Quantum Computing Functions (QCF) for Matlab

Charles Fox
Robotics Research Group
Oxford University

Quantum computing uses unitary operators acting on discrete state vectors. Matlab is a well known (classical) matrix computing environment, which makes it well suited for simulating quantum algorithms. The QCF library extends Matlab by adding functions to represent and visualize common quantum operations. This paper presents a brief overview of QCF, then shows how it can be used to simulate the well-known algorithms of Deutsch, Deutch-Jozsa, and Grover.

Basic notation

There are three basic quantum state notations that are frequently used in the literature: integer kets, binary kets, and vectors. For example, the following are all representations of the same state (in a 3-qubit space):

|3> |011> $[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ ]^T$

The simulator uses vectors as its primary internal representation, but provides functions to convert between notations:

phi = bin2vec(bin)
Converts a single binary string state representation to a state vector.

phi=dec2vec(dec,n)
Convert single decimal state representation to state vector.
n is number of qubits in the vector space.

str = pretty(psi, [bin])
Gives a pretty-printed ket string for a (possibly superposed) state vector *psi*.
By default the integer ket representation is used. The optional *bin* flag gives the binary version.

For example:

```
>> phi_1=bin2vec('011')

     0
     0
     0
     1
     0
     0
     0
     0

>> phi_2 = dec2vec(5, 3)

     0
     0
     0
     0
     0
     1
     0
     0
```

Note that our vector representation is capable of handling superposed states:

```
>> psi = 1/sqrt(2)*phi_1 + 1/sqrt(2)*phi_2

         0
         0
         0
    0.7071
         0
    0.7071
         0
         0
```

…and so are the pretty-print functions:

```
>> pretty(psi)

0.7071|011> + 0.7071|101>

>> pretty(psi, 1)

0.7071|3> + 0.7071|5>
```

Fourier transforms

The Quantum Fourier Transform (QFT) uses *normalized* basis functions (unlike the classical Discrete Fourier Transform) to represent a discrete state vector:

$$|x\rangle = \frac{1}{\sqrt{N}} \Sigma_{j=1:N-1}\, e^{-2\pi i x j/N}\, |j\rangle$$

As the basis is orthonormal, the QFT projections can be computed by the unitary transform:

$$\mathrm{QFT} = \frac{1}{N} \begin{bmatrix} \omega^0 & \omega^1 & \omega^2 & \omega^3 & . \\ \omega^1 & \omega^2 & \omega^3 & \omega^4 & . \\ \omega^2 & \omega^3 & \omega^4 & \omega^5 & . \\ \omega^3 & \omega^4 & \omega^5 & \omega^6 & \\ \cdots & & & & \cdots \end{bmatrix}$$

... where ω is the Nth root of unity, $e^{2\pi i/N}$.

The QCF library provides a *qft* function which creates QFT matrices for any N:

QFT = qft(d)
Creates QFT matrix with d rows and cols, where $d=2^N$.

Note that the QFT is both unitary and hermetian, so the matrix obtained from the *qft* function can be used to transform in both directions.

In the following example we create a QFT matrix for a 3-qubit system:

```
>> Q = qft(2^3)


  Columns 1 through 4

   0.3536             0.3536             0.3536             0.3536
   0.3536             0.2500 + 0.2500i   0.0000 + 0.3536i  -0.2500 + 0.2500i
   0.3536             0.0000 + 0.3536i  -0.3536 + 0.0000i  -0.0000 - 0.3536i
   0.3536            -0.2500 + 0.2500i  -0.0000 - 0.3536i   0.2500 + 0.2500i
   0.3536            -0.3536 + 0.0000i   0.3536 - 0.0000i  -0.3536 + 0.0000i
   0.3536            -0.2500 - 0.2500i   0.0000 + 0.3536i   0.2500 - 0.2500i
   0.3536            -0.0000 - 0.3536i  -0.3536 + 0.0000i   0.0000 + 0.3536i
   0.3536             0.2500 - 0.2500i  -0.0000 - 0.3536i  -0.2500 - 0.2500i

  Columns 5 through 8

   0.3536             0.3536             0.3536             0.3536
  -0.3536 + 0.0000i  -0.2500 - 0.2500i  -0.0000 - 0.3536i   0.2500 - 0.2500i
   0.3536 - 0.0000i   0.0000 + 0.3536i  -0.3536 + 0.0000i  -0.0000 - 0.3536i
  -0.3536 + 0.0000i   0.2500 - 0.2500i   0.0000 + 0.3536i  -0.2500 - 0.2500i
   0.3536 - 0.0000i  -0.3536 + 0.0000i   0.3536 - 0.0000i  -0.3536 + 0.0000i
  -0.3536 + 0.0000i   0.2500 + 0.2500i  -0.0000 - 0.3536i  -0.2500 + 0.2500i
   0.3536 - 0.0000i  -0.0000 - 0.3536i  -0.3536 + 0.0000i   0.0000 + 0.3536i
  -0.3536 + 0.0000i  -0.2500 + 0.2500i   0.0000 + 0.3536i   0.2500 + 0.2500i
```

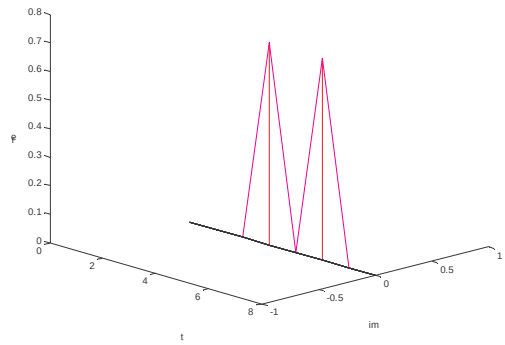
We now apply the QFT to the *psi* state from the earlier example:

```
>> pretty(psi, 1)

0.7071|3> + 0.7071|5>


>> QFT*psi

   0.5000
  -0.3536 + 0.0000i
   0.0000
   0.3536 - 0.0000i
  -0.5000 + 0.0000i
   0.3536 - 0.0000i
  -0.0000
  -0.3536 + 0.0000i

>> pretty(QFT*psi)

0.5|000> + -0.35355+1.3878e-016i|001> + 1.1102e-016|010> + 0.35355-4.7184e-016i|011> + -
0.5+8.8817e-016i|100> + 0.35355-7.7716e-016i|101> + -3.3306e-016|110> + -0.35355+1.0825e-
015i|111>
```

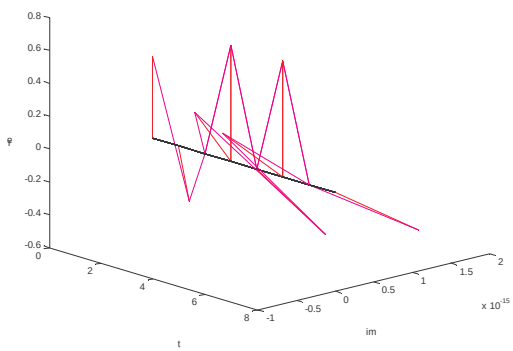
Our QCF library includes the function *iplot* for visualizing complex vectors. In the example below, we use *iplot* to look at the effect of the QFT on *psi*:

> **iplot(psi)**
> Display a 3D complex plane graph of the complex vector *psi*.
> The elements of *psi* are shown as dark spikes and are connected together by a lighter line.

```
>> iplot(psi)
```




```
>> iplot(QFT*psi)
```



Unitary matrix representation of functions

It is often useful to use unitary matrices to represent functions. In this representation, the functions can be performed 'in parallel' on superpositions of inputs. We define the matrix U_f by:

$$U_f\,(\,|x\rangle \otimes |0\rangle^{(n)}\,) = |x\rangle \otimes |f(x \bmod n)\rangle$$

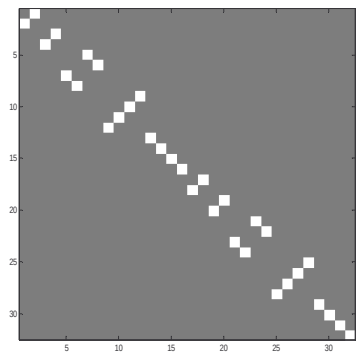
Where there are m bits in the input, x, and n bits in the output. Note that the action of U_f is to replace the zero vector with the value of $f(x)$, whilst leaving the x in tact.

The QCF library function *uf* provides a quick way to construct such unitary matrices. To use it, we define an ordinary function *f*, and pass it as an argument to the *uf* constructor, along with specifications of *n* and *m*. This is necessary because Matlab functions operate on integers, not binary strings – so *uf* must be told how many bits to make space for.

U_f = uf(f,m,n)
Make unitary U_f from function *f*.
f must be of the form $f(x,n)$ where x is a bitstring, and m and n is numbers of input and output bits.

Here we define a simple function, *y*, which returns the modulus of *x*+1. We then display a visualization of the unitary U_y which represents it:

```
function y = f(x,n)
y=mod(x+1, 2^n);

>> U_f = uf('f', 3, 2);
>> qimage(U_f)
```



We now illustrate how the U_f matrix can be used. We form the state $|psi\rangle \otimes |0\rangle^{(2)}$ using the *kron* function for the tensor product. Applying U_f to the result simultaneously computes the superposed results of *f* for the two superposed states comprising $|psi\rangle$.

```
>> pretty(psi)
0.7071|011> + 0.7071|101>

>> pretty(bin2vec('00'))
1|00>

>> pretty(kron(psi, bin2vec('00')))
0.7071|01100> + 0.7071|10100>

>> pretty(U_f * kron(psi, bin2vec('00')))
0.7071|01100> + 0.7071|10110>
```

Deutsch's algorithm

Deutsch's was one of the first quantum algorithms to demonstrate an effect unobtainable by classical computation. Suppose we have four functions: c_0, c_1, b_0 and b_1 as given below:

Input:	0	1
c_0	0	0
c_1	1	1
b_0	0	1
b_1	1	0

The *c* functions are said to be 'constant' and the *b* functions are 'balanced'. Given a black-box which computes U_f, where f is a function chosen from the above set, the task is to determine whether the function represented by U_f is constant or balanced. A classical machine would require two evaluations of the black-box. But Deutsch's algorithm provides a way to do it in just one evaluation, on a quantum computer. The algorithm makes use the Hadamard transform, implemented in QCF by:

H = hadamard(n)
Returns the *n*-qubit Hadamard matrix.

We also use the *measure* function to collapse the final superposition into a single state:

phi = measure(psi)
Measure psi with respect to the standard basis.
(To perform other measurements, transform *psi* and *phi* to and from the required basis.)

The code below implements Deutsch for $f=b_1$. The panel on the right shows the variables during the execution. The results show the values of *psi* at the end of the algorithm, for the four black-box functions. Looking at the first qubit of the final states tells us whether the function is constant (0) or balanced (1).

```
function deutch
psi = bin2vec('01')
U_f = uf('f_b1', 1, 1)
H = hadamard(2)
psi = H*U_f*H*psi
psi = measure(psi)


function y = f_c0(x,n)
y=0;

function y = f_c1(x,n)
y=1;

function y = f_b0(x,n)
y=x;

function y = f_b1(x,n)
y=~x;
```

Results:

function		final *psi*
b_0	à	|11>
b_1	à	|11>
c_0	à	|01>
c_1	à	|01>

psi (initial value) =
0
1
0
0

U_f =

0	1	0	0
1	0	0	0
0	0	1	0
0	0	0	1

H =

0.5000	0.5000	0.5000	0.5000
0.5000	-0.5000	0.5000	-0.5000
0.5000	0.5000	-0.5000	-0.5000
0.5000	-0.5000	-0.5000	0.5000

psi (just before measurement) =
0
0
0
-1.0000

psi (after measurement) =
0
0
0
1

The Deutsch -Jozsa Promise Problem

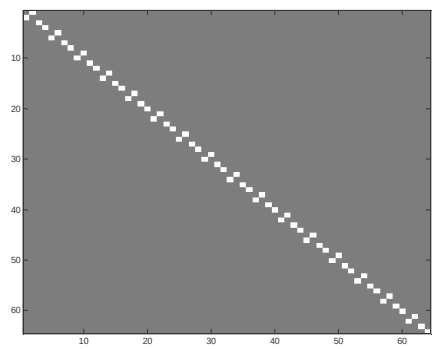
This is a generalization of Deutsch's algorithm to cases where the inputs to the constant/balanced functions have multiple bits. The functions output either 0 or 1, and are 'promised' to be either constant or balanced. Constant now means that the output is the same for any possible input. Balanced means that there are equal numbers of inputs which output 0 as output 1. As with the original Deutsch algorithm, the Deutsch-Jozsa problem can be solved with just one black box evaluation of the U_f.

Matlab/QCF code for Deutsch-Jozsa and some example test functions are shown below. Note the *deutche_joza* function has been defined so that a test function is passed in as an argument. The final, measured, *psi* state is interpreted by looking at the first *N*-1 qubits. If they are all zero then the function is constant. Otherwise, it is balanced. The images on the right are visualizations of two of the U_f matrices.

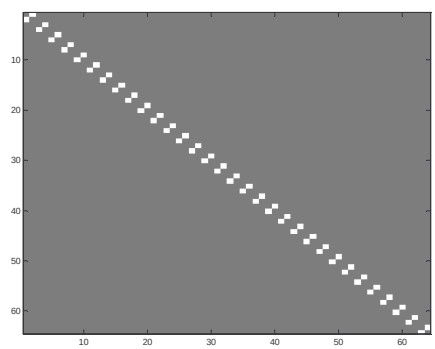
```
function deutsch_jozsa(f)
psi = bin2vec('000001');
U_f = uf(f, 5, 1);
H_6 = hadamard(6);
H_5 = hadamard(5);
I   = identity(1);
psi = (kron(H_5, I))*U_f*H_6*psi;
psi = measure(psi);
pretty(psi)


%the test functions:

function y = c_0(x,n)
y=1;
function y = c_1(x,n)
y=0;
function y = b_0(x,n)
y=x;
function y = b_1(x,n)
y=~x;
```

```
>> deutch_joza('c_0')
 1|000000>

>> deutch_joza('c_1')
1|000001>

>> deutch_joza('b_1')
1|000010>

>> deutch_joza('b_0')
1|000011>
```

By looking a the first four qubits, we see that
So c_0 and c_1 are constant, b_0 and b_1 are balanced.



`U_f` for `b_0`



`U_f` for `c_0`

Grover's Algorithm

Grover demonstrated that quantum computers can perform database search faster than their classical counterparts. In this simple example of Grover's algorithm, we use a function *haystack* to represent the database. We are searching for a 'needle in the haystack', i.e. there is one element of the database that we require. The *haystack* function returns 1 if queried with the correct 'needle' element, and 0 for all the other elements. In this example, *haystack* is defined for a 5-qubit input, and returns 1 for element 8 ('00100').

Grover makes use of the fact that for any function *f* with a binary string input and a Boolean output, we can define a corresponding unitary matrix V_f, whose action on input strings |*psi*> is to invert them if and only if f(*psi*)=1. (Otherwise, it does nothing). The QCF library provides a function to create such matrices:

> **V_f = vf(f, n)**
> Build unitary *f*-conditional inverter for *n* bit input such that:
> $V_f|psi\rangle = (-1)^{f(psi)}|psi\rangle$

Additionally, an 'inversion about the average' matrix, D, is used. The action of D on each element a_i of a state vector *psi* is to replace it with $\text{mean}_j(a_j)-a_i$. Again, QCF has a function to build D for a specified state vector size.

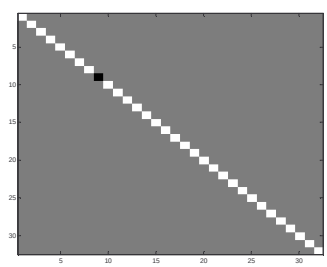
> **D = ia(n)**
> Create 'inversion about average' matrix for *n*-qubit state vectors.

Grover's algorithms works by iteratively applying $V_{haystack}$ and the inversion about the average operator to the current state. Each iteration amplifies the probability of a measurement collapsing the state to the correct 'needle' value.
Grover showed that performing a measurement after $\sqrt{(2^n)}*(\pi/4)$ iterations is highly likely to give the correct result.

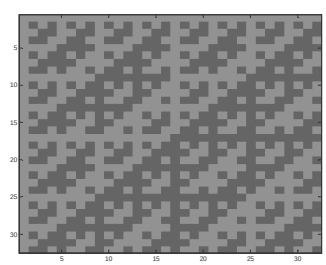
The code for Grover's algorithm is given below.
On the right are visualizations of the matrices used.
The 3D graph on the next page shows the amplitudes of the different 'needle' candidates. Note that the amplitude for 8 (the correct needle value) reaches its peak at about $\sqrt{(2^n)}*(\pi/4)$ iterations, as predicted.

```
function grover

n=5;

V_haystack  = vf('haystack', n);
H             = hadamard(n);
D             = ia(n);

phi = bin2vec('00000');

phi = H*phi;

maxiter = (pi/4)*sqrt(2^n);

for i=1:10

    phi=V_haystack*phi;
    phi=D*phi;

end

phi=measure(phi);
```

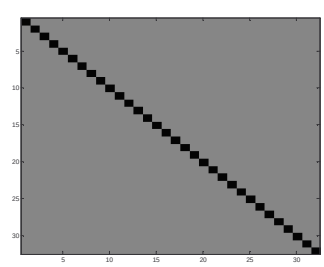
(black=-1; gray=0; white=1+)



$V_{haystack}$

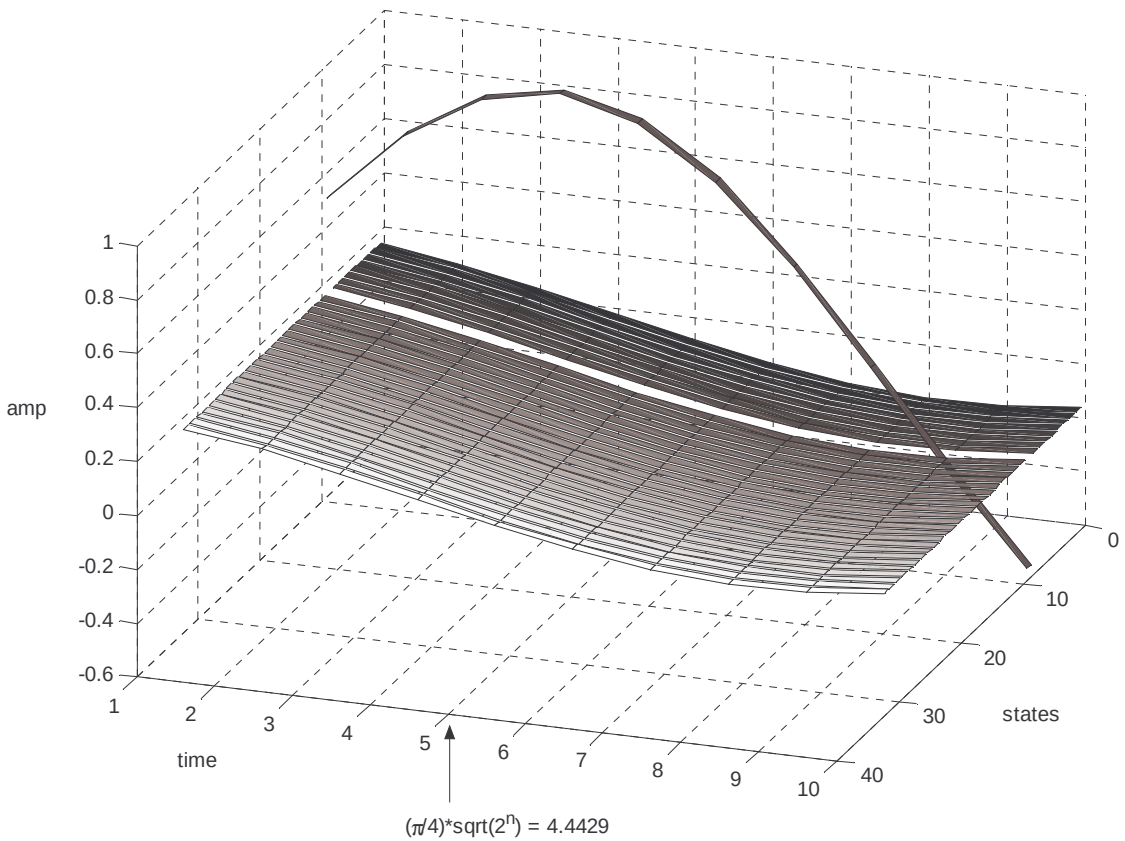


H



D

Amplitudes for needle values during Grover's algorithm iterations:



QCF API Summary

phi = bin2vec(bin)
Converts a single binary string state representation to a state vector.

phi=dec2vec(dec,n)
Convert single decimal state representation to state vector.
n is number of qubits in the vector space.

str = pretty(psi, [bin])
Gives a pretty-printed ket string for a (possibly superposed) state vector *psi*.
By default the integer ket representation is used. The optional *bin* flag gives the binary version.

QFT = qft(d)
Creates QFT matrix with *d* rows and cols, where $d=2^N$.

iplot(psi)
Display a 3D complex plane graph of the complex vector *psi*.
The elements of *psi* are shown as dark spikes and are connected together by a lighter line.

H = hadamard(n)
Returns the *n*-qubit Hadamard matrix.

phi = measure(psi)
Measure psi with respect to the standard basis.
(To perform other measurements, transform *psi* and *phi* to and from the required basis.)

V_f = vf(f, n)
Build unitary *f*-conditional inverter for *n* bit input such that:
$V_f|psi\rangle = (-1)^{f(psi)}|psi\rangle$

D = ia(n)
Create 'inversion about average' matrix for *n*-qubit state vectors.

Consciousness and the Pseudonoumenal

Charles Fox
Robotics Research Group
University of Oxford
charles@robots.ox.ac.uk

Abstract

We define the pseudonoumenal world as our best current phenomenal model of the unknowable noumenal world (cf. Kant, 1781). We argue that unlike other scientific objects, consciousness must logically be noumenal. Noumena are unknowable but pragmatically we should believe our best pseudonoumenal model to be true. Consciousness may therefore be identified with an element of our pseudonoumenal model. Parsimony suggests that identity with an already-postulated pseudonoumenon (property dualism) is more likely than an identity with a newly-postulated pseudonoumenon (substance dualism) if and only if we can empirically observe a correlation. This lends philosophical support to pseudonoumenal identity theories such as Orch-OR (Hameroff & Penrose, 1996) and physical information identities.

This is a condensed presentation of (Fox, 2001).

Consciousness

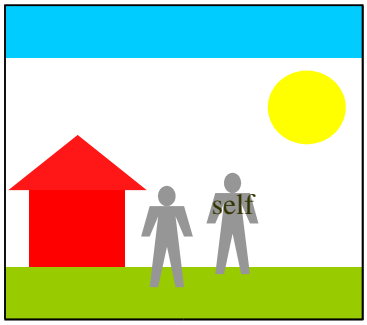
I define 'my consciousness' as the first-person feeling that is common to the experiences that I have when I am awake or dreaming. I assume that others have similar feelings so are able to understand this definition. Once one possesses the concept 'my consciousness', one may then draw an analogy between oneself and others in order to construct concepts of their consciousness. By generalizing all these concepts of other people's consciousnesses, we define the concept 'consciousness'.

This definition is based *purely* on first-person experience. Consciousness, under this definition, has *no function*. Empirically, consciousness may turn out to correlate with functions, or with behaviors or with objects, but that is incidental to the definition.

Phenomena

The experience of consciousness is that of a self and other objects in a world, e.g. as

shown below. These objects are *phenomena* and the world is one's *phenomenal world*. Different people may have different phenomenal worlds and construct different phenomenal objects in them.

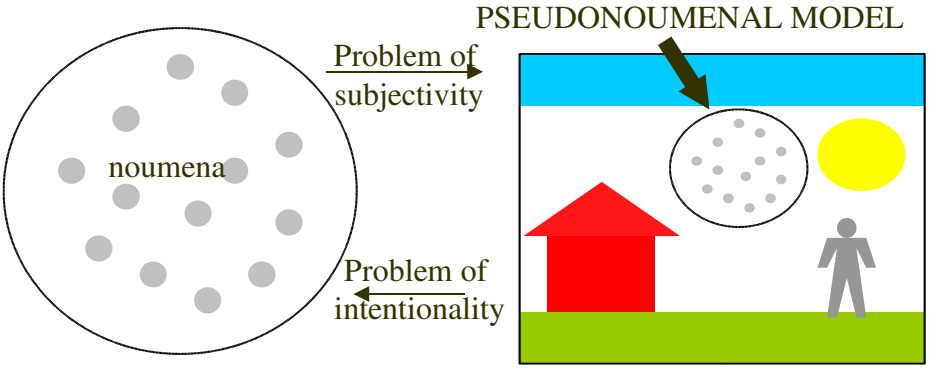


PHENOMENAL WORLD

We construct phenomena that are useful to us. Examples of phenomena include houses, cups, tables, molecules, atoms, love and the self. Everything we perceive is a phenomenon.

Noumena and phenomenal pseudonoumena

In addition to our individual phenomenal worlds, we find it useful to believe in the existence of an observer-independent world called the *noumenal* world, populated by entities called noumena. By definition, noumena are not perceivable as they exist outside of any phenomenology. Pragmatically, at any point in time we have a best phenomenal model of what we think is noumenal. We call such models *pseudonoumenal* models. Historical pseudonoumenal models have included the four Greek elements, atoms, the Mind of God, quarks, the Orch-OR process and superstrings.

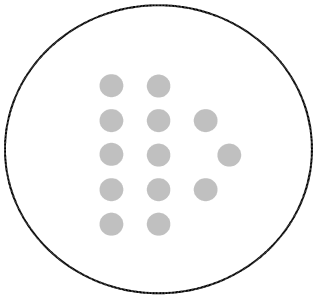


NOUMENAL WORLD PHENOMENAL WORLD

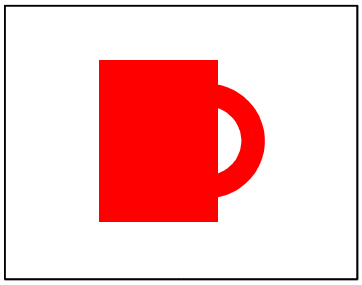
Building on (Velmans, 2000) the above figure shows the relationships between the noumenal, phenomenal and pseudonoumenal worlds. The noumenal world exists independently of any observer. A phenomenal world exists in the mind of an observer, and may contain a pseudonoumenal model of the noumenal world.

Compound objects are phenomenal

Roughly, the noumenal world is the set of all entities which are *fundamental*. For example, cups do not noumenally exist. We can perceive a cup as a single phenomenal object, but we know that noumenally, there is nothing more to the cup than its noumenal parts. If we think, for example, that quarks are fundamental, then we accept that were are noumenally just looking at a bunch of quarks – there is no additional cup entity in the noumenal world. The cup object exists only phenomenally:



NOUMENAL WORLD



PHENOMENAL WORLD

Anything which is not fundamental does not exist noumenally. For, like the cup, if there is a way of breaking it into subcomponents, then we see that nothing more than the subcomponents is needed in the noumenal world. We know that phenomena that are are not pseudonoumenal are just useful fictions. Cups, handles, brains, neurons, functions and behaviours are all useful fictions – we construct them as phenomena because they enable us to make useful predictions.

Consciousness is noumenal

Unlike cups, brains and functions, **consciousness noumenally exists**. It is not just a phenomenal predictive fiction. Like Descartes, I know that it exists non-fictitiously because I *am* it. **I am not a fiction!** (Furthermore, even if I was a fiction, consciousness could still not be phenomenal, because phenomena exist *inside* consciousness, and an entity logically cannot be identical to something inside itself.) Consciousness is therefore a noumenon – the *only* noumenon that we know directly.

Pseudonoumenal identity

The 'hard problem' (Chalmers, 1995) of subjectivity is to account for the existence of the phenomenal world *within* the noumenal world. (Conversely, the problem of 'intentionality' is sometimes conceived of as accounting for referential links the other way.) As we may never know the noumenal directly, the hard problem in practice is to account for the existence of the phenomenal world within the *pseudo*noumenal model. We must have something in the pseudonoumenal world which is identical to consciousness, and the goal of the science of consciousness is to find that identity.

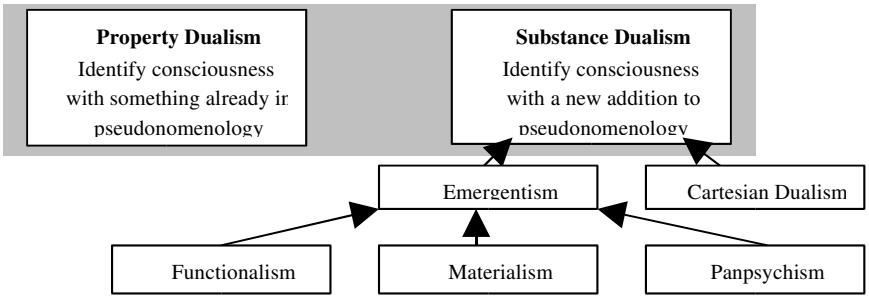
There are two ways to make an identity:

1. Property dualism: Identify consciousness with an existing pseudonoumenon
2. Substance dualism: Postulate the existence of a new pseudonoumenon, identify consciousness with it, and provide bridging laws to connect existing pseudonoumena to it.

As in any other science, parsimony favors the former if and only if an empirical correlation can be found to base the identity upon. The content of contemporary pseudonoumenology is debatable but for example, the Orch-OR model hypothesizes a correlation between a pseudonoumenon (objective state reduction) and consciousness, which may be testable using anaesthetic experiments. If such a correlation could be found, this would provide a highly plausible pseudonoumenal identity.

Reducing the 'isms'

We now demonstrate how some popular identity positions all reduce to substance dualism in the pseudonoumenalist view. Whilst this does not rule out these theories, it renders them less parsimonious than a well-correlated property dualist theory, because they need to postulate additional entities in the pseudonoumenal world.



Recall that we cannot identify consciousness with a phenomenon because phenomena are fictions and I am not a fiction.

Emergentism is the view that 'something extra' comes into existence when a group of parts are brought together, and that something is identified with consciousness. However emergent entities are only phenomena, not noumena: for example the body and the handle of a cup are perceived phenomenally as a new cup object. There is no new observer-independent noumenon that appears – unless one constructs 'strong emergent' laws of physics to postulate the creation of such a noumenon at certain conditions. However this would be a new substance and so strong emergentism reduces to substance dualism.

Similarly, *macroscopic materialist* theories (e.g. identifying consciousness with whole brains, neurons, or coffee cups) are incoherent to the pseudonoumenalist because they try to identity a fictional phenomenon with a noumenon. (Recall, consciousness is noumenal because I exist and I am not a fiction.) Such theories can be rescued by extending them into strong emergent theories (e.g. whenever a brain exists in such a configuration, then a new physical law brings a dualist substance into noumenal existence) but that leads back to substance dualism.

Panpsychism claims that some or all groups of noumena are consciousness. However the concept of 'group of noumena' is a phenomenal construct (otherwise the 'group' part would be redundant as the group would be noumenal already). Again panpsychism can be rescued by postulating new noumena corresponding to such groups, and again this leads back to substance dualism.

Pseudonoumenalism and the 'Hard Problem'

Pseudonoumenalism accepts that we can never make completely certain links between the noumenal and phenomenal worlds. However it replaces that task with the easier task of making an identity *within* the phenomenal world: between a pseudonoumenon and the presence of the phenomenal world itself. This is no worse than identities achieved by other sciences, for example, the pseudonoumenal identity of inertial and relativistic mass. As in any science, it is possible for the pseudonoumenal model and identities to be falsified and improved; but as in any other science we should pragmatically believe in

our current best theory. For example, if experiments with anaesthetics suggest an identity between a pseudonoumenon and consciousness, then we should have confidence in using new anaesthetics designed on the basis of that theory. The pseudonoumenalist is as happy to use consciousness-affecting technology – based on his best model of the pseudonoumenal – as he is to use aeroplanes based on his best theory of aeronautics.

Acknowledgments

Thanks to Paul Schweitzer, Stuart Hameroff, David Chalmers, Douglas Hofstadter and the FARG group for discussions.

References

Chalmers, D. (1995). *J. Consciousness Studies* 2:200-219.

Fox, C. (2001). *MSc Thesis, Edinburgh University.* www.inf.ed.ac.uk/publications

Hameroff, S. and Penrose, R. (1996). *J. Consciousness Studies* 3:36-53.

Kant, I. (1781). C*ritique of Pure Reason*.

Velmans, M. (2000). *Understanding Consciousness*. Routledge.

Genetic Hierarchical Music Structures

Charles Fox
Robotics Research Group
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, UK
charles@robots.ox.ac.uk

Abstract

Music has structure at many levels, from grand arrangements of verses and choruses down to patterns in small riffs and themes. A bracketed L-system — the SARAH language — is used to represent compositions in terms of these hierarchical structures, allowing genetic programming to meaningfully mutate and crossbred compositions at all levels of structure. Automated composition becomes possible, using evolution with human aesthetic judgment as a fitness function. SARAH is also human-readable and can be used as a human tool for rapid structural composition development; or as a semi-automated composition system, mixing human and evolved contributions. The system has bred pleasing compositions starting from basic musical materials and has also been used to crossbreed Bach with the Spice Girls: these examples are presented as audio files.

Introduction

The majority of pop songs share an identical top-level structure (or simple variants of it):

```
POP-SONG => {INTRO, VERSE, CHORUS,
VERSE, CHORUS, MIDDLE 8, CHORUS, OUTRO}
```

They also contain similar structure at lower levels. A `VERSE` may be decomposable as follows:

```
VERSE=>{RIFF-A,RIFF-B,RIFF-A,RIFF-B}
```

And a `RIFF` may be broken down into parts for different instruments (played simultaneously, not sequentially; we notate this by using square brackets instead of braces):

```
RIFF=>[VOCALS,BASS,GUITAR]
```

Structure is present even at the level of individual notes:

```
VOCAL-TUNE=>{G,G,A,A,C#,A,F#,G}
```

Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

We could imagine describing a complete composition using such rules. When considered as a generative system (i.e. when used in conjunction with a compiler that creates actual musical output from the rules) they form an L-system (Prusinkiewicz & Lindenmayer 1990). Further, a *bracketed* L-system allows us to base all the rules on a single musical terminal. We use the terminal `NOTE` to represent a single whole-note tonic in C Major (i.e. middle C), played at medium volume on MIDI channel 1. A bracketed L-system is one which allows transforms in the rule definitions, for example:

```
SONG=>{VERSE,(^3)VERSE,(^5)(~2)(R)VERSE}
```

This describes a song consisting of a verse; then a copy of the verse raised my a major third; then another copy raised by a perfect fifth, timestreched by a factor of 2, and retrograded (played backwards). A simple melody fragment can be represented by transformed `NOTE`s:

```
TWINKLE =>{NOTE,NOTE,(^5)NOTE,(^5)NOTE,
(^6)NOTE,(^6)NOTE,(~2)(^5)NOTE}
```

Holtzman's GCDL language (Holtzman 1980) provided a notation for human composers to rapidly develop structured compositions by specifying production rules similar to those shown above. GCDL is a powerful language, allowing complex function definitions and computations. The present work provides a simpler bracketed L-system notation called SARAH, consisting only of term rewriting with transforms, and using a single terminal. Typically, a SARAH representation of a composition is longer than a GCDL program, requiring more rules and more transforms to compensate for the simpler language. The advantage of this representation (aside from its faster learning curve for human composers) is that simple rules are amenable to be operated on by Genetic Programming (Koza 1992), enabling SARAH to be used for automated and semi-automated composition as well as human composition.

The next section describes SARAH in detail. We then discuss its use as an aid to human composition; then as a tool for semi-automated evolutionary composition. Finally we touch on some cultural and philosophical issues raised by this mode of composition.

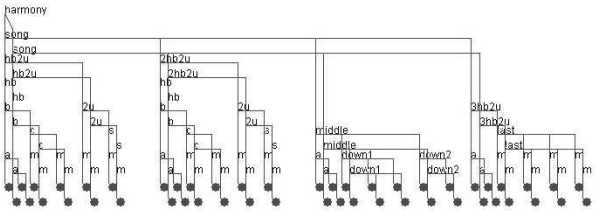


Figure 1: The compiled tree for Happy Birthday

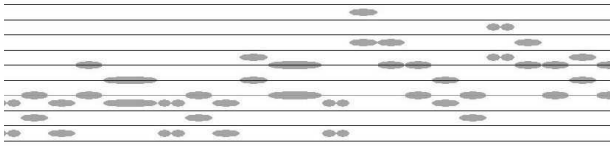


Figure 2: The compiled score for Happy Birthday

The SARAH Language

A composition is represented in SARAH by a list of context-free term-rewrite rules. The right-hand sides of the rules are enclosed in braces or square brackets to represent whether the structure is to be played melodically (in series) or harmonically (in parallel) respectively. Within these brackets is a list of structure names. The names must refer to structures that are previously defined in the list, or may be the terminal `NOTE`. Each name may be prefixed by a list of transforms. For example, here is Happy Birthday written in SARAH:

```
a => {NOTE, NOTE}
m => {(~2)NOTE}
down1 => {(^5)NOTE, (^3)NOTE, NOTE}
down2 => {(^-2)NOTE, (^-3)NOTE}
s => {(~2)m}
c => {(^2)m, m}
b => {a, c}
2u => {m, (^-2)s}
hb => {(^-4)b}
last => {(^3)m, m, (^2)m, m}
hb2u => {hb, 2u}
2hb2u => {hb, (^2)2u}
middle => {(^-4)a, (~2)down1, (~2)down2}
3hb2u => {(^4)a, last}
song => {hb2u, 2hb2u, middle, 3hb2u}
harmony => [song, (^3)song]
```

This generates the tree shown in fig. 1. Applying the transforms to the leaves gives the melody shown in fig. 2. Note that there is no nesting of brackets or transforms *within* each rule, as is generally allowed in bracketed L-systems.

The possible transforms are:

- `(^n)` shift pitch by *n* scale degrees
- `(&n)` shift key by *n* scale degrees
- `(~n/m)` timestretch by rational *n/m*
- `(n/m)` multiply amplitude by rational *n/m*
- `(R)` retrograde (play backwards)
- `(I)` inversion (reflect pitches about the tonic)

We use a *scale-degree* representation of pitch. Each note is represented by a 6-tuple: (key, degree, time, duration, volume, channel). Key is an integer MIDI note number which represents the root note of the (relative) Ionian major scale that the note is in. For example, for a piece in C major, D Dorian or A Aeolian it could be 60 (C3/middle C). For a piece in D Major, E Dorian or B Aeolian it could be 62 (D3). The pitch is then determined by the scale degree within that key. So in key 60 (C3 Ionian), a note of degree 3 would be the major third of the scale rooted on C3, i.e. E3. A note of degree 10 would be major tenth, i.e. E4. At present the scale pattern is fixed to the Ionian scale, allowing melodies in standard western modes to be represented without changing key. To represent accidentals, one must make a key change as well as a degree shift. For example, the following shows how to notate the melody A3,F3,B3 in G Major:

```
TUNE=>{(^2)NOTE,(&-5)(^4)NOTE,(^3)NOTE}
SONG=>{(&5)TUNE}
```

Recall that each `NOTE` starts out as the tonic in C3 Major. So `SONG` shifts the key of `TUNE` up a fifth to G3 Major. `TUNE` itself defines the melody, the second note of which is the fourth degree of a new key - changed down a fifth back to C Major. The accidental is thus conceived of as occurring in a temporary mode change.

We use the scale-degree representation for two related reasons. First, we find it more cognitively useful for human composers to think in terms of scale degrees rather than absolute pitch. It provides a higher level of abstraction for the composer to make use of: once she has changed to F minor she can then think and write in terms of that scale rather than have to worry about how to translate from that scale into absolute pitches. Secondly, we find it gives rise to much more aesthetic compositions from the genetic composition system discussed later. This is because it allows the system to make musical changes at this cognitively meaningful layer. So for example, it may change the key or mode of a whole section of music with a single mutation; or shift the degrees of random notes *within* their home scale.

Human Composition

We have created a GUI environment for human composers to quickly edit, compile and view SARAH compositions (fig. 3). The user can examine the tree and score representations of the complete score or sub-scores, and may also force particular types of genetic operations to occur. This environment is called *Music Genie*. Audio example 1 presents a short minimalist-style composition written by a human using Music Genie. The composer commented that he was able to develop this composition much faster than by hand due to his enhanced abilities to replicate and transform existing structures.

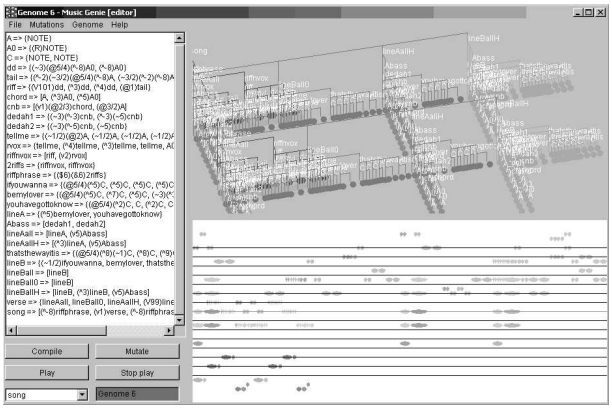


Figure 3: Music Genie's composition environment

Evolution

Following Koza's conception of Genetic Programming (Koza 1992), we consider consider each rule in a SARAH composition to be a gene, and the complete composition as a genome. As with Genetic Algorithms, Genetic Programming requires four operations to be specified: mutation methods, crossover methods, a fitness function and an initial population.

Mutation

We use the following mutation types: Random transforms may be inserted into genes. Existing transforms may have their numeric arguments altered. Transforms may be deleted. Random structure names may be inserted into genes (subject to their structures being defined higher in the genome). Structure names may be removed from genes. Melodic and harmonic structures may be flipped (i.e. the square brackets replaced by braces and vice versa). New random genes may be added to the genome. Genes may be deleted from the genome (subject to all references to them being removed also). Each of these mutation types has a probability, set by hand.

Crossover

Crossover is more complex. We would like to combine two different compositions in a musically meaningful way. Because our genes represent hierarchical structures, and we are using the scale-degree representation, it is possible to do this in an elegant way. To create a child from some parents, we choose random genes from each parent and put them together to form a new genome. When a selected gene from parent *p* refers to other structures from *p* whose genes were also selected, then their names are preserved in the child. When it refers to structures whose genes were not selected, we must change the names to refer to an 'analogous' structure in the child. Choosing the 'most analogous' structure is a difficult problem: analogy-making is an active area of research (Hofstadter 1996), (Gentner, Holyoak, & Kokinov 2002) and we do not yet know the best way to go about it. At present we use a simple heuristic. Recall that the genomes are *ordered* lists. For example, genes may only refer to names of genes defined earlier than themselves in the genome (to prevent circular definitions). This ordering gives an admittedly crude heuristic for analogy-making. Call each genes position in the genome its *level*. To perform crossover, use the following algorithm, whose action is illustrated in fig. 4:

for each child c **do**
  for each level i **do**
    $p \leftarrow$ randomly chosen parent
    $g \leftarrow i^{th}$ gene of p
    copy g into c's genome
    for each name n on the RHS of g **do**
      $gr \leftarrow$ the gene in p with name n
      if gr was previously copied to c **then**
        do nothing
      else
        $j \leftarrow$ the level of gr in p
        $p_{old} \leftarrow$ the parent whose level j gene was copied to c
        $n_{old} \leftarrow$ name of the level j gene in p_{old}
        in c's copy of g, substitute n_{old} for n
      end if
    end for
  end for
end for

Fitness Function

Whilst it may be possible to automate some human aesthetic preferences and encode a fitness function based on them, we take a human-driven approach to evolution. Music Genie includes a GUI (fig. 5) that presents users with a current population of compositions and asks them to select their favorites. These selections are used as parents to breed the next generation. We do however use a single, very basic heuristic that screens out compositions containing less than 5 notes. This was added because the heuristic cross-over system described above can sometimes lead to badly-drawn analogies which produce very short compositions. An interesting open question is to what extent automated fitness functions are possible: could and should we for example screen out compositions that change key too often, or whose rhythmic structures are irregular? There is of course a trade-off here: restricting the space of possible compositions in exchange for requiring the user to spend less time selecting out 'obviously' bad ones.

We have also implemented a web-based client-server version of this GUI that allows many users to collaborate on evolving compositions.

Initial Population

The evolution system may be initialized by loading a set of human compositions as the initial parents; alternatively, single `NOTE` terminals could be used, so that the compositions grow entirely from evolved material. In practice we prefer the first option as it would take many generations of uninteresting compositions before the second method produced interesting results. Cross-over and mutation may also be

```
(1) Align parents:
Parent1: RIFF     =>  NOTE                          Parent2: TUNE1 => NOTE
         CHORUS   => {(^5)RIFF}                              A       => {TUNE1, (&5)TUNE1}
         VERSE    => {RIFF, (^3)RIFF}                        B       => {B, (^5)B}
         SONG     => {VERSE, (&2)VERSE,CHORUS}               RONDO   => {A,B,A,B,A}
(2)Convert to standard form and select genes:
Parent1: 1        =>  NOTE                          Parent2: 1        => NOTE
         2        => {(^5)1}                                 2        => {1, (&5)1}
         3        => {1, (^3)1}                              3        => {3, (^5)3}
         4        => {3, (&2)3,2}                            4        => {2,3,2,3,2}
(3) Create child from selected genes:                (4) Rename with original string names:
        1         =>  NOTE                                   RIFF     =>  NOTE
        2         => {1, (&5)1}                              A        => {TUNE1, (&5)TUNE1}
        3         => {3, (^5)3}                              B        => {B, (^5)B}
        4         => {3, (&2)3,2}                            SONG     => {VERSE, (&2)VERSE,CHORUS}
```

Figure 4: Example of the crossover process.

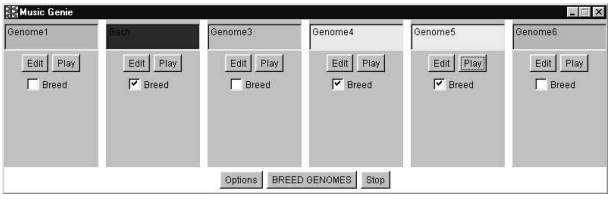


Figure 5: Music Genie's evolution manager

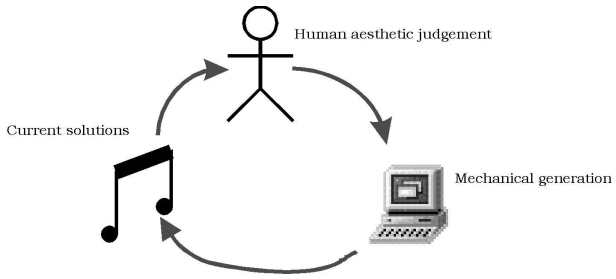


Figure 6: The creative process for semi-automated composition

run manually from the human composition environment in Music Genie, allowing a human composer to generate new works by controlled evolution of his existing works.

Results

Audio example 2 presents a typical example of a user working with the web-based selection system to evolve a new composition. Audio example 3 presents the result of such an evolution after about 100 generations. Audio example 4 illustrates the effects of various random mutations on an extract from Bach's *Well Tempered Clavier*, and the effect of crossbreeding it with an extract from the Spice Girls song *Wannabe*. Most listeners find these compositions to be reasonably aesthetically pleasing.

Discussion

Note that at present, the input compositions must be manually parsed into SARAH, by musically-trained humans. This is time consuming and appears to require considerable intelligence and pattern-recognition skills. Future work could examine the use of grammar induction methods to automate some or all of this process. See for example (Reis 1999) for an agent-based discussion of this area.

We present SARAH and Music Genie not only as a useful tool for human composers and a fun way of evolving short compositions, but as raising questions about the nature of creativity: to what extent should creativity be credited to the machine and to the human(s) who participate in evolutions?

A criticism of machine composition (e.g. (Cope 2001)) is that it cannot currently be called 'artistic' because it does not deliberately *express* anything: it does not start with a conceptual model of emotions or stories to be conveyed by the language of music. If a machine was to produce Beethoven's Ninth by chance, it would still not be artistic unless it had started with a model of what it is like to be hurtling through a thunderstorm in the rain. Rather, this creative credit should be granted to the human *curator* who selects that chance composition — probably after considerable effort spent listening to millions of others — and perceives it as expressing those concepts. The chance composition acts like a 'found object'. Hence the composer becomes synonymous with the curator.

Music Genie suggests a further shift in this direction, from the curator-as-composer to the listener-as-composer (fig. 6). Reminiscient of Turing's notion of mathematical creativity (Turing 1938), listener-composers inject emotion and meaning to the creative process, but the laborious work of creating the actual structures and notes is performed by the machine. The composition could exist as a permanently-evolving population, and every listening acts as a further step in the evolution. The credit for composition should then be assigned to the community of listeners themselves rather than an *èlite* creator or curator.

References

Cope, D., ed. 2001. *Virtual Music: Computer Synthesis of Musical Style*. Cambridge, MA, USA: MIT Press.

Gentner, D.; Holyoak, K. J.; and Kokinov, B. K., eds. 2002. *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA, USA: MIT Press.

Hofstadter, D. R. 1996. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York, NY, USA: Basic Books, Inc.

Holtzman, S. 1980. *Generative Grammars and the Computer Aided Composition of Music*. Ph.D. Dissertation, Edinburgh University.

Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press.

Prusinkiewicz, P., and Lindenmayer, A. 1990. *The algorithmic beauty of plants*. New York, NY, USA: Springer-Verlag New York, Inc.

Reis, B. Y. 1999. A multi-agent system for on-line modeling, parsing and prediction of discrete time series data. In *Intelligent Image Processing, Data Analysis and Information Retrieval*. IOS Press Holland. pp. 164–169.

Turing, A. M. 1938. *Systems of Logic based on Ordinals*. Ph.D. Dissertation, Princeton University.

Appendix

The following is the code for the Bach example:

```
a => {($6)(&5)NOTE}
Bold => {a, (^3)a, (^-4)a, (~2)(^-5b)a,
         (~3)(^-4)a}
Cold => {(~1/2)(^-3)a, (~1/2)(^-2)a, a,
         (~1/2)(^-2)a, (~1/2)(^-3)a, (^-2)a}
Dold => {(^3)Cold, (^6)Cold}
B => {(^5)a, (^5#)(@5/4)a, a,
      (~2)(^-2b)(@5/4)a, (~3)(@5/4)a}
C => {(~1/2)(^2)a, (~1/2)(^3)a, (^4)a,
      (~1/2)(^3)a, (~1/2)(^2)a}
D => {(^3)(~2)a, (~3)a, (~1/2)(^2)a,
      (~1/2)a, (^-2)a, (~1/2)a,
      (~1/2)(^2)a, (^3)a, a,
      (^-3)a, (^4#)a}
E => {(^5)(~3)a, (^6#)a, (^7)a, (^6#)a,
      (^7)a, (^8)a, (^9)a}
t => {B, C, D, E}
tt => {t, (&5)t}
pause => {(@0)(^5)B, (@0)(^5)C, (@0)a}
delayed => {pause, Bold, Cold, Dold}
z => [tt, (v1)delayed]
```

The following is the code for the Spice Girls example:

```
A => {NOTE}
C => {NOTE}
dd => {(~3)(@5/4)(^-8)A, (^-8)A}
tail => {(^-2)(~3/2)(@5/4)(^-8)A,
    (~3/2)(^-2)(^-8)A, (^-2b)(^-8)A}
riff => {dd, (^3)dd, (^4)dd, tail}
chord => [A, (^3)A, (^5)A]
cnb => [(v1)(@2/3)chord, (@3/2)A]
dedah1 => {(~3)(^-4)cnb, (^-3)(~5)cnb}
dedah2 => {(~3)(^-5)cnb, (~5)cnb}
tellme => {(~1/2)(@2)A, (~1/2)A, (~1/2)A,
  (~1/2)A, A, (~1/2)A, (~1/2)A}
rvox => {tellme, tellme, tellme, tellme}
riffnvox => [riff, (v2)rvox]
2riffs => {riffnvox, riffnvox}
riffphrase => {($6)(&6)2riffs}
ifyouwanna => {(@5/4)(^5)C, (^5)C,
               (^5)C, (^5)C}
bemylover => {(@5/4)(^5)C, (^6)C, (^5)C,
              (~3)(^3)C}
youhavegottoknow => {(@5/4)(^2)C, C,
               (^2)C, C, (^3)(~3)C}
lineA => {(~1/2)ifyouwanna, bemylover,
             youhavegottoknow}
Abass => {dedah1, dedah2}
lineAall => [lineA, (v5)Abass]
lineAallH => [lineA, (^3)lineA, (v5)Abass]
thatsthewayitis => {(@5/4)(^8)(~1)C,
 (^8)C, (^9)C, (^5)C, (^6)(~2)C, (^5)(~2)C}
lineB => {(~1/2)ifyouwanna, bemylover,
   thatsthewayitis}
lineBall => [lineB, (v5)Abass]
lineBallH => [lineB, (^3)lineB, (v5)Abass]
verse => {lineAall, lineBall,
          lineAallH, lineBallH}
song => {riffphrase, (v1)(^8)verse,
         riffphrase, (v1)(^8)verse}
```

The MP3 audio examples and the Music Genie software are freely available from `www.charlesfox.org.uk`.

DRUM'N'BAYES: ON-LINE VARIATIONAL INFERENCE FOR BEAT TRACKING AND RHYTHM RECOGNITION

Charles Fox, Iead Rezek, Stephen Roberts
Department of Engineering Science
University of Oxford

ABSTRACT

It is useful for music perception and automated accompaniment systems to perceive a music stream as a series of bars containing beats. We present a proof-of-concept implementation of a Variational Bayesian (VB) system for simultaneous beat tracking and rhythm pattern recognition in the domain of *semi-improvised* music. This is music which consists mostly of known bar-long rhythm patterns in an improvised order, and with occasional unknown patterns. We assume that a lower-level component is available to detect and classify onsets. The system uses Bayesian network fragments representing individual bars and infers beat positions within them. Model posteriors provide principled model competition, and the system may be seen providing a Bayesian rationale for agent-based and blackboard systems. The psychological notion of priming is used to instantiate new candidate models.

1. INTRODUCTION

Perception of beats and bars is difficult because there is scope for much ambiguity. Fig. 1(a) shows an example of *between-bar* ambiguity: a listener familiar with military bands would probably chunk this into 'marching' bars beginning at the points marked '*', but a listener more accustomed to electronic dance music might chunk it into classic 'two-step' drum'n'bass bars beginning at the '†' points. *Within-bar* ambiguity is illustrated in the two bars of fig. 1(b): here the rows represent different onset types; the first bar is a standard 4/4 pattern with the snare on beat 3. The second bar is similar but the snare hit occurs later. This could be interpreted in at least three ways: (1) the pattern could be the same standard 4/4 as in bar one, but with an onset-detection error giving rise to the snare being perceived as late; (2) the pattern may be the same standard 4/4 but the player made a (possibly deliberate) mistake of playing the snare late; (3) it may be a new pattern with the snare on beat 3.5. The difference between (2) and (3) is subtle and even debatable: if the player makes the same 'error' of (2) several times then we begin to perceive it as a new pattern as in (3).

Both types of ambiguity are usually resolved by prior information. The genre of the performance favours certain patterns: a composition is unlikely to switch from military march to drum'n'bass in consecutive bars. We expect – very strongly – that each new pattern will begin at exactly

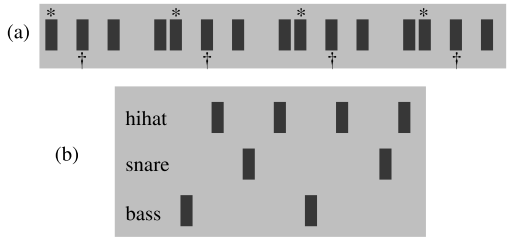


Figure 1. (a) Between-bar and (b) within-bar ambiguity.

the time that the previous pattern ends, and – less strongly – that it will have a similar tempo.

We formalize these ideas using Bayesian network models of single-bar rhythms to infer *within*-bar beat positions, together with a blackboard framework to select *between* these models. The required networks are highly loopy so exact inference is intractable. We have selected the Variational Bayes inference approximation because it seeks an estimate of the joint rather than the marginals, and we are more interested in coherent explanations of whole bars than of individual notes. VB is fast enough to run in real-time, unlike sampling methods which may be very time consuming. VB runs iteratively, so may be terminated early to read off a "best decision so far" which is useful in the real-time setting. Finally, we need to compare rival explanations of bars and VB can provide a computationally cheap lower bound on the model likelihood using byproducts of its parameter computations.

VB is now a standard machine learning technique and we present its application to a novel musical task. We implement the message passing VB computation scheme of [5] and extend it by introducing annealing and undirected sibling rivalry links. We introduce the problem of assigning data points to whole models, and heuristics of priming and pruning for dealing with this.

Real-time music perception systems generally use one of two architectures. *Dynamic state* models (e.g. [2] and the Hidden Markov Model component of [3]) update the state of the music at each point in time (typically each frame of audio). In contrast, *position-in-time* models (e.g. [1] and the Bayesian Network component of [3]) maintain beliefs about the positions of a set of musical events – such as note onset or beat times. Our approach is a position-in-time method, but unlike these previous systems it deals with *semi-improvised* music: rather than require the per-

formance to be aligned with a fixed score, we consider sequences of known (and occasionally unknown) drum patterns played in an improvised sequence. Within each pattern, notes may also be omitted from or inserted into the standard pattern. In particular this means that unlike previous systems, we cannot assume a given mapping from input notes to model notes. We consider multiple rhythm models as hypotheses to explain the data: when the models are viewed as 'agents' this is a similar approach to non-probabilistic agent-based methods such as Goto [6] and Reis [8]. However our models have rigourous probabilistic semantics, for their internal parameters – in the manner of Raphael [3] and Cemgil [1] – but also for their model posteriors. It is the latter that provides a rigourous probabilistic basis for agent competition.

We assume that some lower-level system is available (e.g. [4]) which reports incoming onsets with a belief over their time of occurrence and a discrete classification between bass, snare and hihat type hits. These could be extracted from drum audio or from other rhythmic instruments such as guitar chord strums.

2. VARIATIONAL INFERENCE

Once loops are present in a Bayesian network model M, exact inference of the data-conditioned joint $P(x_1, ..., x_N|d)$ and its Maximum A Posteriori (MAP) solution become intractable and approximations must be used. The VB approximation assumes a mean-field factorization:

$$P(x_1, ..., x_N|d, M) \approx Q(x_1, ..., x_N) = \prod_i^N Q_i(x_i)$$

where the node distributions $Q_i(x_i)$ are of the same form as the transition probabilities $P(x_i|par(x_i))$ and $par(x_i)$ are the parents of x_i. VB seeks parameters of the Q_i so as to minimize the KL divergence $KL[Q||P]$. It can be shown (e.g. [5]) that this may be accomplished by iteratively updating individual nodes distributions with the VB update rule

$$Q_i(x_i) \leftarrow \langle \log P(x_i|mb(x_i)) \rangle_{Q(mb(x_i))}$$

where mb denotes the Markov Blanket of the node. When nodes are conjugate exponential the expectation has an analytic solution and the update is computationally simple. As shown in [5], the model log likelihood, $\log P(d|M)$, has a tractable lower bound:

$$\sum_{i=1}^N \left[\langle \log P(x_i|par(x_i)) \rangle_Q - \langle \log Q_i(x_i) \rangle_Q \right].$$

3. SINGLE MODELS

We model the whole performance as a series of Bayesian networks, stitched together. Each network models a known 4/4 single-bar rhythmic pattern. At each bar, we compare several models to find the best fitting one, and the MAP

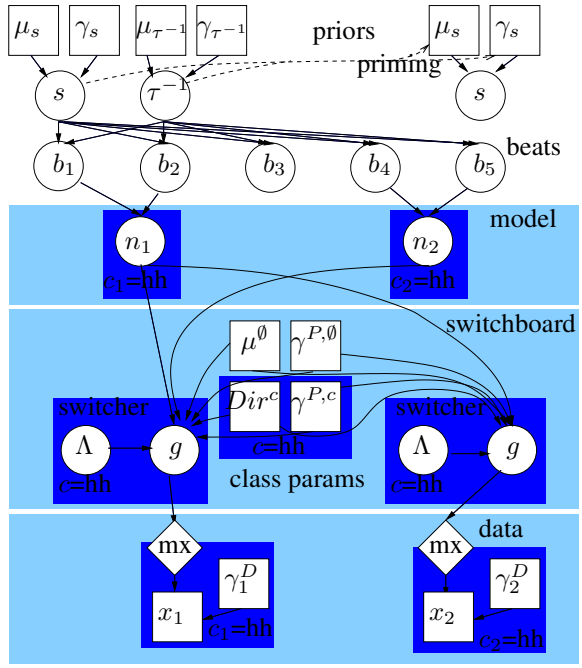


Figure 2. Schematic of a single-bar Bayesian network.

values of its parameters. These parameters may then be used to extract the model's estimate of the past, present and future beat locations as required. (We leave aside the question of how and when to instantiate models, but will return to it in section 4.)

Fig. 2 shows a model for a very simple rhythm consisting of just two hihat (hh) notes. Fixed nodes are shown as squares and hidden nodes are circles. The diamond-shaped *multiplexer* nodes will be discussed later: for now, we may assume they simply forward incoming and outgoing messages unchanged.

The goal is to infer the beat positions $\{b_i\}_{i=1:5}$, along with the start time s and tempo τ. To utilize the efficiency of conjugate-exponential models, we work with the inverse tempo, τ^{-1} instead of τ. Both s and τ^{-1} are modelled as Gaussians, with fixed mean and precision priors.

The beat positions are modelled as linear combinations, $b_i = s + i\tau^{-1}$, and note positions are modelled as the weighted linear combination of their two nearest beats, $n_j = b_{i(j)} + w_j(b_{i(j)+1} - b_{i(j)})$ where $i(j)$ is now a function specifying the latest beat number i before the jth note, and w_j is a rational weight with $0 < w_j < 1$.

Incoming observations are shown in the lowest layer of fig. 2 . We assume that an external system reports onset times x_k with associated (D)etection precisions γ_k^D and discrete classes c_k (with values for bass, snare and hihat). The precisions represent possible errors in the detection process, not in any high-level sources of note deviations such as changes in tempo or performance timing errors. Once reported, x_k, γ_k^D and c_k are modeled as fixed. Note that a more fully Bayesian system would model them as random variables, allowing prior information from other parts of the music model to modify the onset detector's beliefs. In particular, priors from rival rhythm models

could affect the observations, i.e. rival models' parameters would become dependent on each other. This would greatly increase the complexity of inference and it does not appear to give any practical advantages over independent models, as our task is ultimately to select one winning model. Forwarding messages to rival models without the usual co-parent dependency is performed by the special *multiplexer* nodes in the observation layer.

3.1. Switchboard

We want to infer the positions of notes and beats from the set of observations. If each observed note x_k arrived tagged with a marker saying which note n_j of the model it was an instantiation of, this would be easy, as we could then connect the observed x_k to n_j and perform inference. Hence x_k would be treated as a noisy observation of n_j (as in the systems of Cemgil and Raphael). However we do not have such tagging information, we only have the class of the observation. We are told that it is, say, a bass drum note, but not which of the (many) bass drum notes in the bar. Further, we want to allow the possibility that non-model notes may be played. This may occur for several reasons: (1) extra note played by accident, e.g. the performer's hand slipping; (2) performer is playing a variation of a known model, with deliberate extra notes; (3) performer is playing an unknown model. We must try to perceive the latter as a known model as best we can for now, then perhaps consider the known model's failings and update the model or learn a new model from them offline.

We solve the tagging problem by modeling the belief in the observation locations as a Gaussian Mixture Models (GMMs), with one Gaussian source from each of the M model notes of the same class as the observation. The middle shaded layer of fig. 2 is called the switchboard, and consists mostly of clusters of nodes called switchers. (There are also some extra nodes which are shared across switchers.) One switcher is created per incoming observation k, and includes a GMM node, g, together with a discrete mixture weight node Λ, so that its prior is

$$P(g|\{\mu_m\},\{\gamma_m^P\},\{\Lambda_m\}) \propto \sum_{m=1}^{M} \Lambda_m \exp\left[-\frac{\gamma_m^P}{2}(g-\mu_m)^2\right]$$

where γ_m^P is a (P)erformance precision parameter for the mth parent (see section 3.2). This distribution is not in the conjugate-exponential family, but may be approximated:

$$P(g|\{\mu_m\},\{\gamma_m^P\},\{\Lambda_m\}) \propto \sum_{m=1}^{M} \exp\left[-\frac{\Lambda_m\gamma_m^P}{2}(g-\mu_m)^2\right]$$

This moment-matches the original GMM, and has the useful property that it tends towards it as $\Lambda_m \to \Delta_{mm'}$, that is, as the discrete mixing vector becomes a single peak at m'. This is useful for our task, as we wish to ultimately assign each observation to a single class.

All Λ nodes of each class c share a fixed Dirichlet prior, Dir^c, which have hand-set parameters favouring single-source explanations.

3.1.1. Sibling rivalry

In addition to this GMM formalism, we know something more about the relationship between the observed and model notes: each model note will be instantiated at most once in the data. Even if, say, two snare hits occur very close to each other and to an expected model note location, only one of them can be 'the' instantiation of the note, and the other must be a spurious extra note. The standard GMM is not able to capture this 'sibling rivalry' between its children. GMMs are often used in data-intensive statistics where the task is to infer quantities from large numbers of noisy observations of them. In contrast, here we have at most one observation of each model note. The sibling rivalry problem is not unique to GMMs but is a general Bayes net problem: Bayes nets provide an automatic framework for competing *parents* to inhibit one another through the well-known 'explaining away' effect; but do not allow siblings to compete in any similar way.

To model our knowledge of sibling rivalry, we extend the Bayesian network into a Factor Graph, i.e. a graphical model containing both directed and undirected links. For each note class, we fully connect all the Λ nodes in the switchers of that class. We connect with inhibitory links, so that if one switcher lays claim to being an instance of a model note then it discourages other switchers from being instantiations of it. We extend the variational update for Λ nodes to include this undirected term (rightmost):

$$\log Q(\Lambda) \leftarrow \langle \log P(\Lambda|par(\Lambda))\rangle_{Q(par(\Lambda))}$$

$$+ \sum_{ch(\Lambda)} \langle \log P(ch|\Lambda)\rangle_{Q(ch(\Lambda),cop(\Lambda))}$$

$$+ \sum_{riv(\Lambda)} \langle \log \Phi(riv,\Lambda)\rangle_{Q(riv(\Lambda))}$$

As riv and Λ are discrete distributions parameterized by $Q(\Lambda=\Lambda_m)=q_m^{(\Lambda)}$, $Q(riv=riv_m)=q_m^{(riv)}$ with $\sum_m q_m^{(\Lambda)}=1$ and $\sum_m q_m^{(riv)}=1$, we set the additional update terms to

$$\langle \log \Phi(riv,\Lambda)\rangle_{Q(riv)} = \begin{bmatrix} 1-q_1^{(riv)} \\ \dots \\ 1-q_M^{(riv)} \end{bmatrix}$$

where M is the number of components. This is a heuristic whose intent is to encourage pairs of discrete nodes to have large divergences between them (e.g. in the KL-divergence sense). Note that this does *not* assign zero probability to two discrete notes having the same component, because its effect is on the logarithm of the resulting joint, not the joint itself. Instead, it merely discourages such similarity, and we have found it useful enough for our purposes.

Note that this scheme does not force any observations to match up with model notes: it merely discourages multiple observations from explaining the same model note.

3.1.2. Sinks

To allow for the performance of non-model notes, we construct an additional *sink* component for each note type. Sinks may be thought of as null hypothesis explanations for observations. They are modeled by Gaussians having their mean in the center of the bar, and a very wide variance so that the distribution stretches all the way across the bar in one standard deviation. This wide spread has the classic dual effect of (a) catching observations from a wide range, enabling the sink to soak up all observations not well accounted for by other components; (b) assigning a low likelihood to each observation because it is so spread out; hence allowing non-null components to be favoured in their specialist locations.

This may be viewed as a crude approximation to a fuller sink model: there is information contained in non-model notes which we are currently throwing away. In particular, notes do not occur at completely random places in the bar: even if not part of an identifiable model, they still tend to occur on beats or rational subdivisions of beats. In fact many beat identification systems have been constructed in this way. For example Cemgil [1] constructs an intricate prior over subdivisions similar to:

$$P(x) = \frac{1}{Z} \sum_{D=1}^{K} \sum_{N=1}^{2^D} \delta(x; \frac{N}{2^D}) * \exp(-\frac{\gamma}{2} x^2)$$

(The left part of this forms peaks at major subdivisions, whose height increases with importance, as measured by number of factors. The right term convolves (*) with a Gaussian to blur the spikes.) Such priors could in principle be incorporated here as GMMs, with one GMM component per subdivision. The advantage of such a scheme is that tempo and phase may then be inferred in a 'model-less' manner simply from collections of notes and the prior. However this approach would be time-consuming in a real-time system due to the large number of additional components to consider. We also note that 'model-less' is something of a misnomer, as we would effectively be specifying a complete, general pattern based on subdivisions: in practice we can know most common models in advance and exploit this by using our set of particular models instead of this general one. We have found that there is usually enough information in the instances of model notes to infer the beats well, without having to rely on sink notes to improve the accuracy.

There is still one problem with our single wide Gaussian approximation to this intricate structure. As the Cemgil prior is made of peaks, this structure assigns slightly higher priors to sink notes around rational subdivisions than our Gaussian assigns, at the expense of having lower priors than our Gaussian at places away from rational subdivisions. In practice, most (usually all) sink notes *do* occur at rational subdivisions. This means that our approximation will consistently assign lower priors to sink notes than the more accurate structure. To correct for this, we adjust the Dirichlet prior on the Λ nodes to give sinks a slightly higher prior than model notes. Empirically, $u = [0.10, ..., 0.10, 0.12]$ was found to work well, where the 0.12 is the sink component.

As discussed above, we constrain the GMMs with sibling rivalry links to prevent multiple observations mapping to a single model note. For sinks there is no such constraint: we allow multiple observations to be generated by the same sink. To accomplish this, the $\langle \log \Phi(riv, \Lambda) \rangle_{Q(riv)}$ term above is modified so as not to penalize sink-sharing.

3.2. Precisions

The switchboard contains two kinds of (P)erformance precision nodes. First, there is a single fixed precision $\gamma^{P,\emptyset}$ for sinks. This is used as an input to all GMMs, for the precision of their sink component, regardless of their class. It is set so that the standard deviation of the sink is one whole bar. Together with the Dirichlet prior amplification this produces a useful sink model. Second, for each note class c there is a fixed precision $\gamma^{P,c}$ which is shared between all switchers of that class. This performance precision models the player's accuracy in playing notes of class c. For example, the performer may be more accurate on the bass drum than on the hihat. Note that future versions of the system could extend this by allowing each model note to have its own performance precision, in the manner of Raphael's system [3]. This may be useful when the patterns become more complex, and the performer may try harder to play the on-beat notes very accurately at the expense of other more decorative notes. Alternatively, a simplified model might share just one performance precision across all note classes to allow faster inference.

Note that the performance precisions differ from the detection precisions γ^D – the latter being a measure of the detection error rather than the human player's deviation.

3.3. Annealed Variational Message Passing

To reduce the possibility of belief oscillations during message passing, we use variational annealing. In this scheme we do not seek $Q \approx P$ directly, rather we infer a sequence of *heated* distributions $Q^{1/T_i} \approx P^{1/T_i}$. The temperature T begins at a number greater than unity, and is gradually reduced to unity. Further, if an MAP solution is sought rather than a joint approximation, T may be reduced below unity and toward 0, in which case $Q^{1/T}(x) \rightarrow \delta(x; \hat{x})$, giving an estimate of the MAP. In practice, this MAP estimate is a near-delta spike at a local maximum of the posterior. As the cooling schedule slows and the initial temperature is raised, the probability of arriving at the global maximum increases. We extend the variational message passing framework to include annealing by giving each node a temperature parameter (for our conjugate exponential models this simply multiplies the parameter vector.)

Fig. 3 shows a difficult three-onset inference problem: the notes fall in such places as to be highly ambiguous under the position and tempo priors. Without annealing this causes the network to oscillate and diverge during inference; annealing cures this problem. Note that the distributions on the nodes begin by becoming very wide as mes-

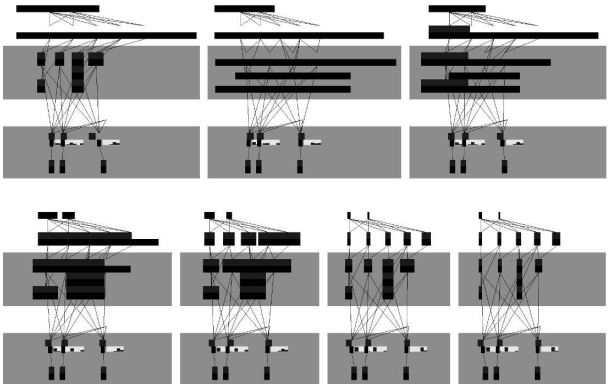


Figure 3. Seven steps of annealed variational inference on an ambiguous three-onset problem. As inference progresses, the heated initial beliefs converge to the correct VB posterior. Without annealing, the network can oscillate and diverge during inference.

sages are received; then gradually reduce as cooling takes place. (See section 5 for a description of the display.)

4. HYPOTHESIS MANAGEMENT

So far we have considered rhythm models of individual bars: test data was created representing a single bar's content of music, and the postulated model had to explain all the data. However in the wild, the situation is more complex: we receive a stream of events over a period of several minutes (the song) and we do not know initially which events belong to which bar and hence to which model.

The ideal Bayesian approach would be to consider all possible models instantiated at all possible points in time-tempo space, together with a prior on which combinations of them are allowable and mixture switches to allow data points to be explained as mixtures of models. The prior would act to allow only sets of rhythm models which are contiguous throughout the whole song, and could further specify Markovian or grammatical priors on the probabilities of particular sequences of models.

This approach is impractical because it requires an infinite number of models, and even quantizing the possible model instances would require a very large number of models. In particular the prior would have to operate on the power set of these models which is intractable. So we must use heuristics to approximate the Bayesian ideal.

The heuristics we use are based on the ideas of priming and pruning presented by [8]. Reis modeled symbolic sequences using a set of *agents,* each agent being a detector for a particular pattern. When an agent *A* successfully completes its pattern (by observing the exact required sequence – Reis' work was non-probabilistic) it competes with completed *rival* agents, and the winner *W* primes all other agents to start looking for their patterns beginning at the point that *W* completed. Rival agents are those which share one or more symbols of the data stream. Losing agents are deleted (*pruned*).

We extend this method to the case of continuous time events (opposed to sequences of symbols). We replace discrete symbol sequence detecting agents with our variational models. Competition between models was handled with a heuristic by Reis but we are able to use the variational lower bound on model likelihood (from section 2) to provide a more rigourous foundation for competition. The principal difficulty is the lack of sharp boundaries between models together with the uncertainty in note arrival times: in the symbolic domain is simply to say 'if the new note time is after the end of agent *A* then add it to the next agent instead of to *A*' but the meaning of this is less clear in our probabilistic domain: there will instead be some probability that the new note lies inside or outside the model. Pragmatically, we simply set a probability threshold and declare the new note to be outside the model if it is exceeded. The pseudocode for our algorithm is thus:

```
prime initial model set
set initial models to incomplete state
for each incoming note n do
  for each incomplete model M do
    if P(n after M < threshold) then
      link n to M
      update temperature T
    else
      set M to completed state
      winner W ← M
      for each rival model R (overlapping M) do
        if R completed then
          if log Q(R|d_R)/|d_R| > log Q(W|d_W)/|d_W|
          then
            W ← R
            prune R and descendants of R
          end if
        end if
      end for
      prime from W
    end if
  end for
end for
```

where $d_M$is the set of data points attached to model M and $|d_M|$ is the size of this set.

There is an issue regarding model comparison when two rivals *R* and *W* have different numbers of data points. Instead of raw model posteriors, we compare the log posteriors *per data point*, dividing the lower bounds by the number of data. This statistic may be thought of as an approximate log probability *density* per unit time, with data points as a proxy for time. (Note that log posterior densities are typically negative, so dividing them *increases* their value.) The reason for comparing *log* probability per data point is as follows. Suppose we have N models in a sequence, $M_1, M_2, ..., M_N$ which together cover a complete song. The issue is that data points near model boundaries may fall under the model at either side of the boundary. We do not wish to penalize either case in the ideal *global* distribution of models. So we note that our statistic has this property: the score for the whole song is

$$\frac{\sum_{i=1}^{N} \log P(d_{M_i}|M_i)}{\sum_{i=1}^{N} |d_{M_i}|}$$

From the point of view of the global distribution, the denominator is constant so does not penalize any particular assignments of data to models. However recall that we are using a heuristic to approximate the global distribution, namely that we make a hard decision about which models to keep and reject at the end of each bar. And from the viewpoint of these individual bars, the statistic *does* make a difference, namely that of transforming our measure of log likelihood into log likelihood per data point explained, which aids our greedy search through the space of sequences of models as required.

Note that we have included the annealing schedule as part of the node linking cycle. We want the model to start as 'hot' when first created, then to cool as *real* time progresses, so that it reaches the true probability scale as it completes. We use the time of the latest received note as a proxy for elapsed time, and have found that a useful schedule is to cool linearly from $T = 8$ to $T = 1$ during the course of the bar.

4.1. Priming

In the present work we have used a simple Markov transition matrix to specify priors on sequences of models. (e.g. the low probability of a drum'n'bass pattern following a military march pattern.) Note there is an important practical difference between transitions of zero probability and those of small but non-zero probability. Conceptually these are similar, but the former will *not be primed* whereas the latter are primed but are unlikely to be chosen as the final winner. This makes a huge difference to the amount of computation that takes place, but there is the classic tradeoff that in setting unlikely transitions to a hard zero, we rule out the ability to use them to explain uncommon situations, so in the unlikely event that those situations occur, the model will fail. (Primed models are 'known unknowns', as they still contain much uncertainty but are under active consideration'; unprimed models are 'unknown unknowns'.)

Newly primed models are instantiated having a start time with mean equal to the MAP end time of the previous (completed) model – shown by the dotted line in fig. 2 – and inverse tempo mean equal to the MAP of the previous inverse tempo. Parameters specify precisions of these beliefs, which at present are global and hand-set. However it would be conceptually simple to use an EM method as in [3] to learn precisions for each bar and also to learn priors on tempo transitions.

5. EVALUATION AND DISCUSSION

We believe it is more instructive to present a walkthough of a real test run than to give only statistics of its results. The images in figs. 4 and 5 show the result of a complete run on a reasonably challenging data set. The run is quasi-realtime, meaning that simulated time runs at half the speed of the recorded input to allow more message passing cycles. No effort has been made to optimize the interpreted Lisp research code, though the run was performed on a 1.6GHz Pentium with a quasi-time factor of only two which suggests real-time optimization is possible. The data set was recorded by a human drummer using a MIDI kit, and standard deviations of 100ms attached to notes of all types. It consists of 48 bars of rhythms lasting about 2 minutes and including over 250 hits, which is a typical amount of data for beat tracking evaluations, (e.g. [7]). The rhythms begin as simple known patterns and get progressively more complex, and more errors are introduced. A knowledge base of six models was used, including three regular rhythms and three fills. The data begins with a simple four hihat count-in to initialize the priors; the program is initialized with a single count-in model at the correct time and roughly correct tempo, much as a human musician would require. The following description gives a walkthrough of the activity in the first 12 bars only.

Each row of the figures shows four bars. Only the winning models are shown. Each model has the same visual form as fig. 2: at the top left is the s node, to its immediate right is τ^{-1}. Below these are the beats. Time is shown horizontally on each row, and the standard deviation of each node is shown by its width. The τ^{-1} nodes are shown at one beat away from the s nodes. The three shaded areas contain the model notes, the switchboard, and the observations. Within each shaded area are three sublevels for hihat (top), snare (middle) and bass (bottom) – the same format as fig. 1(b). Each switchboard node has a small bar-chart on its right. This shows the value of its switch node where the nth value is the nth model note of the matching type with the rightmost being the sink. All nodes have upper and lower halves showing their prior and posterior respectively – the deviation from the prior is visible only in s and τ^{-1} when inference has converged.

Bars 1 and 2 form a pair of `rock1` and `rock2` patterns. There is a high prior on such a transition. Note that in bar 1, an additional bass drum hit is played on beat 4. This is is not part of the pattern so is assigned to the sink. In bar 4 we see a performance of the highly syncopated `fill-offbeat1` pattern: four snare hits played between the beats instead of on them. This kind of pattern is likely to confuse conventional beat trackers that operate by looking for correlations and phase of onsets and assuming beats match strong onsets. But the system knows this pattern is a fill, albeit with a low occurrence prior, and is able to infer the beat positions correctly.

Bars 5 and 6 see another standard `rock1`-`rock2` pair. Note the change in s belief in bar 5: the posterior (lower half) s is later than one standard deviation away from the prior (upper half), due to the player loosing time when recovering from the offbeat fill pattern. Bars 7 and 8 contain a further `rock1`-`rock2` pattern, but in bar 8 the snare hit has been intentionally played half a beat late: it is assigned to the sink (the rightmost bar in its barchart). A possible extension here could be: instead of modeling such

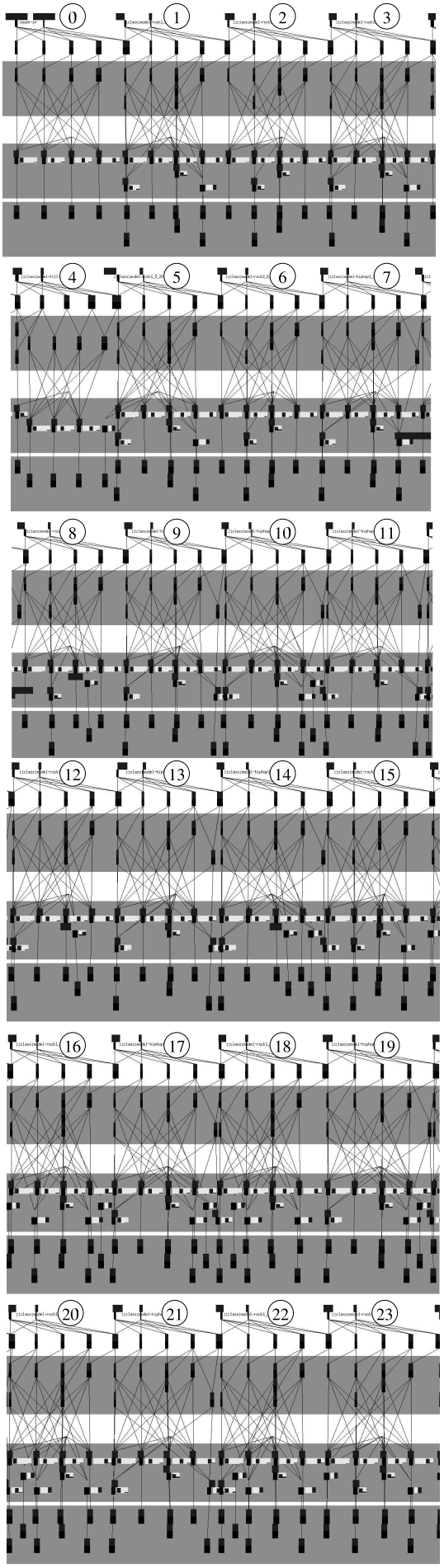


Figure 4. Bars 0-23 of the test run. Bar 0 is the hihat count-in. Bar numbers are shown above the center of each bar.

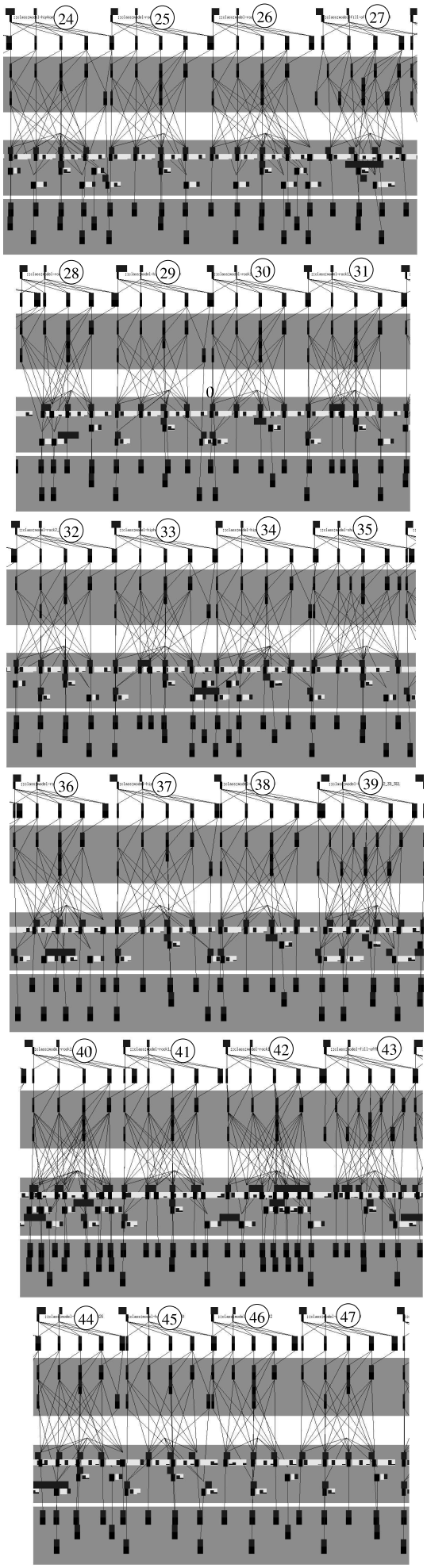


Figure 5. Bars 24-47 of the test run.

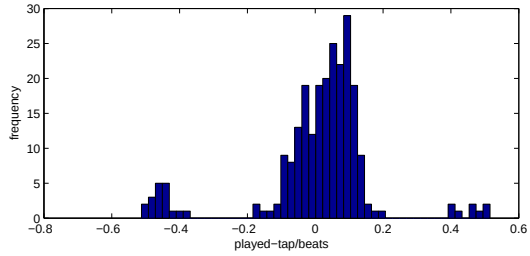


Figure 6. Error of played onsets from hand-tagged beats.

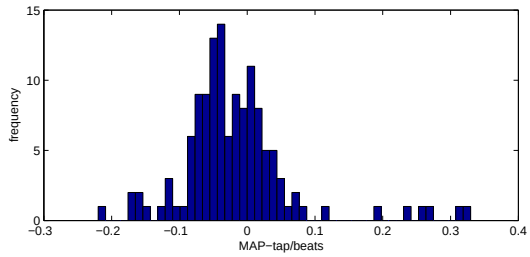


Figure 7. Error of played onsets from auto-tagged beats.

notes from the sink, we might explicitly model them as played intentionally late; a human listener primed to expect `rock2` is likely to hear the note in this way and in fact it is this feeling of lateness that creates the laidback mood of this pattern, sometimes called a 'lounge' beat.

Bars 9-14 switch to a hiphop feel, due to extra hits at beat 4.5. Note again the late snare in bar 12, beat 3.5. It is interesting to consider the 4.5 hit in the same way as in the extension discussed above: currently the system just knows the whole pattern `hiphop1` which contains this 4.5 hit. But humans (especially if unfamiliar with hiphop) may initially hear the 4.5 hit as the first beat of the next bar but played early – in the same way that the 3.5 snare was heard as being from beat 3 but late. In hiphop, the 4.5 hit is quickly followed by a 5.0 hit marking the true start of the next bar. But for half a beat there is a brief feeling that the music has jumped forward a little, before being snapped back. (Perhaps it is this feeling that gives this rhythm its characteristic mood and edgy dancing moves? Again, future work could try to model these ideas.)

The reader may examine the rest of the run, noting the near-losses of tracking around bars 26-28 and 36 and subsequent recovery; and the eventual failure due to complex unknown rhythms at the end.

Fig. 6 shows the deviation of the played onsets from 'ground truth' beats (as labeled by hand, offline), and can be seen as a measure of the accuracy of the playing. The x axis measures fractions of a beat deviation from the nearest labeled beat. Note the clusters around +/- 0.5 beats which are notes played half way between beats. Excluding these notes, the standard deviation (player error) from the beat is 0.07 beats, corresponding to 35ms at 120bpm.

Fig. 7 shows a histogram of errors of the system's MAP beat positions from the ground truth. Only the correctly-tracked bars 1-38 are included. The standard deviation of the errors is 0.12 beats, which corresponds to a musical semi-quaver, or 60ms for a track played at 120bpm. However this large deviation includes the outliers shown in the graph which arise during the near-loss of tracking around bar 27. Excluding these incorrect models gives a deviation of 0.056 beats over 129 data points, equivalent to 28ms at 120bpm. This is a significant (a χ^2 test gives $P(s^2 \leq 0.056^2 | \sigma^2 = 0.07^2, N = 129) = 0.0004$) improvement from the 35ms player latency above (for example if the raw onsets had been used as beat locations.) Typical music software has a latency of around 10ms in comparison. Inspection of the worst errors shows them to occur in bars where the tempo changes within the bar rather than at the barline, this is of course expected as our current model assumes a single tempo within each bar.

While our proof-of-concept implementation of the variational and blackboard ideas described here is not yet at contest standard, we have explained how and why the approach is useful for solving higher-level rhythm and beat perception tasks than those tackled by current beat-trackers. By incorporating psychological ideas of model-based perception, priming and a global workspace, the approach may also lead to clarifications of perceptual music psychology concepts and could even serve as a framework for empirically modeling human rhythm perception, especially for interesting ambiguous percepts.

6. REFERENCES

[1] A. T. Cemgil, H. Kappen, P. Desain, and H. Honing. "On tempo tracking: Tempogram Representation and Kalman filtering" *JNMR*, 28:259-273, 2001.

[2] N. Orio and F. Dechelle. "Score Following Using Spectral Analysis and Hidden Markov Models" *Proceed. ICMC*. 2001.

[3] C. Raphael. "Music Plus One: A System for Flexible and Expressive Musical Accompaniment" *Proceed. ICMC*. 2001.

[4] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. Sandler. "A tutorial on onset detection in music signals." *IEEE Trans. Speech and Audio Proc.* 13(5):1035-1047, 2005.

[5] J. Winn and C. Bishop. "Variational Message Passing" *J. Machine Learning Res.* 6:661-694, 2005.

[6] M. Goto. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds" *JNMR*, 30(2):159-171, 2001.

[7] N. Collins. "DrumTrack: Beat Induction from an Acoustic Drum Kit with Synchronised Scheduling" *Proceed. ICMC*. 2005.

[8] B. Reis. "A multi-agent system for on-line modeling, parsing and prediction of discrete time series data" *Intelligent Image Processing, Data Analysis and Information Retrieval*, IOS, Holland, 1999.

HOW TO BE LOST: PRINCIPLED PRIMING AND PRUNING WITH PARTICLES IN SCORE FOLLOWING

Charles Fox

Robotics Research Group
Dept. Engineering Science
University of Oxford
charles@robots.ox.ac.uk

John Quinn

Institute for Adaptive and
Neural Computation
School of Informatics
University of Edinburgh

ABSTRACT

Previous work in score following has provided methods for aligning a skilled live performance to a symbolic or audio score. In the Bayesian framework, ideal generative models require $O(n)$ computations at each real time step where n is the length of the score. In practice, heuristic thresholds have been used to consider only a subspace of generative models with high priors conditioned on the previous state. These heuristics work well for skilled performances but fail when large errors are made by amateur musicians. We present a novel Priming Particle Filter for audio scores which places the order-limiting heuristic on a firm foundation and adds the ability to recover from large errors by using psychologically-inspired bottom-up priming in addition to regular sequential importance sampling.

1. INTRODUCTION

Automated score following is the task of listening to a human performance of a predetermined score and playing a predetermined accompaniment in real-time which adapts to tempo changes and errors of the player.

Dannenberg [1] applied Dynamic Time Warping (DTW) to follow MIDI performances, allowing for insertion and omission of notes. The DTW path gradient is used to approximate tempo and hence provide accompaniment. In real-time, the optimal path is noted at each step and the search on the next row is constrained to nearby elements within an arbitrary window. Hence the complexity at each time point is $O(1)$ rather than $O(n)$ where n is the score length. This approach was extended to audio alignment: score and observation audio are cut into frames, and 'chroma' vectors are extracted which shift all pitches to within a single octave. A distance measure is defined on pairs of chroma vectors, and used as the local distance in DTW.

DTW is a likelihood-based approach to alignment. To perform it exactly requires each incoming live observation to be dot-producted with every score position, then the shortest path through this matrix is found. There is no explicit prior over possible paths: we can consider it flat. In practice, it is intractable to perform comparisons with the whole score, so DTW algorithms typically consider only a fixed window (of perhaps 100 audio frames) about the previous best point. This can be thought of as specifying a finite plateau transition prior centered around the previous MAP point. Importantly, this heuristic specifies not only a prior belief but also a *computational* simplification: it tells us to *consider* only those hypotheses in the plateau rather than all possible generating score states.

Moving from the plateau prior to an arbitrary state transition matrix, DTW becomes a general Hidden Markov Model. Rather than use score audio frames as states (as in DTW), score-following HMM researchers have tended to use a reduced model of the score, with one or several states per note ([2], [3]). Transitions are typically allowed between consecutive score states, self-transitions, and jumps to the end of the note, and their parameters fitted to previous performances.

While HMMs provide explicit prior beliefs about transitions, they do not address the computational problem of which beliefs to *consider* in computations. As in DTW, practical HMMs generally apply some kind of cutoff threshold to limit the space of score states under consideration. For example, Raphael [5] uses a heuristic which prunes all but a fixed number of most probable posterior hypotheses at each time step. These states tend to be closely clustered. This works well for Raphael's intended users: professional performers who are unlikely to make large errors, and whose small-scale tempo deviations from previous performances can be tracked with high accuracy. But for amateurs who may make very large leaps around the piece it runs the risk of becoming irrecoverably lost.

Heuristic hypothesis management is the 'dirty secret' of much of Bayesian inference. Particle Filtering [8] is a more principled approach to hypothesis management in dynamic Bayesian networks including HMMs. Rather than keeping the most probable hypotheses, it samples from them at each time step. This can allow for less probable paths to be explored in the hope that they may eventually provide a better global path than locally probable ones. Particle filtering has been applied to model-based score-following HMMs by [7]. This allows for larger leaps in the score than a 'take only the best' approach, but it is still possible for particle filters to become lost ('diverge') and

they are then unlikely to recover.

Human musicians do not just rely on a prior information to select location hypotheses to consider: hypotheses can also be *primed* [9] bottom-up from observed features. For example a song may have a distinctive chord change at the start of a the chorus: when this change is heard, it primes us to consider that we might be at that location – even if our prior beliefs were focused elsewhere. Within the framework of particle filtering, we define priming to be the injection of new samples into the system driven only by bottom-up features, regardless of their priors. We present a novel 'Priming Particle Filter' (PPF) which uses this technique to recover from being lost by recognizing and allowing a probability for its own error, then creating new hypotheses based on bottom-up note changes. Unlike existing HMM score-followers, we have returned to the DTW-style audio-based score, and avoid the need for explicit note-based score models. To demonstrate the priming particle filter we have used relatively simple features for likelihood computations, but we suggest that the PPF could be a useful addition to all state-of-the-art score followers ([4], [6]) with more advanced features.

2. PARTICLE FILTERING IN HMMS

Hidden Markov Models assume a discrete hidden state $x_i[t]$ at discrete time steps t with linear Markovian transitions, $P(x_i[t+1]) = \sum_j P(x_i[t+1]|x_j[t])P(x_j[t])$, together with observations $P(y[t]) = f(x[t])$. The filtering posterior at each step is given recursively by $P(x_i[t]|y[1:t]) = \frac{1}{Z}P(y[t]|x_i)\sum_j P(x_i[t]|x_j[t-1])P(x_j|y[1:t-1])$ where the sum is over all possible hidden states, so scales with the length of a score. Particle Filters can be used to approximate the sum by maintaining a limited set of samples from each $P(x[t]|y[1:t])$. We base our Priming Particle Filter on a standard particle filtering technique, Sequential Importance Resampling, whose algorithm is:

for each time step t **do**
  for $p = 1 : N$ **do**
    sample $x^p[t] \sim P(x_i|x^p[t-1])$
  end for
  for $p = 1 : N$ **do**
    $w^p[t] := \frac{1}{Z}w^p[t]P(y[t]|x^p[t])$
  end for
  if $\frac{1}{\sum_{p=1}^{N}(w^p)^2} < N_{thresh}$ **then**
    resample $x^p[t] \sim P(x[t] = i) = w^i[t]$
  end if
end for

where N is the number of particles and N_{thresh} is a hand-set threshold for number of effective particles. The resampling step becomes necessary in the case of degeneracy (i.e. when most particle weights $w^p[t]$ become small). See [8] for details.

3. SCORE FOLLOWING MODEL

3.1. Tempo model

At each (discrete) frame of live performance time $t^{(l)}$, the hidden state is the score position in frames, $t^{(s)}$ and the current tempo. These hidden state variables are modeled as continuous and are lazily discretized only when a hard output frame decision is required. We write the score time as a function of the live time, $t^{(s)}[t^{(l)}]$, and write the current tempo as $\dot{t}^{(s)}[t^{(l)}]$. The hidden state evolves as the damped switching stochastic process:

$$t^{(s)}[t^{(l)}] = t^{(s)}[t^{(l)}-1] + \dot{t}^{(s)}[t^{(l)}-1] + \epsilon_{t^{(s)}}$$
$$\dot{t}^{(s)}[t^{(l)}] = \rho(\dot{t}^{(s)}[t^{(l)}-1] + \epsilon_{\dot{t}^{(s)}} - 1) + 1$$

where ρ is a damping coefficient and $\epsilon_{t^{(s)}}$ are from the Gaussian $N(0, \sigma_{t^{(s)}})$ and $\epsilon_{\dot{t}^{(s)}}$ are from the switching mixture of Gaussians $\alpha_1 N(0, \sigma_{\dot{t}^{(s)}}) + \alpha_2 N(0, \sigma_{\dot{t}^{(s)}})$ with $\alpha_1 + \alpha_2 = 1$. The two mixture components model a small tempo drift due to player or tracking errors and a separate large tempo change due to performer style changes. A mixture is used to encourage occasional, sudden large changes (e.g. at new phrases) whilst discouraging gradual large tempo changes (e.g. during notes). We do not perform any learning of parameters – they are set by hand. Particles contain the 2-element state $(t^{(s)}[t^{(l)}], \dot{t}^{(s)}[t^{(l)}])$, i.e. a position and speed in the score.

3.2. Likelihoods

Likelihoods are approximated using chroma features similar to [1]. Offline score $y^{(s)}$ and online live $y^{(l)}$ audio is sampled at 2756.3Hz (=44.1kHz/16) and cut into 512-point frames $y[t]$ with 448 point overlaps. The signals are differentiated to remove linear components then Hann-windowed and the power FFTs are computed. The total energy $e[t]$ and normalized power FFT $Y[t]$ are stored:

$$Y[t] = \frac{1}{Z}|FFT(Hann(y'[t]))|^2$$
$$e[t] = \sum_{\omega}|(FFT(Hann(y'[t]))|^2$$

where Z is a normalizing constant and ω sums over frequency components. A filterbank of note detectors is projected onto the power FFT with each detector having a 10-peaked harmonic series:

$$n_i[\omega] = \frac{1}{Z}\sum_{k=1}^{10}\exp(-k)\Phi(k\omega; 2^{\frac{i}{12}}\omega_0, \sigma)$$

where $\Phi(\omega; \mu, \sigma)$ is the Gaussian pdf, ω_0 is the frequency of the lowest considered pitch, and σ is chosen by hand to give a reasonable spread but without interfering with neighboring frequencies. (Ideally σ would be derived from acoustic theory or fit to data.)

Note detectors over a two-octave range are summed to give a 12-valued octave-independent chromacity vector:

$$N_i(Y[t]) = \sum_j \langle n_{i+12*j}|Y[t]\rangle$$

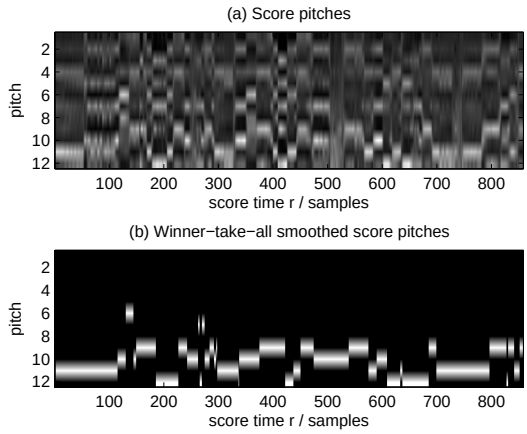


Figure 1. Chroma and change points of score performance

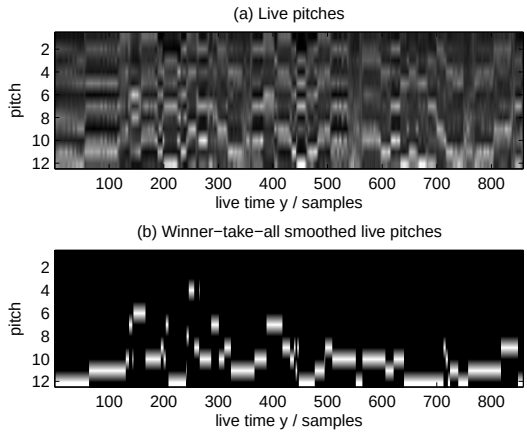


Figure 2. Chroma and change points of live performance

We use an inner product between chroma vectors as an approximate spectral likelihood (i.e. the probability of the observed spectrum at live time $t^{(l)}$ given score time $t^{(s)}$):

$$\lambda_s(Y^{(s)}[t^{(s)}], Y^{(l)}[t^{(l)}]) = \frac{\langle Y^{(s)}[t^{(s)}] | Y^{(l)}[t^{(l)}] \rangle}{\sqrt{e^{(s)}[t^{(s)}] e^{(l)}[t^{(l)}]}}$$

Energy values are normalized by local windowed (length L) mean and variance to compute novelty features:

$$v[t] = \frac{e[t] - \mu_{j=t-L:t}(e[j])}{\sigma_{j=t-L:t} e[j]}$$

We use a Gaussian approximation of energy likelihoods:

$$\lambda_v(v^{(s)}, v^{(l)}) = \frac{1}{Z} \exp \frac{-(v^{(s)} - v^{(l)})^2}{2\sigma_v^2}$$

and assume that chroma distance and energy are sufficient statistics for the total likelihood:

$$P(t^{(l)} | t^{(s)}) \approx \lambda_s(Y^{(s)}[t^{(s)}], Y^{(l)}[t^{(l)}]) \lambda_v(v^{(s)}[t^{(s)}], v^{(l)}[t^{(l)}])$$

3.3. Injecting primed particles

A fundamental assumption in Bayesian inference is that we posses the exhaustive set of hypotheses $\{H_i\}$ to explain some data D. It is this exhaustiveness that allows us to convert likelihoods into posterior probabilities. However in realtime score following we do not have computational resources to consider all possible score position hypotheses, we must consider only a subset of them. Previous models have used heuristics to choose this subset: we (with [7]) use sequential importance samples (particles) to make this choice in a principled way. The standard particle filter assumes that the particles form the exhaustive hypothesis set at each step. But this does not allow the model to represent the possibility of its own failure: in practice it is possible for a particle filter to get lost, and for the best hypothesis (and even the whole area around it) to be excluded from consideration. But the particle filter is blind to this. What we would like to do is compute the probability that all particles have failed, $P(lost)$, and treat 'being lost' as an alternative explanation. When we know we are lost we will then consider a number of newly injected particles $\{H_j^*\}$ based on bottom-up priming, with

$$P(H_j^* | D, C) = P(lost | D, C) P(H_j^* | lost, D, C)$$

$$P(H_j^* | lost, D, C) = \frac{1}{Z} P'(H_j^* | feature(D)) P'(H^* | C)$$

where C is the context (i.e. the previous state for dynamic networks like HMMs), $feature(D)$ is some bottom-up priming feature computed from the data, and P' are probability *factors*. In practice, all of the above terms are intractable (in the sense that computing them would require an exact solution of the very inference problem that the particle filter is approximating) but can be approximated from domain knowledge or historical statistics.

We estimate $P(lost | D, C)$ by looking at a windowed running average data likelihood. We use an exponentially weighted moving average of $P(D[t] | \hat{H}[t])$, the evidence of the MAP hypothesis at each step. When this average falls below a threshold we consider that we might be lost with a fixed probability. In practice we approximate this by removing the worst m particles from the set and replacing them with newly primed particles, where m is the number of these primes. (More detailed methods could model how probable being lost is given how far below threshold we are, and learn this model from historical data.)

$P'(H_j^* | feature[D])$ is approximated by a simple bottom-up priming scheme. We use sparse Boolean features and assume that all score states having the feature are equally likely given the presence of the feature when lost. For our score-following task, we use changes in the maximum windowed running average pitch as features. These *roughly* correspond to onsets of new notes. We maintain a cache of known positions of these features in the score, and consider these positions as hypotheses when the same features are found in a lost live performance. Speeds are created for these particles assuming that the tempo has been constant since the point of becoming lost.

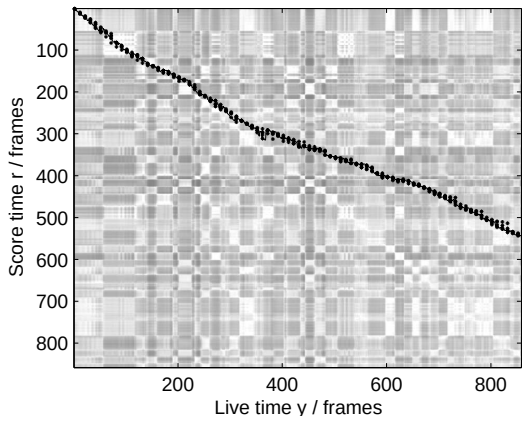


Figure 3. Alignment without priming.

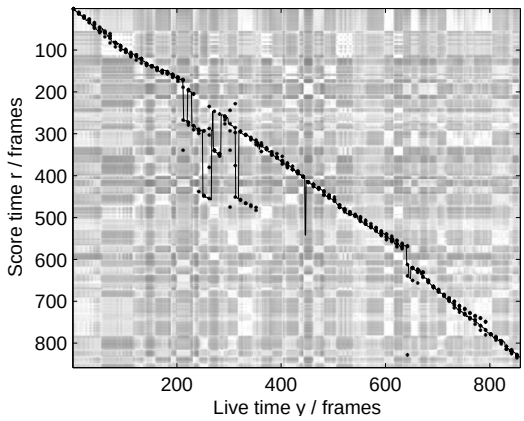


Figure 4. Alignment with priming.

$P'(H^*|C)$ can be approximated in score-following by considering how far removed the primed hypothesis H* is from where the particle filter's previous MAP estimate was, and penalizing very large jumps. Our system uses a simple plateau function to allow only primes within a fixed distance of the previous MAP.

Our method corresponds to altering the distribution from which particles are sampled from in the SIR algorithm. If we know that we are lost, prior knowledge about likely 'primed' positions is used to obtain a more appropriate set of samples.'

4. RESULTS

We demonstrate the priming particle filter on the first seven bars of the *Andante Cantable* from the Rimsky-Korsakov *Trombone Concerto*. Recall that no symbolic score is required: an audio recording is used as the score. Fig. 1 shows the chroma vectors and priming features from the audio score; fig. 2 shows the same for the live recording.

The live recording contains a short error around frame 200. Running the standard particle filter with 5 particles results in getting lost at this point as shown in fig. 3. The figure shows the complete spectral similarity matrix in the background, in which the central white diagonal stripe is the true path. The particles are plotted as dots and the MAP particle path is drawn as a line. Fig. 4 shows the same example running under the Priming Particle Filter. The large jumps are where lostness recognition and priming has occurred. It can be seen that the filter gets lost several times but is able to recover every time.

5. DISCUSSION

We have demonstrated the Priming Particle Filter on a simple example for illustration purposes only, and running with more than 5 particles on this example generally avoids getting lost in the first place. Our demonstration PPF is a relatively simple score follower using very basic likelihood features and a Markov hidden state in contrast to state-of-the-art score followers such as [4], [6] which have more advanced features and higher-level state models. However we suggest that all state-of-the-art models could benefit from the simple addition of a PPF to allow them to recover from being lost when used by non-professional performers.

6. ACKNOWLEDGMENTS

Thanks to Matthew Dovey, Chris Raphael and Diemo Schwarz for discussions.

7. REFERENCES

[1] R. Dannenberg and N. Hu. "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors" *ICMC*. 2003.

[2] C. Raphael. "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models" *IEEE PAMI*. 21(4):360-370. 1999.

[3] N. Orio and F. Dechelle. "Score Following Using Spectral Analysis and Hidden Markov Models" *ICMC*. 2001.

[4] C. Raphael. "Music Plus One: A System for Flexible and Expressive Musical Accompaniment" *ICMC*. 2001.

[5] C. Raphael. "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models" *IEEE PAMI*. 21(4):360–370, 1999.

[6] A. Cont and D. Schwarz. "Score Following at Ircam" *MIREX Score Following Contest*, 2006.

[7] A. Cont. "Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative constraints and Hierarchical HMMs" *IEEE Int. Conf. Acoustics and Speech Sig. Proc.*, 2006.

[8] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice* Springer. 2001.

[9] C. Wiggs and A. Martin. "Properties and mechanisms of perceptual priming." *Cur. Op. Neurobiology (CNS). (8) 227-233.* 1998.

Local Quantum Computing for Fast Probably MAP Inference in Graphical Models

Charles Fox Iead Rezek Stephen Roberts
Robotics Research Group
Department of Engineering Science
University of Oxford
charles@robots.ox.ac.uk

Abstract

Maximum a Posteriori (MAP) inference in graphical models is a fundamental task but is generally NP-hard. We present a quantum computing algorithm to speed it up. The algorithm uses only small, local operators, and is based on a physical analogy of striking the nodes in a network with superposed 'coolants'. It may be viewed as a quantum version of the Gibbs sampler, utilising entanglement and superposition to represent the entire joint over N variables using only N physical nodes in 2^N states of superposition, and exploring all MCMC trajectories simultaneously. Information is carried away by the coolants, making essential use of decoherence – often thought of as a hindrance rather than a benefit to quantum computation. The algorithm creates a superposition whose amplitudes represent *model* probabilities, then cools it to leave a sharp peak at the MAP solution. As quantum *observation* probabilities are *squares* of these amplitudes, the probability of observing the MAP solution is amplified quadratically over the classical case. Proof-of-concept simulation is presented.

Introduction

Quantum computing has become increasingly popular in the physics community and others. However, there is little work framing probabilistic inference tasks in a quantum computational setting. We present a quantum computational algorithm capable of performing MAP inference in in graphical models. The algorithm quickly finds a probably MAP or near-MAP configuration with a quadratic speedup over classical Gibbs sampling, in the sense that after the same number of sampling steps it yields observation probabilities $\frac{1}{Z}Pr_{obs}(a)^2$ of states a where $Pr_{obs}(a)$ are the classical Gibbs observation probabilities and Z is a normalising coefficient. The algorithm uses only small local operators and makes essential use of decoherence – which is usually thought of as a hindrance rather than a feature in quantum computation. Phase information is not used.

We work with an undirected, possibly loopy, pairwise Markov Random Field (MRF) with N Boolean-valued nodes X_i. Any graphical model can be converted into this form (e.g. (Myllymaki & Tirri 1993)). Let nodes have potentials ϕ_{ii} and links have potentials ϕ_{ij} so that the joint is

$$P(x_{1:N}) = \frac{1}{Z}\prod_i \phi_{ii} \prod_{ij} \phi_{ij}$$

where Z is a normalisation coefficient. We restrict our discussion to potentials which may be defined by integer-valued *energies* E_{ii} and E_{ij} such that

$$\phi_{ii} = \exp(-E_{ii}), \phi_{ij} = \exp(-E_{ij})$$

We wish to find the MAP configuration $\hat{x}_{1:N}$ to maximise $P(x_{1:N})$. This is generally NP-hard (Shimony 1994).

Overview of quantum computing

We briefly review some key concepts in quantum computation. For a full introduction see (Nielsen & Chuang 2000).

A classical register of bits exists in a single state, for example, a three-bit register could store a configuration of three Boolean MRF nodes such as 101 (also notated as the decimal 5). Such classical states may also be thought of as basis vectors in 2^3-dimensional Boolean space. To emphasise this view we use 'ket' notation:

$$|101\rangle \equiv |5\rangle \equiv [00000100]'$$

where the rightmost term is a column vector (the prime symbol notates matrix transposition) whose elements are the eight classical states from $|0\rangle$ to $|7\rangle$, so they are all zero except for the sixth element which represents the integer 5. Operations that map register states to register states (such as a particular state-change in a Gibbs sampler) may then be represented by a 2^3 permutation matrix.

When two registers are brought together and considered as a single system, the tensor product of their state vectors gives the state vector for the combined system, e.g.:

$$|101\rangle \otimes |001\rangle \equiv |101\rangle\,|001\rangle \equiv |101001\rangle$$

Quantum computing allows systems to exist at any point on the unit radius hypersphere in the state space, rather than just the classical basis vectors. For example a register can be in a superposition of basis states $|000\rangle$ and $|110\rangle$:

$$|\Psi\rangle = \alpha_{000}\,|000\rangle + \alpha_{110}\,|110\rangle$$

where the α are chosen to describe a hypersphere location, $\sum_i |\alpha_i|^2 = 1$, and each $|\alpha_i|^2$ describes the probability of an observation collapsing the register into state $|i\rangle$. In general the α_i may be complex, but our algorithm uses positive real values only. General unitary matrices may operate on the state vectors rather than just the special permutation

cases. (Unitary matrices have a 'reversible computing' property, and may be pictured as rotations on the hypersphere.)

Our notation uses upper and lower case letters in bare and ket forms as follows: A refers to a state space; $|A\rangle$ is a general vector in that space; a is an integer (in decimal or binary notation) denoting a basis state; $|a\rangle$ is the basis vector in A with coding a. (Such binary codings will be used to represent Boolean MRF configurations.) We write true and cooled *model* probabilities as $P(a)$ and $Q(a)$ in contrast to quantum *observation* probabilities $Pr_{obs}(a) = |\alpha_a|^2$.

Representing joints by superpositions

A quantum register of N qubits ('quantum bits') is able to store and represent amplitudes of all its 2^N possible configurations simultaneously. A fundamental problem in machine learning is that the joint distribution of N variables has exponential size. The structure of quantum amplitudes is very similar to that of joints so we can use the α_i to store our probabilities (up to normalisation; recall that our probabilities normalise to 1 but amplitudes normalise so that $\sum_i |\alpha_i|^2 = 1$), requiring only N qubits instead of exponential resources. We will try to set the α_i so that $\alpha_i = \frac{1}{\sqrt{Z}} Q(i)$ where Q is some probability distribution of interest over the space of binary registers I, and Z is chosen for correct normalisation of the amplitudes. Distributions Q of interest include the true joint $Q(i) = P(i)$ and the MAP-Dirac Delta joint, $Q(i) = \delta(i; \hat{i})$. The latter is of interest because it guarantees that an observation on the system will yield the MAP state, $|\hat{i}\rangle$. We may move gradually from a flat $Q(i)$ to $Q(i) = P(i)$ then to $Q(i) = \delta(i; \hat{i})$ via a series (parametrised by simulated 'temperature' T) of *cooled* distributions, $Q(i) = P(i)^{1/T}$.

We have seen that it is possible to represent joints in linear resources on a quantum register. The problem is how to bring the register into such a representational state.

Grover's method (Grover 1996) performs quantum unstructured search over N items in $O(\sqrt{N})$. However this algorithm requires the use of large operations over the whole space, applied in series. It also only finds target values whose target-ness is a function of their value only, rather than of their relationship to other values as in optimisation problems. In contrast, we present an algorithm using small local operators which may be applied in parallel during each overall sequential step, as in the local node updates of an annealed Gibbs sampler or Boltzmann machine (Hinton & Sejnowski 1986); and which seeks targets that are global and near-global MAP states.

Side-stepping the unitary requirement by extending the state space

The key idea is to construct and apply operators which transfer low-probability (high-energy) states to high-probability (low-energy) states. A difficulty with this is that quantum operators must be unitary. This means that as much amplitude must flow out of each state as flows in (similar to 'global balance' in Markov Chain theory). Initially this may seem to prevent our approach being possible, as the high-energy states must ultimately flow back to low-energy states.

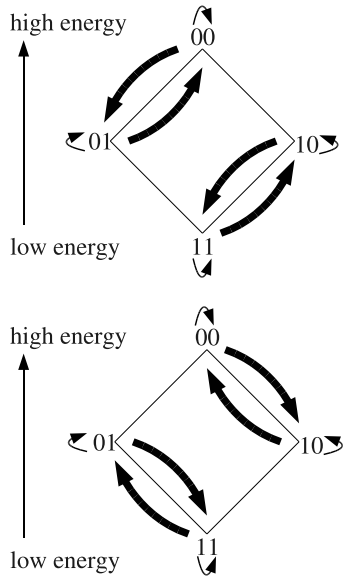


Figure 1: Top: A unitary operator that maps states to a superposition of themselves and the state reached by flipping their rightmost bit. Bottom: A similar operator for the leftmost bit.

This is indeed the case when a fixed-size system is considered. Consider fig. 1(top) and fig. 1(bottom), which respectively illustrate unitary operations acting on single qubits to flip the left and right qubits of a two-qubit register. In general, single-qubit operators may have some portion of self-transition as well, shown by the thin loops. The widths of the arrows illustrate the transition probabilities. (Note that this state transition diagram is not an MRF, but could represent moves between *configurations* of a Boolean MRF with two nodes.) Suppose the two variables are elements from an MRF whose configuration energies satisfy

$$E(|00\rangle) > E(|01\rangle) = E(|10\rangle) > E(|11\rangle)$$

where $E(|\psi\rangle) = \sum_i E_{ii}(|\psi\rangle) + \sum_{i,j} E_{ij}(|\psi\rangle)$ with the second sum over linked pairs of nodes.

In this case we would like to design operators that tend to make probability amplitudes flow away from the high-energy state $|00\rangle$ down towards the low-energy state $|11\rangle$. This is impossible with ordinary unitary operators on the 2-qubit space because unitary operators must have as much flow out of states as into them.

However by adding *new* bits to the system at *every* operation we will be able to make the state space grow and keep on providing new places for the accumulating high-probability states to flow to. The algorithm is reminiscent of – and inspired by – the idea of blowing a constant stream of *coolant* particles over a machine to cool it. The particles then flow away as *garbage* (or 'exhaust'), carrying away information (cf. Feynman's heat computers, (Feynman 1998)). They can then be ignored for the final measurement, but their existence is the key part of the algorithm. This process of spreading the superposition across a large garbage space is a form

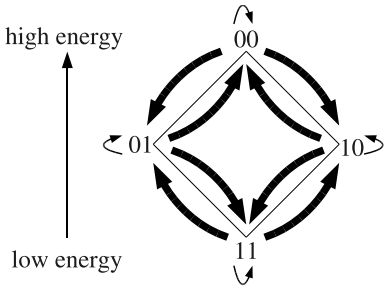


Figure 2: The same operator as fig. 1, plotted as a single figure. Note that unitarity is equivalent to the requirement that there is an equal amount of total flow into and out of each node.

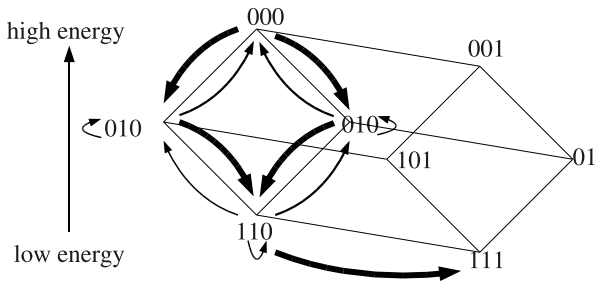


Figure 3: The rightmost qubit is new, introduced to extend the state space.

of *decoherence* (e.g. see (Nielsen & Chuang 2000)).

As we consider only operators which (like the Gibbs sampler) flip only single qubits (and/or don't flip them) we may depict whole collections of operators by a single diagram, as in fig. 2. Depending which of the bits is chosen to be considered next, one may read off the appropriate flow arrows without ambiguity.

Fig. 3 shows how the 2-variable system from fig. 2 may be extended with a garbage qubit to achieve the desired flows. This qubit is notated by the rightmost bit in the state labels, and is initialised to $|0\rangle$. Our node operators now act on the combined space of the node together with the garbage bit – i.e. the garbage bit's value may change as well as the node of interest. The figure shows how the low-energy state $|11\rangle$ may now act an attractor: the flow out of $|11\rangle\,|0\rangle$ does equal its flow in, but most of this flow is into $|11\rangle\,|1\rangle$, which maintains the $|11\rangle$ state of the nodes. We don't care about the resulting value of the garbage bit. For each operation a new, fresh garbage qubit, initialised to $|0\rangle$, is introduced to the system.

Because we are introducing *new* qubits at each step, we are able to control all of their initial states – setting them to $|0\rangle$. So we know that the initial complete state will have all its amplitude in the left half of the diagram. We can set up our unitary operator to have large flows from the left half to the right half, and vice versa. But we know that the flows from right to left (called 'shadow flows' and not shown on the diagram) will never actually be used. The states in the right half are called 'shadow states' and are never realised as initial states, only as resulting states.

Intuitive Explanation

The algorithm is based on a simplified analogy to physical atoms being bombarded by photons. Each new coolant is like a photon; each node is like an atom. At each step, one coolant and one node (and its neighbours) interact, and like Gibbs sampling, the node may or may not flip as a result.

If the energy of the coolant/photon is exactly zero and an energy emission is possible from the atom/node, then the node flips and energy {emission} occurs. If the coolant has zero energy but the node is unable to emit energy then {no emission} occurs.

If the coolant's energy is exactly the amount to raise the atom/node to a *higher* energy state, then the node flips and {absorption} occurs. If the coolant has energy but not the exact amount then {no absorption} occurs.[1]

The terms in braces above refer to cases in algorithm 1. Note that they refer to transitions between basis configurations of nodes and coolants: in the general quantum setting, nodes and coolants will be in a superposition of states, so all of these cases can occur simultaneously. As in classical Gibbs sampling, we would like to encourage the emissions to occur more often, and absorptions to occur less often, whilst still allowing energy increases to escape from local minima. We can control the rates of emissions and absorptions by manipulating the composition of the coolant superpositions. Initially we create 'hot' coolants with relatively high amplitudes of high energy states. Then we reduce the temperature to make the low-energy states more likely. Note that use of the term 'temperature' is analogical: the algorithm itself may run on a standard quantum computer.

As discussed above, all operators are required to be unitary. As well as 'energy bits', our coolants contain 'change bits' which act as in fig. 2 to construct shadow states and transitions. As before these states are never realised as inputs, but exist as outputs; they are a book-keeping device to allow unitarity to be maintained. Algorithm 1 is a constructive proof that this is possible.

Formal algorithm description

We consider Boolean MRFs so each node may exist in state $|0\rangle$ (false) or $|1\rangle$ (true). Initially, the system state space Ψ_0 is just the state space of the nodes:

$$\Psi_0 = X_1 \otimes X_2 \otimes ... \otimes X_N$$

This system state is initialised by an N-bit Hadamard transform H_N to a superposition of all possible configurations

[1] There is one special subcase where flipping the node would 'emit' zero energy, i.e. flipping makes no energy difference. In this case we make a superposed transition, {both flip and not flip} simultaneously. Amplitude thus disperses over equiprobable states, and this is required to escape from plateaux in the PDF. An alternative would be to always flip in such cases, but this would lead to undesirable random-walk MCMC behaviour as discussed in (Neal 1995).

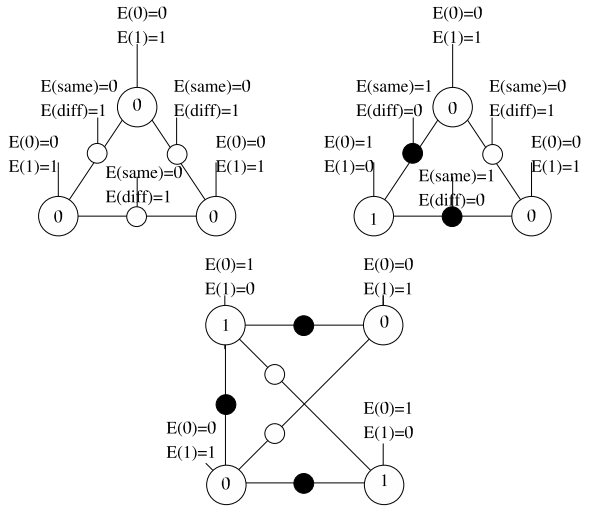


Figure 4: Top left: the $\hat{x} = 000$ test MRF. Top right: $\hat{x} = 010$ test. Bottom: $\hat{x} = 1010$ test. White and black links prefer same and differing nodes respectively. Nodes are labelled with optimal values.

(this may be though of as the limit of high temperature, the flat distribution):

$$|\Psi_0\rangle := H_N |00...0\rangle = \frac{1}{\sqrt{Z}} \sum_{\psi=0}^{2^N-1} |\psi\rangle$$

At each step s of the algorithm we extend the state space with a new *coolant* state space $(\Gamma_s \otimes C_s)$, so as the algorithm progresses, the space grows very large:

$$\Psi_s = \Psi_0 \otimes (\Gamma_1 \otimes C_1) \otimes (\Gamma_2 \otimes C_2) ... \otimes (\Gamma_s \otimes C_s)$$

Each coolant has two parts: $|\Gamma\rangle$ is generally made of several qubits, and is called the *energy register*. It is analogous to a fresh particle being blown over the system to heat or cool it; or to a photon striking an atom to transfer energy in or out of it. $|C\rangle$ is a single qubit and is called the *change bit* because we will use it as a book-keeping device to record whether the node changed state (allowing us to create 'shadow states' and to make our operators unitary). Initially each of the coolants are created in the state:

$$|\Gamma C\rangle = \frac{1}{\sqrt{Z}} \sum_{\gamma=0}^{\Delta_{max}E-1} \exp(-\gamma/T) |\gamma\rangle |0\rangle$$

The amplitudes follow a Boltzmann distribution. T is a temperature parameter, and is chosen so that $\sum_{\gamma=1}^{\Delta_{max}E-1} \exp(-\gamma/T) > 1$.[2] As in classical annealing, T will be gradually reduced. It should be follow a similar schedule as used in classical Gibbs; these schedules are generally heuristic. We will show that the quantum algorithm produces a speedup for the same schedule as its classical counterpart. $\Delta_{max}E$ is chosen to be the first power of two above the highest possible change in network energy that would occur from flipping one Boolean node, $\log_2 \Delta_{max}E = \max_{i,x_i,m_i} \lceil \log_2 \Delta E \rceil$, with $\Delta E = |E_i(x_i, m_i) - E_i(\bar{x_i}, m_i)|$, described below.

[2]This is to encourage more energy emission than absorption.

Algorithm

First construct an operator U_i for each node X_i as detailed in the next section and algorithm 1. At each iteration k, choose a random node ordering (as in classical Gibbs sampling). For each node X_i in the ordering, create a new coolant and add it to the system. The size of the state space thus keeps growing. Apply the node's associated operator U_i to the subspace consisting of X_i and its neighbours M_i. Ignore old coolants – they are garbage. When all nodes in the ordering have been operated on, lower the temperature according to the cooling schedule and progress to the next iteration. With appropriate hardware, each iteration's operations could be applied in parallel instead of random series.

The system converges to a superposition of local minima, and usually to a global minimum with high amplitude. Unlike the classical Gibbs sampler, all possible histories of proposal acceptances and rejections are explored in parallel by the superposition.[3]

Constructing the local operators

Before running the quantum iterations we first define one operator U_i corresponding to each node X_i. These operators are local in the sense that they act only on the reduced state space Ψ_i comprising the node X_i, its neighbours $M_i = \bigotimes_{X_j \in neigh(X_i)} X_j$, and a coolant system $(\Gamma \otimes C)$:

$$\Psi_i = X_i \otimes M_i \otimes \Gamma \otimes C$$

Define $E_i(x_i, m_i)$ to be the energy contribution due to the prior on X_i and the links to its neighbours:

$$E_i(x_i, m_i) = E_{ii}(x_i) + \sum_{X_j \in neigh(X_i)} E_{ij}(x_i, x_j)$$

Each U_i is constructed by algorithm 1.

Results

Our simulations are performed using the QCF quantum computing simulator for Matlab (Fox 2003). Due to the exponential resources required to simulate quantum devices, we are restricted in the present work to giving only proof of concept demonstrations. We consider the 3-node and 4-node Boolean MRFs in fig. 4. The first net has unit energy penalties for nodes with value 1 and for links whose neighbours are different, i.e. $E_i = x_i$, $E_{ij} = \Delta(x_i, x_j)$, so its MAP state is 000. The second net has asymmetric penalties and MAP state 010. The third is a more complex net with MAP state 1010. Fig. 5 shows the three operators constructed for the $\hat{x} = 010$ task.

[3]This is reminiscent of an 'exact particle filter', which adds new particles at each step to ensure all possible histories are explored. Information carried away by coolants could be used to recover the trajectory histories.

Algorithm 1 Pseudo-code for the construction of the U_i operator matrices. This is a classical algorithm which simply inserts numbers into matrices. A quantum algorithm will later apply these operators to the nodes and coolants. Refer to the text for descriptions of the various cases. QCF/Matlab versions of both classical and quantum algorithms are available from the author.

for each x_i configuration; each m_i configuration; each $\gamma = 0 : \Delta_{max}E$ **do**
  $\Delta E := E_i(\bar{x}_i, m_i) - E(x_i, m_i)$
  {network's energy gain by flipping}
  if $\Delta E \neq 0$ **then**
    if $\gamma = 0$ **then**
      {coolant carries no energy – possible cooling}
      if $\Delta E < 0$ **then**
        $U\,|x_i, m_i, 0, 0\rangle := |\bar{x}_i, m_i, -\Delta E, 1\rangle$
        {emission}
        $U\,|\bar{x}_i, m_i, -\Delta E, 1\rangle := |x_i, m_i, 0, 0\rangle$
        {shadow transition}
      end if
      if $\Delta E > 0$ **then**
        $U\,|x_i, m_i, 0, 0\rangle := |x_i, m_i, 0, 0\rangle$
        {no emission}
        $U\,|x_i, m_i, 0, 1\rangle := |x_i, m_i, 0, 1\rangle$
        {shadow transition}
      end if
    else
      {coolant carries energy – possible absorb}
      if $\Delta E = \gamma$ **then**
        $U\,|x_i, m_i, \gamma, 0\rangle := |\bar{x}_i, m_i, 0, 1\rangle$
        {absorption}
        $U\,|\bar{x}_i, m_i, 0, 1\rangle := |x_i, m_i, \gamma, 0\rangle$
        {shadow transition}
      else
        $U\,|x_i, m_i, \gamma, 0\rangle := |x_i, m_i, \gamma, 0\rangle$
        {no absorption}
        $U\,|x_i, m_i, \gamma, 1\rangle := |x_i, m_i, \gamma, 1\rangle$
        {shadow transition}
      end if
    else
      {equi-energy special case – flipping node has no energy effect}
      $U\,|x_i, m_i, \gamma, 0\rangle := \frac{1}{\sqrt{2}}\,|x_i, m_i, \gamma, 0\rangle + \frac{1}{\sqrt{2}}\,|\bar{x}_i, m_i, \gamma, 0\rangle$
      {flip and not flip}
      $U\,|x_i, m_i, \gamma, 1\rangle := \frac{1}{\sqrt{2}}\,|x_i, m_i, \gamma, 1\rangle - \frac{1}{\sqrt{2}}\,|\bar{x}_i, m_i, \gamma, 1\rangle$
      {shadow transitions}
    end if
  end if
end for

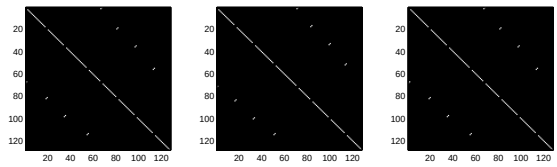


Figure 5: Operators U_0, U_1 and U_2 constructed for the $\hat{x} = 010$ task by the algorithm 1. There is one operator for each node. In this simple case the matrix representations of the operators are all of size 120x120. Matrix elements are always 0 or 1 (except in special equi-energy cases). After the operators are constructed, the quantum computing itself may run. Applying U_i to node i and its neighbours is analogous to a classical Gibbs step of updating one node.

Note that a naive state-vector-based computer simulation would require resources exponential in the number of iterations, due to the expanding state space from the new coolants (e.g. of size about 2^{50} after 4 iterations of the 4-node net). However as we don't care about the resulting states of the coolant – they are garbage – we may make use of the density matrix formulation of quantum computing (e.g. see (Nielsen & Chuang 2000)), which allows us to 'trace out' over these coolant states. This essentially provides us with the analogous operator to the 'sum' in the 'sum-product' algorithm (the 'products' corresponding to the unitary operators). Note that it is the lack of ability to sum out states – which reduces information – that prevents us from using quantum algorithms without garbage bits to do inference. Summing these traces uses the bulk of computing time on a classical computer simulation.

The performance of the quantum algorithm was compared to the analogous classical Gibbs sampler, whose acceptance probabilities are $\frac{1}{Z}\exp(-E/T)$. The same arbitrary cooling schedule was used in both algorithms, and was chosen prior to the experiments: at the kth iteration we set $T = \frac{1}{k+1}$. Each classical sampler was run for 100,000 trials, each trial initialised with a random configuration. The table below shows the empirical probability of the classical sampler (C) reaching the MAP solution after the final iteration, compared with the (deterministic) quantum observation probabilities (Q), for the three tasks with MAP targets 000, 010 and 1010. The quantum algorithm appears to lag the classical sampler for the first two iterations before overtaking.

	000		010		1010	
step	C	Q	C	Q	C	Q
1	.68	.51	.68	.43	.63	.35
2	.80	.76	.81	.70	.84	.72
3	.84	.84	.82	.86	.89	.92
4	.84	.87	.84	.92	.92	.98
5	.84	.89	.84	.94	.94	.99
6	.85	.89	.84	.96	.95	.999

Discussion

Comparison to other quantum methods

Quantum optimisation methods have featured in the physics literature in recent years (e.g. (Hogg & Yanik 1998), (Durr & Hoyer 1996), (Castagnoli & Finkelstein 2003)). However this work has not generally been framed in the context of discrete local quantum computation and the MAP-MRF task. The novel contribution of the present work is to give a purely quantum-computational algorithm for graphical model MAP optimisation (for example there are no continuous Hamiltonians, integrals, or *physical* temperatures in this paper). Recall that any graphical model can be reduced to a Boolean MRF which the algorithm may then operate upon. Our algorithm uses only local operators, in contrast to inference methods such as (Ricks & Ventura 2004) which apply Grover's algorithm to parameter search.

A recent development in quantum optimisation is the *adiabatic* method (Farhi *et al.* 2000), which uses no garbage bits and matches Grover's $\sqrt{N}$ unstructured search speedup (Roland & Cerf 2002). This speedup is dependent on prior information on the difference in probability between the first and second most likely states, and would not be achieved for general optimisation such as MAP inference. It is an open question how it could be exploited in those particular inference tasks where we do have some such information. All the above should be distinguished from operator formulations of classical message-passing (Rezek & Roberts 2003) and from 'quantum annealing' (Kadowaki 2002), a quantum-inspired search method for classical computers.

Comparison to classical Gibbs sampling

A fundamental problem in machine learning is that the joint over N nodes generally requires exponential resources to represent it – whether as a brute-force table or the classical Gibbs sampler's use of ensembles of N-node configurations over exponential time. We have shown how by using quantum amplitudes to represent joint probabilities, it is possible to represent the exponential-sized joint of N nodes instantaneously using only N physical nodes. However, moving the state vector into this representation is non-trivial and we showed how to use a unitary quantum analog of Gibbs sampling to get there. Further, we showed how it can be cooled towards the MAP solution, $Q(x) \to \delta(x; \hat{x})$.[4]

If amplitudes are considered to represent probabilities, then the algorithm can be thought of as behaving exactly as a classical annealing Gibbs sampler, with transition probabilities identical to the classical Gibbs sampler's. The difference between classical 'annealing' and our 'cooling' is that in the quantum domain, information cannot simply be discarded as in a classical computer, but must be accounted for within the unitary operations. The coolants are a method for producing annealing behaviour in the quantum domain, and achieve this by extending the state space at each step.

[4] $Q(x) \approx P(x)^{1/T}$ is represented implicitly as the marginal of the explicit $Q(\tau) = Q(x, \gamma_{1:s}, c_{1:s})$. The coolant space was introduced as it was not possible to carry out marginalisation explicitly by unitary operators.

Comparison to variational inference

Although we have presented the algorithm largely as a quantum extension of the classical Gibbs sampler, an alternative and interesting view is as a quantum extension of classical variational methods (Bishop 2006) such as Variational Bayes and loopy (Bethe) message passing. Such methods approximate the true joint $P(x_{1:N})$ by a product $Q(x_{1:N}) = \prod_{i=1}^{N} Q_i(x_i)$ which ignores correlation structure but is typically more tractable to compute. In the Gibbs analogy, we view our algorithm as a multitude of Gibbs samplers, running in 'parallel worlds'. An alternative view is to consider the amplitudes over individual node subspaces as being like variational Q_i factors. However they are extended from classical variational methods because they are entangled with the other nodes, so representing the correlation structure. A re-derivation of the algorithm from this perspective could be interesting future work.

The source of the quadratic speedup

The algorithm does not make use of phase cancellations (all amplitudes are positive real), so trajectories in the state space starting from each of the superposed initial states evolve almost completely independently of each other.[5] When we make a measurement after the cooling, we are therefore selecting just a single trajectory history to have occurred, and discarding the others.[6] It may be asked what advantage this has over the classical Gibbs method of simply sampling a single s-step trajectory $\tau = \{x_{1:N}\}_{1:s}$ without the need for quantum hardware – what is the use of the unobserved trajectories? The answer is that as we are representing cooled *model* probabilities $Q(\tau)$ by *amplitudes*, α_τ, our probability of *observing* a particular τ is given by the (renormalised) *squared* probability of its occurrence in the corresponding classical sampler, $Pr_{obs}(\tau) = \alpha_\tau^2 = Q(\tau)^2$. This gives a quadratic amplification of the probabilities $Q(\tau)$ from the classical sampler. This quadratic speed-up appears to fit in spirit with Grover's quadratic speedup, and suggests some rationale for the quadraticty: we may work linearly in the amplitude domain but then obtain a free quadratic amplification at observation. Full mathematical analysis of this idea could form the basis of future work. Such analysis could also try to explain why our first two iterations give poorer results than the classical sampler: we believe this is due to the sub-optimal nature of our sampling, as for simplicity we excluded the possibility of low but non-zero energy states absorbing energy.

Implications

We present the algorithm primarily as a potential quantum method to speed up the machine learning community's general task of MAP inference in graphical models, and to illustrate the quantum ability to represent complete N node joints instantaneously using only N physical nodes. Unlike many quantum algorithms it requires only small local

[5] The unimportant exception being the special plateaux case.

[6] The computation is reversible, so reachable basis states including coolants specify historical trajectories.

operators that would be easy to implement. Also unusual is the essential *requirement* for decoherence into coolants – rather than being a hindrance, decoherence is necessary to disperse information: due to the reversibility requirement, uncertainty is transferred out of the cooled nodes and into the coolants. Highly speculatively, the algorithm may be of interest to quantum biologists, as if it does provide a useful computational speedup then it seems *likely* (in the Bayesian sense) that evolution would have found a way to use it – though whether this is in fact the case is of course an empirical question.

Acknowledgements

Many thanks to Anuj Dawar, David Deutsch, Stuart Hameroff, Stephen N.P. Smith and Amos Storkey for discussions; and two reviewers for comments. This work was supported by an EPSRC studentship.

References

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Castagnoli, G., and Finkelstein, D. R. 2003. Quantum-statistical computation. *Proceedings of the Royal Society A* 459(2040):3099–3108.

Durr, C., and Hoyer, P. 1996. A quantum algorithm for finding the minimum, `quant-ph/9607014`.

Farhi, E.; Goldstone, J.; Gutmann, S.; and Sipser, M. 2000. Quantum computation by adiabatic evolution.

Feynman, R. P. 1998. *Feynman Lectures on Computation*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Fox, C. 2003. QCF: Quantum computing functions for Matlab. Technical Report PARG-03-02, Robotics Research Group, Oxford University.

Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 212–219.

Hinton, G. E., and Sejnowski, T. J. 1986. Learning and relearning in Boltzmann machines. In *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1: foundations*. Cambridge, MA, USA: MIT Press. 282–317.

Hogg, T., and Yanik, M. 1998. Local search methods for quantum computers, `quant-ph/9802043`.

Kadowaki, T. 2002. *Study of Optimization Problems by Quantum Annealing*. Ph.D. Dissertation, Department of Physics, Tokyo Institute of Technology.

Myllymaki, P., and Tirri, H. 1993. Massively parallel case-based reasoning with probabilistic similarity metrics. In K. Althoff, M. R., and Wess, S., eds., *Proceedings of the First European Workshop on Case-Based Reasoning*.

Neal, R. 1995. Suppressing random walks in Markov Chain Monte Carlo using ordered overrelaxation. Technical report, Dept. of Statistics, University of Toronto.

Nielsen, M., and Chuang, I. 2000. *Quantum computation and quantum information*. Cambridge.

Rezek, I., and Roberts, S. 2003. An operator interpretation of message passing. Technical Report PARG-03-01, Robotics Research Group, University of Oxford.

Ricks, B., and Ventura, D. 2004. Training a quantum neural network. In Thrun, S.; Saul, L.; and Schölkopf, B., eds., *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.

Roland, J., and Cerf, N. J. 2002. Quantum search by local adiabatic evolution. *Phys. Rev. A* 65(4):042308.

Shimony, S. E. 1994. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence* 68(2):399–410.

A Causal Bayesian Network view of Reinforcement Learning

Charles Fox
Robotics Research Group
Engineering Science
University of Oxford

Neil Girdhar
Google

Kevin Gurney
Adaptive Behaviour Research Group
Department of Psychology
University of Sheffield

Abstract

Reinforcement Learning (RL) is a heuristic method for learning locally optimal policies in Markov Decision Processes (MDP). Its classical formulation (Sutton & Barto 1998) maintains point estimates of the expected values of states or state-action pairs. Bayesian RL (Dearden, Friedman, & Russell 1998) extends this to beliefs over values. However the concept of values sits uneasily with the original notion of Bayesian Networks (BNs), which were defined (Pearl 1988) as having explicitly causal semantics. In this paper we show how Bayesian RL can be cast in an explicitly Bayesian Network formalism, making use of backwards-in-time causality. We show how the heuristic used by RL can be seen as an instance of a more general BN inference heuristic, which cuts causal links in the network and replaces them with non-causal approximate hashing links for speed. This view brings RL into line with standard Bayesian AI concepts, and suggests similar hashing heuristics for other general inference tasks.

Introduction

Reinforcement Learning

An MDP is a tuple (S, A, p_s, p_r) where $s \in S$ are states, $a \in A$ are actions, $p_s(s'|s,a)$ are transition probabilities and $p_r(r|s,a)$ are reward probabilities. The goal is to select a sequence of actions $\{a_t\}$ (a *plan*) over time t to maximise the expected value $\langle v_t \rangle = \langle \sum_{t=1}^{T} \gamma^t r^t \rangle$, where T may be infinite, and each action is selected as a function of the current, observable state $a_t = a_t(s_t)$. We consider the case where p_s and p_r are unknown. Classical Reinforcement Learning approximates the solution using some parametric, point estimate function $\hat{v}(s,a;\theta)$ and seeks $\hat{\theta}$ to best approximate

$$\hat{v}(s_t, a_t; \hat{\theta}) \approx \max_{a_{t+1:T}} \langle \sum_{\tau=t}^{T} \gamma^\tau r_\tau \rangle = \langle r_t + \max_{a_{t+1}} \hat{v}(s', a; \theta) \rangle.$$

It runs by choosing a_t at each step (which may be $a = \arg\max_a \hat{v}(s,a)$ if best available performance is required; or randomised for ad-hoc exploratory learning), then observing the resulting r_t and s_{t+1} and updating θ towards a minimised error value (with $w_0 + w_1 = 1$):

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

$$\theta \leftarrow w_0\theta + w_1 \arg\min_{\theta'} (\hat{v}(s_t, a_t; \theta')) - [r_t + \gamma \max_{a_{t+1}} \hat{v}(s, a; \theta)])^2$$

Bayesian RL uses a larger parameter set ϕ to parametrise and learn a full belief over values $Q(v|s,a;\hat{\phi}) \approx P(v|s,a)$. (So classical RL is a special case where this parametric probability function is assumed to be a Dirac Delta function, $Q(v|s,a;\phi) = \delta(v; \hat{v}(s,a;\theta))$)

Causal Bayesian Networks

A Directed Graphical Model (DGM) is a set of variables $\{X_i\}$ with directed links specified by parent functions $\{pa(X_i)\}$, and a set of conditional probabilities $\{P_i(X_i|pa(X_i))\}$ so the joint is $P(\{X_i\}) = \prod_i P(X_i|pa(X_i))$. A Causal Bayesian Network (CBN) is a DGM together with a set of operators $do(X_i = x_i)$ which when applied to the model, set $pa(X_i) = \emptyset$ and $P(X_i) = \delta(X_i; x_i)$. The do operators correspond (Pearl 2000) to the effects of performing an intervention on the system being modelled. A DGM of a system 'respects causal semantics' if its corresponding CBN faithfully models interventions. (The name 'Bayesian Networks' originally referred (Pearl 1988) to CBNs.)

While DGMs and CBNs are generally treated as complementary, we will show how a hybrid net with some causal and some acausal links is a useful way to think about Reinforcement Learning algorithms, and suggests generalisations for creating other approximate inference algorithms.

The model

Perceived backwards-in-time causality is possible

The general MDP problem can be drawn as the CBN shown in fig. 1(a). Each (s,a) pair causes a reward and the next state. The task is to choose a plan to maximise the expected value. Classical AI tree-search methods ignore the v nodes and literally search the whole space of plans, computing the expected summed discounted rewards for each plan. This is generally intractable. (Exact polynomial dynamic programming can be used for cases where the set of possible states is the same at each step, but the polynomial order is the number of dimensions of s which may be large, rendering exact solution impractical though technically tractable.)

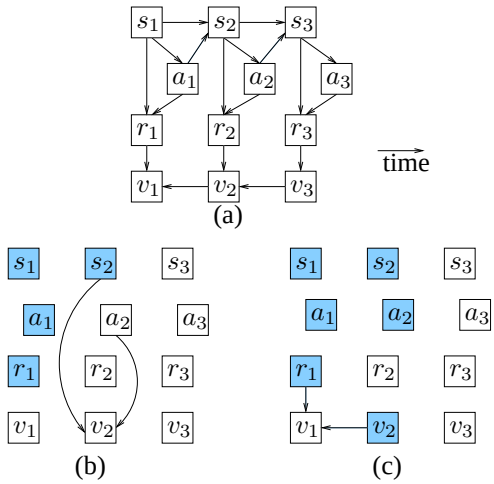


Figure 1: (a) Full inference. (b) RL action. (c) RL learning.

In the unknown p_s, p_r case they must also infer these probabilities from the results of their actions. Tree-searching can equivalently be conceived of as using the deterministic $v_t = \gamma v_{t+1} + r_t$ nodes to perform the summation. Inference proceeds as before: at each time t, for each plan (ordered by the classical search algorithm), we instantiate the plan nodes and infer the distributions over the v nodes, then perform the first action from the plan with the highest $\langle v_t \rangle$. Using message-passing algorithms (Pearl 1988), information must propagate all the way from the left to the right (via the a nodes) of the network and back again (via the v and r nodes) to make the inference. This is time-consuming.

The network respects causal semantics, and despite this it includes arrows pointing backwards through time. This is not a paradox: the expected discounted reward v_t at time t really is caused by future rewards: that is its meaning and definition. Note that v_t does not measure anything in the physical world, rather it is a 'mental construct'. So there is no backwards causation in the *physical* world being modelled. But there *is* backwards causation in the perceptual world of the decision maker: its percepts of future rewards cause its percept of the present v_t. For example, if v_t is today's value of a financial derivative, and we are able to give a delayed-execution instruction now that will somehow guarantee its value next month, then the causal do semantics would give a faithful model of this future intervention and an accurate current value. However value is a 'construct' rather than an object in the physical world, so no physical causality is violated.

Approximation method

We now present RL as a heuristic method for fast inference and decision making in the previous CBN. The two steps of RL, action selection and learning, (also known as the 'actor' and 'critic' steps) correspond to two DGMs derived from the CBN. We change the computation of the v nodes, from being exact to approximate expected discounted rewards. (Their semantics are the same, they still compute value, but their computational accuracy changes.)

In the action selection network shown in fig. 1(b), we sever all the causal links and replace them with two acausal links. So we assume an approximate, parametric distribution $Q(v_t|s_t, a_t; \phi)$. This is not a causal link: rather it is some function, whose parameters are to be learned, that will approximate the true causal $P(v_t|s_t, a_t, \hat{a}_{t+1}, ...\hat{a}_{t+N})$ where $\hat{a}_i$ are the actions in the optimal plan given a_t. It can be thought of as a 'hashing' link, making the best available guess under limited computational resources.

Once a_t is thus determined (though not yet executed) the learning step is performed. We remove the hashing links and reinstate the two causal links required to exactly update the parameters of v_t, shown in fig. 1(c). We assume that the best action was selected and will result in v_t from the action-selection step (which may be a Delta spike in classical RL or a belief distribution in Bayesian RL). The previous step's reward r_{t-1} is already known, so v_{t-1} is updated accordingly, using the standard Classical and Bayesian learning methods described earlier.

Following the learning step, the results of a_t are observed (r_t and s_{t+1}) and the actor phase begins again for $t + 1$.

Discussion

Exact inference for action selection in the exact CBN is time-consuming, requiring inference across the complete temporal network and back again to evaluate each candidate in a plan searching algorithm. We have seen how RL can be viewed as a heuristic which cuts the causal links of the CBN and replaces them with acausal hashing links, mapping directly from the current state to an approximate solution. We suggest that having viewed RL in the context of CBNs, a similar method could be applied to more general inference problems when cast as CBNs: isolate the time-consuming parts of the network and replace them with trainable acausal hashing functions. The process of switching between causal and hashing links can also be seen in the Helmholtz machine (Dayan *et al.* 1995) in a similar spirit to RL.

We also saw how the causal do semantics allow for coherent backwards-in-time causality, where the caused entities are perceptual 'constructs' rather than physical entities. This makes an interesting contrast to other conceptions of causality such as (Granger 1969) which assume that causality must act forwards in time.

References

Dayan, P.; Hinton, G. E.; Neal, R.; and Zemel, R. S. 1995. The Helmholtz machine. *Neural Computation*.

Dearden, R.; Friedman, N.; and Russell, S. J. 1998. Bayesian Q-learning. In *AAAI/IAAI*, 761–768.

Granger, C. W. J. 1969. Investigating causal relations. *Econometrica* 37:424–438.

Pearl, J. 1988. *Intelligient Reasoning with Probabalistic Networks*. Morgan Kaufmann.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Sutton, R., and Barto, A. G. 1998. *Reinforcement Learning*. MIT.

ThomCat: A Bayesian Blackboard model of Hierarchical Temporal Perception

Charles Fox
Robotics Research Group
Engineering Science
University of Oxford

Abstract

We present a Bayesian blackboard system for temporal perception, applied to a minidomain task in musical scene analysis. It is similar to the classic Copycat architecture (Hofstadter 1995) but is derived from rigourous modern Bayesian network theory (Bishop 2006), with heuristics added for speed. It borrows ideas of priming, pruning and attention and fuses them with modern inference algorithms, and provides a general theory of hierarchical constructive perception, illustrated and implemented in a minidomain.

Introduction

Real-time perception of musical structures is an exemplar of hierarchical temporal perception, and is a useful domain in which to demonstrate general theories of scene analysis which emphasize the temporal element. Time is notably absent in many other test domains for scene analysis, especially static visual scenes as opposed to video.

We use perception of semi-improvised musical performance from an 'almost-known' grammar as a specially selected minidomain to illustrate a theory of temporal hierarchical perception. A minidomain is a simplified but non-'toy' problem which is chosen to capture properties of a general task. Here we aim to illustrate ideas about general temporal perception and have chosen a relatively simple but non-trivial task in which to work. *Semi-improvised* music is generated by a performer from a probabilistic context *sensitive* grammar, based on a stochastic context-free grammar (SCFG) such as

$$\begin{aligned} p(SONG \rightarrow VERSE, CHORUS, VERSE) &= 1 \\ p(VERSE \rightarrow A, A, B) &= 0.75 \\ p(VERSE \rightarrow B, B, A, A) &= 0.25 \\ p(CHORUS \rightarrow K, L, K, L) = 1; p(A \rightarrow c) &= 1 \\ p(B \rightarrow c, g) = 1; p(K \rightarrow f, am) = 1; p(L \rightarrow f, g) &= 1 \end{aligned}$$

In addition to these context-free term-rewriting probabilities, music superimposes various global probability factors. The most notable is the global *key* which acts to prefer coherent sets of chords and notes throughout the whole musical scene. Similarly, melodic themes are often repeated throughout a performance, with a bias to preferring rewrite rules that have previously been applied. This is typically the case in jazz, rock and many non-Western art musics: a composition is a fluid entity consisting of recognizable chunks and conventions which are combined in real time by the player. At the structural level, performers may decide on the fly to insert extra verses and choruses if the performance is going well and the audience is wanting more; or to cut out sections if it is going badly. At the low level of chords, rhythm players are likely to improvise around the standard chords, with different changes having probabilities. An *almost-known* grammar means that the performance is generated from a grammar which is mostly similar to the grammar known by the perceiver, but may include additional and differing rules and differing probabilities. The perceiver must thus allow for the fallibility of its own knowledge of the grammar and be prepared to learn new rules on the fly.

The context-sensitive nature of the global factors, and the required allowance for one's own failure preclude the use of the fast dynamic programming algorithms used in simplified language processing tasks such as the Inside-Outside algorithm, and more general scene analysis techniques are needed. Indeed this was a reason for selection of the microdomain as representative of general scene analysis.

We consider musical scenes comprised of observations of low-level chords, assuming one chord per bar, and that chords are the terminals of the almost-known grammar, and influenced by global key. This minidomain task is enough to capture and illustrate the general context-sensitive and 'almost-known' requirements of general scene analysis. In particular we do not consider perception of melodies or individual notes in the present work. Our architecture, ThomCat, is largely inspired by the Copycat blackboard system (Mitchell 1993a), which itself was explicitly based upon (and cites) Ising models, Gibbs sampling and simulated annealing (Hofstadter 1987). This statistical mechanics basis is generally under-appreciated due to Copycat's use of non-standard terminology and heuristics. ThomCat is an explicitly Bayesian version of the Copycat architecture applied to the music minidomain. Creation and destruction of hypotheses by agents is translated into a simple rule which restricts

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the search space of the Gibbs sampler at each step to the neighbors of active nodes in the previous step. Rivalry is dealt with in the rigorous Bayesian sense of hypothesis comparison, and makes use of a central 'thalamus' lookup table for speed. ThomCat shows how to view and extend Copycat with modern machine learning concepts, and provides a useful method for high-level musical understanding. It may be of interest to music psychologists who could try setting its probabilities to model human perception; and to neuroscientists as a rough plan for some neural functions. While we focus on the domain of musical understanding, the architecture could be carried over to other forms of scene analysis.

Since Kant, perception has been seen as the process of interpreting low level sense data (e.g. pixels in an image or chords in a musical performance) *as* instantiations of known concepts in perceptual space (which may be spatial or temporal or both). The relativistic ideas of (Whorf 1956) emphasized that different subjects may have different sets of concepts through which to view the data, learned from different, unique sets of experiences. (James 1907) emphasized the Pragmatist view that the subject should learn and use the concepts which are the most useful in its typical environments. The use of our minidomain can provide insight into these ideas by showing quantitatively how they may be implemented. In our miniomain, the sense data consist of chord likelihoods in musical bars, and the concepts to be instantiated are the musical structures found in the grammar, along with the concepts of different musical keys.

A key distinction to be made is between perception *of* time and perception *in* time. Time in the former refers to the states of the world over time, while the latter refers to the times at which computations and inferences are made.

We give a model of temporal perception which maintains and beliefs about a *region* of time and updates them as new information arrives and is processed. We note than confusion between perception *of* and *in* has led to many apparent philosophical problems such as those raised by the experiments of (Libet 2004) in which future events appear to be perceived before they occur. Our model of temporal perception shares the *of* and *in* distinction with the Multiple Drafts analysis of (Dennett 1998) but differs in that like Blackboard systems, and Copycat in particular, it constructs a single coherent scene interpretation at each inference step rather than multiple drafts. We restrict on-line inference to a temporal window around the current time, 'freezing' the results outside it. This is a simple version of a 'spotlight of attention' (Crick 1984) and gives insight into how attention could work in general scene analysis.

ThomCat currently runs best using Gibbs sampling similar to Copycat (though extended to use cluster sampling), but we see that once Copycat's architecture is brought into the Bayesian network framework, the path is opened for other modern inference algorithms such as Variational Bayes to run on the same structures. The ability for unseen rules to be perceived in terms of free-floating known components provides a framework to further extend the architecture to learn new rules and probabilities.

Formal task specification

The task is perceive a sequence of incoming chord observations as the maximum a posteriori (MAP) single coherent structure which generated them, or as close an approximation to it as possible. We assume that a musical bar (measure) and chord detector module is available. At each bar t, we receive a set of chord likelihoods $\{(c_i, \lambda_i\}_{i=1}^{N_t}$, where c_i are chord symbols (such as *em*, *g*, *c*) and λ_i are likelihoods $\lambda_i = P(D|c_i)$ with D being the raw data received by the chord detector, such as a sound wave or MIDI stream. We allow for major and minor chords from each of the 12 possible root notes.

We are given a prior SCFG generation model, modified to become context sensitive with a set of key coherence factors and to allow for 'almost-known' structures generated by rules which are missing from the grammar. The SCFG is a set of rewrite rules of the form $X_i \rightarrow \{Y_k\}_k, p_i, \pi_i^{\emptyset}$ where X are symbols. If a symbol ever appears as the left hand side (LHS) X_i of some rule i then it is called a non-terminal. If it does not appear in this way it is called a terminal, and is assumed to be a lowest-level chord symbol such as *em*, *g*, *c*. The probability p_i is $P(\{Y_k\}_k|X_i,)$, that is, the probability that the LHS term is rewritten as the RHS sequence rather than as some other RHS sequence, given that the LHS already exists.

The $\pi_i^{\emptyset}$ values are *free-floating probabilities*, used in understanding 'almost-known' structures which allow the possibility of occasional new rules being used in the performance which are not known to be part of the grammar. We consider the case of new rules whose RHS are comprised of known symbols. To allow the possibility of learning such rules, we must first be able to perceive the sequence of RHS structures as existing in perceptual space, but with no parent structure (e.g. the K in fig. 2). Such percepts are called *free-floating*, and we allow each known symbol X_i to free-float with probability $\pi_i^{\emptyset}$. This is an important feature in general scene analysis tasks. For example, in visual scenes there is usually no explicit top-level grammatical structure, rather the scene is made of objects at arbitrary positions in space, which in turn are comprised of hierarchies of sub-objects. This is a key difference between scene analysis and language processing, the latter almost always assuming the existence of some top level 'sentence' structure which ultimately generates everything in the linguistic 'scene'.

We assume that the whole performance is in a single musical key κ, which affects the chord observations via compatibility factors, $\prod_i \phi(\kappa, c_i)$ so the total probability of a scene interpretation is given by the probability product of all the rewrite and floating rules used in the derivation, multiplied by this factor and normalized. We assume a flat prior over 12 keys (assuming relative major and minors to be equivalent).

We require quasi-realtime inference, meaning that complexity should not increase as the performance progresses. (Though as we present ThomCat as perceptual model rather than a practical application, we do not currently require it to run in real-time on a desktop PC). Scene analysis is generally used to make predictions of the near-future, so to illustrate this we will require a near-optimal prediction of the

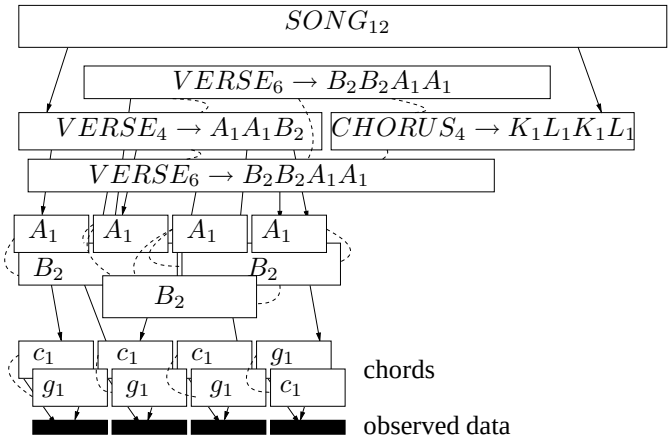


Figure 1: A network instantiated from the test grammar

musical structure of the next bar, and preferably a rough prediction of the structure of the entire future performance.

Architecture: static structures

We view the scene generation process as (causal) Bayesian networks as shown in the example of fig. 1. The existence of each musical structure (called a *hypothesis*), such as a *SONG* causes its components to exist, with a probability given by the grammatical rewrite rule. Each possible hypothesis can also exist as a free-floater, with a probability[1] $\pi^{\emptyset}$. The probability of a hypothesis is given by the noisy-OR function of its parents and its free-floating probability,

$$P(h|pa(h)) = 1 - (1 - \pi_h^{\emptyset}) \prod_{x \in pa(h)} I(x)(1 - P(h|x))$$

where x ranges over the parents $pa(h)$ of hypothesis h, and $I(x)$ is an indicator function which is 1 if hypothesis x is true and 0 if it is false.

The lowest level hypotheses are the chords reported by the external bar and chord classifier. To handle these in the same framework as the rest of the grammar, we construct a lowest-level set of nodes called 'observed bars' or *ObsBars* which represent the raw data observed by the external classifier. ObsBars then have transition probabilities conditioned on the reported possible chords c_i, $P(obsBar|c_i)$ which are set to the reported likelihoods from the classifier.

The dashed links in the figure connect pairs of mutually exclusive, or *rival* hypotheses. As a single coherent explanation of the scene is required, we cannot allow the existence of pairs of hypotheses which overlap in time and level of explanation. Formally, these links represent pairwise probability factors, $\phi(h_1, h_2) = 1 - I(h_1)I(h_2)$, which when multiplied into the scene joint distribution, set scenes with overlapping rivals to have probability zero, and have no effect otherwise.

[1] In our microdomain these are probabilities as the possible structure locations and sizes are discretized by the bars. In more general scene analysis this is not the case, and probability *densities* would be used over continuous space and size.

The hypotheses are of the form 'there is a structure of type X which exists in temporal space starting at time t and having length L'. In general, grammars allow the same type X to exist having different lengths, depending on how it becomes expanded later. In our framework this is important because such alternative expansions give rise to rival hypotheses such as the two $VERSE$ structures at the start of the figure, of lengths four and six bars. So hypotheses do not use the raw grammars as seen so far, but instead make use of compiled *lengthened* grammars (*l-grammars*), constructed from SCFGs. l-grammars split up SCFG rules which can exist in different lengths into sets of individual rules which make the lengths explicit, and convert the transition and free-float probabilities appropriately. Details of the l-grammar conversion process using an offline message passing algorithm are given in the appendix. Here we show the result of the l-grammar compilation for the previous example SCFG:

$$S_{16} \rightarrow V_4C_8V_4 : p = 1.0, \pi^{\emptyset} = 0.253$$
$$S_{18} \rightarrow V_4C_8V_6 : p = 0.5, \pi^{\emptyset} = 0.084$$
$$S_{18} \rightarrow V_6C_8V_4 : p = 0.5\pi^{\emptyset} = 0.084$$
$$S_{20} \rightarrow V_6C_8V_6 : p = 1, \pi^{\emptyset} = 0.028$$
$$V_4 \rightarrow A_1A_1B_2 : p = 1, \pi^{\emptyset} = 0.15$$
$$V_6 \rightarrow B_2B_2A_1A_1 : p = 1, \pi^{\emptyset} = 0.05$$
$$C_8 \rightarrow K_2L_2K_2L_2 : p = 1, \pi^{\emptyset} = 0.2$$
$$A_1 \rightarrow c_1 : p = 1, \pi^{\emptyset} = 0.05; B_2 \rightarrow c_1g_1 : p = 1, \pi^{\emptyset} = 0.05$$
$$K_2 \rightarrow f_1am_1 : p = 1, \pi^{\emptyset} = 0.05; L_2 \rightarrow f_1g_1 : p = 1, \pi^{\emptyset} = 0.05$$

The $\pi^{\emptyset}$ in the *unlengthened* grammar are here assigned automatically based on the hierarchical level of the rule. We assume that large-scale structures are more likely than lower-level structures to appear as free-floaters: the performer is very likely to play a top-level $SONG$ and is more likely to insert an extra $VERSE$ into that song than to insert extra A and B riffs into the $VERSE$ and extra bars of chords into those riffs.

In addition to musical structure hypotheses, we also model beliefs about the global key. There are 12 possible keys, each is either true or false, and all are mutual rivals so that only one key is allowed in a single coherent scene. Undirected links are added to the chord-level hypotheses, connecting them to the 12 key nodes and applying the pairwise compatibility potentials. A flat prior is placed over the keys by giving each key an equal $\pi^{\emptyset}$ probability factor.

Discussion of the static structure

The model is a causal Bayesian network in the sense that it respects Pearl's *do* semantics (Pearl 2000). It is a 'position-in-time' model rather than a 'dynamic state model', as it conceives of the nodes as existing in a temporal space, and makes inferences about what exists in each part of space. (In contrast, hierarchical dynamic-state models such as (Murphy 2001) would represent the explicit set of structures active at each individual time point. Algorithmically this can

be made to perform the same inference computations as the position-in-time model, but is less satisfactory as a theory of perception: position-in-time models construct an explicit phenomenal space and populate it with objects, which fits well with subjective perceptual experience.) We also note here the presence of much causal independence (Zhang & Poole 1996) in the network which might be exploitable by inference algorithms we choose to run on it.

The multiple hypotheses that may be instantiated from these lengthened rules are similar to those which appear in the context-free Inside-Outside algorithm, which instantiates *every* possible rule at *every* possible start time and *every* possible length. The l-grammar method is generally more efficient as it only ever instantiates hypotheses which can actually ever exist. For example, in our example grammar, there is no way that a $SONG$ of length less than 12 can ever exist, so it is wasteful to consider such hypotheses. This speedup is especially important when we move away from polynomial solvable context free structures to NP-hard context sensitive ones. Ideally, we would instantiate all possible hypotheses allowable by the l-grammar, giving rise to a single, large Bayesian network (not as large as that used by Inside-Outside – but lacking polynomial-time dynamic programming algorithms) and run inference on it. However such a network would still grow as $O(RT^2)$ like Inside-Outside, with R the number of lengthened rules and T the total number of bars, both of which are likely to be large in music and other domains of scenes, rendering even approximate inference impractical (Dagum & Luby 1993). Worse still, to generalize away from the music microdomain to continuous scene analysis domains such as vision, an infinite continuum of hypotheses would be needed which is clearly uncomputable.

Architecture: structural dynamics

Rather than instantiating all possible hypotheses from the outset, we turn to the classical AI architecture of Blackboard Systems for heuristics. Blackboard systems such as Copycat aimed to *construct* a single unitary percept by allowing multiple agents to add, change and remove parts of the percept. Each agent, when called upon, searches the blackboard for any changes it could contribute. If the changes improve the percept, they are made; if they make it worse, they may be made anyway, according to some heuristic, to help escape from local minima. Randomizing the orderings of agents to call upon also adds stochasticity which helps to escape such minima. In Copycat, the acceptance probabilities are made using simulated tempering, based on statistical mechanics. Factors in the scene (analogous to our rewrite probabilities, free-float priors, chord likelihoods, rivalry and key compatibilities factors) define energy contributions, and the objective is to minimize the energy. Changes proposed by agents are accepted with approximate Gibbs probability $P = \frac{1}{Z}\exp(-E/T)$ where T is a variable temperature parameter (Copycat's exact tempering equations are found in the functions `get-temperature-adjusted-probability` and `update-temperature` in source file `formulas.l` in (Mitchell 1993b)). Extending the simulated annealing algorithm to tempering, Copycat defined temperature as a function of the current 'stuckness' of the inference: if there is little change, suggesting a local minimum, then temperature is increased to escape from it.

Classical blackboard systems always maintained a single coherent state, even if that state is unlikely and is part of a path between local minima. Each proposal in sequence was either accepted and updated on the blackboard, or rejected and forgotten. In Copycat, proposals occur rapidly and repeatedly, resulting in a 'flickering halo' (Hofstadter 1995) of accepted proposals flashing briefly into existence around the local minimum. This set of nearby potential but usually-unrealized percepts is analogous to William James' notion of the *fringe* of awareness: the set of percepts which *could* be easily perceived if required – and which may now be quantified using psychophysical priming experiments, which show faster responses to fringe members than to non-members (Wiggs & Martin 1998). In Copycat and other classical blackboards, this fringe is an implicit notion, defined by an observer of the program to be the set of hypotheses which tend to flicker around the current solution.

ThomCat provides a new formulation of the fringe, derived from a heuristic to make its structures tractable. We discussed above how instantiating all possible hypotheses is intractable. But luckily there is some domain knowledge in scene analysis tasks that may inform a heuristic to limit the size of this search space *at each step* of inference. The implicit search space will remain the same, but at any point during inference, only a small subset of it will be instantiated and used to run inference. Recall that the task is to seek MAP solutions, not full joints. Consider a candidate solution set X of hypotheses that together explain the data (i.e. a set of nodes all having value 'true'). Now consider solution set $X' = X \cap x_d$ where x_d is a 'dead' hypothesis. A *dead* hypothesis is one with no 'true' parents or children. Recall that all hypotheses in our models have *small* free-float priors, in this case $\pi_\emptyset(x_d)$. Because the hypothesis is floating, setting its value to 'true' will always decrease the total network configuration probability. So as we only seek the MAP solution we know that we can exclude configurations with floating nodes from consideration and computation.

This fact gives a rigorous foundation for a *Bayesian blackboard* approach. We treat the hypothesis nodes as objects on a blackboard, which may be inserted and removed *during* inference. *Cues* in 'true' low-level hypotheses *prime* (i.e. instantiate) higher-level hypotheses and vice-versa, known as bottom-up and top-down priming respectively. This gives rise to a 'spreading activation' of hypotheses, starting at the lowest level of received observations, and spreading up the levels of abstraction, and down again to their top-down predictions. Some of these top-down predictions will eventually reach the lowest level of perception and provide us with the 'what chords comes next' prediction that is our ultimate goal. Note then that hypotheses from the exhaustive set now fall into *three* classes: instantiated-and-'true' (the current percepts, as in classical blackboards); instantiated-but-'false' (the fringe, now explicitly instantiated); and uninstantiated.

Priming may be done by any associative hashing-type function, which rapidly and approximately maps from a

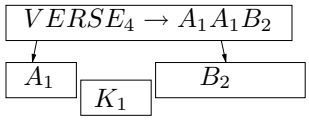


Figure 2: A missing child, and a floating replacement

central object to its fringe. In classic blackboard systems, Hopfield-like association networks and semantic networks were popular; classical neural networks are also useful hash functions for priming. The grammatical nature of the music minidomain allows an especially simple form of priming: when a (lengthened) term is asked to prime, we simply search the grammar for all rules containing it in their LHS or RHS, and instantiate them at the appropriate locations. Pruning (uninstantiation; 'fizzling' in Copycat) is performed in two ways. First, *Conway* pruning (after Conway's Game of Life) removes hypotheses which are 'false' and have no 'true' neighbors. Second, a garbage-collection sweep removes any resulting islands of nodes having no chain of connectivity to observed data.

In music as with most scene domains, it is unlikely that an object (hypothesis) would exist without one of its constituent parts. Initially it may appear that the model lacks a mechanism for penalizing such omissions. In a visual example, a $TABLE$ with a missing LEG is improbable. But if there was data and *instantiated* hypotheses only about three legs, and the scene probability was $\pi^{\emptyset}(TABLE)\prod_{i=1}^{3} P(LEG_i|TABLE)$, then this scene would be assigned a *higher* probability than a similar scene with data about four legs, because the latter scene would include an extra term in the product. To handle such cases, we may wish to include extra probability factors into the joint (as with keys) which specifically penalize the absence of components. The former visual scene would then be assigned probability $\frac{1}{Z}\pi^{\emptyset}(TABLE)\phi_{mc}\prod_{i=1}^{3} P(LEG_i|TABLE)$ where ϕ_{mc} is an explicit 'missing component' penalty. Early implementations of our system included such penalties, but they are in fact superfluous in the music minidomain because we are guaranteed that every bar will contain some chord, and require some explanation. Fig. 2 shows an example: the hypothesis corresponding to rule $VERSE_4 \rightarrow A_1 A_1 B_2$ is instantiated, but one of the A structures is missing, and a free-floating K takes its place in explaining that bar. However for the K to be 'true' it must have been generated by the free-float prior, which is small compared to the top-down prior on its rival A, so the scene as a whole does receive a penalty. In practice, the A will generally be instantiated anyway if the parent $VERSE$ is 'true' due to the fringe priming method, in which case a penalty will be received because $P(A|VERSE)$ is high so $1 - P(A|VERSE)$ is low.

Inference

We have used hypothesis states 'true' and 'false' informally, as the discussion has been on the network structure and construction. We now consider inference over the structures, which gives meaning to these node states. In the present implementation, hypotheses contain an abstract *inferlet* object, which provides a standard event-based API which may be implemented by various inference algorithms. Events include 'hypothesis just created', 'request made to run a local inference step (fire)', and 'tell me if you are 'true' or 'false".

The nearest inference method to classical blackboard systems is annealed Gibbs sampling, in which each instantiated node has a single Boolean state, updated according to the Gibbs probabilities given its Markov blanket. In this case, 'true' and 'false' are simply these Boolean Gibbs states at the moment the node is queried. We extend standard Gibbs to sample from *clusters* of coparents rather than single nodes. The rivalry links impose a strong form of causal independence on the network: we know a priori that pairs of rivals will never be on together, so we need not consider these scenarios – they are not part of the 'feasible set'. Care is taken to ensure that all probabilities are soft to ensure detailed balance of the sampler.

Unlike Copycat, ThomCat runs online, updating its percept on the arrival of each new bar (about 2s of music). Inference runs under an annealing cycle: at the initial arrival of the chord likelihood data, the temperature is raised to its maximum. Sampling and cooling run during the bar, and a schedule is chosen so that it reaches near-freezing just before the arrival of the next bar. At this point the Gibbs distribution is almost a Dirac Delta, at a local (and hopefully global) maximum scene probability. An additional 'collapse' step is performed which sets this to an exact Delta, yielding the MAP estimate.

We have implemented two other inference algorithms, Bethe and Variational Bayes (Bishop 2006), using different inferlet classes. These algorithms maintain local distributions $Q_i(h_i)$ over hypotheses h_i rather than simple Boolean states, and are updated according to the Markov blankets $mb(h_i)$. Variational Bayes locally minimizes the relative entropy $KL[Q||P]$ by node updates of the form

$$Q(h_i) \leftarrow \exp\langle \log P(h_i|mb(h_i))\rangle_{Q(mb(h_i))}$$

and the Bethe variational method assumes the Bethe approximation to the free energy leading to updates of the form

$$Q(h_i) \leftarrow \langle P(h_i|mb(h_i))\rangle_{Q(mb(h_i))}.$$

(The Bethe updates are the same as Pearl used in polytree-structured networks, but applied to the loopy case as an approximation.) In noisy-OR networks, the expectations lack analytic solutions and must be computed by brute-force summation over the $2^{|mb(h_i)|}$ configurations of the Boolean Markov blanket nodes (The QMR approximation of (Jaakkola & Jordan 1999) gives a method for faster approximation in particular cases of noisy-OR messages, which could perhaps be extended to ThomCat's networks). The notion of hypotheses being 'true' or 'false' as used in the priming mechanism now becomes vaguer, and we use the heuristic that hypotheses with $Q_i(h_i) > 0.5$ are classed as 'true' for this purpose only. Many typical grammars have troublesome symmetries in their generating probabilities, which can cause variational methods to get stuck between two solutions, or converge very slowly. Empirically it was found

that adding a small stochastic component to each node's $P(h_i|pa(h_i))$ at each step is a useful way to break these symmetries and allow convergence to a single solution.

Attention

We require that inference complexity does not increase over time as the scene grows, so we cannot keep growing the network and running inference on all of it as more and more data is received. Rather, it is better to focus *attention* on a fixed-size area of interest. Attention in this case means computing power used to run inference on nodes. It is not usually useful to focus on distant past events, rather we should focus on the region around the current events. However occasionally we may have to backtrack, as both current and past events are influenced by their shared high-level parents. It is also likely that in these 'mind-wandering' moments a higher temperature will be needed to break out of any minima: we must allow inference to become more fluid and risk-taking at these times. This could occasionally result in a minor 'paradigm shift' where the large-scale structure of the perceptual theory is overturned and replaced with something better. Similarly, for some tasks we may need to make time for 'blue-sky' thinking about objects in the distant future: when driving a car or preparing a jazz solo it is useful to have at least a rough map of where the whole journey is going in order to inform percepts and plans about immediate actions. But these distant past and future percepts change rarely – because little recently-received evidence is relevant for changing them – so do not need to be attended to often.

In the music minidomain, the region of interest consists of a few bars either side of the most recently received bar: this is where it is worth applying most effort to get good MAP approximations. We typically take six bars of the recent past, and two bars of the immediate future as the 'window' of attention. Hypotheses having any part inside the window are treated as active, and their inferlets participate in inference. Interactions between attention and priming and pruning need to be handled carefully. Fig. 3 shows an example of the window around the current time. It is important that the high level nodes such as $VERSE$ still receive likelihoods from out-of-attention past events that are 'true', such as the leftmost A structure. Such information acts as part of the prior when interpreting the current window. However, 'false' past hypotheses can be pruned once they fall outside the window, as they contribute no such prior and can never become 'true' again (unless a movable, 'mind-wandering' window was implemented – but in that case they would become re-primed when their region is attended to.) Similarly, new hypotheses are not primed outside the window, including in the future. We do however prime hypotheses in the *near* future inside the window, such as the K, f and am of fig. 3. Inference runs as usual on these hypotheses (though there are no *ObsBars* yet to provide bottom-level likelihoods) and the inferred state gives a prediction of the future, which could be used for example to schedule automated accompaniment actions to play along with the performer. The window moves along by one bar each time an *ObsBar* is received – at the same time as the annealing schedule is reset to its starting maximum temperature.

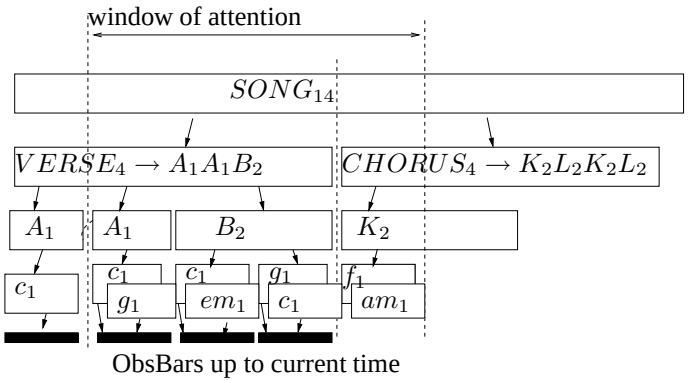


Figure 3: Attention includes the recent past and near future.

Practical issues

After naive implementation of the system with simple data structures and profiling, it became apparent that structural algorithms were consuming more processor time than inference proper, particularly during the single task of searching for pairs of rivals in a global hypothesis list. A large speed gain was made by constructing a central 'thalamus' topographical map of the temporal space, and using it to cache the locations of all hypotheses to allow rapid lookup of rivals. A similar performance gain was made by pre-caching hash maps of all possible primes from each l-grammar rule in a 'hippocampus' to avoid online lookup. The structures are so-called due to their possible suggested similarly to the corresponding biological structures. Once these speedups were implemented, the number of inference rounds per structural update was balanced so as to make the structural updates insignificant with respect to inference proper.

Results

Fig. 4 shows snapshots of the Gibbs state during a Gibbs inference run, using the grammar presented earlier. A deliberate mistake was introduced in the performance: the g chord in bar two has the highest likelihood, where the $SONG$ should generate a c. (a), (b) and (c) show annealing steps on the first five bars. The soft probabilities allow some pairs of rival nodes to be true together at high temperatures, but as the network cools this becomes less probable. Its final state (c) consists of a single free-floating $SONG$, causing a $VERSE$ component which causes riffs and chords. The erroneous g chord is perceived *as* a c due to large top-down pressures. The global KEY is identified as $Cmajor$, further encouraging c chords especially. (d) shows the state later in the performance, when attention has moved along with newly received bars, and now focuses on the new current bar, 17. A new round of annealing has just begun, so the window is at high temperature. Past hypotheses outside the window have been pruned if false, and frozen if true. In the latter case they still contribute likelihood to the large scale $SONG$ structures. The window thus contains a denser concentration of hypotheses undergoing inference, as the full fringe of primed objects is active. By this stage in the performance, the prior information from the past is very strong

so correct convergence occurs in just a few annealing steps following each new bar, if the performance coheres with the current percepts. The attention window includes two bars of the near future, which give the current prediction of what happens next. We do not yet have significant comparison results, but early trials suggest Gibbs tends to beat variational methods given similar time and hand-optimized annealing schedules. This is perhaps due to large correlations between node states, not well captured by variational methods.

Discussion

We have presented a model of hierarchical temporal perception in a musical minidomain. The model is intended to be generalizable to other forms of scene analysis – such as vision – and the minidomain was selected to include general features such as global context sensitivity and free-floating structures. The blackboard heuristics of priming and pruning are similar to those of Copycat, as are the Gibbs acceptance probabilities and annealing, but we have placed these ideas on a Bayesian foundation, viewing them as approximation heuristics to exact probabilistic MAP inference. The minidomain uses discretized bars of time which simplifies hypothesis construction, but similar architectures could prime and prune in a continuous space, for example in visual scenes.

The current implementation is slow, requiring more than real-time for accurate perception. This is partly due to lack of software optimization, but Gibbs sampling is inherently slow, with even Copycat's simple *micro*domain tasks taking several minutes to run. Switching from sampling to variational methods removes the need to visit long sequences of states but requires evaluation of exponential configurations of Markov Blankets. It is possible that the inference algorithms could be further tuned to improve accuracy, for example using Metropolis-Hastings steps instead of Gibbs at high temperatures; QMR-style variational approximations; adjusted annealing schedules and tradeoff between attention window size and accuracy. But the problem is inherently NP-hard so will always require some approximation.

With further work, the MAP scenes could be used to generate and schedule *actions*. The task of learning and tuning new rules from scenes containing free-floaters could also be researched. We touched upon the idea of movable 'mind-wandering' attention: this could be explored, for example if the listener realizes it made a mistake in the distant past, it may 'look' back to what happened, and *reconstruct* the past in the light of later data. Reconstruction is literal as we prune away most past hypotheses when attention moves on: so some limited memory of past events (either the low level data or the high-level structures that were constructed) could be used to start the inference, which could then also incorporate messages from later events.

We tentatively suggest that parts of the ThomCat architecture could be implemented neurally as theories of biological computation. The cortical column is hypothesized to function as some form of Bayesian network node, and we suggest an analogous role to ThomCat's structural hypotheses. This shows a clear distinction between 'mean-field' and 'spike coding' neural models analogous to variational and Gibbs inference. Gibbs-like 'spreading activation' has been observed in V1 and Thalamus. We found the use of a central 'thalamus' especially useful to speed up rivalry lookups, and an associative 'hippocampus' which primes fringe concepts ready for inference. Similar 'hippocampal' associations could be useful in storing episodic memories of the type required to reconstruct the past during 'mind-wandering' discussed above.

Like Copycat, ThomCat is intended to capture and refine concepts about human perception, including the nature of priming and pruning, attention, rivalry, and perception of time and in time. It is a relativist, constructivist model, in which different subjects may optimally perceive the same data *as* different scenes depending on their known concepts (in this case, grammatical rules) and priors within those concepts. A strange consequence of this approach to music perception in particular is that composition is a relatively simple task in comparison to listening well: a good listener is one whose prior grammar is accurate for the genre, and who does therefore not expect to hear unmusical scenes. Such a listener should then be able to compose within the genre simply by sampling forwards from the prior distribution – a much simpler task than NP-hard 'backwards' inference! (Fox 2006) used a ThomCat-like grammar to create variations on existing compositions parsed by human experts – this approach could perhaps be fused with ThomCat to automate the perceptual part of the compositional process – leaving the acceptance function still performed by humans or to be further automated.

Python source code for ThomCat is available under the GPL from `5m.org.uk`.

Appendix: lengthened grammar compilation To convert a SCFG to an l-grammar we use a novel message-passing scheme. First, a simple algorithm constructs all possible lengths for each symbol, and creates all possible lengthened rules, with no probability information. We then performing the following for each of these rules.

```
create top level hypotheses node from the rule
while any leave nodes contain nonterminals do
  for each leaf node l do
    x ← the leftmost nonterminal in l
    lookup all rules which rewrite x
    for each of these rules r do
      create child of l: RHS of r replaces x
    end for
  end for
end while
```

We then perform a top-down, dissipative (α) round of message passing followed by an upwards agglomerative (β) round with $\alpha_n = \alpha_{par(n)} T_{G(par(n) \to G(n))}$ and

$$\beta_n = \begin{cases} \sum_{m \in ch(n)} \beta_m & \text{: for non-leaves} \\ \alpha_n & \text{: for leaves} \end{cases}$$

where n ranges over nodes and we introduce the notation $G(par(n)) \to G(n))$ to indicate the rule R in the unlengthened grammar G whose LHS is the symbol in $par(n)$ which was substituted in the transition to the RHS of R. At the end of this process we obtain β values in the first layer of chil-

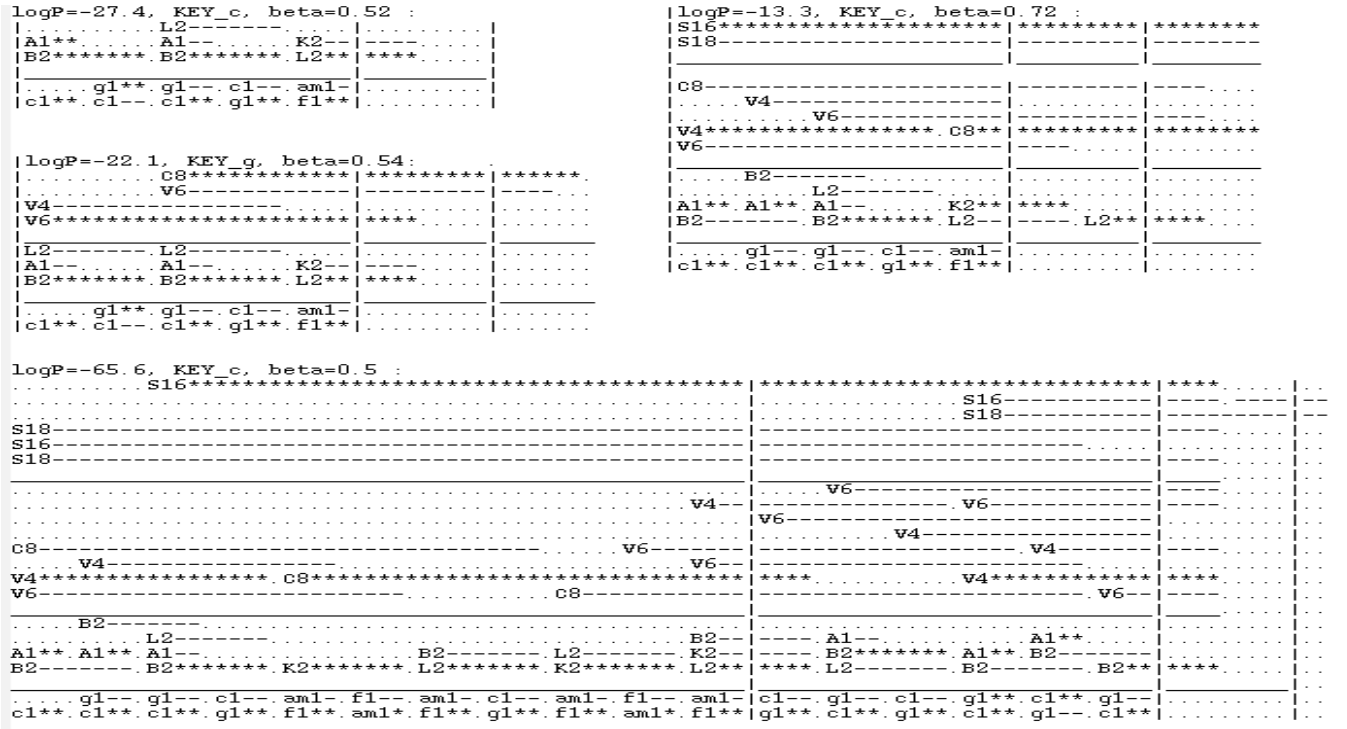


Figure 4: Snapshots of states during inference. Instantiated hypotheses are shown as hyphens (false states) and asterisks (true states). The vertical lines show the start of the attention window, the present time, and the end of the window, from left to right.

dren which summarize the the probabilities of those parameterizations of the top-level rule, and sum to 1 (so β at the root node is 1). The child β values are the required transition probabilities to insert into the lengthened grammar. Having formed the tree above and passed messages, it is simple to obtain the lengthened $\pi^{\emptyset}$ by $\pi_{\emptyset}(S_{9.5}) = \pi_{\emptyset}(S).\beta(S_{9.5})$. This works because the top-level $\beta(LR)$ for lengthened rules LR compute the prior $P(LR|R)$ where R is the corresponding unlengthened rules.

References

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Crick, F. 1984. Function of the thalamic reticular complex: The searchlight hypothesis. *Proceedings of the National Academy of Sciences* 81:4586–4590.

Dagum, P., and Luby, M. 1993. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence* 60(1):141–153.

Dennett, D. C. 1998. *Consciousness Explained*. Penguin.

Fox, C. 2006. Genetic hierarchical music structures. In *FLAIRS06*.

Hofstadter, D. 1987. A non-deterministic approach to analogy, involving the ising model of ferromagnetism. In *The Physics of Cognitive Processes*. ed. E. Caianiello. World Scientific.

Hofstadter, D. 1995. *Fluid Concepts and Creative Analogies*. Basic Books.

Jaakkola, T., and Jordan, M. 1999. Variational methods and the QMR-DT database. *Journal of Artificial Intelligence Research* 10:291–322.

James, W. 1907. *Pragmatism: A New Name for Some Old Ways of Thinking*. Reprinted Dover, 1995.

Libet, B. 2004. *Mind Time: the temporal factor in consciousness*. Harvard Press.

Mitchell, M. 1993a. *Analogy-Making as Perception*. MIT.

Mitchell, M. 1993b. Copycat source code. http://web.cecs.pdx.edu/ mm/ccat-src/.

Murphy, K. 2001. Linear time inference in hierarchical HMMs. In *NIPS*. MIT Press.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Whorf, B. L. 1956. *Language, Thought and Reality*. MIT Press.

Wiggs, C., and Martin, A. 1998. Properties and mechanisms of perceptual priming. *Current Opinion in Neurobiology (CNS)*.

Zhang, N., and Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*.

TOWARDS TEMPORAL INFERENCE FOR SHAPE RECOGNITION FROM WHISKERS

Charles Fox, Mat Evans and Tony Prescott
Adaptive Behavior Research Group
University of Sheffield

ABSTRACT

Previous work on shape recognition from whisker-like sensors has taken two approaches. First, static analysis from beam theory has been used to estimate contact points. Second, dynamic analysis has estimated contact points by considering vibrations set up on contact. This paper proposes additional methods which utilize full time-series data from whiskers. Full generative models may be intractable but approximated by discriminative classifiers. Early results show that object location and velocity may be discriminated simultaneously.

I. INTRODUCTION

The recognition of shape is an useful sub-task of robot navigation and other cognitive tasks. For example, commercial robot vacuum cleaners navigate using point collisions only (e.g. [1]) and could be improved by locating classes of objects in space rather than identical-looking points. The use of tactile whisker sensors for mobile robots has been explored by several authors [2], [3], [4], [5] and has advantages of functioning in foggy environments and of being covert.

The mechanics of whisker behavior are described by the theory of beams, as used in Civil Engineering [6] and briefly reviewed below. On making contact with an external object, whiskers bend, giving rise to measurable curvature at their bases. Whiskers also vibrate when they are stuck by object contact, as described by the theory of mechanical vibration analysis [7]. Static and dynamic analysis of beams are often considered as different fields of Engineering, and this divide has continued into whisker sensor research, yielding two distinct methods for shape detection based on the two theories.

This paper reviews static and dynamic whisker analysis then considers forward simulation models of whiskers and inverse models for obtaining single contact points from whisker sensors. It describes new generative and discriminative time-series approaches which may be capable of making inferences about shapes and motion rather than single points, and gives results from a preliminary discriminative implementation.

II. WHISKER MECHANICS

II-A. Static analysis

The beam theorem of Euler and Bernoulli. For a *pure* bending moment (i.e. pairs of forces M at the ends of the beam which rotate as the beam bends, and the center of mass pinned), a small segment of a beam of Young's modulus E and cross-sectional moment of inertia I bends circularly about a center of curvature at radius R, with

$$\frac{1}{R} = \frac{M}{EI}.$$

(A standard proof is given in the appendix). For example, one of the bending pair could be a contact force when a whisker is bent by an external object; the other bending force and the pin are provided by the base of the whisker. The *area* moment of inertia $I = I(x)$ is that of the cross-sectional *disc*, about the neutral axis. For homogeneous non-tapered whiskers it is constant along x but in general it could be a function of x.

Rather than work with radii of curvature, it is often useful to work with spatial derivatives. By geometry alone, we have for a circle of radius R,

$$\frac{1}{R} = \frac{d^2v/dx^2}{(1 + (dv/dx)^2)^{3/2}}$$

where v is orthogonal axis to x in the whisker's bending

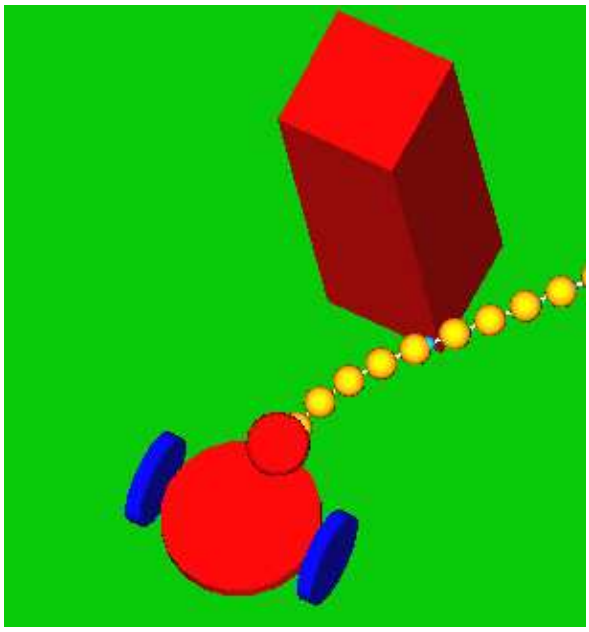


Figure 1. Physical simulation of whiskered robot.

plane. For small deflections dv/dx this reduces to

$$\frac{1}{R} \approx \frac{d^2v}{dx^2}.$$

The *Double integration method* may be used to find the shape of a bending whisker if we are given the moments applied at each point along it. (There is usually a single contact point, which excerpts a variable but effectively unlimited force to oppose the whisker pushing into it. Whiskers are small compared to most external objects so we ignore the effect of the opposite force acting on the object.)

$$v(x) = \int\int \frac{M(x)}{E(x)I} dx^2 + c_0$$

The constant c_0 is set by the boundary condition that the whisker base is fixed in a known position.

Whiskers typically make contact with objects at a single point only, due to their curvature. If a contact force W is applied at distance C from the whisker base then the moment experienced at each point x along the whisker is

$$M(x) = \{ \begin{array}{ll} W(C-x) & : x < C \\ 0 & : x \geq C \end{array}$$

Inserting this moment form into the double integration equation shows that the whisker bends cubically up to the contact point, and is linear afterwards:

$$\frac{EI}{W} v(x) = \begin{array}{ll} \frac{1}{2}Cx^2 - \frac{1}{6}x^3 & : x < C \\ \frac{1}{2}C^2x - \frac{1}{6}C^3 & : x \geq C \end{array}$$

The $x < C$ condition describes the shape from the base of the beam to the contact, and a small part of this region close to the base is typically observable by sensors. The cubic equation has two unknowns, W and C, so requires two observations to solve. These could be observations of the position $v(x)$ at two points, or its gradient and curvature,

$$\frac{dv}{dx} = \frac{W}{EI}(Cx - \frac{1}{2}x^2)$$
$$\frac{d^2v}{dx^2} = \frac{W}{EI}(C - x)$$

or other variables from which any of the above may be derived. Note that the position and gradient (but not curvature) are always zero at the base itself, so are not useful measurements. Kaneko [2] used base torque and an angle moved between first contact with an object and at its final location as such a measurement pair. Similarly, Birdwell et al. [3] use the base angle and torque; or alternatively the first temporal derivatives of the same.

If a series of two-variable data points are received, and it is assumed that the whisker is static at each, then a collection of contact points may be located, and used as the basis for shape recognition. In addition, each contact point also contains the information that there is no object in any of the space swept out by the whisker before reaching its static cubic spline. In particular this is useful for distinguishing smooth curved surfaces from sharp edges, as discussed by [4] and [5].

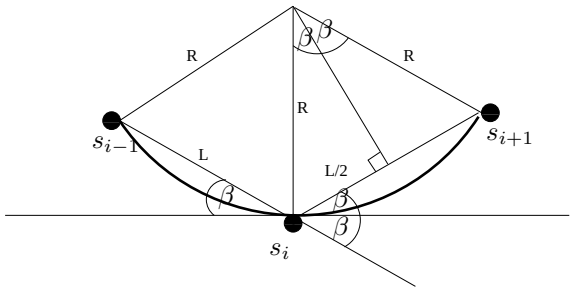


Figure 2. Construction used to derive spring-mass model from the beam theorem.

In practice, for real-world whisking speeds, the whisker is not static at each point, but carries kinetic energy which affects its trajectory. One approach to shape recognition is to estimate the noise level in sensor measurements from their equilibrium values introduced by these deviations from equilibria. The estimates of the contact point coordinates $(C, v(C))$ may then be computed together with error bounds propagated from the sensor uncertainties. There appears, qualitatively, to be a trade-off between speed of whisking (and thus the number of contact points gathered) and the accuracy of the points due to kinetic effects: a whisker driven infinitely slowly would be perfectly accurate. However this approach appears to be throwing away information, as we may have good models of the whisker dynamics and thus be able to account for the 'noise' in terms of the actual dynamic state of the whisker over time.

II-B. Dynamic analysis

Some previous work has considered the use of *simple* dynamic models. Kaneko [8] analyzed a massless beam with a large mass at the tip. When such a beam strikes a contact point, it oscillates in two distinct phases. First – with the beam in contact – the region from the contact point to the tip oscillates with fundamental frequency

$$\omega(C) = \sqrt{\frac{k(C)}{M}} = \sqrt{\frac{12EI}{ML^3(4 - C(3-C)^2)}}$$

where k is the effective spring constant. In the second phase, the beam is not in contact, and oscillates at the natural frequencies of the whole beam. Transitions between the phases may be observed in the base torque, and the contact location deduced from them, under the important assumption that all motion is at stable natural frequencies. Kaneko [8] then proceeded to a non-massless beam, deriving similar equations to estimate C.

In particular this method assumes that there are no transient components due to the initial contact. A low-pass filter was used by Kaneko to remove this 'noise' – which may in fact contain useful information. As with the previous static methods, there is no notion of uncertainty in the estimate. Beams without large masses at the tip often produce ambiguous, bi-valued estimator functions, and the wave equation used in the analysis assumes only

small displacements, which are not valid for large deflections as often experienced by real whiskers. The method must also wait for a complete cycle of both phases to begin making estimates. As with static methods, it is unable to handle moving objects.

Non-generative dynamic models have searched for empirically useful features in the frequency domain of strain signals to predict contact surface properties, usually with regard to classifying texture rather than contact location. In the texture case, it has been noted [9] that the 'ringing' transients produced as the whisker makes and leaves contact with the surface contain useful information. This is the same part of the whisk signal discarded as 'noise' by Kaneko.

III. LIMITATIONS OF CURRENT MODELS

We have reviewed various methods for gaining information about contact location. Static analysis assumes that each data point is at static equilibrium, and discards dynamic effects as noise. State-of-the-art dynamic analysis assumes that all oscillations are at stable natural frequencies, and discards transients as noise; it also only makes use of one sensor. Heuristic classifier-based approaches have found information in transients but do not fuse it with all other available information. All previous methods make point estimates only, not handling uncertainties in various sensors or allowing the possibility to fuse information from multiple sensors. No previous approach has handled moving targets. Other dynamic effects ignored by previous approaches include surface properties such as collision elasticity and surface friction. We thus propose a class of generative time series models capable of accounting for all these effects. We will outline the form of these models, then consider computational inferential and discriminative approximations.

IV. FORWARD SIMULATION

We have constructed a simulation model of a whisker as a series of masses on rotational springs, based on the Open Dynamics Engine (ODE; `www.ode.org`), as shown in fig. 1. Such systems consist of more than two bodies so lack analytic solutions. The number of segments, N, is an important parameter as it controls the complexity of the simulation (and later, of inference) and affects the spring parameters used. Spring constants can be derived as follows. Consider approximating the curved beam with a series of length L segments $\{s_i\}_{i=1}^N$ as in fig. 2. The continuous curvature at segment s_i is now replaced by deflection angle $\theta_{i,i+1} = 2\beta$ at each segment. (We write $\theta_{0,1}$ for the angle between the whisker base on the robot and the first segment.) By geometry:

$$R \sin \beta = \frac{L}{2}$$

for small β:

$$R\beta = \frac{L}{2}$$

$$R = \frac{L}{\theta_{i,i+1}}$$

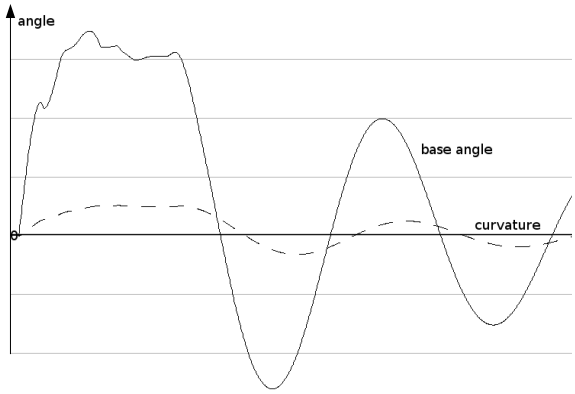


Figure 3. Base angle $\theta_{0,1}$ (solid) and curvature $\theta_{1,2}$ (dashed) during contact with a moving object.

Substitute in from the beam theorem:

$$M = \frac{EI}{L}\theta_{i,i+1}$$

This is now in the form of a rotational spring equation, with

$$k = \frac{EI}{L}$$

The springs also have damping coefficient γ. This may be set using the *Q-value* of an isolated segment, which for light damping is the number of oscillations performed during each loss of $1/e$ of the oscillator energy:

$$Q = \frac{m}{\gamma}\sqrt{\frac{k}{m} - \frac{\gamma^2}{4m^2}}$$

When Q is zero or imaginary the system is critically or heavily damped and no oscillations occur. For light damping (small γ):

$$Q \approx \sqrt{m}\frac{\sqrt{k}}{\gamma}$$

The segment Q-value could be computed analytically from beam theory, measured empirically from a real whisker segment of length L, or – for simulations – simply prescribed. In particular its is often useful to prescribe critical damping.

Collision detection is handled by ODE, as are constraint forces to hold joints together. However we have found that such whisker systems are extremely sensitive to numerical errors and require small time steps to simulate correctly, and we note empirically that there appears to be a monotonic relationship between the joint Q-value and the number of steps per simulation second to avoid error; and between the number of whisker segments and the steps per simulated second. The problem can be reduced to some extend by introducing linear spring elements alternating with the rotational springs. This reduces the magnitude of the required Lagrangian constraint satisfaction forces which are the principal cause of numerical instability. We are considering the design

of a custom ODE joint which combines rotational and linear springs to reduce error further.

The simulation allows us to record sensor-like observations, for example of base angle $\theta_{0,1}$ and base curvature, approximated by $\theta_{1,2}$. It also allows for small random noise forces to be applied at all segments, as would be experienced due to wind and self-motion of the robot body.

Figs. 4 and 3 shows the screen-shots and simulated base angle and curvature for a passive whisker being stuck by a *moving* square object, whose motion is perpendicular to the line of the whisker. The first oscillation includes the contact period; the others occur after the object has passed by the whisker. The shape, contact location and motion of the object affect the time series during the contact period and the subsequent oscillations. (Real whiskers have a much lower Q than simulated here; however this requires much longer computation time to simulate as discussed above, so for development purposes a low Q is used.) Note in particular that in step 5 of fig. 4 the whisker has received large momentum from the object and continuous to move away from it, before springing back to make contact a second time in step 6.

V. INVERSE TIME-SERIES INFERENCE

Fig. 5 shows a slice of a dynamic Bayesian network model for the whisker system. It consists of segment states s_i which each comprise current position and velocity; contact object state c which also comprises position and velocity along with other object properties such as shape, rotation, surface elasticity and friction. The observed base angle $\theta_{0,1}$ and curvature $\theta_{1,2}$, and small noise nodes representing additional forces applied to the segments and contact object. Two time slices are shown horizontally; in reality the model consists of many such slices over the simulation time. The transition relation for the contact object is such as to update its position according to its previous velocity; and allow for changes in velocity according to some prior, as well as any noise forces acting on it. The transition relations for the segments are complex as they include constraint satisfaction forces, but may be modeled implicitly by ODE physics simulation steps. Similarly the observations are obtainable from such simulations.

Drawing samples from the model is simple, and effectively amounts just to running the ODE physics simulation, with appropriate random noises added. A much more challenging task is to infer the most probable explanation of given observed time series. Exact inference is clearly intractable due to the loopiness, continuous-valued nodes, and highly non-linear transitions. However approximate inference by tracking of hypothesis subsets (i.e. particle filtering) [10] may be possible. The whisker domain is especially interesting as the number of segments in the model may be varied, trading off physical accuracy for the number of variables (and hence dimensions) represented in each particle. After drawing random noises, the transition steps can be performed deterministically by calling out to the ODE physics engine to simulate the evolution of each particle.

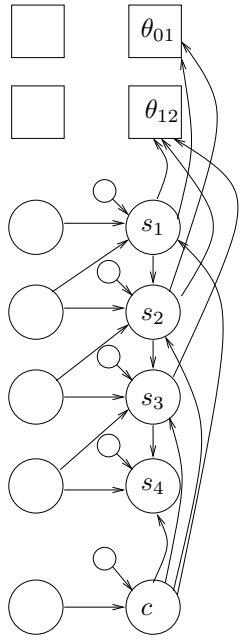


Figure 5. Dynamic Bayesian network slice for the physical whisker model. Small nodes are noise sources; square nodes are observations.

VI. DISCRIMINATIVE APPROXIMATION

We have shown how to set up inference from whiskers as a dynamic Bayesian network based on physical simulation, and suggested that approximate particle filtering inference may be useful. However it is an open question whether any such inference is feasible in real-time. An alternative approach is to construct and train discriminative functions – such as neural networks – mapping directly but approximately from the time-series data to the object state. As well as potentially replacing full inference models, such discriminative methods may be useful as heuristics within them: for example by suggesting new hypotheses to 'inject' into particle filters [11].

VI-A. Template-based discrimination

To demonstrate that at least some useful information is quickly obtainable from the time series by discriminative methods, we here present baseline results for estimating the contact distance d to a moving object using the curvature, $\theta_{12}(t)$, series only. The object is the square box moving at a vertical, fixed velocity as shown in fig. 4. The only variable is the horizontal distance of the box from the whisker base.

Fig. 6 shows examples of curvature series for three contact distances. (Time is unit-less but steps may be though of as milliseconds.) The series make reasonably smooth shape transitions as the contact distance varies. Series were obtained for distances of 1.5cm to 3.0cm, in 1mm steps. (Contacts closer than 1.5cm are difficult to simulate and were discarded.) The first, last and

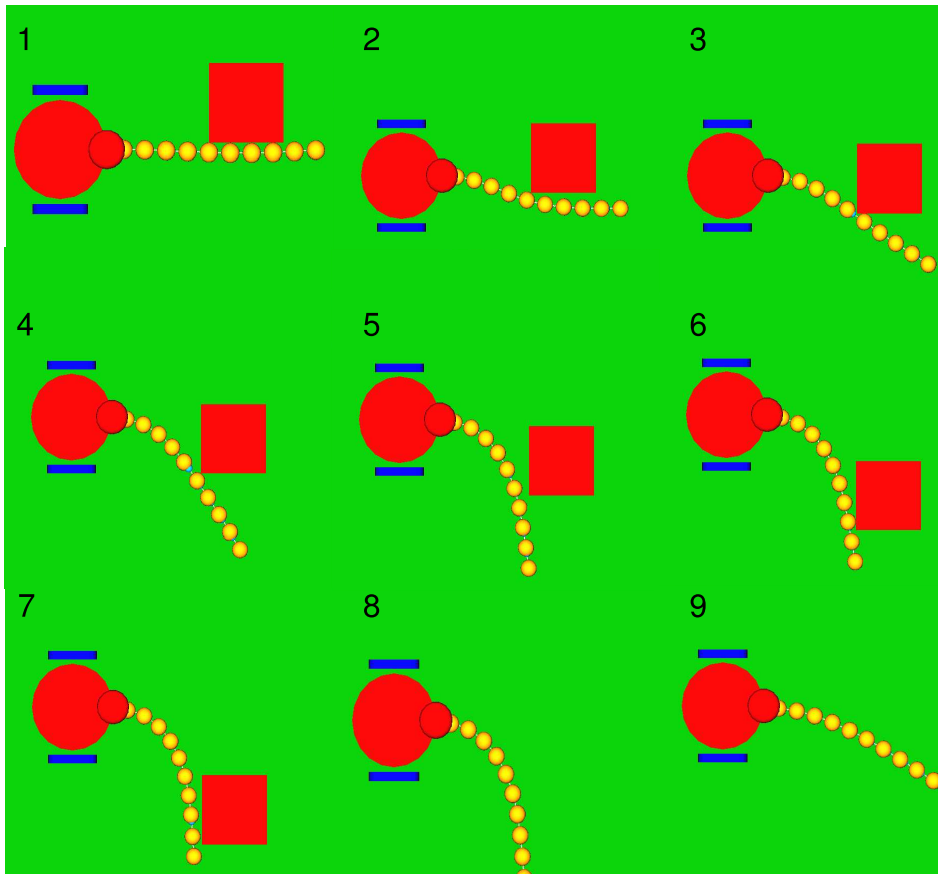


Figure 4. Screen-shots during contact with a moving object. The whisker length is 3cm.

central (2.3cm) points were used as template models. The remaining points' contact distances were then estimated as follows. Define the responsibility $\lambda_o(i)$ of template $m_i(t)$ for an observed series $o(t)$ as the Gaussian

$$\lambda_o(i) = \frac{1}{Z} \exp - \frac{\sum_t (m_i(t) - o(t))^2}{2\sigma^2}$$

where a (unit-less) variance of 5 was chosen by hand as a useful value and Z normalizes the responsibilities over i. Responsibilities for the observed contact distances are shown in fig. 7. Taking the average of the template models' contact distances – weighted by the responsibilities – as an estimate of contact distance, gives the model estimates shown in fig. 8, which have a mean error of 1.2mm.

In a second experiment, both the contact distance and the object vertical velocity were varied together. Nine templates were used, at combinations of minimum, medium and maximum of the two variables, and the same methods as above was applied to estimate both parameters simultaneously. The estimated contact distances *across all velocities* are shown in fig. 9 and the estimated velocities *across all contact distances* are shown in fig. 10.

This is the first published result to our knowledge on simultaneous contact distance and object velocity

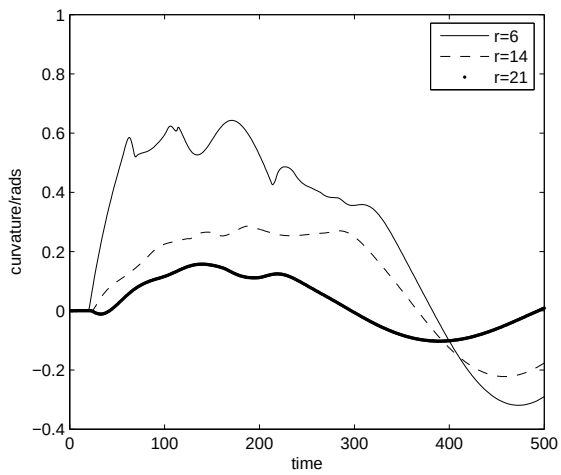


Figure 6. Strain series for three contact distances.

estimation – all previous methods have considered static objects only. It is intended as a proof of concept to show that useful information is contained in whisker time series. However the degradation of accuracy using the simple template method when a second variable is included suggests that this simple method may not

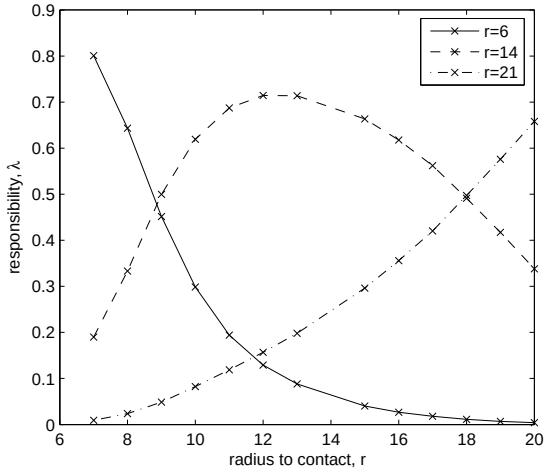


Figure 7. Responsibilities of the three template models for observations at various contact distances (in cm).

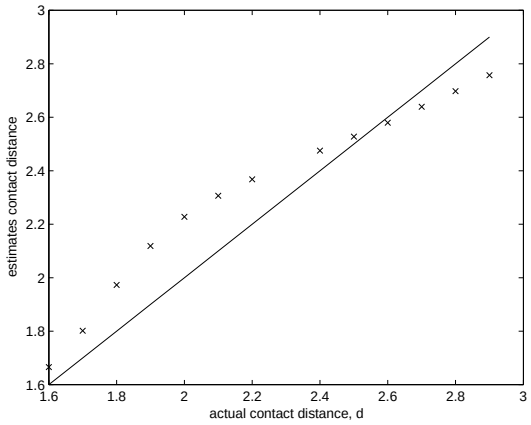


Figure 8. Model contact distance estimates (crosses) against actual contact distances (line), in cm. The mean error is 1.2mm.

generalize to large numbers of object parameters, and hence more detailed models should be more useful.

VII. DISCUSSION

To build and test both generative and discriminative models, it is useful to work with a series of tasks of increasing difficulty. In this case, difficulty mostly corresponds to the size of the parameter space. In addition to reducing the number of segments used in the generative model, the dimension of the contact object may be restricted. A simple scenario is to assume that the only type of object in the world is a frictionless, perfectly elastic plane, moving towards the whisker at known constant velocity. Such an object has only three dimensions: initial shortest distance from the whisker base and rotation. Alternatively, the object could be stationary and the whisker moving *rotationally* into it. Simple shapes such as squares and circles are a little harder, having extra configuration dimensions.

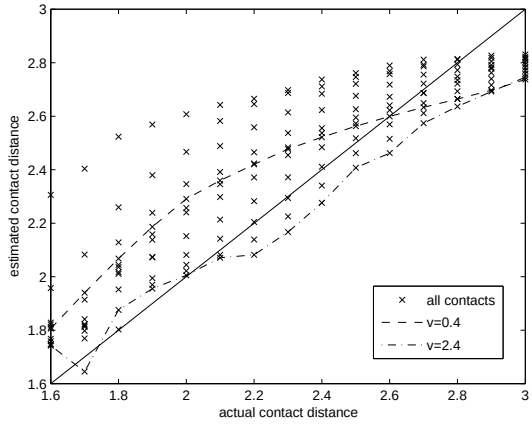


Figure 9. Ground truth (solid line) and estimated (crosses) contact distances d (in cm) under varying velocity (in cm/ms). The minimum and maximum velocities curves are shown (dashed lines). The mean error is 2.0mm.

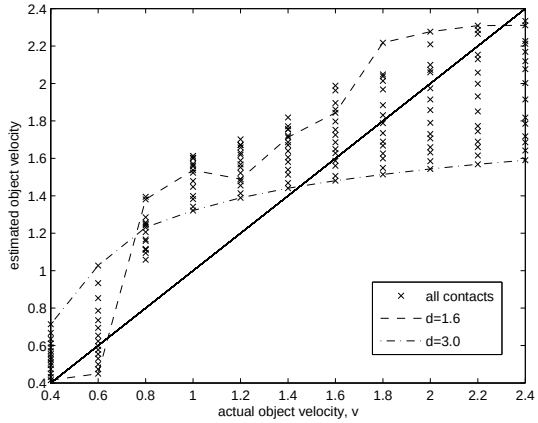


Figure 10. Ground truth (solid line) and estimated (crosses) velocities v (in cm/ms) under varying contact distance. The minimum and maximum contact distance curves are shown (dashed lines). The mean error is 0.28mm/ms.

Many interesting collisions in the real world involve three physical dimensions rather than the 2D collisions produced by simple shapes. For example the act of 'plucking' a whisker (like a musical instrument string) involves pushing the whisker back but simultaneously lifting the object upwards, away from the whisker. Rather than use a full 3D simulation (which massively increases the hidden state space size), a close approximation is to restrict the whisker motion to the horizontal plane while allowing contact object to move in 3D space.

The linear template model demonstrated here is a crude form of discrimination, and more advanced template-based discriminators exist such as classification on Principal and Independent Component strengths. Template models in general are not invariant to shifts

in the time domain, so a different class of Markovian discriminators may be useful instead. These include Hidden Markov Models and recurrent neural networks, which may be trained to filter time-series data into class labels online. Such models have similar dynamic network structure to the full generative model, but are acausal, more easily trainable, and faster to run. In particular, the structural similarity may allow some of their parameters to be set or constrained by knowledge about the generative model.

Previous whiskered shape recognition systems yield only point estimates of contact locations. In contrast generative methods (and discriminative approximations to them) can produce a posterior belief over the configurations of whole objects in the world. These beliefs may then be used as input to higher-level systems. For example localization and mapping tasks [10] and higher-level object recognition from parts [12] are improved by allowing for uncertainty. Uncertainty may be reduced by fusing information from multiple whiskers.

VIII. SUMMARY

Previous approaches to whiskered shape recognition have used only limited subsets of information available from the whiskers. They have further neglected aspects such as the motion, elasticity, and friction of objects. Their outputs have been point estimates of contact points with no posterior distributions, and they have not taken account of uncertainties in sensors or noise in the environment such as wind and self-motion effects.

We have shown how to set up the task as a full generative temporal model, which may in theory account for all available time-series data, and make inferences about the additional object and environment properties. Unusually, the model makes use of an external physics engine to make deterministic state transitions once simple noise terms are drawn. We have shown how the whisker beam can be approximated in this model by a series of masses on springs and discussed some practical simulation issues, in particular noting an empirical trade-off between the computational simulation time and the simulatable stiffness of the whiskers. We propose the use of particle filters for generative inference and for calibrating and testing discriminative methods. Once trained, the latter may in turn suggest new particles to inject into online particle filters.

The utility of full time-series data in recognition was demonstrated by our results showing simultaneous recognition of contact distance and velocity. All previous approaches have assumed static objects. Our result was achieved by a simple template-based discriminator, but suggests that more advanced discriminative and generative models will also be useful and possibly more accurate.

The simulation is available under GPL license from `sourceforge.net/projects/freebots`.

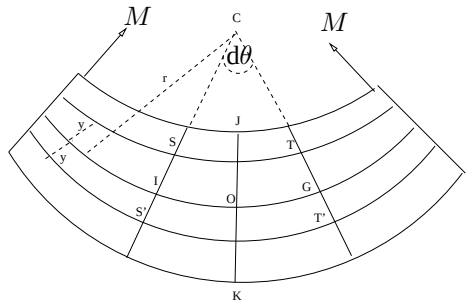


Figure 11. Construction on cross-section of a beam, used to derive the beam theorem.

APPENDIX

Consider fig. 11 which shows the bending. For *pure bending*, the forces rotate with the beam as it bends so they are always perpendicular. The top of the beam experiences negative strain (due to compression stress) and the bottom experiences positive strain (due to tension stress). There will be some 'neutral axis' IOG whose length is unchanged. ST is a short, thin slice at height y from the neutral axis. Consider the strain ϵ_{ST} at ST:

$$\epsilon_{ST} = \frac{\Delta ST}{ST_0} = \frac{ST - IG}{IG} = \frac{(R-y)d\theta - Rd\theta}{Rd\theta} = \frac{-y}{R}.$$

Assuming Hooke's law holds for the stress σ_{ST},

$$\sigma_{ST} = -E\frac{y}{R}.$$

Consider a thin horizontal slice around from ST to $ST + dy$, having cross-sectional area dA, and let the cross-sectional area of the whole beam be A. As there is no axial load, the *total* force horizontally through the beam must be zero (because the beam is static). So

$$F = \int_A \sigma dA = 0$$

(From this it follows by symmetry that the neutral axis is in the center of the beam.) The strain at $S'T'$ is similarly the tensile

$$\epsilon_{s'T'} = \frac{+y}{R}.$$

Imagine that the beam is split into two at JOK. Across ST there is a compressive force

$$F_{ST} = \sigma_{ST} dA = -\frac{Ey}{R} dA$$

acting leftwards on the RHS of the beam and at $S'T'$ there is a similar tensile force $F_{S'T'} = \sigma_{S'T'} dA$ acting rightwards on the RHS of the beam. (And vice versa for the LHS of the beam.) This pair of forces is equivalent to moments about O on the RHS, and the *total* moment on the RHS at location JOK is given by summing them along JOK:

$$M = \int_A y.\frac{Ey}{R} dA = \frac{E}{R}\int_A y^2 dA = \frac{EI}{R}$$

This must be equal to the original bending moment M applied to the RHS and the theorem follows.

IX. REFERENCES

[1] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simulataneous localization and mapping without predetermined landmarks," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.

[2] M. Kaneko, "Active antenna," in *Fourth IEEE International Conference on Robotics and Automation*, 1994, pp. 2665–2671.

[3] J. Birdwell, J. Solomon, M. Thajchayapong, M. Taylor, M. Cheely, R. Towal, J. Conradt, and M. Hartmann, "Biomechanical models for radial distance determination by the rat vibrissal system." *Journal of Neurophysiology*, vol. 98, no. 4, pp. 2439–55, October 2007.

[4] D. Kim and R. Möller, "Biomimetic whiskers for shape recognition," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 229–243, 2007.

[5] G. R. Scholtz and C. D. Rahn, "Profile sensing with an actuated whisker," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 124–127, 2004.

[6] T. Megson, *Structural and Stress Analysis*. Butterworth-Heinemann, 2005.

[7] S. Rao, *Mechanical Vibrations*. Prentice Hall, 2004.

[8] N. Ueno and M. Kaneko, "Dynamic active antenna - a principle of dynamic sensing," in *IEEE ICRA*, 1994, pp. 1784–1790.

[9] C. Fox, M. Pearson, B. Mitchinson, T. Pipe, and T. Prescott, "Simple features for texture classification from robot whisker strains," *Barrels XIX, Atlanta. Abstract in Somatosensory and Motor Research*, vol. 24, no. 3, pp. 139–162, 2007.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[11] C. Fox and J. Quinn, "How to be lost: Principled pruning and priming with particles for score following," in *Proceedings of the International Computer Music Conference*, 2007.

[12] C. Fox, "ThomCat: A Bayesian blackboard model of hierarchical temporal perception," in *Proceedings of the AAAI International FLAIRS Conference*, 2008.

BMC Neuroscience

BioMed Central

Poster presentation

Open Access

Hippocampal formation as unitary coherent particle filter

Charles Fox* and Tony Prescott

Address: Adaptive Behaviour Research Group, University of Sheffield, UK

Email: Charles Fox* - charles.fox@sheffield.ac.uk

* Corresponding author

from Eighteenth Annual Computational Neuroscience Meeting: CNS*2009
Berlin, Germany. 18–23 July 2009

Published: 13 July 2009

BMC Neuroscience 2009, **10**(Suppl 1):P275 doi:10.1186/1471-2202-10-S1-P275

This abstract is available from: http://www.biomedcentral.com/1471-2202/10/S1/P275

© 2009 Fox and Prescott; licensee BioMed Central Ltd.

A standard approach to localization, recall and prediction from sensors is the Hidden Markov Model (HMM, [1]), with location as hidden state and sense data as observations. Assuming that hippocampus performs a similar task, we present a new top-down mapping of this function onto its anatomy. In localization models of CA3 [2], firing rates of individual place cells encode probabilities over current location and recurrent connections may represent transitions probabilities between them. In auto-associative models [3], recurrent connections bring the network into a vector-coded memorized state. The existence of cells encoding rewards and locations of external objects supports this view [3].

Consider localization and auto-associative models in the light of HMMs, which represent probabilities over the complete set of discrete hidden states, as in localization models. However, inferences about high-level sensory data, including locations of external objects and rewards, require that the structure of these states be complex: for example, a state could include conjunctions of the agents' own location and heading with the configurations of many external objects, rewards, even the agent's own actions. Such states are best represented by configurations of large numbers of Boolean variables, playing a similar role to auto-associative model nodes.

We present a resolution between localization and auto-association models. HMMs become intractable for complex states and it is common to approximate them by particle filters (PFs, [4]). PFs approximate posteriors by samples. In particular, a "unitary" particle filter maintains just a single sample. CA3 could unitarily sample posteriors from a complex state space containing Boolean variables coding presence or absence of objects at locations.

A standard PF problem is getting "lost:" when the samples no longer include the true state, it becomes difficult for inference to return there even if strongly suggested by the data. To recover, a useful heuristic [4] is to monitor matches between predictions and input, and flatten the prior over hidden states when during prolonged poor matches. We model subiculum performing such monitoring, and its resultant septal ACh flattening the transition prior in CA3. This contrasts with ACh models [5] that disable CA3 recurrency for learning rather than inference.

We model superficial and deep entorhinal cortex representing likelihood and posterior percepts; dentate gyrus (DG) extracting features to change the representation basis, and CA1 performing the inverse transform to DG. Preliminary plus maze simulations (figure 1) show subiculum enabling CA3 recovery of correct state estimates following being lost.

References

1. Rabiner LR: **A tutorial on hidden Markov models.** *Proc IEEE* 1989, **77:**257-286.
2. Arleo A, *et al.*: **Place cells and spatial navigation.** *Proc NIPS* 2001:13.
3. Rolls ET: **Memory, attention and decision making.** *Chapter 2. OUP* 2008.
4. Fox C, Quinn J: **How to be lost.** *Proc Int Computer Music Conf* 2007.
5. Hasselmo M, *et al.*: **Dynamics of learning and recall.** *J Neurosci* 1995, **15:**5249-5262.

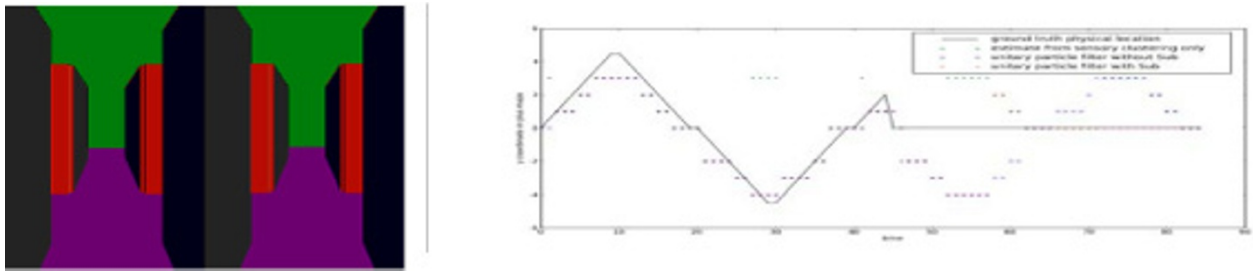


Figure 1
Left: screenshot of maze simulation. The agent visits N-S-E-W arms in turn. *Right*: ground truth and estimates of *y* component of location. The agent is lost during the E arm; only with Sub enabled does it recover for the W arm.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

BioMedcentral

Global Expectation-Violation as Fitness Function in Evolutionary Composition

Tim Murray Browne[1] and Charles Fox[2]

[1] Computing Laboratory
University of Oxford*
tim.murraybrowne@elec.qmul.ac.uk
[2] Adaptive Behaviour Research Group
University of Sheffield
charles.fox@gmail.com

Abstract. Previous approaches to Common Practice Period style automated composition – such as Markov models and Context-Free Grammars (CFGs) – do not well characterise global, context-sensitive structure of musical tension and release. Using local musical expectation violation as a measure of tension, we show how global tension structure may be extracted from a source composition and used in a fitness function. We demonstrate the use of such a fitness function in an evolutionary algorithm for a highly constrained task of composition from pre-determined musical fragments. Evaluation shows an automated composition to be effectively indistinguishable from a similarly constrained composition by an experienced composer.

1 Introduction

Mechanical composition in the Common Practice Period (CPP) style dates at least to Mozart's dice game, which presented 170 bar-length musical fragments and a first-order Markov transition matrix for assembling them into sequences [20]. The Markovian approach to automated composition has continued to the present day, and n-gram Markov models may be found in both generative [6] and evaluative [12] stages of composition. An alternative class of automated composition methods is the use of context-free grammars (CFGs) [4, 9].

Both Markovian and CFG approaches can produce 'correct' sounding compositions, but on extended listening can still sound dull and formulaic. Both approaches tend to lay out notes without considering the global context which is, of course, everything that is heard before *and after* a given note. The need for such a global structure in algorithmic composition is often identified as an open research question [8, 13]. A recent state-of-the-art evolutionary model for rhythm generation [7] identified but circumvented this issue by composing only rhythms and taking their cues from global structures provided in human-composed tracks of the composition.

* Tim Murray Browne is now a research student at Queen Mary University of London.

A new idea in automated composition is to provide a global structure by systematically violating listeners' expectations. Minsky [15] presented the idea of listening to music as exploring a scene, where we absorb different ideas and infer their structural relations. To maintain interest we must always be learning new ideas and relating them to previous ones, which gives some clue as to why Markov and rule-based processes that imitate what has already been heard produce dull results. The problem is that, whilst Markov chains and grammatical parsing are effective models of the *local* listening process, they are not so of the generative process and global perception. In this view, composing is about *manipulating* the listening processes and so whilst the composer must be able to evaluate their composition by listening, there are distinct mechanisms for creating music [18]. The listener needs to be taught ideas then surprised when they are broken or extended. Abdallah and Plumbley [1] recently modelled listening with information measures of a model-based observer. The Kullback-Leibler (KL) divergence, between before and after a note is heard, measured the average rate at which information arrives about the future. Applied to minimalist music, this showed that the listener is constantly receiving *new* information about the rules the music is following, and so learning its structure. This raises questions about the completeness of reductionist approaches to composing music such as CFGs. A composition cannot be broken down into ever smaller sections as ideas that are presented early on must develop and interact, and any part of a composition is always heard in the context of its absolute position within the piece. Analogously to linguistic generation, we may use knowledge of grammar to produce grammatically correct sentences, and with some advanced linguistics perhaps even make them coherent, but by themselves they does not produce interesting or moving prose, let alone poetry.

We present a new method for aesthetic analysis of harmony based on emotional tension, and its use as a fitness function in an evolutionary algorithm for a simplified compositional task. We combine Abdallah and Plumbley's notion of expectation-violation with a classic neural-network listening model [2]. Both of these models were intended for analysis and are here novelly redeployed for use in creative composition. Our compositional task is deliberately constrained, and the present work is intended to illustrate the use of the fitness function and suggest its incorporation into more advanced existing composition systems – we do not intend the simplified compositional task to be a full compositional system in itself.

2 The Global Structure of Expectation-Violation

To form an aesthetic judgement we require tools that work at a human level, emotionally and semantically. Meyer describes the aesthetic experience of listening to music as the *manipulation* of expectation involving the buildup of tension and subsequent release [14]. We will follow this approach and adopt a formal measure of tension based on Abdallah and Plumbley's unmet expectation of resolution. The importance of addressing tension and resolution in composition

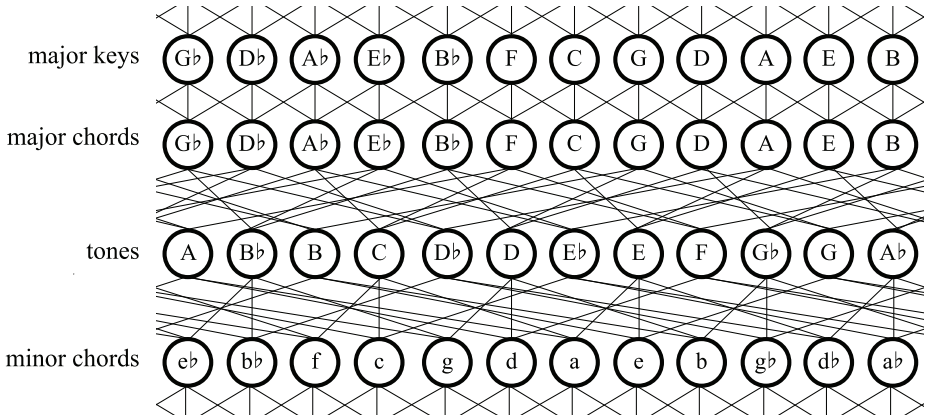


Fig. 1. Bharucha's network, which we redeploy for tension measurement. Links at the sides wrap around.

systems has been noted in recent research as well as the analysis of such qualities with respect to their context [17].

Throughout CPP music many common chord progressions occur. If we play the beginning of a common chord sequence the listener will experience anticipation of which chord is coming next. While we maintain this expectation we build up tension. To prevent our listener from becoming bored we require consistent build up and release of tension and also, to prevent our piece from becoming segmented, we require an increase in the intensity of these build-ups throughout the piece. As demonstrated in [1], if there is no recognisable ground laid, no expectation is produced in the listener and the music is just noise. Likewise if every expected event is granted, the music is predictable and boring. It is the *manipulation* of this expectation that allows us to build up tension.

Bharucha [2] presented the undirected, spreading-activation neural network of fig. 1 as a model of how *listeners* of CPP music infer structural associations between keys by using only the triad chord as foundation. Given a group of simultaneously sounded tones, activation around the network stabilised around related keys that would have been suggested by traditional music theory. Further studies established that people listening to a group of tones do indeed 'prime' related chords in a similar fashion to the network, suggesting expectation [3]. Note that, although based around the triad, the network does indeed capture the relationship between modified chords through how far they are connected. We will here consider Bharucha's past converged network states as specifying a transition prior for the forthcoming key state.

Bharucha's network uses a hierarchical structure of keys, chords and tones, with tones linked to chords they occur in and chords linked to keys they occur in (see fig. 1). At each time step, nodes corresponding to groups of the latest 'heard' tones are activated, then activation spreads through the network in update cycles until an equilibrium is reached. For example, if the tones of a C Major chord {C,

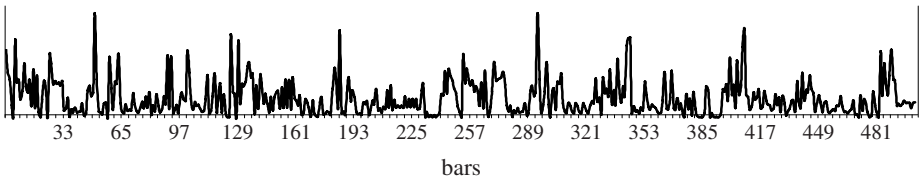


Fig. 2. Graph showing the value of V given in (3) over the entire first movement of Beethoven's fifth symphony.

E, G} are 'heard' together, the key tones reach equilibrium with energy levels monotonically decreasing as we move away in fifths from C.[3] The converged state of the network nodes, when appropriately normalised, may then be considered to represent a prediction on the next time step's key, chords and notes.

We define *key state* X_t at time step t as the normalised activation vector over keys k once the network is stable, and this is interpreted as a probability density function over the current key. Key state alone is used for making and testing predictions. During convergence, the activation x_j of node j is updated via weights w_{ij} as follows:

$$x_j = \sum_{i:i\to j} w_{i,j} x_i \tag{1}$$

where $\{i : i \to j\}$ are the nodes connected to j. $w_{i,j}$ are as in [2]: 0.244 between major chords and keys, 0.22 between minor chords and keys and 0.0122 between tones and chords.

In order to use this network as a model of harmonic expectation the analyser divides a score into temporal units and works through the piece in order, activating the nodes of the tones in each unit as it is heard, and allowing the network to stabilise at each stage. At time t we consider the key state X_t as our observation. We produce a prediction E_{t+1} of the next state by smoothing over all previous observations. We wish predictions to decay by a factor of γ over each beat if not reinforced giving:

$$E_{t+1} \propto \gamma^{1/n} E_t + X_t \tag{2}$$

where we have n units of time per beat. This is a two-variable, first-order Markov process (and is similar to the energy form of a Hidden Markov Model update).

This notion of expectation is meant at a physiological level rather than as a result of logical reasoning and is equivalent to cognitive 'priming' in [3]. We gain a measure of how far our harmonic expectations were violated by considering the KL-divergence of our prediction and observation of the key state X_t:

$$V(t) = \sum_k X_t(k) \log \frac{X_t(k)}{E_t(k)} \tag{3}$$

[3] Whilst we talk of notes in terms of absolute pitch rather than harmonic function (e.g. tonic, dominant), the network's symmetry makes these concepts equivalent within a fixed key.

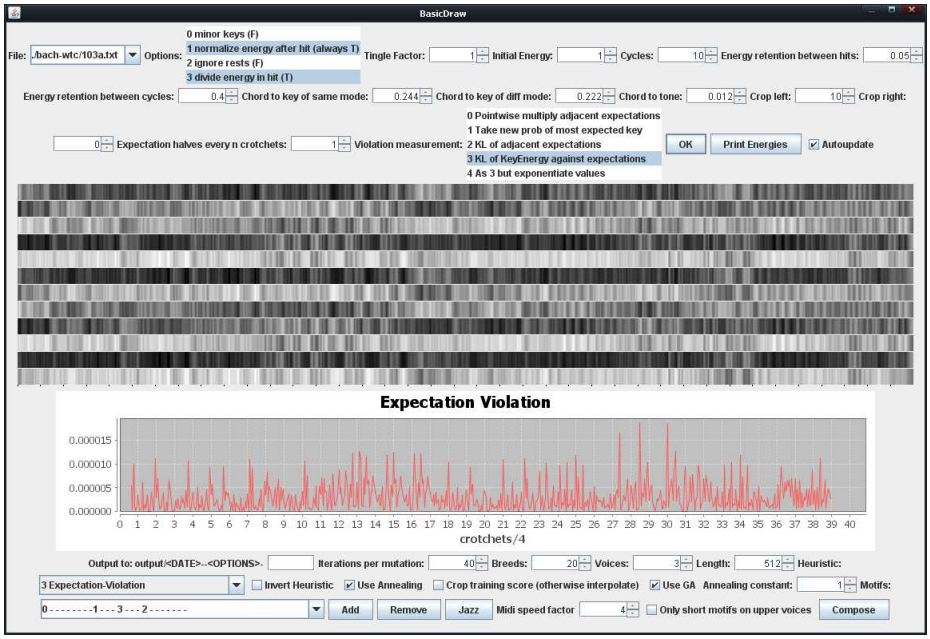


Fig. 3. The interface of the fragment arranger. The central graphic displays the normalised activation vector across the 12 key nodes after the network has stabilised after each unit of time. White is the maximum value and black the minimum.

As t increases, the model's expectation decays if not reinforced so it is not possible to maintain high tension with random notes. Fig. 2 is an example of V over time.

We now use this *local* measure of violation to construct a *global* fitness function R by considering its temporal structure. R compares the temporal tension profile of a candidate score c with that of a training score s:

$$R(c) = \left(\sum_t |V_s(t) - V_c(t)| \right)^{-1} \tag{4}$$

R thus assigns high fitness to new compositions having similar tension profiles to a known training composition, chosen for its pleasing tension profile.

3 The Fragment Arranging Task

As our performance measure is mostly concerned with harmony and less so with rhythm or melodic development we decided to reduce the problem of composing into the easier fragment arranging task. In this task, the program is presented with a number of pre-written motifs and is required to arrange them into a score in a musically desirable fashion. (More advanced motif-arranging systems may

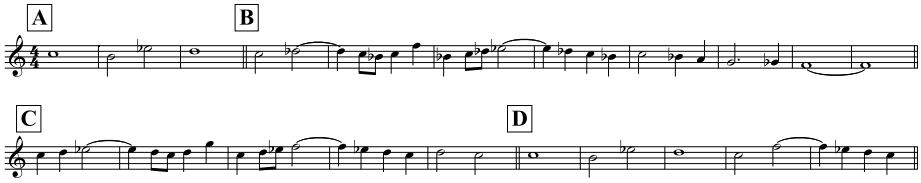


Fig. 4. Fragments used in the evaluation, taken from Fugue 4, Book I of Bach's *Well-Tempered Clavier*.

be found in [10] and the work of Cope [22].) A motif is defined as a sequence of notes and is typically a fragment of a melody that might be used several times by a composer. Arranging motifs into a score means deciding when and at what starting pitch they should be played (where motifs may be used any number of times, and may begin at any multiple of halfbeats through a measure). A score is made up of a number of voices and a voice may only be playing one motif at any one time. Far from being a 'toy problem' this demands an ingenious approach to harmony and produces interesting results quite different from the training material.

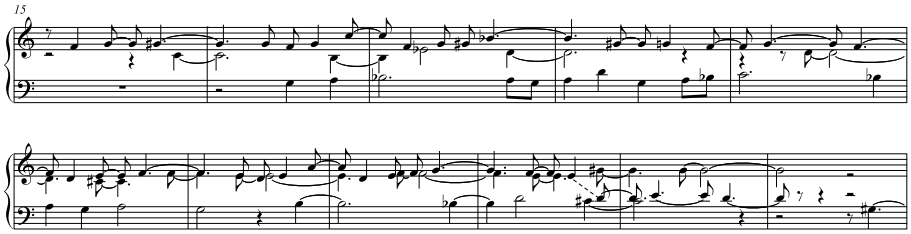
The task is therefore a state-space searching problem where we can move about the space by adding or removing motifs from the score. We implemented evolutionary search using a Metropolis-Hastings (MH) algorithm with simulated annealing [19]. In the MH algorithm the state space is randomly sampled for a proposed alternative to the current state (in our case sampling only involves adding or removing motifs). We accept any proposal that is better than the current state. We also accept some that are worse probabilistically based on how much worse and our *annealing schedule*, which reduces the chance of acceptance as the algorithm progresses. The probability of proposal p being accepted over the current score s was:

$$\Pr(accept) = e^{100(R'(p)-R'(s)).\beta} \tag{5}$$

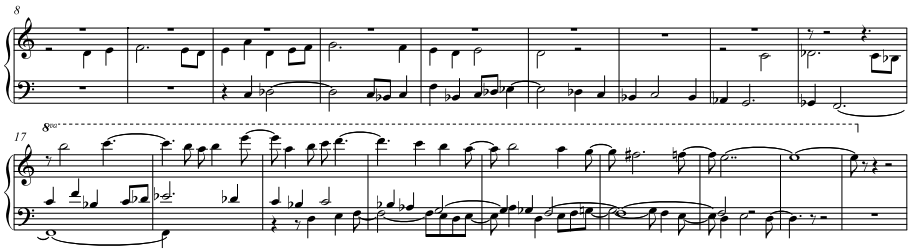
where β decreases linearly from 1 to 0 over the course of the algorithm. The fitness function $R' = RS$ used eqn. (4) together with a density evaluator S, which penalised scores with more than three quarters or less than a quarter of their units as rests. As the fitness function does not differentiate between notes of the same tone in different octaves, the octave that each fragment was placed in was decided by hand afterwards. (The MH algorithm may be viewed as a particular case of Genetic Algorithms: maintaining a population of one; using mutation but no crossover; and always generating two children, one of which is identical to the parent [11].)

4 Evaluation

Although the potential for computers in composition extends well beyond imitating human composers, the task of composing lends itself naturally to a Turing



(a) Computer



(b) Human

Fig. 5. Excerpts from the evaluation scores. The full performances may be listened to at `www.elec.qmul.ac.uk/digitalmusic/papers/2009/MurrayBrowne09evo-data`.

Test. The nature of the fragments task is reminiscent of a fugue so we decided for the evaluation to extract four short (2-8 measure) motifs from a Bach fugue (see fig. 4). The output piece was specified to be in 4/4 time, 32 measures long and in three voices for piano.

We also wished to demonstrate that the performance measure is quite abstracted from the notes and details of the training score, so we are actually modelling the listener's emotional response rather than comparing how similar the candidate score is to our training score. To do so, the same fugue that provided the fragments was used as the training score, to see if the arranger would try to reproduce it.

An experienced composer, Zac Gvirtzman, was given two days to write a piece to the same restrictions with the same input, having heard no program output. Both pieces were recorded by a pianist. We randomly approached 22 Oxford music students in their faculty, told them the composition brief, played them both pieces and then asked them to decide which was written by a human and which by computer. Presentation order was alternated. Excerpts from the two scores are shown in fig. 5. Having heard both pieces, nine answered correctly, 12 answered incorrectly and one refused to answer. Assuming a flat prior on

the population's probability of making a correct decision, the observations yield a Beta posterior having mean 0.42 and standard deviation 0.10, showing the human and machine to be effectively indistinguishable.

5 Discussion

The fragment task is by no means trivial, as confirmed by the composer. However, fragment-arranging is just one of many compositional tools that may be used by humans and larger composing programs, and the human composer commented that he felt very limited without having control over the dynamics and tempo. Even with such a small number of building blocks the program output sounded less repetitive than many Markovian and rule based approaches. Some outputs show original and resolving harmonies and are notable for their tolerance of dissonance that enhance the resolution. Despite not utilising any grammar the outputs have a strong sense of structure, which has been learnt from existing compositions. Interestingly, both computer and human scores used in the evaluation followed an ABA structure, with A drawing exclusively on fragment **A** (see fig. 4). By deliberately violating the listener's harmonic expectation in a controlled way they build up and release global tension, in a similar way to Abdallah and Plumbley's analysis of expectation in notes. They do still however lack a sense of purpose, which will be in part due to the model not having any learning capabilities. Future work may expand on the expectation provided by the listening model by incorporating it alongside Markov chains applied to melody, harmonic progressions and rhythm.

The abstraction of the fitness function from the training material is demonstrated by how, even with both the training material and the set of motifs to be arranged sourced from the same fugue, the output has not attempted to imitate it in any recognisable sense. Further work is required to determine how the function V given in (3) may be used without any training material. Experimenting with the program showed that obvious ways of analysing V such as considering its average over a large corpus of material or attempting to maintain some proportional cumulative value (e.g. $\sum_{i=1}^{t} V_c(i) = \lambda t$ for some λ) does not produce good results. Averaging V over a large corpus of work such as the *Well-Tempered Clavier* results in a fairly flat graph, dissimilar to any of the individual pieces in the work. This was the reason we used just a single composition for our fitness function rather than a set.

The space of scores searched by our evolutionary algorithm were intensionally constrained, and we do not claim to have constructed a full automated composition system. However the performance of the expectation-violation fitness function on the constrained task is encouraging, and suggests that it could be used to enhance most existing composition systems, if weighted appropriately and incorporated into their fitness functions.

References

1. Abdallah, S. and Plumbley, M.: "Information dynamics and the perception of temporal structure in music", Connection Science Journal, *in press*.
2. Bharucha, J. "MUSACT: A Connectionist Model of Musical Harmony", *Proc. Ninth Annual Meeting of the Cognitive Science Society*, Hillsdale, New Jersey, 1989.
3. Bharucha, J., Bigand, E. and Tillmann, B. "Implicit Learning of Tonality: A Self-Organising Approach", *Psychological Review*, 2000.
4. Fox, C.W. "Genetic Hierarchical Music Structures", *Proceedings of the International Florida Artificial Research Society Conference*, 2006.
5. Gill, S. "A Technique for the Composition of Music in a Computer", *The Computer Journal*, 1963.
6. Hiller, L. and Isaacson, L. "Musical Composition with a High-Speed Digital Computer", *J. Audio Eng. Soc.*, 1958.
7. Hoover, A.K., Rosario, M.P. and Stanley, K.O. "Scaffolding for Interactively Evolving Novel Drum Tracks for Existing Songs" *Proceedings of the Sixth European Workshop on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART)*, Springer, 2008.
8. P. Husbands and P. Copley and A. Eldridge and J. Mandelis. "An Introduction to Evolutionary Computing for Musicians", *Evolutionary Computer Music, 1-27, Springer, 2007.*
9. Keller, R.M. and Morrison, D. R. "A Grammatical Approach to Automatic Improvisation", *Proceedings of the Fourth Sound and Music Conference*, 2007.
10. Keller, R., Hunt, M., Jones, S., Morrison, D., Wolin, A., and Gomez, S. "Blues for Gary: Design Abstractions for a Jazz Improvisation Assistant", *Electronic Notes in Theoretical Computer Science)*, 193:47-60, 2007.
11. Laskey, K. and Myers, J. "Population Markov Chain Monte Carlo", *Machine Learning*, 2003.
12. Lo, M.Y. and Lucas, S.M. "Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions", *IEEE Congress on Evolutionary Computation*, 2006.
13. McCormack, J. "Open problems in evolutionary music and art.", *Proceedings of the Sixth European Workshop on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART)*, Springer, 2005.
14. Meyer, L. *Emotion and Meaning in Music*, University of Chicago Press, 1956.
15. Minsky, M. "Music, Mind and Meaning", *CMJ*, 1981.
16. Moorer, J. "Music and Computer Composition", *Communications of the ACM*, 1972.
17. Papadopoulos, G and Wiggins, G. "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects", *AISB Symp. Musical Creativity*, 1999.
18. Pearce, M. and Wiggins, G. "Aspects of a Cognitive Theory of Creativity in Musical Composition", *Proceedings of the ECAI02 Workshop on Creative Systems*, 2002.
19. Rao, R. P. "Bayesian computation in recurrent neural circuits", *Neural Computation*, 2004.
20. Schwanauer, S. and Levitt, D. *Machine Models of Music*, MIT Press, 1993.
21. Sundberg, J. and Lindblom, B. "Generative Theories in Language and Music Description", *Cognition*, 1976.
22. Wiggins, G. "Review of Computer Models of Musical Creativity by David Cope" *Literary and Linguistic Computing*, 2007.

Technical Integration of Hippocampus, Basal Ganglia and Physical Models for Spatial Navigation

Charles Fox [a,*] Mark Humphries [a] Ben Mitchinson [a] Tamas Kiss [b] Zoltan Somogyvari [b] Tony Prescott [a]

[a] *Adaptive Behaviour Research Group, Department of Psychology, University of Sheffield, UK*

[b] *Department of Biophysics, KFKI Research Institute for Particle and Nuclear Physics, Hungarian Academy of Sciences, Budapest, Hungary*

Abstract

Computational neuroscience is increasingly moving beyond modeling individual neurons or neural systems to consider the integration of multiple models, often constructed by different research groups. We report on our preliminary technical integration of recent hippocampal formation, basal ganglia and physical environment models, together with visualisation tools, as a case study in the use of Python across the modelling tool-chain. We do not present new modeling results here. The architecture incorporates leaky-integrator and rate-coded neurons, a 3D environment with collision detection and tactile sensors, 3D graphics and 2D plots. We found Python to be a flexible platform, offering a significant reduction in development time, without a corresponding significant increase in execution time. We illustrate this by implementing a part of the model in various alternative languages and coding styles, and comparing their execution times. For very large scale system integration, communication with other languages and parallel execution may be required, which we demonstrate using the BRAHMS framework's Python bindings.

Key words: hippocampus, basal ganglia, spatial navigation, place cells, plus-maze, BRAHMS, Python

1. Introduction

As computational resources inexorably grow, computational neuroscience is increasingly moving beyond modeling individual neurons or neural systems to consider the integration of multiple models, often constructed by different research groups. At the software level there is a drive towards inter-operability of simulators at both model specification (Goddard et al., 2001) and run-time (Cannon et al., 2007) stages. However, these efforts have concentrated on creating small networks of different multi-compartment models (Gleeson et al., 2007), or large networks of different single-compartment spiking neuron models (Cannon et al., 2007).

Our focus here is on a third strand that can take advantage of growth in computing power: the integration of multiple neural models that form components of a brain-wide system, and the testing of that integrated model in an embodied form. Embodiment often takes the form of a robot and a test environment, whether simulated or real. Requiring the neural models to generate appropriate behavioural output using only inputs available in the environment is a strong test of the proposed computations of that neural system (Humphries et al., 2005; Prescott et al., 2006). In such large simulations, development time is as much an issue as computation time — to implement and test the models, construct simulated environments, implement realistic sensors, and so on. This paper shows how Python provides an excellent solution to both development and computation time problems; we also discuss how Python can work with platforms designed for such large-scale integration (Mitchinson et al., 2008).

As a case study, we report on our preliminary integration of recent hippocampal formation and basal ganglia models, both proposed components of the neural system for spatial navigation (Redish and Touretzky, 1997). The hippocampal formation's role in spatial navigation is not controversial: "place" cells within CA1/CA3 encode position in space (O'Keefe and Conway, 1978; Wiener, 1996); "grid"-cells in entorhinal cortex provide metric information for path-integration via a tessellating rhomboid pattern (Hafting et al., 2005; McNaughton et al., 2006); and hippocampal lesions impair (but not necessarily abolish) rats' abilities to navigate in open environments (Whishaw, 1998). The basal ganglia's main input nucleus — the striatum — is a major target of hippocampal formation output, and

* Charles Fox, Department of Psychology, University of Sheffield, Western Bank, Sheffield, S10 2TP. charles.fox@sheffield.ac.uk

also appears necessary for unimpaired spatial navigation: lesioning the connecting fibres impairs accurate navigation in open environments (see e.g. Whishaw et al., 1995; Devan et al., 1996; Gorny et al., 2002), and blocking plasticity in the region of striatum targeted by hippocampal fibres prevents acquisition of paths to targets (Smith-Roe et al., 1999; Sargolini et al., 2003).

A recurring theme in the basal ganglia literature is that they form a selection mechanism for motor programs (Mink and Thach, 1993; Hikosaka et al., 2000) or, more generally, for "actions" (Redgrave et al., 1999). Thus, the specific hypothesis underlying our integrated model is that the basal ganglia select movement direction based on current spatial position provided by the hippocampal formation input.

The system described below is a preliminary technical integration of the action-selecting basal ganglia model of Gurney et al. (2001a,b) with the hippocampal navigation model of Ujfalussy et al. (2008). The basal ganglia model may be used to select between any types of action, but simple pre-defined saliencies between two target locations are currently used. The hippocampus model may run using any form of sensory input: at present we use visual input, but report on the implementation of physical simulation of tactile whisker-like sensors as an example of developing advanced sensors in Python, which could form a further input in future. Neither the inputs to the models or the placeholder function connecting them are intended to be biologically realistic at this stage. The neural models control a mobile rat-like robot in a standard plus-maze environment with external landmarks, all implemented in a 3D simulator built using existing Python modules. The purpose of this paper is to illustrate a complete neural and physical simulation system, detailing the specific libraries and packages in the tool-chain that were found useful, and not to make any new claims about the biological models. We hope that it will provide a guide for others who wish to implement similar systems, as it can be difficult for newcomers to select the best tools from the plethora of open-source Python extensions.

2. Computational models

We are updating prior models of hippocampal formation-basal ganglia interactions (Arleo and Gerstner, 2000; Chavarriaga et al., 2005) by including the entire basal ganglia circuit and by using a grid-cell driven model of hippocampus. In addition, prior models assumed a direct, modifiable, projection from place cells to the striatum (Arleo and Gerstner, 2000; Chavarriaga et al., 2005). However, such a projection, if it exists, is minor compared to input from other regions of the hippocampal formation, particularly the subiculum, suggesting further stages of processing between the basic representation of position and the striatum (see e.g. van Groen and Wyss, 1990; Groenewegen et al., 1999). In the current integrated system, we provide a simple spatial decoding scheme as a proxy for detailed models of the intervening structures to follow. We do not here present new results from the invididual models (Gurney et al., 2001a,b; Ujfalussy et al., 2008), but report on systems integration at a technical level using Python and BRAHMS.

2.1. *Basal ganglia*

The basal ganglia are a group of inter-connected subcortical nuclei, which receive massive convergent input from most regions of cortex, and output to targets in the thalamus and brainstem (Bolam et al., 2000). We have previously shown how this combination of inputs, outputs, and internal circuitry implements a neural substrate for a selection mechanism (Gurney et al., 2001a,b; Humphries and Gurney, 2002; Gurney et al., 2004; Prescott et al., 2006; Humphries et al., 2006). Figure 1 illustrates the macro- and micro-architecture of the basal ganglia, highlighting three key ideas underlying the selection hypothesis: that the projections between the neural populations form a series of parallel loops — *channels* — running through the basal ganglia from input to output stages (Alexander and Crutcher, 1990); that the total activity from cortical sources converging at each channel of the striatum encodes the salience of the action represented by that channel; and that the selection of an action is signalled by a process of *disinhibition* — the selective removal of tonic inhibition from cells in the basal ganglia's target regions that encode the action (Chevalier and Deniau, 1990).

We use here the population-level implementation of this model from Gurney et al. (2001b). The average activity of all neurons comprising a channel in a population is represented by a single unit that changes according to

$$\tau \dot{a} = -a + u, \tag{1}$$

where τ is a time constant and u is summed, weighted input. We use $\tau = 40$ ms. The normalised firing rate y of the unit is given by a piecewise linear output function

$$y = F(a, \epsilon) = \begin{cases} 0 & a \leq \epsilon \\ a - \epsilon & \epsilon < a < 1 + \epsilon \\ 1 & a \geq 1 + \epsilon \end{cases} \tag{2}$$

The following describes net input u_i and output y_i for the ith channel of each structure, with n channels in total. Net input is computed from the outputs of the other structures, except cortical input c_i to channel i of striatum and STN. The striatum is divided into two populations, one of cells with the D1-type dopamine receptor, and one of cells with the D2-type dopamine receptor. Many converging lines of evidence from electrophysiology, mRNA transcription, and lesion studies suggest a functional split between D1- and D2-dominant projection neurons and, further, that the D1-dominant neurons project to SNr, and the D2-dominant neurons project to GP (Gerfen and Wilson, 1996; Surmeier et al., 2007).

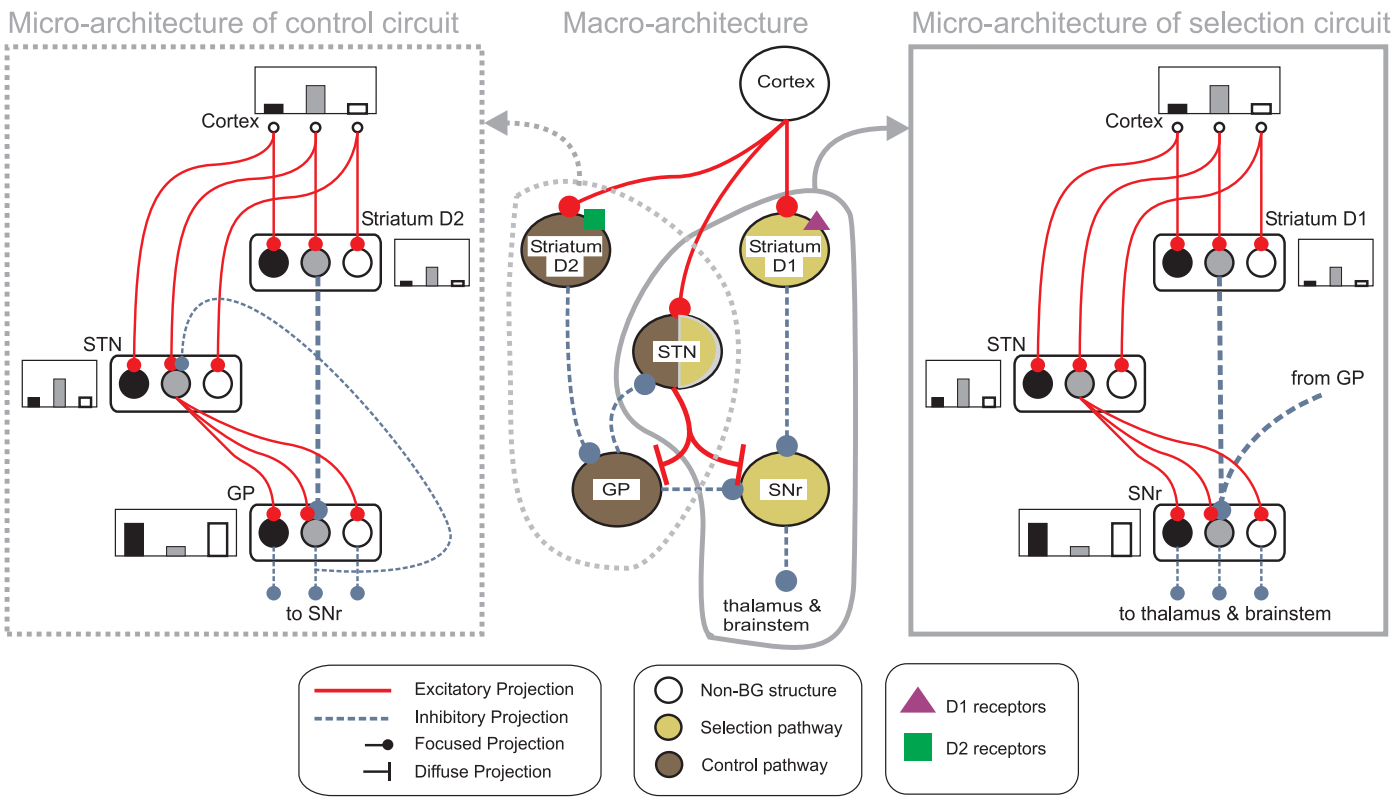


Fig. 1. Architecture of the basal ganglia model. The main circuit (centre) can be decomposed into two copies of an off-centre, on-surround network: a selection pathway (right) and a control pathway (left). Three parallel loops — channels — are shown in both pathways, with example activity levels in the bar charts to illustrate the relative contributions of the nuclei (the three channels are colour-coded black/grey/white, corresponding to the example bar charts). Note that, for clarity, full connectivity is only shown for the second channel. Briefly, the selection mechanism works as follows. Constant inhibitory output from substantia nigra pars reticulata (SNr) provides an 'off' signal to its widespread targets in the thalamus and brainstem. Cortical inputs representing competing saliences are organised in separate channels (groups of co-active cortical neurons), which project to corresponding populations in striatum and subthalamic nucleus (STN). In the selection circuit, the balance of focussed (1-to-1) inhibition from striatum and diffuse (1-to-many) excitation from STN results in the most salient input suppressing the inhibitory output from SNr on that channel, signalling 'on' to that SNr channel's targets. In the control circuit, a similar overlap of projections to globus pallidus (GP) exists, but the feedback from GP to the STN acts as a self-regulating mechanism for the activity in STN, which ensures that overall basal ganglia activity remains within operational limits as more and more channels become active. For quantitative demonstrations of this model see (Gurney et al., 2001b, 2004; Humphries et al., 2006).

Activation of these receptors has opposite effects on striatal input: D1 activation increases the efficacy of the input; D2 activation decreases the efficacy of the input (see Gurney et al., 2001b, for full details). Let the level of tonic dopamine be λ: then the increase in synaptic efficacy due to D1 receptor activation is given by $(1+\lambda)$; the decrease in synaptic efficacy due to D2 receptor activation is given by $(1-\lambda)$. Normal dopamine levels were indicated by $\lambda = 0.2$, and dopamine-depletion by $\lambda = 0$, following previous work (Gurney et al., 2001b; Humphries and Gurney, 2002). The full model is thus given by:

$$\text{Striatum D1:} \quad u_i^{d1} = c_i(1+\lambda), \tag{3}$$

$$y_i^{d1} = F(a_i^{d1}, 0.2), \tag{4}$$

$$\text{Striatum D2:} \quad u_i^{d2} = c_i(1-\lambda), \tag{5}$$

$$y_i^{d2} = F(a_i^{d2}, 0.2), \tag{6}$$

$$\text{STN:} \quad u_i^{stn} = c_i - y_i^{gp}, \tag{7}$$

$$y_i^{stn} = F(a_i^{stn}, -0.25), \tag{8}$$

$$\text{Globus pallidus:} \quad u_i^{gp} = 0.9 \sum_i^n y_i^{stn} - y_i^{d2}, \tag{9}$$

$$y_i^{gp} = F(a_i^{gp}, -0.2), \tag{10}$$

$$\text{SNr:} \quad u_i^{snr} = 0.9 \sum_i^n y_i^{stn} - y_i^{d1} - 0.3 y_i^{gp}, \tag{11}$$

$$y_i^{snr} = F(a_i^{snr}, -0.2) \tag{12}$$

Full details for the chosen constants can be found in (Gurney et al., 2001a), and are summarised here. Thresholds for striatal output were set $\epsilon > 0$ so that a large positive input would be required for any output from these neu-

rons, modelling the large input required to push the striatal projection neuron into its firing-ready "up-state" (Gerfen and Wilson, 1996). The STN, SNr, and GP all had $\epsilon < 0$, as each of these has tonic output at rest (Bolam et al., 2000). Non-unity weights (0.3,0.9) on inputs were set to be within analytically-derived bounds for stable operation of the model (Gurney et al., 2001a).

We used forward Euler to simulate this system for a two-channel model, with the same time-step of 10 ms as was used for the discrete equations of the hippocampus model (see below).

The model was implemented in Python using an object-oriented hierarchy. Neuron objects contain Dendrite objects, which store modulated and unmodulated weights, and references to parent neurons. Neurons also store their parameters (ϵ, τ, s) (where s is the sign of dopamine action) and state (u, a, y). The neuron class contains methods to apply dopamine modulation and determine the unit's output. A Population class groups units together, and contains methods to instantiate sets of one-to-one (e.g. GP→SNr) or diffuse (e.g. STN → SNr) links to other Populations. These methods automatically construct Dendrite objects and update references.

We have found Python's default and named arguments to be especially useful in this type of modeling. Neurons may be given many default parameter values which remain invisible in the user-level code unless specifically overridden. For example, the sign s of dopamine action is assumed to be zero (meaning no effect) unless an easy-to-read named parameter is passed:

```
STN = Population(n, epsilon=-0.25)
STN.addParPopOneToOne(Cx, w_Cx_STN)
D2 = Population(n,dopamineAction=-1,epsilon=0.2)
D2.addParDopamine(SNc)
D2.addParPopOneToOne(Cx, w_Cx_D2)
```

2.2. *Hippocampal formation model*

The hippocampal formation comprises the entorhinal cortex (EC), dentate gyrus (DG), fields CA3 and CA1 of the hippocampus proper, and the subiculum. These form a feed-forward loop of connections that starts and ends in the entorhinal cortex. Though all structures are thought to contribute, the hippocampal model of Ujfalussy et al. (2008) instantiates just the minimum putatively required for the hippocampal formation to act as a memory store (following Treeves and Rolls, 1994); for spatial navigation, the memory formed is considered the place code created by the place cells. Figure 2 shows the basic structure, formed by just the entorhinal cortex, dentate gyrus, and CA3.

Following previous models (e.g. Treeves and Rolls, 1994), the model of Ujfalussy et al. (2008) makes three key assumptions. First, the dentate gyrus region is a preprocessing stage for CA3, acting as a competitive network that creates a sparse and clustered code of the pre-synaptic entorhinal cortex input, which — similarly to other neocortical regions — realizes a denser representation. This sparse, orthogonal code is in turn used as a teaching signal for the CA3 region. Second, the CA3 region acts as an auto-association memory, which stores memory traces in its extensive recurrent local collaterals for later retrieval. Third, many previous hippocampal models (Rolls, 1995; Arleo and Gerstner, 2000; Treeves and Rolls, 1994) assume that the hippocampus operates in two distinct modes during learning and retrieval, which are also incorporated into the present model. As in these models, switching between the two modes is performed manually (contrary to models such as Hasselmo et al. (1995, 1996) which explicitly address the separation between learning and recall). Figure 2 shows the connections that change between the modes.

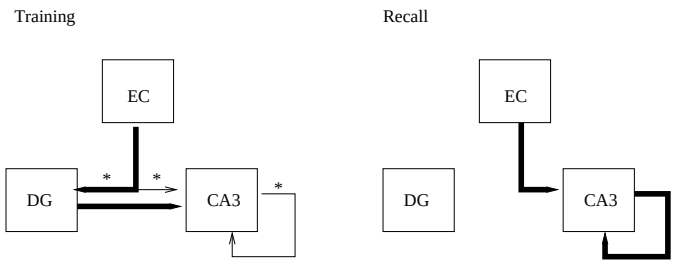


Fig. 2. Basic structure of the hippocampal model. Different connections are active during learning and recall modes. Learning is performed on the asterisked connections only. Thin lines indicate connections which do not drive their targets, but perform learning only.

2.2.1. *Entorhinal cortex*

Grid cells in entorhinal cortex are modeled as having firing rates that are functions of the agent's actual physical position $\mathbf{r} = (x, y)$ in the simulated environment. Grid cells each have two parameters, determining the phase and scale of their receptive fields. The output of the i, jth grid cell is

$$g(\mathbf{r})_{i,j} = \sqrt{\frac{1}{3}\sum_k \cos^2\left[\mathbf{w}_k.(s_i\mathbf{r} - \theta_j)\right]}, \qquad (13)$$

where s_i and θ_j are the ith scale factor and jth phase shift respectively, and $\{\mathbf{w}_k\}_{k=1:3}$ are unit vectors at 60° from each other. We used an ordered set of scale factors from 0.5 to 2.5 in steps of 0.5, and an ordered set of phases from 0 to π in steps of $\pi/5$; i and j are indices into these sets. Figure 3 shows that (13) produces receptive fields with the characteristic rhomboid or "double triangle" tesselation of grid cells (Hafting et al., 2005).

The entorhinal cortex relays input from several cortical areas (Marr, 1971) to the hippocampal formation, and is thus often treated (e.g. Rolls, 1995; Rolls et al., 2006), as in the present model, as the input source for all sensory information. Thus, as well as comprising a large population of grid cells, entorhinal cortex is modelled with an additional population of sensory cells. We use 100 visual cells, whose activations are set by 10×10 grayscale images.

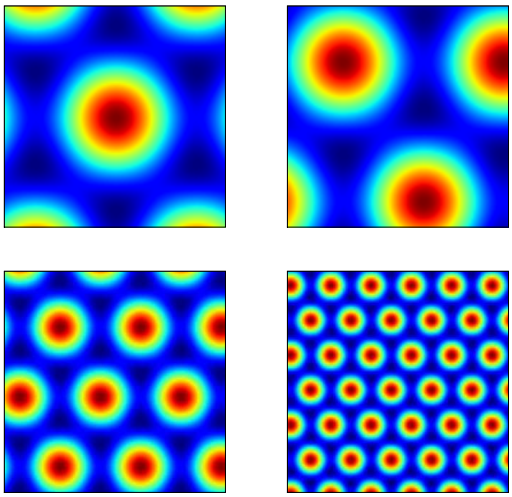


Fig. 3. Grid cell receptive fields from the model, over physical 2D space. These are plotted with Pylab's Matlab-style `imagesc` command.

2.2.2. *Dentate gyrus*

The dentate gyrus is thought to perform a principal-components-like dimensionality reduction of input from the entorhinal cortex (Lorincz, 1998). Writing w_{ij} for weights on inputs y_i, the jth DG unit's neural activation is given by

$$a_j = \sum_i w_{ij} y_i, \tag{14}$$

where the sum is taken over all EC inputs. Output firing rates $\{y_j\}$ are given by a m-best function $\{y_j\} = F\{a_j\}$ which preserves the m largest activations, linearly re-maps them to the interval $[0, 1]$, and sets the others to zero.

During the training phase only, every $EC \rightarrow DG$ weight is updated at each time-step using the standard Hebbian learning rule,

$$\Delta w_{ij} = \alpha y_j (y_i - w_{ij}), \tag{15}$$

where α is the learning rate, and again j represents the DG cell population and i represents the afferent EC population.

2.2.3. *CA3 place cells*

CA3 functions differently during training and recall. During training, CA3 is driven only by input from DG; hence unit activity is updated according to (14) with j representing the CA3 cell population and i representing the afferent DG population. CA3 output is computed from these activations with the same m-best function used for the DG output.

Despite being driven by DG only, no learning is performed on this connection. Instead, learning is performed on the otherwise dormant EC→CA3 and CA3→CA3 pathways. EC→CA3 weights are altered by (15), where y_i is entorhinal cortex output, and y_j is CA3 output. Following Rolls (1995), each recurrent CA3→CA3 weight is altered by the gated Hebbian rule,

$$\Delta w_{ij} = \alpha y_i y_j (1 - w_{ij}) - \beta w_{ij} \tag{16}$$

where β sets the "forgetting rate", and i and j now both refer to cells within the CA3 population.

During the recall phase, the entorhinal cortex input is used to initiate retrieval of a stored memory pattern. First, the activation of CA3 units is computed from the entorhinal cortex inputs only, using (14) with j representing the CA3 cell population and i representing the afferent EC population; their output is then computed using the m-best function. Second, this initial output vector was used as the cue to retrieve the memory trace in the CA3 autoassociative network. The activity of the kth CA3 unit is then the weighted sum of total output from the entorhinal cortex and the recurrent connections

$$a_k = \sum_{i \in EC} w_{ik} y_i + \sum_{j \in CA3} w_{jk} y_j, \tag{17}$$

with CA3 output y_k again computed by applying the m-best function. Activation (17) and output calculations of the CA3 units were iterated I times to bring the CA3 close to an attractor state as in Hopfield-style networks (Hopfield, 1984).

We used eight DG cells and 30 CA3 cells with: learning rate $\alpha = 0.05$, forgetting rate $\beta = 0.00002$, sparsity $m = 20$ and $I = 5$ recurrent iterations.

2.2.4. *Implementation in Python*

The hippocampal model uses simple rate-coded units and linear weights, in contrast to the basal ganglia's leaky integrators. For this reason the population activations and firing rates are amenable to fast implementation as vectors rather than as attributes of individual objects. Multiplication of population firing rates by weight matrices may then be performed by matrix algebra. This style of programming is common in Matlab, and may be performed in Python using the Numpy library (numpy.scipy.org). Numpy emulates much of Matlab's matrix syntax, including notation for slicing matrices (e.g. `A = M[:,1:5]`), addressing (`M[2,3]=4`) and performing operations such as element-wise addition (`B=A+1`) as well as matrix algebra (`C = dot(A,B)`).

We have also made use of two further libraries: SciPy (www.scipy.org) provides a library of higher-level mathematical functions similar to Matlab's toolboxes; and Pylab (matplotlib.sourceforge.net) provides interactive plotting commands. For example, figures 3 and 4 were plotted using Pylab. Pylab emulates many of Matlab's graphics commands including 2D and 3D graphs, and image viewers. The Matlab APIs are replicated almost literally, using the same function names and argument conventions where possible, such as `clf`, `plot` and `imagesc`.

2.2.5. *Training*

Training of the EC→DG, EC→CA3 and CA3→CA3 weights was performed over 5 epochs. Weights were initialised to random real values from a uniform distribution ranging from 0 to 1. Grid and visual cell input data was

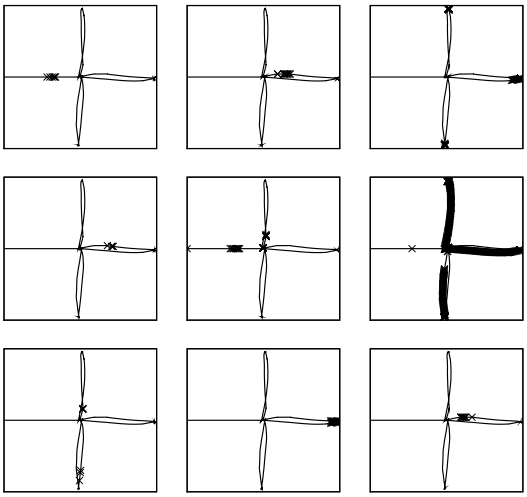


Fig. 4. Receptive fields for nine CA3 place cells, superimposed on the robot's path around the plus maze. Crosses show locations where cells firing rates are in the top 5% of their activity throughout the path. Plotting was performed with Pylab's `plot` command, which has similar syntax to Matlab.

collected from a simulated robot moving to a sequence of pre-determined points in a plus-maze environment (see section 3 for simulator details). The robot enters the maze from the open arm, then visits each of the other arms in turn and comes to rest at the center. About 1,500 data points were sampled during this motion. After training, Python's standard cPickle library provided a simple way to serialize and save the trained Hippocampus object, using only the following code:

```
file=open(‘‘myfile’’,‘‘r’’)
cPickle.dump(myObject, file)
file.close()
```

The effect of training the hippocampus model with the grid cell and visual input was to generate place fields in CA3, such as those shown in Figure 4, which shows the locations of strongest firing for nine of the 30 CA3 cells, superimposed on the robot's path. Of the 30 cells simulated, 11 responded to single places, 13 to two or more places, and 6 were silent at all places (where a 'single' place is defined as a contiguous series of strong activations).

2.3. *Decoding place*

We used a placeholder function for decoding hippocampal place representations into striatal input, as a proxy for detailed models of the intervening structures (e.g. CA1, subiculum) to follow. A simple linear regression was used to find a linear mapping from the vector of place cell activations to the Cartesian (x, y) spatial positions. SciPy provides such regression in its linear algebra sub-package (function `linalg.lstsq`).

3. Building and using the 3D simulator

3.1. *The plus-maze environment*

We used Python to construct a plus-maze environment in which to test our current and future forms of the integrated basal ganglia-hippocampus model. The plus-maze environment was chosen as it is widely used for neural recording studies that probe the roles of striatum, hippocampus, and their interactions in spatial tasks (Albertin et al., 2000; Tabuchi et al., 2000, 2003; Mulder et al., 2004; Khamassi, 2007). Following these studies, the simulated plus-maze comprised a symmetric arrangement of walled arms, and two extra-maze cues (Figure 5).

The neural model was used to control the "ICEAsim" simulated robot, a differential wheels robot in a rat-like form, created by Cyberbotics (Lausanne) for the ICEA project (www.iceaproject.eu). ICEAsim was initially created under Cyberbotics' Webots simulator (Michel, 2004), but was readily imported into a Python simulation via the standard VRML format. Wheel commands are sent via a higher-level (and non-biological) function which takes as input a requested target location to which to move. Our implementation of ICEAsim added two whisker sensors, which output the angle and curvature at their bases for use in tactile perception algorithms (see sections 3.2 and 3.4). We added realistic whisker-like sensors as the basis for future studies: while rats can successfully navigate in the dark, and corresponding place fields are formed in the hippocampal formation, this has been attributed entirely to idiothetic (self-motion) cues (Quirk et al., 1990; Rossier et al., 2000); surprisingly little attention has been paid to the potential role of rats' whiskers in constructing spatial maps in the dark.

For the purposes of this paper, we used a simple task and a placeholder function to test that the models were correctly implemented and technically integrated. After hippocampal training (a separate task, not involving basal ganglia), the robot was simply required to successfully navigate to the end of a maze arm, starting from the entrance of the maze. The basal ganglia model received input saliences $\{c_1, c_2\}$ on two channels, corresponding to two actions ("go to left arm" and "go to right arm"), and which – in this preliminary system – were assigned predefined time series. The placeholder function monitored the hippocampal position estimate, and when this estimate was close to the center of the maze, the action corresponding to the basal ganglia output channel (in SNr) with minimum value was selected and executed to completion. A "go to left arm" or "go to right arm" routine is called, which uses hippocampal output to estimate the required path to follow and sets the robot's differential wheel speeds accordingly. Figure 5 illustrates the simulated robot's behaviour: video of the robot's movement, and corresponding activity in the integrated neural models, are available as Supplemental Data. Future biological models of basal ganglia-hippocampus in-

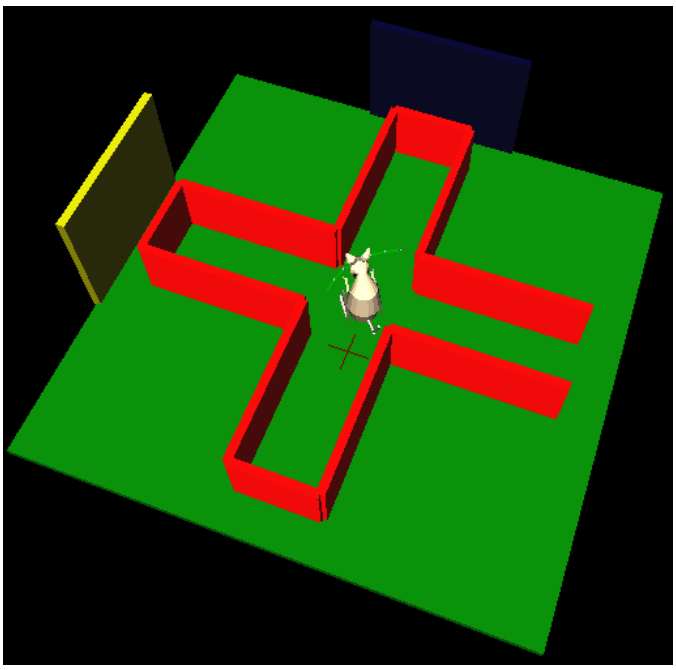


Fig. 5. The simulated plus-maze environment. The hippocampus reports the current estimated location, shown by the cross on the floor. When this estimate is close to the center of the plus-maze, the basal ganglia is consulted for an action to turn. 3D physical simulation and visualisation uses PyODE and Pivy.

teractions may of course replace the predefined time series and placeholder function with more complex and ongoing interactions between the models, using the technical integration framework presented here.

3.2. *Python physics implementation*

To simulate tactile whisker sensors requires realistic physics modeling, as the precise bending (Birdwell et al., 2007), vibration (Ueno and Kaneko, 1994) and other dynamics (Ritt et al., 2006) of whiskers are crucial in making inferences from touch. The Open Dynamics Engine (ODE) is an excellent open-source (BSD license) physics engine, and we use the PyODE wrapper (pyode.sourceforge.net) to use it from Python. ODE provides primitive objects such as cubes, spheres and cylinders, which may be combined and transformed to produce objects such as the walls of the plus-maze and the parts of the robot. PyODE wraps all the major ODE functions for shapes, kinematics and collision handling, and provides access to ODE's standard set of flexible joints. We use the latter to construct rotating wheels, and whiskers. The whiskers are modelled as a series of spherical or cylindrical segments, connected by joints with rotational Hooke's law springs. ODE handles the constraint forces required to keep joints together automatically; however very small time steps (and hence long simulation times) are needed when the number of segments is above three. For example with three segments per whisker the simulation requires about three minutes to run stably on a 1.6GHz machine; with four segments it requires about 10 minutes.

3.3. *Visualization*

3D visualization is important in robotics simulation, both to ensure that the simulation is behaving as intended, and also to provide realistic visual input to robot sensors, for processing by neural models.

OpenGL is a standard 3D graphics application programmer interface (API) (www.OpenGL.org), and is implemented by the free software Mesa and by many hardware-specific graphics drivers. OpenGL provides low-level graphics commands to draw lines, triangles and polygons, and position lights and cameras. The OpenGL API is wrapped in Python by pyOpenGL (pyopengl.sourceforge.net).

Higher-level graphics commands – such as drawing cubes, cylinders and cones using scene graphs – are provided by the OpenInventor API, implemented by the free software Coin (www.coin3d.org). Coin has been wrapped for Python by the Pivy binding (pivy.coin3d.org; Fahmy, 2006) which we use here. Pivy allows raw pyOpenGL commands to be mixed into its higher-level structures where necessary.

To simulate vision (for input to the hippocampus model) we read back images from simulated cameras attached to the robot. Pivy wraps Coin's `SoOffScreenRenderer` function to perform this task. (Modelers are advised that use of this function may be incompatible with the use of direct rendering on some graphics hardware. Disabling direct rendering solves this problem but reduces execution speed.) If graphical output is required in video form only – such as for presentation but not as data for neural models – the free program Yukon (dbservice.com/projects/yukon) is able to export OpenGL graphics to `.avi` movies, whose frame-rate and resolution may be edited with the free Avifix program (www.transcoding.org).

In addition to the main 3D representation of the physical world, it is often useful to attach additional graphical monitors to show the internal state of the neural models in real-time. Pivy – like Coin – takes control of program flow, calling back user functions to draw and update the world. When multiple displays are required – or when handing over control is too intrusive – it is useful to instantiate several processes running Pivy. We use the Python Remote Objects package (PyRo; pyro.sourceforge.net) to handle communication between such processes. PyRo allows an object from one process to appear on another as if it was resident there, allowing function to be called and data to be passed easily. PyRo processes communicate via TCP/IP so the monitors may run on different machines to the main simulation. Figure 6 shows a screen-shot of the state visualisation tool we built for the integrated basal ganglia-hippocampus model.

3.4. *Comparison to PyRobotics*

Our simulation is constructed using Python wrappers for ODE and OpenGL. An alternative approach to simula-

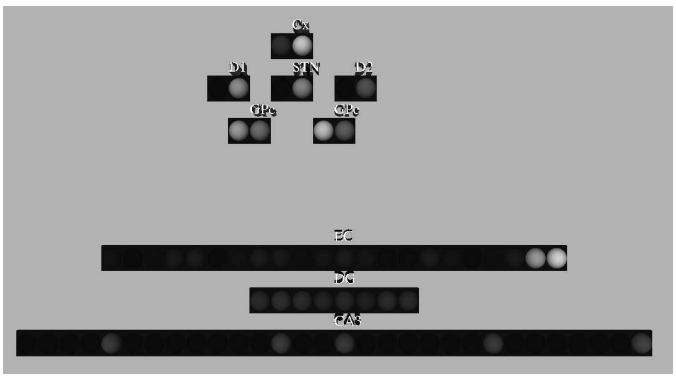


Fig. 6. Real-time graphical neuron monitor, showing basal ganglia and hippocampus model populations. The monitor runs remotely from the simulation over TCP/IP using Pyro, and displays graphics using Pivy.

tion would be to use the higher-level Player interface and Gazebo simulator (playerstage.sourceforge.net) which are available though the PyRobotics (www.pyrobotics.org) integrated robotics simulator. PyRobotics allows worlds and robots to be built from standard components using XML specification, and controllers written in Python. This approach is recommended for simulations requiring standard physics, but our use of the lower-level APIs was determined by the need to write custom physics code for the whisker sensors. Whiskers are difficult to model and coding with ODE directly allows finer control over the contacts and forces that are simulated than would be available in a higher level simulator. Our custom simulations are not intended to be an integrated robotics simulator, but they serve as an example of lower-level PyODE and Pivy simulation.

4. Speed comparisons

Python is often thought of as being a 'slow language' and if this is the case then it would be a barrier to its use in large scale, computationally-intensive neural simulations. However, various libraries and programming styles exist that can improve performance. We investigated a variety of these to evaluate Python's suitability for large simulations. We chose execution of the previous basal ganglia model alone as a benchmark representative of many neural simulation tasks, and used a model with 100 channels running over 1000 time steps to provide a sizable task requiring time of the order of seconds. Neural models are commonly implemented in high-level Matlab code (and its open-source equivalent, Octave), or in low-level C code. C code allows and requires the user to perform their own memory management, leading to greater development time but often faster running times. We re-implemented the basal ganglia model in these languages, writing the fastest code our skills allowed.

In addition to the object-oriented Python model described earlier, we also re-implemented Python models using the Numpy, Pyrex and Weave libraries. As described above, Numpy provides Matlab-like data structures, operations and syntax, to the extent that the Matlab program can be ported to Numpy with only minor syntactic modifications. Pyrex (www.cosc.canterbury.ac.nz/greg.ewing/) is a Python-like language for writing Python extension modules, which provides C-like manual typing and data structures. As with C, Pyrex increases development time by adding work to the programmer's load, but may increase execution time as a result. Programming Pyrex is conceptually similar to writing C programs, but using a Python-like syntax and allowing very simple integration into pure Python code. Weave (part of SciPy) allows inline C code to be embedded directly into Python files, and its 'converters' library automates data type conversion between languages. We implemented inline Weave code within the body of the main Numpy simulation loop.

Another way to improve Python speed is to use more advanced compilers and virtual machines. There is much current research into such tools but a popular system is Psyco (psyco.sourceforge.net). We used Psyco to run the pure Python, object-oriented model (it has negligible effect on Numpy code, in which most of the computation is performed by external numerical C libraries).

Table 1 shows the average execution times for the above implementations (and a BRAHMS version discussed below). Execution was performed on a 1.6GHz, 1.5GB Ubuntu system and time averages were taken over 5 runs. No calls were made to platform-specific BLAS or random-number generator libraries within the simulation loops (such calls are not required or useful in implementing the basal ganglia model's equations). It can be seen that for the Matlab and C-like programming styles (i.e. Numpy and Pyrex respectively) Python is about 4 times slower than the non-Python alternative. Weave is only a fraction slower than raw C, the overhead being due to type conversions. The object-oriented version has a much larger run-time – as expected of this style of programming – and the time is reduced by about 25% using Psyco.

These results suggest that Python is not inherently 'slow' – a factor of four is not large in such comparisons – though it can be used to write slow but conceptually meaningful, human-readable, object-oriented code if desired. Alternatively, if human comprehension is less important, then Matlab-like and C-like programming styles can be used to regain speed. In most cases it is desirable to work on an easily comprehendible 'reference implementation' of a model at first, then develop a faster implementation once the research is complete. Python eases this often difficult transition as Numpy, Weave and Pyrex commands may be gradually mixed into and replace the research code: the more traditional replacement of Matlab by C programs requires a complete rewrite from scratch.

5. Subjective experiences with Python

The above has considered architectural and computational features of Python and its associated libraries that

Table 1
Computation times for the basal ganglia model implemented in different languages and programming formats.

language/format	time (s)
Object-oriented Python	66.1
as above, with Psyco	48.6
Octave	1.31
Numpy Python + BRAHMS	0.89
Numpy Python	0.82
Pyrex	0.22
Matlab	0.21
Scipy.weave.inline	0.05
Raw C	0.04

are useful in embodied neural modeling. However these are not the only criteria for choosing a language for development: at least as important are the more subjective aspects of the system during development and debugging. Here we offer our experiences of hands-on development of the neural and physical models.

We have found that Python supports a wide range of coding styles. In particular, it is possible to code almost literal line-by-line translations of Matlab programs by making heavy use of Numpy's matrices and Pylab's plotting facilities. A key feature is the ability to use interpreted Python from a command line, enabling Matlab-like exploitation of data, testing of functions, and calculator-style calculations. There is typically a little more keyboard typing than when using Matlab. Throughout we have drawn explicit parallels between using Python and Matlab, as Matlab (or Octave) is often the preferred choice for rapid model development and analysis.

Python's class system allows Java-like object-oriented construction of dendrites, neurons and populations. Stylistically its use is similar to Java, or C++ with passing by reference. We have found it a natural but relatively slow-execution way to model neural systems.

The physical and neural simulation, with OpenGL interface, runs at comparable speed to commercial robotics simulators such as Webots (Cyberbotics, Lausanne). We have found development time to be much improved over C++, and comparable with Matlab. However Python gives more versatility than Matlab, allowing easy integration with many open source libraries and the underlying operating system. Our development has used Emacs with its Python mode. In particular, this integrates with the Python debugger, pdb, to allow visual stepping through code and command-line interaction as in Matlab. This type of interaction can be especially important in neural and AI programs, whose states and interactions can become very complex in unpredicted ways.

6. Large-scale integration with BRAHMS

All of the components discussed above (basal ganglia, hippocampus, 3D simulator) were implemented as stateful functions in Python. Thus, integrating them into a computational system was straightforward, by writing a simple Python 'main' function that called these objects in turn to progress them through time. Such an approach to integration is effective, so long as there is no requirement for integration across more complex boundaries. One example of a more complex boundary is cross-language: integrating between functions written in Python, C, Java, or Matlab, for instance, is not generally straightforward. Whilst Python might be a suitable language for large portions of a development, bottleneck computations may benefit from being recoded in a lower-level language such as C. Besides, contributing authors may not all share competence and/or enthusiasm for Python development.

Other obstacles to integration include different component authors, particularly in different groups. This can be problematic since different authors tend to design different interfaces for their components and, in the world of research, rarely have time to properly document these interfaces. Integrating through time — that is, using code written some years ago with code written today — can throw up the same problems as integrating across authors, particularly if documentation is lacking. Cross-platform integration is sometimes necessary, particularly as emphasis shifts to high-performance or embedded computing, and this is far from trivial.

As such multi-module eclectic models become prevalent, and with growing interest in widely varying use cases (high-performance, desktop, embedded), a general solution to the integration problem is urgently required. One such solution is the BRAHMS Modular Execution Framework (brahms.sourceforge.net; Mitchinson et al., 2008). BRAHMS consists of a supervisor, which is analogous to the simple Python 'main' function mentioned above, a fixed supervisor interface against which software components can be developed (currently available in C, C++, Matlab and Python), and a user-extensible set of data types for passing data between software components (forming the inter-process interface). Components need not agree between themselves on implementation: they need only conform to these two interfaces provided and made public by the framework. A BRAHMS system, constructed from processes authored as described below, can be parallelised across computer cores sharing memory or connected by an MPI layer or LAN; alternatively, it can be run on an embedded system, since BRAHMS is lightweight. Here we describe the BRAMHS Python language binding.

6.1. *A BRAHMS process in Python*

The current BRAHMS Python binding (called "1262") requires that the process be implemented as a function; this

function is rendered stateful by passing in and out a reference to a dictionary object, called `persist`. The function is a handler for framework events, so its body consists of a switch block on the event type. The 1262 template provided with BRAHMS handles four events.

The first, (`EVENT_MODULE_INIT`), returns information about the process to the framework, and is already implemented completely in the template. The developer can update the author information as appropriate and familiarise themself with the two possible process flags (discussed below). The second, (`EVENT_STATE_SET`), passes the component its state, which is obtained by the framework from the system document. This 'state' typically consists only of process parameters for initialisation. The third event, (`EVENT_INIT_CONNECT`), requires that the process validate its inputs and create its outputs (discussed below). The fourth, (`EVENT_RUN_SERVICE`), requires that the process service its inputs and outputs (read input data, write output data) at some time, t. This implies that the process must complete its computations at least up to time t (a process is free to progress its state beyond t for any reason). This last event (discussed below) is received multiple times during execution and is, effectively, the process *step* function.

6.1.1. *Connectivity*

In general, a system to be computed may include any number of processes, each of which has each of its outputs dependent on some subset of its inputs. A valid (fully specified) system may have arbitrary (including recursive) output structure dependencies. Since processes are responsible for instantiating their own outputs this requires, in general, multiple calls from the framework to each process in the system to request that it create outputs. The BRAHMS supervisor takes care of making these calls (by sending event `EVENT_INIT_CONNECT`), guarantees that more inputs will be available on each subsequent call (with zero to N available on the first call, and exactly N available on the last), and requires that each process follow a simple algorithm on receiving each call. The algorithm is: (a) observe (and validate, if necessary) the structure of any newly presented inputs; (b) create *as many outputs as possible.* This algorithm will successfully instantiate any valid system. When required dependencies are not met, the framework will raise a 'deadlock' error.

6.2. *Example*

We have constructed and executed successfully a second version of the integrated basal ganglia, hippocampus and physical world simulation in which these three components are implemented as separate BRAHMS modules. Such conversion is straightforward, with each module being pasted into the template and modified such that it expresses the interface described above. Wrapper code linking a Hippocampus Python object to BRAHMS is given in the Appendix. We tested the overhead introduced by the BRAHMS framework by running a BRAHMS-wrapped version of the Numpy basal ganglia model used in the previous speed comparisons. Table 1 shows that the overhead of using BRAHMS is very small; yet using it will now allow extensions to the basic integrated model in any (currently supported) language or level of modelling detail.

7. Conclusions

For large-scale integration and testing of neural models, Python can achieve an excellent balance between development time and computational run-time. The flexibility offered by its modules allows programmers to adopt the style most comfortable to them, without a strong penalty in computation time. We have shown here how all these aspects have contributed to the construction of both an integrated basal ganglia-hippocampal formation model for spatial navigation and its embodiment. Moreover, Python either forms the basis for (PyNN; neuralensemble.org/trac/PyNN), or is compatible with (BRAHMS; Mitchinson et al., 2008), platforms that address larger-scale integration across modelling levels and hardware. Thus, Python is a crucial part of the neuroinformatics toolbox: flexible, usable, readable, and scalable.

Supplemental Data

Videos and source code from the simulation and speed comparisons are presented in the Supplemental Data.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgements

This work was supported by the European Union Framework 6 IST project 027819 (ICEA project: www.iceaproject.eu) and the European Union Framework 7 ICT project 215910 (BIOTACT project: www.biotact.org).

References

Albertin, S. V., Mulder, A. B., Tabuchi, E., Zugaro, M. B., Wiener, S. I., 2000. Lesions of the medial shell of the nucleus accumbens impair rats in finding larger rewards, but spare reward-seeking behavior. Behav Brain Res 117, 173–183.

Alexander, G. E., Crutcher, M. D., 1990. Functional architecture of basal ganglia circuits: neural substrates of parallel processing. Trends Neurosci 13, 266–272.

Arleo, A., Gerstner, W., 2000. Spatial cognition and neuromimetic navigation: a model of hippocampal place cell activity. Biological Cybernetics 83, 287–299.

Birdwell, J., Solomon, J., Thajchayapong, M., Taylor, M., Cheely, M., Towal, R., Conradt, J., Hartmann, M., October 2007. Biomechanical models for radial distance determination by the rat vibrissal system. Journal of Neurophysiology 98 (4), 2439–55.

Bolam, J. P., Hanley, J. J., Booth, P. A., Bevan, M. D., 2000. Synaptic organisation of the basal ganglia. J Anat 196 (Pt 4), 527–542.

Cannon, R. C., Gewaltig, M.-O., Gleeson, P., Bhalla, U. S., Cornelis, H., Hines, M. L., Howell, F. W., Muller, E., Stiles, J. R., Wils, S., Schutter, E. D., 2007. Interoperability of neuroscience modeling software: current status and future directions. Neuroinformatics 5, 127–138.

Chavarriaga, R., Strösslin, T., Sheynikhovich, D., Gerstner, W., 2005. A computational model of parallel navigation systems in rodents. Neuroinformatics 3, 223–241.

Chevalier, G., Deniau, J. M., 1990. Disinhibition as a basic process in the expression of striatal function. Trends Neurosci 13, 277–280.

Devan, B. D., Goad, E. H., Petri, H. L., 1996. Dissociation of hippocampal and striatal contributions to spatial navigation in the water maze. Neurobiol Learn Mem 66, 305–323.

Fahmy, T., 2006. Pivy - embedding a dynamic scripting language into a scene graph library. Master's thesis, Vienna University of Technology.

Gerfen, C., Wilson, C., 1996. The basal ganglia. In: Swanson, L., Bjorklund, A., Hokfelt, T. (Eds.), Handbook of chemical neuroanatomy. Vol 12: Integrated systems of the CNS. Part III. Elsevier, Amsterdam, pp. 371–468.

Gleeson, P., Steuber, V., Silver, R. A., 2007. neuroConstruct: a tool for modeling networks of neurons in 3D space. Neuron 54, 219–235.

Goddard, N. H., Hucka, M., Howell, F., Cornelis, H., Shankar, K., Beeman, D., 2001. Towards NeuroML: model description methods for collaborative modelling in neuroscience. Philos Trans R Soc Lond B Biol Sci 356, 1209–1228.

Gorny, J. H., Gorny, B., Wallace, D. G., Whishaw, I. Q., 2002. Fimbria-fornix lesions disrupt the dead reckoning (homing) component of exploratory behavior in mice. Learn Mem 9, 387–394.

Groenewegen, H. J., Mulder, A. B., Beijer, A. V. J., Wright, C. I., Lopes Da Silva, F. H., Pennartz, C. M. A., 1999. Hippocampal and amygdaloid interactions in the nucleus accumbens. Psychobiology 27, 149–164.

Gurney, K., Prescott, T. J., Redgrave, P., 2001a. A computational model of action selection in the basal ganglia I: A new functional anatomy. Biol Cybern 85, 401–410.

Gurney, K., Prescott, T. J., Redgrave, P., 2001b. A computational model of action selection in the basal ganglia II: Analysis and simulation of behaviour. Biol Cybern 85, 411–423.

Gurney, K. N., Humphries, M., Wood, R., Prescott, T. J., Redgrave, P., 2004. Testing computational hypotheses of brain systems function using high level models: a case study with the basal ganglia. Network 15, 263–290.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., Moser, E. I., 2005. Microstructure of a spatial map in the entorhinal cortex. Nature 436, 801–806.

Hasselmo, M., Schnell, E., Barkai, E., 1995. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region ca3. Journal of Neuroscience 15, 5249–5262.

Hasselmo, M., Wyble, B., G.V. Wallenstein, G., 1996. Encoding and retrieval of episodic memories: role of cholinergic and gabaergic modulation in the hippocampus. Hippocampus 6, 693–708.

Hikosaka, O., Takikawa, Y., Kawagoe, R., 2000. Role of the basal ganglia in the control of purposive saccadic eye movements. Physiol Rev 80, 953–978.

Hopfield, J. J., 1984. Neurons with graded response have collective computational properties like those of two-state neurons. Proc Natl Acad Sci USA 81, 3088–3092.

Humphries, M. D., Gurney, K., Prescott, T. J., 2005. Is there an integrative center in the vertebrate brainstem? a robotic evaluation of a model of the reticular formation viewed as an action selection device. Adaptive Behavior 13, 97–113.

Humphries, M. D., Gurney, K. N., 2002. The role of intra-thalamic and thalamocortical circuits in action selection. Network 13, 131–156.

Humphries, M. D., Stewart, R. D., Gurney, K. N., 2006. A physiologically plausible model of action selection and oscillatory activity in the basal ganglia. J Neurosci 26, 12921–12942.

Khamassi, M., 2007. Complementary roles of the rat prefrontal cortex and striatum in reward-based learning and shifting navigation strategies. Ph.D. thesis, Univeristy Paris 6.

Lorincz, A., 1998. Forming independent components via temporal locking of reconstruction architectures: a functional model of the hippocampus. Bio Cybernetics 79, 263–275.

Marr, D., 1971. Simple memory: a theory for archicortex. Phil trans Royal Soc London. Series B Bio 262, 23–81.

McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., Moser, M.-B., 2006. Path integration and the neural basis of the 'cognitive map'. Nat Rev Neurosci 7, 663–678.

Michel, O., 2004. Webots(tm): Professional mobile robot simulation. Int J Adv Robot Sys 1, 39–42.

Mink, J. W., Thach, W. T., 1993. Basal ganglia intrinsic circuits and their role in behavior. Curr Opin Neurobiol 3, 950–957.

Mitchinson, B., Chan, T., Chambers, J., Humphries, M., Gurney, K., Prescott, T., 2008. Brahms: Novel middleware for integrated systems computation. In: Frontiers in Neuroinformatics. Conference Abstract: Neuroinformatics 2008. DOI:10.3389/conf.neuro.11.2008.01.051.

Mulder, A. B., Tabuchi, E., Wiener, S. I., 2004. Neurons in

hippocampal afferent zones of rat striatum parse routes into multi-pace segments during maze navigation. Eur J Neurosci 19, 1923–1932.

O'Keefe, J., Conway, D. H., 1978. Hippocampal place units in the freely moving rat: why they fire where they fire. Exp Brain Res 31, 573–590.

Prescott, T. J., Montes Gonzalez, F. M., Gurney, K., Humphries, M. D., Redgrave, P., 2006. A robot model of the basal ganglia: Behavior and intrinsic processing. Neural Netw 19, 31–61.

Quirk, G. J., Muller, R. U., Kubie, J. L., 1990. The firing of hippocampal place cells in the dark depends on the rat's recent experience. J Neurosci 10, 2008–2017.

Redgrave, P., Prescott, T. J., Gurney, K., 1999. The basal ganglia: A vertebrate solution to the selection problem? Neuroscience 89, 1009–1023.

Redish, A. D., Touretzky, D. S., 1997. Cognitive maps beyond the hippocampus. Hippocampus 7, 15–35.

Ritt, J., Andermann, M., Skowronski-Lutz, E., Moore, C., 2006. Characterization of vibrissa motion during volitional active touch. In: Barrels XIX. Atlanta, USA.

Rolls, E., 1995. A model of the operation of the hippocampus and cortex in memory. Int J Neural Systems 6, 51–71.

Rolls, E., Stringer, S., Elliot, T., 2006. Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. Network 17, 447–465.

Rossier, J., Kaminsky, Y., Schenk, F., Bures, J., 2000. The place preference task: a new tool for studying the relation between behavior and place cell activity in rats. Behav Neurosci 114, 273–284.

Sargolini, F., Florian, C., Oliverio, A., Mele, A., Roullet, P., 2003. Differential involvement of NMDA and AMPA receptors within the nucleus accumbens in consolidation of information necessary for place navigation and guidance strategy of mice. Learn Mem 10, 285–292.

Smith-Roe, S. L., Sadeghian, K., Kelley, A. E., 1999. Spatial learning and performance in the radial arm maze is impaired after n-methyl-d-aspartate (NMDA) receptor blockade in striatal subregions. Behav Neurosci 113, 703–717.

Surmeier, D. J., Ding, J., Day, M., Wang, Z., Shen, W., 2007. D1 and D2 dopamine-receptor modulation of striatal glutamatergic signaling in striatal medium spiny neurons. Trends Neurosci 30, 228–235.

Tabuchi, E., Mulder, A. B., Wiener, S. I., 2003. Reward value invariant place responses and reward site associated activity in hippocampal neurons of behaving rats. Hippocampus 13, 117–132.

Tabuchi, E. T., Mulder, A. B., Wiener, S. I., 2000. Position and behavioral modulation of synchronization of hippocampal and accumbens neuronal discharges in freely moving rats. Hippocampus 10, 717–728.

Treeves, A., Rolls, E., 1994. Computational analysis of the role of the hippocampus in memory. Hippocampus 4, 374–391.

Ueno, N., Kaneko, M., 1994. Dynamic active antenna - a principle of dynamic sensing. In: IEEE ICRA. pp. 1784–1790.

Ujfalussy, B., Eros P.and Somogyvari, Z., Kiss, T., 2008. Episodes in space: A modelling study of hippocampal place representation. In: Asada, M., Hallam, J., Meyer, J.-A., Tani, J. (Eds.), From Animals to Animats 10. Vol. 5040 of LNAI. Springer-Verlag, Berlin, pp. 123–136.

van Groen, T., Wyss, J. M., 1990. Extrinsic projections from area CA1 of the rat hippocampus: olfactory, cortical, subcortical, and bilateral hippocampal formation projections. J Comp Neurol 302, 515–528.

Whishaw, I. Q., 1998. Place learning in hippocampal rats and the path integration hypothesis. Neurosci Biobehav Rev 22, 209–220.

Whishaw, I. Q., Cassel, J. C., Jarrad, L. E., 1995. Rats with fimbria-fornix lesions display a place response in a swimming pool: a dissociation between getting there and knowing where. J Neurosci 15, 5779–5788.

Wiener, S. I., 1996. Spatial, behavioral and sensory correlates of hippocampal CA1 complex spike cell activity: implications for information processing functions. Prog Neurobiol 49, 335–361.

8. Appendix

The following shows the code used to link the Hippocampus model to the BRAHMS framework. The code implements four BRAHMS events. The persistent state consists of an instance of a pre-trained `Hippocampus` object, created in `EVENT_STATE_SET`. Servicing (`EVENT_RUN_SERVICE`) consits of reading the BRAHMS inputs, passing them in an appropriate format to the `Hippocampus` object, and passing its output back to BRAHMS. The other events are described in section 6.1.

```
import brahms
from hc import *
def brahms_process(persist, input):
 output = {'info':{},'operations':[],'event':{'response': 0}}

  if input['event']['type'] == EVENT_MODULE_INIT:
    #these flags inform BRAHMS that this process
    #needs all inputs to be available before it can initalise,
    #and that the process does not change the sample rate.
  output['info']['flags']=F_NEEDS_ALL_INPUTS+F_NOT_RATE_CHANGER
    output['info']['component'] = (0, 1)
    output['info']['additional'] = ''
    output['event']['response'] = C_OK

  elif input['event']['type'] == EVENT_STATE_SET:
    #create an instance of the Python Hippocampus object
    pars = persist['state']
    persist['ptHC'] = loadHippocampus()
    output['event']['response'] = C_OK

  elif input['event']['type'] == EVENT_INIT_CONNECT:
    #check the data types of the BRAHMS inputs
    p = input['iif']['default']['ports']
    if len(p) != 1:
      output['error'] = 'expects one input'
      return (persist, output)
    if p[0]['class'] != 'dev/std/data/numeric':
```

```
    output['error'] = 'expects data/numeric INPUT'
    return (persist, output)
  if p[0]['structure'] != 'DOUBLE/REAL/102':
    output['error'] = 'expects real double 2x1 input'
    return (persist, output)
  #create a BRAHMS output
  persist['hOut']=brahms.operation(
  persist['self'],OPERATION_ADD_PORT,
  '','dev/std/data/numeric',
  DOUBLE/REAL/'+str(persist['state']['n_out']),
  out')
  output['event']['response'] = C_OK

 elif input['event']['type'] == EVENT_RUN_SERVICE:
  ptHC = persist['ptHC']  #retreive my persistent state
  #retreive my current inputs from BRAHMS
  ins = input['iif']['default']['ports'][0]['data']
  x=ins[0]
  z=ins[1]
  img=ins[2:102]
  img=array(img.ravel())
  img.shape=(100,1)
  #call to the Python Hippocampus object
  x_hat, z_hat = ptHC.step(x,z,img)
  #create output and send it to BRAHMS
  myOutput = numpy.array([x, z, x_hat, z_hat], numpy.double)
  brahms.operation(persist['self'], OPERATION_SET_CONTENT,
              persist['hOut'], myOutput)
  output['event']['response'] = C_OK

#return the output and the modified persistent state
return (persist, output)
```

Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task.

Mat Evans[1], Charles W. Fox[1], Martin J. Pearson[2] and Tony J. Prescott[1]

***Abstract*—**
Whisker-based floor texture classifications could form an important input to mobile robots for use in navigation and object recognition, especially in harsh and covert environments. Previous work [9], [11], [13] showed that artificial whiskers can be used for texture discrimination in controlled situations. In these experiments both the movement of the whisker and the textures used were carefully selected to constrain the problem. In contrast, the present study investigates whiskered texture discrimination in a practical, real-world task. An artificial whisker was attached to an iRobot Roomba vacuum cleaner, and its base displacement was sampled while the robot moved in different directions on four different real-world floor surfaces: rough and smooth carpet, vinyl and tarmac. Surface discrimination is possible using a template based classifier. The discrimination is dependent on knowing robot motion type, with different movement strategies providing better signals for certain texture discriminations.

I. Introduction

The rat whisker system has been the subject of a great deal of biological research [1], [7], [15], [5]. Numerous behavioural experiments have determined that rats are tactile discrimination specialists, both of object location in the whisker field [20], [21] and especially of surface texture [3], [4]. In the dark, rats can extract the identity of a 30um grating texture based on just one to three touches per whisker and can display accurate judgments of a texture within 100 ms of initial whisker contact [2]. In an effort to understand and emulate the whiskered sensing capabilities of rats, many researchers have investigated the use of artificial whiskers for tactile discrimination.

Recent work has explored the capabilities of whiskered sensors from an engineering perspective. Whisker based tactile sensors have the potential to be uniquely useful in a broad range of tasks. As opposed to vision and echo-location, tactile sensation can be performed in complete darkness, in dusty or cloudy environments, and is covert (i.e. does not emit light or sound). Whisker sensors are mechanically very simple, they require very little energy to run, and can work in confined spaces. These properties make whisker sensors ideal for mobile robotics, and would for example be useful when exploring adverse underground environments. Due to their cheapness and simplicity, whiskers could also make effective additions to domestic robots such as vacuum cleaners.

Studies of object location in the whisker field have investigated the use of static beam equations [12], [2], deflection magnitude and time series differences [14]. Hipp et al [11] looked for the features in the whisker signals that allowed the rat to make such fine texture discriminations, and showed that artificial whiskers can be used to discriminate different grades of sandpaper using the power spectral density of the signal.

Navigation, or any high level object recognition task would require some low level feature recognition. In the tactile domain, local texture provides a convenient and robust low-level feature set for building higher level models of object recognition and map building for navigation. For a mobile robot in an office or home environment, floor surface identity provides an ideal cue to location. Recognition of a floor surface as carpet versus vinyl flooring could indicate that the robot is in a living room or kitchen, and boundaries between surfaces within a room are valuable landmarks. For a cleaning robot, floor surface recognition could further activate different cleaning mechanisms, and provide estimates of current odometry noise associated with motion on different surfaces.

Previously we have shown that artificial whiskers can be used for texture discrimination of above-floor objects, by a mobile robot platform making three constrained types of contact with four texturally homogeneous surfaces [9]. If the textured surface's location relative to the platform is known, and the speed of the robot's movement moderated, it is possible to use a range of classifiers to distinguish rough from smooth surfaces. In contrast, the present paper is an 'in the wild' study, with texture data gathered in a real-world office environment.

II. Materials and methods

A. The Whisker

Our whisker sensor consisted of a flexible plastic (Acrylonitrile butadiene styrene (ABS)) whisker shaft (200mm long, 2mm diameter, E $\approx$ 2.3GPa) mounted at its base into a short, polyurethane rubber filled, inflexible tube called a follicle case (see fig 1). ABS plastic was used for the whisker shaft because of its flexibility, appropriate mechanical match to scaled-up biological whiskers, and suitability for rapid prototyping using the Fused Deposition Modeling (FDM) approach. The use of rapid prototyping technology to build the whisker will also contribute significantly to any future investigation into the morphology of the whisker shaft itself.

A magnet was bonded to the base of the whisker shaft in such a way that when the follicle case/whisker shaft assembly was located into the whisker mount (see fig 1), the magnet was positioned directly above a tri-axis Hall effect sensor IC

[1] Active Touch Laboratory, Department of Psychology, University of Sheffield, Western Bank, Sheffield. S10 2TN, UK. *mat.evans@shef.ac.uk*
[2] Bristol Robotics Laboratory, Du Pont Building, Bristol Business Park, Coldharbour Lane, Frenchay, Bristol. BS16 1QD, UK

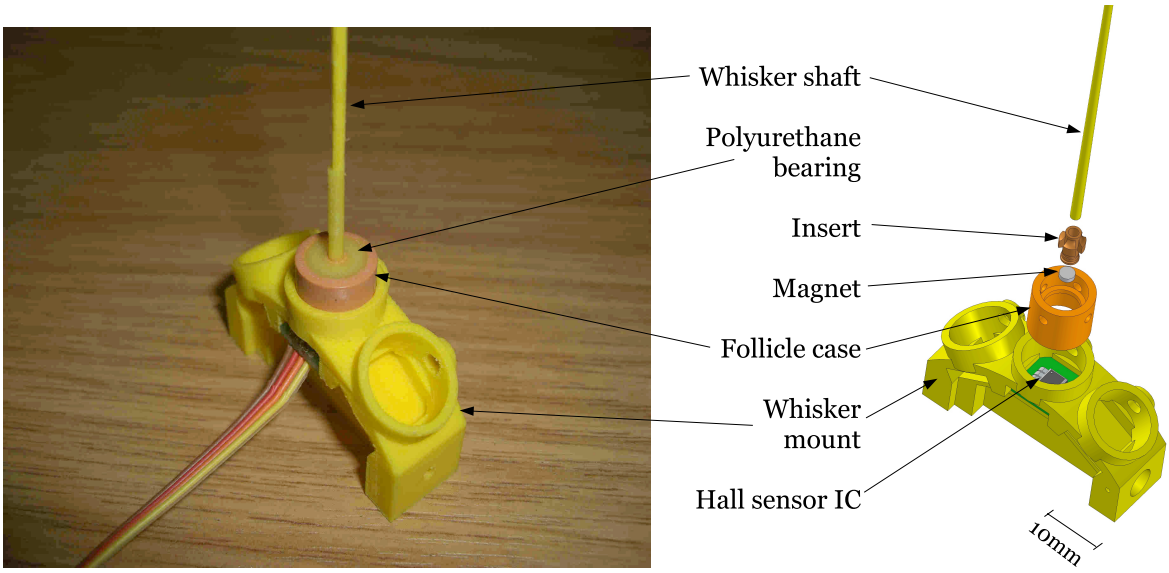


Fig. 1. Diagram of whisker follicle sensor construction

(Melexis MLX90333 [16]). Hall effect sensors measure the change in voltage across a conductor in response to changes in the strength of a nearby magnetic field. The tri-axis Hall effect sensor used here can measure the voltage changes in 3 orthogonal axes, i.e., *x, y, z*. This sensor technology was chosen for its robustness (no physical coupling between sensor and environment), ability to measure displacements in two dimensions, cost ($\approx$ 6 euros), size (S08 IC package 3x4mm) and reprogramability, which allows the sensitivity of the sensor to be adjusted after manufacture to best suit the current experiment or application.

The Hall effect sensor IC was programmed to generate 2 voltages, the magnitudes of which being proportional to the two orthogonal displacement angles (α, β) of the magnet from its resting position above the sensor (see fig 3). As forces are applied to the whisker shaft, the moment experienced at the base will rotate the magnet around the pivot point, nominally in the centre of the polyurethane bearing. A trigonometric operation in the DSP core of the Hall sensor IC decouples the alpha and beta angles and removes the z-component introduced by the arc of travel of the magnet, as indicated by the blue dotted line in fig 3. This operation ensures that the output voltages from the IC are linearly proportional to the tangent component of the alpha and beta angles, or x and y as they will be refered to hereafter.

To set the operating range of the sensor a calibration stand was constructed to allow a fixed deflection, d, to be applied to the whisker shaft at a known radial distance, l, from the base in the 2 dimensions (refer to fig. 3). The output voltage from the IC was then scaled to $\pm d$ in both dimensions, with d=0 set to 50% of the maximum output voltage Vdd, i.e., 2.5V. For the whisker sensor used in the experiments reported here, the voltage range was set as 5 - 95% Vdd through $\pm d$ = 60mm applied at l = 150mm.

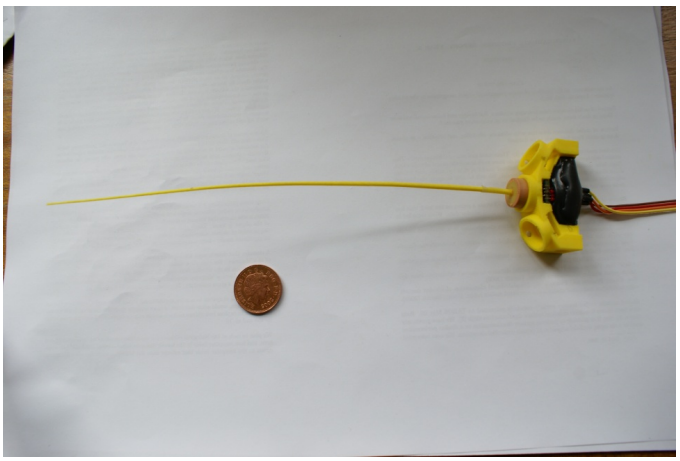


Fig. 2. Artificial whisker shaft and follicle, with a 1p coin for scale

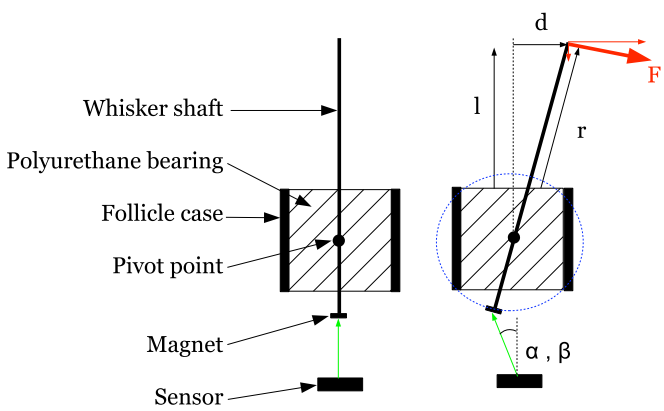
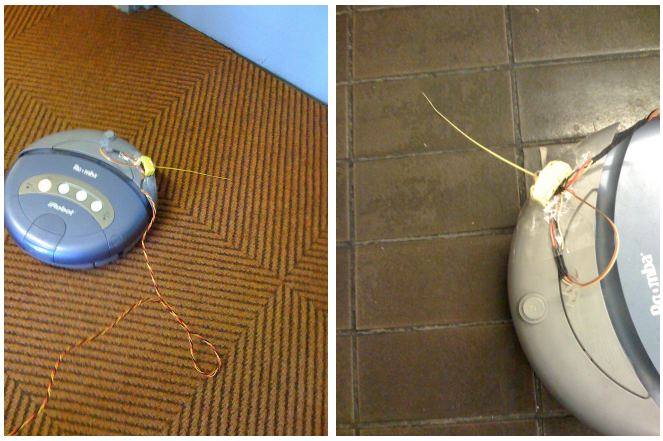


Fig. 3. Diagram of the artificial whisker Hall effect sensor



(a) (b)

Fig. 4. (a) iRobot Roomba with the artificial whisker sensor attached. (b) A close up of the whisker angled to make contact with the floor

B. The robot

An iRobot Roomba vacuum cleaner (iRobot, 8 Crosby Drive, Bedford, MA 01730) was chosen as the platform for these experiments (fig. 4). This robot was ideal for the task as it would ultimately be the candidate system for any behavioural output from the classification. The Roomba is a commercial robot vacuum cleaner that is programmed to clean carpets autonomously, using three modes chosen by the user, and control is modified by the environment it encounters. A remote control is also provided with the Roomba which allows more direct control over the robot's behaviour. Without access to the preprogrammed algorithms it is difficult to automatically determine the robot's motions during execution of its cleaning modes, so the present work focuses on simple motions produced with the remote control.

The whisker was mounted on the the front of the robot, at 45 degree azimuth from the forwards direction of travel, and a slight downwards elevation sufficient to make constant contact with the floor during movement (see fig. 4). Thus the whisker transduces a constant stream of deflection information as long as the robot is moving.

C. The task

Four surfaces were chosen for classification: two carpets of different roughnesses, a lino surface and a tarmac surface (fig. 5). These surfaces were chosen because they were appropriately generic for a real world experiment, and they provide a range of surface types, while also being sufficiently similar to make classification difficult.

Two primary behavioural conditions were chosen with the robot moving either anticlockwise only, or clockwise only. This provided enough data for developing classifiers for known robot motions. In order to demonstrate the difficulty of classification without knowledge of the robot motion, we also recorded data for each surface during the Roomba's 'spot' cleaning programme. This consists of a series of (externally) unpredictable clockwise, anticlockwise and forward motions.

As the robot moved the whiskers were swept across the floor. Any deflections of the whisker were transmitted through the hall effect sensors through a LabJack UE9 USB data acquisition card (www.labjack.com) at a rate of 2 kHz for each of the x and y directions. This data was sent to a computer through the BRAHMS middleware (brahms.sourceforge.net) for analysis in MATLAB (www.mathworks.com). Example data from the four floor surfaces is shown in fig. 6.

III. Results

A. The data

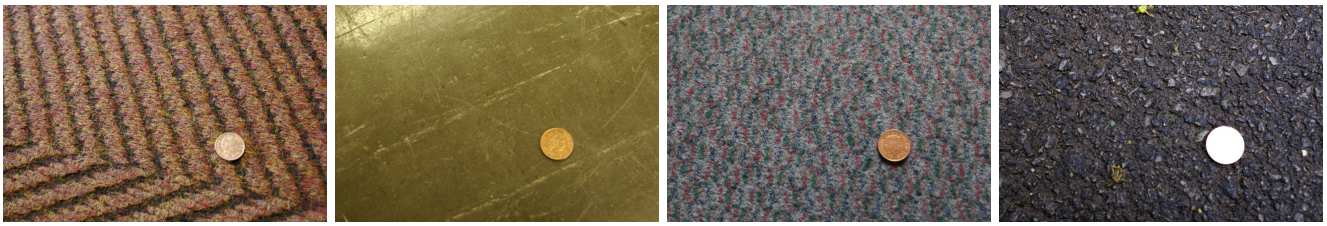
For each behavioural condition the Labjack sampled data while the robot moved in a specific direction for 16 seconds. Each 16-second trial was then repeated 4 times for each of the behavioural conditions, and 4 times for the spot command. Data from the Labjack is passed to the computer in packets of 160 recordings each (at 2kHz. 0.08s, or 200 packets per 16 second trial) to ensure seamless acquisition of the data with no bottleneck at the Labjack. When the data is sampled there is no guarantee that the robot will be moving, so to aid the classification the packets acquired when the robot was stationary were removed. A MATLAB script was written to remove any packets where the variance in the signal was below $5\text{x}10^{-5}$. This automated process provided a clean signal for classification. The remaining packets were restructured to windows for classification that were 800 recordings long (0.4s, or 40 per 16 second trial). These restructured windows were then separated evenly into training and test data sets by assigning all odd numbered samples to the training set, and all even numbered samples to the test set. Only the training set was then used to develop any classifiers. Classifiers were then applied to the test set, and their performance measured using confusion matrices.

B. Analysis

Classification was performed using a template based classifier on the fast fourier transfer (FFT) of the windows. The FFT was carried out on each window in the training dataset. The absolute values of each squared FFT'd signal were taken and the first value was set to zero to remove the mean signal component. The spectra were then smoothed by convolving with a 10-point uniform window, then L1-normalized. Templates were generated by averaging all FFT'd windows in the training set for each surface (rough and smooth carpet, vinyl and tarmac) and behavioral condition (clockwise, anticlockwise and spot). Templates from the clockwise condition are shown in fig. 7. The templates were compared to the FFT of each window in the test data set in turn using a sum of squared errors over the spectra, and hard classification was made by selecting the template with the lowest error.

C. What we found

Table (I) and (II) show confusion matrices for four-way texture discrimination in the anti-clockwise and clockwise movement conditions respectively. Though mean classification performance is good in both movement



(a) Rough carpet (b) Vinyl Flooring (c) Smooth Carpet (d) Tarmac

Fig. 5. Photos of all the surfaces, with 1p coins for scale.

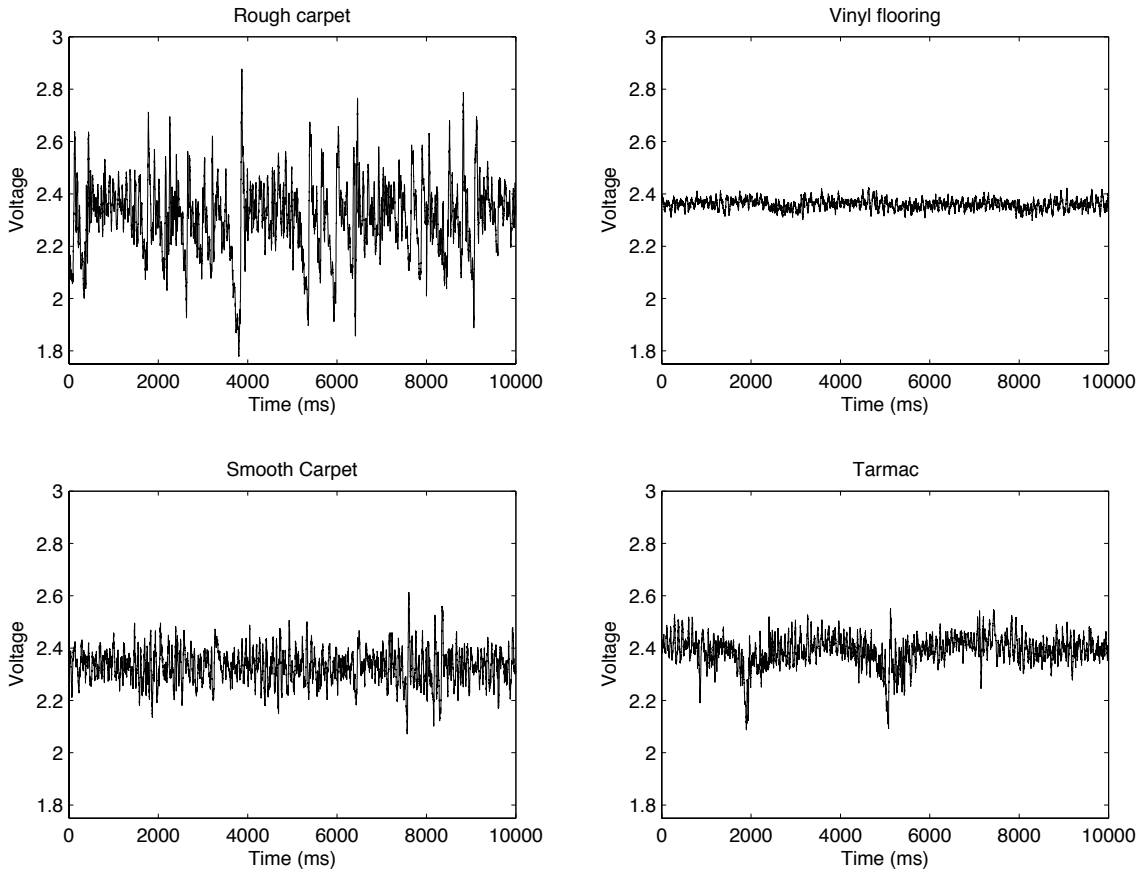


Fig. 6. Example signals from the artificial whisker sampled from the four floor surfaces.

conditions, there is quite a significant difference in performance between the conditions. Overall, classification in the anti-clockwise movement condition is better (Mean correct = 72%, vs 64%), though classification of the vinyl flooring is more accurate in the clockwise condition (79% vs 68% correct). Smooth carpet classification is extremely poor in the clockwise movement condition (33% correct), as can be seen in the third column of table (II).

Table (III) shows the performance of the classifier in the spot program movement condition. Though overall performance is poorer in this condition than the previous single direction of movement tests, it is still well above chance. Also, the confusion matrix clearly shows that performance overall is very good in three of the four conditions, with misclassification of smooth carpet (11% correct) accounting for the low mean correct score.

In a further analysis, we tried classifying each behavioural condition with the templates from the other, clockwise templates on the anti-clockwise data and vice versa, to see how robust these templates were. The performance in this test was much poorer, with the mean correct scores on both datasets falling to 52%.

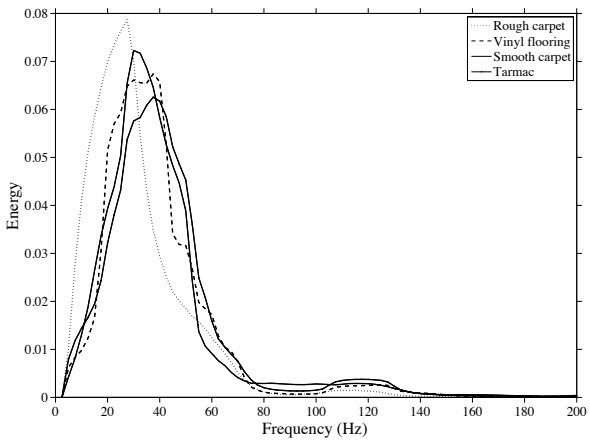


Fig. 7. Templates used for surface classification in the clockwise movement condition

TABLE I

CONFUSION MATRIX FOR CLASSIFICATION IN THE ANTI-CLOCKWISE DIRECTION OF MOVEMENT CONDITION. MEAN CORRECT = 72%

AC	Rough carpet	Vinyl	Smooth carpet	Tarmac
Rough carpet	78	6	12	10
Vinyl	1	54	4	12
Smooth carpet	0	1	43	4
Tarmac	1	19	21	54
Correct %	97.5%	67.5%	54%	67.5%

TABLE II

CONFUSION MATRIX FOR CLASSIFICATION IN THE CLOCKWISE DIRECTION OF MOVEMENT CONDITION. MEAN CORRECT = 64%

C	Rough carpet	Vinyl	Smooth carpet	Tarmac
Rough carpet	64	6	26	11
Vinyl	0	63	0	9
Smooth carpet	11	1	27	8
Tarmac	5	10	27	52
Correct %	80%	79%	33%	65%

TABLE III

CONFUSION MATRIX FOR CLASSIFICATION IN THE 'SPOT' MOVEMENT CONDITION. MEAN CORRECT = 56%

Spot	Rough carpet	Vinyl	Smooth carpet	Tarmac
Rough carpet	60	7	48	7
Vinyl	0	48	1	2
Smooth carpet	13	3	9	9
Tarmac	7	22	22	62
Correct %	75%	60%	11%	77.5%

IV. DISCUSSION

These results show that the performance of these classifiers are dependent on the movement of the robot. Though classification performance is good on most surfaces across all conditions, movement would need to be taken into account for classification of all surfaces to be optimised. There are two reasons for this discrepancy in classification for different movement conditions. Firstly, as can be seen in fig. 4, the whisker was not mounted precisely perpendicular to the outer case of the Roomba, so movement in one direction or another would deflect the whisker differently. For example by moving in an anticlockwise direction the whisker tip is dragged behind the rest of the shaft, whereas clockwise movement forces the tip forwards slightly into the ground. These two different contact patterns could affect the signals coming from the whiskers, by either magnifying or suppressing small bumps in the surface. Secondly the movement of the robot itself was quite different in the two conditions, with anti-clockwise movement being combined with a slight forward movement to produce a smooth spiraling motion. On the other hand clockwise motion was a pure rotation, and as a result the motion of the robot was much less smooth. Without taking the robot apart, or knowing what the Roomba's operating system actually commands, we cannot know the reasons for these differences exactly. We can suggest that the spiraling anti-clockwise movement combined with a rotational clockwise avoidance movement would provide enough flexibility for efficient cleaning while keeping the movement choices at any one time low. This would be desirable for a robot with the Roomba's operational goal to work robustly and efficiently.

What is clear is that for texture discrimination in a mobile robot the movement of the system, and as a result the trajectory and geometry of the whisker contact with the surface, would need to be known before an accurate decision can be made about the surface. Given that any robot movement would be controlled by the same system that is hoping to process the whisker information this motor-efference signal would be available, making deciphering the signals easier. Robot movement could be controlled to accentuate certain aspects of the signal, in this case by turning in the most discriminating direction. For example, if a general 'rough' category decision has been made, the robot could then move in a way that provides the most accurate 'rough carpet'/ 'tarmac' discrimination. Following on from our findings here, the robot could always move clockwise upon start-up to accurately check if the surface is vinyl flooring, then move anti-clockwise to better discriminate the three other surfaces. In this way, all of the robot's movements could be optimised for a given task, and decisions will be as quick and accurate as possible.

Additional, biologically inspired, methods could be implemented in future to highlight any differences between very similar textures. Arabzadeh et al [6] showed that the pattern of spikes in the primary afferents provide more information about the texture encountered by the whisker

than spike number alone. They suggest that the pattern of spikes in the trigeminal ganglion and cortex are representing the temporal pattern of deflections at the whisker tip. Others have proposed that the rats may sweep their whiskers along textured surfaces in such a way as to amplify the frequency of the texture by monitoring the resonance of the different whiskers [18], [19]. Rats adjust the movement of their whiskers to extract the most information possible from the surface by ensuring frequent and numerous light touches [17], and by controlling the speed and spread of the whiskers before initial contact and between multiple contacts [10]. Whisking speed would affect the fundamental frequency of the whisker deflections in the signal much like a record being played at the wrong speed, with higher or lower speed causing frequencies to increase or decrease, respectively.

V. Concluding remarks and future work

Simple real-world floor texture classification is possible and could for example be used to adjust the Roomba's mechanical cleaning mode for different surfaces. A more general application of texture classification is as an observation category for probabilistic navigation models (e.g. [22]). In navigation, the task is to infer the current location from noisy observations and odometry. Knowing the surface type allows estimation of the noise in the odometry and allows more accurate fusion with sensor data. For tactile-only robots, such as those operating covertly in hostile environments, regions of texture and boundaries between them could form useful observations to provide location likelihoods.

Template based classification is a discriminative method, and its success suggests that whisker signals could provide immediate information about the surfaces in the world with minimal processing. This is in contrast to generative modelling which needs to consider all possible candidate causes of the signal and search for the best top-down explanation. As they work for artificial whiskers, discriminative methods might also be found in low-level biological whisker sensory-motor loops for prey capture or avoidance, having less latency than hierarchical cortical processing.

In previous work using constrained contacts with *above*-floor objects [9] we showed that the accuracy of several texture discriminators is greatly improved when given knowledge of the geometry of the contact. (This is not an issue with floor surfaces, which are constantly in contact with the whisker.) We have shown in simulation [8] that larger-scale templates can be used to estimate the contact geometry with above-floor objects (alternatively, generative beam-theoretic methods also exist for such estimation [2]). Future work will explore how to fuse such geometric knowledge with texture classifiers to improve the accuracy in real-world, mobile, above-floor texture discrimination.

VI. Acknowledgements

The authors would like to thank members of the Active Touch Laboratory at Sheffield (ATL@S) and the Adaptive Behaviour Research Group (ABRG) for their suggestions, knowledge and patience.

This work was supported by the EPSRC Doctoral Training Scheme and EU grants ICEA (IST-027819) and BIOTACT (ICT-215910).

References

[1] Ehud Ahissar and David Kleinfeld. Closed-loop neuronal computations: focus on vibrissa somatosensation in rat. *Cereb Cortex*, 13(1):53–62, 2003.

[2] J Alexander Birdwell, Joseph H Solomon, Montakan Thajchayapong, Michael A Taylor, Matthew Cheely, R Blythe Towal, Jorg Conradt, and Mitra J Z Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *J Neurophysiol*, 98(4):2439–2455, 2007.

[3] G E Carvell and D J Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *J Neurosci*, 10(8):2638–2648, 1990.

[4] Mathew E Diamond, Moritz von Heimendahl, and Ehsan Arabzadeh. Whisker-mediated texture discrimination. *PLoS Biol*, 6(8):e220, 2008 Aug 26.

[5] Mathew E Diamond, Moritz von Heimendahl, Per Magne Knutsen, David Kleinfeld, and Ehud Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[6] Stefano Panzeri Ehsan Arabzadeh and Mathew E. Diamond. Deciphering the spike train of a sensory neuron: Counts and temporal patterns in the rat whisker pathway. *The Journal of Neuroscience*, 26(36):9216–9226, 2006.

[7] Daniel E Feldman and Michael Brecht. Map plasticity in somatosensory cortex. *Science*, 310(5749):810–815, 2005.

[8] Charles W Fox, Mat Evans, Martin Pearson, and Tony J Prescott. Towards temporal inference for shape recognition from whiskers. In Subramanian Ramamoorthy and Gillian M. Hayes, editors, *Towards Autonomous Robotic Systems*, pages 226 – 233, 2008.

[9] Charles W Fox, Ben Mitchinson, Martin Pearson, Anthony G Pipe, and Tony J Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.

[10] Robyn A. Grant, Ben Mitchinson, Charles W. Fox, and Tony J. Prescott. Active touch sensing in the rat: Anticipatory and regulatory control of whisker movements during surface exploration. *Journal of Neurophysiology*, 101:862–874, 2009.

[11] Joerg Hipp, Ehsan Arabzadeh, Erik Zorzin, Jorg Conradt, Christoph Kayser, Mathew E Diamond, and Peter Konig. Texture signals in whisker vibrations. *J Neurophysiol*, 95(3):1792–1799, 2006.

[12] Makoto Kaneko. Active antenna. *Robotics and Automation, IEEE Transactions on*, pages 2665–2671, 1994.

[13] DaeEun Kim and Ralf Moller. A biomimetic whisker for texture discrimination and distance estimation. In S. Schaal et al, editor, *From Animals to Animats 8, Proceedings of the International Conference on the Simulation of Adaptive Behaviour*, pages 140–149. MIT Press, 2004.

[14] DaeEun Kim and Ralf Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55:229–243, 2007.

[15] David Kleinfeld, Ehud Ahissar, and Mathew E Diamond. Active sensation: insights from the rodent vibrissa sensorimotor system. *Curr Opin Neurobiol*, 16(4):435–444, 2006.

[16] Melexis. *www.melexis.com/assets/mlx90333_datasheet_5276.aspx*.

[17] Ben Mitchinson, Chris J Martin, Robyn A Grant, and Tony J Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.

[18] Maria A Neimark, Mark L Andermann, John J Hopfield, and Christopher I Moore. Vibrissa resonance as a transduction mechanism for tactile encoding. *J Neurosci*, 23(16):6499–6509, 2003.

[19] Jason. Ritt, Mark L. Andermann, and Christopher Ir. Moore. Embodied information processing: Vibrissa mechanics and texture features shape micromotions in actively sensing rats. *Neuron*, 57:599–613, February 2008.

[20] Marcin Szwed, Knarik Bagdasarian, and Ehud Ahissar. Encoding of vibrissal active touch. *Neuron*, 40(3):621–630, 2003.

[21] Marcin Szwed, Knarik Bagdasarian, Barak Blumenfeld, Omri Barak, Dori Derdikman, and Ehud Ahissar. Responses of trigeminal ganglion neurons to the radial distance of contact during active vibrissal touch. *J Neurophysiol*, 95(2):791–802, 2006.

[22] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, March 2002.

Autonomous Robots manuscript No.
(will be inserted by the editor)

Contact type dependency of texture classification in a whiskered mobile robot

Charles W. Fox · Ben Mitchinson · Martin J. Pearson · Anthony G. Pipe · Tony J. Prescott

Received: date / Accepted: date

Abstract Actuated artificial whiskers modeled on rat macrovibrissae can provide effective tactile sensor systems for autonomous robots. This article focuses on texture classification using artificial whiskers and addresses a limitation of previous studies, namely, their use of whisker deflection signals obtained under relatively constrained experimental conditions. Here we consider the classification of signals obtained from a whiskered robot required to explore different surface textures from a range of orientations and distances. This procedure resulted in a variety of deflection signals for any given texture. Using a standard Gaussian classifier we show, using both hand-picked features and ones derived from studies of rat vibrissal processing, that a robust rough-smooth discrimination is achievable without any knowledge of how the whisker interacts with the investigated object. On the other hand, finer discriminations appear to require knowledge of the target's relative position and/or of the manner in which the whisker contact its surface. (146 words)

Keywords whiskers · vibrissae · tactile sensing · texture discrimination

Charles W. Fox, Ben Mitchinson and Tony J. Prescott
Adaptive Behavior Research Group
Department of Psychology
University of Sheffield
Tel.: +44(0)114 222 6591
Fax: +44 (0)114 276 6515

Martin J. Pearson and Anthony G. Pipe
Bristol Robotics Laboratory
Du Pont Building
Bristol Business Park
Coldharbour Lane
Frenchay
Bristol
Tel.: +44(0)117 328 6321

1 Introduction

The long facial whiskers (macrovibrissae) of rodents such as rats and mice provide these animals with a sophisticated and multi-functional tactile sensory apparatus [21], [8]. Using their vibrissae, rats are able to make fine-grained discriminations based on surface texture with a resolution similar to that achieved by the human fingertips [10]. The exploration of texture by the vibrissae is an active process which involves sweeping ('whisking') the whiskers across the target surface at a rate of 5-15Hz during typical exploratory behaviour [4]. The brushing of each vibrissa against the target surface induces vibrations in the whisker shaft that are transduced into neural signals by mechanoreceptors in the whisker follicle [23]. The neural encodings of these vibration patterns, formed across the vibrissal array, therefore provide the basis for the texture discrimination capabilities shown by the rat [1], [16]. High-speed videographic observations suggest that contact on only a small number of macrovibrissae over 3-5 whisking cycles is sufficient for reliable texture discrimination [10].

Tactile-based texture discrimination capabilities, similar to those of rodent vibrissae, would be useful in a variety of robot applications. For example, intelligent household appliances could use whisker-based sensing to optimise cleaning strategies to suit different surface types. Similarly, autonomous vehicles such as planetary rovers could use them to assess surface properties such as friction in order to better control locomotion and exploration [33]. Whisker sensors will be especially useful for robot local navigation and object recognition in any circumstances were the capacity to use vision sensors, or active sensors such as lasers or infra-red, is compromised by poor air quality (smoke- or dust-filled), or by turbid water. Such circumstances are often encountered

during search and rescue missions, in environments that are hazardous to humans, making the capacity to employ autonomous robots alongside human search teams particularly useful.

Research into robotic whisker systems began in the mid 1980s, and has continued since as a small but flourishing sub-field of bio-inspired engineering. The investigated solutions have differed considerably in a number of important respects. First, choice of material for the vibrissal shaft has ranged from steel wires [31], [18] through actual rat vibrissae [22], to specially-moulded composites [28]. Sensor transduction has likewise used a variety of solutions. Early robotic whisker implementations used potentiometers to measure the torque of steel whiskers as they made contact with surfaces [18]; [32]. More recent work has used electret microphones [22], resistive arrays [35], strain gauges [28], [36], piezoelectric [19], and magnetic ('Hall effect') sensors [19]; [20]. A number of systems have employed actuated whiskers in order to better simulate the active sensing capacities of the rat vibrissae. Solutions to the actuation problem have included miniaturised conventional electric motors, and actuators with more muscle-like properties such as shape-memory alloys [28] or 'air-muscles' [37]. In functional terms, previous work on robotic whisker sensing has provided proof of principle that artificial vibrissal systems can compute estimates of distance and shape [37], [36], [32], [20] and can distinguish between textures with different spatial frequencies [12], [19]. In the following we briefly summarise past research specifically related to the texture discrimination problem.

Lungarella et al. [22] investigated texture discrimination by gluing either rat whiskers, or similar-sized synthetic whiskers, to the diaphragm of an electret microphone thereby transducing vibration of the whisker shaft into acoustic signals. In one set of experiments [13] actuated whiskers, attached to a stationary robot 'head', were made to brush against a rotating textured drum. Data was analyzed as time-averaged windowed-spectra and discrimination performed by clustering vectorised spectra. When the contact period between the whisker shaft and the drum surface was known the system was able to discriminate eight distinct textures using eight whiskers with exposure to ten whisk cycles per texture (80 individual contacts); with unknown contact periods this discrimination capacity fell to five textures for the same number of whisks. In a second series of experiments [12] a 3-layer sigmoidal regression neural network was used for classification having 10 input bins for each of 6 whiskers, 10 hidden nodes, and 4 output classes. Using a controlled set of seven known head positions the system achieved 70% accuracy in classifying four textures using data from six whiskers each making a single whisk (i.e. six individual contacts). No significant gain was obtained by increasing the number of hidden nodes. In a further experiment where the robot platform was made mobile (i.e. uncontrolled head positions), classification was described as "not entirely reliable" and no quantitative results were given.

Kim and Möller [19] performed similar experiments using a non-actuated, 26mm steel whisker attached to a fixed base contacting a rotating, textured drum. Direct measurements of direction and amplitude of deflection of the whisker shaft were made with magnetic and piezo sensors. Deliberately choosing textures whose spectra did not interfere with the theoretical resonant whisker frequencies, a neural network was again used to classify the low-band spectra achieving a success rate of 85% accurate classification over seven textures. These authors noted that classification was better with shorter whiskers and that the classifier fails when the whisker base was allowed to move.

Schultz et al. [33] described a robot with 190mm whiskers whose movement was detected using strain sensors, and presented spectra for five surfaces of three types, that can be distinguished by eye. No quantitative analysis of the data were provided.

The Darwin IX robot, described by Seth et al. [35], used 40mm polyamide whiskers that, unlike natural vibrissae, can directly detect the amount of bending along the whole whisker, not just at the base. Unlike the other robots above, data was analyzed in the time domain, with 20 lagged curvature inputs fed into a biologically inspired neural network classifier. The robot was given the task of distinguishing between two textures consisting of patterns of pegs embedded in walls. In a mobile robot setting, two-way classification reached 97% accuracy, though it is unclear how many whisk cycles were typically used to make each decision.

The above systems have applied both generative methods, such as solving the beam equations to analytically remove whisker resonant frequencies [19], and data-mining methods, such as neural network training algorithms [19], [12], [22] to address the texture classification problem. An alternative approach – to be taken here – is to search for useful statistics of the data or 'features' that can then be provided as inputs to standard classifiers. Such features may be identified by careful analysis of the data, by hand or using machine methods where appropriate. This type of hypothesize-and-test method of feature discovery has produced contest-winning results in other time-series classification domains such as music retrieval [11]. A particular method for feature hypothesis generation is to examine the codes and processing mechanisms used by biological classification systems. In the domain of

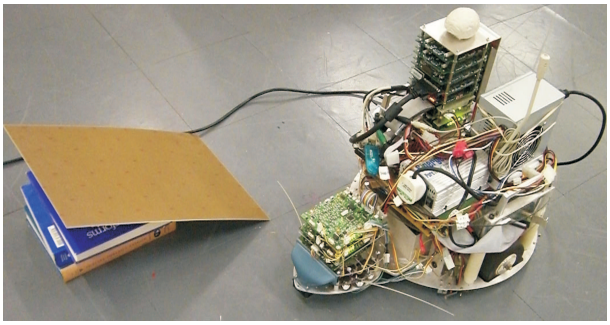


Fig. 1: WhiskerBot, whisking on a surface during the experiments.

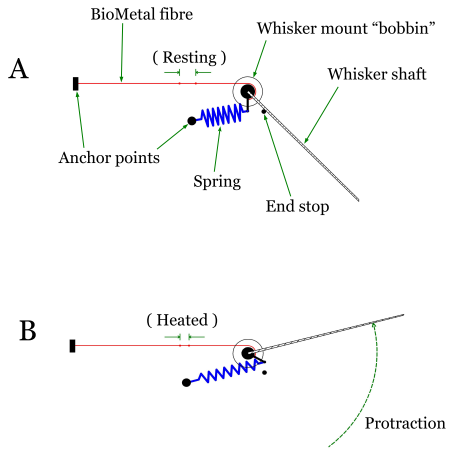


Fig. 2: WhiskerBot's actuation mechanism, in its resting (A) and heated (B) states. The BioWire contracts when heated, rotating the whisker.

vision, for instance, a classifier modeled on the primate visual cortex has recently out-performed several benchmark computer vision systems on a large database of natural images [34]. Here we will take inspiration from studies of coding in the neural pathways of the rat whisker system as one source of candidate features for resolving texture classifications problems for artificial vibrissae. One previous study [17] has taken a similar approach in which frequency-based features inspired by the analysis of activity in whisker ("barrel") sensory cortex were used to successfully classify signals obtained from an actuated 80mm metal whisker moved against textured surfaces from a known fixed starting position. The features identified by [17], along with others inspired by our own investigations of coding in the vibrissal primary afferent neurons [23], will be evaluated below for their usefulness in discriminating texture in a robotic whisker context. The performance of the above methods will also be compared with a neural network classifier.

In summary, in previous work, only [12] and [35] have attempted texture classification using artificial whisker signals obtained from unconstrained whisker-base positions and neither reported quantitative results under these circumstances. This raises the question of how robust whisker-based texture classifiers are to differences in the way in which the artificial whisker interacts with the target surface. This, in turn, is dependent on the relative position and orientation of the whisker with respect to the surface and on the nature of the whisker actuation strategy used. In the current study we therefore investigated the problem of classification of signals from different surface textures obtained using robot-mounted actuated whiskers in four different experimental settings. In the first two settings, the robot (and hence the whisker base), was stationary at one of two fixed distances from the target surface. In the third, the position of the robot with respect to the target surface was variable, as the consequence of the robot's prior movement from a fixed starting point. Finally, in the fourth setting, data was obtained from 500 arbitrary positions of the robot relative to the surface.

2 Methods

2.1 Artificial whisker system

As part of a more general investigation of functional capabilities of artificial whisker systems we previously constructed an active whisker array attached to a mobile robot platform [28, 29], see figure 1. In this system each artificial whisker was a 200mm moulded glass-fibre composite rod approximately four times larger than, but respecting the general form of, a rat whisker, and gauged at the base to measure strain (bending) in two dimensions. The whisker was 200mm in length, tapered from 2mm in diameter at its base to approximately 0.7mm diameter at its tip, and the radius of curvature was 250mm.

Each whisker was actuated in a single axis (anterior-posterior) using a shape metal alloy wire (BioMetal) and a spring, shown in figure 2. Heating the BioMetal wire, using a pulse width modulated current source, caused a longitudinal contraction of the BioMetal which was translated into a rotational force around the drive shaft onto which the whisker was mounted, thereby actively protracting the whisker. When the current through the BioMetal was switched off, the wire would cool and return to its original length assisted by the spring to generate a passive retraction phase.

The drive signals to the BioMetal actuators were directly derived from a pair of coupled motor pattern gen-

erators, implemented as a spiking neural network, nominally operating as an open-loop controller. Inhibitory and excitatory projections from the strain sensors at the base of each whisker would modulate the behaviour of the pattern generators when contacts were made with the environment, thus affecting the drive signals to the BioMetal wires. This feedback control system was adopted as it best fits both anatomical (no proprioception in mystacial musculature [14]) and ethological observations (contact mediated adaptation [24]) of rats. For further details of the robot whisker morphology, actuation, and transductions mechanisms see [26], [29].

For the experiments described below a single whisker moved (whisked) at about 1Hz, and strains were sampled at 10kHz. Note that our use of strain data as input to a whisker-based texture classification system differs from previous work which has generally used direct recordings of angles and deflection amounts as classifier input.

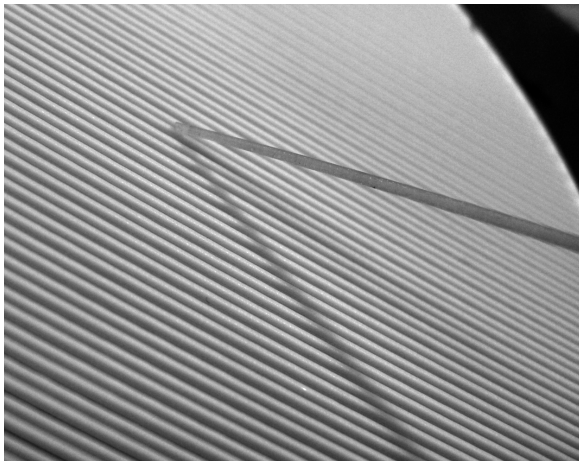


Fig. 3: Close-up of the whisker tip (and its shadow) during a typical brushing contact on the rough surface, showing the angle of contact. For brushing, the distance from robot base to surface is set to allow the whisker to sweep over the surface, making contact without sticking; for dabbing, the distance is reduced so that the whisker is pressed into a point on the surface.

2.2 Data collection

Four distinct data-sets were obtained each containing multiple trials of whisking against different textured surfaces, the difference between sets being the robot's *experimental setting*, that is, the position and movement of the robot platform relative to the investigated surface. These datasets were:

i. Fixed Brushing (FB): Here the robot platform and surface are fixed in the same positions over all trials, allowing the whisker to gently brush past the surface for about half of the whisk cycle.

ii. Fixed Dabbing (FD): Again the platform and surface are fixed in the same positions over all trials, but placed closer together than above, so the whisker covers only around 5-10mm of surface before getting stuck for the rest of the contact period. Feedback control, as described above, was switched on for fixed dabbing.

iii. Mobile Brushing (MB): The robot begins each trial away from the surface, then travels towards it before stopping and performing brushing. The initial robot positions are roughly similar across trials: starting about 1m from the surface, with a standard deviation radius of about 10mm and standard deviation orientation of about 10 degrees. The robot stops moving as soon as a surface is found, then brushing behavior is achieved by controlling whisker actuation using sensory feedback as described above.

iv. Random positions: Whisks were performed at random relative positions and orientations, as chosen by a human instructed to position the surface by hand at as wide a range of locations as possible relative to a fixed whisker base. This data thus includes both brushing and dabbing type contacts, with azimuth, elevation and radius drawn from an approximately flat distribution within the potential contact area of the whisker.

For each of (i-iii), whisks were performed against four different types of textured surface of increasing roughness: a smooth perspex sheet, fine-grade P600 sandpaper, medium-grade P240 sandpaper, and a rough perspex sheet with regular 2mm-spaced grooves. Note that sandpaper grit is non-regular, having some average number of bumps per centimeter, but not at a fixed spatial frequency. 65 whisks were obtained for each FB surface type; 52 for each FD surface type, and 34 for each MB surface type. These whisks were split into equal sized training and test sets. Contact was made by the whisker tip or within 30mm of the tip, and a typical contact is shown in fig. 3. The whisker was in contact with the surface for around 20mm of travel during brushing, and did not travel during dabbing. Heuristics were used to remove bad data which included the whisker becoming stuck on the surfaces and failing to make contact (both especially prevalent in the mobile setting). For (iv), 500 whisks each were performed against the smooth and rough perspex surfaces only.

Electronic appendix 1 shows video examples of the interaction between the whisker and smooth, P240 and rough surfaces, for settings FB, FD and MB respectively.

2.3 Feature extraction

Fig. 4 show examples of typical strain time-series data collected during the experiment in settings (i-iii). At least within each dataset, it appears possible to distinguish between data acquired from rough and smooth surfaces by eye. It is also possible to play the whisks as audio files and identify some differences by ear. Intuitively one might think of the whisker as like the needle in a record player, picking up the audio details from the record groove, however this analogy is far from perfect—in all but highly constrained situations, whiskers exhibit 'sticks, slips, skips and rings' [30] rather than smoothly brushing over the surface. For instance, both natural and artificial vibrissae often get stuck on rougher textures, then slip quickly off the sticking point and resonate (ring) at their natural harmonic frequency. Skips are like stick-slip combinations but where the whisker is not completely stopped.

We will present three pairs of simple features in addition to a replication of the neural network classifier used by [12]. The first pair of features is inspired by inspection of sampled whisker signals. The other two pairs are inspired by previous neurophysiological [2] and modeling [17], [23] investigations of the rat vibrissal system. In each case we present raw scatter plots of the data classes under the features, and confusion matrices obtained using a standard Gaussian classifier [7]. The scatter plots show all recorded data; the confusion matrices are constructed by splitting the data into equally sized (for each class) training and test sets, and reporting test performance. The Gaussian classifier operates by assuming that for each class c , data points x_i in feature space are generated by a Gaussian with unknown mean μ_c and covariance Σ_c,

$$p(x|\mu_c, \Sigma_c) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \quad (1)$$

$$Z = (2\pi)^{\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \quad (2)$$

and for each class finds maximum likelihood parameters

$$\mu_c, \Sigma_c = \arg\max_{\mu_c \Sigma_c} \prod_i P(x_i|\mu_c, \Sigma_c) \quad (3)$$

for the training data in that class. On test data x_j , it assigns the maximum likelihood ('winner-take-all') class

$$c_j = \arg\max_c \prod_j p(x_j|\mu_c, \Sigma_c). \quad (4)$$

3 Results

3.1 Neural network replication

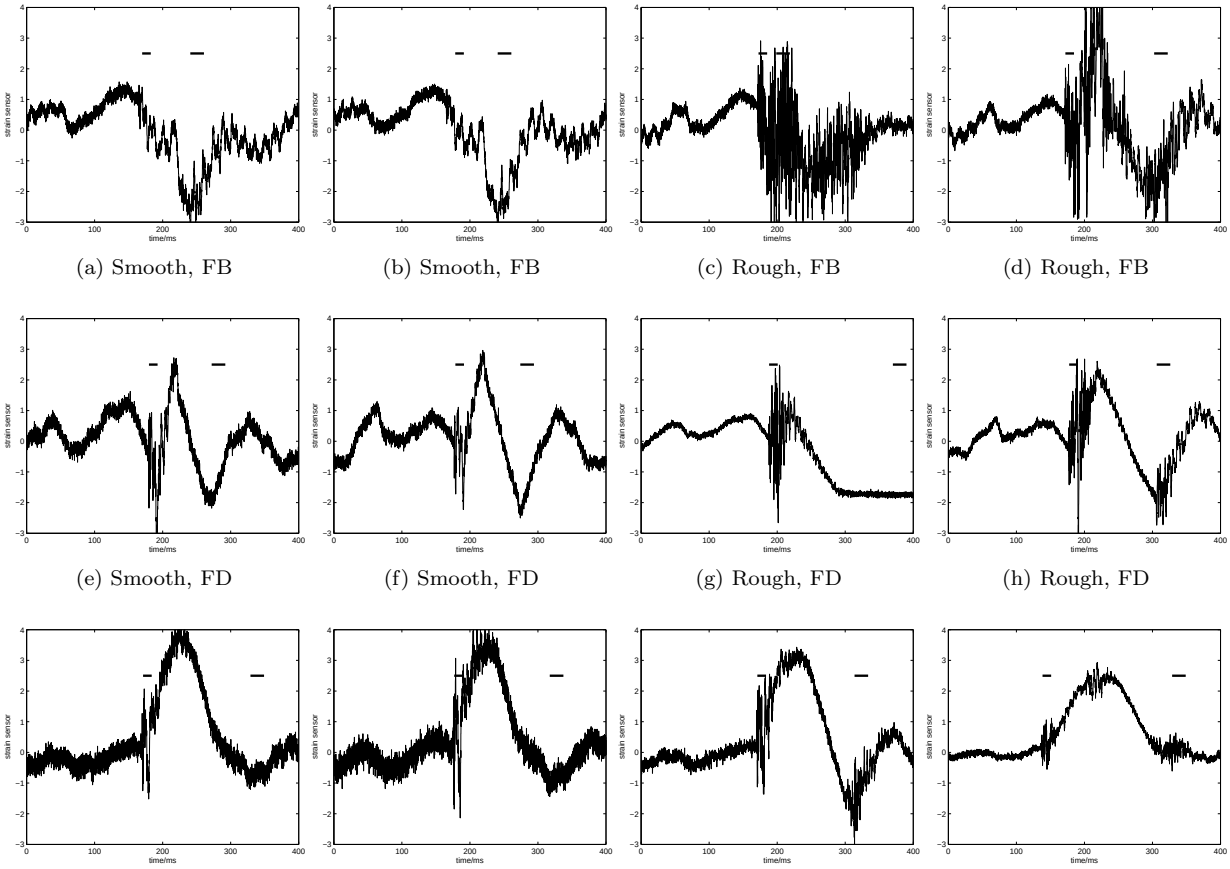
We first give results for a standard neural network classifier as used in previous studies. Following the methodology of [12], the strain time-series from a portion of the contact period was Discrete Fourier Transformed (DFT), convolved with a 70 point Blackman window, and quantized into 10 bins. Unlike Fend et al., our data does not contain information about the start and end of the contact periods, and contact periods are highly variable in the mobile setting especially. Therefore we made use of the onset detector described below to estimate the start point, and took a conservative 1024 points of time series following it for the analysis. As in Fend et al., a three-layer multilayer perceptron was used, having between 10 and 14 hidden nodes. We used neurons with the transfer function $y = 2/(1 + \exp(-2x)) - 1$ for hidden and output layers with one output node per class, and target values of ± 1. Classification was by winner-take-all and training used the Levenberg-Marquardt numerical optimisation algorithm [7]. (Window size, transfer function and output representation were not detailed by Fend et al., therefore we chose the best-performing values found by trial and error.) The training regime of Fend et al. was replicated: the network was trained ten times from random initial weights and on random test and training partitions of the data, and the best test set performance is reported.

The confusion matrices in Table 1 give results for the individual settings (*FB*=Fixed Brushing; *FD*=Fixed Dabbing; *MB*=Mobile Brushing) and for a single network trained on all three (*ALL*) settings together. The confusion matrix rows represent the true classes and the columns the predicted classes (hence perfect classification would correspond to zeros in all table elements except the diagonals) Assuming a reference Beta prior [5] on the probability of correct classification, the posterior expectation and standard deviation (a confidence measure) of this probability are presented as percentages in the accompanying text.

We note that the neural network method gives good classification performance *within* each setting – including the mobile brushing case – but does not generalize particularly well across the settings.

3.2 Onset and Offset features

On listening to whisks from the dabbing experiment by playing them as sound waves it was apparent that whisks of rough surfaces featured pronounced clicks during the onset and offset of contact.



(a) Smooth, FB (b) Smooth, FB (c) Rough, FB (d) Rough, FB

(e) Smooth, FD (f) Smooth, FD (g) Rough, FD (h) Rough, FD

(i) Smooth, MB (j) Smooth, MB (k) Rough, MB (l) Rough, MB

Fig. 4: Examples of whisker base strain time series data in various settings and on various textures. Two examples are shown for each texture-setting combination to illustrate variability. The horizontal axis measures time in samples (1 sample=0.1ms) and the vertical axis measures strain sensor output. The two short horizontal lines indicate the onset and offset regions, defined in section 3.2.

Further analysis (see fig. 5) showed that the onset click corresponds well to increased energy in the 2-3kHz band, in a 128 sample (12.8ms), normalized onset period. The start of the onset may be defined heuristically as the first point at which the signal's 50-sample short-term standard deviation is more than ten standard deviations away from its long-term average since the start of the whisk. Formally, for strain signal $x[t]$, and sample variance function Var, we define the onset time t_{on} by

$$\sigma^2[t] = Var(x[t-127:t]) \tag{5}$$

$$\gamma^2[t] = Var(\sigma[0:1000]) \tag{6}$$

$$t_{on} = \min\{t.\sigma[t] > 10 * \gamma[t]\} \tag{7}$$

and the value of the *onset* feature by

$$X_t[\omega] = FFT(\frac{x[t-127:t] - \langle x[t-127:t]\rangle}{Var(x[t-127:t])}) \tag{8}$$

$$onset = \sum_{\omega=25}^{35} X_{ton}[\omega] \tag{9}$$

Given this distinctive response to rough surfaces, this feature would appear to be useful for general rough/smooth discrimination. It is conceivable that the precise power distribution within this band may lead to classifiers for different surface texture frequencies such as P240 and P600 sandpaper, however we have not found such a classifier and there is much discouraging within-class variation. On the other hand, an advantage of classifying from onset data only is that is invariant to the rest of the whisk, which may vary by setting.

Energy increase in the offset as shown in fig. 6 did not appear to be limited to a particular band, but was well captured simply by the variance of the time domain signal in the 10ms period after the offset start (t_{off}, defined as the time of maximum counter-whisk-direction strain following the onset):

$$t_{off} = \arg\min t\{x[t].t > t_{on}\} \tag{10}$$

Table 1: Confusion matrices for neural network classifier. Means and standard deviations of the posterior percentage correct classification are: $FB = 86 \pm 3\%, FD = 77 \pm 5\%, MB = 85 \pm 5\%, ALL = 65 \pm 3\%$.

FB	*smooth*	*rough*	*p*600	*p*240
smooth	23	0	1	1
rough	0	23	1	1
p600	1	0	20	4
p240	0	0	4	20
FD	*smooth*	*rough*	*p*600	*p*240
smooth	15	1	3	1
rough	1	15	4	0
p600	0	2	18	0
p240	1	3	2	14
MB	*smooth*	*rough*	*p*600	*p*240
smooth	9	1	0	0
rough	1	7	0	2
p600	0	0	10	1
p240	0	0	1	10
ALL	*smooth*	*rough*	*p*600	*p*240
smooth	49	3	1	2
rough	5	31	14	5
p600	4	7	25	20
p240	2	1	13	39

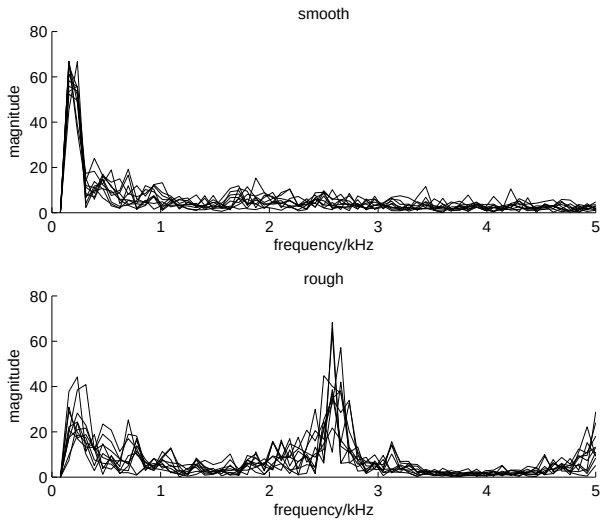


Fig. 5: FFT magnitudes for the onset region showing rough and smooth spectra for typical 128-point contact windows in the FB setting. The Nyquist frequency is 5kHz; the rough surfaces have increased power (a visible peak) around 2-3kHz which is absent from the smooth surfaces.

$$offset = Var(x[t_{off} : t_{off} + 200]) \tag{11}$$

This feature is therefore useful to discriminate between coarser surfaces such as the rough perspex and the P240 sandpaper. The offset feature may be due to the whisker sticking at the final bump of the surface – where one exists – then slipping and ringing as a result. Particularly on our rough surface, the whisker may noticeably stick in a groove and require a buildup of force to escape.

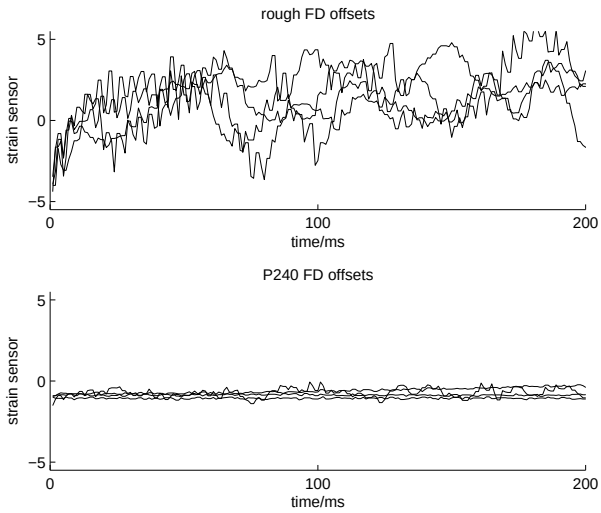


Fig. 6: Closeups of the time-domain signal at the offset ('release') stage of the whisk, for whisks against typical grooved perspex (*above*) and P240 sandpaper (*below*) in the FB setting. It can be seen that the rougher surface has much higher variance.

Table 2: Confusion matrices for the onset-offset (OO) classifier. Means and standard deviations of the posterior percentage correct classification are: $FB = 87 \pm 3\%, FD = 67 \pm 6\% MB = 52 \pm 8\%$.

FB	*smooth*	*rough*	*p*600	*p*240
smooth	28	0	1	3
rough	0	32	1	0
p600	1	0	26	5
p240	3	0	4	26
FD	*smooth*	*rough*	*p*600	*p*240
smooth	19	0	2	5
rough	1	25	0	0
p600	3	1	17	5
p240	6	1	10	9
MB	*smooth*	*rough*	*p*600	*p*240
smooth	12	0	2	2
rough	0	10	1	5
p600	1	7	7	2
p240	2	9	0	6

Fig. 7 shows that both of the above features are useful across the first three experimental settings. Specifically, the binary rough *vs.* non-rough onset regions are in similar positions across all datasets. This is important as it shows that they are not due to non-texture-related artifacts in individual settings. Conversely, the more finer-grained, offset discrimination between sandpaper and rough perspex is possible *within* each setting, but these regions vary greatly *between* settings.

We applied the same maximum-likelihood ('winner-take-all') Gaussian classifier and confusion matrix analysis as in the neural network model, to give the results of Table 2 for the individual settings.

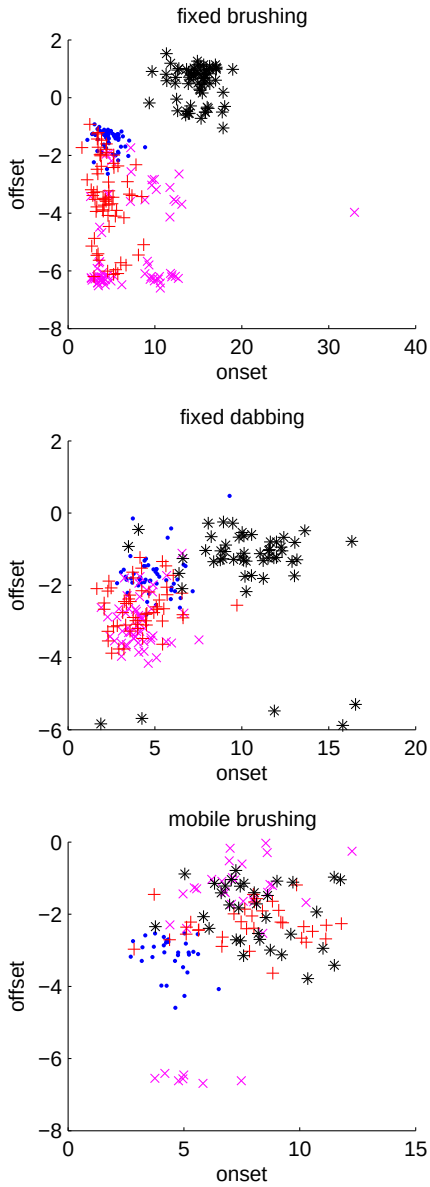


Fig. 7: Onset and offset (*OO*) feature values for data sets i–iii. Textures are: smooth perspex sheet (dots), fine-grade P600 sandpaper ('x's), medium-grade P240 sandpaper ('+'s), and rough perspex sheet with regular 2mm-spaced grooves ('*'s).

3.3 Biologically inspired frequency features from Cortical responses

Recordings of neuronal firing patterns obtained from cells in the whisker sensory ("barrel") cortex of anesthetized, head-fixed rats whisking against textured surfaces suggest the possibility of using frequency-based features to perform texture classification. *In vivo* animal experiments [2] played pure sinusoidal vibration stimulations into whiskers and found cortical cells responded to the product $X\omega$ of the amplitude and frequency of these vibrations. Texture experiments [3] generalized this, playing back pre-recorded texture vibrations, and the best found feature responded to by cortical neurons was

$$\langle X\omega \rangle = \int_{\Omega} \int_{T} |X(\omega, t)| \omega d\omega dt \tag{12}$$

This is equal to the amplitude-weighted mean frequency, or *centroid*, averaged over time. This same feature has been found useful in musical audio analysis [11], as it can represent the average musical pitch. Musical classifiers often use it in conjunction with a 'total energy' feature which gives information about the onset type of musical nodes, analogous to the onsets of whiskers contacting a surface: this suggests trying the energy feature in the whisker texture domain too. In our analysis, the average over time is implicit as we work with a large-window DFT containing all time points of interest.

[17] proposed that the centroid and energy of the contact period spectrum could correspond to the *physical* frequency and depths of bumps in the texture surface. Hipp et al. then tested this hypothesis in experiments in which actuated 80mm steel whiskers were moved against a variety of textured surfaces. For each whisk the spectrum of the whole whisk cycle was then analysed for the centroid and energy features which were input into a Gaussian classifier as in section 2.3. This system achieved 75% of the classification accuracy of a Gaussian classifier based on the entire binned spectrum (similar to our neural network in section 3.1), showing it to be a useful statistic of the spectrum with respect to the classification task.

Following these precedents we investigated the suitability of frequency-based classification of data from our own robotic whisking data. To obtain the required features we first low-pass filtered the lower fifth of the DFT spectrum to remove overall bending effects as in [17]. Here the DFT was taken over the whole contact period, excluding onset and offsets, which in the mobile robot setting (iii) was of variable length, therefore the centroid was normalized in the spectral domain (as the DFT is of variable size), and the energy normalized per unit time as follows:

$$centroid = \frac{1}{N} \sum_{\omega=N/5}^{N} \omega . X[\omega] \tag{13}$$

$$energy = \frac{1}{N} \sum_{\omega=N/5}^{N} X[\omega] \tag{14}$$

Table 3: Confusion matrices for centroid-energy (CE) classifier. Means and standard deviations of the posterior percentage correct classification are: $FB = 56 \pm 5\%, FD = 42 \pm 7\%, MB = 41 \pm 9\%$.

FB	*smooth*	*rough*	*p600*	*p240*
smooth	25	0	5	2
rough	0	30	0	3
p600	20	1	2	9
p240	8	8	2	15
FD	*smooth*	*rough*	*p600*	*p240*
smooth	22	0	3	1
rough	11	5	5	5
p600	3	2	9	12
p240	2	1	15	8
MB	*smooth*	*rough*	*p600*	*p240*
smooth	9	4	3	0
rough	1	7	5	3
p600	3	4	7	3
p240	4	3	6	4

Results using these frequency features are shown in Figure 8 and Table 3. For the fixed brushing experiment, the features performed excellently, separating smooth perspex, P240 sandpaper and rough perspex. However for the dabbing and mobile brushing settings their utility deteriorates greatly. In the dabbing case the difficulty is that there is only a short non-onset contact period in which to examine the spectrum before the whisker becomes stuck against the surface. The mobile case is more interesting, however, here variability in the distance and angle to the surface appears to detract from the usefulness of frequency features as the same surface at a larger distance will produce a higher frequency due to the increased whisk radius.

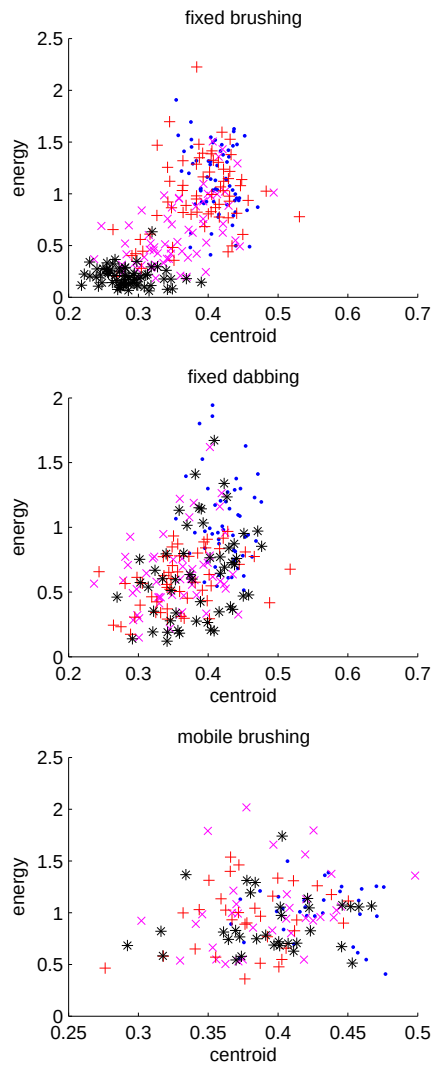


Fig. 8: Frequency centroid and energy (CE) features for experimental settings i-iii. Textures are: smooth perspex sheet (dots), fine-grade P600 sandpaper ('x's), medium-grade P240 sandpaper ('+'s), and rough perspex sheet with regular 2mm-spaced grooves ('*'s).

3.4 Biologically inspired features from the whisker primary afferents

In previous work [23], [27] we constructed a biomimetic model of transduction in the whisker follicle. This electromechanical model transforms patterns of mechanical whisker stimulation into their initial encoding as patterns of neural firing in primary afferent neurons (PAs) associated with each whisker follicle. This PA activity is passed up to the brainstem trigeminal sensory complex via the trigeminal nerve. Statistics of the spike trains from these simulated PAs may be used to drive a texture classifier.

The first part of the model is a mechanical simulation of the components of the follicle, including the whisker itself. External inputs are forces from the muscles and elastic tissues driving the follicle capsule to whisk, along with forces applied to the whisker shaft or tip by environmental obstructions. The output of the mechanical model is the strains in the tissues that are the seat of various whisker follicle mechanoreceptor families. In particular, the model derives tissue strains that are Rapidly Adapting (RA) or Slowly Adapting (SA) to tonic stimuli, leading to the most often reported dichotomy in observed PA responses. The SA and RA populations in the model each comprise 20 neurons arranged in a ring around the base of the whisker, responding to strains in different orientations.

These strains form the input to the second part of the model, consisting of a number of pre-processing operations, followed by a bank of conventional integrate-and-fire models representing the PAs themselves. The pre-processing operations are derived from observations of PA responses to time-varying stimuli. Three govern instantaneous response: gain, projection onto a vector representing maximal sensitivity (the whisker deflections are vectors), and a nonlinear response curve. In addition, temporal response is represented through stimulus adaptation and stimulus memory.

Responses of the model to simple passive whisker deflections are not complex: for example, step deflections generate phasic and tonic responses through RA and SA afferents, respectively. Responses to more complex stimuli are much less intuitive and comparable biological data are generally lacking. The results presented here use the model of [23] – the revised model of [27] would give similar values for the simple features used here.

Examples of simulated RA and SA responses obtained from the whiskered robot in the FD setting are shown in Figures 9a, 9b and 9c, along with the raw strain data for comparison. We next examine the utility of these responses as features.

On inspection, the local RA rates (summed over the 20 neurons in the population) appear to function well as an onset detector, so can be used to replace our previous, hand-picked onset detector. We use a fixed threshold to first detect the *presence* of an onset, and then employ the mean firing rate during the following 10ms period as our onset *feature* (for use in texture classification). Following the onset, SA cells continue to fire at rates which depend on both the amplitude and frequency of the whisk wave. We therefore used the mean firing rate of the SA cells (summed over the population) during a 100ms post-onset period a second feature. Fig. 10 and Table 4 show the performance of this RA/SA feature pair. As with the other features, the PA classifier separates well within each setting, but the classification boundaries are different across settings (note the changes of scale on the axes especially).

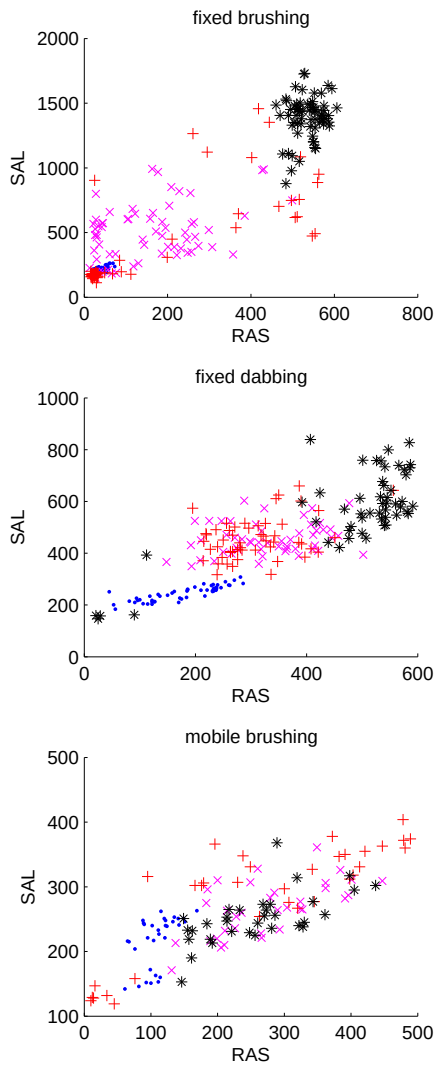
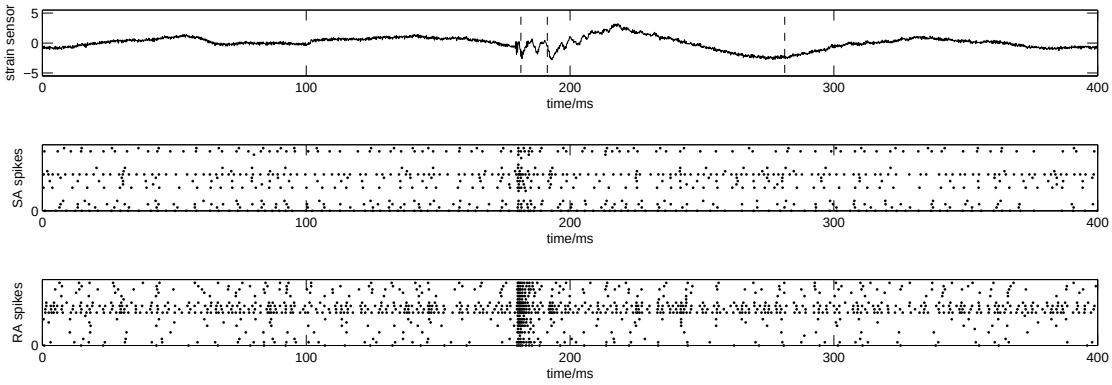


Fig. 10: Slowly Adapting (SAL) and Rapidly Adapting (RAS) simulated primary afferent (PA) cell firing rate features. Textures are: smooth perspex sheet (dots), fine-grade P600 sandpaper ('x's), medium-grade P240 sandpaper ('+'s), and rough perspex sheet with regular 2mm-spaced grooves ('*'s).

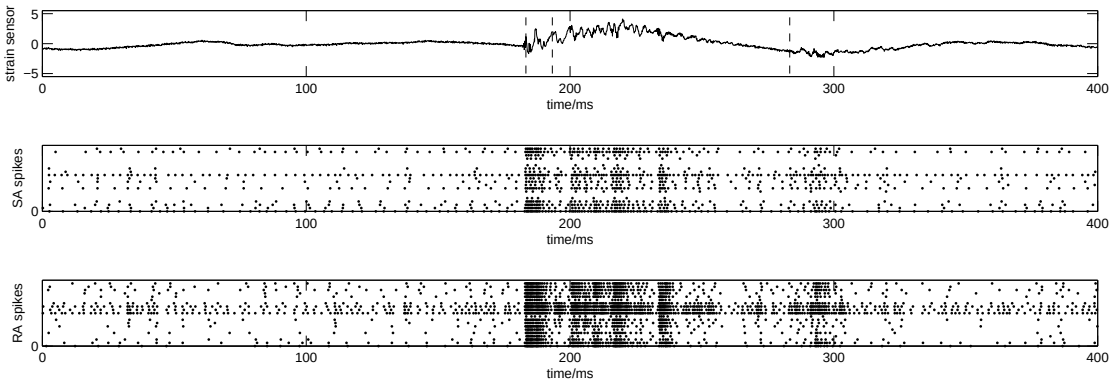
3.5 Across-setting results

While we do not have enough data from settings (i-iii) to form a sufficient sample from all possible whisker-base position and angle settings, it is still of interest to plot all three of these data sets on the same axes, in order to visualize how the features perform in a more general environment. Recall that most previous animal and robotic work has performed classification in some single 'head-fixed' setting only, raising the concern that such classifications may not generalize beyond those specific circumstances. Results from this across-setting analysis are shown in fig. 11. The subsequent three confusion matrices correspond to the three feature pairs (OO=onset-offset; CE=centroid-energy; PA=primary afferent) when used on the inter-setting data as in the scatter plots.

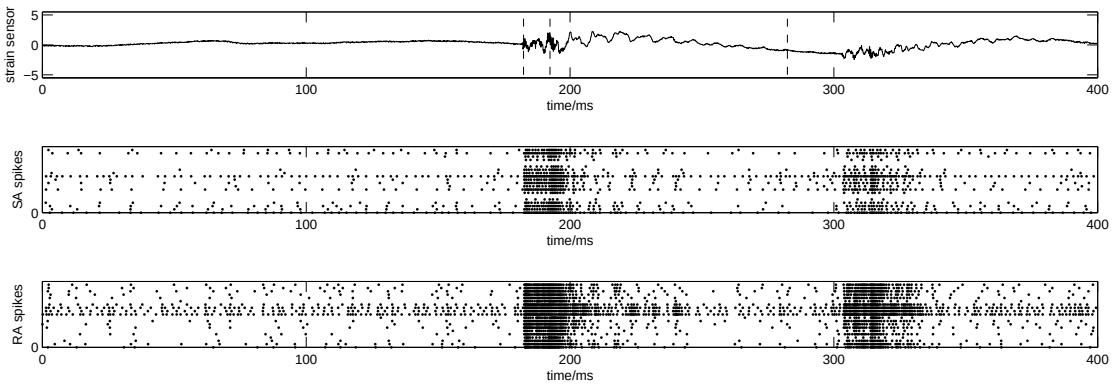
All the features except offset seem reasonably useful for generalized binary rough-smooth classification: however the 3 frequency-based features (including SA



(a) Smooth perspex



(b) P240 sandpaper



(c) Rough perspex

Fig. 9: Examples of raw base strain, Slowly Adapting spikes and Rapidly Adapting spikes for three textures, all in the Fixed Dabbing (*FD*) setting. Both neural populations are made of 20 neurons arranged in a circle around the whisker base, and which therefore respond to deflections in different orientations. The 20 neurons are shown along the vertical axes of the spike plots. The dashed lines indicate the start of the fixed-length contact window whose start location is determined using the RA onset detector. Summed RA activity for a short period following initial onset, to the second dashed line, forms the RA feature. Summed SA activity from the first to the third dashed lines forms the SA feature.

Table 4: Confusion matrices for primary afferent (PA) classifier. Means and standard deviations of the posterior percentage correct classification are: $FB = 76 \pm 4\%, FD = 73 \pm 5\%, MB = 71 \pm 7\%$.

FB	*smooth*	*rough*	*p600*	*p240*
smooth	32	0	0	0
rough	0	33	0	0
p600	19	1	7	5
p240	0	0	5	28
FD	*smooth*	*rough*	*p600*	*p240*
smooth	26	0	0	0
rough	1	21	0	4
p600	0	2	14	10
p240	0	1	10	15
MB	*smooth*	*rough*	*p600*	*p240*
smooth	15	0	1	0
rough	1	11	0	4
p600	2	1	11	3
p240	1	3	3	10

Table 5: Confusion matrices each classifier, over all settings. Means and standard deviations of the posterior percentage correct classification are: $OO = 60 \pm 4\%, CE = 46 \pm 4\%, PA = 61 \pm 4\%$.

OO	*smooth*	*rough*	*p600*	*p240*
smooth	60	2	0	12
rough	4	60	4	7
p600	18	12	25	20
p240	28	7	6	35
CE	*smooth*	*rough*	*p600*	*p240*
smooth	65	2	3	4
rough	21	44	1	0
p600	35	17	7	16
p240	22	26	8	20
PA	*smooth*	*rough*	*p600*	*p240*
smooth	67	0	4	3
rough	11	52	2	10
p600	22	9	8	36
p240	11	4	5	56

rate, which depends on frequency) suffer in setting ii and iii due to the problems of short dabbing distances and variable whisk radius discussed above.

3.6 Combining classifiers

It is often possible to construct a more powerful classifier by combining features. Table 6 shows results using a 6-dimensional Gaussian classifier, operating on all three of our feature pairs (onset, offset, centroid, energy, SAL and RAS) over all settings, as in the previous section.

It can be seen that the combined classifier gives additional power in 4-way classification over all settings, and beats the neural neural network model of section 3.1. We performed a similar analysis for the case of 2-way (rough/smooth) classification, shown in table 7. In the two-way case, we found that the combined classifier

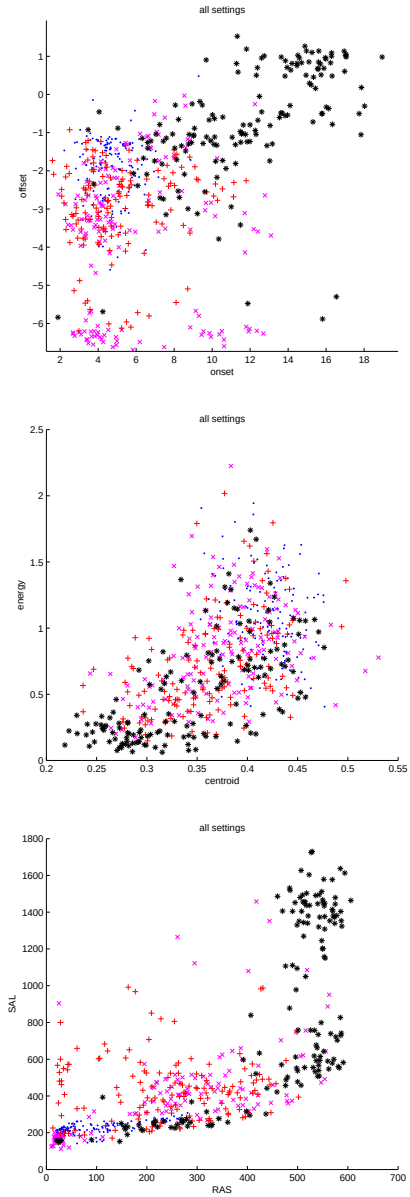


Fig. 11: Samples from datasets (i-iii) combined analysed using (a) onset-offset features (b) frequency features (c) RA/SA features as before

Table 6: Confusion matrix for the 6-dimensional, combined feature classifier, running on all settings together. The mean and standard deviation of the posterior percentage correct classification is $72 \pm 3\%$.

ALL	*smooth*	*rough*	*p600*	*p240*
smooth	67	3	3	1
rough	5	55	11	4
p600	0	32	37	6
p240	1	4	14	57

Table 7: Confusion matrices for the 6-dimensional combined feature classifier; for the three feature pairs; and for single onset and offset features; each over all settings together. Means and standard deviations of the posterior percentage correct classification are: $COMBINED = 91 \pm 2\%, OO = 92 \pm 2\%, CE = 81 \pm 4\%, PA = 91 \pm 2\%, onset = 91 \pm 2\%, offset = 78 \pm 4\%$.

$COMBINED$	$smooth$	$rough$
smooth	67	7
rough	6	69

OO	$smooth$	$rough$
smooth	69	5
rough	7	68

CE	$smooth$	$rough$
smooth	8	6
rough	22	53

PA	$smooth$	$rough$
smooth	74	0
rough	12	63

$onset$	$smooth$	$rough$
smooth	69	5
rough	8	67

$offset$	$smooth$	$rough$
smooth	68	6
rough	26	49

has no significant advantage over the onset-offset classifier or the PA classifier alone. Furthermore, training one-dimensional Gaussian models on the individual onset and onset features showed that the onset classifier alone achieves similar power to the combined classifier. This is of interest because the onset feature may be computed rapidly from initial contact data only, without waiting for the whisk to complete.

3.7 An online classifier

Having shown that the onset feature alone performs as well as the combination of all features in rough-smooth classification over all settings, we stress-tested this feature in the highly challenging task of classifying the 500 samples collected in setting (iv) from random whisking positions and orientations using the smooth and rough surfaces. Classification of this data was performed in real-time, requiring the use of fast cumulative update forms of the mean and (unbiased) standard deviation calculations of the T most recent strains x passed to the onset classifier:

$$\mu_t := \mu_{t-1} + \frac{1}{T}(x_t - x_{t-T-1}) \tag{15}$$

$$a_t := a_{t-1} + \frac{1}{T-1}(x_t^2 - x_{t-T-1}^2) \tag{16}$$

$$\sigma_t^2 := a_t - \frac{T}{T-1}\mu_t^2 \tag{17}$$

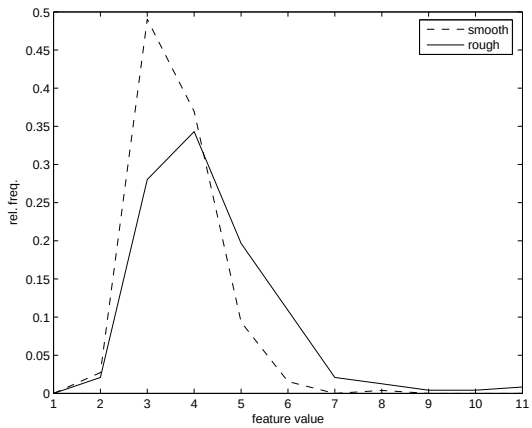


Fig. 12: Histogrammed values of a single discriminator function, from 500 whisks at random hand-held orientations and distances.

These update equations use internal state variables to avoid the need for the large sums $\mu_t := \sum_{i=t-T}^{t} x_i$ and $\sigma_t^2 := \sum_{i=t-T}^{t}(x_i - \mu)^2$ to be recomputed from scratch on the arrival of each new data point. As in the offline studies, it is necessary to obtain mean and standard deviations for every possible window location, as they are needed by the onset detector, in order to detect the *location* of the onset before classification proper begins. Fig. 12 shows the empirical distribution of the onset features in the rough and smooth classes.

The distributions are significantly different (ANOVA shows the means are different ($F(1, 495) = 53.096, p < 0.001$, partial eta squared=0.097) and Levene's test shows the variances are also different $(1, 494) = 51.466, p < 0.001$). We can compute the expected strength of belief in the correct class c after n whisks by

$$\langle p(c|d_{1:n})\rangle_{c,d} = \langle \frac{\pi(c)\prod_{i=1}^{N} p(d_i|c)}{\sum_c \pi(c)\prod_{i=1}^{N} p(d_i|c)}\rangle_{c,d} \tag{18}$$

$$= \sum_c \int_d \frac{\pi(c)^2 p(d_i|c)^{2N}}{\sum_c \pi(c) p(d_i|c)^N} \tag{19}$$

Assuming that the data were generated by the maximum likelihood parameters of two class-conditional Gaussians, and a flat prior on the classes; and computing the integral numerically with Gaussian Monte Carlo samples, we find the probability of a Gaussian classifier giving the correct class after a single whisk is 0.562, and after n whisks as shown in fig. 13. Such a classifier can be seen to achieve an expected 90% belief in the correct class after 9 whisks, 95% strength after 12 whisks, and 97.5% after 15 whisks.

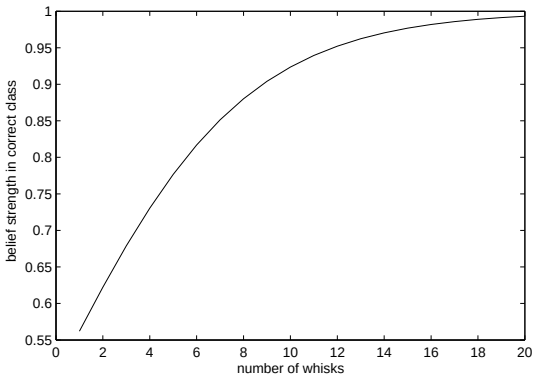


Fig. 13: Expected correct belief strength after n whisks in the random position task, using the onset feature only.

4 Discussion

This is the first work to present quantitative results in mobile whiskered texture discrimination, and the second (after [12]) to consider cross-setting discrimination. Previous studies with artificial whiskers produced good classifiers using low-level DFT features with generic parametric classifiers (e.g. 85% accuracy across 7 classes [19]) but under constrained conditions in which the interaction of the whisker with the surface was under tight experimental control. To overcome this limitation we tested several higher-level candidate features within and across three different experimental settings. Our principal findings can be summarised as follows. All candidate features tested were found to be useful within settings, i.e. where the interaction of the whisker with the surface was highly constrained. However, the frequency-based classifiers did not generalize particularly well across settings. The two best performing classifiers (see Table 7) were those based on the primary afferent model (PA) and on the onset feature. Combining all of the features produced a small accuracy improvement over the neural network replicated from [12], over all settings. The onset feature has the advantage that it can be computed very quickly following the initial contact with the surface, and gives a similar accuracy to the combined classifier in the two-class (rough,smooth) case across the settings.

The onset feature was further found to be useful for two-class (rough,smooth) discrimination across a set of highly randomized positions, requiring around 7 whisks to achieve 85% correct beliefs and 10 whisks to achieve 95% correct beliefs. In comparison, animal experiments [10] found a 85% success rate for trained rats making similar discriminations, typically making 1-5 whisks with multiple whisker contacts per whisk (46% of contacts were made with a single whisker; 36% with 2 whiskers; 18% with 3 whiskers) before the decision. For a *single* whisk (typically having two whisker contacts) the rats achieved 73% accuracy; in contrast our classifier requires three contacts to reach a similar accuracy, and reaches about 65% accuracy with only two contacts. So in this task our classifier is comparable to the rat, though the success rate is marginally lower.

We conjecture that the other features could be of more use if position and angle relative to the surface were known a priori, as we have seen that once the setting is known, good classifiers can be built using them. It is further possible that by introducing latent variables such as distance and angle from the surface, a good classifier may be found in the resulting higher-dimensioned space, generalizing over all possible settings. Frequency features in particular should be expected to be dependent on the radius and angle of contact with the surface, as physical frequencies in the texture will give rise to different whisker vibrations depending on this positioning. Methods for estimating the contact location are known (see below) and could in future be used to provide such prior information to the texture classifiers. Alternatively, the position information could be used to affect the agent's movements in order to bring the object to some standard position in the whisker field, then a simpler classifier, independent of position information, could be used. However for rapid decision-making, the onset feature provides a simple and effective way to make crude rough/smooth judgments before these more advanced inferences take place.

Which of the two above positional strategies – higher-dimensional classification and active positioning – are used by whiskered animals such as the rat? An answer to this question could inform the choice of strategy for robot platforms. There is some evidence in support of each strategy.

First, using biomechanical modeling and data obtained using plucked rat whiskers, Birdwell et al. [6] have shown that the radial distance to a point of contact can, in principle, be calculated by measuring the rate of change of moment (or equivalently curvature) at the whisker base. Further, this method has been shown to be useful for recovering distance information in an artificial vibrissal sensing system using steel-wire whiskers, gauged for strain, and fitted to a columnar whisking apparatus exploring a complex three-dimensional shape [36].

Second, behavioral data obtained by Carvell and Simons [10], in animals that were trained to discriminate texture for reward using only their macrovibrissae, provides evidence of an "active touch" strategy [25] that rats may use to optimise their texture discrimination capabilities. Specifically, they observed that while some

of the longer, more caudal, macrovibrissae were whisked back and forth across the target surface, presumably to obtain texture-related information, a number of the smaller rostralmost hairs remained in a "relatively motionless, protracted state [...] never losing contact with the surface" (p. 2641). It seems possible, then, that this use of the smaller whiskers could provide the rat with reference signals that might allow it measure, or control for, variance in the position of the larger whiskers relative to the target.

Finally, it is worth mentioning that, when allowed to do so, animals will often investigate textures with their shorter microvibrissae and other tactile sensory surfaces around the mouth [15]. In these circumstances the longer macrovibrissae may serve an important role in object detection and coarse feature discrimination, whilst also acting as a trigger for orienting movements that bring surfaces of interest into the 'foveal zone' of the shorter and more densely arranged microvibrissae (see [15] and figure 14). Contacts with this microvibrissal array, in addition to being more numerous, should be at distances and orientations that are both less variable and more predictable, thus signal features such as those considered in this article could be used – perhaps via a continuum of slow to rapid adapting afferents – to perform detailed texture classification. Brecht et al. [9] have similarly proposed that the microvibrissae may be the rats preferred tactile apparatus for making judgements regarding fine spatial detail, whilst Hartmann [15] has described rat exploration strategies in which micro- and macrovibrissal placements are either synchronized or co-ordinated in predictable sequences. Further experiments with a range of artificial whisker sensors are needed to determine whether some similar mix of short and long sensing elements, with the latter supporting a detecting/orienting capacity, might also be useful in autonomous whiskered robots.

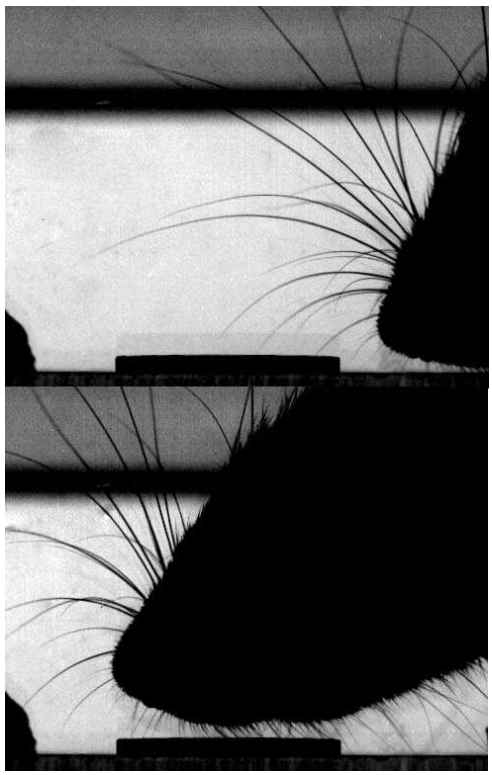


Fig. 14: Frames from a high-speed video recording (full recording in supplementary video 4) showing a rat detecting a complex textured surface of interest (a small coin) with its macrovibrissae (upper frame), and then, in a subsequent whisk cycle, investigating that surface with its microvibrissae on the chin and upper lip (lower frame). Many similar high-speed recordings obtained in our laboratory indicate that the macrovibrissae are frequently used in this manner to detect, and then orient the microvibrissal array towards, surfaces of interest.

5 Acknowledgements

The authors would like to thank members of the Active Touch Laboratory at Sheffield and the Bristol Robotics Laboratory for discussions, particularly Chris Melhuish and Robyn Grant. This work was supported by EU grants ICEA (IST-027819) and BIOTACT (ICT-215910).

References

1. M. L. Andermann and C. I. Moore. A somatotopic map of vibrissa motion direction within a barrel column. *Nat Neurosci*, 9(4):543–51, 2006.
2. E. Arabzadeh, S. Panzeri, and M. E. Diamond. Whisker vibration information carried by rat barrel cortex neurons. *J Neurosci*, 24(26):6011–20, 2004.
3. E. Arabzadeh, E. Zorzin, and M. E. Diamond. Neuronal encoding of texture in the whisker sensory pathway. *PLoS Biology*, 3(1):e17, 2005.
4. R. W. Berg and D. Kleinfeld. Rhythmic whisking by rat: retraction as well as protraction of the vibrissae is under active muscular control. *J Neurophysiol*, 89(1):104–17, 2003.
5. J. M. Bernardo and Adrian Smith. *Bayesian Theory (Wiley Series in Probability and Statistics)*. John Wiley and Sons Ltd, March 2000.
6. J.A. Birdwell, J.H. Solomon, M Thajchayapong, M.A. Taylor, M Cheely, R.B. Towal, J Conradt, and M.J. Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *Journal of Neurophysiology*, 98(4):2439–55, October 2007.
7. C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
8. M. Brecht. Barrel cortex and whisker-mediated behaviors. *Curr Opin Neurobiol*, 17(4):408–16, 2007.
9. M. Brecht, B. Preilowski, and M. M. Merzenich. Functional architecture of the mystacial vibrissae. *Behavioural Brain Research*, 84(1-2):81–97, 1997.
10. G. E. Carvell and D. J. Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *Journal of Neuroscience*, 10(8):2638–2648, 1990.

11. J. Stephen Downie, Kris West, Andreas Ehmann, and Emmanuel Vincent. The 2005 music information retrieval evaluation exchange. In *Proceedings of the 6th International Symposium on Music Information Retrieval*, pages 320–323, 2005.
12. M. Fend. Whisker-based texture discrimination on a mobile robot. In *Proceedings of the 8th European Conference on Artificial Life (ECAL)*, Springer Lecture Notes in Computer Science, 2005.
13. M. Fend, S. Bovet, H. Yokoi, and R. Pfeifer. An active artificial whisker array for texture discrimination. In *IEEE/RSJ Conf on Intelligent Robots and Systems (IROS)*, Las Vegas, 2003.
14. B. T. Fundin, F. L. Rice, K. Pfaller, and J. Arvidsson. The innervation of the mystacial pad in the adult-rat studied by anterograde transport of hrp conjugates. *Experimental Brain Research*, 99(2):233–246, 1994.
15. M. J. Hartmann. Active sensing capabilities of the rat whisker system. *Autonomous Robots*, 11:249–254, 2001.
16. J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, and P. Konig. Texture signals in whisker vibrations. *J Neurophysiol*, 95(3):1792–9, 2006.
17. J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M.E. Diamond, and P. Konig. Texture signals in whisker vibrations. *Journal of Neurophysiology*, doi:10.1152/jn.01104.200, 2005.
18. Makoto Kaneko. Active antenna for contact sensing. *IEEE Transactions on Robotics and Automation*, 14(2):278–291, April 1998.
19. D. Kim and R. Möller. A biomimetic whisker for texture discrimination and distance estimation. In S. Schaal, editor, *From Animals to Animats 8 (Proc. 8th Intl. Conf. on the Simulation of Adaptive Behavior)*, number 8, pages 140–149. MIT Press, 2004.
20. D. Kim and R. Möller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55(3):229–243, 2007.
21. D. Kleinfeld, E. Ahissar, and M. E. Diamond. Active sensation: insights from the rodent vibrissa sensorimotor system. *Curr Opin Neurobiol*, 16(4):435–44, 2006.
22. M. Lungarella, V. Haffner, R. Pfeifer, and H. Yokoi. An artificial whisker sensor for robotics. In *15th Conference on Intellient Robots and Systems*, pages 2931–36, 2002.
23. B. Mitchinson, K. N. Gurney, P. Redgrave, C. Melhuish, A. G. Pipe, M. Pearson, I. Gilhespy, and T. J. Prescott. Empirically inspired simulated electro-mechanical model of the rat mystacial follicle-sinus complex. *Proc Biol Sci*, 271(1556):2509–16, 2004.
24. B. Mitchinson, C. J. Martin, R. A. Grant, and T. J. Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–41, 2007.
25. B. Mitchinson, C.J. Martin, R.A. Grant, and T.J. Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc. Biol. Sci.*, 274:1035–1041, Apr 2007.
26. B. Mitchinson, M. Pearson, C. Melhuish, and T. J. Prescott. A model of sensorimotor coordination in the rat whisker system. In *From animals to animats 9: Proceedings of the Ninth International Conference on Simulation of Adaptive Behaviour, LNAI*, volume 4095. Springer Verlag, 2006.
27. Ben Mitchinson, Ehsan Arabzadeh, Mathew E. Diamond, and Tony J. Prescott. Spike-timing in primary sensory neurons: a model of somatosensory transduction in the rat. *Biological Cybernetics*, Online, January 2008.
28. Martin J. Pearson, Ian Gilhespy, Chris Melhuish, Ben Mitchinson, Mokhtar Nibouche, Anthony G. Pipe, and Tony J. Prescott. A biomimetic haptic sensor. *International Journal of Advanced Robotic Systems*, 2(4):335–343, 2005.
29. Martin J. Pearson, Anthony G. Pipe, Chris Melhuish, Ben Mitchinson, and Tony J. Prescott. Whiskerbot: A robotic active touch system modelled on the rat whisker sensory system. *Adaptive Behavior*, 15(3):223–240, 2007.
30. J. Ritt, M. Andermann, Skowronski-Lutz, E., and C. Moore. Characterization of vibrissa motion during volitional active touch. In *Barrels XIX*, Atlanta, USA, 2006.
31. R. Andrew Russell. Closing the sensor-computer-robot control loop. *Robotics Age*, 6(4):15–20, April 1984.
32. R. Andrew Russell. Using tactile whiskers to measure surface contours. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 1295–1299, 1992.
33. A.E. Schultz, J.H. Solomon, M.A. Peshkin, and M.J. Hartmann. Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, pages 2588 – 2593, 2005.
34. Thomass Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, April 2007.
35. A. Seth, J. McKinstry, G. Edelman, and J. Krichmar. Texture discrimination by an autonomous mobile brain-based device with whiskers. In *Proceedings of the International IEEE Conference on Robotics and Automation*, volume 5, pages 4925–30, May 2004.
36. J. H. Solomon and M. J. Hartmann. Biomechanics: robotic whiskers used to sense features. *Nature*, 443(7111):525, 2006.
37. J.F. Wilson and A. Chen. A whisker probe system for shape perception of solids. *ASME Journal of Dynamic Systems, Measurement and Control*, 117:104–108, 1995.

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/23500680

Active Touch Sensing in the Rat: Anticipatory and Regulatory Control of Whisker Movements During Surface Exploration

Article *in* Journal of Neurophysiology · December 2008

DOI: 10.1152/jn.90783.2008 · Source: PubMed

CITATIONS
101

READS
41

4 authors, including:

Robyn Grant
Manchester Metropolitan University
23 PUBLICATIONS **435** CITATIONS

SEE PROFILE

Ben Mitchinson
The University of Sheffield
57 PUBLICATIONS **981** CITATIONS

SEE PROFILE

All content following this page was uploaded by Robyn Grant on 29 November 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Active Touch Sensing in the Rat: Anticipatory and Regulatory Control of Whisker Movements During Surface Exploration

Robyn A. Grant, Ben Mitchinson, Charles W. Fox and Tony J. Prescott

J Neurophysiol 101:862-874, 2009. First published 26 November 2008;
doi: 10.1152/jn.90783.2008

You might find this additional info useful...

Supplementary material for this article can be found at:
http://jn.physiology.org/http://jn.physiology.org/content/suppl/2008/12/08/90783.2008.DC1.html

This article cites 33 articles, 13 of which you can access for free at:
http://jn.physiology.org/content/101/2/862.full#ref-list-1

This article has been cited by 14 other HighWire-hosted articles:
http://jn.physiology.org/content/101/2/862#cited-by

Updated information and services including high resolution figures, can be found at:
http://jn.physiology.org/content/101/2/862.full

Additional material and information about ***Journal of Neurophysiology*** can be found at:
http://www.the-aps.org/publications/jn

This information is current as of June 13, 2013.

Journal of Neurophysiology publishes original articles on the function of the nervous system. It is published 12 times a year (monthly) by the American Physiological Society, 9650 Rockville Pike, Bethesda MD 20814-3991. Copyright © 2009 the American Physiological Society. ISSN: 0022-3077, ESSN: 1522-1598. Visit our website at http://www.the-aps.org/.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

J Neurophysiol 101: 862–874, 2009.
First published November 26, 2008; doi:10.1152/jn.90783.2008.

Active Touch Sensing in the Rat: Anticipatory and Regulatory Control of Whisker Movements During Surface Exploration

Robyn A. Grant, Ben Mitchinson, Charles W. Fox, and Tony J. Prescott
Department of Psychology, University of Sheffield, Sheffield, United Kingdom

Submitted 18 July 2008; accepted in final form 24 November 2008

Grant RA, Mitchinson B, Fox CW, Prescott TJ. Active touch sensing in the rat: anticipatory and regulatory control of whisker movements during surface exploration. *J Neurophysiol* 101: 862–874, 2009. First published November 26, 2008; doi:10.1152/jn.90783.2008. Animals actively regulate the position and movement of their sensory systems to boost the quality and quantity of the sensory information they obtain. The rat vibrissal system is recognized to be an important model system in which to investigate such "active sensing" capabilities. The current study used high-speed video analysis to investigate whisker movements in untrained, freely moving rats encountering unexpected, vertical surfaces. A prominent feature of rat vibrissal movement is the repeated posterior–anterior sweep of the whiskers in which the macrovibrissae are seen to move largely in synchrony. Here we show that a second significant component of whisking behavior is the size of the arc, or "spread," between the whiskers. Observed spread is shown to vary over the whisk cycle and to substantially decrease during exploration of an unexpected surface. We further show that the timing of whisker movements is affected by surface contact such that *1*) the whiskers rapidly cease forward protraction following an initial, unexpected contact, and may do so even more rapidly following contact with the same surface in the subsequent whisk cycle, and *2*) retraction velocity is reduced following this latter contact, leading to longer second-contact durations. This evidence is taken to support two hypotheses: *1*) that the relative velocities of different whiskers may be actively controlled by the rat and *2*) that control of whisker velocity and timing may serve to increase the number and duration of whisker–surface contacts while ensuring that such contacts are made with a light touch.

INTRODUCTION

Adult rats sweep their large facial whiskers (macrovibrissae) back and forth in a rhythmic behavior known as whisking, observed under most conditions of natural locomotion and exploration (Carvell and Simons 1990; Gustafson and Felbain-Keramidas 1977; Hartmann 2001; Vincent 1912). Recent research has demonstrated that exploratory whisking, in freely moving animals, often diverges from the regular, largely bilaterally symmetric and synchronous motor pattern that has been recorded when immobilized rats are trained to whisk in air (e.g., Bermejo et al. 2002; Gao et al. 2001). Specifically, asymmetries, asynchronies, and changes in whisk amplitude and frequency have been documented (Berg and Kleinfeld 2003; Carvell and Simons 1990; Mitchinson et al. 2007; Sachdev et al. 2003; Sellien et al. 2005; Towal and Hartmann 2006, 2008) that may be the consequence of "active sensing" control strategies. Such strategies use sensor information, or task knowledge, to control the position and movement of the sensory apparatus in a manner likely to boost the amount of useful sensory information obtained (Aloimonos et al. 1988; Ballard 1991; Gibson 1962; Lungarella et al. 2005). In the current study we set out to discover whether other aspects of whisker control might also be subject to this kind of active regulation and to identify some of the likely sensory consequences of these forms of anticipatory control.

Our specific focus concerns how the rat modifies the movements of its whiskers, and of its head and body, over the course of a sequence of whisk cycles during which contact is made with an unexpected vertical surface. We have previously demonstrated immediate, contact-induced changes in the timing of whisker control and longer-term, anticipatory modulations of whisk amplitude (Mitchinson et al. 2007). Here we will present evidence of changes in a further whisking parameter: the relative spacing, or "spread," between adjacent whiskers. We will provide data suggesting that this measure not only captures a significant, and previously unquantified, fraction of variance in whisking behavior, but that it also appears to be modified by active control mechanisms following contact and in anticipation of further contacts.

During exploratory whisking, the initial contact with a vertical surface triggers a rapid cessation of whisker protraction (and commencement of retraction) such that the whiskers palpate the surface with a relatively light touch (Mitchinson et al. 2007). We have termed this active control strategy "minimal impingement," since whisker movements appear to be modulated to reduce whisker bending (impingement) against the contacted surface. The investigation of such whisking control strategies is important to understanding the nature of the tactile stimuli that are processed in upstream neural centers, such as the somatosensory cortex, during natural behavior. For instance, evidence for "minimal impingement" strategies suggests that the whisker stimuli that are transmitted to these structures may be less intense than might otherwise be expected, whereas "maximal contact" strategies that involve regulating whisk amplitude or spread, could lead us to expect greater numbers of whisker–surface contacts than if control is not subject to such modulation. A further important parameter for understanding upstream sensory processing is the duration of whisker–surface contacts: while the whiskers are in contact with the surface the animal may be able to extract useful information about surface shape or texture. Our investigation will therefore also consider how differential control of whisker movements during interactions with surfaces could influence

Address for reprint requests and other correspondence: T. J. Prescott, The University of Sheffield, Department of Psychology, Western Bank, Sheffield S10 2TN, UK (E-mail: t.j.prescott@sheffield.ac.uk).

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

 0022-3077/09 $8.00 Copyright © 2009 The American Physiological Society

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

these important determinants of the tactile signals processed by the rat brain.

METHODS

Experimental procedures

ANIMALS. Ten male Royal College of Surgeons (RCS) dystrophic rats, aged 6–18 mo and weighing 250–350 g, were used. All animals had genetic retinal degeneration (dystrophy) and thus minimal vision; they were therefore strongly reliant on tactile information from their whiskers during locomotion and exploration behavior. Hetherington et al. (2000) previously showed that the capacity to orient to whisker stimulation is unimpaired in dystrophic RCS animals compared with other rat strains, although they did note a deficit in orienting to a tactile stimulus on the flank. Observations in our laboratory of dystrophic and nondystrophic RCS animals and of sighted Hooded Lister rats suggest that whisking control in dystrophic animals does not deviate in any marked way from that of normally sighted rats. All animals were kept in a 12-h dark/light cycle at 22°C, with unrestricted access to water and food; tested during the dark (active) part of their daily cycle; and handled prior to being placed in the experimental arena. All procedures were approved by the local Ethics Committee and UK Home Office, under the terms of the UK Animals (Scientific Procedures) Act of 1986.

APPARATUS. Digital video recordings were made using a Photron Fastcam PCI camera, recording at 500 frames/s, shutter speed of 0.5 ms, and resolution of 1,024 × 1,024. The camera was suspended from the ceiling above a custom-built rectangular (40 × 40 cm) viewing arena with a glass floor, ceiling, and end-wall (see Supplemental Fig. S1).[1] The arena contained a front-silvered mirror, positioned behind the end-wall, to afford two viewpoints to the camera. The camera was also positioned so that it looked directly down the end and side walls in the overhead view; the front-silvered mirror was angled at about 45° so that the camera looked along the surface of the floor in the reflected, end-on view. The arena was illuminated from below by a custom-built, high-power light box, and from the far end of the arena facing the end-wall by a second, suitably angled mirror. The design and geometry of the arena thus provided uniform backlighting in two dimensions, with the field of view of the camera covering close to 20 × 20 cm in the overhead view and 8 × 20 cm in the end-on view. To obtain an accurate measure of camera/arena geometry a recording was also made of a three-dimensional (3D) calibration tool, with known shape and location, following initial positioning of the camera.

RECORDING. Opportunistic recordings, each 3–4 s in length, were taken of awake, unrestrained animals engaged in active, exploratory behavior. Each animal was filmed on between 2 and 11 occasions with ≤12 clips recorded, per rat, in each filming session. Each recording was initiated by an experimenter viewing the camera scene on a monitor window and pressing a trigger when the animal entered the field of view. In all, 334 recordings were obtained in this way with between 15 and 82 per animal.

Data selection

Video clips were selected for whisker tracking and detailed analysis as follows. To qualify for analysis each video was required to contain three consecutive whisks during which the rat approached and contacted an identified arena wall, with whiskers on the side of the snout ipsilateral to the initial contact clearly visible in the overhead view throughout. Furthermore, these three whisks were required to be of the following types:

1) An initial *precontact whisk*, in which the whiskers did not contact the identified wall

2) A *first-contact whisk*, in which at least two whiskers made contact with this vertical surface

3) A *second-contact whisk*, in which the same two whiskers, and possibly others, made a further contact with the same surface

The constraint that there should be contacts on at least two whiskers in each cycle was enforced to reduce the possibility of including clips in which surface contacts were so slight that they might be below the rat's detection thresholds (Stuttgen et al. 2006). The requirement that the same two whiskers made the initial contact with the wall, in both the first- and second-contact whisks, was enforced to exclude some of the more atypical interactions with surfaces. In all, 60 clips (between 1 and 15 per animal) were identified that satisfied these criteria.

We note that the clips selected for analysis do not, in general, record the animal's first experience of the experimental arena. However, in sightless animals, anticipation of a surface prior to whisker contact would require the synthesis of path integration information with acquired knowledge of the arena configuration. Due to the accumulation of error in neuronal path integration mechanisms and evidence that such mechanisms may be specialized toward encoding position with respect to a home base (Etienne et al. 1996), it is unlikely that the RCS rat will have access to an accurate estimate of the relative position of a wall following a period of roaming, during which there was no tactile contact with that surface. In the clips we analyzed, then, the initial contact with the wall, in the first-contact whisk, was considered to be "unexpected" in the sense that the animal is assumed to lack accurate prior knowledge of its own position relative to that surface prior to contact. More generally, our results do not require that the rat has no prior knowledge of nearby surfaces, but rather that the initial whisker contact provides information that greatly improves the animal's capacity to localize a nearby surface and is therefore useful for directing subsequent exploratory behavior.

Whisker tracking and head tracking

For the 60 selected clips, and for each of the three selected whisks, five whiskers, ipsilateral to the initial contact, were tracked on an LCD flat-screen monitor by a human observer using uncompressed video and a purpose-built tracking tool. Whisker tracking, in this case, used only the overhead view and involved tracking two points, one near the base of the whisker, the other two thirds of the way out along the whisker shaft (see Supplemental Fig. S1). This is referred to as the overhead tracking set in the following text. The tracked whiskers were assumed to correspond to whisker columns 0–4, although our results do not depend on this assumption. To estimate head movements, the tip of the snout and the midpoint of the head were also tracked in the overhead view and the height of the snout above the floor in the end-on view. Note that, although technologies have recently been developed for automated tracking of whisker movement (Knutsen et al. 2005, 2008; Voigts et al. 2008), the difficult problem of tracking multiple whiskers in a complete, intact whisker field has yet to be automated. No smoothing was performed on the tracked whisker angular position or head-movement data; for whisker velocity a moving average filter that computes a running average of three adjacent points was used to reduce any effects of tracking inaccuracies.

The overhead tracking data for each clip were analyzed to obtain the descriptive measures listed in Table 1. For the calculation of per-whisk summary measures (Table 1, *B* and *D*) the point of minimum protraction between two successive protraction–retraction phases was taken as the separation boundary between consecutive whisks. The clips were also examined to determine whether contacts with the target surface, in the first-contact whisk, occurred on one side or both sides of the snout. Two subsets were identified for further analysis: *1)* the unilateral set ($n = 25$) in which contact, during the

[1] The online version of this article contains supplemental data.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

TABLE 1. *Definitions of whisking, contact-related, and head movement measures*

Name	Unit	Description
A. Whisker movement measures per video frame (overhead view)		
Angular position	degrees	Angle from the whisker shaft to the head midline, such that forward movement (protraction) causes an increase in angle
Mean angular position	degrees	Mean angular position across all five tracked whiskers
Velocity	deg/ms	Rate of change of the mean angular position
Spread	degrees	Difference between the largest and smallest instantaneous angular positions of the five tracked whiskers
B. Summary whisker movement measures per whisk (or per specified portion of a whisk)		
Minimum and maximum protraction	degrees	Minimum and maximum values of the mean angular position. Minimum protraction is also sometimes referred to as peak retraction or the whisking "set point"
Mean, minimum, and maximum spread	degrees	Summary measures of spread
Mean protraction and retraction velocities	deg/ms	Summary measures of velocity calculated separately for the protraction and retraction phases of the whisk
Whisk duration	ms	Time from the previous minimum protraction to the next
C. Measures relating to the initial contact with the wall within a whisk		
Number of contacts on tracked whiskers	2–5	Total number of tracked whiskers making contact with the wall over the course of the whisk
Angular velocity at contact	deg/ms	Instantaneous angular velocity of the whisker making first contact calculated from the two frames preceding the contact
Angular position at contact	degrees	Mean angular position in the frame immediately prior to the first surface contact
Contact duration	ms	Time from initial contact to the first contacting whisker becoming detached from the surface (measured for the most rostral whisker if two whiskers make simultaneous first contact)
Time from contact to maximum protraction	ms	Time from the initial contact to maximum protraction averaged across all five tracked whiskers
D. Head position measures per whisk		
Snout elevation	mm	Mean distance from the snout tip to the floor (vertical view) over the course of the whisk
Snout vertical velocity	mm/ms	Absolute value of the rate of change in snout elevation
Head orientation	degrees	Mean angle of the head relative to the wall (overhead view), with zero being perpendicular and all angles positive, over the course of the whisk
Head angular velocity	deg/ms	Absolute value of the rate of change in head orientation
Distance to wall	mm	Mean length of the perpendicular from the snout tip to the wall (overhead view) over the course of the whisk
Velocity toward wall	mm/ms	Rate of change in distance to wall
Velocity along wall	mm/ms	Rate of change of head position parallel to the wall and floor (overhead view)

first-contact whisk, was on one side only, and *2*) the bilateral set (n = 35) in which there were whisker–surface contacts on both sides of the snout in the first-contact whisk. Further details concerning our tracking and data analysis methods are provided in the Supplement, including raw data (whisker trajectories) for two exemplar clips (Supplemental Fig. S4).

3D whisker-tip trajectory reconstruction

To better evaluate the contribution of head movements to the patterns of whisker movement observed in the overhead view, four representative clips were selected for 3D reconstruction of whisker-tip trajectories. Tracking and trajectory reconstruction, using a stereo correspondence algorithm, were performed as described in the Supplemental Material.

Statistical considerations

The primary focus of this investigation was on contrasts within triplets of consecutive whisks; data were therefore pooled across animals with the video clip of each tracked whisking episode taken as the fundamental unit for analysis. Except where otherwise specified the results presented were calculated for the 60 selected clips in the overhead tracking data set. MANOVA was used to test for overall differences between whisk types in relation to each of the main classes of data, with univariate ANOVAs and appropriate post hoc tests used to examine differences on specific variables. An alpha-level of 0.05 (two-tailed) was used for statistical tests corrected for multiple comparisons, when required, using a Bonferroni correction. Significant results are summarized in the following text and figures, with SE bars included in graphs where appropriate. Further details of statistical analyses, including tables of all univariate ANOVA tests, including means, SDs, F-values, and effect sizes, are provided in the Supplemental Material.

RESULTS

Our presentation of results is structured as follows. In the next section we provide a principal components analysis (PCA) of our whisker position data and show that the two main components of these data, determined algorithmically, correspond with the theoretically meaningful measures of the mean angular position of the whiskers and whisker spread. Next, in *Variation across whisk types in summary measures of whisker control, whisker–surface contacts, and head position*, we quantitatively characterize the whisking behavior of the rat during exploration of an unexpected surface by analyzing variation in summary measures of whisk control (Table 1*B*), surface contact (Table 1*C*), and head movement (Table 1*D*). Armed with a good understanding of the differences between the three whisk types, *A closer look at whisker spread* then considers whisker spread in more detail, evaluating several alternatives to the hypothesis that spread is controlled using differential whisker velocities and in anticipation of surface contacts. Finding that anticipatory control of whisker velocity is supported by the available data, *Sensory consequences of whisker*

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

control then examines whether such control is consistent with a "maximizing contact" strategy. We then turn our attention to the "minimal impingement" aspect of our active touch hypothesis and consider whether this is supported by our new data concerning the timing of cessation of protraction following contact. Finally, we look at the duration of whisker–surface contacts and explore the relationship of this important variable to several elements of whisker control, particularly the angular velocity of the whiskers during the retraction phase.

Principal components of whisker configurations

To better characterize the nature of rat whisking control a PCA analysis was run on the raw whisker angular position data from all five tracked whiskers and across all three specified whisk types (precontact, first contact, second contact). Three components were found to be present in this data set, as illustrated in Fig. 1.

- *Component 1* explained 80.4% (precontact whisk), 76.8% (first contact), and 70.1% (second contact) of the variance in whisker positions (Fig. 1, *top*) and was extremely well correlated ($r > 0.99$) with the mean angular position across all whisk types. Thus as might be expected since whisker movement is strongly synchronized, the average protraction angle of the whisker field strongly predicts the positions of all the tracked whiskers.
- *Component 2* explained 13.4% (precontact whisk), 15.5% (first contact), and 20.2% (second contact) of the variance in whisker positions (Fig. 1, *center*) and was well correlated with whisker spread (precontact: $r = 0.90$; first contact: $r = 0.91$; second contact: $r = 0.92$). Therefore the overall change in the angular separation of the whiskers constitutes the second most important parameter for describing the spatial arrangement of the whiskers at any time during the whisk.
- *Component 3* explained 3.1% (precontact whisk), 3.8% (first contact), and 4.6% (second contact) of the variance in whisker positions (Fig. 1, *bottom*), but was not significantly correlated with any of the measured whisking parameters. Component 3 is best described as the middle whisker columns (1–3), tending to be more protracted than the two outer columns (0 and 4). Although the proportion of variance explained is small, this component was present in every whisk type, suggesting it is unlikely to be an artifact of the measurement/analysis process (in contrast, components 4 and 5 did not have a consistent shape across whisk types and so are probably not meaningful).

In summary, then, although the whiskers largely move as a group (70–80% of variance captured by their mean angular position), there is important variation in relative whisker positions (13–20%) that is well characterized by the measure of whisker spread. Using these two components together can account for 90–94% of all the variance in the whisker position data seen in the overhead view. In other words, given just these two values for each frame we could reconstruct the actual angular positions of all five tracked whiskers to this level of accuracy.

Having established mean angular position and spread as the two most informative parameters for describing whisker con-

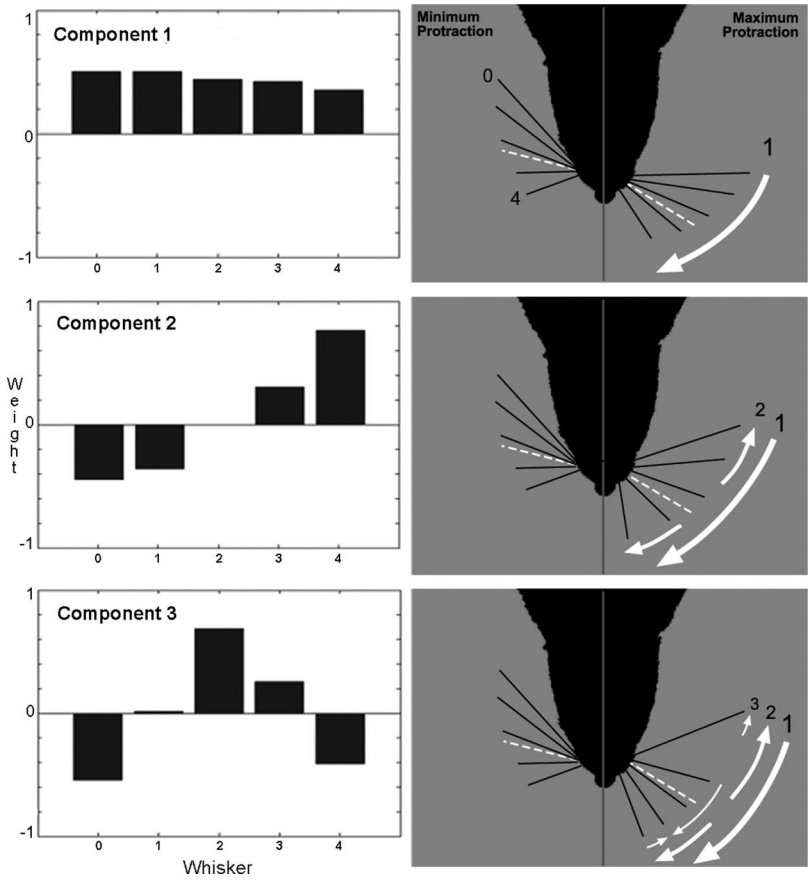


FIG. 1. The principal components of rat whisker configurations. The first 3 principal components account for 70–80% (component 1, *top*), 13–20% (component 2, *center*), and 3–5% (component 3, *bottom*) of the variance in whisker angles seen in the overhead view. Here the histograms on the *left* indicate the weighting on each tracked whisker (0: most caudal; 4: most rostral) for each component. The images on the *right* illustrate the components graphically and cumulatively (*top*: component 1; *center*: 1 + 2; *bottom*: 1 + 2 + 3). Each image shows the whiskers at minimum protraction on the left-hand side of the rat snout and at maximum protraction, as determined by the effects of the principal components (white arrows), on the right-hand side. Dotted white lines indicate the mean angular position. Note that the first 2 components are highly correlated with mean angular position and whisker spread, respectively, and together account for 89–93% of the variance in observed whisker positions.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

figurations it is useful to briefly consider the phase relationship of the two measures over the course of the whisk cycle. In Supplemental Fig. S5 we show, using cross-covariance analysis, that in the precontact and first-contact whisks, spread peaks reliably 0–6 ms before angular position. However, this relationship breaks down somewhat in the second-contact whisk, where we see a much weaker coupling between the two measures with out-of-phase relationships (>10-ms difference between the angular position and spread peaks) in 26 (43%) of the 60 whisks. The temporal relationship between spread and angular position seen in the precontact and first-contact whisks is consistent with observations by Sachdev et al. (2002), made in head-fixed animals whisking in air, who described the spread as being maximal at, or close to, maximum protraction. That this relationship is much less in evidence for the second-contact whisk is consistent with the hypothesis, explored in the following text, that active control mechanisms may be influencing whisker spread when the rat is able to anticipate contact with a vertical surface.

Variation across whisk types in summary measures of whisker control, whisker–surface contacts, and head position

VARIATION IN WHISKER CONTROL ACROSS WHISK TYPES. To begin our investigation of changes in whisk control following unexpected surface contacts a MANOVA was conducted using whisk type as a three-way within-subjects factor and the summary whisker movement measures (Table 1*B*) maximum protraction angle, minimum protraction angle, mean spread, mean protraction velocity, mean retraction velocity, and whisk duration as dependant variables. This analysis revealed a significant multivariate difference between whisk types ($P < 0.001$). To determine which aspects of whisking were contributing to this result, separate three-way univariate ANOVAs were then carried out on each of these six variables as detailed in Supplemental Table S1. All of the analyzed variables showed reductions between the precontact and second-contact whisk, with the first-contact whisk being, on the whole, more similar to the precontact whisk than to the second-contact whisk (as indicated by post hoc tests). Two parameters, in particular, stood out as showing large changes by the second-contact whisk—the mean spread and the mean retraction velocity. These measures fell by 21% ($P < 0.001$) and 22% ($P < 0.001$), respectively, over the course of the three whisks. Smaller but still significant changes also occurred in minimum and maximum protraction angles of the whisk (7 and 10%, $P < 0.001$). Figure 2 provides graphs illustrating each of the significant trends in whisk protraction angle, spread, and velocity. Examples of the reduction in overall spread are also provided in Supplemental Video S1 and in the sequence of snapshots in Fig. 3. Comparison of clips with bilateral and unilateral first contacts found a significant interaction effect between whisk type and whether the initial contact was on one side or both sides ($P < 0.001$; see Supplemental Fig. S6), indicating that the reduction in spread across whisks was more pronounced on occasions where there were bilateral surface contacts.

VARIATION IN WHISKER–SURFACE CONTACT ACROSS WHISK TYPES. We next examine the relationship between whisker movement and surface contact for the two contact whisks. Four variables that have potentially important sensory consequences—and therefore deserve attention—are the number of whisker contacts, the angular velocity and angular position of the whisker at the point of contact, and the duration of whisker–surface contact (see Table 1*C* for definitions). Further, since we previously proposed "minimal impingement" as a general characteristic of exploratory whisking (Mitchinson et al. 2007) we will also examine the time from contact to maximum protraction, predicting that control of whisker movement following contact will show rapid cessation of protraction in both the first- and second-contact whisks. Two limitations of the analyzed data should be made clear: First, our measure of the number of whisker contacts is the number of tracked whiskers touching the vertical surface during the whisk cycle of interest and not the total number of contacting whiskers (which is difficult to measure from video recordings). Second,

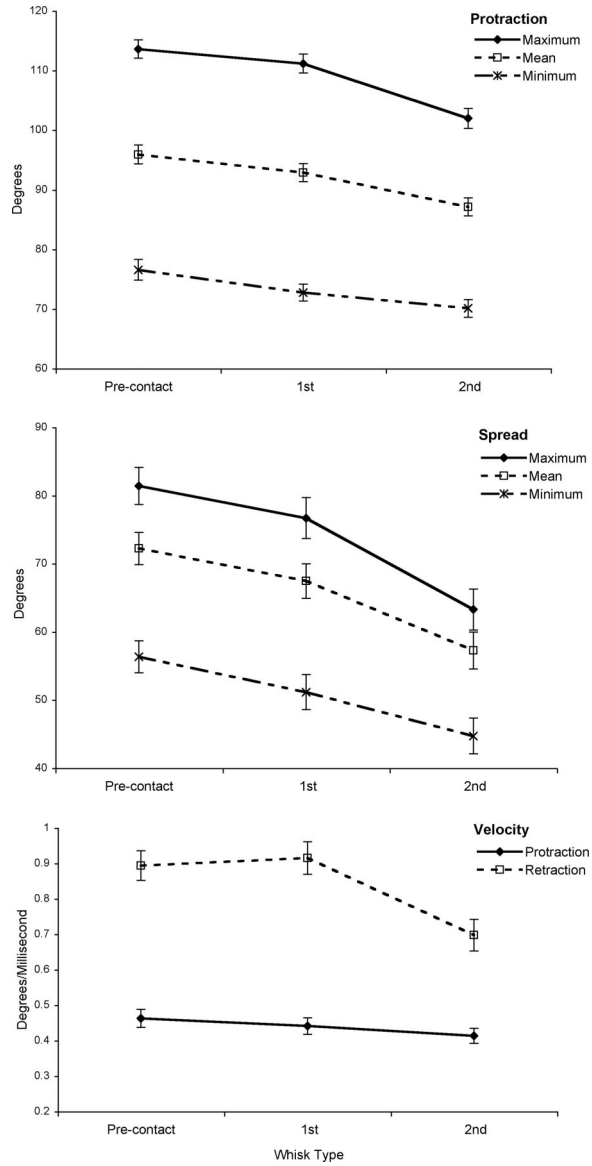


FIG. 2. Changes in whisk protraction, spread, and velocity on encountering an unexpected vertical surface. From *top* to *bottom*: maximum, minimum, and mean protraction (degrees); maximum, minimum, and mean spread (degrees); and mean protraction and retraction velocities for precontact, 1st-contact, and 2nd-contact whisks. All of these measures, except protraction velocity, are reduced significantly by the 2nd-contact whisk.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

FIG. 3. Snapshots of whiskers at maximum protraction for 3 consecutive whisks. From *bottom* to *top*: precontact, 1st contact, and 2nd contact. Whisker spread is significantly reduced by the 2nd-contact whisk. These snapshots are taken from the clip provided as Supplemental Video S1.

contact duration was calculated for only the first whisker to touch the surface, although this is also generally the last whisker to leave the surface. Furthermore, in 25 (42%) of the second-contact whisks contact continued beyond the end of the whisk cycle. Contact duration measures for these whisks were therefore calculated as the time from the initial contact, in that whisk, to the end of the whisk cycle.

A MANOVA for the preceding contact-related measures again showed a large difference between whisk types ($P < 0.001$). Subsequent univariate analyses (Supplemental Table S2) found that there were more contacts on tracked whiskers in the second-contact whisk (+14%, $P = 0.001$), that second contacts occurred at smaller angular positions (−9%, $P < 0.001$), and that the mean duration of contacts was markedly longer in second-contact whisks compared with first-contact whisks (+40%, $P < 0.001$). Mean time from contact to maximum protraction was found to be <15 ms in both whisks (14.3 and 11.3 ms for the first- and second-contact whisks, respectively), consistent with the "minimal impingement" hypothesis. The findings of more contacts, longer contact durations, and rapid cessation of protraction will each be subsequently examined further in *Sensory consequences of whisker control*. To conclude the current section, however, we briefly consider the impact of head movements on whisking control and on whisker–surface contacts.

VARIATION IN HEAD POSITION AND MOVEMENT ACROSS WHISK TYPES. Although our primary focus here is on changes in whisker movements it is important to consider whether there may be consistent patterns in the positioning and movement of the rat head during exploratory whisking. This matters both because head movement can influence the sensory consequences of whisking (movement toward a surface, for instance, will increase bending of any contacting whiskers) and because head/body movement is likely to be an important element of the overall control strategy used by the rat to position its whiskers. The head position and movement parameters listed in Table 1*D* were selected for quantitative analysis as being most relevant to understanding whisker movement and positioning with respect to a vertical surface. An initial MANOVA again showed a strongly significant difference between whisk types ($P < 0.001$), whereas subsequent univariate ANOVAs (Supplemental Table S3) showed that the mean distance to the wall and the mean velocity toward the wall both decreased by >50% ($P < 0.001$) by the second-contact whisk. Meanwhile, the vertical velocity of the snout (a useful indicator of the change in head tilt) increased 56% by the second-contact whisk ($P = 0.021$). By including measures of head rotation and movement along the wall (parallel to the floor), we can estimate that, during a typical approximately 100-ms whisk cycle, each rat moved 9–12 mm horizontally, raised or lowered its snout by 6–10 mm, and rotated its head through 5–6°. In other words, these animals were rarely stationary and often moved at quite significant speeds while exploring nearby surfaces with their whiskers.

EFFECTS OF HEAD MOVEMENT ON CONTACT VELOCITY. The above-cited findings suggest that the rat's head and body movements during whisking may be of sufficient magnitude to influence the nature of whisker deflections. These impacts could arise either by changing the effective speed of the whiskers as they meet the contacted surface, by changing the duration of contact, or by causing movement of the whisker shaft as it "sticks" or "slips" (Ritt et al. 2008) across a surface. Here we consider the likely impact of head movement on the effective velocity of the initial contact; effects during contacts will be briefly considered in *Sensory consequences of whisker control*.

Because the surfaces in our arena are flat, the vast majority of whisker–surface contacts in our data set were observed to begin at or near the tip of the whisker. We therefore make the simplifying assumption that the velocity at the whisker tip is a good approximation for the velocity at the point of contact on the whisker shaft. Using measurements of the lengths of contacting whiskers estimated over three successive video frames, and the formula

$$\text{Whisker tip velocity} = (\text{Whisker angular velocity}) \times (\text{Whisker length}) \times (2\pi/360)$$

we estimate that the mean velocity at the tip, immediately prior to contact, was 0.44 mm/ms in both the first- and second-contact whisks (SD 0.32 and 0.38, respectively). Mean head angular velocity of about 6°, in both contact whisk types, suggests that the effective tip velocity could increase or decrease by ±5% as the result of head rotation in a typical whisk. Combining whisker tip and head translation velocities (which were predominantly positive in both whisks) suggests that movement toward the wall increased the effective tip velocity at contact by around 27% to 0.60 mm/ms (SD 0.34) in the first-contact whisk and by 13% to 0.55 mm/ms (SD 0.38) in the second. Overall, then, we can conclude that head movements

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

contributed significantly to the velocity of whisker–surface impacts.

A closer look at whisker spread

In the previous two sections, whisker spread was first identified as the second largest component of whisking behavior, and then as one of the two whisker movement parameters (the other being retraction velocity, which we will subsequently consider) that change most significantly when a rat explores an unexpected vertical surface. We propose the hypothesis that variation in observed whisker spread is at least partly the consequence of differential control of whisker velocity and that significant changes in spread arise in response to surface contacts and in anticipation of future contacts. Evidence in support of this view will be provided in this section by evaluating, and presenting evidence against, three alternative explanations of the data presented so far: *1*) that variations in apparent spread occur primarily as the consequence of head movements (particularly head tilt) and thus do not arise from differences in how the whiskers themselves are controlled; *2*) that variation in spread across whisks arises as the result of correlated changes in other whisker control parameters and can be reduced or eliminated by controlling for these covariates; and *3*) that changes in spread occur only after surface contacts and not prior to, and in anticipation of, such contacts.

CAN APPARENT CHANGES IN SPREAD BE EXPLAINED BY HEAD MOVEMENT? As previously noted, a change in the orientation of the whisking plane can substantially affect the apparent spacing between whiskers observed in the overhead view; thus the differences in whisker spread that we have identified could have arisen partly, or wholly, as the consequence of changes in head position. To investigate this possibility we performed three analyses.

First, we looked separately at the protraction velocities for the most rostral and caudal tracked whiskers across each of the whisk types. We know from Fig. 2 that the whiskers are, overall, moving slightly more slowly in the second-contact whisk, although for spread to be reduced in this whisk the most caudal whiskers will have to match their velocities more closely to those of the rostral ones. This is true, of course, only if the changes in spread are the consequence of differential control of whisker velocities. If changes in observed spread are the result of head movement, then we should expect to see corresponding changes in whisker velocity, but all whiskers should be affected equally as they move in approximately parallel trajectories (see, e.g., Bermejo et al. 2002 and Fig. 6). In other words, head-movement and whisker-movement explanations of spread make quite different predictions concerning individual whisker velocities. A 3×2 ANOVA of the relevant whisker velocity data is provided in Supplemental Table S4 and the key results are summarized in Fig. 4. Here we see that although the more rostral whiskers did protract faster than the most caudal ones overall ($P < 0.001$), there was a significant interaction ($P < 0.001$) such that by the second-contact whisk, both rostral and caudal whiskers were moving at approximately the same speed and the most caudal whiskers were moving faster than in the preceding whisks. Since this result cannot be explained by head movement we therefore conclude that a significant part of the observed variance in whisker spread is due to the way in which the whiskers themselves are controlled.

Second, to investigate the possible effects of changes in head position on observed spread we plotted a scattergram, shown in Fig. 5, of snout elevation (used as an indicator of degree head tilt) versus mean spread for all three whisk types in all 60 clips. We then fitted separate polynomial curves to this distribution for each whisk type. Two observations arise from inspecting these figures. First, and most important, it can be seen that the curve for the second-contact whisk data lies beneath that for the other two types and does not overlap them at any point. This confirms that the observed reduction in spread in that whisk type, compared with the others, is relatively independent of this measure of head position. Second, it is noteworthy that all three curves in Fig. 5 are mildly U-shaped, with the largest spread values occurring when the snout is near the floor or the ceiling and the smallest when the snout is at medium height. Since the same five whiskers are tracked in all frames, this is the opposite of what one would expect if spread, as recorded in the overhead camera, was changing solely, or primarily, as the result of changes in head tilt. In that case, spread should be maximal somewhere in the central range of elevations, where whisker motion is parallel with the viewing plane of the camera, and should be smallest at the extremes,

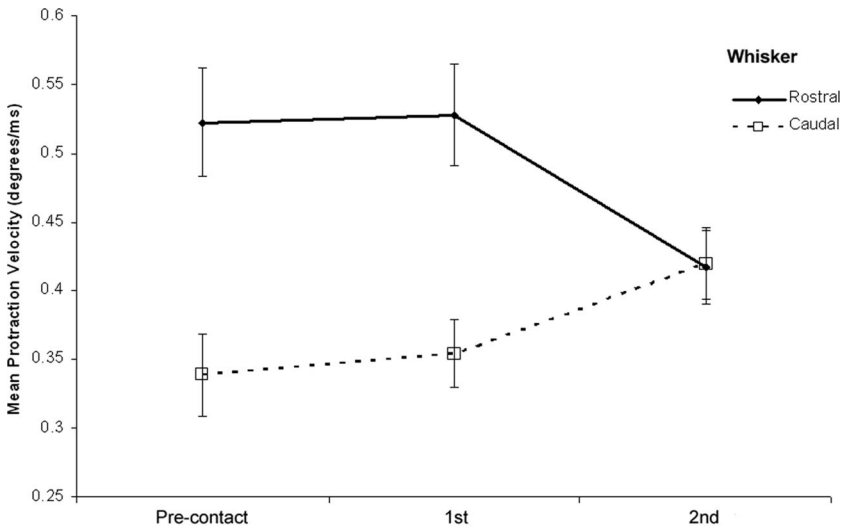


FIG. 4. Changes in the protraction velocities of the most rostral and most caudal tracked whiskers on encountering an unexpected surface. In the precontact whisk the most rostral whisker moves significantly faster than the most caudal one; however, the protraction velocities converge by the 2nd-contact whisk, consistent with a substantial reduction in whisker spread in that whisk.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

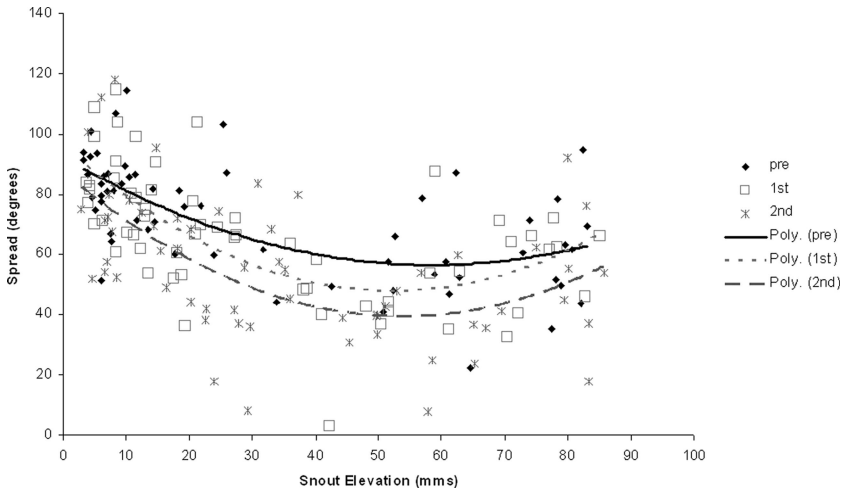


FIG. 5. Scattergram of whisker spread against snout elevation with best-fit curves for different whisk types. The plot shows that whisker spread (y-axis) is reduced in the 2nd-contact whisk across the full range of snout elevation (x-axis values). Curve fitting used 2nd-order polynomials. That the best-fit curves are mildly U-shaped suggests that tracking in the overhead view may lead to underestimation, rather than overestimation, of the extent to which spread varies within the plane of the whiskers.

where the whisking plane angles away from, or toward, the camera. That the range of spread values for different snout elevations shows the opposite trend implies that tracking in the overhead view may be causing us to underestimate, rather than overestimate, the extent to which spread is varying within the plane of whisker movement at different head tilts.

Finally, we looked at some illustrative data that compare whisker spread, as measured in the overhead view, with an alternative estimate of relative whisker spacing calculated within a coordinate frame defined by the whiskers themselves. Specifically, two representative clips were chosen for tracking in both overhead and end-on views and 3D trajectories calculated for a subset of whisker tips as described in the Supplemental Material. For both clips we then computed a per-frame, head-invariant spread measure as the average distance between all pairs of tracked whisker tip positions. A comparison between this new estimate of whisker spacing and the original viewpoint-dependent spread measure, for both whisking episodes, is shown in Supplemental Fig. S7. These graphs show a reasonable match between the two measures (Pearson's r correlations of 0.72 and 0.54 for the two clips) and a similar reduction in spread over the three whisk types, confirming that spread, as measured in the overhead view, captures a significant portion of the variance of a measure of whisker spacing that is independent of head movement.

ARE APPARENT CHANGES IN SPREAD THE RESULT OF CORRELATED CHANGES IN OTHER WHISK PARAMETERS? In a multivariate analysis it is important to control for the possible effects of covariates on a measure of interest. In this case, when calculated across all whisk types (i.e., $n = 180$), whisker spread was found to have a significant, though relatively small, positive correlation with maximum protraction ($r = +0.18$, $P = 0.015$) and a nonsignificant but positive correlation with minimum protraction ($r = +0.14$, $P = 0.068$) (correlations with other measured whisking parameters were negligible). These relationships are potentially important because we know that both maximum and minimum protractions are reduced on the second-contact whisk (Fig. 2). To establish whether variation in spread is independent of these other differences between whisk types, a univariate ANOVA (see Supplemental Table S5) was performed using values of mean spread calculated solely within the arc x to y degrees of the protraction phase, where x = max (minimum protraction) and y = min (maximum protraction), and where x and y were calculated separately within each clip across all three whisk types. After controlling for differences in minimum and maximum protractions in this way, mean spread was still found to decrease substantially by the second-contact whisk ($P < 0.001$) and by 17.9% compared with the precontact whisk, which is a similar reduction to that seen overall in the protraction phase (16.5%). We therefore conclude that the variation in spread seen across whisk types is not explained by differences in the protraction start and end positions.

DO CHANGES IN SPREAD OCCUR PRIOR TO SURFACE CONTACTS? Differences in observed spread between precontact and contact whisks could occur throughout the whisk cycle or could arise, primarily or wholly, as the result of changes in whisker movement taking place after the whiskers have touched the surface. To determine whether there were significant changes in whisker spread prior to any surface contact, an analysis (see Table S5) was conducted in which mean spread was calculated solely within the arc x to z degrees of the protraction phase where x = max (minimum protraction) (as above) and z = min (mean angular position prior to contact). Again, as in the previous test, mean spread still showed a significant decrease (-14.3%, $P < 0.001$) by the second-contact whisk, confirming that there is a significant reduction in whisker spacing prior to surface contact. In this context it is also worth looking briefly at differences in spread in the retraction phase of the whisk. Again, and as expected, this showed a significant reduction across the whisk types ($P < 0.001$); however, notably, here the reduction in spread began in the first-contact whisk (-14.1% compared with precontact). This result suggests that, following the initial contact, whisker control begins to adapt almost immediately to provide reduced whisker spread during further exploration of the contacted surface.

Sensory consequences of whisker control

The active touch hypothesis for rat whisking postulates that the control of whisker positioning is regulated on a moment-to-moment basis to provide the animal with better or more task-relevant tactile sensory signals. In this final subsection we examine three aspects of whisker–surface interactions—the quantity of whisker–surface contacts, the amount of bending during contacts, and the duration of contacts—and attempt to

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

pinpoint some of the elements of whisker control that could be actively regulated to modify each of them.

INCREASING THE QUANTITY OF SURFACE CONTACTS BY CONTROLLING WHISKER SPREAD AND HEAD TILT. Reducing spread through differential control of whisker movement brings the whisker tips closer together and should allow the rat to increase the density of sensory signals from targeted regions of space where objects or surfaces are expected. To verify this "maximizing contact" effect we calculated correlations between the number of contacts on tracked whiskers and a range of potential predictor variables for the combined set of 120 first- and second-contact whisks; a stepwise multiple linear regression was then performed using the most promising candidate predictors. This analysis showed that the number of contacts increased both with the inverse mean spread (lower values of spread are most discriminative), which showed a part correlation of +0.202 ($P = 0.019$) after removing the affects of other predictors, and with increasing snout elevation (part correlation +0.195, $P = 0.024$). Further details of these analyses, together with residual plots confirming that the relationships were approximately linear, are provided in the Supplemental Material. We conclude that the rat can increase the number of surface contacts by reducing whisker spread. The increase in contacts due to greater snout elevation remains to be explained and is briefly considered next.

Figure 6 shows the whisker tip trajectories for a first-contact whisk (*left*) and a second-contact whisk (*right*), calculated using tracking data from both camera views using our 3D trajectory reconstruction algorithm, and then rotated (in the bottom half of the figure) so that the trajectories can be observed in a plane perpendicular to both the wall and the arena. For illustrative purposes the tracked video clips used to generate these trajectories are also provided as Supplemental Videos S2 and S3. The *left panel* shows that during a whisk with low elevation many whiskers, and particularly the more caudal ones, make contact with the floor and would do so even if the rat were closer to the wall. Increased floor contact and reduced wall contact are likely simply because the rat's head tilts down toward the floor at a significant angle. For the whisk with higher snout elevation, shown in the *right panel*, whisker movement is close to being perpendicular to the wall, and the head position near horizontal, so we can expect, and do observe, many more wall contacts and few floor contacts. What happens for still higher elevations? When the head tilts above the horizontal this will bring some of the longer more caudal whiskers closer to the wall, increasing the likelihood that these whiskers will touch even as some of the dorsal whiskers rotate away from the wall (and may begin to touch the ceiling). We can thus conclude that tilting the head upward is, overall, a useful "maximizing contact" strategy for exploring walls, whereas tilting downward is clearly favorable for whisking during floor traversal.

CONTROLLING BENDING AGAINST SURFACES THROUGH RAPID CESSATION OF PROTRACTION FOLLOWING CONTACT. In our previous study (Mitchinson et al. 2007) we found that, following a unilateral surface contact, the whiskers on that side of the snout cease protraction on average 13 ms after the initial touch. In the current study we used a larger sample size (60 vs. 22 clips), a higher frame rate (500 vs. 250 frames/s), and measured the average time from contact to maximum protraction across all five tracked whiskers. Performing this analysis for data from the first-contact whisk produced a mean time to maximum

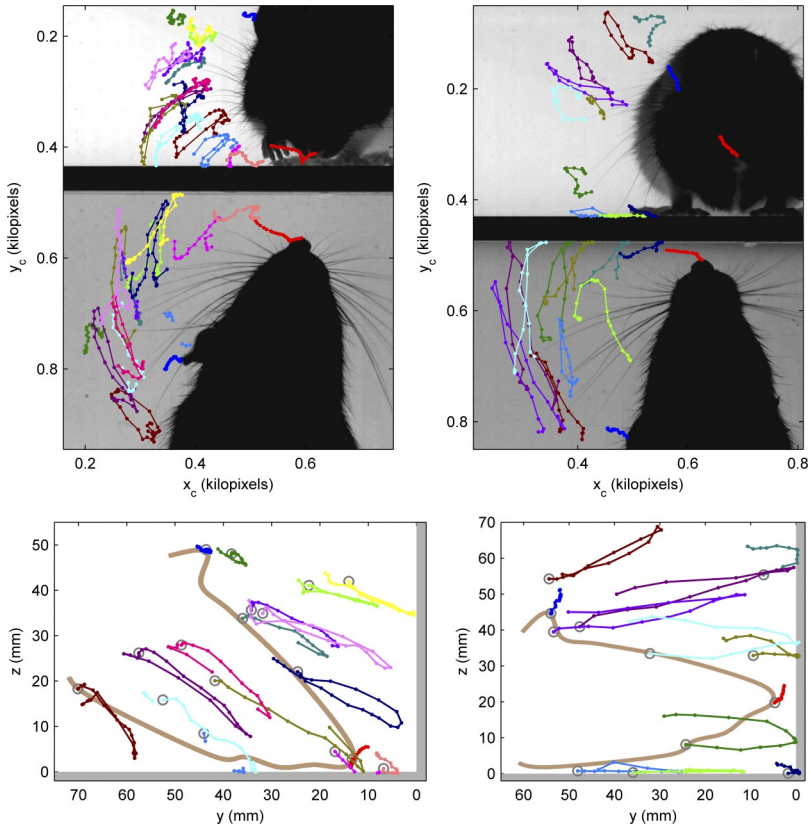


FIG. 6. Whisker tip trajectories for a 1st-contact (*left*) and a 2nd-contact whisk (*right*). *Top*: a high-speed video frame from each clip with whisker tip trajectories overlaid in both the side-on (mirror) and overhead views. Colored lines show trajectories of individual whiskers matched across the 2 views using a stereo correspondence algorithm. The tip of the snout (red) and of one ear (blue) were also tracked in the both whisks. *Bottom*: reconstruction of whisker tip trajectories in a rotated view showing movement in a plane perpendicular to both the floor and the end wall (surfaces shown by thick gray outlines). Axes show distance to wall and height above the floor in millimeters. The approximate shape and position of the head are illustrated by the brown outline. The video clips from which the trajectories were reconstructed are provided as Supplemental Videos S2 and S3 with tracking overlaid.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

protraction of 14.3 ms—i.e., a result that is well within the expected margin of error of our previous investigation (±2 ms). When the same calculation was performed for the second-contact whisk, interestingly, the time to maximum protraction fell to 11.3 ms, a latency that was marginally faster than that for the first contact, although this result should be treated with some caution due to the relatively high P value (0.038, which exceeds the Bonferroni corrected alpha of 0.01). To further test the robustness of these estimates we separately examined the time to maximum protraction for each of the tracked whisker columns. As shown in Supplemental Fig. S9, the difference between first- and second-contact whisks was found consistently across all columns, whereas there was also an interesting trend for the more rostral whiskers to reach maximum protraction earlier than the more caudal ones by up to 5 ms. Time to maximum protraction was also compared for whisker columns that did, and did not, make contact with the wall; these were found to be very similar ($P = 0.974$), indicating that the contacts themselves were not significantly distorting this measure. An analysis by contact type (bilateral vs. unilateral; see Supplemental Fig. S10) also found no significant effect on the timing of maximum protraction ($P = 0.095$). Overall then we can conclude that rapid cessation of protraction is a general and consistent feature of whisker–surface contacts during exploratory whisking, which may possibly be more pronounced (faster) in subsequent whisks that in the initial contact, and whose effect will be to reduce bending of the whiskers against the surface compared with nonmodulated whisks.

CONTACT DURATION—A FURTHER CANDIDATE FOR ACTIVE CONTROL? On first inspection, the finding that whisker–surface contacts have much longer duration appears to be inconsistent with the rapid cessation of protraction and reduced head velocity toward the wall in the second-contact whisk. However, the likely explanation for these longer contact times is easily found by reviewing Fig. 2—here we saw that whisk retraction velocity was at a significantly slower pace on the second-contact whisk than on the first. To understand this further, we next divided the retraction phase into two halves and found that the decrease in retraction velocity in the second-contact whisk was much more evident in the first half of the retraction phase (−54%, $P < 0.001$) than in the second half (−9%, $P = 0.237$). This establishes that slower retraction occurs during the period of the whisk immediately following peak protraction and is thus most likely to influence contact duration. To confirm that retraction velocity was genuinely slower following the second contact, and not simply that drag on the contacting whiskers created the appearance of slower retraction, we looked at retraction velocity on the most caudal tracked whisker, since this rarely contacts the vertical surface (only three contacts in the 60 second-contact whisks). For this whisker, too, velocity during the first half of the retraction phase was also considerably slower on the second-contact whisk (−49% compared with −47% for the most rostral whisker, both $P < 0.001$). Details of each of these analyses are provided in Supplemental Table S6.

To further establish which control factors most influence contact duration we performed a stepwise multiple linear regression using head- and whisker-movement parameters that were likely predictors of duration. The dependent variable was contact duration measured for the 95 first- and second-contact whisks in which contact with the surface ceased before the end of the whisk cycle (60 first-contact and 35 second-contact whisks). This analysis showed that contact duration was most strongly predicted by time from contact to maximum protraction (part correlation of $r = +0.487$), retraction velocity ($r = -0.293$), snout vertical velocity ($r = -0.246$), and snout elevation ($r = +0.183$). Further details of this analysis, including residual plots confirming that the relationships were approximately linear, are provided in the Supplemental Material. The high positive correlation with the time from contact to maximum protraction confirms the importance of controlling protraction cessation, using sensory feedback, to the tactile experience of the rat. The negative correlation with retraction velocity shows that this parameter is a further important element of whisking strategy in determining the duration of whisker–surface contact signals. We leave to the DISCUSSION consideration of why retraction velocity is reduced in the second-contact whisk and whether increasing contact duration through this mechanism should be considered as an additional active touch strategy.

DISCUSSION

Sensory signals are generally ambiguous, sometimes entirely meaningless, in the absence of knowledge of how the sensor that generated those signals was controlled. This is particularly true of touch, where signals are obtained only through physical contact and where the trajectory of the sensor with respect to the surface codetermines, with the surface properties of the object, the nature of the signals that are obtained. Increasing evidence, both in this study and in other studies, indicates that the sensor (whisker) trajectories of the rat vibrissal system are carefully controlled, modified on the basis of recent sensory experience and in anticipation of future experience, and directed at obtaining high-quality, task-relevant information. This type of active sensing control is also apparent in human fingertip touch (Chapman 1994; Lederman and Klatzky 1993; Smith et al. 2002), suggesting that the rat whisker system can be a useful model in which to investigate "sensorimotor contingencies" (O'Regan and Noe 2001) similar to those underlying our own tactile experience of the world. In the following, our new findings in relation to active touch sensing in the rat are summarized and evaluated, beginning with the evidence that whisker spread is actively controlled, then turning to the active touch sensing strategies that appear to be used by the rat.

Control of whisker spread

WHISKER SPREAD EXPLAINS A SIGNIFICANT PORTION OF WHISKING VARIANCE. Although changes in the horizontal spacing of whiskers have been previously noted (Sachdev 2002), this is the first study to have quantified the contribution of these changes to the overall observed whisking pattern. We found that 13–19% of the variance in whisker angular position data for freely moving animals, tracked from overhead, can be accounted for in terms of one summary parameter—the "spread" or angle of arc between the rostral-most and caudal-most tracked whisker. Further, together with the mean angular rotation of the whisker field these two parameters can account for ≤93% of the variance seen in the overhead view.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

WHISKER SPREAD VARIES ACROSS DIFFERENT WHISK TYPES. It is important to be clear that we regard the spread parameter as simply a descriptive measure that usefully summarizes some of the observed changes in whisking behavior. We have no direct evidence that the rat brain encodes spread as a specific control parameter (any more than it encodes amplitude, frequency, or set point as control parameters, all of which are also descriptive concepts). That the spacing between whiskers changes with time is consistent with, but does not necessarily require, active control of underlying mechanisms. Passive, rather than active, control (such as might be provided by purely mechanical properties of the vibrissal system) would imply that whisker spread should vary across the whisk cycle in a predictable and consistent manner across all types of whisk. To obtain evidence that observed changes in spread are not simply passive we needed to demonstrate variability across different whisking contexts; therefore we investigated whether anticipation of a proximal vertical surface could change the pattern of spread changes over the course of a whisk cycle. Our data demonstrate such variability. Specifically, we have shown that in the second-contact whisk, whisker spread and angular position can become decoupled, that spread is typically reduced compared with preceding whisk cycles, and most reduced following bilateral whisker–surface contacts.

EXPLAINING THE OBSERVED VARIANCE IN SPREAD ACROSS WHISK TYPES. Observed spread thus appears to vary across whisking contexts, but what is the source of this variation? In our analysis we tested and provided evidence against several alternative explanations, concluding that observed changes in spread were not simply the consequence of head movements, changes in other aspects of whisker control, or the result of the whiskers bending against the contacted surface. Most tellingly, perhaps, we found evidence of differences in whisker velocity indicating that the most rostral whiskers move substantially faster than the most caudal ones in precontact and first-contact whisks, but not in second-contact whisks. This result is difficult to explain other than by assuming some capacity for differential control of the whiskers themselves.

COULD CHANGES IN WHISKER SPREAD INVOLVE CONTROL OF THE WHISKING MUSCULATURE? Although we have ruled out the most plausible alternatives, based on the current data we cannot directly test the hypothesis that the apparent changes in whisker spread involved control of the whisking musculature. Results from other laboratories (Berg and Kleinfeld 2003; Dorfl 1982; Hill et al. 2008; Klein and Rhoades 1985; Wineski 1985), however, show that the rat does have sufficient degrees of freedom of whisker control to effect some differential movement of either individual whiskers or whisker columns. Moreover, some divergent movement of the whiskers has been observed in animals trained to make texture discriminations (Carvell and Simons 1990) and in head-fixed animals trained to whisk for reward (Sachdev et al. 2002). That the rat has the capacity to focus its whisker field toward a target has also been suggested before in the context of the "foveal whisking" behavior described by Berg and Kleinfeld (2003). In the following we briefly compare some of the observations made in that study with our current results.

WHISKING MODULATION AND WHISKING MODES. Berg and Kleinfeld (2003) used high-speed videography and electromyographic recording of the whisking musculature to investigate whisking behavior in rats that were trained to explore a maze to obtain food rewards. They described two general modes of whisking behavior that were distinguished both by their spectral properties (whisk amplitude and frequency) and by differential patterns of activation in the whisking musculature. The first mode, termed "exploratory whisking," consisted of bouts of relatively large amplitude whisks occurring at a frequency 5–15 Hz. In the second, less frequent "foveal whisking" mode, rats exhibited bouts of relatively small amplitude but high-frequency (15–25 Hz) whisking. During general maze traversal, while the animals searched for a food resource, they exhibited exploratory whisking; however, the animals shifted to the foveal mode when required to "crane their necks" across a gap to reach a food tube.

The changes in whisking control during exploration of surfaces described herein are more subtle than the marked switch from one whisking mode to another described by Berg and Kleinfeld (2003). Indeed, in the current data, the frequency and velocity of the whisker movement changed relatively little on surface contact and, instead, we saw differences in some less well studied whisking parameters such as the whisker spread and the mean retraction velocity. We consider, then, that the whisking patterns we have observed here fall within the general class of "exploratory whisking" described by Berg and Kleinfeld (2003), and we suggest that, within this mode, the rat has the capacity to modulate whisking control on a per-whisk basis, and, to some degree, per-whisker (or whisker column) basis. It is worth noting that Berg and Kleinfeld (2003) described foveal whisking as involving the vibrissae being "clustered in front of the head in a relatively dense pattern." Although this clustering is not precisely quantified it does seem consistent with what we are calling a change in whisker spread. If differential use of the whisking musculature can bring about the substantial changes in whisker movement seen following the transition to foveal whisking, it seems reasonable to suppose that similar but subtler changes in muscular control could also underpin the reduction in whisker spread we have observed when rats explore a proximal surface.

Active touch sensing in the rat

We have previously proposed (Mitchinson et al. 2007) that rat whisking uses active control strategies that serve to increase the number of whiskers contacting surfaces of interest ("maximizing contact") while controlling the amount of bending against those surfaces ("minimizing impingement"). In the current study, the reduction in whisker spread and associated changes in head tilt, found in the second-contact whisk, are consistent with both strategies because they allow an increased number of whiskers to make contact with the vertical wall without requiring that the whiskers necessarily press harder against that surface. Several other aspects of whisking and head control that might also be considered to be part of the rat's active sensing strategy are considered in the following text.

MINIMIZING IMPINGEMENT BY CONTROLLING CESSATION OF PROTRACTION. Consistent with our earlier findings (Mitchinson et al. 2007), the current study found that whisker protraction ceased rapidly following an initial contact with a surface (mean of 14.30 ms from contact to maximum protraction in the

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

first-contact whisk). In the case of unilateral contacts, we previously found a difference between the ipsilateral (to the contact) whisker field, where protraction stopped soon after contact, and the contralateral field (where there was no contact) where it did not. From this we inferred the existence of a fast sensory feedback loop controlling the timing of whisker protraction to implement a "minimal impingement" strategy and ensure that contacts were made with a relatively "light touch." Our new data suggest that rapid cessation of protraction occurs for both unilateral and bilateral contacts and may even be quicker on the second-contact whisk (mean latency 11.25 ms). The latter finding, if supported by future studies, would imply some additional element of anticipatory control. The literature on classical conditioning shows that the latency of a reflex response, such as the rabbit eyeblink, is significantly reduced when the animal can anticipate the timing of the unconditioned stimulus (Gormezano et al. 1983). Thus similarly, the rat's ability to anticipate a forthcoming surface contact could influence the control circuitry underlying the proposed whisking sensory feedback loop, enabling it to respond more rapidly when an expected whisker deflection takes place.

CONTROLLING THE DURATION OF WHISKER SURFACE CONTACTS. An unanticipated finding of the current study was that the duration of contacts with surfaces was generally much longer in the second-contact whisk. Duration of contact was found to be best predicted by two whisking control parameters—the time from contact to maximum protraction (discussed earlier) and the retraction velocity (especially in the first half of the retraction phase). These two parameters oppose each other, but, by their interplay, it would appear that the rat could control the duration of contact, the speed at which the whisker is drawn across the surface, and the amount of bending in the whisker shaft. We think it is possible that the slower retraction during the second-contact whisk can be understood as an active touch strategy aimed at prolonging contact and thereby aiding the extraction of information about surface characteristics such as texture. Thus perhaps, the first-contact whisk could be thought of as locating the surface in space and the second as discerning more details concerning the nature of that surface. However, it is important to consider that there may be alternative explanations of reduced retraction velocity on the second-contact whisk that are not directly concerned with contact duration or with the sensory consequences of this contact. For instance, one possibility is that there might be compensatory mechanisms within the whisker pattern generator that act to reduce whisk retraction velocity following early cessation of protraction due to surface contact. Such a mechanism might conceivably operate to prevent a strong mismatch in phase between the left and right whisker fields, since it can be generally observed that the two fields have a strong tendency to return to synchronized movement following perturbation. Evaluation of this alternative will require a better understanding of the coupled motor pattern generators that generate whisker movements in the two fields and of their modulation by sensory signals. Future research on this topic should also benefit from the investigation of generative computational and robotic models of whisker geometry, musculature, and neural control systems (e.g., Hill et al. 2008; Mitchinson et al. 2006; Pearson et al. 2007).

COMBINING HEAD AND WHISKER MOVEMENT TO MAKE EXPLORATION MORE EFFECTIVE. Although going somewhat beyond the current data, we now propose the following hypothesis, con-

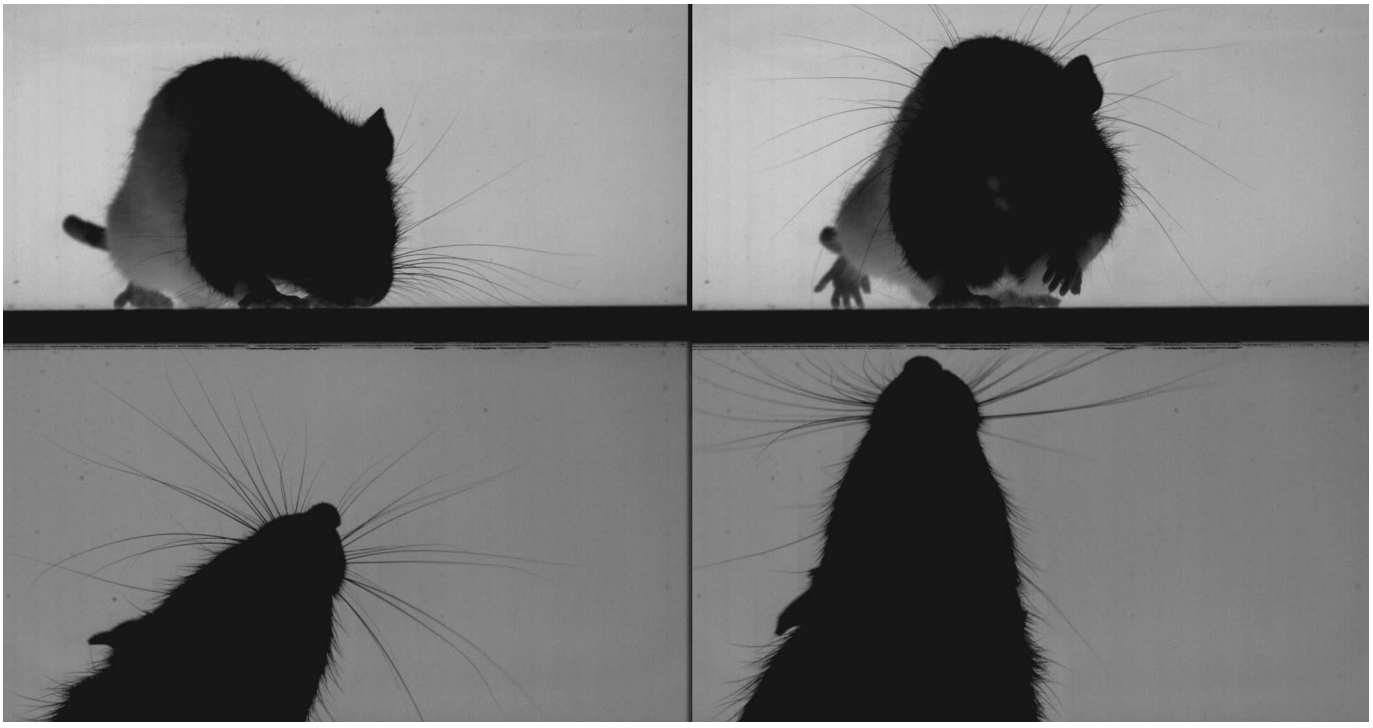


FIG. 7. Control of whisker spread and head position in active touch. Snapshots consistent with the hypothesis that the rat uses it body, neck, and whisker musculature so as to reduce whisker spread perpendicular to a surface of interest, while increasing spread parallel to that surface. Thus when the rat is moving across the floor, spread is reduced in the end-on view (*left top*) and increased in the overhead view (*left bottom*), whereas when exploring a wall, 2 whisks later, spread is increased in the end-on view (*right top*) and reduced in the overhead view (*right bottom*).

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

sistent with the above-cited active sensing strategies that could serve as a further simplifying principle for understanding whisking control in exploring animals.

The rat appears to control its whiskers so that the spacing between the whiskers is reduced relative to the surface of interest. Thus if this is a vertical surface we see reduced spread in the overhead view (as demonstrated by our data); if a horizontal surface (e.g., the floor) we see reduced spread in the end-on view (consistent with our informal observations of video recordings but remaining to be demonstrated quantitatively). A strong version of this hypothesis would suggest that the rat also seeks to increase spacing parallel to the surface of interest, to simultaneously explore as much of that surface as possible. Thus when proceeding across the floor the whiskers appear relatively spread out when viewed from above and directed at the area of the floor around and immediately in front of the animal in its direction of motion (see, e.g., Fig. 7, *left*). In contrast, when investigating a vertical wall, the whiskers appear close together in the overhead view and much more widely separated in the end-on view (see, e.g., Fig. 7, *right*). These changes in whisker spacing are likely brought about partly through differential control of the whiskers and partly through controlled positioning of the head with respect to the surface of interest. Thus obtaining a better understanding of the interaction between head movements and whisking movements will be important to be able to fully characterize the active touch sensing strategies of the rat.

ACKNOWLEDGMENTS

We thank P. Redgrave for scientific advice and M. Simkins, L. Hetherington, A. Ham, and M. Benn for technical support.

GRANTS

This research was supported by European Union Integrating Cognition, Emotion, and Autonomy Grant IST- 027819 and Biomimetic Technology for Vibrissal Action Touch Grant ICT-215910. R. A. Grant was funded by a doctoral training grant from the United Kingdom Engineering and Physical Sciences Research Council.

REFERENCES

Aloimonos JY, Weiss I, Bandopadhay A. Active vision. *Int J Comput Vision* 1: 333–356, 1988.

Ballard DH. Animate vision. *Artif Intell* 48: 57–96, 1991.

Berg RW, Kleinfeld D. Rhythmic whisking by rat: retraction as well as protraction of the vibrissae is under active muscular control. *J Neurophysiol* 89: 104–117, 2003.

Bermejo R, Vyas A, Zeigler HP. Topography of rodent whisking. I. Two-dimensional monitoring of whisker movements. *Somatosens Mot Res* 19: 341–346, 2002.

Carvell GE, Simons DJ. Biometric analyses of vibrissal tactile discrimination in the rat. *J Neurosci* 10: 2638–2648, 1990.

Chapman CE. Active versus passive touch: factors influencing the transmission of somatosensory signals to primary somatosensory cortex. *Can J Physiol Pharmacol* 72: 558–570, 1994.

Dorfl J. The musculature of the mystacial vibrissae of the white-mouse. *J Anat* 135: 147–154, 1982.

Etienne AS, Maurer R, Seguinot V. Path integration in mammals and its interaction with visual landmarks. *J Exp Biol* 199: 201–209, 1996.

Gao P, Bermejo R, Zeigler HP. Whisker deafferentation and rodent whisking patterns: behavioral evidence for a central pattern generator. *J Neurosci* 21: 5374–5380, 2001.

Gibson JJ. Observations on active touch. *Psych Rev* 69: 477–491, 1962.

Gormezano I, Kehoe EJ, Marshall BS. Twenty years of classical conditioning research with the rabbit. In: *Progress in Psychobiology and Physiological Psychology*, edited by Sprague JM, Epstein AN. New York: Academic Press, 1983, p. 197–275.

Gustafson JW, Felbain-Keramidas SL. Behavioral and neural approaches to the function of the mystacial vibrissae. *Psychol Bull* 84: 477–488, 1977.

Hartmann MJ. Active sensing capabilities of the rat whisker system. *Auton Robots* 11: 249–254, 2001.

Hetherington L, Benn M, Coffey PJ, Lund RD. Sensory capacity of the Royal College of Surgeons rat. *Invest Ophthalmol Vis Sci* 41: 3979–3983, 2000.

Hill DN, Bermejo R, Zeigler HP, Kleinfeld D. Biomechanics of the vibrissa motor plant in rat: rhythmic whisking consists of triphasic neuromuscular activity. *J Neurosci* 28: 3438–3455, 2008.

Klein BG, Rhoades RW. Representation of whisker follicle intrinsic musculature in the facial motor nucleus of the rat. *J Comp Neurol* 232: 55–69, 1985.

Knutsen PM, Biess A, Ahissar E. Vibrissal kinematics in 3D: tight coupling of azimuth, elevation, and torsion across different whisking modes. *Neuron* 59: 35–42, 2008.

Knutsen PM, Derdikman D, Ahissar E. Tracking whisker and head movements in unrestrained behaving rodents. *J Neurophysiol* 93: 2294–2301, 2005.

Lederman SJ, Klatzky RL. Extracting object properties through haptic exploration. *Acta Psychol* (*Amst*) 84: 29–40, 1993.

Lungarella M, Pegors T, Bulwinkle D, Sporns O. Methods for quantifying the informational structure of sensory and motor data. *Neuroinformatics* 3: 243–262, 2005.

Mitchinson B, Martin CJ, Grant RA, Prescott TJ. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci* 274: 1035–1041, 2007.

Mitchinson B, Pearson M, Melhuish C, Prescott TJ. A model of sensorimotor coordination in the rat whisker system. In: *From Animals to Animats 9: Proceedings of the 9th International Conference on Simulation of Adaptive Behaviour, SAB 2006, Rome, Italy, September 25–29, 2006*, edited by Nolfi S, Baldassarre G, Calabretta R, Hallam J, Marocco D, Miglino O, Meyer J-A, Parisi D. Berlin: Springer-Verlag, 2006, p. 77–88.

O'Regan JK, Noe A. A sensorimotor account of vision and visual consciousness. *Behav Brain Sci* 24: 939–1031, 2001.

Pearson MJ, Pipe AG, Melhuish C, Mitchinson B, Prescott TJ. Whiskerbot: a robotic active touch system modeled on the rat whisker sensory system. *Adapt Behav* 15: 223–240, 2007.

Ritt JT, Andermann ML, Moore CI. Embodied information processing: vibrissa mechanics and texture features shape micromotions in actively sensing rats. *Neuron* 57: 599–613, 2008.

Sachdev RN, Berg RW, Champney G, Kleinfeld D, Ebner FF. Unilateral vibrissa contact: changes in amplitude but not timing of rhythmic whisking. *Somatosens Mot Res* 20: 163–169, 2003.

Sachdev RN, Sato T, Ebner FF. Divergent movement of adjacent whiskers. *J Neurophysiol* 87: 1440–1448, 2002.

Sellien H, Eshenroder DS, Ebner FF. Comparison of bilateral whisker movement in freely exploring and head-fixed adult rats. *Somatosens Mot Res* 22: 97–114, 2005.

Smith AM, Gosselin G, Houde B. Deployment of fingertip forces in tactile exploration. *Exp Brain Res* 147: 209–218, 2002.

Stuttgen MC, Ruter J, Schwarz C. Two psychophysical channels of whisker deflection in rats align with two neuronal classes of primary afferents. *J Neurosci* 26: 7933–7941, 2006.

Towal RB, Hartmann MJ. Right-left asymmetries in the whisking behavior of rats anticipate head movements. *J Neurosci* 26: 8838–8846, 2006.

Towal RB, Hartmann MJ. Variability in velocity profiles during free-air whisking behavior of unrestrained rats. *J Neurophysiol* 100: 740–752, 2008.

Vincent SB. The function of the vibrissae in the behaviour of the white rat. *Behav Monogr* 1: 1–82, 1912.

Voigts J, Sakmann B, Celikel T. Unsupervised whisker tracking in unrestrained behaving animals. *J Neurophysiol* 100: 504–752, 2008.

Wineski LE. Facial morphology and vibrissal movement in the golden hamster. *J Morphol* 183: 199–217, 1985.

Downloaded from http://jn.physiology.org/ by guest on June 13, 2013

ORIGIN-DESTINATION ANALYSIS ON THE LONDON ORBITAL AUTOMATED NUMBER PLATE RECONGITION NETWORK

Charles Fox
Adaptive Behaviour Research Group, University of Sheffield, UK
Peter Billington
Telematics Technology LLP, Sheffield, UK
Dominic Paulo
Mouchel, London, UK
Clive Cooper
Highways Agency, Dorking, UK

ABSTRACT

The UK Highways Agency (HA) has a specific interest in understanding the actual journeys taken by drivers using the London orbital motorway (M25) in order to assist with planning and management of strategic routes. It is thought that the majority of traffic on Public Service Agreement (PSA) Routes, rather than travelling end to end is joining and leaving mid-route. We provide a tracking algorithm to detect journeys on the routes using data from the existing automated number plate recognition (ANPR) network. This network was designed for other purposes and in particular only covers subsets of lanes at sites, requiring inference of traffic behaviour in the non-covered lanes. Furthermore, its cameras translate plate numbers into non-unique hash values which give rise to many spurious matches. A full-lane calibration study was performed, and fused with induction-loop flow data to obtain parameters for such inferences.

Introduction

The UK Highways Agency (HA) has a specific interest in understanding the actual journeys taken by drivers using the London orbital motorway (M25) in order to assist with planning and management of strategic routes. It is thought that the majority of traffic on Public Service Agreement (PSA) Routes, rather than travelling end to end is joining and leaving mid-route, possibly travelling between only one or two junctions.

Motorway Incident Detection and Signalling (MIDAS) data is available which contains count, speed and occupancy information, the analysis of which provides an understanding of the stresses on individual links and the turning movements of traffic at specific junctions. However, this data does not readily provide useful estimates of where vehicles start and end their journeys on the M25.

The HA's goal is the construction of an Origin-Destination (O-D) matrix, in the form of a look-up table having all possible origins along one axis and all possible destinations along the other, each cell in the matrix giving the number or relative proportion of vehicles making the journey between the relevant orthogonal origin/destination pair. The provision of such a table would enable a greater

understanding of origin-destination movements and the operational implications of the real routes taken by traffic around the M25 and on relevant arterials.

The present ANPR infrastructure (National Traffic Control Center, NTCC) on the M25 and its arterials was designed for other purposes and in particular does not provide full lane coverage. Instead, single or pairs of cameras are positioned at sites covering subsets of the lanes. The proportion of traffic detected by these partial lanes varies according to the flow of traffic. For example, a camera in lane 1 of a three-lane motorway will detect most of the traffic during low flows, but only a third of the traffic when the motorway is saturated. Conversely, a pair of cameras on lanes 2 and 3 will detect a low proportion of traffic during low flows but a higher portion nearing saturation. A single camera in lane 2 may show a more complex flow/detection relationship.

The situation is further complicated by the hashing scheme used by the existing infrastructure. The existing cameras use a hashing scheme which represents each number plate by a 24-bit tag (recently upgraded from 18 bit tags). A 24 bit value provides a maximum of 16,777,216 possible tags to represent approximately 30 million vehicles in the UK, so it might be expected that tags matched between any two specific locations would have a high degree of uniqueness.

However, a process known as “character merging” is applied prior to generation of the hash value from the plate characters. In this process, character sets which are commonly mis-read by ANPR systems (such as O, D and 0, or 8, B and 3) are merged (e.g. all Ds are replaced by Os etc) which has the effect of reducing significantly the uniqueness of the hash value. Thus matches between tags at the origin and destination of a route must be processed to removed spurious matches due to hashing clashes (where different plates give rise to the same hash value), as well as to adjust for the flow-affected detection proportion.

In this study, we recorded one day of calibration data from 12 routes of different lane configurations, using higher quality cameras (Telematics Technology LLP, Sheffield, UK) which have a higher detection rate and report raw plate numbers rather than hashes. The calibration cameras were placed to cover all lanes of the calibration routes, and allow us to report how the proportion of traffic in each lane varies as a function of total flow as measured by MIDAS data. These multipliers are used in conjunction with a tracking algorithm to obtain estimates of origin-destination route profiles for a variety of day types.

Ontology

Each physical NTCC camera is positioned over one lane of a road, which may be a motorway, arterial or slip-road. Typically there are one, two or occasionally three cameras covering lanes of the same road, facing in the same direction. Such a group of cameras is called a site. A site thus defines a road having a number of lanes, a direction of travel, and a subset of those lanes which are covered by cameras. An outstation is a physical box which collects data from one or more sites (e.g. two sites facing in opposite directions) and relays the data to NTCC. A lane configuration describes which subset of lanes is covered for an *n*-lane road. The most common lane configurations for 2-lane roads in the NTCC network are lanes (1) and (1,2). The most common lane configurations for 3-lane roads are lanes (2),

(1,2) and (1,3). An origin-destination configuration is an ordered pair of lane configurations corresponding to on origin and a destination. A route consists of an original site and a destination site. A calibration route consists of origin and destination locations corresponding to sites but uses data collected from all lanes using Telematics calibration cameras rather than by NTCC site cameras. A profile is a graph showing the number of journeys on a route as a function of time of day.

A link is a short (100m) section of a road in a direction, and refers to all lanes of the road. Flow data (MIDAS) measures the number of vehicles per 5-minute interval at each link.

Each calendar day in the study was given a day type from 0-11. For example, 0 is working Mondays, 1 is working Tuesdays, 8 is holidays and 9 is the first day of school holidays. The present study finds profiles for 108 routes on and around the M25, using eight weeks (8Gb) of NTCC data.

The diamond-shaped icons in Figure 2 represent NTCC ANPR camera locations. The triple-segmented orange lines between camera locations in Figure 2 represent the routes over which profiles have been evaluated.

Matching algorithms

Both the calibration study and the 8Gb NTCC profiling require matching vehicles to be identified at the origin and destination. In the calibration study we have access to (noisy observations of) the plate numbers themselves, given by the Telematics cameras. In the profiling we have noisy 20-bit plate hashes. Both data sets are vulnerable to containing large numbers of false positive matches, which we filter using a journey time tracking algorithm.

Running for one route on one day's data, the algorithm begins by retrieving all candidate matches using a large SQL query, such shown in Figure 1. (The bulk plate log contains times, sites and hashes of ANPR detections).

```
SELECT spl_d.hashPlate,
    spl_o.plateTimestamp_GMT AS timeOrig,
    spl_d.plateTimestamp_GMT AS  timeDest,
       spl_o.siteId AS siteOrig,
       spl_d.siteId AS siteDest,
       spl_d.plateTimestamp_GMT-spl_o.plateTimestamp_GMT
    AS travelTimeSecs
FROM BulkPlateLog AS spl_d INNER JOIN
                  BulkPlateLog AS spl_o
                  ON spl_d.hashPlate = spl_o.hashPlate
WHERE spl_d.plateTimestamp_GMT>spl_o.plateTimestamp_GMT
AND             spl_o.siteId = '30320545'
AND             spl_d.siteId = '30320549'
AND spl_o.plateTimestamp_GMT > '2009-06-30 22:00:00'
AND spl_o.plateTimestamp_GMT < '2009-07-02'
AND spl_d.plateTimestamp_GMT > '2009-06-30 22:00:00'
AND spl_d.plateTimestamp_GMT < '2009-07-02'
ORDER BY  spl_d.plateTimestamp_GMT
```

Figure 1: SQL query used to recover all possible matches for a route on a date.

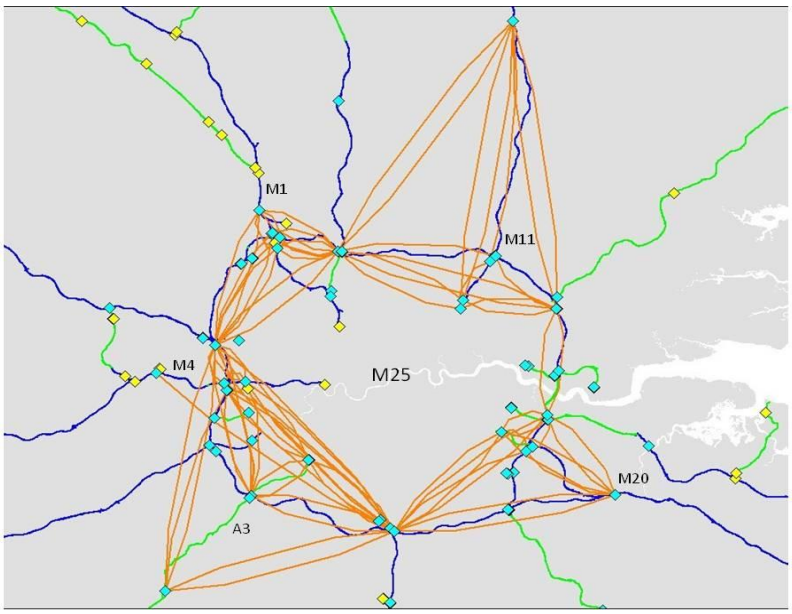


Fig.2 Map of the M25 London orbital, showing routes used.

Careful use is made of SQL clustered indexing to allow such queries to run in a few seconds over the 8Gb data set. The query returns all pairs of origin-destination plate detections having the same plate number or hash, which occur on the date (beginning at 22:00 the previous day), and where the destination detection occurs after the origin detection. This gives rise to a set of candidate journey durations, *y(t)* (where *t* ranges over the discrete set *T* of times at which matches occur). As the day progresses, the number and journey durations of spurious matches increases, as shown in Figure 3.

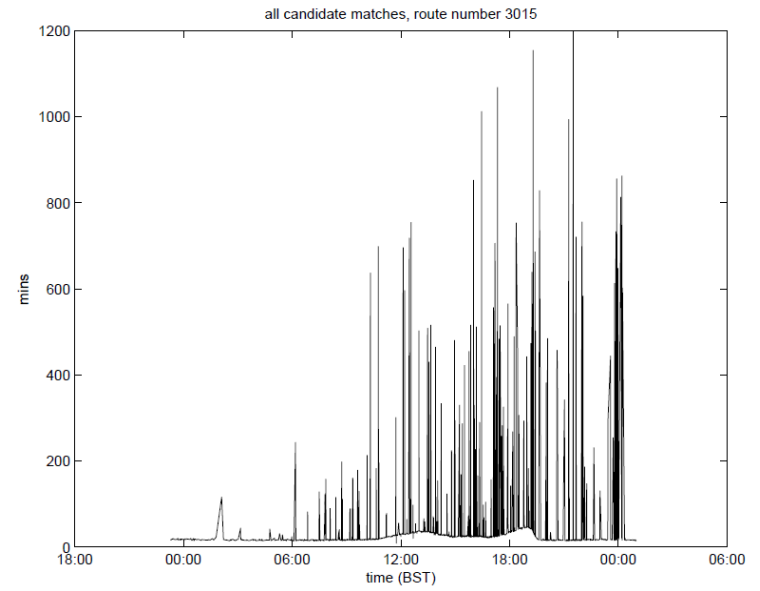


Figure 3: All matches, including many spurious ones, returned by the SQL query.

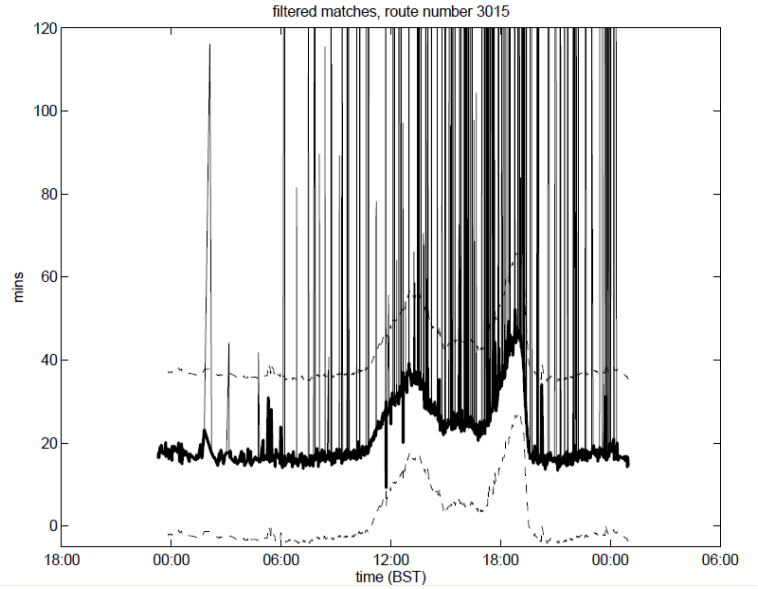


Figure 4: Accepted matches after filtering out the spurious ones.

A form of tracking is then used to sequentially filter out the spurious matches, whose output is shown in Figure 4. The thick line shows the accepted matches; dotted lines show the acceptance window. (Note change of *y*-axis scale from previous figure.)

The algorithm maintains an exponentially weighted moving average journey time throughout the day,

$$\hat{x}(t) = \lambda y(t') + (1-\lambda)\hat{x}(t-1), t \in T'$$

$$\hat{x}(t) = \hat{x}(t-1), t \notin T'$$

where *t'* ranges over all accepted match times, *T'*. This quantity is a point estimate of a hidden variable *x(t)*, the population duration at each match time, which is likely to change considerably during rush hours and quiet periods. The algorithm also maintains an acceptance window *σ(t)*. At each candidate match *y(t)*, we assume that

$$t \in T' \Leftarrow |y(t) - \hat{x}(t-1)| \leq \sigma(t)$$

This process may be viewed as a form of online EM algorithm (Dempster et al., 1977) which, for each point in sequence, alternates between making a hard classification (E step; assigning *t* to *T'* or $\bar{T}'$) and a parameter update (M step; updating $\hat{x}$).

As there are few spurious matches at the start of a day, we make an initial estimate of journey time at midnight from the mean of the first ten matches,

$$x(0) = \frac{1}{N}\sum_{1}^{10} y(t).$$

To estimate a useful value of σ, it is noted that the histogram of journey times typically consists of a relatively dense Gaussian-like region of good matches, followed by a much sparser and less populated region of high-duration spurious matches. Taking the median of this distribution gives a point somewhere in the Gaussian, and we take its value as an initial window width (so σ is half of this median).

To deal with unusual data cases (e.g. broken cameras, lane and road closures) we also impose hard maxima and minima on the tracked duration and window, $0 < \hat{x} < 180$mins and $15\text{mins} < \sigma < 30$mins.

Iterative lostness detection and re-tracking

The above algorithm proved sufficient for use on the calibration data, as it contained relatively few spurious matches due to its reports of raw plate numbers. The NTCC data, consisting of 24-bit hashes with character merging, gave rise to many more spurious matches which on occasion could result in the catastrophic loss of tracking, i.e.,

$$|\hat{x}(t) - x(t)| \gg \sigma(t).$$

as illustrated in Figure 5. During lostness, the acceptance corridor moves away from the bulk the candidate matches, resulting in a reduced density of matches over the day. Due to the hard limits imposed, tracking is always regained at some later stage in our data, so lostness has only been noted to occur during the rush-hour and similar rapid peaks, when the journey time increases faster than the size of the acceptance window.

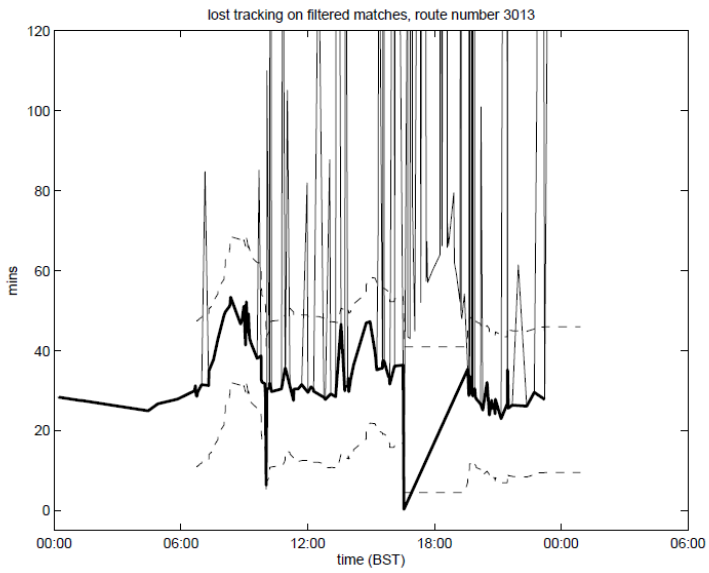


Figure 5: Loss of tracking.

To deal with this issue, up to five iterations of a lostness detection and restoration process are performed.

Assuming that lostness only occurs at peaks, and that peaks only occur between 7am and 7pm, we build a histogram of matches over this time period, having 2-hour bins. During lostness there are zero or very few (false) matches, so we test for bins in the histogram where the number of matches falls below ten per cent of its median. If this condition occurs, we re-run the tracking, using a widened acceptance window, $\sigma \leftarrow 2\sigma$, for the duration of the lost bin. In most cases this is sufficient to restore tracking (at the cost of accepting a small number of false positives); in more difficult cases it restores tracking for a period, gets lost again, and further iterations are needed to complete the tracking. Figure 6 shows an iteratively recovered version of Figure 5.

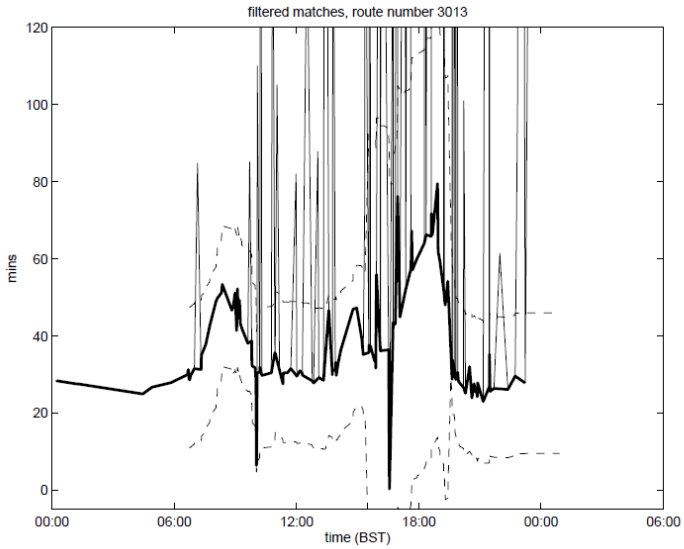


Figure 6: Recovery by detecting lostness then widening the acceptance window.

Calibration Data

The aim was to find how the proportion of routes detected by the NTCC cameras varies as a function of the lane configurations and of the flow. To do this, ANPR cameras were placed on all lanes of 12 calibration routes - selected to represent most typical routes - for one day. In particular these included motorway to motorway, dual carriageway to dual carriageway, dual carriageway to motorway and vice versa. (The calibration routes were all between Kingston, Guildford and the two M25 junctions near to them.)

The matches found by the subset of cameras in each origin-destination configuration of each calibration route were compared to the matches found using all cameras in that route. This enables estimation of the proportion of detections given by each origin-destination configuration, as a function of MIDAS flow. For example, one origin-destination configuration is the case of an origin on a 3-lane motorway with cameras on lanes 1 and 3 and a destination on a dual carriageway with a camera in lane 1. As MIDAS flow (measured at the origin at the journey start time) changes, so does the proportion of journeys along this route that are picked up by the particular cameras in the configuration.

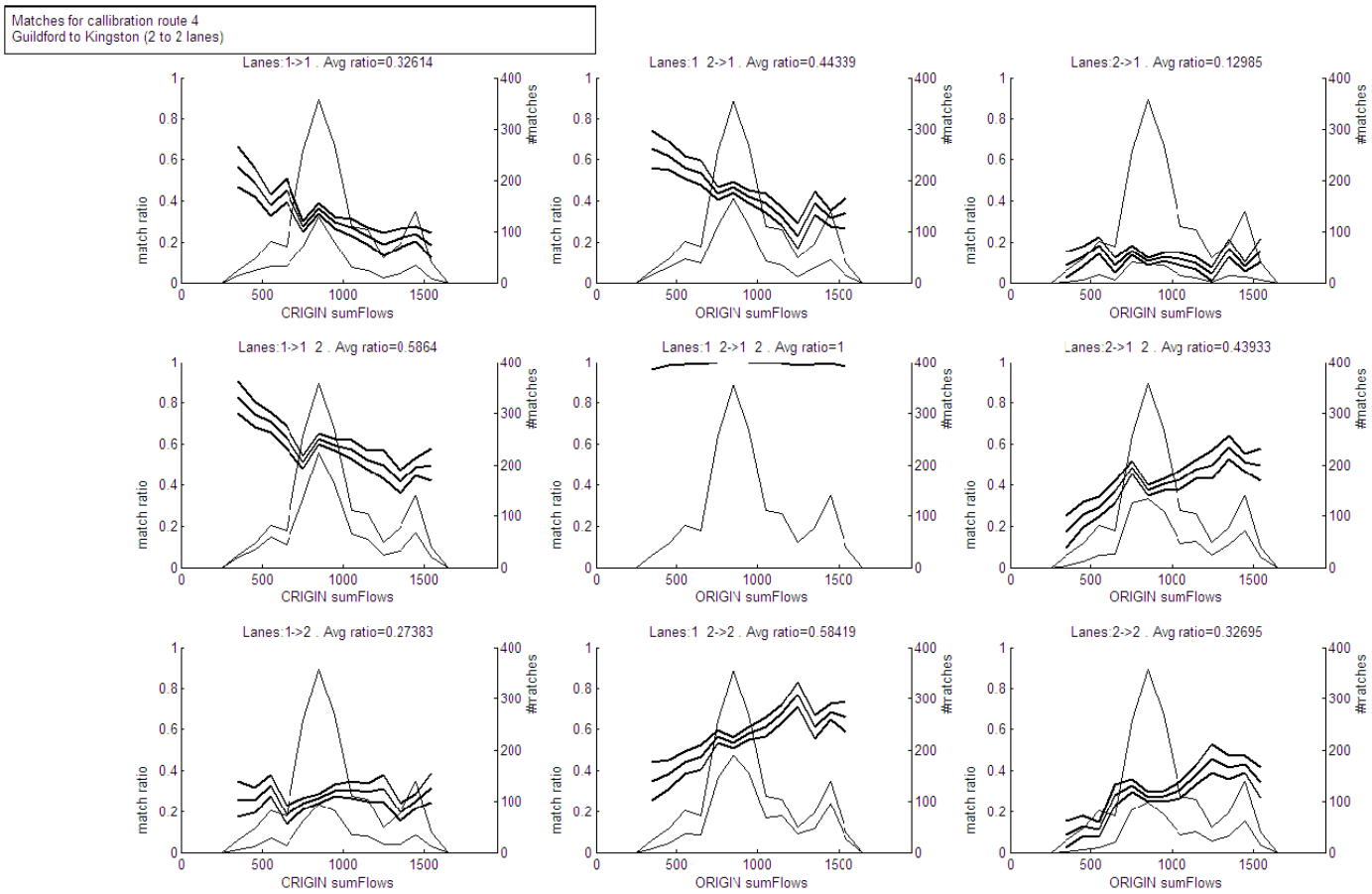


Figure 7: Detection ratios by flow for a single calibration route.

Figure 7 shows the complete set of origin-destination configurations for the simplest case, a dual-carriageway to dual-carriageway route. The thick lines show mean belief and one standard deviation confidence intervals. The thin lines show the number of matches in each flow bin, which affect the confidence intervals. There are nine configurations as both the origin and destination may have cameras in lane one, two or both. The plots show the number of matches detected by all cameras (upper thin plot) and by the configuration cameras (lower thin plot) as a function of the MIDAS flow, in a histogram with bins of 100 vehicles per hour. They also show an estimate of the ratio between these matches, with error bounds (thick lines).

For a single calibration route on a single day, the ratio θ is estimated as follows. We assume that the flow bins are mutually independent, and that each flow bin has a Beta distribution over our belief in its ratio,

$$P(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}.$$

We begin with a flat (Gelman et al., 2004) prior, ($\alpha = 1$, $\beta = 1$) over its matching ratio θ. We assume that the total (all lanes) number of matches that the number of matches observed is an exact estimate of the population of matches at that flow, N, and that the number of matches observed by the configuration cameras, k is a sample from the number that could have been observed. Under these assumptions, k is drawn from a Binomial distribution with unknown parameter θ. i.e. the situation is identical to estimating the heads-probability θ of a weighted coin, from N

observations, k of which are heads. On observing N and k, the Beta posterior is updated by:

$$\alpha \leftarrow \alpha + k,$$

$$\beta \leftarrow \beta + N - k.$$

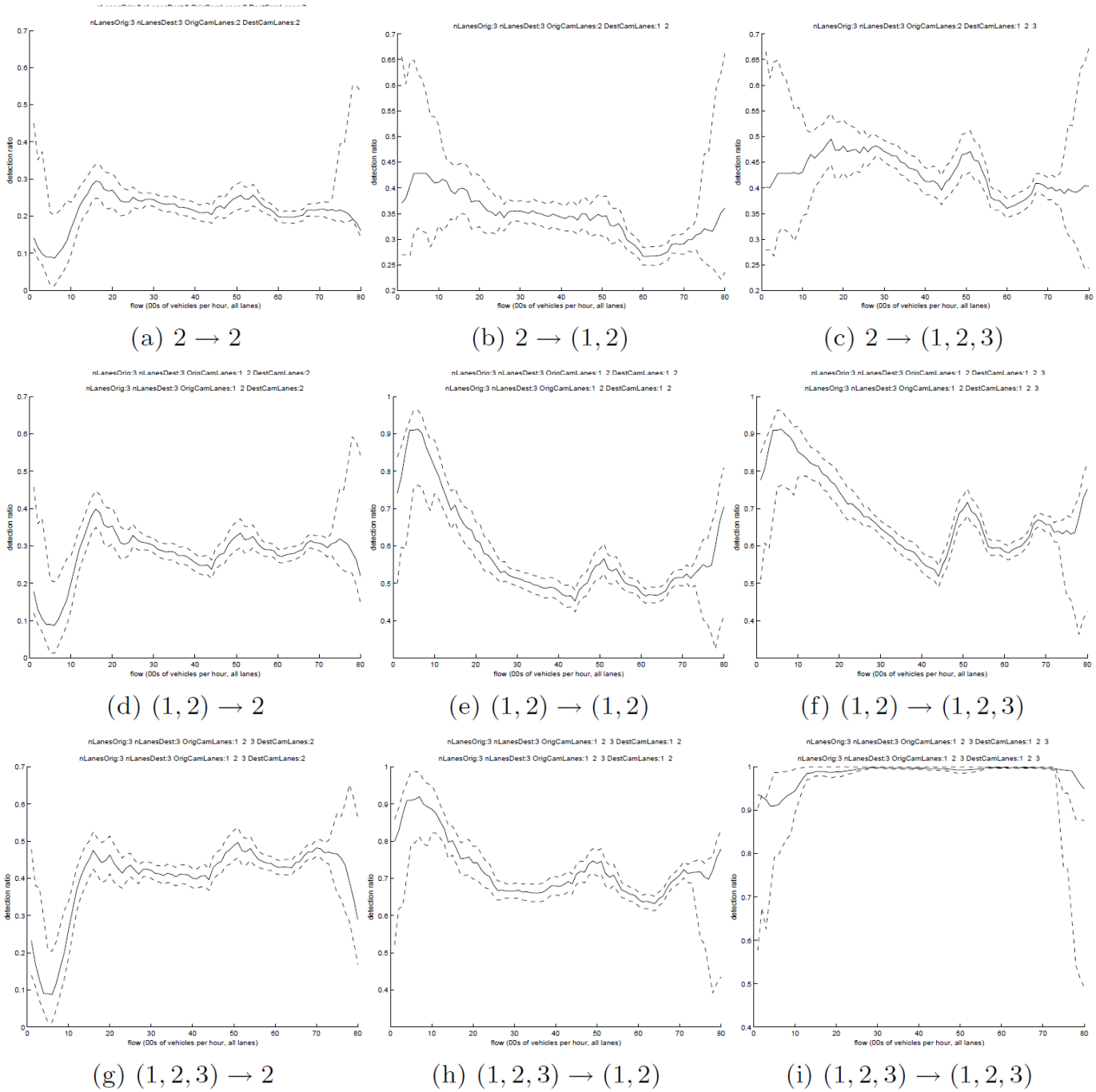
Furthermore, information may be fused from all routes sharing configurations into a single posterior for each configuration and each flow bin, using the same update equation. For example, two routes between three-lane motorways with cameras at lanes 1 and 2 at the origin and lane 2 at the destination. Thus for each origin-destination configuration, we may infer the match ratio as a function of flow. Such inferred functions are presented in Figure 8 for the most common NTCC configurations between 3-lane motorways, and form the principal empirical result of this exercise. As well as providing calibration for the subsequent analysis of NTCC data, they also contain information about how traffic adapts across lanes according to the overall flow. In addition to the ratios in Figure 8, many more ratios have been computed for routes between pairs of roads of different lane numbers and camera configurations, using the same method.

Bulk Data

The reason for estimating match ratios is to provide corrections to aid in estimating the number of origin-destinations routes within the existing NTCC ANPR network. This network consists of 24-bit plate-hashing cameras, positioned at sites in lane configurations. The calibration study suggests how to correct the number of matches on routes, from the sample of matches detected by the existing lane configurations, to form estimates of the actual number of journeys made. The iterative matching algorithm described previously was used to filter the matches made between the origin and destination lane configurations, and the number of journeys at each 5 minute time bin throughout the day was estimated by applying the following adjustments. First, the number of matches in each time bin was divided by the calibration match ratio for origin flow at the start of the journey. Second, it was multiplied by a constant factor of (1/0.75) to account for the ANPR detection rate of individual cameras. (This factor was found to vary somewhat as a function of MIDAS flow, see Figure 9, due to the cameras' processing becoming overloaded by high traffic volumes. However this has been ignored in the present study.) Finally, some NTCC sites monitor roads having $N>3$ lanes, for which we have no calibration data. These cases are treated as three-lane roads, and then their number of matches is crudely multiplied by $N/3$.

Profiles

ANPR data were made available by NTCC for 108 routes, for the eight weeks commencing 22 June, 2009, a dataset some 8Gb in size. The matching and filtering algorithm required about three hours to process this data on an Intel Centrino Duo machine, with the majority of time being spent on inserting the results from Matlab into SQL server via ODBC.



(a) $2 \rightarrow 2$ (b) $2 \rightarrow (1,2)$ (c) $2 \rightarrow (1,2,3)$

(d) $(1,2) \rightarrow 2$ (e) $(1,2) \rightarrow (1,2)$ (f) $(1,2) \rightarrow (1,2,3)$

(g) $(1,2,3) \rightarrow 2$ (h) $(1,2,3) \rightarrow (1,2)$ (i) $(1,2,3) \rightarrow (1,2,3)$

Figure 8: Journey detection rates for common lane configuration pairs on three-lane motorway to three-lane motorway routes.

The final objective was to develop origin-destination route profiles for various day types, so journeys from multiple days of the same type have been averaged out, smoothed with a 15-point convolution, and their number over time of day tabulated for subsequent use with the interactive visualisation tool. Figure 10 shows example route profiles which have been produced by this method, and which show rush hour peaks in the morning and evening.

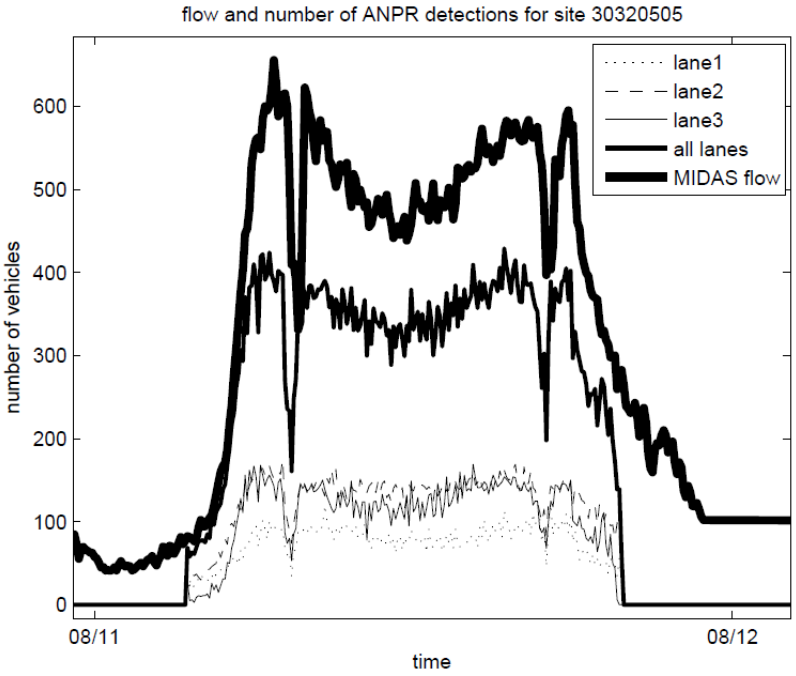


Figure 9: Number of ANPR detections in each lane, and all lanes, compared against MIDAS flow data.

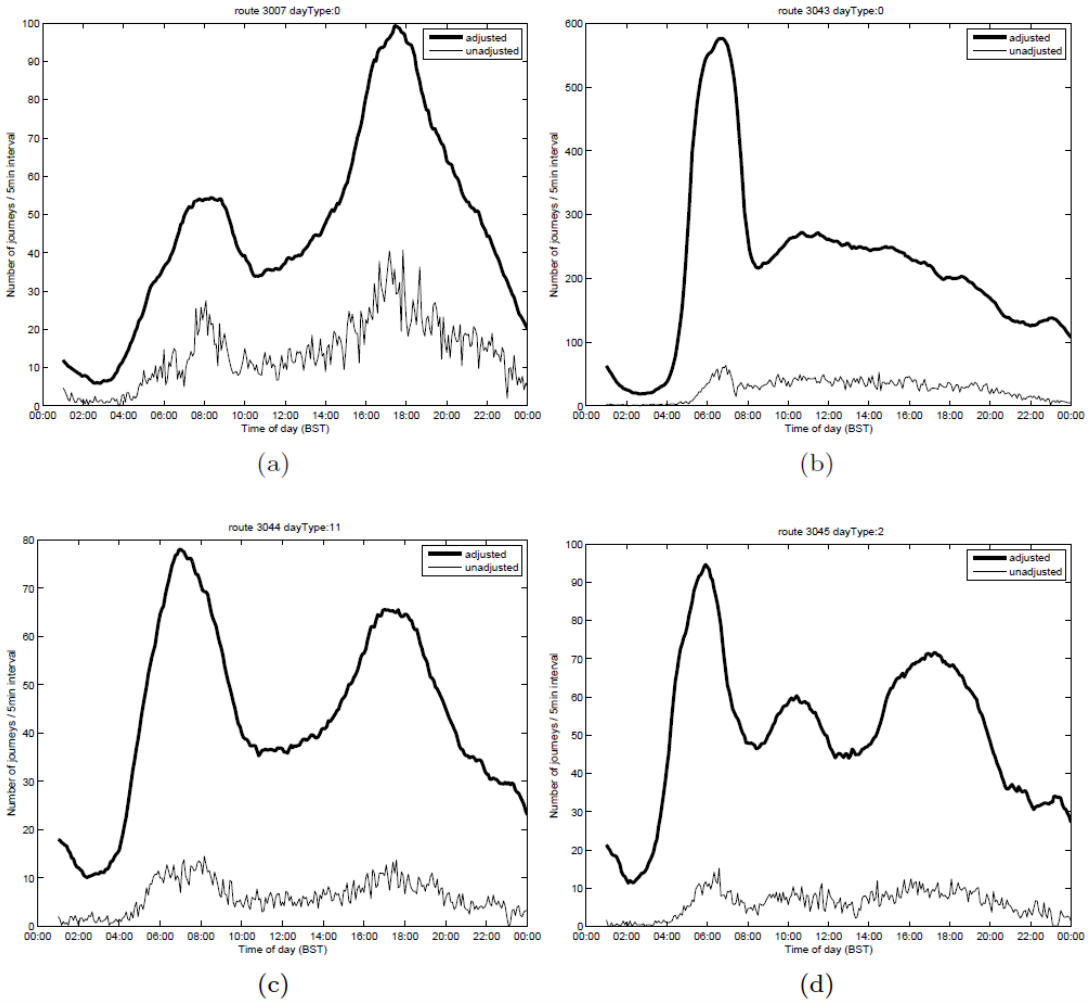


Figure 10: Typical origin-destination profiles.

The unadjusted data shown in these graphs represents the actual matches detected by the NTCC cameras monitoring a subset of lanes. The adjusted data is the result of applying the calibration characteristics for the appropriate start and end lane configurations which have been indexed according to the flow at the appropriate time of day. Such flows have been found from flow profile data supplied by the NTCC for the network links on which the cameras at the origins of each route are situated. The adjusted data therefore represents an estimate of the actual number of O-D movements which has been derived from a combination of NTCC camera observations, flow profiles and calibration characteristics.

Other than where NTCC tag data was not available (for example due to ANPR sites being out of service) a complete set of profiles has been produced for each O-D route and each day type. A graphical user interface for selecting and viewing the profiles has been constructed and is illustrated in Figure 11.

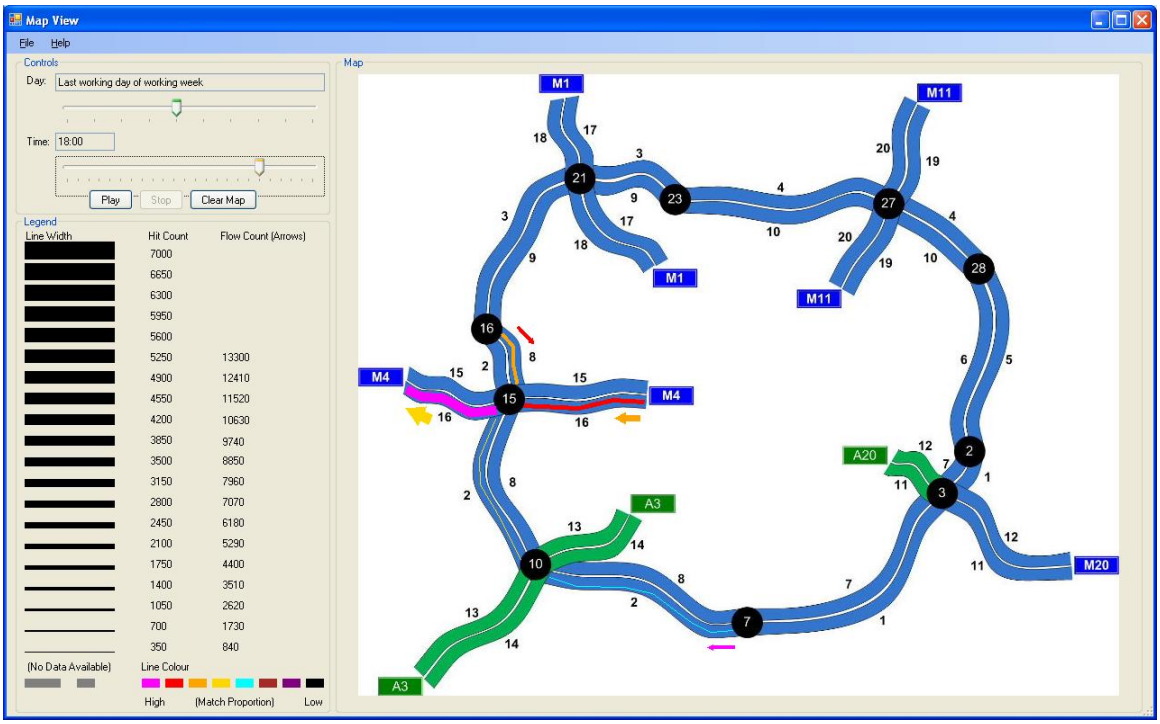


Figure 11: Screenshot from O-D Profile Graphical Visualisation tool.

Discussion

We have presented methods for rapid (3 hours to process 8Gb of data) matching and filtering of noisy ANPR origin-destination data, and for calibrating partial lane configuration matches, as a function of MIDAS traffic flow, to estimate total journey number profiles for routes over time and type of days.

The methods could be further improved by more detailed data modelling. Calibration data could be obtained for roads with four and more lanes; and anomalies such as

lane closures and hard-shoulder running could be accounted for. The assumption of independence of match ratios in neighbouring flow bins was pessimistic, though heuristically smoothed out later in the profile process. This combination of independence and smoothing could perhaps be replaced by (or shown to be equivalent to) a Gaussian Process (Rasmussen and Williams, 2005) which explicitly models local correlations. Finally we noted an effect of MIDAS flow on the detection rates of individual cameras which could be taken into account.

The output profiles could for example be used by the Highways Agency to identify particular popular routes between two arteries of the M25 and hence make recommendations about constructing new direct links (roads or public transport) between their origins and destinations, reducing congestion on the orbital.

References

A.P.Dempster, N.M. Laird, and D.B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.

A. Gelman, J. B. Carlin, H. S. Stern and D. R. Rubin. *Bayesian Data Analysis,* CRC Press, 2004.

C.E. Rasmussen and C.Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Acknowledgements

The co-operation and assistance of the National Traffic Control Centre in the provision of bulk data is gratefully acknowledged.

Learning in a unitary coherent hippocampus

Charles Fox[1] and Tony Prescott[1]

Adaptive Behaviour Research Group, University of Sheffield S10 2TN, UK

Abstract. A previous paper [2] presented a model (*UCPF-HC*) of the hippocampus as a unitary coherent particle filter, which combines the classical hippocampal roles of associative memory and spatial navigation, using a Bayesian filter framework. The present paper extends this model to include online learning of connections to and from the CA3 region. Learning in the extended neural network is equivalent to learning in a temporal restricted Boltzmann machine under certain assumptions about neuromodulatory effects on connectivity and learning during theta cycles, which suggest detailed neural mappings for Bayesian inference and learning within sub-stages of a theta cycle. After-depolarisations (ADP) are hypothesised to play a novel role to enable reuse of recurrent prior information across sub-stages of theta.

1 Introduction

Anatomy. The principal input structures of the hippocampus are the superficial layers of Entorhinal Cortex (ECs). ECs projects to Dentate Gyrus (DG) which is believed to sparsify the encoding of ECs. Both ECs and DG project to CA3, which also receives strong recurrent connections that are disabled [3] by septal ACh. CA3 and ECs project to CA1, which in turn projects to the deep layers of Entorhinal cortex (ECd), closing a loop if ECd sends information back to ECs. ECs, CA1 and ECd outputs appear to share a coding scheme, as evidenced by one-to-one topographic projections. In contrast, DG and CA3 outputs are thought to work in a second basis or latent space. In a second loop, ECs and CA1 both project to Subiculum (Sub), which projects to the midbrain Septum (Sep) via fornix. Septal ACh and GABA fibres project back to all parts of hippocampus.

UCPF-HC model. A previous paper [2], mapped this hippocampal circuit onto a modified Temporal Restricted Boltzmann machine (TRBM, [8]). The TRBM assumes Boolean observation vectors (including a bias node), z'; Boolean hidden state vectors (including a bias node), x'; weight matrices $W_{x'z'}$ and $W_{x'x'}$, and specifies joints,

$$P(x_t, x_{t-1}, z_t) = \frac{1}{Z} \exp \sum_t (-x'_t W_{x'x'} x'_{t-1} - x'_t W_{x'z'} z'_t). \tag{1}$$

Unlike the standard TRBM, UCPF-HC uses the following deterministic update to obtain maximum *a posteriori* estimates:

$$\hat{x}_t \leftarrow \arg\max P(x_t | \hat{x}_{t-1}, z_t) = \{\hat{x}_t(i) = (P(x_t(i) | \hat{x}_{t-1}, z_t) > \frac{1}{2})\}_i \tag{2}$$

which is the zero-temperature limit of an annealed sequential Gibbs sampler.

The noisy inputs $z_t = y_t + \epsilon_t$ are mapped to the combined ECs and DG, where the DG activations are functions of the ECs activations, $z_t = (ECs_t, DG_t(ECs_t))$. CA3 is mapped to the hidden state, x_t. CA1 performs a partial decoding into the DG basis. Finally the estimated de-noised output is mapped to ECd, $\hat{y}_t = ECd_t$. Each neural population is a Boolean vector at each discrete time step t.

A major problem with UCPF-HC tracking is tracking loss, as it approximates the whole posterior with a single sample. To deal with this, performance of the filter is monitored to heuristically detect when tracking is lost – by thresholding a moving average of discrepancy between observed and denoised sensors – then the priors are disabled when lostness is detected. In UCPF-HC, the Subiculum-Septum circuit performs this monitoring. Sub then compares the partially decoded CA1 information against the original ECs input, receiving one-to-one connections from both regions. If they differ for an extended period of time, this indicates loss of tracking. *Tonic* cholinergic projections from Sep, activated via Sub, are well-placed to disable the CA3 priors when lostness occurs, as they are known [3] to disable the recurrent connections in CA3.

The present study presents a new version of UCPF-HC, using the plus maze environment detailed in [2], and extendend with ADP to perform learning in CA3. The plus-maze consists of 13 discrete locations as shown in fig. 4(a). The agent sees unique visual markers if facing two of the arms; it also has touch sensors to report walls to its immediate left, right and front. The original UCPF-HC model included mechanisms to perform path integration in the grid and heading cells using odometry and denoised ECd states – to simplify the present study we assume that grid and heading cells give uncorrelated noisy (Global Positioning System style) estimates of location and orientation, as would be obtained if the UCPF-HC's outputs were always perfect or known to be lost but the odometry was noisy.

ADP Physiology. CA3 pyramidal cells [1] exhibit a single cell short-term memory effect called after-depolarisation (ADP), illustrated in fig. 4(b). A spike (1) in membrane potential, V, is followed by a fast after-hyper-polarisation (AHP, 2), then an after-depolarisation (ADP, 3) and a second, slower AHP (4), before returning to its resting potential (5). (See [7] sections 5.2.5 and 5.3.5 for a detailed review.) ADP has previously been suggested [5] as a basis for multiplexed short-term memories in hippocampus, enabling around seven patterns to be stored simultaneously by re-activating themselves after other patterns, using the ADP gain plus an external excitatory oscillator. We will suggest a related but novel role for ADPs, allowing priors to be restored during separate wake and sleep cycles [4] in a temporal network. ADP is dependent on the presence of ACh or 5HT [6], and septal *phasic* ACh has been suggested to play a role in the hippocampal theta rhythm [3].

2 On-line learning for the UCPF-HC model

The previous version [2] of UCPF-HC did not perform any realistic learning. CA3 cell semantics were specified by hand – for example cells were specified to respond to conjunctions of places, headings and light states. Ideal CA3 responses were computed offline, from these hand-set specifications and ground truth data sets, then weights for

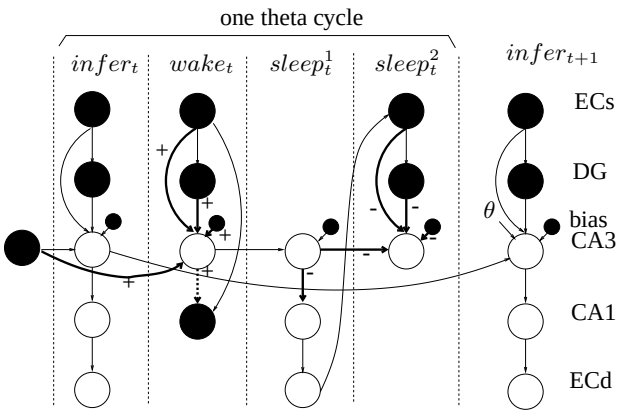


Fig. 1. The four substeps, $infer, wake, sleep^1, sleep^2$ within one theta cycle in the learning neural network model. Circles denote populations of neurons. Arrows indicate fully connected neural network projections (this is *not* a Bayesian network diagram). Thick arrows indicate connections whose weights are updated with Hebbian (+) and anti-Hebbian (-) rules. Dotted arrows indicate where learning occurs but no information is projected (i.e. when the child population is clamped from elsewhere). Filled-in nodes are fixed values at each substep, unfilled nodes are to be computed. The bias population contains a single neuron which is always on, and abstracts the threshold values in CA3. In the first substep of the next cycle, $infer_{t+1}$, CA3 receives a θ signal which disconnects the recurrent connections and switches to ADP recurrent activation.

each input population to CA3, pop, were set using independent wake-sleep [4] updates,

$$\Delta w_{ij} = \alpha(\langle CA3_i pop_j \rangle_{\hat{P}(pop,CA3|b)} - \langle CA3_i pop_j \rangle_{P(pop,CA3|b)}) \tag{3}$$

where $\hat{P}$ is the empirical data distribution including the hand-set ideal hidden values; P is the model's generative distribution; and b is the set of hidden nodes biases, preset empirically to model priors on the handset semantics. This was not indented as a realistic learning model, rather just a computational method to set the weights. In particular the computation was greatly simplified by having access to ground-truth hidden states, which made the weights mutually independent given the bias. In reality the agent does not have access to ground truth hidden states – only to sensors.

We do not give the new model access to ideal CA3 states or hand-set their semantics – this time the semantics must be learned. The semantics of DG and CA1, and hence the weights $W_{EC \to DG}$ and $W_{CA1 \to EC}$ remain set by hand – we focus only on extending the model to learn all connections to and from CA3: namely $W_{ECs \to CA3}$, $W_{DG \to CA3}$, $W_{CA3 \to CA3}$ and $W_{CA3 \to CA1}$.

To simplify both the presentation and implementation of the learning model, we will first present the hippocampal learning algorithm for the UCPF-HC neural network as a *fait accompli*, then describe a graphical model simplification used in the implementation. The graphical model formulation also provides insight into the purposes of the neural network processes, which were in fact derived from the graphical model during development. The algorithm is based on the wake-sleep process but is now intended as a biological model.

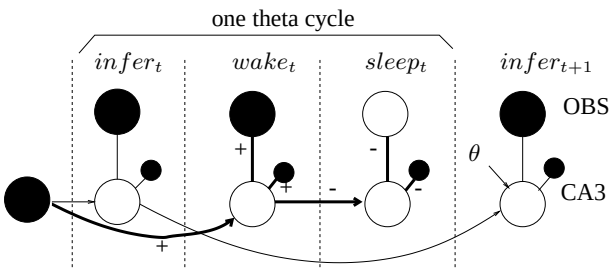


Fig. 2. The reduced TRBM model. This network is equivalent to the neural network, but includes undirected connections, and groups the information from EC,DG and CA1 into a single observable population, OBS.

2.1 Neural network model

The neural learning algorithm is based on the wake-sleep algorithm [4] and is illustrated in fig. 1. It assumes that for every discrete time step t there are four substeps, $infer_t, wake_t, sleep_t^1, sleep_t^2$ corresponding to different phases of one hippocampal theta cycle. These substeps have differing connectivity and learning dynamics, which might be controlled by neuromodulators during the theta cycle. The substeps occur sequentially. But importantly, CA3 activation during the $infer_t$ substep is required to directly influence CA3 at $wake_t$; and CA3 activation at $infer_t$ is required to directly influence CA3 at $infer_{t+1}$; as shown by the arrows in the figure. We tentatively suggest that ADP, discussed in section 1, might play a role in such temporally incontiguous transmission of information.

The $infer_t$ substep is identical to inference in the UCPF-HC model. ECs sensor data is observed; deterministic DG activations (via handset $W_{EC \to DG}$) are computed, and thus act as observations too. We assume that the state of CA3 at $infer_{t-1}$ was inferred exactly and correctly by the UCPF, and is available as an input via recurrent transmission weights $W_{CA3 \to CA3}$. Using these inputs, CA3 is updated with a Gibbs sampling step at temperature $T = 0$. CA1 and ECd decode it to retrieve denoised sensor estimates.

In the $wake_t$ substep, the same input vector is maintained in EC and DG; and CA1 activation becomes clamped by a training signal from the ECs input. We assume that conjunctions of facts from ECs are represented perfectly in CA1 by this process, as in DG. We require a delayed copy of the recurrent CA3 input from $infer_{t-1}$ as was received in the $infer_t$ step – *not* a recurrent CA3 input from $infer_t$ – as the recurrent input to CA3 in $wake_t$. CA3 is resampled at $T = 1$ and Hebbian learning is performed at all synapses to and from CA3. In the $sleep_t^1$ substep, the recurrent CA3 connections are used directly so that CA3's state now is influenced by its previous state, $wake_t$. Its connections from ECs and DG are made ineffectual. CA3 is sampled again at temperature $T = 1$, then CA1 and ECd is decoded from it. In $sleep_t^2$ we assume that ECs becomes clamped to the ECd result – feeding back the denoised output into the input. CA3 is resampled again at $T = 1$ and antihebbian learning is performed in all synapses to and from CA3. The theta cycle is now complete, and the next one begins at $infer_{t+1}$.

2.2 Reduced undirected model

We next explain why the neural network is equivalent to the reduced graphical model shown in fig. 2. It is a new variant of the temporal restricted Boltzmann machine [8].

DG consists entirely of cells whose receptive fields are copies or conjunctions of ECs fields. In the reduced model, we form a single population, OBS, which contains both DG and ECs cells. CA1 in the neural model consists of cells with identical fields to DG cells, which are thus also implicitly contained in OBS. The weights $W_{OBS-CA3}$ are undirected as in the TRBM, though the steps of learning them correspond to the steps learning the weights in the neural model.

In phase $infer_t$, CA3 is driven by inputs from EC and DG in the directed neural model, which is equivalent to the undirected link to the observed OBS population in the reduced model. (The bias link is also changed from directed to undirected in the reduced model – again this is an equivalence as the bias is always observed.) In this phase, the temperature is zero so the inferences are always the MAPs. This gives the best denosied estimate of the state of the world.

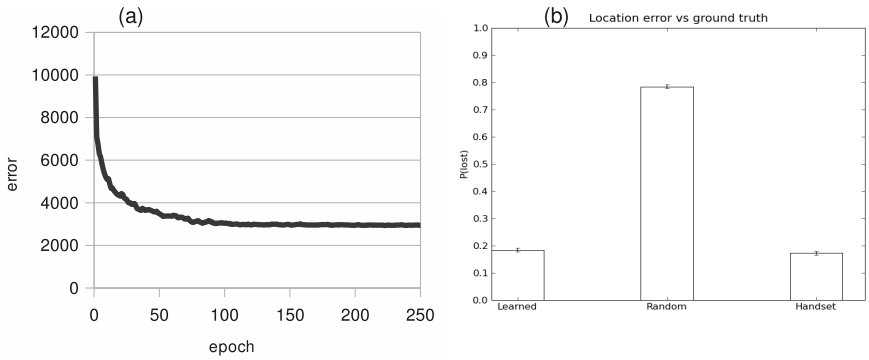


Fig. 3. (a) Training errors. Error is the sum of ECs-ECd discrepancies over all training data in each epoch. (b) Lostness probabilities in learned, random and handset-semantics weights. Error bars show one standard deviation of uncertainty about the population mean.

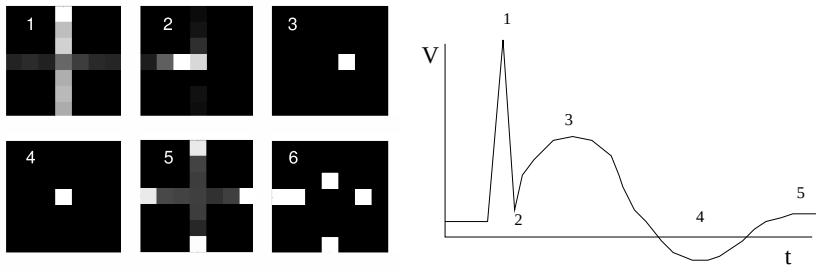
In phase $wake_t$, the drivers of CA3 are the same, but the temperature is $T = 1$. In the neural model, CA1 is clamped to EC, and Hebbian learning occurs in $W_{EC\to CA3}$, $W_{DG\to CA3}$ and $W_{CA3\to CA1}$. This is equivalent to clamping OBS again in the reduced model, and Hebbian learning on $W_{CA3-CA1}$. As in the neural model there is also Hebbian learning on the recurrent CA3 connections.

In phase $sleep_t$ of the undirected model, a CA3 sample is drawn conditioned on its recurrent state only. Then an OBS sample is drawn conditioned on CA3, and finally a new CA3 sample is drawn conditioned on its recurrents and on the OBS sample. Antihebbian learning is performed on all connections to CA3. This is equivalent to the process in the neural model's $sleep_t^1$ and $sleep_t^2$, and is a standard TRBM sleep step.

Phase $infer_{t+1}$ is the start of the next cycle, and like the neural model, requires historical CA3 input from $infer_t$, as might be obtained using ADP.

3 Results

We tested the learning algorithm in the plus-maze world (see [2]), using a path of 30,000 random walk steps. The path was replayed for several epochs until the weights converged. For computational simplicity, learning was performed used the equivalent reduced model, though inference was performed with the full neural model, sharing the learned weights. Python code for the simulation is available from the authors. There is some subtlety in handling learning for cases where the Sub-Sep lostness circuit is activated, which is detailed in the appendix. Fig. 5(a) shows the training errors during learning – using a learning rate of $\alpha = 0.001$ – most of the learning takes place in the first 10 epochs. As in the original [2] UCPF-HC model, the neural network is used to infer denoised ECd estimates of position and sensors. Fig. 5(b) shows the average rate of location errors using the learned weights, compared against the handset semantic of the original UCPF-HC model. A run with randomised, untrained weights is shown for comparison. Inspecting the receptive fields of CA3 cells learned by the training, we find cells in fig. 4(a) responding to individual places (3 and 4); regions around a place (2); the ends of the arms (5); and less well defined fields (1 and 6).



(a) Examples of learned CA3 receptive fields, over the plus maze.

(b) Typical membrane time course following a spike under ADP dynamics.

Fig. 4.

4 Discussion

We have presented a top-down mapping of a wake-sleep learning algorithm onto the biological hippocampal circuit and existing UCPF-HC model. The UCPF-HC model was extended by adding detailed substeps within theta cycles, which specify the connectivity and learning operations required by the algorithm, as biological hypotheses. Our revised model demonstrates a biological plausible online learning mechanism for CA3 pyramidal cells, and thus lends support to our general hypothesis that the hippocampal system may operate as a unitary coherent particle filter. This type of mapping necessarily makes strong predictions about what neurons would be required to do to implement the algorithm. In particular we have relied on specific timing features of ADP and on

ACh to switch between recurrent CA3 activation and ADP-based CA3 memories. It remains to be seen whether biological ADP and ACh are able to provide these functions.

The inference step was performed at zero temperature, separately from higher temperature wake and sleep steps. There are several possible variations on this theme. First, both wake and sleep could be performed at zero temperature, removing the need for a separate inference step, and resulting in a different type of optimisation during learning (minimising $KL[Q^T||P^T]$ rather than $KL[Q||P]$. In the limit $Q = P$ they would give the same result). Second, wake and sleep steps could be extended to run for several steps. This would result in longer sequences of uninterrupted tracking of observations, alternating with longer 'hallucinated' sequences of generated samples. The latter would resemble sequence replay and preplay known to take place in CA3.

Future work could implement the neural learning model directly, in place of the reduced simplification. It could also consider memories of the agent's own actions as way to increase the predictability of plus maze sequences. Finally the agent should iteratively estimate the amount of error in its own location estimates rather than rely on the artificial noisy GPS assumption used in this proof-of-concept implementation.

References

1. M. Ceaser, D.A. Brown, B.H. Gahwiler, and T. Knopfel. Characterization of a calcium-dependent current generating a slow afterdepolarization of CA3 pyramidal cells in rat hippocampal cultures. *Eur. J. Neuroscience*, 5:560–569, 1993.
2. C.Fox and T.Prescott. Hippocampus as unitary coherent particle filter. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2010.
3. M.E. Hasselmo, E. Schnell, and E. Barkai. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience*, 15:5249–62, 1995.
4. G.E. Hinton, P. Dayan, B.J. Frey, and R.M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.
5. O. Jensen, M.A.P. Idiart, and J.E. Lisman. Physiologically realistic formation of autoassociative memory in networks with theta/gamma oscillations: role of fast NMDA channels. *Learning and Memory*, 3:243–256, 1996.
6. J. Lisman and M.A.P. Idiart. Storage of 7+/-2 short-term memories in oscillatory subcycles. *Science*, 267:1512–1515, 1995.
7. N. Spruston and C. McBain. Structure and functional properties of hippocampal neurons. In P. Andersen, R. Morris, D. Amaral, T. Bliss, and J. O'Keefe, editors, *The Hippocampus Book*. Oxford University Press, 2007.
8. G.W. Taylor, G.E. Hinton, and S.T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *NIPS 19*. 2007.

Towards biomimetic vibrissal tactile sensing for robot exploration, navigation, and object recognition in hazardous environments

Tony Prescott[1], Martin Pearson[2], Charles Fox[1], Mat Evans[1], Ben Mitchinson[1], Sean Anderson[1], Tony Pipe[2]

1

[2]Bristol Robotics Laboratory

Abstract. Arrays of actuated, whisker-like tactile sensors could prove useful for the guidance and control of robots in hazardous environments, particularly where the effectiveness of conventional vision sensors is compromised. This paper summarises recent research on biomimetic vibrissal sensing arrays for mobile robots, focusing on the design of the SCRATCHbot whiskered robot platform, and on progress towards algorithms for tactile object detection and recognition, and for robot navigation using vibrissal signals.

In order to cope with nocturnal or poorly-lit environments mammals have evolved a range of non-visual sensory capacities many of which have not been successfully replicated in robots. One such capacity is the tactile hair (vibrissal) sensory system [1, 2]. Tactile hairs are found in all mammals, except for man, and are highly developed in many rodent species (such as rats and mice) and in a variety of aquatic mammals such as seal, walruses, and manatees. Research interest has mainly focused on the facial vibrissae, or *whiskers*. In rodents (such as the rat shown in figure 1) there are two arrays of long facial whiskers surrounding the snout, that endow these animals with the capacity for short-range, but high-speed tactile object detection and recognition. Rodents can also use their vibrissae to navigate and locomote on difficult terrain in the absence of light. A similar sensory capacity in mobile robots could lead to increased versatility and performance in hazardous environments, such as smoke- or dust-filled buildings, or where covert operation in darkness is required. Borrowing inspiration from marine mammals, similar systems might also find applications in aquatic environments particularly in muddy or turgid water.

Bristol Robotics Laboratory, in partnership with the Active Touch Laboratory in Sheffield, have been working on the development of artificial vibrissal systems for robots since 2003. Our latest mobile robot platform, *SCRATCHbot* (see figure 2), has bilateral arrays of 3x3 artificial vibrissae with multiple degrees of freedom of control in the head and whisker-positioning systems, that allow us to orient the vibrissae towards nearby objects or surfaces, and obtain good information sampling rates by rapid forward and backward movement of the vibrissal shafts ("whisking"). We are also developing a modular, actuated artificial vibrissal sensor that can be assembled into different multi-whisker configurations.

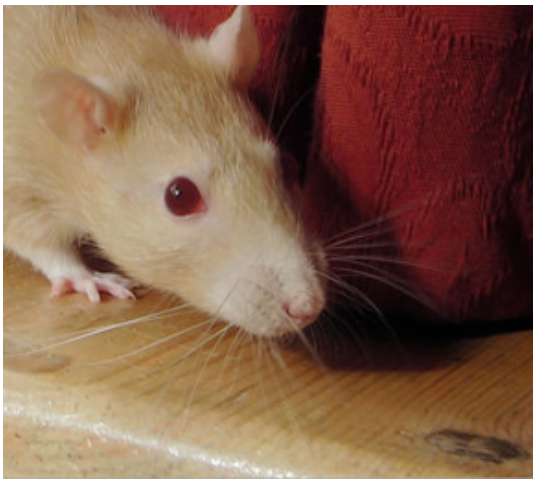


Figure 1. The vibrissal system of the rat consists of bilateral arrays of 30+ actuated whiskers.

Current work is directed towards testing the utility of vibrissal sensing for tactile-guided exploration and mapping of indoor environments. A longer-term goal is to show that these systems could be useful for the guidance of robot locomotion and control in less-constrained and higher-risk settings. In the remainder of this paper we briefly describe the current status of three main strands of work: (i) the design and

function of the SCRATCHbot whiskered robot, (ii) the development of pattern recognition systems for vibrissal signals, and (iii) initial progress towards a biomimetic tactile spatial navigation system.

SCRATCHbot (Spatial Cognition and Representation through Active TouCH robot)
The specification for our current whiskered robot platform was developed through experiments and observations of laboratory rats [1, 2, 3] aimed at understanding how these animals control the movements of their whiskers, and at discovering what sorts of behavioural tasks vibrissal information may be useful for.

Two specific co-ordinated motor actions were identified as pre-requisites for effective active vibrissal sensing: (i) the generation of bouts of rapid whisker movement that can be mediated by environmental contact [2], and (ii) the ability to quickly and accurately reposition the head so as to orient the whisker array towards objects of interest. To support these behaviours (as well as others) the robot platform (see figure 2) was built from 3 main components: a head, on to which the whisker arrays would be mounted; a body, to carry the computing resources, locomotion and power supply; and a neck, to allow the head to be moved independently from the body. Each of these components are described in more detail next.

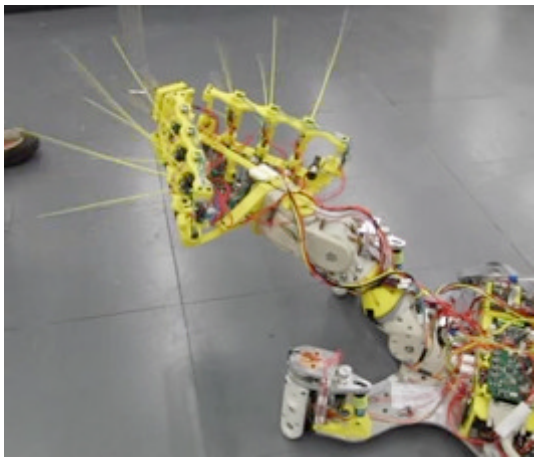


Figure 2. The SCRATCHbot robot platform has two 3x3 array of actuated whiskers on a 3-DOF head. Control uses biomimetic algorithms based on reverse-engineering of rat neural systems for vibrissal sensory processing and actuation.

Head
The head was designed to carry six independent columns of three whiskers, each driven in a single axis (anterior-posterior) by a small dc motor and gearbox. The whisker columns are arranged into two arrays of three columns, each array projecting from opposing sides of the head chassis, and coupled for a second axis of rotation (array tilt). A third, non-actuated, array of nine short whiskers is mounted between the bi-lateral active arrays. The non-actuated array is analogous to the shorter, but high density, *microvibrissal* array found on the chin and lips of the rat, and the actuated bilateral arrays to the longer *macrovibrissae* which emerge from the cheeks (mystacial pads) of the animal. The macrovibrissae on the platform were built from ABS plastic using a rapid prototyping machine. The cross-sections of the whiskers are tapered toward the tip and their lengths (160 - 220mm) are approximately four times larger than the macrovibrissae of a typical adult rat. To measure deflections of the whisker shaft caused by contact with the environment a small magnet is bonded to the base of each whisker and a tri-axis Hall effect sensor IC used to sample the displacements of the magnet in two axes. To maintain the pose of each whisker, and to return it to its resting angle after deflection, the whisker base is mounted into a small plug of polyurethane rubber. The non-actuated whiskers (microvibrissae) have the same transduction technology and polymer return mechanism, however, the whisker shafts are shorter (approximately 80mm) and mounted into a single casting of polyurethane.

Each of the bilateral arrays of macrovibrissae have an associated microcontroller to sample all nine whiskers and to control the rotation of the three columns and the tilt angle of the array (using separate software PID position controllers for each of these). The sensory information from each array is passed to the main computing resources of the platform, located on the body, via serial buses through which are returned the desired angles of each of the controlled axes. Another microcontroller marshals the sensory information from the microvibrissal array and a third bridges output from a three-axis accelerometer, mounted on the tip of the snout, to the platform-wide CAN bus.

Neck
The neck component was designed and built by an external robotics sub-contractor [4] to enable the head to be moved with three degrees of freedom—elevation, pitch and yaw. Each axis is actuated by a brush-less dc motor and harmonic

drive gearbox, and is controlled using a micro-controller based PID position controller coupled to the platform CAN bus. Desired angles for each axis are broadcast to the CAN bus at regular intervals from the central computing resources of the platform.

Body

The main chassis of the platform is a single sheet of aluminium onto which the motor drive units and neck are mounted. Each of the three motor drive units are independently controlled and can steer through +/-90 degrees from straight ahead. The desired speed and direction of each motor drive unit are broadcast onto the platform wide CAN bus at regular intervals, and the power supply for the platform can be either from Lithium Polymer batteries or 240VAC mains. The central computing resources consist of a PC-104+ reconfigurable computing platform, composed of a Single Board Computer (SBC) and a closely coupled array of FPGAs for hardware accelerated processing.

Processing architecture

The processing architecture implemented on the robot takes inspiration from the neural pathways identified in the rat whisker sensory system [1, 2, 5]. Neural structures such as the *trigeminal sensory complex*, *superior colliculus* and *basal ganglia* are modelled and developed in software, at various levels of modelling abstraction, and integrated into a unified system for testing using the *BRain And Head Modelling System* (BRAHMS) execution framework [5, 6]. A BRAHMS process within this architecture acts as the interface to the hardware of the robot through which sensory information is made available and motor commands are sent. Function calls from the platform API within this process communicate with the FPGA bridge and thus act as the real-time regulator to the software BRAHMS system. To allow independent development of robot hardware and software neural models, a platform simulator has been written which can be inserted into the BRAHMS system in place of the robot interface. Figure 3 is a block diagram of the components that make up the current processing architecture. The arrows indicate control loops within our model whisker sensory system [5] which correspond to current understanding of the real sensory system [1, 2]. The blue arrow indicates the control of whisking whilst the green arrow shows the control loop for the 'orient to point of contact' behaviour described in the next section. The orange arrow highlights one of the directions for future work, namely, cortical modelling and the development of tactile spatial mapping.

Orient to point of contact

Through observation of rat behaviour [1, 3] the tendency of rats to direct their snout and micro-vibrissal array toward unexpected macrovibrissal contacts was chosen as a behaviour that was suitable for investigation by physical modelling. Our control system implements the hypothesis that a region of the mammalian brain known as the *superior colliculus* (SC) is used by the rat to control orienting to tactile stimuli [7]. A model SC was designed, implemented in software, and integrated into the BRAHMS processing framework for demonstration on the SCRATCHbot platform. This model integrates whisker deflection information (from the Hall effect sensors) with shaft encoding of whisker column angles in order to map environmental contacts onto a 3-D representation of the space surrounding the robot's head. The most salient contact point then primes a request for an appropriate series of orienting motor commands that move the tip of the snout to that position. The request to perform orienting competes with other salient behaviours for control of the motor plant. This competition is resolved using an action selection mechanism modelled on a group of brain structures known as the *basal ganglia* that are thought to implement a form of centralised switch [8].

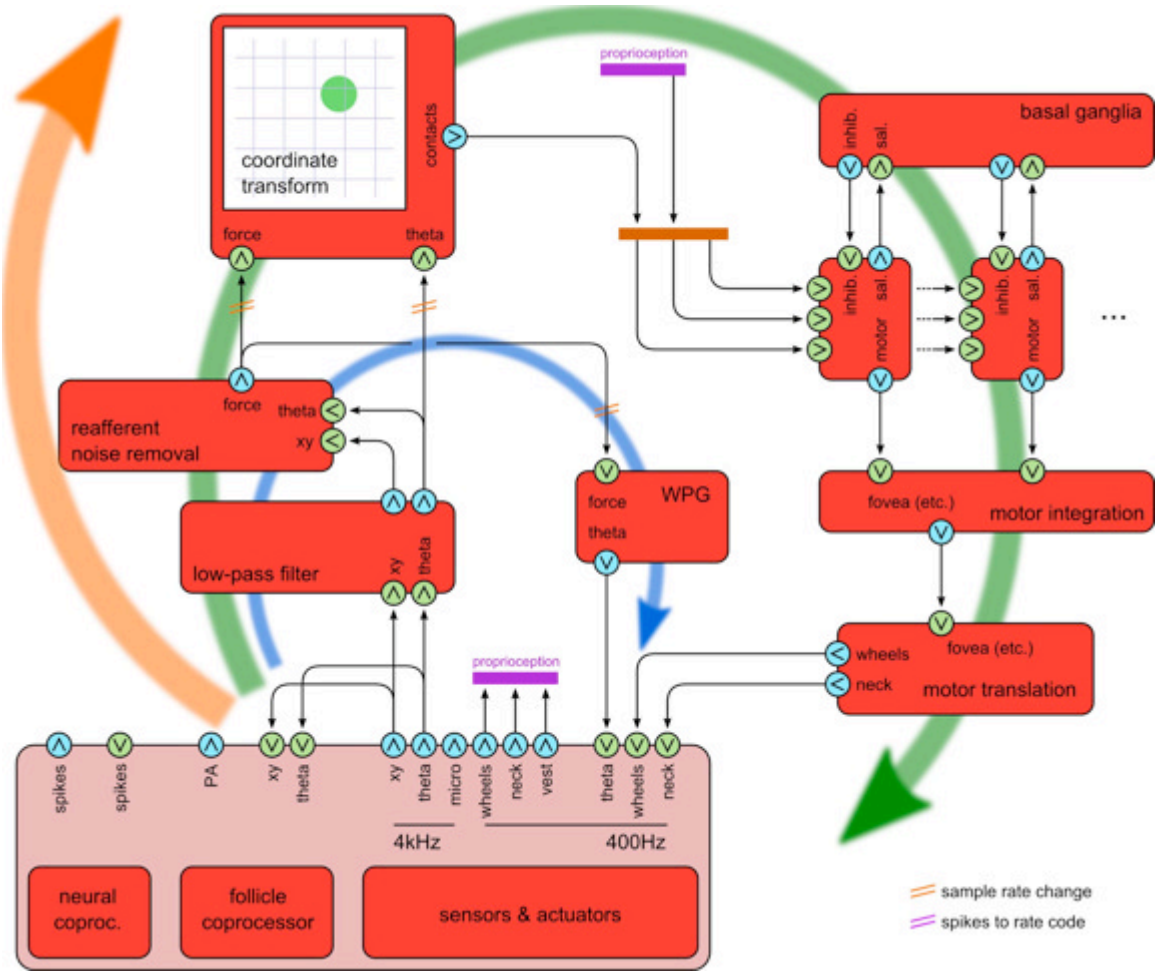


Figure 3. Block diagram of the current processing architecture of the SCRATCHbot platform. Each red block indicates a separate BRAHMS process, with inputs as green circles and outputs as blue. The coloured arrows indicate the two control loops that have been built to date—whisking control (blue) and orient to contact (green); the orange arrow indicates planned future developments.

Figure 4 shows video stills from a typical robot experiment demonstrating the orient to contact response. Implementing this task for our whiskered robot provided insight into some additional problems that the rat must also encounter and has overcome through the mechanisms of evolution and neural plasticity. Specifically, it was evident that there is a significant noise component in the whisker deflection signals that is due self-motion (i.e. by the whisking and head movements) and can cause the robot to make orients to 'ghost' objects that are not actually present. This motivated us to look for brain structures that might function to remove this noise, a prime candidate being the *cerebellum*. Interestingly, the cerebellar-inspired algorithms that we have implemented to successfully remove this re-afferent noise essentially learn the dynamics of each whisker as it is moved. Therefore, if a whisker shaft were to be damaged or replaced, the new dynamics would be acquired and integrated into the control system without the need for manual calibration. This tolerance to damage of individual whiskers and the gradual degradation in performance afforded by an array-based system could provide significant advantages to platforms operating in remote or hostile environments. Clearly, the SCRATCHbot platform itself would be inappropriate for such applications, however, our novel vibrissal sensor technology could be applied to future, more robust, land- or submarine-based vehicles.

Vibrissal pattern recognition for tactile object discrimination

Vibrissal object recognition must begin with low-level tactile feature discrimination. When an

object contacts a whisker, that particular contact may carry information about a number of object properties such as its speed and direction of movement, distance from the base of the shaft, as well as the object's surface orientation and texture. Discriminating these lower level features is key to building up higher-level perceptions of objects and environments.

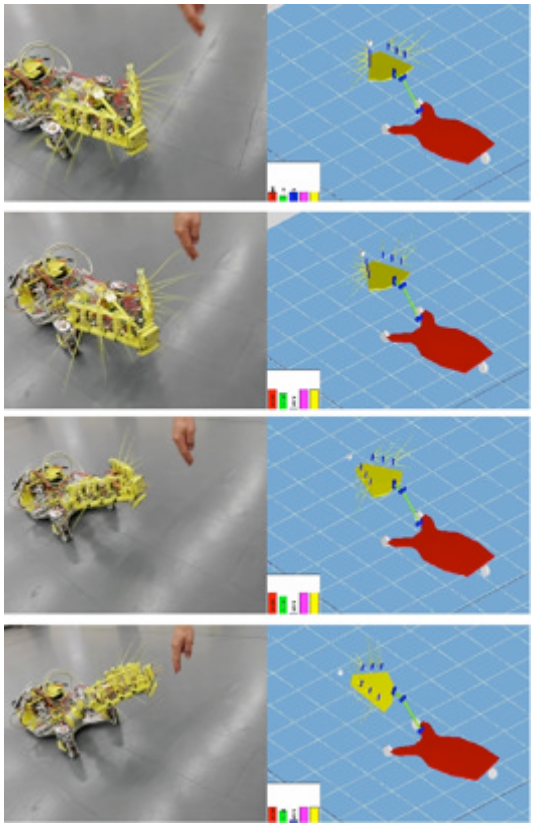


Figure 4. A sequence of video frames from an experimental run of the SCRATCHbot platform demonstrating an autonomous orient toward a point of contact made by the active macrovibrissal array. The animations to the right of the video frames were constructed using odometry collected during the live run. The coloured bars in the lower left corner of the animation indicate the current level of inhibition projecting to each of the robot behaviours currently competing for control of the motors. In this example, the upper whisker of the rear column makes contact with an unexpected obstacle. The orient response competes for, and wins, control of the motor plant. A series of motor commands are then issued which brings the tip of the robot snout (the microvibrissal array) to the point of contact.

In previous work with mobile robot platforms [9, 10] we have demonstrated that actively-controlled vibrissal sensors can discriminate different texture types (e.g. grades of sandpaper). However, more rigorous investigation of tactile pattern recognition calls for precise control of the whisker-object interaction. To this end we have built a 2-DOF robotic positioning system (see figure 5) that allows us to present objects to a SCRATCHbot vibrissal array in a very accurate and repeatable manner. In initial experiments, using this setup, we have presented a straight-sided object to a single vibrissal sensor at a range of radii, speeds, and orientations. These investigations have demonstrated that deflection signals from the Hall effect sensor at the whisker base are sufficiently transparent to be used as inputs to a classifier without any pre-processing. For instance, after presenting 900 unique deflections a template-based nearest-neighbour classifier is capable of 90% accuracy in distinguishing radial distance to contact, and the speed and orientation of the object, for contacts that are at least 5mm in from the whisker tip.

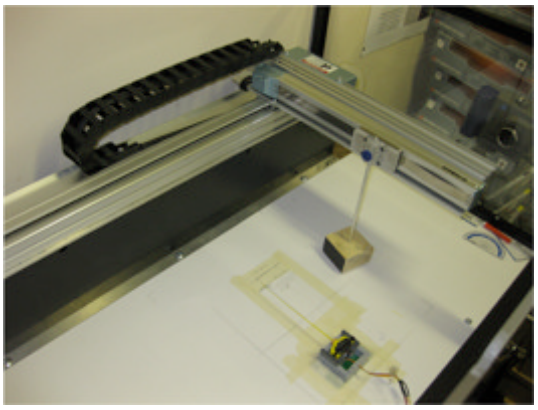


Figure 5. 2-DOF positioning robot for precise exploration of whisker-object interactions. The robot arm is capable of moving up to 1000mm/s, has a range of 650x300mm, and is repeatable to ± 0.02mm.

Our current work is aimed toward the development of feature-based classifiers. Inspection of the signals produced from object contacts reveals features of the signal that consistently predict object properties. For example, the magnitude of deflection predicts object distance and deflection duration reliably predicts object speed. Picking certain features in the data as input to a classifier reduces the complexity of the feature space, allowing the development of more robust and versatile classifiers.

Texture is a key surface property for tactile object recognition and one that rats are able to discriminate, using their vibrissae, with accuracy similar to that of the human fingertip [1, 2]. We have previously shown that vibrissal texture

discrimination is dependent on both surface properties, such as location and orientation [9], and on how the whisker interacts with the surface [10]. Using the feature-based classifiers described above we hope to develop a system for simultaneous extraction of multiple object features, including texture, and ultimately object shape, that will be effective for a wide range of contact situations. As these 'shapelet' classifiers are developed they will be transferred from the XY positioning table to be evaluated and tuned for operation on SCRATCHbot and other multi-whisker vibrissal sensing systems.

Tactile robot navigation and mapping

We are currently constructing a biologically inspired Bayesian-filtering navigation module for SCRATCHbot.

In general, Bayesian filtering computes

$$P(x(t) \mid z(1:t)) = P(z(t) \mid x(t)) \sum_{x(t-1)} P(x(t) \mid x(t-1)) P(x(t-1) \mid z(1:t-1))$$

at discrete steps, where *x(t)* is the pose, and *z(t)* are its sensor measurements (which may include odometry). Exact computation of this filter is an intractable problem and approximation techniques may be constructed in several ways. For instance, sensor models may use raw data or extracted/reduced features, the state model may use assumed parametric form (such as Faussians) or brute-force grid histograms, filter-based navigation may be passive or may include active control of information-gathering explorations. Vibrissal sensors present their own particular challenges in making each of these choices. In developing the navigation model we have also drawn on the rat neurobiology literature by seeking to build a high-level approximation to a further set of rat brain structures—the *hippocampal* system.

Many navigation algorithms [see 11] assume that distances to a fixed set of identifiable, distinct landmarks are always available, as would be the case for range-finding beams and certain visual feature-detection systems. Whiskers differ in providing only very local and ambiguous information. In particular, the *shapelet* reports discussed above will contain information about the position, orientation and texture of walls and other objects within the whisker field and about textures on the floor. Office environments typically contain very few distinctive whisker features that would uniquely identify a location, rather, they yield many repeated shapelets such as edges, corners and floor textures. Hence, prior information about previous locations visited and path-integration mechanisms may be more important for tactile navigation than for wayfinding based on other modalities.

Common heuristics underlying probablistic reasoning systems for navigation [11] typically assume that the distribution of pose beliefs is either Gaussian (leading to the *Kalman filter* and its relatives) or represented by a set of samples (leading to *particle filters*). Our biologically-inspired model [12] assumes that perception is unitary, i.e. that we perceive a single state of the world rather than a "Bayesian blur" of probabilities. This constraint suggests the following algorithm, illustrated in figure 6, which may be though of as a Particle filter with a single particle or as an extended Kalman filter with all posterior uncertainty moved into the transition noise.

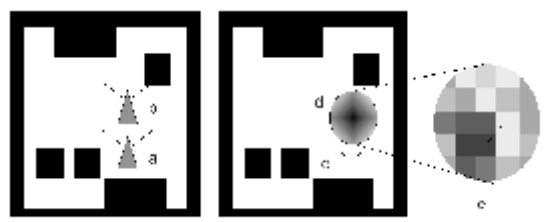


Figure 6. Using a biologically-inspired Bayesian approach to solve the problem of tactile navigation and mapping (see text for full explanation).

In the example shown in the figure, the model robot has moved from *a* to *b* and has detected a shapelet with its right whisker, triggering an update. Assume that the robot's initial position is known exactly as shown by *c*. All odometry since the previous observation is summed, and a corresponding Gaussian error term is computed that is dependent on the length of the summed path. As the shapelet reports are highly nonlinear functions of location, we then switch from the Gaussian parametrisation to a quantised, grid-based method. We consider a trust region, *d*, centred around the middle of the prior, having a radius of two standard deviations (see magnified inset on right). This region is then quantised into a 2D grid, and likelihoods found for the data in each grid position. Fusing these likelihoods with the Gaussian prior gives an approximate posterior, *e*, whose weighted mean location is taken as the next unitary state estimate (shown by the cross is the magnified grid).

Unitary tracking encounters a well known problem. If tracking is ever lost, then it becomes impossible to relocalise without some additional mechanism. A standard solution [13] is to monitor the difference between the sensors and their predicted values, and switch temporarily to purely likelihood-driven sampling if this difference is large for a prolonged period. In our recent theoretical model [12] we postulated a role for the *subiculum-septum* hippocampal pathway in performing such a function. The rat hippocampus is known to be capable of replaying and generating short sequences of states [14]. A possible function of these replays may relate to recovery when lost. If the robot remembers a recent time at which it was not lost, and remembers the sequence of observations since (in non-hippocampal working memory), then this sequence can be replayed and an alternative unitary sequence tested for the possibility that it could provide a better current localisation. This could be repeated multiple times, thus providing a form of biologically-inspired backtracking. Forward generation of possible future trajectories should also be possible, and could prove valuable in hazardous environments where it would allow the robot to simulate and evaluate the risks of moving to different locations before actually doing so.

Conclusion

Current work has brought us to the point where we can begin to evaluate the practicality of active vibrissal sensing for robot tactile object detection/recognition and navigation in indoor environments. We believe that our vibrissal sensors also have great promise for use in riskier environments, particularly in situations where vision can provide only degraded or ambiguous input. In future research we hope to combine vibrissal sensing with artificial sensing in modalities such as vision, audition and olfaction, and with a locomotion system that will allow the robot to negotiation rough terrain. This will bring us closer to the possibility of building autonomous robots with similar capabilities to one of the most successful and versatile animals on the planet—the common rat.

Acknowledgements

Preparation of this paper, and some of the work described, was supported financially by the European Union integrated projects—*Integrating Cognition, Emotion and Autonomy* (ICEA) (IST-027819) and *BIOmimetic Technology for vibrissal Active Touch* (BIOTACT) (ICT-215910) projects (see http://www.biotact.org). The authors thank Chris Melhuish, John Sullivan, and Jason Welsby for their contributions to the design and construction of the Scratchbot robotic whisker system, and Robyn Grant for help with rat observation experiments.

References

[1] T. J. Prescott, M. J. Pearson, B. Mitchinson, J. Sullivan, and A. G. Pipe (2009). Whisking with robots: From rat vibrissae to biomimetic technology for vibrissal active touch. *IEEE Robotics and Automation Magazine*, 16:42-50.

[2] D. Kleinfeld, E. Ahissar and M. E. Diamond (2006). Active sensation: insights from the rodent vibrissa sensorimotor system. *Current Opinion in Neurobiology*, 16:435-444.

[3] B. Mitchinson, C. J. Martin, R. A. Grant and T. J. Prescott, (2007). Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact, *Proceedings of the Royal Society, B*. 274:1035-1041.

[4] Elumotion Ltd. (2009) Biologically inspired robotic devices (http://www.elumotion.com).

[5] M. J. Pearson, A. G. Pipe, C. Melhuish, B. Mitchinson and T. J. Prescott (2007). Whiskerbot: A Robotic Active Touch System Modelled on the Rat Whisker Sensory system. *Adaptive Behavior*, 15(3):223-340.

[6] B. Mitchinson, T. Chan, J. Chambers, M. J. Pearson, M. D. Humphries, C. W. Fox, K. Gurney and T. J. Prescott, (In press). BRAHMS: Novel middleware for integrated systems computation. *Advanced Engineering Informatics.*

[7] M. E. Helmelt and A. Keller (2007). Superior sensation: superior colliculus participation in rat vibrissa system. *BMC Neuroscience,* 8(12).

[8] T. J. Prescott, F. M. Gonzalez, K. Gurney, M. D. Humphries, and P. Redgrave (2007). The robot basal ganglia: Behavior and intrinsic processing. *Neural Networks,* 19: 31-61.

[9] C. W. Fox, B. Mitchinson, M. Pearson, A. G. Pipe, and T. J. Prescott (2009). Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 26:223-239.

[10] M. Evans, C. W. Fox, M. J. Pearson and T. J. Prescott. Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task. *Proceedings TAROS*, 19-24, 2009.

[11] S. Thrun, W. Burgand and D. Fox. (2005). *Probabilistic robotics*. Cambridge, MA:MIT Press.

[12] C. W. Fox and T. J. Prescott (2009). Hippocampus as unitary coherent particle filter, *SfN Abstract*.

[13] S. Lenser and M. Veloso (2000). Sensor resetting localization for poorly modelled mobile robots. *Proceedings of the International Conference on Robotics and Automation*.

[14] E. Pastalkova, V. Itskov V, A. Amarasingham, and G. Buzsáki (2008). Internally generated cell assembly sequences in the rat hippocampus. *Science* 321, 1322–1327.

Naive Bayes novelty detection for a moving robot with whiskers

Nathan F. Lepora, Martin J. Pearson, Ben Mitchinson, Mat Evans, Charles Fox,
Anthony Pipe, Kevin Gurney and Tony J. Prescott

***Abstract*— Novelty detection would be a useful ability for any autonomous robot that seeks to categorize a new environment or notice unexpected changes in its present one. A biomimetic robot (SCRATCHbot) inspired by the rat whisker system was here used to examine the performance of a novelty detection algorithm based on a 'naive' implementation of Bayes rule. Naive Bayes algorithms are known to be both efficient and effective, and also have links with proposed neural mechanisms for decision making. To examine novelty detection, the robot first used its whiskers to sense an empty floor, after which it was tested with a textured strip placed in its path. Given only its experience of the familiar situation, the robot was able to distinguish the novel event and localize it in time. Performance increased with the number of whiskers, indicating benefits from integrating over multiple streams of information. Considering the generality of the algorithm, we suggest that such novelty detection could have widespread applicability as a trigger to react to important features in the robot's environment.**

I. Introduction

A major goal of robotics is to build autonomous devices that can explore and recognize their environment. Novelty detection could be a key component for achieving these tasks, by allowing the robot to notice environmental features that it has never experienced before; these features can then be prioritized for further exploration, after which they are recognizable and thus no longer novel. Another aspect of novelty detection is that, like their biological counterparts, successful biomimetic robots should be able to identify and react to a changing environment. Novelty detection offers a general way to notice such changes and trigger an appropriate change in behavior, such as avoidance or approach.

The present approach to novelty detection is based on a previous study of texture classification [1] with a single biomimetic whisker that sensed various floor surfaces [2]. Given training data from each class of texture, a test dataset was then classified into one of the previously experienced categories using a naive Bayes classifier. This classifier used Bayes rule to calculate the (posterior) probabilities of the test data being from each class of training data, 'naively' assuming statistical independence to simplify the computation. Such classifiers relate to several influential proposals for how humans and animals integrate evidence to make decisions [3]–[6] and are renowned for being effective despite their computational simplicity [7], [8]. This effectiveness was confirmed in the above study, with the classifier achieving success rates close to 90% and many of the errors actually due to data artifacts (such as collisions with objects) rather than misclassifications.

Here we adapted the above naive Bayes classifier of surface texture to novelty detection and applied it to data from a biomimetic robot (SCRATCHbot) inspired by the rat whisker system [9]–[11]. The key idea is that whereas classification recognizes which previous event a new experience is most like, novelty detection finds when a new experience is *not* like a previous one. Novelty, or outlier, detection has many practical applications, such as discovering financial fraud or computer hacking, using a wide variety of statistical, neural network and machine learning methods [12]–[15]. The naive Bayes method is appropriate for making efficient decisions over multiple channels of time series data, motivating its present use in a mobile robot with multiple whisker sensors.

To test the applicability of naive Bayes novelty detection

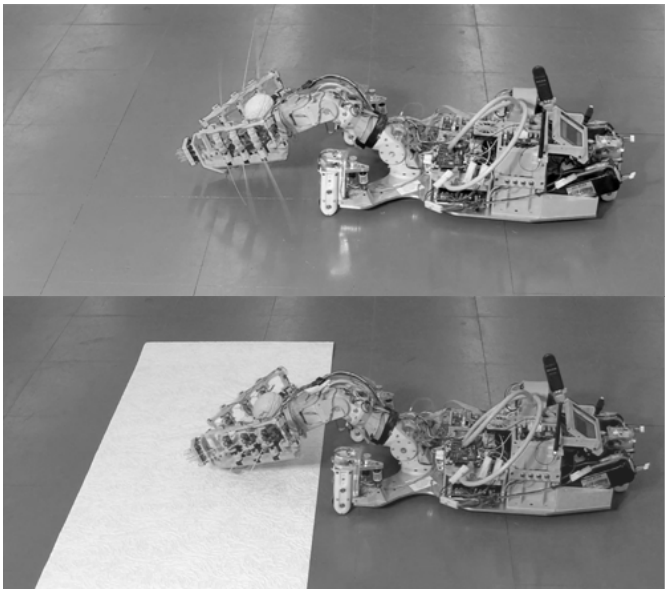


Fig. 1. Novelty detection with SCRATCHbot.
Panel A shows a control trial, where the robot traversed a plain vinyl floor while sweeping its whiskers across the surface. Panel B shows a test trial, where a textured strip was placed in the robot's path. Note that in the bottom panel the whiskers are brushing against the textured strip, even though they cannot be seen clearly against the background.

N. Lepora, B. Mitchinson, C. Fox, M. Evans, K. Gurney and T. Prescott are with the Adaptive Behavior Research Group, Department of Psychology, The University of Sheffield, Western Bank, Sheffield S10 2TN, UK. (email: n.lepora@shef.ac.uk, b.mitchinson@shef.ac.uk, c.fox@shef.ac.uk, mat.evans@shef.ac.uk, k.gurney@shef.ac.uk, t.j.prescott@shef.ac.uk).

M. Pearson and A. Pipe are with the Bristol Robotics Laboratory, Du Pont Building, Bristol Business Park, Coldharbour Lane, Frenchay, Bristol BS16 1QD, UK (email: martin.pearson@brl.ac.uk, tony.pipe@brl.ac.uk).

for a robot task, we used an experimental approach in which a robot was first trained in a familiar environment and then tested in a novel situation (Fig. 1). In training, the moving robot performed some trials in which it swept its whiskers across an empty floor. For testing, the robot then had to detect a textured strip placed in its path. In all cases, the detection algorithm distinguished the novel event based only on previous experience of the familiar situation, and accurately localized this event in time.

II. Materials and methods

A. Data collection

The SCRATCHbot platform [9]–[11] consists of three main components: a head, onto which the whisker arrays are mounted; a neck, that allows the head to move independently from the body; and a body that contains the computing resources, locomotion systems and power supply. The processing architecture of the robot is based on the neural pathways identified in the rat whisker system, and integrated into a unified system using the BRain And Head Modelling System (BRAHMS) execution framework [16]. The biomimetic whisker sensors are engineered to have shape and material properties similar to those of rat whiskers, while scaled to larger sizes appropriate for autonomous robots (see ref. [17] for details).

The head was designed to carry six independent columns of three whiskers, with each column driven by a small motor to whisk forwards and back along the anterior-posterior axis of the head. These columns are arranged in three-by-three arrays, projecting from the left and right sides of the head. To measure deflections of the whisker shaft caused by environmental contact, a small magnet is bonded to the base of each whisker, so that a Hall effect sensor can simultaneously sample the components of the whisker displacement in two directions (denoted by X and Y). The sensory information from each array is passed to the computing resources on the platform body via serial buses. For the present experiment, the data was sampled and recorded at 2kHz.

Each of the bilateral arrays of whiskers has an associated microcontroller to sample all nine whiskers and to control the rotation of the three columns and tilt angle of the array. This rotation was controlled with the Whisking Pattern Generator, based on a model of the activity underlying the rhythmic whisker motion observed in behaving rats [11], [18]. This self-motion introduces a low-frequency artifact in the whisker signal that will make object detection more difficult. Here we preprocessed the whisker data by simply subtracting the low-pass filtered signal off line (using a gaussian smoother of width 5ms). Note, though, that more general and effective adaptive noise cancelation methods are becoming available for this platform that can achieve the subtraction online [19].

The experimental setup for investigating novelty detection consisted of a length of vinyl flooring over which SCRATCHbot traveled whilst sweeping all eighteen whiskers across this surface. As the robot moved forwards, its head also angled from side-to-side to more fully sample the environment in front of it. In addition to this periodic left-right motion, the whiskers also whisked back and forth according to their pattern generator. This generated a total of thirty six data streams (18 whiskers each with two directions) about the floor surface over which the robot was traveling.

Three experimental situations were considered, using eight trials of 10 seconds. The first two trials were the control situation, and consisted of just normal vinyl flooring (Fig. 1A). The next three trials (A1-A3) had the robot travel over a thin strip of rough textured surface placed over the vinyl floor, occurring between about four and seven seconds after the beginning of the trial (Fig. 1B). The final three trials (B1-B3) were similar to the previous three, but instead used a smoother strip that was harder to distinguish from the floor.

The goal of the novelty detection task is for the robot to notice the textured strips in trials A1-A3 or B1-B3 based only on its previous experience of the two control trials. In this sense, the two control trials are considered as *training* data on which to tune the novelty detection algorithm, and the other six trials are *test* data for validating its performance.

B. Analysis methods

1) Probability distributions: The methods described here rely on using the probability distributions of the measured time series values, calculated from the empirical frequency with which each value occurs in training data of the control (familiar) situation. These sensor measurements are sampled in time at frequency f_s and their range is binned into N equal-width intervals.

Denoting the training data of the familiar situation by F, the conditional probability of a quantized measurement q being from it is

$$P(x|\mathrm{F}) \equiv P(q(x)|\mathrm{F}) = \frac{n_q}{\sum_{q=1}^{N} n_q}, \tag{1}$$

where n_q is the total number of times that the value q occurs in the quantized time series. The conditional probability $P(x|\mathrm{F})$ is commonly referred to as a likelihood of the sensor measurement x occurring.

Note that to reduce sampling bias from having too few samples in the distribution tails, the normalized frequencies n_q were convolved with a Gaussian smoother of width 10 quantized intervals before applying Eq. 1, as described in [1].

2) Bayes novelty detection (single measurement): Given a new set of test data, Bayes Theorem states that the (posterior) probability $P(\mathrm{F}|x)$ for a measurement x being drawn from the training data is proportional to the likelihood of that measurement $P(x|\mathrm{F})$ estimated from the training data

$$P(\mathrm{F}|x) = \frac{P(x|\mathrm{F})P(\mathrm{F})}{P(x)}, \tag{2}$$

where $P(\mathrm{F})$ is the (prior) probability of the data being familiar and $P(x)$ is the (marginal) probability of measuring x given no other information.

Here we consider a novelty detector that finds when the posterior probability for a measurement x being familiar

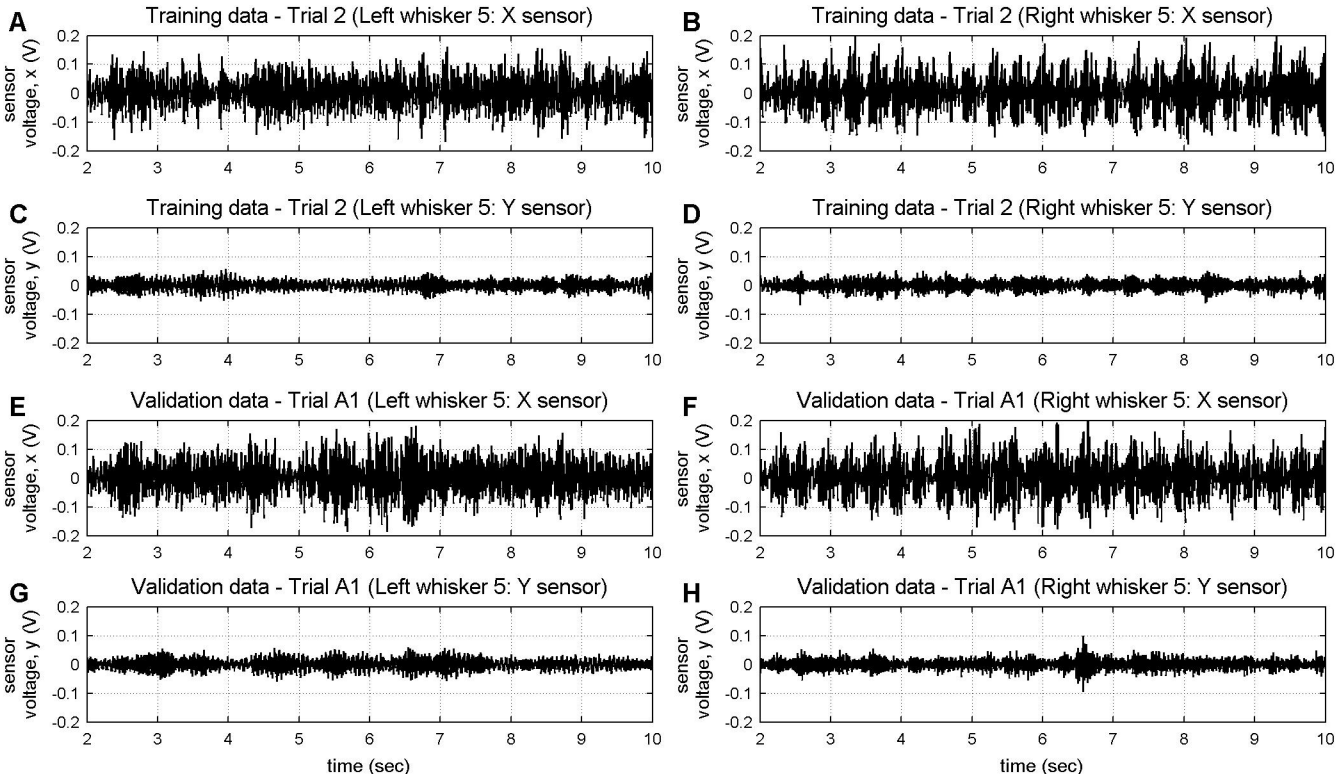


Fig. 2. Example training and test data.
Panels A-D show the X- and Y-sensor measurements for one left and one right whisker in trial 2 of the training data. Panels E-H show the corresponding measurements for the same whisker in trial A1 of the test data. The data was preprocessed to remove the low-frequency artifact due to whisker self-motion.

passes below a fixed threshold θ weighted by the marginal probability. By Bayes theorem, a measurement x is novel if

$$P(\mathrm{F}|x) = \frac{P(x|\mathrm{F})P(\mathrm{F})}{P(x)} < \frac{\theta}{P(x)} \quad \Rightarrow \quad \text{Novel}. \tag{3}$$

For the following arguments, it is convenient to use the logarithm of the posterior probability, which since $\log(x)$ (monotonically) increases with x does not affect the inequality in Eq. 3. Then with a little algebra, the above novelty condition can be rewritten in terms of the log-likelihood

$$\log P(x|\mathrm{F}) < \eta \quad \Rightarrow \quad \text{Novel}, \tag{4}$$

where the novelty threshold $\eta = \log[\theta/P(\mathrm{F})]$ implicitly contains also the contribution from the prior probability.

How is the novelty threshold determined? A simple criterion followed here is to suppose that it is the minimum value of η for which the novelty detector in Eq. 4 would always infer familiarity when tested on the training data. This threshold lies just above the log-likelihood values for all single measurements x,

$$\eta = \sup_{x \in \mathrm{F}} \log P(x|\mathrm{F}). \tag{5}$$

where sup denotes the supremum (least upper bound) over the familiar training set F.

3) Naive Bayes novelty detection (many measurements): A potentially more powerful method for detecting novelty is to make the decision over many measurement values, either in time and/or in parallel across different whisker sensors. The above arguments using Bayes theorem are unaffected if the single measurement x is replaced by a time series $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, of vectors $\boldsymbol{x}_i = (x_i^{(1)}, ..., x_{(i)}^K)$. Then the novelty detection criterion from Eq. 4 becomes

$$\log P(\boldsymbol{x}_{n_s}, \cdots, \boldsymbol{x}_{n_f}|\mathrm{F}) < \eta, \tag{6}$$

and the novelty threshold from Eq. 5 is now

$$\eta = \sup_{\boldsymbol{x}_i \in \mathrm{F}} \log P(\boldsymbol{x}_{n_s}, \cdots, \boldsymbol{x}_{n_f}|\mathrm{F}), \tag{7}$$

with $n_s \leq i \leq n_f$ and n_s, n_f the window start and finish.

An important simplification occurs if all these measurements are assumed independent, because the overall conditional probability factorizes into a product of individual conditional probabilities,

$$P(\boldsymbol{x}_{n_s}, \cdots, \boldsymbol{x}_{n_f}|\mathrm{F}) = \prod_{k=1}^{K} \prod_{i=n_s}^{n_f} P(x_i^{(k)}|\mathrm{F}). \tag{8}$$

Consequently, the classification in Eq. 6 can be rewritten as

$$\sum_{k=1}^{K} \sum_{i=n_s}^{n_f} \log P(x_i^{(k)}|\mathrm{F}) < \eta, \tag{9}$$

and the novelty threshold is now

$$\eta = N \sum_{k=1}^{K} \sup_{x_i^{(k)} \in \mathrm{F}} \log P(x_i^{(k)}|\mathrm{F}), \tag{10}$$

with $N = n_f - n_s + 1$ the length of the time-series. The novelty detector finds when the summed log-likelihoods passes a novelty threshold, given by N times the sum of the individual thresholds (Eq. 5) over all dataset dimensions.

III. Results

A. Properties of the (familiar) training data

Two control trials for training were considered as examples of the familiar situation, in which the moving robot swept its whiskers over a plain vinyl floor (top panel of Fig. 1). This data was preprocessed to reduce the low frequency artifact due to the whiskers' periodic motion relative to the floor. Example traces are shown in Figs 2A-D for the X- and Y-sensor measurements of the whisker deflection, showing one left and one right whisker. The peak reading for the X-sensor was about 0.2V and that for the Y-sensor about 0.05V, consistent with the greater displacement of the whiskers in the horizontal plane during whisking. Visual inspection of recordings from the other whiskers and others trial revealed no obvious qualitative differences from those in Fig. 2.

Data from the two training trials were pooled to determine the probability distributions of the measured X- and Y-sensor voltages for each of the 18 whiskers. The range of sensor voltages were binned into widths of 2mV and the number of values in each bin totaled. These totals are the empirical frequencies of the measurements, from which the probability distributions of the sensor values are estimated (Methods, Eq. 1). All 36 probability distributions resembled Gaussian profiles centered on zero deflection. The amplitudes and widths of these Gaussians were similar across the left and right sensors, but differed between X- and Y-sensors. These probabilities are interpreted as the likelihoods of the X- and Y-sensor measurement for each whisker.

B. Novelty detection over (non-familiar) test data

Given test data that may include a new event, the conditional probability distributions over the familiar training data can be used to estimate the occurrence of a novel event. This estimation uses a novelty score calculated with the naive Bayes' rule (Methods, Eq. 2), which represents the log-probability that the test data was drawn from the same distribution as the training data. (More precisely, this score is the log-posterior probability estimated from the log-likelihood for each test data measurement.) The utility of the naive Bayes assumption is that allows the evidence for novelty to be summed over both the time-window and different whisker sensors (Methods, Eq. 9). In the following, the temporal window was fixed at 0.5 seconds to be consistent with related work on texture classification [1].

The novelty score was first calculated over the familiar training data, using the likelihood determined from that same training data. For both training trials, this resulted in a constant log-posterior value overlaid with random fluctuations (Fig. 3A). A novelty threshold η was then determined as the minimum log-posterior value over both training trials, corresponding to the most unlikely window of training data. Note that the novelty scores plotted in Fig. 3 show the log-posterior normalized by this threshold, and thus the novelty threshold is depicted at unity (dashed line); in addition, because the novelty threshold is negative, the log-posterior/threshold ratio is less than unity over the training data.

This normalized novelty score was then calculated over the six trials of test data, again using the likelihood determined from the training data. As described in methods, this test data consisted of the vinyl floor (like the training data), but with the robot moving over a novel strip in the central portion of the trial (robot pictured in Fig. 1B). Two types of strip were considered: type A, roughly textured, and type B, a smoother texture more similar to the vinyl floor itself. There were three test trials, A1-3 and B1-3, for each strip.

In trials A1-A3, for the roughly textured strip, the normalized novelty score rose above the unit threshold in the central portion of each trial and was below it otherwise (Fig. 3B). Thus, by only having knowledge of the familiar situation, the robot is able to tell that a novel event has occurred and accurately localize it in time. A graphical depiction of the associated novelty detection is shown in Fig. 4A. Novelty detection is clearly a success in this situation.

In trials B1-B3, for the smoother strip, the normalized

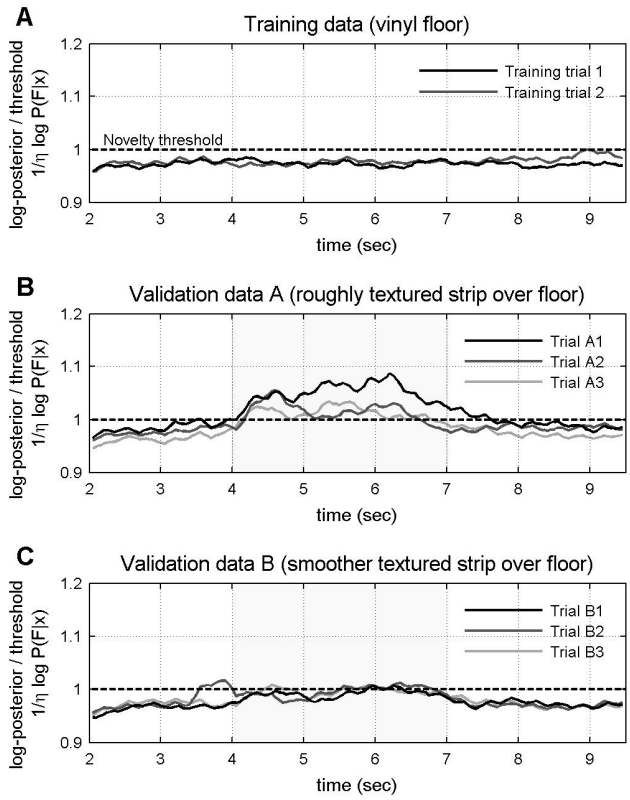


Fig. 3. Novelty score over training and test data.
Panel A shows the novelty score for the training data, while panels B and C are for the two types of test data. This novelty score was the log-posterior probability for a window of data being drawn from the conditional probability distributions for the training data, normalized by the threshold for novelty detection. The approximate times of encountering the novel texture are shown in gray.

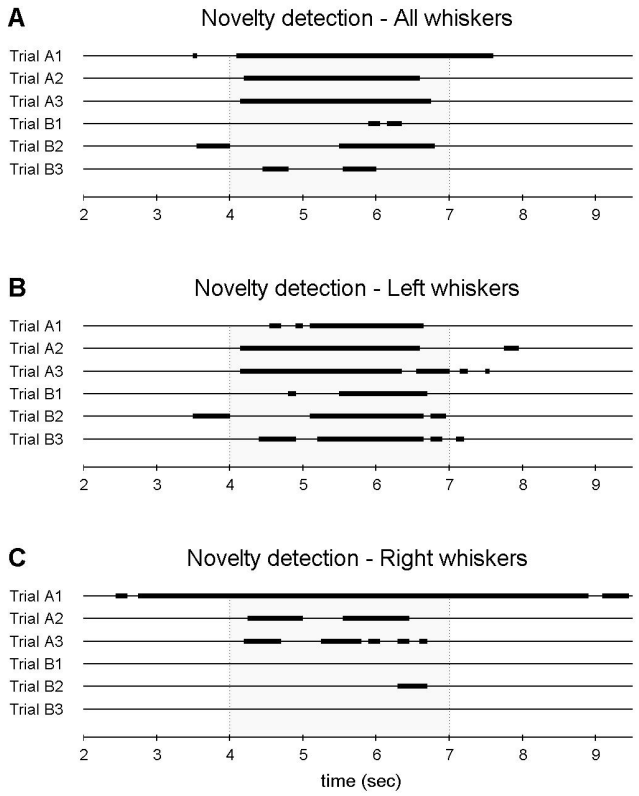


Fig. 4. Times of novelty detection.
In each panel, the thin lines denote the time course of each trial A1-B3 and the bold lines denote times of novelty detection (corresponding to the novelty scores rising above the threshold shown in Fig. 3). Panel A shows the times of novelty detection using all 18 whiskers and corresponds to Figs 3B,C. Panels B and C show the corresponding times for only 9 whiskers grouped to the left-side or right-side of the robot's head. The approximate times of encountering the novel texture are shown in gray.

novelty score again rose above unit threshold in the center of the trial, but was less pronounced than for the textured strip (Fig. 3C). The characterization of novelty now seems more difficult, presumably because the strip is similar to the familiar situation. Examining the graphical depiction of when novelty detection occurs (Fig. 4A), the principal detection times are at the initial and final contacts with the strip. Because the strip is placed over the floor, it seems likely that the novelty score is detecting the sharp contact with its edges. In contrast, the rest of the strip is similar to the familiar situation and is thus not detected as novel.

C. Reliability of novelty detection with whisker number

Thus far the novelty detection has been both trained on and tested against data using all 18 whiskers. Given that each whisker sensor measures both the X and Y deflection, this detection uses a total of 36 distinct information streams.

To examine whether all this information is necessary, we checked whether novelty could be detected reliably using only single whiskers. For the present platform and experiments, this was found to not give a reliable way of detecting novelty, with many false negatives and false positives resulting. This outcome was expected because the novel whisker signals could not be obviously distinguished from the familiar signals by eye (*e.g.* Fig. 2).

Next, we considered an intermediate test using half of the whiskers, grouped to either the left or right sides of the robot's head. The detection performance for the three whisker groups (all, left and right) is depicted graphically in Fig. 4. The top panel (Fig. 4A) shows the control group using all whiskers. The plotted bold lines show when the normalized novelty score rose above unit threshold for the six test trials from Figs 3B,C. As discussed above, the novel events in trials A1-A3 were easily distinguishable, whereas only the initial and final contacts were detectable in trials B1-B3.

For the left whiskers, the novelty detection (Fig. 4B) was slightly over-sensitive compared with the control group of all whiskers. In consequence, there were false-positives in trials A2 and A3, although these inaccuracies were somewhat compensated by more reliable detections in trials B1-3. For the right whiskers, the novelty detection (Fig. 4C) was far poorer that the control group. In particular, trial A1 was detected as novel across most of its duration and trials B1 or B3 had no novel events. Therefore the left whisker group was slightly over-sensitive to novelty, while the right group was a poor detector of novelty.

This unreliability with fewer whiskers indicates that the naive Bayes novelty detector performs best if it is given more information, as one might expect intuitively. Given there were no discernable difference between the left and right whiskers from the training data, there was also no obvious way to choose the better (left) group before testing. Instead, the best decisions were made by pooling over all data. Then if a misleading signal occurred in one data stream, it could be outweighed by reliable information from the other channels.

IV. Discussion

The performance of a naive Bayes novelty detection algorithm was examined in several experiments using a biomimetic robot (SCRATCHbot) based on the rat whisker system. These experiments consisted of two control trials, in which the robot swept its whiskers over an empty floor to characterize a familiar situation. These were followed by six test trials in which a strip was placed across the robot's path, with the first three using a roughly textured strip easily distinguishable from the floor and the last three using a smoother strip more similar to the floor. Using information from all 18 whiskers, the detection algorithm was able to identify the novel event in all test trials. In general, the textured strip had a strong novelty signature that was easy to identify, whereas the smoother strip was mainly evident from the initial and final contacts with its edge.

A crucial aspect of the detection algorithm was that it could integrate information over multiple streams of whisker data. Initial results were thus obtained with all 36 channels (18 whiskers with two directional displacements each). To check the reliability of the detection if less data were used, novelty was reassessed using only 9 whiskers on either the left or right sides of the robot's head. Although the novel event was still noticeable in many test trials, the results

were degraded with many false-positive and false-negative detections resulting. Meanwhile, testing with single whiskers gave no useful information about novelty. In general, robust decisions resulted from pooling as much data as possible, which allowed a misleading signal in one channel to be outweighed by the other evidence.

A. A common system for novelty detection and classification

The present approach to novelty detection is based on a classification method that was previously used to recognize textures [1]. Because the same mathematical structure underlies these two types of algorithm, both tasks can in principle be achieved with a common computational architecture. Thus a robot could test for recognition (classification) and non-recognition (novelty detection) of its environment using a single, hybrid classification/novelty system.

For example, suppose a robot navigates using whiskers to sense texture. It has previously been trained on some representative surfaces, say smooth vinyl and rough carpet, and the memory of each texture is stored as the log-likelihoods of the sensor measurements. On each set of training data, the largest novelty threshold that characterizes this data as familiar has also been calculated and stored. Now, as the robot moves around its environment, it continually streams data to two parallel modules, one for vinyl and one for carpet, that each calculates a log-posterior from their stored log-likelihoods and thresholds this value to test for novelty. The robot can then make various inferences from these results: to recognize vinyl or carpet, it identifies the largest log-posterior; or to determine novelty, it examine whether both modules conclude novelty. These decisions can trigger different behaviors, such as continued navigation in a familiar environment, or exploration/avoidance when there is novelty.

In principle, such a combined approach for classification and novelty detection could be extended to an arbitrary number of classifiers with a corresponding linear increase in computational cost. As the robot moves around its environment, it would instead stream the data to n parallel modules, each of which calculates a log-posterior. Recognition would involve identifying the largest log-posterior, while novelty would involve detecting when the log-posterior from every module passes below their respective thresholds. This more general framework for novelty detection could be tested with new data sets using many textures and objects other than strips across a floor.

B. Novelty-triggered learning

How might an autonomous robot use novelty detection to characterize a new environment? One method would be to use novelty detection to trigger an exploring behavior, where novel aspects of the environment are examined repeatedly to provide training data for future classification. This exploration ceases when the environment no longer appears novel, as judged from applying the novelty detector trained on this data to the environment being explored. The robot can then return to its normal behavior until novelty is again encountered. Such a behavioral strategy would, in effect, intrinsically motivate the robot to reduce the perceived novelty of its environment.

Acknowledgments

The authors thank members of ATLAS (the Active Touch Laboratory at Sheffield), the Bristol Robotics Laboratory and the BIOTACT (BIOmimetic Technology for vibrissal ACtive Touch) consortium. This work was supported by EU Framework projects BIOTACT (ICT-215910) and ICEA (IST-027819) and by the UKRC project REVERB (EPSRC-EP/C516303/1).

References

[1] N. F. Lepora, M. Evans, C. W. Fox, M. E. Diamond, K. Gurney, and T. J. Prescott, "Naive Bayes texture classification applied to whisker data from a moving robot," *Proceedings IEEE World Congress on Computational Intelligence*, pp. 2870–2877, 2010.

[2] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott, "Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task," *Proceedings TAROS*, pp. 19–24, 2009.

[3] J. I. Gold and M. N. Shadlen, "Neural computations that underlie decisions about sensory stimuli," *Trends in Cognitive Sciences*, vol. 5, no. 1, pp. 10–16, 2001.

[4] J. B. Tenenbaum and T. L. Griffiths, "Generalization, similarity, and Bayesian inference," *Behavioral and Brain Sciences*, vol. 24, no. 04, pp. 629–640, 2002.

[5] K. Kording, "Decision theory: what 'should' the nervous system do?" *Science*, vol. 318, no. 5850, p. 606, 2007.

[6] R. Bogacz and K. Gurney, "The basal ganglia and cortex implement optimal decision making between alternative actions," *Neural computation*, vol. 19, no. 2, pp. 442–477, 2007.

[7] D. J. Hand and K. Yu, "Idiot's BayesNot So Stupid After All?" *International Statistical Review*, vol. 69, no. 3, pp. 385–398, 2001.

[8] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001, pp. 41–46.

[9] M. J. Pearson, A. G. Pipe, C. Melhuish, B. Mitchinson, and T. J. Prescott, "Whiskerbot: a robotic active touch system modeled on the rat whisker sensory system," *Adaptive Behavior*, vol. 15, no. 3, p. 223, 2007.

[10] T. J. Prescott, M. J. Pearson, B. Mitchinson, J. C. W. Sullivan, and A. G. Pipe, "Whisking with robots from rat vibrissae to biomimetic technology for active touch," *IEEE Robotics and Automation Magazine*, vol. 16, no. 3, pp. 42–50, 2009.

[11] M. J. Pearson, B. Mitchinson, J. Welsby, A. G. Pipe, and T. J. Prescott, "Scratchbot: Active tactile sensing in a whiskered mobile robot," *From Animals to Animats SAB2010* [In Press].

[12] M. Markou and S. Singh, "Novelty detection: a review–part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[13] ——, "Novelty detection: a review–part 2: neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[14] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[16] B. Mitchinson, T. S. Chan, J. Chambers, M. Pearson, M. Humphries, C. Fox, K. Gurney, and T. J. Prescott, "BRAHMS: Novel middleware for integrated systems computation," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 49–61, 2010.

[17] M. J. Pearson, I. Gilhespy, C. Melhuish, B. Mitchinson, M. Nibouche, A. G. Pipe, and T. J. Prescott, "A biomimetic haptic sensor," *International Journal of Advanced Robotic Systems*, vol. 2, no. 4, pp. 335–343, 2005.

[18] B. Mitchinson, M. Pearson, C. Melhuish, and T. J. Prescott, "A model of sensorimotor coordination in the rat whisker system," *From Animals to Animats 9*, pp. 77–88, 2006.

[19] S. R. Anderson, M. J. Pearson, A. G. Pipe, T. J. Prescott, P. Dean, and J. Porrill, "Enhanced detection of robot whisker contacts by adaptive cancelation of self-generated signals," *IEEE Transactions on Robotics*, vol. 99, pp. 1–12, 2010.

Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems

Mathew H Evans[1], Charles W Fox[1], Martin J Pearson[2], and Tony J Prescott[1]

[1] Active Touch Laboratory, Psychology Department,
Western Bank, Sheffield, S10 2TP, UK
mat.evans@shef.ac.uk
[2] Bristol Robotics Laboratory, Du Pont Building, Bristol Business Park,
Coldharbour Lane, Frenchay, Bristol, BS16 1QD, UK

Abstract. Rats and other whiskered mammals are capable of making sophisticated sensory discriminations using tactile signals from their facial whiskers (vibrissae). As part of a programme of work to develop biomimetic technologies for vibrissal sensing, including whiskered robots, we are devising algorithms for the fast extraction of object parameters from whisker deflection data. Previous work has demonstrated that radial distance to contact can be estimated from forces measured at the base of the whisker shaft. We show that in the case of a moving object contacting a whisker, the measured force can be ambiguous in distinguishing a nearby object moving slowly from a more distant object moving rapidly. This ambiguity can be resolved by simultaneously extracting object position and speed from the whisker deflection time series – that is by attending to the dynamics of the whisker's interaction with the object. We compare a simple classifier with an adaptive EM (Expectation Maximisation) classifier. Both systems are effective at simultaneously extracting the two parameters, the EM-classifier showing similar performance to a handpicked template classifier. We propose that adaptive classification algorithms can provide insights into the types of computations performed in the rat vibrissal system when the animal is faced with a discrimination task.

Introduction

Rats, mice and other whiskered mammals can discriminate a variety of tactile object properties using only their facial whiskers (vibrissae). For instance, rats are able to discriminate surface textures, with different degrees of roughness, with similar acuity to the human fingertip [4]; the Etruscan shrew - the smallest living mammal - can recognise and localise prey animals (insects) from a small number of fleeting whisker contacts, sufficient to allow fast and precisely targeted attacks [1]; and sea mammals such as seals and walruses are able to make judgements about object size, shape, and direction of movement, using only tactile signals from their vibrissae [5].

Several properties of the vibrissal system make it stand out as an interesting model system in which to investigate theories about the sensory guidance of behaviour. First, in tactile sensing systems generally, the sensory apparatus is usually brought into contact

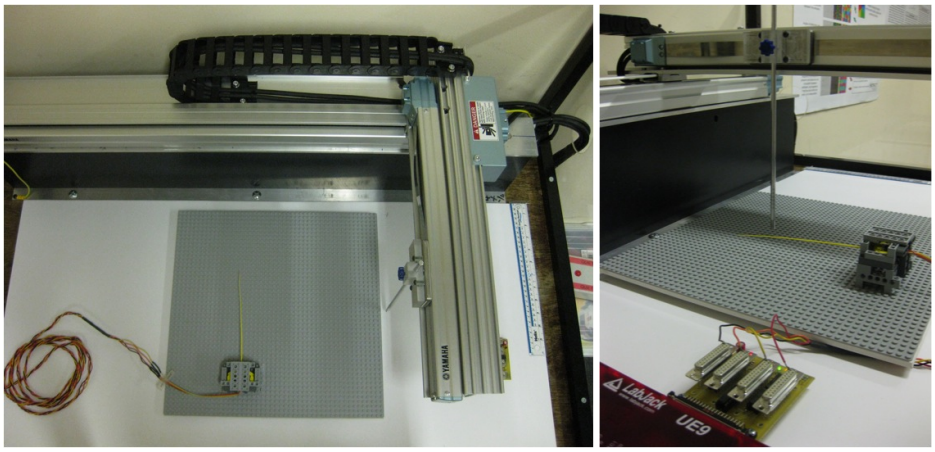
with objects in a deliberate and controlled manner. Whether it is a person exploring an object contour with their fingertips or a rat palpating (whisking) its vibrissae against a surface, purposive control or active sensing is key to information acquisition [14]. Second, the processing of tactile signals may require a relatively small number of stages. For instance, in the rat, there are multiple closed loops connecting vibrissal signals to actuation mechanisms such that new information can begin to influence behaviour after passing through just a small number of synapses [12]. Processing can also be very fast. For instance, whisker contact signals can reach the barrel cortex – from where they can begin to effect processing in behaviour-related areas such as the motor cortex, the superior colliculus, and cerebellum – in just 7 milliseconds [7]. The rat brain thus appears to be tuned to pick out the behaviourally-relevant aspects of vibrissal signals rapidly and in just a small number of steps.

Inspired by the vibrissal systems of mammals, we are working to develop artificial whisker systems for fast and accurate tactile discrimination – which could be useful for mobile robots – and at the same time can be used to test theories of mammalian sensorimotor control. Previous work has shown that information about texture, distance to contact, and shape can be extracted from signals obtained when an artificial whisker is moved against a surface [9] [8] [11] [6]. Our current work extends these findings in several directions. First, by exploring how classifier systems can be trained to extract a range of different tactile properties from whiskers signals with relatively little preprocessing of the sensor input. Second, by showing that such systems can extract multiple features simultaneously from the same signal. Third, by investigating the effects of active control of the sensor apparatus on the problem of tactile feature discrimination. At the same time, though described elsewhere, we are developing decision-making algorithms for these systems that can optimise the speed-accuracy tradeoff [13].

Developing models of whisker based perception has been problematic. In passive sensory modalities such as vision and audition it is generally quite easy to present stimuli to a passive sensor on a robot, or images and tones can be simulated and used to train a computational model. There is no obvious analog for tactile stimuli, and the true nature of tactile stimuli is too poorly understood to be simulated accurately. Whiskers are especially difficult to simulate accurately, as they have very low mass but high spring constants when modelled as a series of masses on rotational springs, leading to numerical instabilities. Additionally when the parameters of a whisker-object contact become more numerous (e.g. speed and radial distance to contact, surface texture, orientation and softness etc) it becomes very difficult to constrain the contact and generate reliable signals in either simulated or physical robots. For these reasons acquiring sufficient examples of carefully controlled whisker contacts with tactile stimuli to train models and classifiers has proved difficult. To facilitate the study of artificial vibrissal sensing we therefore present here a novel system for generating large sets of tactile stimuli and deflection signals. An XY positioning robot is programmed to move objects into an artificial whisker sensor in an accurate and highly repeatable manner (Fig.1). Deflections for the whisker are streamed to a PC, and can be processed in real time to control subsequent movement of the robot arm. Under passive deflections the object moved by the robot arm makes contact with the artificial whisker and deflects the whisker through a large angle. When deflection reaches a critical point the whisker loses friction with the

object, deforms and deflects past the object and goes through oscillatory ringing until the energy dissipates and the whisker comes to rest. However, in addition to passive touch experiments we are also able to use our experimental setup to investigate active sensing. In this case we mimic a control policy that we have observed in rats in our own laboratory whereby the protraction of a whisker ceases rapidly on contact with a surface and whisker then begins to retract [14]. In contrast to the passive case, this policy, which we call Minimal Impingement (MI), keeps the amplitude and duration of whisker deflection within a limited range, and also keeps whisker ringing after contact to a minimum. An additional benefit is that the forces acting on the whisker are much smaller, meaning whisker breakage is less likely – an important consideration for autonomous robotics.

The remainder of this paper motivates and describes a discrimination algorithm developed using this test-bed and shows its utility under differing parameters for accurately extracting object features from vibrissal deflection signals. In future work these algorithms will be transferred to operate in less constrained circumstances onboard mobile robot platforms such the SCRATCHbot robot described in our partner paper [15].



(a) (b)

Fig. 1. The XY positioning robot (a) from above, to show the range of movement available. (b) From the side

Simultaneous radial distance and speed task

The object properties we chose to manipulate and attempt to recognize were radial distance to contact from the base, and contact speed. The task is to recognise these two parameters *simultaneously*. To investigate this problem we presented a vertical pole to the whisker at a range of radial distances from the base, and at different movement

speeds. We show that, depending on the features used for classification, radial distance detection is confounded with contact speed. Previous work [3] [10] has shown that a rat could encode the radial distance to contact along a whisker by monitoring the magnitude of forces (or moments) at the base. Others have suggested that the increased firing rate of cells in the whisker sensory nerve, for contacts close to the base, could be due to the increased moments at the whisker base. Static beam equations, and analyses relying on instantaneous measures of moments do not account for the dynamic properties of objects. If an object collides with a whisker at the same location but at different speeds it will induce different forces at the base. For example, under the right conditions the moment at the base will be the same for a slowly moving object contacting near the base, and a fast object near the tip (see Fig.2 for a demonstration of this). This ambiguity in the signal cannot be accounted for with a single observation, an additional observation or feature must be found in order to discriminate these two properties of the collision. Successful classification relies either on finding the contact speed before conducting a radial distance estimation, or discriminating both properties simultaneously. In the analysis we assess a simple template-based classifier and compare its performance to a classification using an adaptive template classifier, or EM (Expectation Maximisation) algorithm [2] based classifier. Previously we have shown that templates can be used for discriminating tactile features in simulation [8], and that spectral templates can be used to discriminate whisker deflection signals from floor surface textures in a real world environment [6]. In these cases the templates were hand-picked from the data set for better classification. In the present study we show that template based classification can be used to successfully discriminate ambiguous whisker signals in hardware, and that the templates can be found adaptively using a simple Hebb-like learning algorithm.

Methods

The XY positioning robot. An XY positioning robot (Yamaha-PXYX, Yamaha Robotics) (see Fig.1) was used to move objects into the whisker. The robot has a movement range of 350x650mm, and can move up to 720mm/s. Repeatability of the robot is ± 0.01mm, and the maximum load it can carry is 1.5kg. Objects are carried by the robot into an artificial whisker fixed to the table, as this allows us to control the contact as carefully as possible. Moving the whisker into an object would cause the whisker to oscillate unpredictably during movement between contacts, and as a result each contact would be slightly different. A controller (Yamaha RCX 222, 2-axis robot controller) takes instructions from a PC through an RS232 cable, and the controller interprets the instructions, completes path integration, and drives the motors. Instructions for the robot are generated inside a Matlab (www.mathworks.com) loop, and can be easily updated during robot operation, depending on the whisker input.

The Whisker. A whisker sensor was taken from the SCRATCHbot robot platform (described in detail in [15] [16]). Technical details of the whisker can be found in [6]. A tapered, flexible plastic whisker, $\approx$5 times scale models of a rat whisker, was mounted into an inflexible rubber-filled 'follicle' case (Fig.3(a)). A tri-axis hall effect sensor mounted in the follicle case measures the deflection of a magnet fixed to the base of the whisker shaft (Fig.3(b). The hall effect sensor IC was programmed to generate two volt-

ages, corresponding to the magnitude of the whisker base deflection in two directions, x and y.

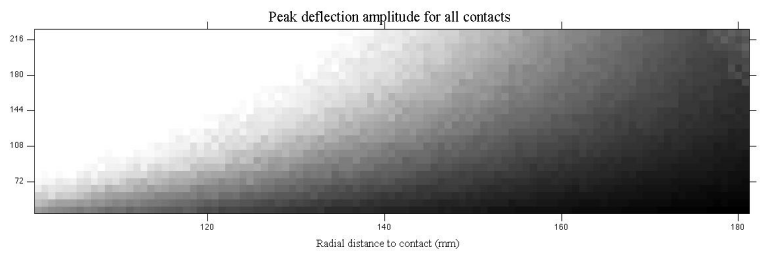
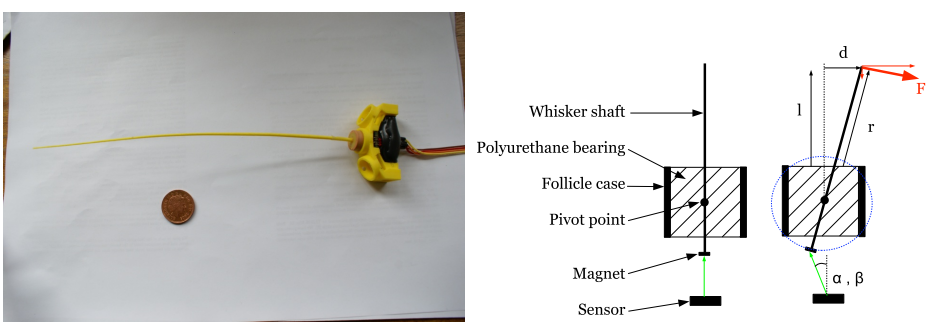


Fig. 2. Peak deflection magnitude for each speed-radial distance pair. Deflection magnitude (brightness of pixel), or force, has been used in the past as a discriminator of radial distance to contact. Here a given radial distance results in differing deflection magnitudes depending on movement speed. If deflection magnitude were a sufficient criteria for radial distance estimation, the brightness would decrease evenly across all speeds (Y axis).



(a) (b)

Fig. 3. Artificial whisker shaft and follicle, with a UK 1 pence coin for scale (b)Diagram of the artificial whisker Hall effect sensor.

Data. Deflections of the whisker were transmitted through the hall effect sensors to a LabJack UE9 USB data acquisition card (www.labjack.com) at a rate of 1 kHz for each of the x and y directions. Each trial lasted 4s. This data was sent to a computer through the BRAHMS middle- ware (brahms.sourceforge.net) for analysis in Matlab.

Robot control. Minimal impingement was implemented by instructing the robot to move an object into the whisker at a given speed until a deflection threshold is crossed, at which point the robot retracts the object as fast as possible (720mm/s). Temporal latency for the loop is $\approx$ 300ms from initial contact due to the controller duty cycle. Though this latency is short enough for the present study, we are working to reduce this latency by gaining more direct control over the robot motors.

The task. Preliminary investigations highlighted that the closest contact that could be made by the whisker at any reasonable speed without saturating the Hall effect sensor was $\approx$80mm from the base. Contacts at less than 5mm from the tip did not deflect the base of the whisker for long enough before slipping past to allow an MI type contact. Therefore, the 185mm length whiskers provide a 100mm range of radial distances. Contact speeds above 216mm/s either cause the whisker to slip past the object before a retraction, or saturates the sensors. 36mm/s was the lower bound on the speed here. Contacts were sampled at radial distance intervals of 1mm, and speed intervals of $\approx$7mm/s over the previously described ranges. In total 101 radial distances and 26 speeds were sampled, giving 2626 different radial distance and speed combinations. Contact combinations were randomly interleaved to limit any affects of changing whisker properties. For each contact combination the whisker was deflected by the robot in both a clockwise and anticlockwise directions (-ve and +ve in x), ensuring that the whisker did not bend over time through repeated unilateral deflections. The experiment was performed twice to generate sufficient data for classification.

Data from each trial was stored separately. Deflections from the clockwise robot movement trials (-ve in x) were converted so all data samples were equivalent. Trials were ordered into arrays by speed and radial distance to contact. Each trial was aligned to peak deflection, and shortened to only the 325ms either side of the peak deflection (751 data samples in total).

Analysis

The data were separated into training and test sets that were each complete data sets of 26 speeds and 101 radial distances. Signals were placed in the training or test sets at random from the original data. In each case classifiers were developed on the training sets, and performance was determined on the test set.

Template based classification. Template based classification involves recording example sensory data as templates during a training phase, and comparing the stored templates to novel data during the test phase. By systematically comparing the novel data to signals encountered previously, a classification can be made by declaring which of the stored templates the novel signal is most similar to. In the present study each template corresponds to a speed-radial distance pair. Classification based on these templates is therefore simultaneous classification of both speed and radial distance. From the training data set a subset of trials – representative of the larger set – were stored in an array as templates. The number of templates chosen were dependent on the experimental condition. During the test phase, trials were taken at random from the test set as inputs to the classifier. An element-wise sum of squared errors calculation was made between the input I and each template T_i,

$$e(T_i) = \sum_{t=1}^{n} (I(t) - T_i(t))^2. \tag{1}$$

where n is the length of the template, in samples. The template with the lowest sum of squared error was determined the winner, and a recording was made in an output array of the estimated speed and radial distance to contact of the input trial. We conducted the experiment with a full complement of templates (26x101) as a benchmark for classification, as well as with two reduced sets of templates, (13x50 and 7x26 templates respectively).

Adaptive EM template based classifier. Picking templates by hand, though successful, is an inefficient method of developing a classifier. A better method is to adaptively learn a set of templates in an unsupervised way, to find a set that covers the bulk of the data space for a given number of templates. Such adaptation can be performed with an EM algorithm [2]. Each time a classification is made it is verified against the true value of the two contact parameters. If the classification was correct, the winning template is modified to be more like the input pattern. If the classification was incorrect, the appropriate losing template is modified to be more like the input pattern. This template modification is achieved by taking a weighted mean of the template and the input,

$$T = \frac{(I + \alpha T)}{1 + \alpha} \tag{2}$$

where α is the learning rate of the classifier. By increasing $alpha$ it is possible to learn quicker, but it is less likely that an optimal value will be reached. (The positive and negative modifications to templates may be viewed as Hebbian and anti-Hebbian learning respectively, for templates as linear neurons.) We decided to run the adaptive classifier with a reduced set of templates (13x50), to see if it was possible to achieve performance close to or equivalent to that of hand picked templates.

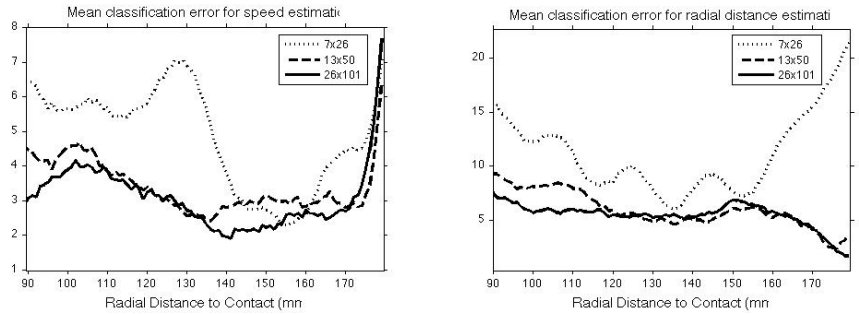
Results

Template based classifier. The template based classifier with 13x50 templates was capable of successfully classifying 65% of inputs to within 50mm/s of speed and 10mm of radial distance. This performance decreases as the number of templates used is reduced.

Adaptive EM template classifier. We found that after training the Adaptive EM template classifier was capable of classifying 66% of inputs to within 50mm/s of speed and 10mm of radial distance.

Fig.5 shows the results in graphical format. Input signals are shown, arranged for speed and radial distance. Pixel brightness indicates the value assigned to that input signal. A completely correct classification of speed (Fig.5(a) would appear as a gradual transition from dark to light vertically along the Y axis, and a corresponding transition from light to dark horizontally along the X axis for correct radial distance detection (Fig.5b). We can see that both classifiers achieve some degree of correct classification, as well as a number of mistakes. Classification tends to be best for larger radial distances (contacts nearer the tip, lower right region of each plot). Contacts at high speed and near

the base tend to be systematically misclassified (upper left region of each plot). Fig.4 shows the mean error of classification for each point along the whisker. Error is lowest in the region near, though not at, the whisker tip. Performance of the classifiers with reduced sets of templates is good, though the performance of the classifier with the fewest (7x25) templates is less reliable over certain regions of the space.



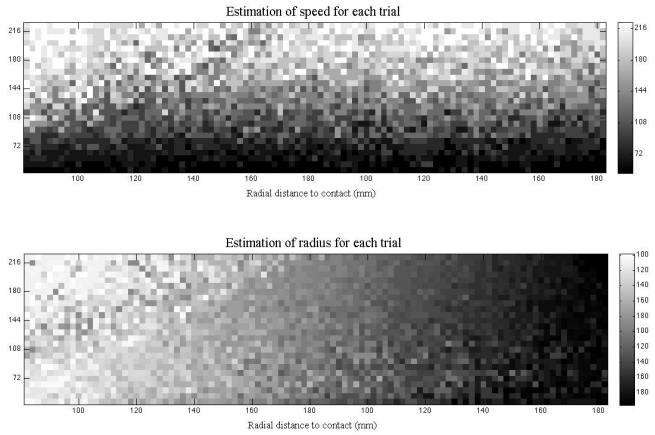
(a) (b)

Fig. 4. Mean classification error for each radial distance along the whisker, for different numbers of templates. Y axis corresponds to mean distance between the real and predicted values of speed (a) and radial distance (b). Error is at its lowest in the portion of the whisker near, though not at, the tip.

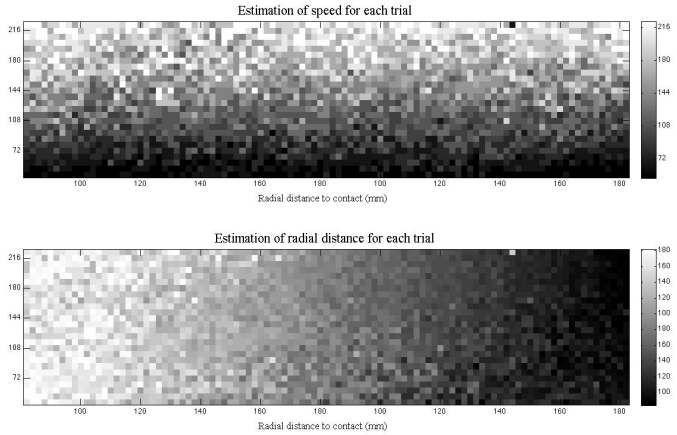
Discussion

By collecting a large data set and exhaustively tiling a feature space we have shown that certain features in the whisker signal that are ambiguous in isolation can be simultaneously discriminated using a robust, and computationally cheap adaptive classified system. Previously we have shown [9][6] that successful classification of surface properties, such as texture, is highly dependent on knowing the location of the surface and the nature of the contact. We believe that the first steps towards the goal of simultaneously extracting a range of relevant object properties have now been taken. The success of the classifiers at contacts nearer the tip may also go some way to explain the whisking behaviour seen in rats. Rats appear to control their whiskers so as to make contacts at or near the whisker tip [14], possibly because this creates signals that vary more predictably across contact parameters allowing the animal to make better judgements. More generally, keeping whisker deflection amplitude and duration within a limited range using active sensing strategies will allow the development of classifiers that are more sensitive to smaller changes in the input. Indeed, the data presented here suggests that contacts of this type are easier to discriminate over a particular section of the whisker, suggesting a 'sweet spot' of whisk speed and whisker contact location. The development of fast, adaptive classifiers for tactile feature discrimination could also provide insights into signal processing in areas of the vibrissal system such as

the barrel cortex where the rat is known to be able to rapidly extract behaviourally-relevant properties of the stimulus in a small number of processing stages. Ullman et al [17] have proposed that the visual system operates through hierarchies of progressively more complex adaptive feature-matching templates. The ideas investigated in the current paper might therefore be considered as the first step towards identifying a similar, general scheme for understanding cortical processing in the domain of touch.



(a)



(b)

Fig. 5. Classification performance of the EM template classifier (a), against that of a classifier with hand picked templates.

References

1. Farzana Anjum, Hendrik Turni, Paul G H Mulder, Johannes van der Burg, and Michael Brecht. Tactile guidance of prey capture in etruscan shrews. *Proc Natl Acad Sci U S A*, 103(44):16544–16549, 2006.
2. A.P.Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
3. J Alexander Birdwell, Joseph H Solomon, Montakan Thajchayapong, Michael A Taylor, Matthew Cheely, R Blythe Towal, Jorg Conradt, and Mitra J Z Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *J Neurophysiol*, 98(4):2439–2455, 2007.
4. G E Carvell and D J Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *J Neurosci*, 10(8):2638–2648, 1990.
5. G. Dehnhardt and G. Ducker. Tactual discrimination of size and shape by a california sea lion (zalophus californianus). *Animal Learning and Behavior*, 24(4):366–374, 1996.
6. Mat Evans, Charles W Fox, Martin J. Pearson, and Tony J Prescott. Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In T Kyriacou, U. Nehmzow, C. Melhuish, and M. Witkowski, editors, *Proceedings of Towards Autonomous Robotic Systems (TAROS 2009)*, pages 19–24, 2009.
7. Isabelle Ferezou, Sonia Bolea, and Carl C H Petersen. Visualizing the cortical representation of whisker touch: voltage-sensitive dye imaging in freely moving mice. *Neuron*, 50(4):617–629, 2006.
8. Charles Fox, Mat Evans, Martin Pearson, and Tony J Prescott. Towards temporal inference for shape recognition from whiskers. In Subramanian Ramamoorthy and Gillian M. Hayes, editors, *Towards Autonomous Robotic Systems*, pages 226 – 233, 2008.
9. Charles W Fox, Ben Mitchinson, Martin J Pearson, Anthony G Pipe, and Tony J Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.
10. M. Kaneko, N. Kanayama, and T Tsuji. Active antenna for contact sensing. *Robotics and Automation, IEEE Transactions on*, 14(2):278–291, Apr 1998.
11. DaeEun Kim and Ralf Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55:229–243, 2007.
12. David Kleinfeld, Ehud Ahissar, and Mathew E Diamond. Active sensation: insights from the rodent vibrissa sensorimotor system. *Curr Opin Neurobiol*, 16(4):435–444, 2006.
13. N. F. Lepora, M. Evans, C. W. Fox, M. E. Diamond, K. Gurney, and T. J. Prescott. (submitted) naive bayes texture classification applied to whisker data from a moving robot. In *World Congress on Computational Intelligence.*, 2010.
14. Ben Mitchinson, Chris J Martin, Robyn A Grant, and Tony J Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.
15. Martin J. Pearson, Ben Mitchinson, Jason Welsby, Tony G Pipe, and Tony J. Prescott. Scratchbot: Active tactile sensing in a whiskered mobile robot. In *Submitted to SAB*, 2010.
16. T.J Prescott, M.J. Pearson, B Mitchinson, J.C.W. Sullivan, and A.G. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.
17. Ullman S, Vidal-Naquet M, and Sali E. Visual features of intermediate complexity and their use in classification. *Nat Neurosci.*, 5((7)):682–7, July 2002.

Hippocampus as unitary coherent particle filter

Charles Fox and Tony Prescott
Adaptive Behaviour Research Group
Department of Psychology, University of Sheffield
charles.fox@sheffield.ac.uk

***Abstract*— We present a mapping of the hippocampal formation onto a Temporal Restricted Boltzmann Machine [23] based architecture, running a deterministic version of Gibbs sampling, and extended with a lostness detection and recovery circuit modelled on subiculum and septal acetylcholine (ACh). The mapping approximates Bayesian filtering, which infers both auto-associative de-noised percepts and temporal sequences, the latter including sequences of places during navigation. Inference may be viewed as a neurally implemented particle filter with a single particle – as suggested previously [5] as a purely behavioural animal model.**

I. Introduction

The classical view of hippocampus is as a single loop. The principal input structures of the hippocampus are the superficial layers of Entorhinal Cortex (ECs). ECs projects to Dentate Gyrus (DG) which is believed to sparsify the encoding of ECs. Both ECs and DG project to CA3, which also receives strong recurrent connections that are disabled [10] by septal ACh. There is a recently-discovered back-projection from CA3 to DG [20]. CA3 and ECs project to CA1, which in turn projects to the deep layers of Entorhinal cortex, ECd. ECs, CA1 and ECd outputs appear to share a coding scheme, as evidenced by one-to-one topographic projections. In contrast, DG and CA3 outputs are thought to work in a second basis or latent space. There is a also a second loop. ECs and CA1 project to Subiculum (Sub), which projects to the midbrain Septum (Sep) via fornix. Septal ACh and GABA fibres project back to all parts of hippocampus.

We make a distinction between two broad schools of hippocampal modelling: spatial sequences, and auto-associative. They differ in their representation of the world, and in the role attributed to the recurrent connections.

Spatial sequence models (e.g. [21], [2], [4]) are based on the existence of rodent cells responding to place. They use the activity of the ith CA3 (or sometimes CA1) cell to represent a strength of belief, $P(i)$, in the agent being at the ith discrete place,

$$CA3(i) = P(i). \tag{1}$$

Weights connect the place cells to sensory inputs, and possibly to features extracted from inputs by DG. Asymmetric recurrent CA3 weights may then specify probabilities of moving from one place to another. Assuming that the places are mutually exclusive and exhaustive, such models are similar to the Hidden Markov Model which we discuss in section III, inferring a complete distribution of place probabilities $\{P(i)\}_i$ from a vector of inputs and prior transitions.

In contrast, *auto-associative models* based on primate physiology [15], [12], [18], [22] use vector-coded discrete or continuous attractors to represent a single ('unitary') but complex ('coherent') state of the world. For example, each unit in a Hopfield network [12] may correspond to the presence of an object, and the network stores and recalls patterns of many objects which occur together. Thus an episode corresponds to a collections of facts experienced simultaneously, which may include the agent's location and its percepts. To construct stable attractors, the recurrent connections are generally assumed to be symmetric, $w_{ij} = w_{ji}$ (there is no biological evidence for this assumption), and trained so that an energy function,

$$E = \sum_{ij} w_{ij} CA3(i) CA3(j), \tag{2}$$

is optimised at stored episodes. By encoding the world state as a vector, we note that only a single world state,

$$\arg\max_{\{i\}} P(\{i\}_i), \tag{3}$$

can be represented at each point in time – not a probability distribution *over* world states.

A recent model in machine learning, the Temporal Restricted Boltzmann Machine (TRBM) [23], has been proposed [3] as a hippocampal model, and uses a rate-coded, mean-field assumption to approximate whole belief distributions over a vector-coded CA3. This combines aspects of both the above modelling approaches. Our model uses a similar architecture to the TRBM but uses a different inference algorithm, and extends the architecture with a lostness detection and recovery module, mapped to Sub and cholinergic Sep functions. Our model is also similar to that of Samu et al. [19] which uses a CA to correct rate-coded grid cells. As with the TRBM, this model treats CA as recognising and de-noising configurations of inputs. Our model differs from Samu et al. by specifying precise probabilistic semantics; adding recurrent CA3 connections; using sampling instead of rate coding; adding DG, CA1, Sub and Sep; and de-noising the whole input vector rather than just the inputs that are grid cells.

We do not model biological learning in the present study, rather we set weights by hand and with a machine learning algorithm. The aim is purely to illustrate inference and lostness detection and recovery in a pre-trained network. Previous experiments and models [10] suggest that septal ACh is involved with learning: we suggest an additional, complementary function for this signal in lostness recovery.

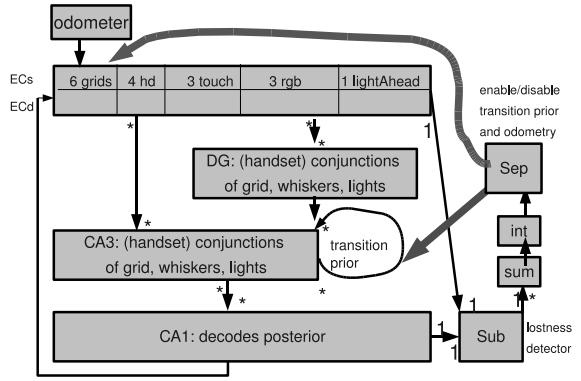


Fig. 1. Hippocampal network architecture used in the model. Connections are labelled with UML notation indicating many-to-many fully connected links (* → *), one-to-one links (1 → 1) and many-to-one links (* → 1). Thick lines are ACh projections, thin lines are glutamate.

II. Bayesian filtering

The objectives of both auto-associative and spatial sequence memories are combined by a general Bayesian filter with noisy observations, which infers the hidden state of the world x_t at each time from a series of noisy observations $z_t = y_t + \epsilon_t$ of ideal, deterministic sensor states $y_t = f(x_t)$ through the update

$$P(x_t|z_{1:t}) = \frac{1}{Z}\left(\sum_{y_t} P(z_t|y_t)P(y_t|x_t)\right) \times \left(\sum_{x_{t-1}} P(x_{t-1}|z_{1:t-1})P(x_t|x_{t-1})\right). \quad (4)$$

where Z is a normalising coefficient. Such a filter is further able to provide a de-noised version of the sensors,

$$P(y_t|z_{1:t}) = \frac{1}{Z}\sum_{x_t} P(y_t|x_t)P(x_t|z_{1:t}). \quad (5)$$

which is a form of auto-association, but one which also incorporates prior knowledge from the previous step in a sequence of inferences. If part of the sensor observation relates to place then spatial sequences can be captured, but also sequences of other senses.

We would like to map the Bayesian Filter to a high-level model of the hippocampus circuitry, to allow it to perform both auto-associative and spatial sequence inference. Let us first consider how a purely spatial sequence, place-cell based model may be mapped onto such a filter. In this case, the state of the world is a *single* variable representing the agent's location, rather than a complex coherent set of associations. The ith CA3 cell could each represent the ith location,

$$CA3_t(i) = P(x_t = i|z_{1:t}), \quad (6)$$

and each of the EC and DG cells $z_t(j) = (EC_t, DG_t)$ represents the presence or absence of a sensory feature. As the hidden states are exhaustive and mutual rivals, this special case of the Bayesian filter is a Hidden Markov Model (HMM). Neural HMM mappings may be subsequently be constructed [17] so that the activation of the CA3 cells can represent the location posteriors. Such mappings allow optimal tracking during navigation, assuming a set of discrete places, but unfortunately cannot account for the presence of other CA3 cells in the hidden state such as those found in biology responding to odours [6] and objects [7].

III. Unitary coherent particle filtering

Auto-associative models, on the other hand, can represent complex hidden states of the world, such as:

isAt(self,location1) AND isAt(edge,myRight) AND isVisible(light) AND isAt(light,location3),

which denotes the agent's own allocentric position at location 1, the egocentric presence of a boundary on the agent's right, the immediate visibility of light, and the position of a light in allocentric space. Each logical term can be represented by the activation of a CA3 cell, and thus the conjunction of terms by the activity of the whole CA3 vector. Such representation is *unitary* because it represents a single state of the world, rather than a pdf over many states. It is *coherent* because it comprises several facts which mutually cohere with one other.

Such a scheme can represent exponentially more unique states than the HMM mapping above, but by forfeiting the ability to represent a belief distribution over states. The presence of cells responding to particular facts is in agreement with the biological CA3. We would thus like to utilise such representations in a hippocampus model, as in auto-associative memories. But we would also like to retain an ability to track sequences of states, including locations, as in the HMM mapping.

We combine these ideas using a variant of the Temporal Restricted Boltzmann Machine (TRBM) architecture [23]. The TRBM assumes weight matrices W_{xz} and W_{xx}, Boolean vectors for the hidden state $x_t(i) \in \{0,1\}$ and observations $z_t(j) \in \{0,1\}$, and specifies joint probabilities,

$$P(x_t, x_{t-1}, z_t) = \frac{1}{Z}\exp\sum_t(-x_t W_{xx} x_{t-1} - x_t W_{xz} z_t - b_x x_t - b_z z_t), \quad (7)$$

where b are biases that may be viewed as specifying the priors on each of x and z. If we extend each population vector v to a vector $v' = (v, 1)$, appending an additional node which is always on, then the biases may be moved inside the weight matrices:

$$P(x_t, x_{t-1}, z_t) = \frac{1}{Z}\exp\sum_t(-x'_t W_{x'x'} x'_{t-1} - x'_t W_{x'z'} z'_t).$$

Assumption 1. Unlike the standard TRBM – which uses rate coding – we will assume when inferring x_t that both the sensors and previous hidden state are *observed*. That is, we assume that the previous inference step produced an exact, correct estimate $\hat{x}_{t-1} = x_{t-1}$. Under this assumption, all

links in the TRBM graphical model [23] become effectively directed, and we may write

$$P(x_t|x_{t-1}, y_t) = \frac{1}{Z} \exp \sum_t (-x_t' W_{x'x'} x_{t-1}' - x_t' W_{x'z'} z_t').$$

A previous mapping of TRBM to hippocampus [3] used the rate-coded, variational approximation to the full posterior,

$$Q(x_t) = \prod_i Q(x_t(i)) \approx P(x_t|z_{1:t}), \tag{8}$$

as in the original TRBM [23]. In contrast, we suggest the following update as a model of CA3 function,

$$\begin{aligned} CA3_t = \hat{x}_t &\leftarrow \arg\max P(x_t|\hat{x}_{t-1}, z_t) \\ &= \{\hat{x}_t(i) = (P(x_t(i)|\hat{x}_{t-1}, z_t) > \frac{1}{2})\}_i \end{aligned} \tag{9}$$

This is a deterministic update which may be viewed as the zero-temperature limit of an annealed Gibbs sampler. As we update to a unitary coherent state at each time step, all nodes are Boolean valued and the conditioning on $\hat{x}_{t-1}$ becomes definite, satisfying Assumption 1. Therefore the cells $\hat{x}_t(i)$ become independent over i and may be updated locally and individually, using sigmoidal threshold-and-fire units with

$$\begin{aligned} P(x_t(i)|\hat{x}_{t-1}, z_t) &= \frac{P(x_t(i)|\hat{x}_{t-1}, z_t)}{P(x_t(i)|\hat{x}_{t-1}, z_t) + P(\neg x_t(i)|\hat{x}_{t-1}, z_t)} \\ &= sig(W_{x'x'}(i)\hat{x}_{t-1}' + W_{x'z'}(i)z_t'), \end{aligned} \tag{10}$$

where $W(i)$ denotes the ith row of matrix W, $sig(x) = (1 + \exp(-x))^{-1}$ is the sigmoid function, and the output of eqn. 10 is used with a threshold of $\frac{1}{2}$ to set the Boolean vector $\hat{x}_t$ as in eqn. 9. This tracking method may be viewed as a particle filter (PF) with a single particle, and the zero-temperature annealed Gibbs distribution as its proposal distribution. The particle has a complex, vector-coded coherent state, hence we call the algorithm a *unitary coherent particle filter (UCPF)*. Unitary particle filters have previously been proposed as a model of external behaviour [5] – here we give an internal neural implementation which could give rise to such behaviours.

We map the noisy inputs to the combined ECs and DG, where the DG activations are functions of the ECs activations, $z_t = (ECs_t, DG_t(ECs_t))$. Finally we map the estimated de-noised output to ECd, $\hat{y}_t = ECd_t$. Each neural population is a Boolean vector at each discrete time step t, which may be viewed as an abstracted spike or absence of spike within a theta cycle.[1]

We will later describe subpopulations within each region. For example, ECs contains a subpopulation of grid cells, which in turn is indexed by 2D row and column coordinates r, c. We use chains of functions to index such populations, for example, ECs denotes the whole ECs population, $ECs(grid)$ denotes the subpopulation of grid cells, and $ECs(grid)(r, c)$ denotes a particular grid cell at row r and column c. We may also write $z_t(ECs)(grid)(r, c)$ to emphasise that ECs is itself a subpopulation of the input z_t.

[1]The precise timing of theta oscillations and phase-precession within them in hippocampus have attracted much modelling interest but we do not consider them in the present model.

IV. Lostness detection and recovery

A major problem with particle filters having small numbers of particles – and especially a unitary particle – is getting lost. PFs approximate the posterior at each step by a small number of samples, and the UCPF approximates it with a single sample. If the sample set drifts away from the true state, then it becomes difficult or impossible to regain tracking. An approach to dealing this problem used in robotics [13] is to monitor the performance of the filter and heuristically detect when tracking is lost, for example by thresholding a moving average,

$$e_t = \alpha \sum_{i \in \neg o} (z(i)_t - \hat{y}(i)_t)^2 + (1 - \alpha)e_{t-1}, e_0 = 0, \tag{11}$$

where the sum ranges over the non-odometric inputs only (discussed further in section V). If tracking is lost, then priors (and odometry-dependent sensors, discussed in section V) should be disabled and an alternative proposal distribution used, based only on immediate sensor likelihoods.

We hypothesise that the Subiculum-Septum circuit performs such monitoring. We model CA1 as performing partial decoding of the CA3 hidden state into the de-noised, posterior beliefs about the sensors, which are then relayed or fully decoded in ECd. Sub is then well-placed to compare the de-noised CA1 information against the original ECs input, receiving one-to-one connections from both regions. If the posterior CA1 decodings are sufficiently different from the ECs likelihoods for a period of time, this indicates loss of tracking. The cholinergic projections from Sep, activated via Sub, are especially well-placed to disable the CA3 priors as described above, as they are known [10] to disable the recurrent connections in CA3.

V. Treating odometry as noisy GPS observations

Including odometry information in the sensory input z_t is somewhat problematic for an HMM-like architecture. In the Bayesian filter, inputs must be independent of one another conditioned on the hidden states. One allowable type of place sensor would be a *noisy GPS* system, which gives noisy estimates $(\theta_t + \epsilon_t)$ of pose θ_t, being a vector comprised of 2D position and angular heading. However animals do not posses such a sensor, rather they must integrate a series of noisy differential odometry measurements $\Delta\theta_t + \epsilon_t$ to obtain a sense of location. If summed naively, a sequence of heavily error-correlated estimates of pose is obtained. Such correlation violates the Bayesian filter requirements.

To force the place and heading observation inputs to respect HMM semantics, our ECs assumes that that the previous hippocampal output estimate of pose is so accurate as to be perfect, so when added to the latest differential odometry gives an input equivalent to a noisy GPS observation, appropriate for the Bayesian filter. This is a similar assumption as used in the UCPF update: assuming that the previous inference was perfect. Of course – as with the UCPF

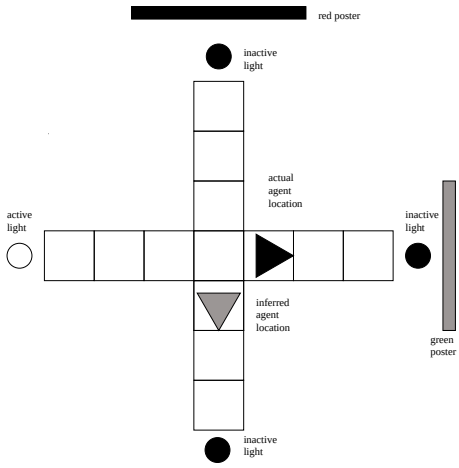


Fig. 2. Plus-maze environment used in the demonstration.

update – the noisy-GPS assumption will break when tracking is lost. We already have a system for lostness detection described above. So its output can be reused to completely disable the noisy GPS input until tracking is regained, in addition to its role in disabling the prior in the UCPF itself.

To provide the capability of switching between full inference (using odometry, other senses, and CA3 priors) and inference from odometry-independent sensors only, care must be taken in setting up the TRBM and its biases in particular. We partition the input vector z_t into a pure sensory component $z_t(\neg o)$ which is independent of all odometry, and a component dependent on odometry, $z_t(o)$. These vectors are extended to $z'_t(\neg o)$ and $z'_t(o)$ respectively, by appending a 1 at the end, to provide separate bias terms. We partition the de-noised version y_t similarly. (This is an orthogonal partition to $(z_t(EC), z_t(DG))$, as both ECs and DG will contain both odometry dependent and independent variables.) We then define a global bias on the hidden TRBM nodes,

$$b_x = sig^{-1}(P(x)), \tag{12}$$

yielding the first-order priors on the hidden states under the Boltzmann distribution,

$$P(x) = \frac{1}{Z} \exp b_x. \tag{13}$$

Next we define separate sets of weights for the three information sources, conditioning on the global bias, so that the joint posterior is

$$\log Z P(x_t, x_{t-1}, z_t) =$$

$$\sum_t (b_x + x'_t W_{x'x'} x'_{t-1} + x'_t W_{x'z'(\neg o)} z(\neg o)'_t + x'_t W_{x'y(o)'} z(o)'_t).$$

As addition in the weight domain corresponds to distribution fusion in the probability domain, the terms in this sum correspond exactly to the factors in

$$P(x_t, x_{t-1}, z_t) = \frac{1}{Z} P(x) P(x_{t-1}|x_t) P(z(o)_t|x_t) P(z(\neg o)_t|x_t). \tag{14}$$

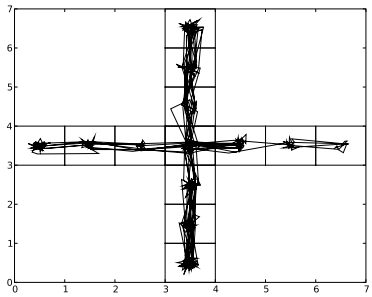


Fig. 3. The first 500 steps around the maze in the simulation. The x and y locations shown here correspond to the top two graphs in figs. 5 and 6.

It is necessary to define the global bias first before learning the other weights, so that the bias terms in the other weight matrices exclude the contribution from the global prior and thus form likelihood terms which may be fused correctly with it. In particular this then allows the odometry-dependent and recurrent prior information to be removed using

$$P(x_t, z(\neg o)_t) = \frac{1}{Z} \exp \sum_t (b_{x,} + x'_t W_{x'z(\neg o)'} z(\neg o)'_t). \tag{15}$$

The terms in this exponent then correspond, as desired, to

$$P(x_t, z(\neg o)_t) = \frac{1}{Z} P(x) P(z(\neg o)_t|x_t). \tag{16}$$

We may write equation 15 as

$$P(x_t, z(\neg o)_t) = \frac{1}{Z} \exp \sum_t (b_x + x'_t (0 \times W_{x'x'}) x'_{t-1} \tag{17}$$

$$+ x'_t W_{x'z'(\neg o)} z(\neg o)'_t + x'_t W_{x'z(o)'} (0 \times z(o)'_t)), \tag{18}$$

where $(0 \times W_{x'x'})$ is equivalent to disabling the recurrent CA3 connections, and $(0 \times z(o)'_t)$ is equivalent to inhibiting all activity in ECs and DG areas receiving odometry-dependent input. Importantly, both of these disablings assume that the source population includes an always-on bias node. In contrast, the effect of the global bias may easily be moved into the firing threshold for CA3 cells.

VI. Plus-maze example

We illustrate the UCPF with Subicular lostness detection and recovery using a simple microworld based on a standard environment used in hippocampal research [1]. Fig. 2 shows a simulated plus-maze, consisting of 13 discrete locations. An agent can move between these locations, and rotate to face the four discrete compass headings. The north and east arms of the maze have coloured posters placed some distance behind them, to provide visual cues when facing in those directions. Two components of the state of the world are the agent's own place, $place \in [0 : 13]$, and heading, $hd \in \{N, E, S, W\}$. In order to illustrate more complex coherent world states, lights are placed at the ends of each arm. One light is on at each time step. When the agent reaches the end of the arm with the active light, it is

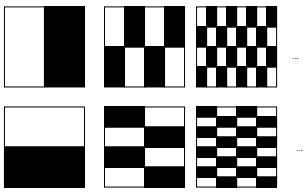


Fig. 4. Walsh-like basis functions used as simplified grid cell receptive fields. Each of the six cells shown here has a Boolean valued receptive field ranging over the 2D space occupied by the maze.

switched off and another light, from a fixed 4-light sequence, is activated. So a third component of the world state is, $light \in \{N, E, S, W\}$, describing which of the four lights is currently on. The complete world state is thus the triple, $(place, hd, light)$.

The present model is concerned only with hippocampal inference, so no learning is modelled, and the agent moves in a random walk. The task is to infer the world state from its noisy sensors. The model network is shown in fig. 1 and weight types summarised in table I.

A. Entorhinal inputs

The agent has $touch$ sensors detecting the presence of boundaries on its *left*, *right* and *forward*; a *lightAhead* sensor which is active when facing the active light; and *red*, *green* and *blue colour* sensors, active when facing a coloured poster. The values of these sensors are all Boolean and are placed into ECs. For example, we have cells $ECs(touch)(left)$ and $ECs(colour)(red)$.

The other two senses are history-dependent, based on odometry information under the noisy GPS assumption. First, medial ECs is known to encode position using grid cells [9] rather than unique place cells – we model a population of six simplified grid cells, $ECs(grid)$, having Boolean valued receptive field functions of 2D location (denoted by r, c, for row and column),

$$EC(grid)(n, 1) = (r \bmod 2^{n+1} < 2^n)(c \bmod 2^n < 2^{n-1})$$

$$EC(grid)(n, 2) = (c \bmod 2^{n+1} < 2^n)(r \bmod 2^n < 2^{n-1}),$$

for $n \in \{1, 2, 3\}$, as shown in fig. 4. Secondly, there are four head direction cells $EC(hd)(dir)$ responding to $dir \in \{N, S, E, W\}$ headings.

The effects of odometry on the ECs grid and head direction cells are modelled only functionally. The previous location estimate is read from ECd, odometry is added, and the resulting locations is encoded and placed in the ECs population,

$$EC(grid)_t = EC(grid)_{t-1} + odom_t$$

$$EC(hd)_t = EC(hd)_{t-1} + gyro_t, \tag{19}$$

where the addition operator is here defined as decoding the grid or head direction cells, adding the Cartesian odometry ($odom$) or angular gyroscopy ($gyro$), then re-encoding new grid or head direction activations. (See [16], [8] for neural-level models of this process.)

B. Dentate Gyrus

DG has handset weights, $W_{EC \to DG}$, which form a sparse encoding of entorhinal feature combinations including: combinations of head direction and light ahead; left and right touches together; left, right and center touches together; and the 13 discrete places in the maze (given by combinations of grid cell activity). We denote these by, for example $DG(hd \otimes lightAhead)(N)$ for the head direction and lightAhead combination responding to north facing and the light ahead; $DG(touch \otimes touch)(left)(right)$ for the combination of two boundaries on the left and right. We write $DG(place) = DG(grid \otimes grid \otimes grid \otimes grid \otimes grid \otimes grid)$, defining place as a combination of grid activations to reduce notation.

C. CA3

The *semantics* (not the weights) of the CA3 cells are set by hand. We specify CA3 cells responding to each of the 13 discrete locations, $CA3(place)$; combinations of place and head direction, $CA3(place \otimes hd)$; the state of the light sequence $CA3(light)$; and combinations of location and light sequence, $CA3(place \otimes light)$ (Thus the micro-domain illustrates CA3 cells responding to simple facts about the world, and to combinations of those facts, cf. [7]).

The weights to CA3 are partitioned into $(W_{x'z'(\neg o)}, W_{x'z'(o)}, W_{x'x'})$ as in equation 15, where $\hat{x} = CA3$, $z(o)$ is the collection of EC and DG cells dependent on odometry, and $z(\neg o)$ is the collection of EC and DG cells that are independent of odometry.

The weights are fit using a version of the wake-sleep algorithm [11] as follows, used for convenience rather than as a biological model of learning. World states $(place, hd, light)$ and noisy sensors (including a genuine noisy GPS training-only sensor) from a random walk of 30,000 steps are recorded. Using the prescribed CA3 semantics, ideal CA3 activations are determined directly from the world states, and global biases b_x computed from their occurrence frequencies. Populations in EC and DG are pooled to form $z(o)$ and $z(\neg o)$, according to whether they include odometry information. In wake steps, ECs_t, $CA3_t$ and the previous $CA3_{t-1}$ are all thus directly observed, and Hebbian learning is performed, conditioned on b_x. In sleep steps, predictions of the next CA3 and next EC are made from the ideal CA3, and anti-Hebbian learning is performed, conditioned on b_x.[2]

D. CA1 and ECd

ECd consists of the same population types as ECs (being the deep layers of the same cortical columns as the ECs units), but stores de-noised sensor values $\hat{y}_t$.

[2]We found that it was important to train all three weight matrices together, in the same wake-sleep cycles. Theoretically they could be trained separately, all conditioned on the same global bias, to learn the likelihoods in equation 14. However in practise the likelihoods are never learned exactly, and variations in weight strength can occur during separated training, which combine to give inaccurate predictions when fused together.

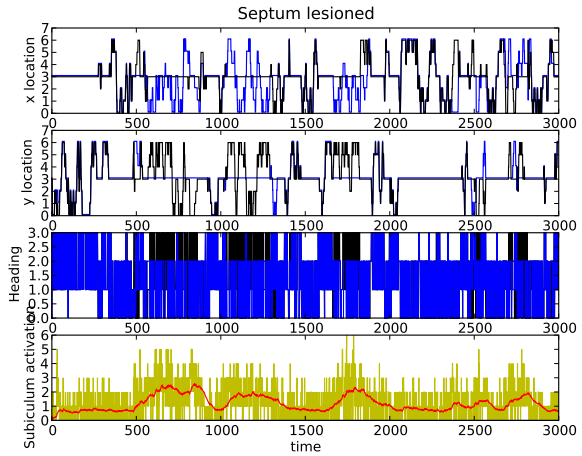


Fig. 5. Results with septum lesioned. Top two graphs show the actual (dark) and inferred (light) x and y locations in the plus maze respectively, during 3000 steps of a random walk. These are the same x, y coordinates shown in 2D in fig. VI. The walk consists roughly of journeys to arms N,E,W,N,S,N,W,S. Tracking performs well until $t = 500$ and then becomes lost – i.e. the actual and inferred lines diverge – until regained by chance alone at $t = 800$. Tracking is lost again at $t = 1000$ and regained by chance at $t = 1300$. Is is lost a third time at $t = 1700$ and regained by chance around $t = 1900$. It is lost a fourth time at $t = 2600$ and regained at $t = 2700$. The third, heading, graph shows actual (dark) and inferred (light) head directions, which have four discrete values for N,E,S,W and also diverge during lostness. The lower graph shows Subicular immediate and averaged activation. This activation is seen to be highest at times when the agent is lost, i.e. when there is divergence between the actual and inferred locations. With Sep lesioned the Sub output has no effect on the inference.

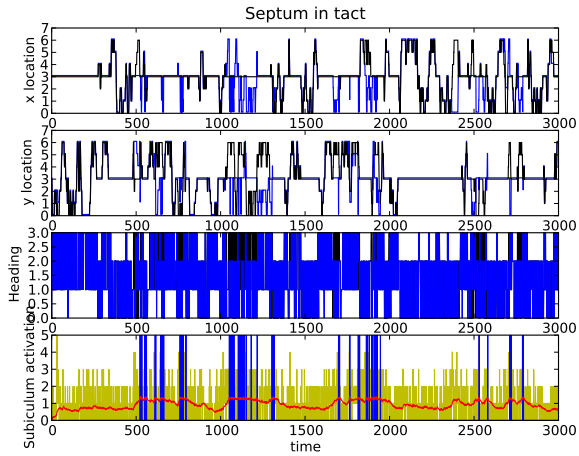


Fig. 6. Results with septum intact. When the moving average subicular output reaches a threshold, it causes Sep to fire (shown as spikes on the lower plot), releasing ACh into CA3 and ECs. This disables the recurrent prior connections in CA3 and inhibits all grid cell activity in ECs, causing inference to proceed using sensory likelihoods only for the next step. This sometimes allows the network to jump to the correct solution. If recovery fails, then subiculum continuous to fire until recovery succeeds. For example around $t = 1100$ there are two long Sep bursts before recovery occurs; in contrast at about $t = 2500$ there are two errors which are corrected almost immediately. Comparing with fig. 5 it can be seen that periods of lostness begin at the around the same times, but last for shorter periods due to the corrections.

A naive decoding method would be to map the unitary TRBM posteriors $\hat{y}(EC)_t$ directly to ECd, using

$$ECd = \hat{y'_t}(EC) = (sig(W^T_{x'y'}\hat{x}'_t) > \frac{1}{2}), \qquad (20)$$

and ignoring the $\hat{y}(DG)_t$ components. This can be done, but the results are poor because the TRBM assumes the sensor units to be independent of one another. However we know that some of the y_t subpopulations – place, heading and whisker *combinations* – in fact always have exactly one active unit, as they represent mutually rival and exhaustive hypotheses. Much of this structure is contained in the $\hat{y}(DG)_t$ components that are discarded in the naive decoding.

We can exploit the structure by using a two stage decoding processes, mirroring the two-stage (EC,DG) encoding. Our CA1 contains all the same population types as our DG, representing combinations of EC features. For example, $CA1(place)$ is the de-noised version of $DG(place)$. (It also contains populations mirroring the EC populations that do not benefit from mutual rivalry, namely $lightAhead$ and $colour$.) The mutual rival features in each population s are then sampled using the winner-take-all rule,

$$CA1(s)(i) \leftarrow (P(CA1(s)(i)) > P(CA1(s)(j)); \forall j \neq i)$$

(such an update may be implemented neurally using winner-take-all feedback.)

The composite features in DG and CA1 are all conjunctions of EC features, so it is simple to handset weights $W_{CA1 \rightarrow ECd}$ to decode them back into the EC basis in ECd. (An ECd cell is on if any of its conjunctive parents is on.) In the case of non-rival populations, CA1 activations are simply relayed to the corresponding ECd populations.

E. Subiculum and Septum

We postulate that Subiculum consists of four parts: a decoder, comparator, an accumulator and an integrator. They are modelled functionally, not neurally. The decoder performs the same computation as ECd, to obtain a copy of the de-noised signals from the partially-decoded CA1. The comparator has cells in one-to-one correspondence with the sense cells in ECs and ECd (excluding place/grid and head direction cells), and which fire when the corresponding ECs and ECd cells differ. (It receives one-to-one input from ECs and decodes input from CA1 in the same way as ECd.) The accumulator computes the integer sum of these differences. The integrator tracks the exponentially weighted moving average of the accumulator to give an overall indication of tracking error. (Accumulation and integration could perhaps take place along the fornix.) When the integrator exceeds a threshold, a single-unit cholinergic Septum unit is activated. The ACh modulation then projects from Sep to ECs and to CA3, having the two effects described by equation 17.

F. Results

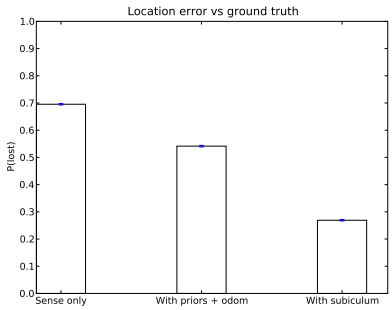
Results of the model running in the plus maze simulation are shown in figs. 5 and fig. 6, with Septum lesioned and intact respectively. Without the septum, there is no recovery from loss of tracking (except when the agent moves by

TABLE I

CONNECTIVITY IN THE MODEL. H=HAND SET WEIGHTS, L=LEARNED WEIGHTS, T=TRANSPOSE OF LEARNED WEIGHTS, R=ONE-TO-ONE RELAY, I=IMMEDIATE INPUTS, O=ODOMETRIC INPUTS. CONNECTIVITY IS SHOWN IN FIG. 1. SUB RECEIVES INPUTS FROM (ECS,CA1).

pop,region	ECs	DG	$CA3$	$CA1$	ECd	Sub
$grid$	o				h	r,h
hd	o			t	r	r,r
$touch$	i				h	r,h
$colour$	i			t	r	r,r
$lightAhead$	i			t	r	r,r
$place$		h		t		
$hd \otimes lightAhead$		h				
$touchCombis$		h		t		
$place$			l			
$place \otimes hd$			l			
$light$			l			
$light \otimes hd$			l			

Fig. 7. Mean errors made about places. (Error bars are vanishingly small as a large amount of simulation data is used.)



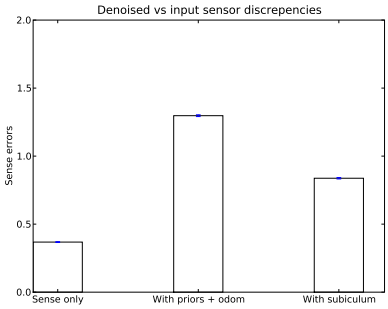
chance to the location that it believes in). For example, at $t = 500, t = 1000, t = 1700$ and $t = 2700$ the ground truth and estimate lines diverge and do not rejoin for several hundred steps. With Septum intact, tracking is able to recover following the detection of failure and deactivation of CA3 recurrent connections via septal ACh. At each of the time above, the Sub-Sep circuit becomes active and the priors are removed. This often leads to burst of activation, as the likelihood-driven estimates may also be wrong, but eventually the correct location is found, the error falls, then tracking returns to normal.

The error introduced by the Sub-Sep circuit is less than the tracking error that it corrects, as shown in fig. 7. We also show the error in a purely likelihood based model, which has both CA3 priors and odometry permanently removed. The transitions priors alone do improve the place error over the likelihood-only model; but a further significant improvement is gained by using the Sub-Sep circuit. Fig. 8 shows the decreased error in the odometry independent sensor denoising when the Sub-Sep circuit is active.

VII. DISCUSSION

The UCPF model combines and extends ideas from Samu et al. [19] and the TRBM hippocampal model of Becker and Hinton [3]. Samu et al. use a hidden CA population (assumed to encompass all of DG,CA3 and CA1) to modulate and

Fig. 8. Discrepancies between input and de-noised sensors. These may be due to genuine de-noising, or to lostness producing incorrect inferences. (Error bars are vanishingly small as a large amount of simulation data is used.)



correct activity in a single EC population of grid cells. Our model also uses hippocampus to modulate the grid cells. However it makes a more detailed distinction between the ECs input and ECd output populations, and adds recurrent temporal priors. Under the UCPF's probabilistic semantics, ECs and ECd are well-defined representations of the noisy input z_t variables and the estimated denopised $\hat{y}_t$ respectively. This allows us to interpret both grid and sensor deep layers as de-noised versions of the superficial layers. These semantics are drawn from the TRBM model, which the present work extends with a more detailed mapping to hippocampus, and the new sampling and lostness inference algorithms. We have used the temporal sequence semantics of the TRBM (i.e. recurrent CA3 connections as priors), for example to learn the places and sensations that typically follow a given place or sensation. This draws together the notion of asymmetric weights as spatial transitions between place cells – used in purely spatial models – with the notion of the weights as compatibilities between variables – used in auto-associative models.

The wake-sleep algorithm in particular learns whatever weights give the best model of the temporal sequence of vector-valued data, so will incorporate both temporal and associative factors. (We note that if each input vector was held constant for a time tending to infinity, then the recurrent weights would tend towards symmetry, thus the symmetric auto-associative models are a limiting case of temporal sequence models.) The wake-sleep algorithm was used here merely to set the computational weights, and not as part of the biological model. However it has previously been suggested as a theory of on-line learning, with learning and anti-learning triggered by tonic ACh in the theta cycle. (In contrast, the septal signal here is the phasic ACh level.)

Three biological connections were not modelled: ECs→CA1, the CA3→DG back-projection, and Sub→EC. We speculate that ECs→CA1 could play a role in learning: CA1 must learn to decode CA3 back to the EC basis, so this connection could act as the supervised training signal. The CA3→DG back-projection might likewise play a role in learning, perhaps part of a wake-sleep-like algorithm where DG sleep states are sampled given CA3 (though full

wake-sleep would also require sampling from ECs).

The mapping would have been more elegant if the Sub comparator received input from ECs and ECd, instead of ECs and CA1. The present model requires Sub to perform the same decoding from the CA1 basis to the EC basis, as well as the comparison itself. An early version of the model assigned the full decoding task to CA1, which then simply relayed all information to ECd, and allowed Sub to act as a simple one-to-one comparator. While this is a possible alternative mapping, we found that using an extra layer of partial decoding produced better results, as it allowed mutual rivalry to be expressed and inferred in CA1 during the partial decoding step. It may be possible to move all such rivalry functionality into CA3, allowing CA1 to act as a pure decoder and relay, though this would make the function of ECd rather redundant, and would also require the presence of grid cells in CA1, rather than the CA1 place cells known in the biology.

The biological mechanism for resetting grid cells is unclear – the model merely requires that lostness detection causes reset. We modelled the Sep→ECs connections as performing this role, but it is possible that the Sub→ECs projection could perform a similar function instead. There is some evidence for septal ACh activating inhibitory cells in EC [24], but it is also known to make ECd cells ignore their inputs and maintain their firing patterns [**?**].

ACh is well-known to play an important role in learning in CA3, as well as switching CA1 from responding mostly to CA3, to responding mostly to ECs. Learning and lostness go together in many circumstances, namely those where the agent is in a novel environment, so is both lost and needing to learn. During novelty, ACh would both disable the priors and trigger learning. However we might not always want to disable odometry during novelty. In contrast, in cases of being lost in a known environment, the removal of both priors and odometry is desired but without learning. Whilst lostness and novelty often go together, they are different concepts, and might be represented by combinations of modulators, for example dopamine is also known to play a role in hippocampal learning [14], but its interactions with ACh are not yet understood.

We have seen that the TRBM uses vector coding, but can learn weights modelling both the temporal and auto-associative aspects of the input vectors; we then constructed a unitary coherent particle filter algorithm operating on the TRBM structure and illustrated how tracking failure may be detected and corrected in a UCPF by monitoring input/output difference and removing the effects of transition and priors and odometry when lost. We have showed how to map the UCPF onto the hippocampal EC-DG-CA3-CA1-EC loop, and how tracking correction could be implemented by the Subiculum and Septum, using ACh to deactivate CA3 recurrent connections.

REFERENCES

[1] E. Tabuchi A.B. Mulder and and S.I. Wiener. Neurons in hippocampal afferent zones of rat striatum parse routes into multi-pace segments during maze navigation. *European Journal of Neuroscience*, 19(7):1923–32, 2004.

[2] A. Arleo and W. Gerstner. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83:287–299, 2000.

[3] S. Becker and G. Hinton. Caching and replay of place sequences in a temporal resticted Boltzman machine model of the hippocampus. In *Cosyne (abstract only, Poster II-56)*, 2007.

[4] Ricardo Chavarriaga, Thomas Strsslin, Denis Sheynikhovich, and Wulfram Gerstner. A computational model of parallel navigation systems in rodents. *Neuroinfomratics*, 3:223–241, 2005.

[5] Nathaniel Daw and Aaron Courville. The rat as particle filter. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 369–376. MIT Press, Cambridge, MA, 2008.

[6] E.R.Wood, P.A.Dudchenko, and H.Eichenbaum. The global record of mempoy in hippocampal activity. *Nature*, 397:613–616, 1999.

[7] E.T.Rolls, J.Xiang, and L.Franco. Object,space and object-space representations in the primate hippocampus. *Jounrnal of Neurophyiology*, 94:833–844, 2005.

[8] Alexis Guanella and Paul F. M. J. Verschure. A model of grid cells based on a path integration mechanism. In Stefanos D. Kollias, Andreas Stafylopatis, Wlodzislaw Duch, and Erkki Oja, editors, *ICANN (1)*, volume 4131 of *Lecture Notes in Computer Science*, pages 740–749. Springer, 2006.

[9] T. Hafting, M. Fyhn, S. Molden, M.B. Moser, and E.I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436:801–6, 2005.

[10] ME Hasselmo, E Schnell, and E Barkai. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region ca3. *Journal of Neuroscience*, 15:5249–62, 1995.

[11] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[12] J.Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Science*, 79:2554–2558, 1982.

[13] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, 2000.

[14] J. Lisman and A.A. Grace. The hippocampal-VTA loop: Controlling the entry of information into long-term memory. *Neuron*, 46:703–13, 2005.

[15] D. Marr. Simple memory: a theory for archicortex. *Philosophical Transactions of the Royal Society B*, 262:23–81, 1971.

[16] Bruce L. McNaughton, Francesco P. Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, pages 663–678, 2006.

[17] Rajesh P. N. Rao. Bayesian computation in recurrent neural circuits. *Neural Comput.*, 16(1):1–38, 2004.

[18] E.T. Rolls, S.M. Stringer, and T.P.Trappenberg. A unified model of spatial and episodic memory. *Philosophical Transactions of the Royal Society B*, 269:1097–1093, 2002.

[19] D. Samu, P. Eros, B. Ujfalussy, and T. Kiss. Robust path integration in the entorhinal grid cell system with hippocampal feedback. *Biological Cybernetics*, 101(1):19–34, 2009.

[20] HE Scharfman. The CA3 backprojection to the dentate gyrus. *Progress in Brain Research*, 163:627–37, 2007.

[21] P.E. Sharp. Computer simulation of hippocampal place cells. *Psychobiology*, 19:103–115, 1991.

[22] S.Kali and P.Dayan. The involvement of recurrent connections in area ca3 in establishing the properties of place fields: a model. *Journal of Neuroscience*, 20(19):7463–7477, 2000.

[23] Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, Cambridge, MA, 2007.

[24] Zhaoyang Xiao, Pan-Yue Deng, Chuanxiu Yang, and Saobo Lei. Modulation of gabaergic transmission by muscarinic receptors in the entorhinal cortex of juvenile rats. *J Neurophysiol*, 109:659–669, 2009.

BRAHMS: Novel middleware for integrated systems computation

B. Mitchinson, T.-S. Chan, J. Chambers, M. Humphries, C. Fox, K. Gurney and T. J. Prescott

***Abstract*— Computational modellers are becoming increasingly interested in building large, eclectic, biological models. These may integrate nervous system components at various levels of description, other biological components (e.g. muscles), non-biological components (e.g. statistical discriminators or control software) and, in embodied modelling, even hardware components, all potentially with different authors. There is a need for middleware to facilitate these integrated systems. BRAHMS, a Modular Execution Framework, fills that need by defining a supervisor-process interface and an (extensible) set of process-process interfaces; authors can write to these interfaces, and processes will integrate as required. Additional benefits include reuse (never code the same model twice), cross-user readability, system-level parallelisation on multi-core or multi-node environments, cross-language integration, data logging, performance analysis, and run-stop-examine-continue execution. BRAHMS employs the nascent, and similarly general purpose, model markup language, SystemML. This will, in future, also facilitate repeatability (same answers ten years from now), transparent automatic software distribution, and interfacing with other SystemML tools.**

I. INTRODUCTION

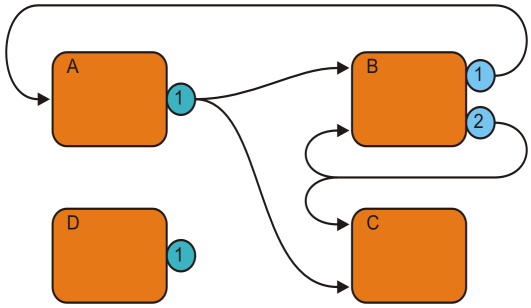


Fig. 1. Example block diagram of an integrated dynamic system built from independent processes. Rectangles represent processes, circles output ports, and arrows links between processes.

Moves are afoot in the world of computational neuroscience towards the construction of large integrated models[1], [2], [3], [4], [5], [6], [7], [8], in the spirit of Daniel Dennett's 'whole iguana'[9]. Multi-region brain models are complex, so researchers necessarily develop them as collections of sub-systems, or 'processes', which must then be 'linked' together to form a 'system' that can be computed, Fig. 1. This aside, modularity is widely considered to be a desirable trait in software design [10]. We define a Modular Execution Framework (MEF)[1] to mean any middleware that facilitates such linkage—a well-known example is Simulink[11]. Neuroscientists are wont to represent processes at different levels of abstraction (biophysical, neuronal, network, control-theoretic, etc.)[12], so it is not generally possible to compute all processes in one computation engine—rather, the task is to link *engines* together[8]. Bespoke integration serves for any single project but, as we will see, a general solution offers much more for less effort, and the startup cost is considerably outweighed by the immediate benefits.

In Section II, we run over the particular challenges and requirements for integrated computation in academic research. We go on in Section III to introduce our proposal, BRAHMS[13], beginning with an overview of its use. In Section IV, we contrast BRAHMS with existing and developing solutions and show it to be well positioned with respect to these. We report on project status in Section V and conclude in Section VI that BRAHMS already offers a solution to most of the identified challenges and will, through planned developments, meet the remainder.

Whilst BRAHMS has its roots in solving problems in and around computational neuroscience, we emphasise that it is not limited by those origins, and we expect it to be of equal interest to researchers in other fields. The problems of integration, of course, become all the more visible when crossing disciplinary boundaries.

II. CHALLENGES AND REQUIREMENTS

A. Varied Development

The primary challenge (as described above) is to integrate software processes, and the primary requirement is, therefore, to offer a middleware platform which will execute processes in concert. Processes may be developed in, on, or by different authors, labs, platforms, programming languages, human languages, programming styles and at different times. The inclusion of non-neuroscience processes generalises the problem across problem domains and technical languages. Without direct communication and refactoring, such disparate offerings will not generally be integrable. Software engineers meet such challenges by offering fixed, public, interfaces to develop against. In this context, an interface requires two facets: one between process and framework, the other between processes; these interfaces must be general (exclude no possibilities), static (or maintain backward compatibility), accessible and available in multiple programming languages on multiple platforms.

The authors are with the Adaptive Behaviour Research Group at the Department Of Psychology, The University Of Sheffield, Western Bank, Sheffield, S10 2TN, UK. Supported by European 6th Framework Grant IST 027819 ICEA and EPSRC Research Grant EP/C516303/1.

Corresponding author b.mitchinson@shef.ac.uk.

[1]We prefer 'Execution' to 'Simulation' since, in general, some processes will not be simulations.

B. Varied Deployment

High-end multiprocessing hardware is increasingly becoming available to research labs, supporting a rapid growth in the development of 'large-scale' models[14] (models with many dynamic states). At the same time, increasing focus on 'embodied modelling'[15], [16] (deployment of behavioural models on robotic hardware) is generating use cases based on low-end hardware. These two trends push the computational envelope at opposite ends, and any solution must be deployable in all these environments. A researcher may develop initially on a desktop machine, for convenience; experimental work may involve large models or parameter spaces and, thus, high-end hardware; embodied models will eventually be deployed on robots, and, particularly if there is an intellectual interest in mobile robotics *per se*, this may mean running on low-end embedded hardware. The requirements are, that a researcher should only have to develop once for such varied deployment cases, and that the middleware should be able to take advantage of the resources of high-end and desktop hardware without becoming unwieldy on low-end hardware.

C. Code Sharing

Computational researchers spend much time authoring software, and disappointingly often this work is repeated in other labs, by other researchers, or even by the same researchers, when documentation, compatible source-code and/or compatible binary code is, or becomes, unavailable. Anecdotal (and more concrete[5], [17]) evidence suggests that '...easier to rewrite it myself than try and obtain/understand/integrate their original code...' is a common story. Any solution should offer great potential for code sharing and reuse, which is to say more than that the code *could* be integrated—it must be *straightforward* to do so. This requires a (preferably, automatic) archiving/distribution mechanism, that shared code be in a form that is immediately usable (rather than having to be compiled, say), and that the solution encourages authors to document their work[18] (facilitating 'intellectual' integration).

In addition, 'background functionality' like parallelisation or data marshalling is neither trivial nor quick to author (and most researchers do not want to become software engineers), so sharing such functionality with all process developers is desirable. Therefore, as much functionality as possible should be subsumed into the middleware—'general' process code should be shared. We use the term 'supervisor' to refer to this shared code, which might, at minimum, be responsible for reading a system document, loading required processes, connecting them together, progressing them through time, collating results, and returning these to the caller.

D. Open Standards

There is more to working with systems than their execution; other possibilities include a system design GUI and an archival/retrieval tool (see also Section II-C). It is, thus, a requirement that the middleware should work with open and extensible data standards. Moreover, the needs of academic research are constantly changing and often cannot wait—the solution must be able to support these changing needs in a timely manner. In practice, therefore, it is a requirement that the solution be open source and extensible by anyone.

E. Adoption

If they are to adopt any proposal, potential users must perceive its advantages to outweigh its costs, both initially and in the long-term. The short-term requirement is met if the startup cost is sufficiently low—interfaces must be few, simple, and well documented; immediately available 'added value' will help to offset this cost. The long-term requirement is met if, in comparison with an equivalent bespoke monolithic design, overall performance does not suffer and per-process development effort is similar or less. Furthermore, an integration solution should by its nature be inclusive, so the solution must be available to all—this means affordable (preferably with no cost). Adoption will be the more willing the more freedom that is given to the developer to do things their way—this means making the interface available in multiple languages, on multiple platforms.

III. BRAHMS

A. Overview

BRAHMS represents part of our commitment to the philosophy and methodologies of neuroinformatics: developing general purpose tools that facilitate large-group working in neuroscience, and sharing and reusing resources. It is an MEF developed in-house during the course of a large-scale, multicentre project (WhiskerBot [3]) which presented many of the challenges outlined in Section II. This was a computational neuroscience project, but from the outset BRAHMS was required to integrate diverse processes (see Section III-H). The design goals of BRAHMS are performance, flexibility and extensibility. BRAHMS is open source and licensed under the GNU General Public License.

BRAHMS operates on systems, Fig. 1, progressing them through time and generating output, Fig. 2, which comprises, in large part, logs of the links between processes. Processes can be developed by dropping state initialization and update code into one of the provided templates (using a programming language that suits the developer). Systems are built from these, processes developed by other researchers, and processes provided with BRAHMS, in just a few lines of script. BRAHMS is invoked to execute these systems, taking advantage of parallel computing resources where available, and the results can easily be pulled into an analysis environment. BRAHMS is not tied to any particular interactive environment but, currently, the support offered for working in Matlab[11] is particularly strong. In time, the library of available processes will increase and additional supervisor functionality will accrue (see Section VI-B for future plans).

B. Systems

A BRAHMS system is a snapshot of a stateful dynamic system in time; that is, a collection of stateful processes and a collection of stateful links connecting them together. A system is loaded from file at invocation, and stored back to

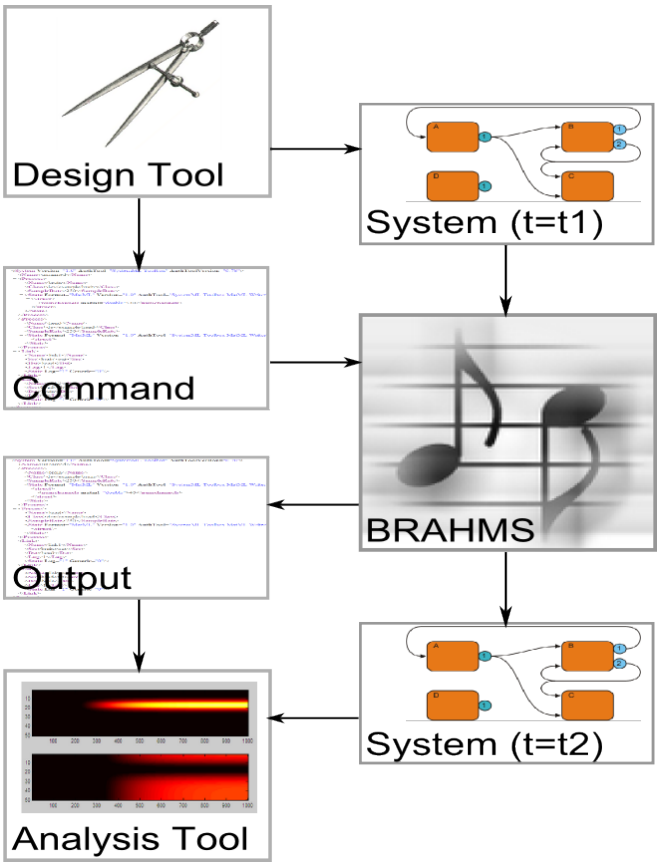


Fig. 2. Typical BRAHMS workflow: system and execution command file design, execution by BRAHMS (progression of the system through time), analysis of final system state and execution output.

file at termination (the file format used is SystemML[19], an open XML-based format for representing stateful dynamic systems). Each process is specified as a class, which indexes an extensible set of process classes each implementing some algorithm, and class-specific state data. A process may publish outputs, each of which specifies a transport protocol (periodic/aperiodic, sample rate) and a data container. Each data container is specified as a class, which indexes an extensible set of data container classes each implementing some data structure, and class-specific state data. Each link specifies a source output (implicitly, thus, a transport protocol and data container class), a destination process, and a transport delay.

The class-specific process or data container state data (termed 'StateML') is understood by design tools for and implementations of those classes (for instance, NeuroML[8] might be the StateML for some neural process classes). These class-specific data are not used by BRAHMS so the set of systems that can be represented is extensible simply by the addition of new process or data classes. Transport protocols are the responsibility of the middleware, however, so the addition of new protocols requires updating BRAHMS.

C. Interfaces

The core of BRAHMS is the supervisor-process interface—C was chosen as the language for this for its durability and interoperability with other languages. Language bindings are provided for C++ and Matlab, and additional bindings are expected to follow (Python bindings are currently under development). The interface as a whole has been designed according to modern API design principles[20], for example with an eye to minimality, extensibility, and backward compatibility. It comprises four aspects, as follows.

First, processes offer an extensible events interface through which the supervisor can invoke process operations (such as initialization, progressing through time). The remaining three aspects are callback interfaces allowing processes to invoke supervisor functionality. They are an implementation of parts of the W3C XML Document Object Model interface[21] (for manipulating StateML, amongst other things), the inter-process interface (for interacting with links), and a general interface of BRAHMS-specific functions.

In addition to the above, data container classes must offer a class-specific interface to allow manipulation of their contents by processes. Inter-process communication then proceeds as follows: the source process publishes a port during initialization, specifying the class and structure of data containers to pass over it, and this forms one end of a link; at run-time, the source process writes data into a container using the class-specific interface, and passes the container into the link; the destination process pulls the container from the other end of the link and reads the contents using the class-specific interface.

D. Modules

Process classes are implemented in modules, in one of the languages for which bindings are provided. A process module must respond to calls from the supervisor on the events interface, performing the requested operations. A simple process module might respond to only two events, first to publish an output, and second to pass data into it on each time step. Data classes are implemented in very similar modules. A data module receives a different set of events from a process module, but otherwise operation is similar. A simple data module might respond to two events, to log its current state for later retrieval, and to return its log to the supervisor. In addition, a data module must offer an interface to its content for use by processes, as mentioned above.

E. Supervisor

The BRAHMS supervisor is authored (in C++) as a standalone executable, so it does not require a virtual machine or scripting engine and is therefore resource-light. It is invoked with a command file, see Fig. 2. It then reads the system file, instantiates the system (by loading modules and passing them their state from the system file), connects processes together, then supervises the progression of the system through time, managing the transport of data through the links. At a predetermined stop time (or following cancellation by the user or by a process), it collates the state of all loaded modules (processes and data containers) and links and writes this back to a system file, also collating logs of each link specified for logging in the command file and writing these

to an output file. Most often, it is this output file of logged inter-process links that is the desired product of an execution.

Automatic parallelisation is provided at a coarse-grained (process) level. Finer-grained parallelisation can be implemented within processes, but it will generally be easier to break processes up and let BRAHMS parallelise them (models that consist of collections of similar objects, like network models, lend themselves particularly to this technique). One implementation of the supervisor, 'Solo', offers lightweight parallelisation in a shared-memory environment using multithreading. Another, 'Concerto', offers multiprocessing for parallelisation over a computing cluster, currently using either TCP/IP or MPI as the communications layer.

The data transport provided by the supervisor might comprise sharing a memory pointer (Solo), or routing a container over ethernet (Concerto), or through a stateful system file to another user, on another platform, at another time. Such transport operations are achieved without onus on the process developer. Background functionality beyond parallelisation includes inter-process data compression (Concerto), windowed data logging, distributed system performance monitoring, and a completely general 'pause & continue' execution model (allowing system snapshots at non-zero time to be coherently stored, exchanged, examined or modified).

F. Accountability

One of the dimensions of Varied Development, above, is time. This means that we should be able to integrate, today, process code that was generated many years ago. However, computing environments change, and researchers forget. BRAHMS offers accountability; that is, everything that bears on the results produced from an execution (details of each loaded library module, external library, of the run-time environment, platform, operating system, etc.) is recorded in an 'execution report'. If a repeat of the execution does not produce identical results, it is possible to identify why (thus, accountability favours repeatability by identifying sources of disagreement).

G. Software Development Kit

A BRAHMS release includes template processes and tutorial examples (corresponding to tutorials in the documentation) authored in all three currently supported languages. Thus, creating a new BRAHMS process involves little more than copying the template for the chosen programming language, and adding the content that performs the actual algorithm intended. Also included is a development version of the BRAHMS 'Standard Library', a collection of processes implementing simple operations (such as sum, product and resample) which are intended to be useful in production systems, whilst doubling as further illustrative material. This library also includes the data container class that will be most useful, 'data/numeric', which is a container for an N-dimensional array of numeric data in a comprehensive (and extensible) range of fixed-bit-width element formats.

Matlab interfaces for the command, system, and output files, and for invocation of the BRAHMS executable, are also include, such that BRAHMS can be called in Matlab using normal function call syntax. These allow the construction of systems from processes and links, and the design of the generic StateML used by the processes in the Standard Library. However, BRAHMS does not know about process state, in general, so additional tools are needed to design StateML for processes that do not use this generic StateML. Interfaces to BRAHMS from other environments are expected to follow in future.

H. Work Experience

The WhiskerBot robot is an embodied model of rat behaviour with an eclectic control model including hardware components (FPGA spiking neural simulators), neural software (leaky integrator models of Superior Colliculus and Basal Ganglia) and non-neural software (heuristic models of fixed behaviours and arithmetic/geometric modules for robot control). The model was developed on high-end desktop hardware and deployed with modification of parameters only on the low-end embedded compute platform of the robot, meeting sub-millisecond real-time constraints consistently. Large parts of the model have been adopted to contribute to the control software of the ScratchBot robot, an artefact of the ICEA project[4]. Other parts are currently in use as part of the large-scale oculomotor control model of the REVERB project[22] which deploys across a compute cluster and a separate robot-control machine. In another aspect of the ICEA project, existing WhiskerBot processes have been interfaced successfully with the Freebots[23] robot simulation environment. BRAHMS has also been chosen as the integration platform for the large European project BIOTACT[24]. Taken together, these use cases illustrate reuse, various dimensions of integration, and varied substrates of deployment.

IV. RELATED PROJECTS

We do not discuss neuroscience-only integration projects, such as NSL[25], NEOSIM and CATACOMB[26], since they do not attempt to solve the integration problem with the level of generality discussed here.

A. Simulink

Simulink[11] has a long history, and is a useful tool for learning about integrated systems. More recently, it offers multi-language support (Matlab, C, C++, Ada, Fortran) but as yet no standard support for parallelisation even within a shared-memory space, and it suffers from computational overhead. The data format is open (though not extensible, since Simulink is proprietary). In the long-term, support may improve in the technical areas where Simulink does not meet the requirements, but it is likely to remain costly, closed source and with a large resource footprint.

B. IKAROS

IKAROS[27] is a project of similar spirit to the BRAHMS project, but has rather different focus. Its positive points include the 'WebUI', which allows real-time monitoring of

system state through a browser, good documentation, and a simple developer interface. But where BRAHMS aims for to meet all the challenges listed above, IKAROS attempts to solve a much more constrained problem, albeit in a straightforward and effective way.

IKAROS is constrained to a single inter-process datatype (2D single-precision matrices), does not support dynamic creation and sizing of outputs based on connectivity, multiprocesses on a single machine only (currently), and the plugin architecture requires the whole system to be rebuilt from source when new modules are added. This last is a particular problem for accountability, since the code that executed to generate archived results is generally no longer available. There is no discussion of bindings for other languages; the IKAROS interface is authored in C++, which might become a problem in the future if the project intended to move towards dynamic loading or accountability. IKAROS also lacks some background functionality available in BRAHMS, though such features can probably, as for BRAHMS, be added without modifying the plugin codebase. In summary, we suspect that IKAROS and BRAHMS are solutions to different, if not quite orthogonal, problems.

C. Large-Scale Modeling Program and MUSIC

The International Neuroinformatics Coordinating Facility (INCF) have recently launched a program to foster infrastructure for researchers working with large-scale neural models. Attendees of the first program workshop[8] noted the large and growing set of neuron simulators available, and agreed on the importance of interoperability and component reuse. Focus was on modularity, particularly the supervisor-process interface, process-process interface, and common file format. Consequently, run-time inter-process communication was also discussed as a necessary future development to integrate computation engines into systems. They also highlighted background functionality (node allocation, communications initialisation) and marshaling of extremely large data sets.

All of these issues are addressed by our proposal. Other interoperability concerns raised in the workshop report are not applicable to BRAHMS; e.g. no application scripting is required since BRAHMS is responsible for procedure. Within the program, a communications library called MUSIC[28] is under development. The MUSIC approach leaves everything but inter-process communication to the process, in stark contrast to BRAHMS, which provides much common functionality. Each approach has its advantages—in particular, large and/or closed-source projects are more likely to suit a MUSIC interface than a BRAHMS front-end (though the latter need not be onerous). In contrast, BRAHMS allows extremely rapid development of powerful cross-platform engines, which MUSIC does not. We do not consider that BRAHMS and MUSIC will be direct competitors, and expect to offer a BRAHMS-MUSIC interface in future.

V. STATUS

BRAHMS has been public since April 2007, and is now approaching version 0.7, expected midsummer 2008. This branch will serve the ICEA, REVERB and BIOTACT projects (for three years). Interface logic is unchanged since December 2007 though minor syntax changes continue as foundations are assured for planned features. At 0.7, syntax will also freeze. Planned features will arrive in upcoming minor branches (0.8, 0.9, ...) towards version 1.0 which will end the initial development cycle. Processes authored against 0.7 will interoperate with future releases.

Solo is now fairly mature, having performed well with only minor changes for some years. Concerto still has alpha status, but is considered sufficiently mature for deployment in the REVERB project. All releases are available for Windows 32-bit, GNU/Linux 32-bit (Ubuntu) and GNU/Linux 64-bit (Debian). We anticipate offering builds for other platforms in time. Aperiodic links, pause & continue execution, and execution reports are not yet fully implemented.

VI. CONCLUSIONS AND FUTURE WORKS

A. Meeting The Challenges

BRAHMS solves the primary challenge of integration across Varied Development by specifying a common, flexible interface in multiple programming languages, against which new computational engines can be developed, and onto which existing computational engines can be imported. It meets the challenge of Varied Deployment through implementation as a lightweight standalone native executable and by allowing processes to be developed in similarly lightweight native code; a version of the supervisor is available without multiprocessing support for a further reduced footprint. BRAHMS offers extensive (and accreting) background functionality in the supervisor, meeting one aspect of the challenge of Code Sharing (a BRAHMS 'hello world' process written in C++ runs to about 25 lines of mostly boilerplate code, yet can distribute its complex processing across massively parallel resources).

BRAHMS supports the pragmatic challenges of sharing process code (by allowing the distribution of pre-compiled binaries rather than source code and by providing accountability) and goes some way to fostering documentation by defining a public process interface (development against a known interface self-documents to some extent, since a naïve reader knows at least some aspects of what a piece of code is intended to do). However, it does not directly address the challenge of sharing process code—see Section VI-B for details of how this will be addressed by future developments. BRAHMS employs Open Standards throughout. Adoption of the pre-release platform continues: the success of BRAHMS as the integration framework for the WhiskerBot project has led to its being chosen for three other major projects, involving varied use cases, and the early adopters have reported finding the workflow agreeable. Indications are that overall performance of systems executed using BRAHMS perform favourably in comparison with monolithic equivalent systems—quantitative metrics will follow in a later report.

New processes benefit from being built into the BRAHMS framework by taking advantage of services provided by the

system, and are constrained in their operation only by the requirements of the supervisor-process interface (i.e. a process is free to interact with the operating system, with hardware, with the user, as required). Integrating existing processes into BRAHMS can be achieved either by 'wrapping' the existing processing engine (contingent on cooperation and/or access to source code) or by meeting the communications interface of the existing software (Freebots was wrapped, for example, by authoring a BRAHMS process that communicated with it over TCP/IP).

None of the other proposals considered meet the requirements for general integration. Some discussed features are, as noted above, incomplete; however, users can begin using BRAHMS immediately and take advantage of feature additions as they become available.

B. Future Work

Above, we mentioned the SystemML file format, which is used by BRAHMS to represent systems. This open file format is a point of interface between BRAHMS and other tools. Beyond that, however, SystemML will also, in future, offer an infrastructure for the publication of processes represented in such systems. Process data in the infrastructure will include specification of parameterisation and of input/output interfaces, as well as of the algorithm itself. The infrastructure will provide archiving, distribution, version control and automatic patching (without breaking accountability) of published processes. The interplay of this infrastructure with accountability will ease the identification and removal of software bugs. This infrastructure will allow BRAHMS to meet the final challenge identified above, that of effective sharing of process code. It will also contribute to the reduction of the problem of algorithmic details becoming lost irrecoverably in undocumented optimised code.

Asked to execute a SystemML document, a naïve installation of BRAHMS will be able to obtain implementations of the specified processes through the SystemML infrastructure and execute the model without human intervention. Thus, processes published to the infrastructure will be immediately available to co-workers. Not only will co-workers be able to execute each other's models, but they will be able to use freely-available expertly-authored components in their own models.

We are committed to completing the development of Concerto, the BRAHMS Standard Library and the Python language bindings, and we expect to develop bindings for Java and Octave in future. In addition, we plan to develop interfaces for the use of BRAHMS from other interactive environments (Octave, for example). Other possibilities include a GUI system designer (operating within the SystemML space entirely, this is actually independent of BRAHMS) and the addition of a service equivalent to the IKAROS WebUI.

A future report will offer efficiency and scaling metrics, including comparison with equivalent monolithic systems.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of the other Adaptive Behaviour Research Group members in testing, and particularly the patience and advice of Martin Pearson as the main end user on the WhiskerBot project.

REFERENCES

[1] P. F. Dominey and M. A. Arbib, "A cortico-subcortical model for generation of spatially accurate sequential saccades", *Cereb Cortex*, 2:153-175, 1992.

[2] J. W. Brown, D. Bullock and S. Grossberg, "How laminar frontal cortex and basal ganglia circuits interact to control planned and reactive saccades", *Neural Networks*, 17:471-510, 2004.

[3] M. J. Pearson, A. G. Pipe, C. Melhuish, B. Mitchinson and T. J. Prescott, "Whiskerbot: A Robotic Active Touch System Modeled on the Rat Whisker Sensory System", *Adaptive Behavior*, 15:223-240, 2007.

[4] *ICEA*, European Union Framework 6 IST-027819, http://www.iceaproject.eu.

[5] J. M. Chambers, *Deciding where to look: A study of action selection in the oculomotor system*, PhD Thesis, The University Of Sheffield, 2007.

[6] J. G. Fleischer, J. A. Gally, G. M. Edelman and J. L. Krichmar, "Retrospective and prospective responses arising in a modeled hippocampus during maze navigation by a brain-based device", *Proc Natl Acad Sci U S A*, 104:3556-3561, 2007.

[7] B. Girard, D. Filliat, J. Meyer, A. Berthoz and A. Guillot, "Integration of Navigation and Action Selection Functionalities in a Computational Model of Cortico-Basal-Ganglia-Thalamo-Cortical Loops", *Adaptive Behaviour*, 13(2):115-130, 2005.

[8] M. Djurfeldt and A. Lansner, *Proceedings of 1st INCF Workshop on Large-scale Modeling of the Nervous System*, Stockholm, Sweden, 2006, *in* Nature Precedings, doi: 10.1038/npre.2007.262.1

[9] D. C. Dennett, "Why not the whole iguana?", *Behavioral and Brain Sciences*, 1:103–104, 1978.

[10] D. L. Parnas, "On the criteria to be used in decomposing systems into modules", *Communications of the ACM*, 15(12):1053-1058, 1972.

[11] Mathworks, *Matlab & Simulink*, http://www.mathworks.com.

[12] K. Gurney, T. J. Prescott, J. R. Wickens and P. Redgrave, "Computational models of the basal ganglia: from robots to membranes", *Trends in Neuroscience*, 27(8):453-459, 2004, doi: 10.1016/j.tins.2004.06.003

[13] B. Mitchinson and T.-S. Chan, *BRAHMS*, http://sourceforge.net/projects/brahms.

[14] M. Djurfeldt, Ö. Ekeberg and A. Lansner, "Large-scale modeling – a tool for conquering the complexity of the brain", *Frontiers in Neuroinformatics*, 2:1, 2008, doi: 10.3389/neuro.11.001.2008

[15] T. J. Prescott, F. M. Montes González, K. Gurney, M. D. Humphries and P. Redgrave, "A robot model of the basal ganglia: Behavior and intrinsic processing", *Neural Networks*, 19(1):31-61, 2006.

[16] B. Webb, *Biorobotics*, AAAI Press, 2001.

[17] R. Chavarriaga, T. Strösslin, D. Sheynikhovich and W. Gerstner, "A Computational model of parallel navigation systems in rodents", *Neuroinformatics*, 3(3):223-242, 2005.

[18] D. L. Parnas, "Software Aging", *in 16th international conference on Software engineering*, Sorrento, Italy, 2004, 279-287.

[19] B. Mitchinson and J. Chambers, *SystemML*, http://sourceforge.net/projects/systemml.

[20] J. Bloch, *How to Design a Good API and Why it Matters*, Javapolis, Antwerp, Belgium, December 12–16, 2005.

[21] Le Hors A. *et al.* (ed.), *W3C Document Object Model Core*, http://www.w3.org/TR/DOM-Level-3-Core/core.html.

[22] *REVERB*, EPSRC Research Grant EP/C516303/1, http://reverb.abrg.group.shef.ac.uk.

[23] ICEA Deliverable D27, http://www.iceaproject.eu.

[24] *BIOTACT*, European Union Framework 7 ICT-215910, http://www.biotact.org.

[25] A. Weitzenfeld, M. A. Arbib and A. Alexander, *The Neural Simulation Language: A System for Brain Modeling*, MIT Press, 2002.

[26] F. Howell, R. Cannon, N. Goddard, H. Bringmann, P. Rogister and H. Cornelis, "Linking computational neuroscience simulation tools : a pragmatic approach to component-based development", *Neurocomputing*, 52-54:289-294, 2003.

[27] C. Balkenius *et al.*, *IKAROS*, http://www.ikaros-project.org.

[28] Ö. Ekeberg, M. Djurfeldt, *MUSIC: Multi-Simulation Coordinator, Request For Comments*, http://www.incf.org, 2008.

Whisker-object contact speed affects radial distance estimation

Mat Evans[1], Charles W. Fox[1], Martin J. Pearson[2], Nathan F. Lepora[1] and Tony J. Prescott[1]

***Abstract*— Whiskered mammals such as rats are experts in tactile perception. By actively palpating surfaces with their whiskers, rats and mice are capable of acute texture discrimination and shape perception. We present a novel system for investigating whisker-object contacts repeatably and reliably. Using an XY positioning robot and a biomimetic artificial whisker we can generate signals for different whisker-object contacts under a wide range of conditions. Our system is also capable of dynamically altering the velocity and direction of the contact based on sensory signals. This provides a means for investigating sensory motor interaction in the tactile domain. Here we implement active contact control, and investigate the effect that speed has on radial distance estimation when using different features for classification. In the case of a moving object contacting a whisker, magnitude of deflection can be ambiguous in distinguishing a nearby object moving slowly from a more distant object moving rapidly. This ambiguity can be resolved by finding robust features for contact speed, which then informs classification of radial distance. Our results are verified on a dataset from SCRATCHbot, a whiskered mobile robot. Building whiskered robots and modelling these tactile perception capabilities would allow exploration and navigation in environments where other sensory modalities are impaired, for example in dark, dusty or loud environments such as disaster areas.**

INTRODUCTION

To operate successfully in nocturnal or poorly-lit environments many animals have evolved non-visual sensory capacities, some of which have yet to be successfully replicated in robots. For instance, rodents, such as mice and rats, have evolved sophisticated tactile sensing systems based around their facial whiskers (known as vibrissae). Behavioural experiments have demonstrated that these animals can make judgements about the tactile properties of objects and surfaces with impressive accuracy. For instance, rats can extract the identity of a 30μm grating texture, in the dark, based on just one to three touches per whisker [5]; can display accurate judgments of a texture within 100 ms of initial whisker contact [3]; and can accurately determine object location in the whisker field [25][26][16]. Similarly, the Etruscan shrew - the smallest living mammal - can recognise and localise prey animals (insects) from a small number of fleeting whisker contacts, sufficient to allow fast and precisely targeted attacks [1]. Processing can also be very fast. For instance in the mouse, whisker contact signals can reach the barrel cortex – from where they can begin to affect processing in behaviour-related areas such as the motor cortex, the superior colliculus, and cerebellum – in just 7 milliseconds [10]. Robots with tactile sensing capabilities approaching those of rodents would excel in environments where many other sensory modalities are impaired. In dusty, smoky or loud environments such as disaster areas and war zones where sensory systems operating in the modalities of sight and sound may be compromised. Whiskers themselves are inexpensive, mechanically simple elements. Whiskers have no sensory elements along their length, deflections are processed by sensors at the base, making them relatively robust to damage. In contrast, an artificial hand will have sensor elements at or near the tips of its digits where they will be much more vulnerable to wear and tear. Together these features make whisker based tactile sensors ideal for mobile robotics.

[1] Active Touch Laboratory, Department of Psychology, University of Sheffield, Western Bank, Sheffield. S10 2TN, UK. *mat.evans@shef.ac.uk*
[2] Bristol Robotics Laboratory, Du Pont Building, Bristol Business Park, Coldharbour Lane, Frenchay, Bristol. BS16 1QD, UK

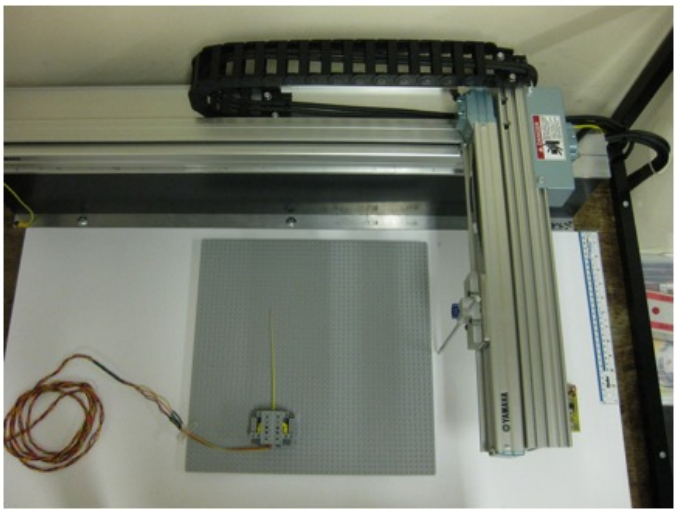


Fig. 1. The XY positioning robot. The whisker was deflected by the robot along the X axis through movements in a clockwise or anticlockwise direction as viewed from above.

Recent years have seen many whiskered robots being developed (for example, SCRATCHbot [20] [21]), addressing numerous aspects of tactile perception (see [22] for a recent review). Previous work has shown that information about texture, distance to contact, and shape can be extracted from signals obtained when an artificial whisker is moved against a surface [12][11][15][7][17][9].

The XY positioning robot

Developing more sophisticated models of whisker-based perception has been problematic. In modalities such as vision and audition it is generally quite easy to present stimuli to a passive sensor on a robot, or images and tones can be simulated and used to train a computational model. There

is no obvious analog for tactile stimuli, and the true nature of whisker-object interactions is too poorly understood to be simulated accurately. Whiskers are especially difficult to simulate, as they have very low mass but high spring constants when modelled as a series of masses on rotational springs, leading to numerical instabilities. Additionally when the parameters of a whisker-object contact become more numerous (e.g. speed and radial distance to contact, surface texture, orientation and softness etc) it becomes very difficult to constrain the contact and generate reliable signals in either simulated or physical robots. For these reasons acquiring sufficient examples of carefully controlled whisker contacts with tactile stimuli to train models and classifiers has proved difficult. We present a novel system for generating large, repeatable, sets of deflection signals from whisker-object contacts. An XY positioning robot is programmed to move objects into an artificial whisker sensor Fig.1. Deflections for the whisker are streamed to a PC, and can be processed in real time to control subsequent robot movement.

Under passive deflections an object moved by the robot arm makes contact with the artificial whisker and deflects the whisker through a large angle. When deflection reaches a critical point the whisker loses friction with the object, deforms and deflects past the object and goes through oscillatory ringing until the energy dissipates and the whisker comes to rest.

However, in addition to passive touch experiments we are also able to use our experimental setup to investigate active sensing. In this case we mimic a control policy that we have observed in rats in our own laboratory whereby the protraction of a whisker ceases rapidly on contact with a surface and whisker then begins to retract [19]. In contrast to the passive case, this policy, which we call Minimal Impingement (MI), keeps the amplitude and duration of whisker deflection within a limited range, and also keeps whisker ringing after contact to a minimum. An additional benefit is that the forces acting on the whisker are much smaller, meaning whisker breakage is less likely – an important consideration for autonomous robotics.

Having established our motivation and methodological direction, the remainder of this paper describes our initial experiments exploring the space of whisker-object contacts, and the feature based classifiers we have developed to reliably encode this space. The classifiers are then verified on a preliminary data set from the SCRATCHbot whiskered robot [20].

METHODS

An XY positioning robot (Yamaha-PXYX, Yamaha Robotics) (see Fig.1) was used to move objects into the whisker. The robot has a movement range of 350×650mm, and can move up to 720mm/s. Repeatability of the robot is ±0.01mm, and the maximum load it can carry is 1.5kg. Objects are carried by the robot into an artificial whisker fixed to the table, as this allows us to control the contact as carefully as possible. Moving the whisker into an object would cause the whisker to oscillate unpredictably during movement between contacts, and as a result each contact would be slightly different. A controller (Yamaha RCX 222, 2-axis robot controller) takes instructions from a PC through an RS232 cable, and the controller interprets the instructions, completes path integration, and drives the motors. Instructions for the robot are generated inside a MATLAB (www.mathworks.com) loop, and can be easily updated during robot operation, for example to react to whisker input.

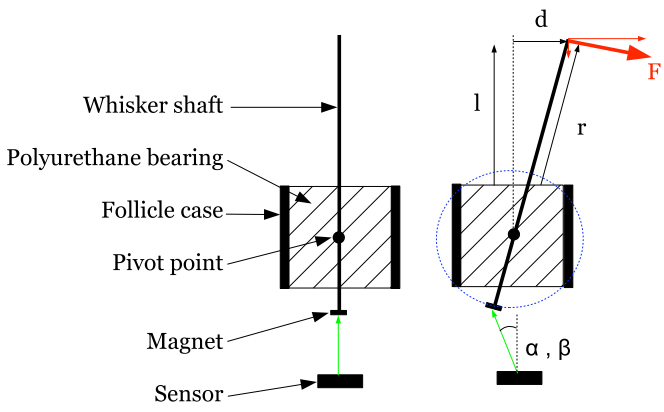


Fig. 2. Diagram of the artificial whisker Hall effect sensor.

The Whisker

A whisker sensor was taken from the SCRATCHbot robot platform [20][22]. Technical details of the whisker can be found in [7]. A tapered, flexible plastic whisker shaft, ≈5 times scale models of a rat whisker (185mm long, linearly tapered from a diameter of 0.5mm at the tip to 3mm at the base), was mounted into an inflexible rubber-filled tube, or 'follicle' case. A tri-axis Hall effect sensor (Melexis MLX90333 [18]) mounted in the follicle case measures the deflection of a magnet fixed to the base of the whisker shaft Fig.2. The Hall effect sensor IC was programmed to generate two voltages, the magnitudes of which being proportional to the two orthogonal displacement angles (α, β) of the magnet from its resting position above the sensor (see Fig. 2). As forces are applied to the whisker shaft, the moment experienced at the base will rotate the magnet around the pivot point, nominally in the centre of the polyurethane bearing. A trigonometric operation in the DSP core of the Hall sensor IC decouples the alpha and beta angles and removes the z-component introduced by the arc of travel of the magnet, as indicated by the blue dotted line in Fig. 2. This operation ensures that the output voltages from the IC are linearly proportional to the tangent component of the alpha and beta angles, or x and y as they will be referred to hereafter.

Data

Deflections of the whisker were transmitted through the hall effect sensors to a LabJack UE9 USB data acquisition card (www.labjack.com) at a rate of 1 kHz for each of the x and y directions. Each trial lasted 4s. This data was

sent to a computer through the BRAHMS middle-ware (brahms.sourceforge.net) for analysis in MATLAB.

Robot control

Minimal impingement was implemented by instructing the robot to move an object into the whisker at a given speed until a deflection threshold is crossed, at which point the robot retracts the object as fast as possible (720mm/s). Temporal latency for the loop is ≈ 300ms from initial contact due to the controller duty cycle.

The task

Preliminary investigations highlighted that the closest contact that could be made by the whisker at without saturating the Hall effect sensor very quickly was ≈80mm from the base. Contacts at less than 5mm from the tip did not deflect the base of the whisker for long enough before slipping past to allow an MI type contact. Therefore, the 185mm length whiskers provide a 100mm range of radial distances. Contact speeds above 216mm/s either cause the whisker to slip past the object before a retraction, or saturates the sensors. 36mm/s was the lower bound on the speed for practical reasons. Contacts were sampled at radial distance intervals of 1mm, and speed intervals of ≈7mm/s over the previously described ranges. In total 101 radial distances and 26 speeds were sampled, giving 2626 different radial distance and speed combinations. Contact combinations were randomly interleaved to limit any affects of whisker properties changing over time. For each contact combination the whisker was deflected by the robot in both a clockwise and anticlockwise directions (-ve and +ve in x, see Fig.1), ensuring that the whisker did not bend over time through repeated unilateral deflections. The experiment was performed twice to generate sufficient data for classification. Data from each trial was stored separately. Deflections from the clockwise robot movement trials (-ve in x) were converted so all data samples were equivalent. Trials were ordered into arrays by speed and radial distance to contact. Each trial was aligned to peak deflection, and shortened to only the 325ms either side of the peak deflection.

Speed of contact confounds radial distance detection

The object properties we chose to manipulate and recognise were radial distance to contact from the base, and contact speed. The task is to recognise these two parameters *simultaneously* under varying conditions.

Previous work [2][14][28] has shown that a rat could encode the radial distance to contact along a whisker by monitoring the magnitude of forces (or moments) at the base. Others have suggested that the increased firing rate of cells in the whisker sensory nerve, for contacts close to the base, could be due to the increased moments at the whisker base, indicating that this moment information is available to the rat [26]. In controlled experiments it has been shown that a rat can discriminate apertures of different widths, down to a difference of ≈3mm [16]. When modelled as a cylindrical elastic beam, the relationship between radial distance to contact r, moment (or torque) at the base M and whisker deflection angle θ can be expressed [13] as;

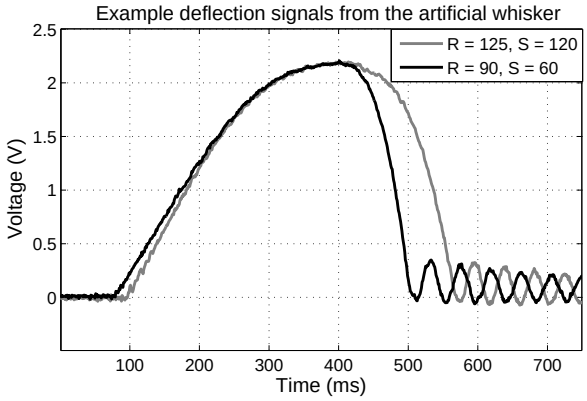


Fig. 3. Example deflection signals from the artificial whisker. Magnitude of deflection, or force, has been used as a discriminator of radial distance to contact. Here the two traces are at different radial distances, but create the same magnitude of deflection.

$$r = 3EI_{\text{base}}\frac{\theta}{M}, \tag{1}$$

where E is the elastic modulus, and I_{base} is the area moment of inertia at the base of the whisker. In this equation the interaction of deflection angle θ and M gives you a value for radial distance to contact. However, the equation would need to be modified to account for situations where contact speed is a variable.

Static beam equations, and analyses relying on instantaneous measures of moments only account for the dynamic properties of objects if two observations are made. If an object collides with a whisker at the same location but at different speeds it will induce different forces at the base. For example, under the right conditions the moment at the base will be the same for a slowly moving object contacting near the base, and a quick object near the tip (see Fig.3 for a demonstration of this).

This ambiguity in the signal cannot be accounted for with a single observation, an additional observation, or invariant feature in the signal, must be found in order to discriminate these two properties of the collision. For example by taking rate of change of moment and θ, or by taking the longitudinal force [24][4].

In the present study we find features in the data that correspond to radial distance to contact and object speed. Successful classification relies either on finding the contact speed before conducting a radial distance estimation, or discriminating two properties simultaneously (e.g. [8]). In the analysis we assess simple feature-based classifiers, and compare the performance of different features in isolation and combination.

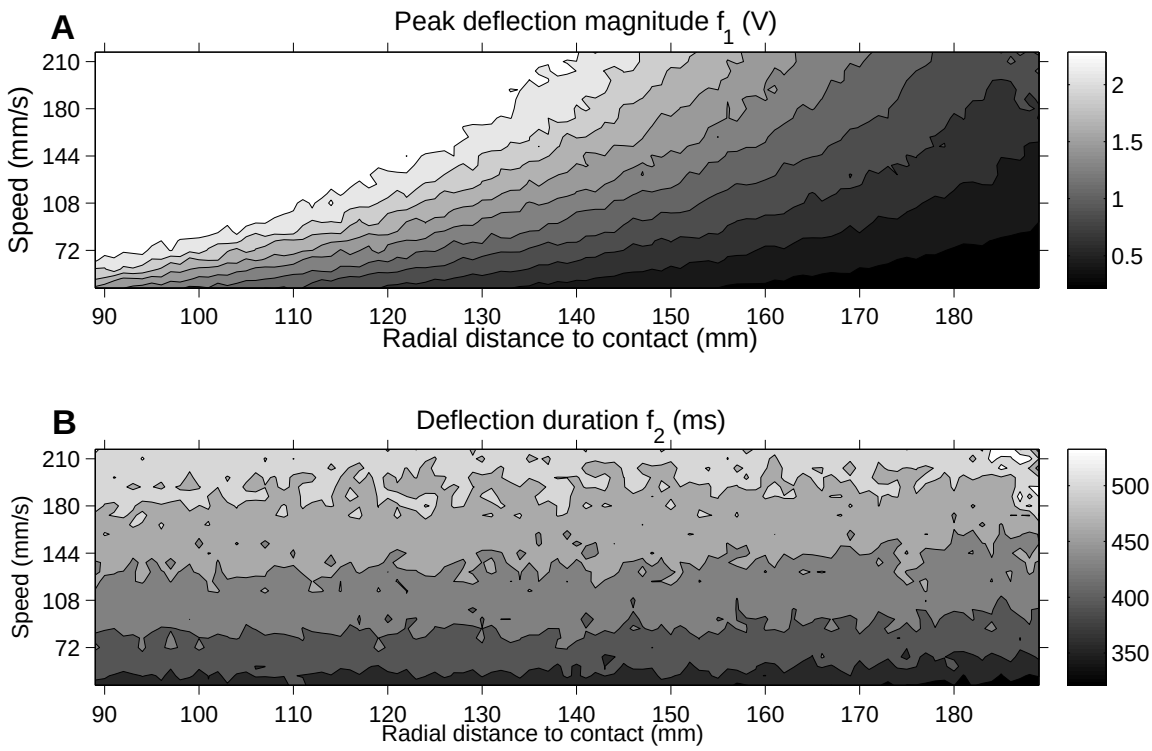


Fig. 4. A contour plot of peak deflection magnitude and duration for each contact. Each point in the image corresponds to a location in the speed-radial distance space which is equivalent in both plots. (a) Peak deflection magnitude f_1, brightness indicates higher deflection magnitude, measured in volts. (b) Deflection duration f_2, brightness indicates greater duration (measured in ms).

RESULTS

The data were separated into training and test sets that were each 2 complete data sets of 26 speeds and 101 radial distances. Signals were placed in the training or test sets at random from the original data. For each feature, classifiers were developed on the training sets, and performance was determined on the test set.

Feature based classifiers

Inspection of the data showed that peak deflection magnitude could be used as a feature for radial distance discrimination at a given speed. Deflection magnitude was taken as the Hall effect sensor output voltage at peak deflection, which is proportional to the bending moment. Feature f_1 can be defined as,

$$f_1 = \max_t \theta(t), \tag{2}$$

where $\theta(t)$ is the deflection magnitude varying with time, measured by the Hall effect sensor in volts. Note that $t(f_1) = t(\max_t \theta(t))$

Similarly, contact speed could be discriminated using deflection duration. Deflection duration was taken as the width of the deflection peak (prominent initial deflection in each trace of Fig.3). Deflection duration was measured using a threshold crossing on the sensor output. When Hall effect sensor output exceeded 0.05V a timer was initiated (t_1), and when Hall output subsequently fell below this threshold the timer was stopped (t_2). Feature f_2 can be defined as,

$$t_1 = \min\{t : \theta(t) \geq \gamma\}, \tag{3}$$

$$t_2 = \min\{t : \theta(t) \leq \gamma, t_2 > t_1\}, \tag{4}$$

$$f_2 = t_2 - t_1, \tag{5}$$

where γ is the threshold and f_2 was measured in ms. Fig.4 shows the object-contact space for f_1 and f_2 in graphical format.

As an additional feature peak rate of change of deflection $\dot{\theta}(t)$ was used. Rate of change of moment $\dot{M}$ and deflection angle $\dot{\theta}$ have each been proposed as biologically plausible alternatives to the absolute values used in elastic beam equations for radial distance detection (Eq. 1) [13] [23]. Since the output of the Hall effect sensor corresponds to deflection magnitude, which is in itself proportional to the bending moment, we found a proxy for rate of change of moment by computing the rate of change of deflection of the whisker, $\dot{\theta}(t)$. To generate $\dot{\theta}$ each input signal was downsampled to 100Hz, a derivative was taken and the peak derivative from the protraction period of each trial was taken as $\dot{\theta}(t)$. Feature f_3 can be defined as,

$$f_3 = max_t \dot{\theta}(t), \tag{6}$$

for each trial the peak $\dot{\theta}$ was taken during the initial deflection of the whisker, $t < t(f_1)$, where $t(f_1)$ is the time at which the f_1 occurs.

A model was generated of the relationship between each feature and the corresponding contact property by fitting a cubic polynomial to the training data in MATLAB, for example to find an estimate of radial distance $\hat{r}$,

$$\hat{r} = a_3 f_1^3 + a_2 f_1^2 + a_1 f_1 + a_0, \tag{7}$$

was fitted to the data with a linear-in-the-parameters regression on the cubic, to find (a_0, a_1, a_2, a_3) by least squares.

In the case of simultaneous radius and speed classification, speed is classified first with f_2, then radial distance is classified with a model fitted to a smaller region of the f_1 or f_3 space.

Classification performance

Fig. 4 shows features f_1 and f_2 for each radius and speed combination. Looking at both plots at the same time shows the interaction of deflection magnitude and duration across the feature spaces. Alone, deflection magnitude f_1 is an ambiguous predictor for radial distance Fig. 4(a). By using deflection duration f_2 as a measure of speed Fig. 4(b), it is possible to limit the region of Fig. 4(a) over which to perform the radial distance discrimination, thus improving classification. The results of the classification are summarised in Table I.

Single feature classification

When used alone, peak deflection magnitude f_1 allows discrimination of 46% of inputs to within 10mm over the 101mm range. Mean error was 13.6 mm. Deflection duration f_2 was capable of classifying 98% of inputs to within 70mm/s for speed, over the 216mm/s range. Mean error here was 20mm/s. Classification of radial distance based on rate of change of deflection magnitude f_3 was successful in 50% of inputs to within 10mm over the 101mm range. Mean error was 15.6mm.

Multi-feature classification

In each case combining classifiers improved radial distance detection. Combining deflection duration f_2 and peak deflection magnitude f_1 results in improved radial distance classification of 74% to within 10mm. Mean error was reduced to 8.4mm.

Combining deflection duration f_2 classification with rate of change of deflection f_3 again results in improved radial distance classification of 54% to within 10mm. Mean error was reduced to 11.4mm.

Histograms of mean errors for classification are shown in Fig.5.

TABLE I

MEAN CLASSIFICATION PERFORMANCE FOR EACH CONDITION OVER DIFFERENT VERIFICATION WINDOWS.

% Correct	f_1	f_1+f_2	f_3	f_3+f_2
$\leq$ 5mm	23.6%	45.2%	28.1%	36.0%
$\leq$ 10mm	46.1%	74.4%	50.0%	54.4%
$\leq$ 15mm	46.2%	74.5%	50.1%	54.5%

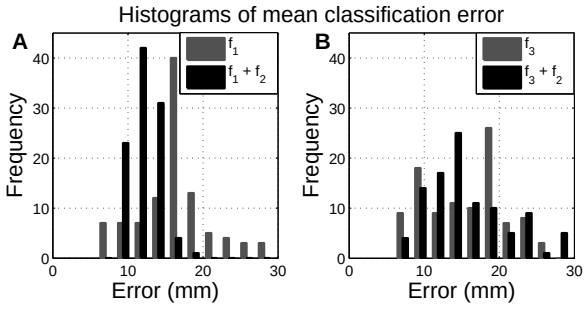


Fig. 5. Histograms of the mean error for radial distance classification in each condition. Combining each classifier with a discriminator for speed improved classification, both by decreasing the mean (moving the histogram peak towards zero) and by reducing variance (width of the histogram).

Verification on SCRATCHbot data

A small dataset was collected on the SCRATCHbot whiskered robot platform [22][21][20]. The robot was kept stationary while it whisked into a vertical pole at 3 different radial distances (70, 100 and 130mm), and 3 whisk speeds (2, 4 and 6Hz). This dataset is too small to do an analysis similar to that conducted on the XY table data, but some useful insights can be gained from it. SCRATCHbot data was inspected to see whether the same features found in the XY table generated signals would be present in a less well constrained situation (see Fig. 7). A key difference between data from SCRATCHbot and that from the XY table is the way whisker speed affects contact duration. Since SCRATCHbot is performing active whisking, increased whisk speed results in a shorter contact duration. However, though the direction of the relationship is reversed, whisk speed still predictably affects contact duration. As in the XY table data, whisking at the same radial distance but different speeds affects peak deflection magnitude (as can be seen in Fig. 7(b)). On SCRATCHbot, as on the XY table, accurate radial distance estimation must involve taking whisker contact speed into account.

DISCUSSION AND CONCLUDING REMARKS

The results presented here compare favourably with the sensory capabilities of rats. Rats have been shown to be capable of discriminating apertures differing in width by 5mm, corresponding to a radial distance difference of 2.5mm per whisker [16]. Since the longest rat whiskers are 50–60mm in length, a 2.5mm discrimination corresponds to an acuity of ≈4–5% of whisker length. A discrimination acuity of 8.4mm on an 185mm whisker corresponds to an acuity 4.5% of whisker length.

It is important to consider that a rat has ≈30 whiskers on each side of its head. Combining information from multiple whiskers may improve the reliability of classification, for example by providing a means to remove independent noise from the signal. This is an approach we hope to investigate in the future.

Rate of change of moment has been proposed [2] as a biologically plausible method of measuring radial distance

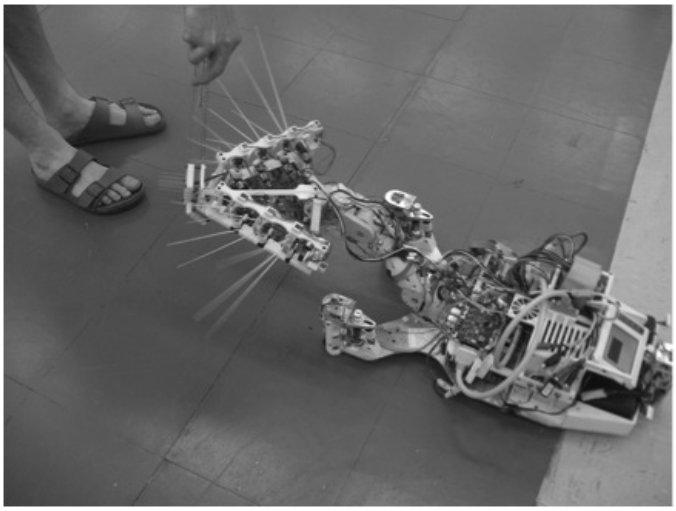


Fig. 6. The SCRATCHbot whiskered mobile robot. To collect data for this experiment the robot platform was kept stationary while it whisked into a pole at varying radial distances to contact, and whisk speed.

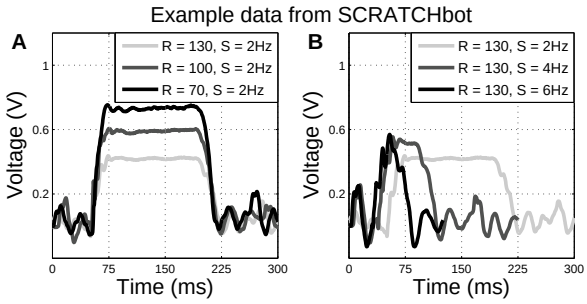


Fig. 7. Properties of the deflections match closely to those from the XY table. (a) 3 deflections at different radial distances (R, in mm), but the same speed (S, in Hz). Peak deflection height varies predictably with radial distance. (b) 3 deflections at the same radial distance but at different speeds. Contact duration varies predictably with speed.

to contact, as biological systems are poor at measuring absolute values. Though performance of a classifier using $\dot{\theta}$ information is poorer than that of a classifier using absolute values, it is still well above chance (chance performance = 20% success to within 10 mm). Classifier performance is improved when combined with an additional speed of contact feature, f_2.

When trying to discern a whisker contact property, such as radial distance to contact, any observation that a classification can be based on will always be affected by the nature of the contact. In the rat whisker system, contacts are adaptively controlled [19]. Though the true purpose of this control is not entirely understood, it is clear that whisker contact speed and force has a large impact on the signals that are produced at the base. We have shown previously that whisker contact speed, and object location have an effect on whisker based texture discrimination [12][7], and that interacting with a surface within certain ranges of speed and at particular regions of the whisker may indicate a 'sweet spot' of whisker object interaction [8]. The rat may be adaptively controlling whisker movement to constrain the ambiguity in the vibrissa deflection signals.

Though in artificial whiskers deflection signals are ambiguous under certain conditions, it is possible to discern certain contact properties from robust features. Subsequent classification can be improved when these contact features are taken into account. Data from the SCRATCHbot robot platform shows that the features found in analysis of XY table data still apply. Going forward we hope to test these classifiers more thoroughly on SCRATCHbot in task based situations. Robust reports of local object features can then be used as inputs to a system of tactile SLAM [6][27].

Steps have been made to explore whisker-object contact space, and in future we hope to utilise the XY table system presently described to investigate further object properties. Knowing contact speed allows better radial distance estimation, which in turn can be used for the discrimination of corners and surface curvature [23]. Discerning the orientation and location of a surface is critical to the discrimination of texture [12]. It is clear that whisker based tactile perception is critically dependent on properties of whisker-object contact. In autonomous robotics, reciprocal sensory motor interaction of this sort can be used to maximise the sensitivity and reliability of a system. Simultaneous speed and radial distance detection is a step towards this goal.

Acknowledgements: The authors would like to thank members of the Active Touch Laboratory at Sheffield (ATL@S) and the Adaptive Behaviour Research Group (ABRG) for their suggestions, knowledge and patience.

References

[1] Farzana Anjum, Hendrik Turni, Paul G H Mulder, Johannes van der Burg, and Michael Brecht. Tactile guidance of prey capture in etruscan shrews. *Proc Natl Acad Sci U S A*, 103(44):16544–16549, 2006.

[2] J Alexander Birdwell, Joseph H Solomon, Montakan Thajchayapong, Michael A Taylor, Matthew Cheely, R Blythe Towal, Jorg Conradt, and Mitra J Z Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *J Neurophysiol*, 98(4):2439–2455, 2007.

[3] George E. Carvell and Daniel J. Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *J Neurosci*, 10(8):2638–2648, 1990.

[4] Hillel J Chiel, Lena H Ting, Orjan Ekeberg, and Mitra J Z Hartmann. The brain in its body: Motor control and sensing in a biomechanical context. *J Neurosci*, 29(41):12807–12814, October 2009.

[5] Mathew E Diamond, Moritz von Heimendahl, Per Magne Knutsen, David Kleinfeld, and Ehud Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[6] Gamini. Dissanayake, Paul. Newman, Steven. Clark, Hugh. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[7] Mat Evans, Charles W Fox, Martin J. Pearson, and Tony J Prescott. Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In T Kyriacou, U. Nehmzow, C. Melhuish, and M. Witkowski, editors, *Proceedings of Towards Autonomous Robotic Systems (TAROS 2009)*, pages 19–24, 2009.

[8] Mat. Evans, Charles. W. Fox, Martin. J. Pearson, and Tony. J Prescott. Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. In *In Proceedings of the International Conference on Simulation of Adaptive Behaviour (SAB), Paris, France*, Preprint available at 5m.org.uk, 2010.

[9] Miriam. Fend, Simon. Bovet, Hiroshi. Yokoi, and Rolf. Pfeifer. An active artificial whisker array for texture discrimination. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings*, volume 2, 2003.

[10] Isabelle Ferezou, Sonia Bolea, and Carl C H Petersen. Visualizing the cortical representation of whisker touch: voltage-sensitive dye imaging in freely moving mice. *Neuron*, 50(4):617–629, 2006.

[11] Charles Fox, Mat Evans, Martin Pearson, and Tony J Prescott. Towards temporal inference for shape recognition from whiskers. In Subramanian Ramamoorthy and Gillian M. Hayes, editors, *Towards Autonomous Robotic Systems*, pages 226 – 233, 2008.

[12] Charles W Fox, Ben Mitchinson, Martin J Pearson, Anthony G Pipe, and Tony J Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.

[13] Venkatesh Gopal and Mitra J Z Hartmann. Using hardware models to quantify sensory data acquisition across the rat vibrissal array. *Bioinspir Biomim*, 2(4):S135–45, 2007.

[14] Makoto. Kaneko, Naoki. Kanayama, and Tsuji Tsuji. Active antenna for contact sensing. *Robotics and Automation, IEEE Transactions on*, 14(2):278–291, Apr 1998.

[15] DaeEun Kim and Ralf Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55:229–243, 2007.

[16] David J Krupa, Matthew S Matell, Amy J Brisben, Laura M Oliveira, and Miguel A Nicolelis. Behavioral properties of the trigeminal somatosensory system in rats performing whisker-dependent tactile discriminations. *J Neurosci*, 21(15):5752–5763, 2001.

[17] Nathan.F Lepora, Mat Evans, Charles. W. Fox, Kevin Gurney, and Tony.J Prescott. Naive bayes texture classification applied to whisker data from a moving robot. In *Proceedings of the 2010 international joint conference on Neural Networks*, pages 2970 – 2877, 2010.

[18] Melexis. www.melexis.com/assets/mlx90333_datasheet_5276.aspx.

[19] Ben Mitchinson, Chris J Martin, Robyn A Grant, and Tony J Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.

[20] Martin J. Pearson, Ben Mitchinson, Jason Welsby, Tony G Pipe, and Tony J. Prescott. Scratchbot: Active tactile sensing in a whiskered mobile robot. In *In Proceedings of the International Conference on Simulation of Adaptive Behaviour (SAB), Paris, France*, 2010.

[21] Tony J. Prescott, Martin J. Pearson, Charles W. Fox, Mat. Evans, Ben Mitchinson, Sean Anderson, and Anthony G Pipe. Towards biomimetic vibrissal tactile sensing for robot exploration, navigation, and object recognition in hazardous environments. In *In Proceedings of 4th IARP workshop: Robots for risky Interventions and Environmental Surveillance- maintenance (RISE)*, 2010.

[22] Tony J Prescott, Martin J. Pearson, Ben Mitchinson, J Charles W. Sullivan, and Anthony G. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.

[23] Joseph H. Solomon and Mitra J Hartmann. Extracting object contours with the sweep of a robotic whisker using torque information. *The International Journal of Robotics Research*, 2009.

[24] Maik C. Stüttgen, Stephanie Kullmann, and Cornelius Schwarz. Responses of rat trigeminal ganglion neurons to longitudinal whisker stimulation. *J Neurophysiol*, 100:1879–1884, August 2008.

[25] Marcin Szwed, Knarik Bagdasarian, and Ehud Ahissar. Encoding of vibrissal active touch. *Neuron*, 40(3):621–630, 2003.

[26] Marcin Szwed, Knarik Bagdasarian, Barak Blumenfeld, Omri Barak, Dori Derdikman, and Ehud Ahissar. Responses of trigeminal ganglion neurons to the radial distance of contact during active vibrissal touch. *J Neurophysiol*, 95(2):791–802, 2006.

[27] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, March 2002.

[28] Naohiro. Ueno, Mikhail.M. Svinin, and Makoto. Kaneko. Dynamic contact sensing by flexible beam. *Mechatronics, IEEE/ASME Transactions on*, 3(4):254–264, Dec 1998.

Naive Bayes texture classification applied to whisker data from a moving robot

Nathan F. Lepora, Mat Evans, Charles W. Fox, Mathew E. Diamond, Kevin Gurney and Tony J. Prescott

***Abstract*— Many rodents use their whiskers to distinguish objects by surface texture. To examine possible mechanisms for this discrimination, data from an artificial whisker attached to a moving robot was used to test texture classification algorithms. This data was examined previously using a template-based classifier of the whisker vibration power spectrum [1]. Motivated by a proposal about the neural computations underlying sensory decision making [2], we classified the raw whisker signal using the related 'naive Bayes' method. The integration time window is important, with roughly 100ms of data required for good decisions and 500ms for the best decisions. For stereotyped motion, the classifier achieved hit rates of about 80% using a single (horizontal or vertical) stream of vibration data and 90% using both streams. Similar hit rates were achieved on natural data, apart from a single case in which the performance was only about 55%. Therefore this application of naive Bayes represents a biologically motivated algorithm that can perform well in a real-world robot task.**

I. Introduction

Many rodents that are nocturnal or crepuscular rely on their whiskers for foraging and navigation in poor light. Their tactile acuity for sensing textures can rival the human fingertip [3]. A working model for this discrimination is that as the rodent whisks across an irregular surface, the relative motion between the whisker and the surface induces a vibration along the whisker shaft that is characteristic of the texture [4]. The kinetic aspects of this vibration are encoded in neuronal activity, from which an internal discrimination is made. Electrophysiological recordings from the barrel cortex of anaesthetized and awake rats support that there is sufficient information in the temporal patterns of neuron firing to allow effective discrimination of whisker vibrations [4]–[7].

To examine possible computational processes underlying this discrimination, time series data from an artificial whisker attached to a moving robot was used to test texture classification algorithms. This approach of implementing a complete system from stimulus to sensor to computation has been applied to whisker-based systems [1], [8]–[10], but only some reported quantitative results [1], [10]. These latter two studies used a spectral classifier to discriminate textures [11], using the whisker vibration power spectrum to distinguish textures with a template matching algorithm. Here we analyzed the data from Ref. [1] using a classifier based instead on the raw whisker vibration signal. Moreover, rather than matching 'features' of the data, as in template-matching, we used the statistical properties of the vibrations to characterize textures. This classification was achieved using a naive Bayes classifier.

Our motivation for using this classifier was that it relates to an influential proposal for the neural computations that underlie decisions about sensory stimuli. For a two-choice classification, the naive Bayes classifier is equivalent to using a (log) likelihood ratio, which Gold and Shadlen said 'is a natural currency for combining sensory evidence obtained from... multiple samples in time' [2]. For two or more choices, the naive Bayes classifier can be interpreted as integrating evidence for each alternative until a decision is made after a given time, which is similar to a number of proposals for how humans and animals make decisions [13]–[15]. Meanwhile, from a machine learning perspective, Naive Bayes classifiers are known for their ease of implementation and effectiveness, despite their naive assumption that samples are statistically independent (see *e.g.* [16], [17]).

We found that naive Bayes is a powerful classifier of robot whisker vibration data that usually achieves hit rates of over 80% on test data. Therefore this simple, biologically motivated algorithm for sensory discrimination can generally achieve good performance when implemented in a robot undergoing a real-world task.

II. Materials and methods

A. Data collection

The texture data was taken from a previous study of an artificial whisker attached to a moving robot [1]. The following methods briefly summarize the data acquisition. Further details can be found in the original reference.

The whisker sensor consisted of a flexible plastic (Acrylonitrile butadiene styrene) whisker shaft (200mm long, 2mm diameter) mounted at its base into a short, polyurethane rubber filled, inflexible tube called a follicle case. The plastic used for the whisker shaft was chosen for its flexibility, appropriate mechanical match to scaled-up biological whiskers, and suitability for rapid prototyping. A magnet was bonded to the base of the whisker shaft so that it was positioned directly above a tri-axis Hall effect sensor (Melexis MLX90333) in the assembled follicle case/whisker shaft on the whisker mount [1, Fig. 1]. The sensor integrated circuit was programmed to generate two voltages with magnitudes linearly proportional to the tangential component of the two

N. F. Lepora, M. Evans, C. W. Fox, K. Gurney and T. J. Prescott are with the Adaptive Behavior Research Group, Department of Psychology, The University of Sheffield, Western Bank, Sheffield S10 2TN, UK. (email: n.lepora@shef.ac.uk, mat.evans@shef.ac.uk, charles.fox@shef.ac.uk, k.gurney@shef.ac.uk, t.j.prescott@shef.ac.uk).

M. E. Diamond is with Tactile Perception and Learning Laboratory, Cognitive Neuroscience Sector, International School for Advanced Studies (SISSA), 34014 Trieste, Italy and Italian Institute of Technology - SISSA Unit (email: diamond@sissa.it).

Fig. 1. The robot.
An iRobot Roomba vacuum cleaner was used as a platform for the experiments. The whisker was mounted on the front of the robot, angled down to make constant contact with the floor.

orthogonal displacement angles (X and Y) of the magnet from its resting position above the sensor.

An iRobot Roomba vacuum cleaner (iRobot, 8 Crosby Drive, Bedford, MA 01730) was chosen as the platform for these experiments (Fig. 1). This robot was ideal for the task as it could also be a candidate system for any behavioral output from the texture classification. The whisker was mounted on the front of the robot using epoxy resin, directed at a 45 degree azimuth from the forwards direction of travel and with a slight downwards elevation sufficient to make constant contact with the floor during movement. Thus the whisker transduced a constant stream of deflection information as the robot moved over the floor.

Four surfaces were chosen for classification: two carpets of different roughnesses, a tarmac surface and a vinyl surface. These surfaces were chosen because they were appropriately generic for a real world experiment and they provided a range of surface types that were sufficiently similar to make classification difficult. Two primary behavioral conditions were chosen with the robot moving either anticlockwise only, or clockwise only. To demonstrate the difficulty of classification without knowledge of the robot motion, data were also recorded for each surface during the Roomba's 'spot' cleaning programme, which consisted of a series of (externally) unpredictable clockwise, anticlockwise and forward motions.

As the robot moved the whiskers were swept across the floor. Any deflections of the whisker were transmitted through the hall effect sensors through a LabJack UE9 USB data acquisition card (www.labjack.com) at a rate of 2kHz for each of the X- and Y-directions. This data was sent to a computer through the BRAHMS middleware (brahms.sourceforge.net) for analysis in MATLAB (www.mathworks.com). Example data from the four floor surfaces is shown in Fig. 2 for the four clockwise and four anti-clockwise trials, each of duration sixteen seconds.

B. Analysis methods

1) Quantized time series: The data are discrete time series of the sensor voltages produced by deflection of the artificial whisker attached to the robot. As this sensor measures both the horizontal and vertical deflections of the whisker, at each sample time there are two distinct sensor voltages x and y associated with these deflections. These voltages are measured at times $t_i = (i-1)/f_s$, where i runs from one to the number of samples n and f_s is the sampling frequency.

For analysis, it will be convenient to quantize the time series values x_i into discrete intervals. Suppose that all signals under consideration vary between minimum $x_{\min}$ and maximum $x_{\max}$ values, with range $\Delta_x = x_{\max} - x_{\min}$. This range is binned into N equal-width intervals, where x is in the qth bin if

$$x_{\min} + \frac{q-1}{N}\Delta_x \leq x < x_{\min} + \frac{q}{N}\Delta_x, \qquad (1)$$

and if $q = N$ the upper bound also includes $x_{\max}$. The continuous-valued time series $x_1, \cdots, x_n$ then becomes a discrete-valued time series $q_1, \cdots, q_n$ with each q_i an integer between one and N. In a similar way, a discrete-valued time series $w_1, \cdots, w_n$ can be constructed for the samples y_i associated with the vertical whisker deflection. Quantization intervals of width 10mV were used here, compared with an approximate 3V spread of sensor measurements.

2) Measurement distributions: The classifier examined here relies on using the probability distributions of the individual time series values, calculated from the empirical frequency with which each measurement value occurs in example (training) data of each texture. If n_q is the total number of times that the value q occurs in the quantized time series, then the normalized frequency is $p_q = n_q/n$. As $\sum_{q=1}^{N} n_q$ equals the total number of samples n, the sum of all normalized frequencies equals one and can be interpreted as a probability distribution.

Denoting the four textures by $\mathrm{T}_1, \cdots, \mathrm{T}_4$ (for rough carpet, smooth carpet, tarmac and vinyl flooring, respectively), the conditional probability of a quantized measurement q from the texture T_l is

$$P(x|\mathrm{T}_l) \equiv P(q(x)|\mathrm{T}_l) = \frac{n_q(\mathrm{T}_l)}{n}. \qquad (2)$$

Here the probability distribution for the sensor measurements, $P(x|\mathrm{T}_l)$, is considered to represent the probability of a measurement x being in the interval $q(x) = q$ given it is from the texture T_l.

In practice, it was necessary to smooth the normalized frequencies n_q to correct for sampling bias in the training data. Without this smoothing, the few samples in the tails of the distributions can lead to large errors in estimating the probabilities, which deteriorates the performance of the classifier. All inferred conditional probabilities were thus

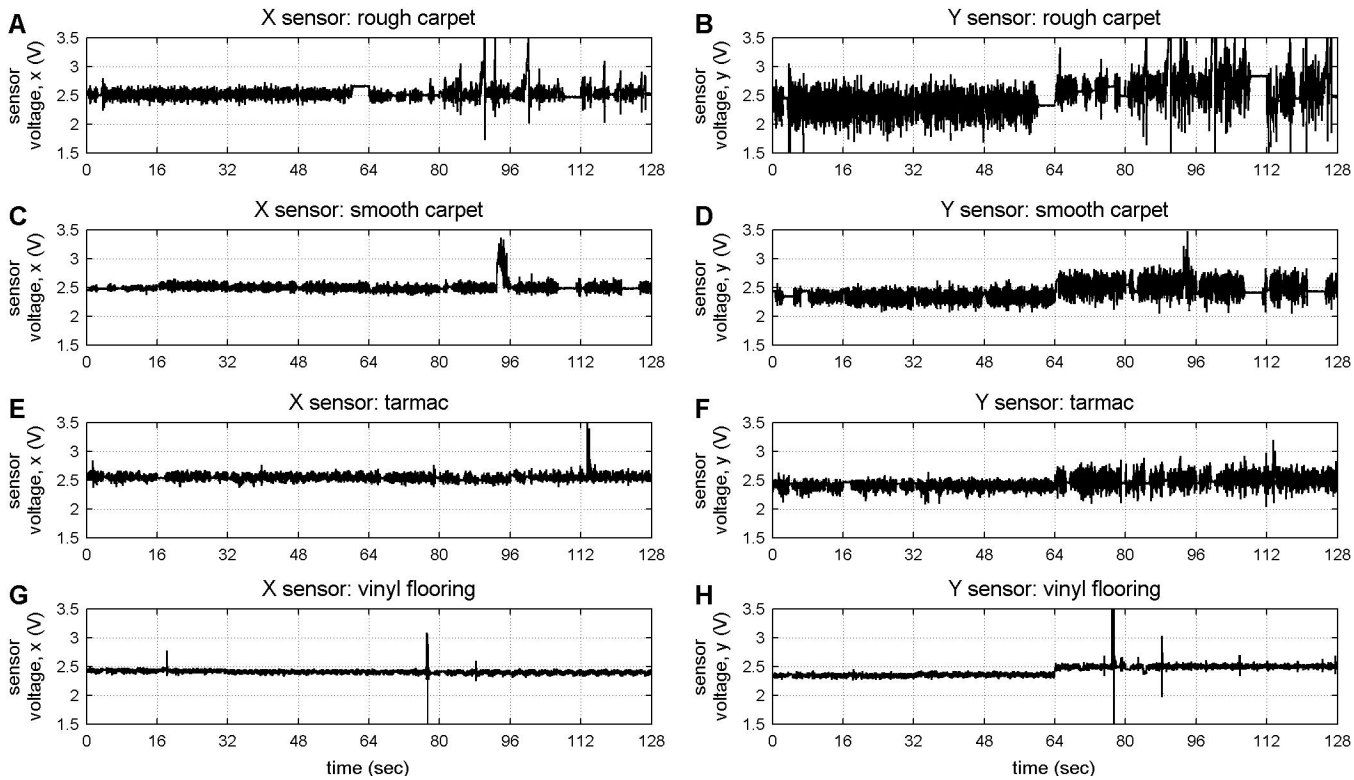


Fig. 2. Example data for four textures
The above data for the four floor surface textures was collected in eight trials each of length sixteen seconds, with the four initial trials for clockwise motion and the four final trials for anticlockwise motion.

convolved with a Gaussian smoother of width $\sigma = 100$mV (10 quantized intervals), which improved classification performance while smoothing on a relatively small scale compared to the overall spread of data.

3) Bayes classifier (single measurement): The method used here is based on: (a) Supposing the conditional probabilities of the measurements $P(x|\mathrm{T}_l)$ are known from estimating them on training data of all textures. (b) Then, given a measurement x, finding for the most probable texture from which it arose. This method compares the conditional (posterior) probabilities $P(\mathrm{T}_l|x)$ for each texture given a measurement x, which are related to the above estimated conditional probabilities by Bayes Theorem,

$$P(\mathrm{T}_l|x) = \frac{P(x|\mathrm{T}_l)P(\mathrm{T}_l)}{P(x)}, \tag{3}$$

where $P(\mathrm{T}_l)$ are the (prior) probabilities of having a particular texture, $P(x)$ are the (marginal) probabilities of measuring x given no information about the textures. The conditional probability $P(x|\mathrm{T}_l)$ is now considered a likelihood of a texture from a measurement.

The classifier finds which texture T has maximum *a posteriori* (MAP) probability given a measurement x. By Eq. 3,

$$\mathrm{T} = \arg\max_{\mathrm{T}_l} P(\mathrm{T}_l|x) = \arg\max_{\mathrm{T}_l} \frac{P(x|\mathrm{T}_l)P(\mathrm{T}_l)}{P(x)}, \tag{4}$$

where arg max denotes the argument of the maximum, namely the texture for which $P(\mathrm{T}_l|x)$ is maximal.

The following arguments assume there is no prior information about which texture is being measured, so that the priors $P(\mathrm{T}_l)$ are all equal and can be ignored when evaluating arg max over the textures. Moreover, since the marginals $P(x)$ are independent of the textures they can also be ignored in the arg max. Finally, it will be convenient to use the logarithm of the posterior probability, and since $\log(x)$ (monotonically) increases with x it also does not affect arg max. Putting these arguments together, the texture classification in Eq. 4 is equivalent to finding

$$\mathrm{T} = \arg\max_{\mathrm{T}_l} \ \log P(x|\mathrm{T}_l), \tag{5}$$

which is just the maximum over the likelihoods determined from the training data given a new measurement x.

4) Naive Bayes classifier (multiple measurements): A potentially more reliable method for classifying textures is to make the decision over many time samples rather than a single value. The arguments in the previous section are unaffected if the single measurement x is replaced by a series of values. Then the equal-prior Bayes texture classification from Eq. 5 becomes

$$\mathrm{T} = \arg\max_{\mathrm{T}_l} \ \log P(x_{n_s}, \cdots, x_{n_f}|\mathrm{T}_l), \tag{6}$$

where n_s and n_f are the start and finish of the window.

An important simplification occurs if the measurements are assumed independent at each time $t(i)$ across the window. Then the overall conditional probability of a series of

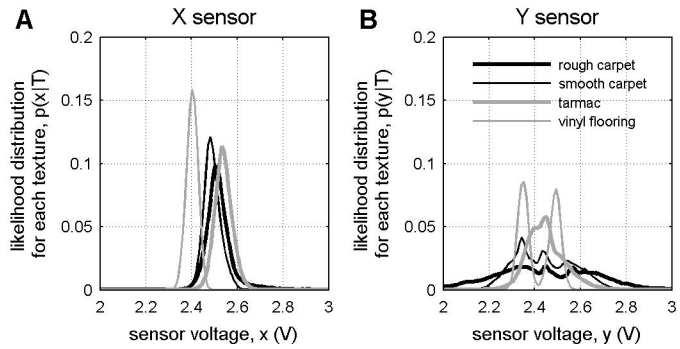


Fig. 3. Conditional probabilities of texture measurements.
The conditional probabilities were calculated from the empirical frequencies with which the measurement values occur in training data for each of the four textures. Each distribution has been smoothed to reduce sampling bias. Data from the X-sensor (panel A) and Y-sensor (panel B) were considered separately.

measurements given a texture factorizes into a product of conditional probabilities for each individual measurement

$$P(x_{n_s}, \cdots, x_{n_f}|\mathrm{T}_l) = P(x_{n_s}|\mathrm{T}_l) \times \cdots \times P(x_{n_f}|\mathrm{T}_l). \quad (7)$$

Consequently, the classification in Eq. 6 can be rewritten as

$$\mathrm{T} = \arg\max_{\mathrm{T}_l} \sum_{i=n_s}^{n_f} \log P(x_i|\mathrm{T}_l), \quad (8)$$

Thus the most probable texture is found from the argument of the maximum over the summed log-likelihoods for the series of measurements.

5) Combined X- and Y-sensor classification: Rather than classifying the X- and Y-sensor data individually, it is possible to make a joint classification using both sensors together. Considering the two sensors to take independent measurements, then the overall conditional probabilities for two measurements also factorize into a product of the individual X- and Y-sensor conditional probabilities

$$P(x, y|\mathrm{T}_l) = P(x|\mathrm{T}_l)P(y|\mathrm{T}_l), \quad (9)$$

Then the classification rule in Eq. 8 can be rewritten as

$$\mathrm{T} = \arg\max_{T_l} \sum_{i=n_s}^{n_f} \log P(x_i|\mathrm{T}_l) + \log P(y_i|\mathrm{T}_l). \quad (10)$$

Thus the most probable texture is found from the argument of the maximum over the summed log-likelihoods for the individual sensor measurements.

III. Results

A. Probability distributions from the training data

Training data from the four clockwise and four anticlockwise trials (Fig. 2) was pooled to determine the probability distributions for the measured X- and Y-sensor voltages. The initial 8 seconds of each trial was used for training data, and the final 8 seconds of each trial saved for later validation of the classification algorithms.

The X- and Y-sensor training data was found to vary between 1.36V and 4.83V. This range was separated into 347 bins of width 10mV, and the measurements quantized into these intervals (Methods, Eq. 1). The number of measurement values within each bin was then totalled to estimate the conditional probabilities of sensor voltage values for the four textures (Methods, Eq. 2).

The resulting probability distribution for each texture is shown in Fig. 3A for the X-sensor and Fig. 3B for the Y-sensor. Notice that the X-sensor probabilities are single peaked, with the means and variances roughly compatible with a visual inspection of the measurements in Fig. 2A,C,E,G. On the other hand, the Y-sensor probabilities have more complicated shapes with two or more peaks. This distribution shape is caused by systematic differences in the response of the Y-sensor to clockwise and anticlockwise motion, as is also visible in Fig. 2.

B. Texture classification on X- and Y-sensor validation data

Given data from an unknown texture, the conditional probability distributions plotted in Fig. 3 can be used as a basis of a classification of which texture is most probable for this data. For validation, the data was separated into discrete segments of fixed temporal window size over which the texture is determined. In general the first half of each trial was used for training and the second half for (holdout) validation, which seemed natural in relation to how a robot or animal might be taught and also allowed the training data to include data correlated beyond the window size. Then the

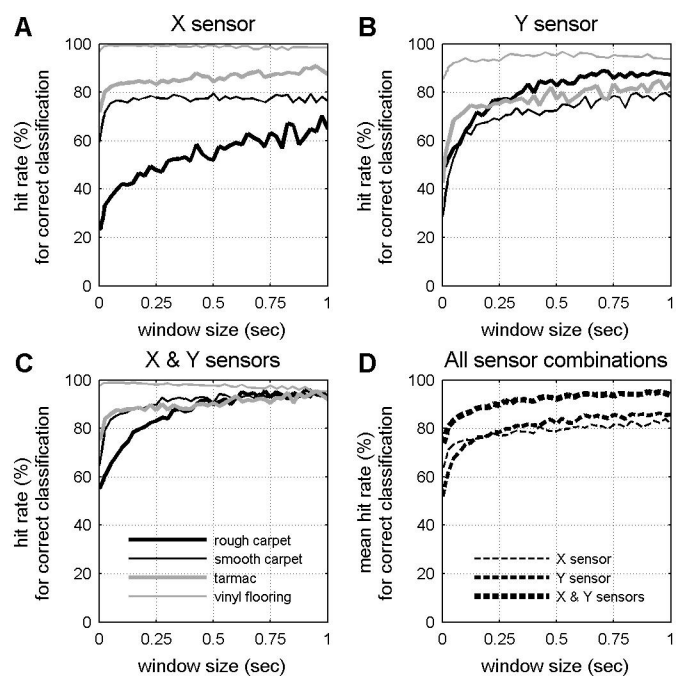


Fig. 4. Hit rates of correct texture identification.
The percentage of correct classifications was evaluated over validation data of the four textures. Each hit rate is shown as a function of window size over which the classification was made. Data from the X-sensor (panel A) and Y-sensor (panel B) were considered separately, followed by combining the X and Y measurements in a single classifier (panel C). The single and dual sensor classifications were then compared by their mean hit rates (panel D)

TABLE I

CLASSIFICATION OF X-SENSOR VALIDATION DATA (0.5S WINDOW). MEAN HIT RATE = 79%

Validation data	Rough carpet	Smooth carpet	Tarmac	Vinyl
Rough carpet	52%	22%	26%	0%
Smooth carpet	18%	79.5%	2.5%	0%
Tarmac	10%	5.5%	84.5%	0%
Vinyl	1%	0%	0%	99%

TABLE II

CLASSIFICATION OF Y-SENSOR VALIDATION DATA (0.5S WINDOW). MEAN HIT RATE = 82%

Validation data	Rough carpet	Smooth carpet	Tarmac	Vinyl
Rough carpet	82.5%	8.5%	1%	8%
Smooth carpet	8%	72.5%	19.5%	0%
Tarmac	1%	19%	77%	3%
Vinyl	2.5%	0%	2.5%	95%

TABLE III

CLASSIFICATION OF X & Y-SENSOR VALIDATION DATA (0.5S WINDOW). MEAN HIT RATE = 92%

Validation data	Rough carpet	Smooth carpet	Tarmac	Vinyl
Rough carpet	90%	8%	3%	0%
Smooth carpet	8%	90.5%	1.5%	0%
Tarmac	0%	9.5%	90.5%	0%
Vinyl	2.5%	0%	0%	97.5%

naive Bayes classifier (Methods, Eq. 8) considered the log-likelihood values for the four candidate textures from the conditional probability distributions at the same measurement value in the training data. These log-likelihoods were summed across the window for each of the four textures, and the maximum one gave the classified texture.

The proportion of correct classifications, or hits, across validation data for each of the four textures is shown in Fig. 4A for the X-sensor and Fig. 4B for the Y-sensor. The classification was clearly a success, with the data from both sensors reaching a good mean hit rate of about 80%. Even though some hit rates improved and some worsened with increasing window size, the mean hit rate improved steadily across the range of window sizes shown (Fig. 4D). This mean hit rate improved substantially as the window size was increased from 0.5ms (single sample) to 100ms, and then modest gains occurred thereafter. Notice also that the two sensors had problems with different textures: the X-sensor classifier was most mistaken for rough carpet, while the Y-sensor classifier was worst on smooth carpet.

Details of the texture classification and misclassification are shown in Table I for the X-sensor data and Table II for the Y-sensor data at a fixed window size of 0.5 seconds. The diagonals of each table were the percentage of correct classifications over the validation data for each texture. Consistent with Fig. 4A,B, vinyl was excellently classified ($> 90\%$) for either sensor, while rough carpet was the most difficult to classify with the X-sensor ($\sim 50\%$) but well classified ($\sim 80\%$) with the Y-sensor. The mean hit rate was calculated over the diagonal of the table, and was about 80% for either sensor. The off-diagonal elements represent how often each texture is misclassified as another texture and can be thought of the degree of confusion. For example, the X-sensor classification confuses rough carpet with tarmac 26% of the time and almost never confuses anything with vinyl. The confusion between rough carpet and tarmac is consistent with visual inspections of the data (*c.f.* Figs 2A and 2C) and the probability distributions (Fig. 3A), both of which are quite similar by eye.

To illustrate when the classifiers make mistakes, Fig. 5 shows the texture classification applied to Y-sensor validation data for the four textures. Hits are shown on the plot by coloring the data black and misses in gray. It is interesting that the instances when a texture was misclassified often coincides with artifacts in the data. For example, the rough and smooth carpets were mainly misclassified on dead-zones and the vinyl texture is misclassified on jumps. In a sense, therefore, the classification algorithm is not really making mistakes, but is instead identifying correctly where the data does not look like a typical example of that texture.

C. Improved classification by combining X- and Y-data

Rather than using validation data from the X- and Y-sensors individually, it is possible to perform a classification using both sets of data together (Methods, Eq. 10). Because more evidence is used in the classification, the combined

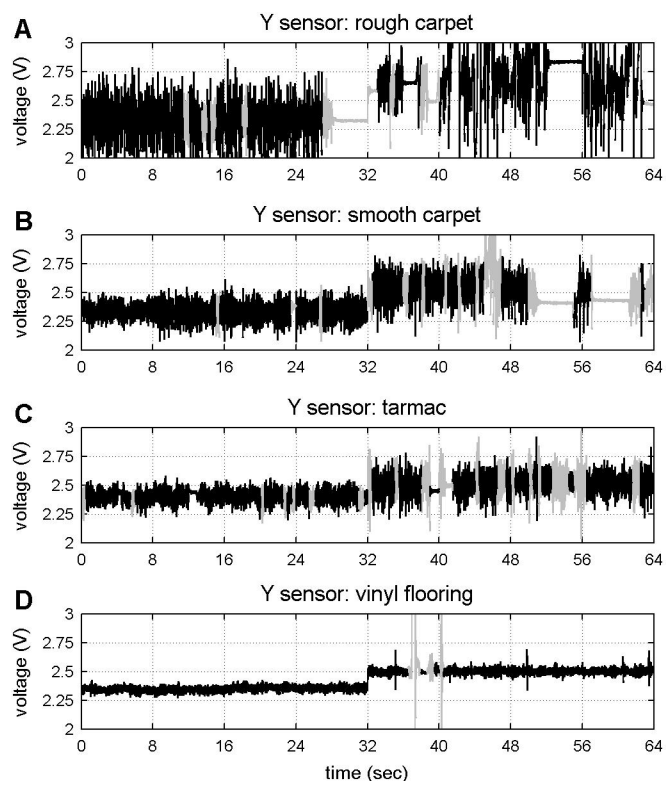


Fig. 5. Examples of misclassification.
The panels show the performance of the classification over a 0.5 second window applied to Y-sensor validation data of the four textures. Hits are shown in black and misses in gray.

TABLE IV

CLASSIFICATION OF X & Y-SENSOR VALIDATION DATA (0.5S WINDOW). ARTIFICIAL TRAINING DATA; NATURAL VALIDATION DATA. MEAN HIT RATE = 68.0%

Validation data	Rough carpet	Smooth carpet	Tarmac	Vinyl
Rough carpet	67%	32%	1%	0%
Smooth carpet	2%	97%	1%	0%
Tarmac	3%	9.5%	87.5%	0%
Vinyl	23%	9.5%	47%	20.5%

TABLE V

CLASSIFICATION OF X & Y-SENSOR VALIDATION DATA (0.5S WINDOW). NATURAL TRAINING DATA; NATURAL VALIDATION DATA. MEAN HIT RATE = 81%

Validation data	Rough carpet	Smooth carpet	Tarmac	Vinyl
Rough carpet	88%	12%	0.5%	0%
Smooth carpet	8%	91.5%	0.5%	0%
Tarmac	6%	2.5%	91.5%	0%
Vinyl	7%	0%	40%	53%

X- and Y-sensor classifier should be more accurate. In particular, a main cause of misclassifications of the individual sensor data was when a sensor picked up an artifact such as a dead-zone or jump. Using data from both sensors together gave more chance of receiving reliable information from at least one of the sensors when the signals are hard to discriminate.

The hit rate of correct texture identification is shown in Fig. 4C for the combined X- and Y-sensor classifier. The combined classification is clearly a substantial improvement over the results for the individual sensors in Figs 4A,B, with all textures reaching an excellent correct identification of about 90% for window sizes of 0.5 seconds or more. The classification of vinyl has suffered slightly compared to that by just the X-sensor, but this is more than compensated by the significant gains in classifying the other textures. These improvements can be seen more clearly in a comparison of the mean hit rate for the X-sensor, Y-sensor and combined XY-sensor classifier in Fig. 4D: the combined sensor classification reaches a mean hit rate of about 90% compared to about 80% for the individual sensors.

Details of the texture classification and misclassification for the combined XY-sensor classification are shown in Table III for a window size of 0.5 seconds. The percentages of correct classifications are shown down the diagonal, and as all are approximately 90% the mean is also about 90%. The percentages of misclassifications are shown by the off-diagonal elements. Roughly speaking, if the individual X- and Y-sensor classifiers confuse two textures, then these are mistaken to a lesser-degree by the combined XY-sensor classification.

D. Application to 'natural' data

Thus far the classification algorithm has been both trained on and validated against data from artificial, stereotyped clockwise or anticlockwise robot motion (with the first half of each trial used for training and the second half for validation). A more difficult task is whether the texture classification is still successful on 'natural' motion, taken from the Roomba's internal programme of a series of (externally) unpredictable clockwise, anticlockwise, stationary and forward motions.

The first classification was for artificial training data and natural validation data (results in Table IV), applying the same combined XY-sensor classifier as in the previous section. The classifier performed excellently on smooth carpet (97%) and tarmac (87.5%), adequately on rough carpet (57%) and terribly on vinyl flooring (20.5%). The mean hit rate (68%) is shifted down relative to that for the artificial validation data (Table III, 92%), mainly by the poor performance on vinyl. Curiously, none of the other textures are confused with vinyl, even though it is frequently confused with rough carpet (23%) and tarmac (47%).

Natural training and natural validation data were then considered, using training data from the first half of each natural trial and validation data from the second half. The results of the combined XY-sensor classification text are shown in Table V. The classification was now excellent ($\sim 90\%$) on all textures except for a moderate performance on vinyl (53%).

Therefore, other than for vinyl, the classification generalized well from artificial to natural data, and even performed adequately if trained on the artificial data and applied to the natural data. Unfortunately, vinyl was not well-recognized on natural data, which was a surprise considering that it was most easily classified for the artificial motion. However, closer inspection of the natural vinyl data revealed a systematic difference between the initial and final four trials of the X sensor data (Fig. 6A) that was absent in the artificial data (Fig. 2G). This difference explains the poor performance (20.5%) when classifying vinyl using artificial training and natural validation data, as the absence of this important feature in the training data made recognition of the natural data impossible. The moderate performance (53%) on natural training and validation data is more difficult to explain, and

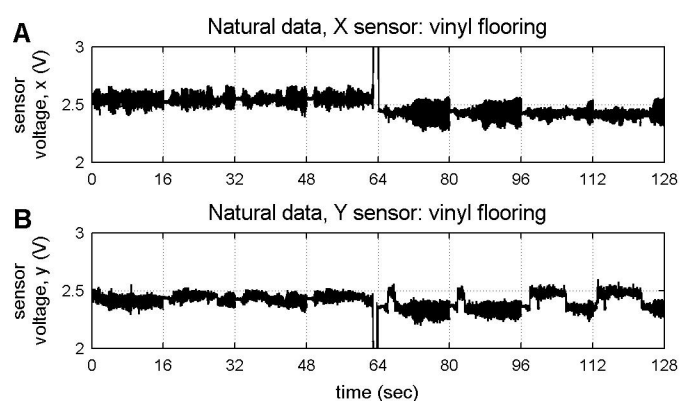


Fig. 6. Natural data for the vinyl texture.
The data was collected in eight trials each of length sixteen seconds, where the robot performed 'natural' motion according to its internal program.

could relate to a limitation of the naive Bayes classifier, such as not using the clearly visible temporal correlations in the natural vinyl data (Fig. 6) or a correlation between the X and Y sensor measurements.

IV. Discussion

To examine the underlying processes involved in sensing surface textures with whiskers, data from an artificial whisker attached to a moving robot was used for testing possible classification algorithms. This data was previously analyzed using a template matching method on the whisker vibration power spectrum [1]. Here a naive Bayes classifier was used on the raw vibration signal: this method estimates the most probable texture to have produced a series of measurements, given estimated likelihoods for single measurements from training data of the textures. The naivety refers to using only the likelihood from single measurements, which ignores any temporal structure in the data. Our reason for using this classifier was that it relates to proposed mechanisms for neural decision making [2], [13]–[15].

The naive Bayes classifier gave excellent results on data from four different textures (rough carpet, smooth carpet, tarmac, vinyl flooring). For stereotyped clockwise and anticlockwise motion, the algorithm could reliably recognize textures with a mean hit rate of about 80% using single stream data from either the X or Y whisker deflection. If both X and Y data streams were considered together, the mean hit rate improved to a very impressive 90%. It was important that enough data was being used to make a decision, with temporal window sizes of about 500ms being necessary to give these hit rates. Furthermore, the times when incorrect decisions were made tended to coincide with artifacts such as dead zones or jumps, consistent with the algorithm then identifying correctly where the data does not look like the texture on which it has been trained.

The classifier was also applied to 'natural' motion, taken from a series of (externally unpredictable) clockwise, anticlockwise, stationary and forward motions. Although there were now problems with vinyl flooring, the classifier could still recognize the other three textures with 85-90% success.

A. Comparison with the spectral template-based classifier

The robot whisker data was previously examined in a study that used a spectral template-based classifier [1]. Spectral classifiers characterize data from the profile of their power spectra. In the previous study, training data of each texture was separated into 400ms segments, from which a template was estimated from the mean power spectrum for each texture (Fig. 7, Ref. [1]). The performance of the template matching was then analyzed on validation data of each texture. First, the power spectrum over each 400ms segment of validation data was found, followed by a determination of the mean square error from each of the texture templates, which then classified the data segment according to the template with the lowest error.

Theoretically, there are two (related) differences between the spectral method and our use of the naive Bayes classifier: (a) The naive Bayes classifier was applied to the raw whisker deflection data, whereas the spectral classifier uses the frequency spectrum of the whisker signal captured by the power spectra; (b) The naive Bayes classifier does not use any information about statistical dependance of the signals at different times, whereas the spectral classifier uses *only* information about the statistical dependance, represented by the power spectrum. Hence, we view the two methods as complementary approaches, since each uses a quite different component of the information available from the whisker. This is important because both methods could in principle be combined together to give a more reliable classifier than either on its own.

The differences in performance between the two classifiers can be seen by direct comparison. For stereotyped texture data from clockwise or anti-clockwise motion, the spectral classifier achieved an overall mean hit rate of 72% (clockwise) and 64% (anticlockwise) using a data from just the X-sensor, whereas naive Bayes achieved about 80% over both motions with either the X- or Y-sensor. Therefore the naive Bayes classifier does seem to be moderately more reliable when the hit rates over all textures are averaged.

Note though that the two classifiers have problems with different textures. For X-sensor data, the spectral classifier had most difficulty with smooth carpet (33% and 54% hit rates) and performed well on rough carpet (78% and 64%) while the naive Bayes classifier performed well on smooth carpet (79.5%) and found rough carpet most problematic (52% hit rate). Moreover, for natural data, the spectral classifier performed well on all textures (hit rates 60-77.5%) except smooth carpet (11%), while the naive Bayes classifier (both sensors) performed well on all textures (hit rates 88-91.5%) except vinyl (53%). This behavior is consistent with the two methods being complementary in their performance in addition to their theoretical basis.

B. Relation to biological decision making

Our use of the naive Bayes classier in the present study is motivated by a proposal for the neural computations that underlie decisions about sensory stimuli. Based upon measurements of neural activity in monkeys performing perceptual decision making (*e.g.* [12]), Gold and Shadlen suggested that the logarithm of the likelihood ratio ($\log \mathrm{LR}$) provides a natural currency for forming a perceptual decision [2]. The likelihood ratio applies only to choices between two alternatives, say H_1 and H_2, with a decision rule that given a sequence of measurements $x_1, \cdots, x_n$,

$$\sum_{i=1}^{n} \log \mathrm{LR}_{1,2}(x_i) = \sum_{i=1}^{n} \log \frac{P(x_i|\mathrm{H}_1)}{P(x_i|\mathrm{H}_2)} \gtrless 0, \qquad (11)$$

where greater than zero supports hypothesis H_1 and less than zero H_2. This decision rule is identical mathematically to the naive Bayes MAP rule in Eq. 8 for two textures. Crucially, both naive Bayes and the log-likelihood ratio are based on

assuming that the measurements x_i are independent over time. In this sense, we interpret naive Bayes' as a direct extension of Gold and Shadlen's proposal to the situation with three or more alternatives. Such a mechanism is known more generally as evidence accumulation or Bayesian integration and has formed the basis for a number of proposals for neural decision making [13]–[15]

Therefore we see the present whiskered robot study as an initial stage in clarifying tactile decision making processes in rodents. We showed that a simple, texture classification algorithm, which is closely related to a number of proposals for perceptual decision making, is an excellent classifier of textures for a robot moving in (its) natural environment. This suggests a functional hypothesis for tactile decision-making in rodents based on a similar evidence accumulation strategy. The log-likelihoods could be stored from previous experience of the textures, and then a decision reached on a new stream of sensory experience by accumulating evidence for each of these texture 'memories'.

A complementary step to investigate such a proposal would be to examine electrophysiological recordings from barrel cortex to see whether whisker vibrations can also be distinguished with similar computational methods. If successful, these studies taken together would give reasonable circumstantial evidence for a related form of decision making in rodents. That being said, a true test of the neural processing could only be achieved by comparing the computational results to the performance of an awake, behaving rat. A behavioral performance that was neither more nor less accurate than that achieved by the classifier on barrel cortex activity would provide strong evidence that the rat uses such a decision making strategy.

Acknowledgments

The authors would like to thank members of the Active Touch Laboratory at Sheffield, the Bristol Robotics Laboratory, and the Tactile Perception and Learning Laboratory in SISSA for discussions. This work was supported by EU grants ICEA (IST-027819) and BIOTACT (ICT-215910).

References

[1] M. Evans, C. W. Fox, M. J. Pearson and T. J. Prescott, "Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task," *Proc. 10th Conference Towards Autonomous Robotics Systems*, Ulster, U.K., Aug. 2009.

[2] J. I. Gold and M. N. Shadlen, "Neural computations that underlie decisions about sensory stimuli," *Trends Cogn Sci.*, vol. 5 (1), pp. 1016, Jan. 2001.

[3] G. E. Carvell and D. J. Simons, "Biometric analysis of vibrissal tactile discrimination in the rate," *J Neurosci.*, vol. 10 (8), pp. 2638–48, 1990.

[4] E. Arabzadeh, S. Panzeri, M. E. Diamond, "Whisker vibration information carried by rat barrel cortex neurons," *J Neurosci.*, vol. 24 (26), pp. 6011-20, Jun. 2004.

[5] E. Arabzadeh, S. Panzeri, M. E. Diamond, "Deciphering the spike train of a sensory neuron: counts and temporal patterns in the rat whisker pathway," *J Neurosci.*, vol. 26 (36), pp. 9216-26, Sep. 2006.

[6] M. von Heimendahl, P. M. Itskov, E. Arabzadeh, M. E. Diamond, "Neuronal activity in rat barrel cortex underlying texture discrimination," *PLoS Biol*, vol. 5 (11), e. 305, Nov. 2007.

[7] A. Lak, E. Arabzadeh, M. E. Diamond, "Enhanced response of neurons in rat somatosensory cortex to stimuli containing temporal noise," *Cereb Cortex.*, vol. 18 (5), pp. 1085-93, May 2008.

[8] A. K. Seth, J. L. McKinstry, G. M. Edelman, and J. L. Krichmar, "Texture discrimination by an autonomous mobile brain-based device with whiskers," *Proc. 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA, April 2004.

[9] M. Fend, "Whisker-based texture discrimination on a mobile robot," In *Lecture notes in computer science. Proc. 8th European conference on artificial life (ECAL).* Berlin: Springer.

[10] C. W. Fox, B. Mitchinson, M. J. Pearson, A. G. Pipe and T. J. Prescott, "Contact type dependency of texture classification in a whiskered mobile robot," *Auton Robot*, vol. 26, pp. 223239, 2009.

[11] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, P. K onig "Texture signals in whisker vibrations," *Neurophysiol.* vol. 5 (3), pp. 1792-9, Mar 2006.

[12] J. I. Gold and M. N. Shadlen, "Representation of a perceptual decision in developing oculomotor commands," *Nature*, vol. 404, pp. 390-394, 2000.

[13] J. B Tenenbaum and T. L Griffiths, "Generalization, similarity, and Bayesian inference," *Behavioral and Brain Sciences*, vol. 4 (24), pp. 629-640, 2002.

[14] K. Kording, "Decision theory: what should the nervous system do?," *Science*, vol. 318, pp. 606-610, 2007.

[15] R. Bogacz, "Optimal decision-making theories: linking neurobiology with behaviour," *Trends in Cognitive Sciences*, vol. 11 (3), pp. 118-125, 2007.

[16] D. J. Hand and K. Yu, "Idiot's Bayes - not so stupid after all?," *International Statistical Review*, vol. 69 (3), pp. 385399, 2001.

[17] I. Rish, "An empirical study of the naive Bayes classifier," *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, U.S., Aug. 2001.

Motiwala et al. BMC Neuroscience 2011, 12(Suppl 1):P257
http://www.biomedcentral.com/1471-2202/12/S1/P257

BMC Neuroscience

POSTER PRESENTATION Open Access

Each to their own beat: periodicity in temporal inference

Asma Motiwala*, Charles Fox, Tony Prescott

From Twentieth Annual Computational Neuroscience Meeting: CNS*2011
Stockholm, Sweden. 23-28 July 2011

Characterizing cortical computation has been a key challenge in neuroscience. It has been suggested that the cerebral cortex is involved in unsupervised learning [1] and the notion that cortical computation is probabilistic is being increasingly accepted [2]. It has also been shown that cortical regions are hierarchically connected [3] and this has important implications on the way information could be represented. Moreover, the classical notion that processing in the cortex is predominantly governed by its feedforward inputs has been largely replaced by the view that feedback signals and other contextual information play a crucial role in the way information is encoded and learnt [2,4,5]. There is increasing emphasis on the influence of ongoing activity [5], ie. the state of the network on the effects of feedforward inputs.

The hierarchical temporal memory (HTM) model of cortical computation [6] was used as the starting point for the current work. HTM is a hierarchical probabilistic model that uses spatial and temporal characteristics of its inputs to represent 'invariant causes' in the world. The key assumptions in the model are that learning in the cortex involves learning sequences of coincident patterns in its inputs and that the temporal patterns in inputs are critical for the brain's ability to learn and infer with ambiguous sensory information.

For the current work we are investigating tactile perception and implementing the HTM model using sensory inputs from artificial whiskers. The HTM model emphasizes that temporal patterns in inputs are extremely important features for learning and inference in the brain. This is especially the case with tactile perception since very little information is available at any single instant; how sensory inputs evolve over time conveys indispensable information about stimuli in the world. In modeling unsupervised learning, 'time' is commonly treated as indexing sequential inputs [7]. However, we are suggesting that learning, inference and prediction with temporal patterns pose a theoretical need to model some mechanism to encode time not only in terms of a temporal succession of events and causes, but also as a metric of periodicity. This theoretical need is perfectly aligned with well established empirical evidence that the brain engages in a plethora of oscillations at several different scales and frequencies. Our implementation of HTM incorporates a mechanism which attempts to mimic hypothesized effects of local neural oscillations in temporal inference.

We are suggesting that cortical networks can dynamically maintain local oscillations to best match the relative frequencies with which inputs tend to occur. Assuming that all inputs to any cortical network are superimposed on its ongoing activity, these local oscillations can discretize the continuous stream of sensory inputs in signaling to subsequent levels in the hierarchy, as well as modulate the signal to suppress or enhance effects of specific components in the inputs based on their timing. We are demonstrating these effects in our implementation of HTM.

Published: 18 July 2011

* Correspondence: a.motiwala@sheffield.ac.uk
Department of Psychology, University of Sheffield, Sheffield, S10 2TN, UK
Full list of author information is available at the end of the article

References

1. Doya K: **What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?** *Neural networks* 1999, **12**:961-974.
2. Knill DC, Pouget A: **The Bayesian brain: the role of uncertainty in neural coding and computation.** *TRENDS in Neurosciences* 2004, **12**:712-719.
3. Felleman DJ, Van Essen DC: **Distributed hierarchical processing in the primate cerebral cortex.** *Cerebral cortex* 1991, **1**:1-47.
4. Thomson AM, Bannister AP: **Interlaminar connections in the neocortex.** *Cerebral Cortex* 2003, **13**:5-14.
5. Arieli A, Sterkin A, Grinvald A, Aertsen A: **Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses.** *Science* 1996, **273**:1868-1871.

BioMed Central

© 2011 Motiwala et al; licensee BioMed Central Ltd. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

6. George D, Hawkins J: **Towards a mathematical theory of cortical micro-circuits.** *PLoS Comput Biol* 2009, **5**:e1000532.
7. Bishop CM: **Pattern recognition and machine learning.** NewYork: Springer; 2006.

doi:10.1186/1471-2202-12-S1-P257
Cite this article as: Motiwala *et al.*: Each to their own beat: periodicity in temporal inference. *BMC Neuroscience* 2011 **12**(Suppl 1):P257.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

BioMed Central

Sensing with Artificial Tactile Sensors: An Investigation of Spatio-temporal Inference

Asma Motiwala, Charles W. Fox, Nathan F. Lepora, and Tony J. Prescott

Adaptive Behaviour Research Group, Department of Psychology
University of Sheffield, Sheffield, UK
{a.motiwala,c.fox,n.lepora,t.j.prescott}@sheffield.ac.uk

Abstract. The ease and efficiency with which biological systems deal with several real world problems, that have been persistently challenging to implement in artificial systems, is a key motivation in biomimetic robotics. In interacting with its environment, the first challenge any agent faces is to extract meaningful patterns in the inputs from its sensors. This problem of pattern recognition has been characterized as an inference problem in cortical computation. The work presented here implements the hierarchical temporal memory (HTM) model of cortical computation using inputs from an array of artificial tactile sensors to recognize simple Braille patterns. Although the current work has been implemented using a small array of robot whiskers, the architecture can be extended to larger arrays of sensors of any arbitrary modality.

Keywords: Pattern recognition, Cortical computation, Hierarchical Temporal Memory, Bayesian inference, Tactile perception.

1 Introduction

Interacting with any environment, real or artificial, involves extracting meaningful information about features in the environment, often from noisy and ambiguous sensory inputs, to guide behavior and appropriately respond to and/or control the environment. Implementing these operations in artificial systems has been a persistent challenge. However, biological organisms implement these with relatively much greater ease, in much less controlled natural environments. Given how difficult it is to design artificial systems that can perform with similar adaptability and precision with real world problems as biological systems, there is increasing motivation to address these problems using strategies that are algorithmically similar to those used by biological systems to solve similar problems in their natural environments.

Although, generating appropriate behavior and motor outputs have been argued to be the main function of brains in biological systems, they can only follow from an accurate understanding of the *state* of the agent's internal and external environment. Hence, the first step in appropriately interacting with any environment is to use sensory information available through different sensors to extract properties of how things change in the world.

R. Groß et al. (Eds.): TAROS 2011, LNAI 6856, pp. 253–264, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Several algorithms of pattern recognition and feature extraction have been suggested to extract meaningful information from sensory data in different contexts and for different types of problems. It has been previously suggested that no algorithm is inherently superior to other algorithms at solving all learning problems, and that the superiority of any algorithm depends largely on the degree to which the assumptions in the algorithm match the properties of the problem space [11]. Hence, given the efficiency with which biological organisms interact with their environment, it can be assumed that their brains have evolved to implement algorithms that best exploit the properties of real world objects and events.

2 Theoretical Background: Hierarchical Temporal Memory and Cortical Computation

The key problem an organism faces in learning about its environment is to extract consistent and stable features in the environment rather than features in the sensory data itself, *i.e.* the organism needs to learn an internal model of *hidden causes* in the world based on information available from its noisy percepts.

The hierarchical temporal memory (HTM) model is a theory of cortical computation. It is a hierarchical probabilistic model, that uses assumptions based on the properties of objects and events in the world to suggest how the cortex learns internal representations of causes from its inputs, and how it uses these for inference with ambiguous sensory information [10]. Based on the*common cortical algorithm assumption* [16], the HTM model attempts to characterize a generic algorithm with which any arbitrary region of the cortex learns and infers, irrespective of the modality or the level of processing. In other words, it aims to characterize a cortical algorithm that best exploits the statistical properties of causes in the world, using a tight set of constraints regarding the known anatomy and physiological behavior of the cortex.

The Problem of Invariance in Learning Representations: Why the Hierarchy Needs to be Spatio-temporal

The HTM model argues that representing a spatio-temporal hierarchy of causes is critical for the brain's ability to infer with ambiguous sensory information. It has been previously shown in the visual cortex that detecting spatial coincidences in its inputs allows a cortical region to represent more complex spatial patterns than those in lower levels in the cortical hierarchy [14]. However, the problem in learning consistent and stable representations of causes is that a single cause can give rise to several spatially dissimilar percepts. For example: people are exceptionally good at recognizing faces. Face recognition is invariant to the size, position and view of the face, despite the fact that different combinations of these generate widely different visual patterns.

The *temporal slowness principle* argues that causes in the world change slower than noisy percepts [13]. Based on this property of temporal slowness, it has been suggested that the cortex learns to represent invariant and complex representations not just by matching spatial similarity of its inputs but by generalizing over successive spatial patterns in time [8,3]. This *temporal proximity constraint*, that spatial patterns that consistently occur close in time are likely to be associated with the same cause in the world, is a key principle in the HTM model [10].

Spatio-temporal Representations

The cortical hierarchy has been modeled as a hierarchy of HTM nodes where every node implements the same algorithm, and at every level in the hierarchy each node pools information over a sub-set of nodes in lower levels. To maintain a spatio-temporal hierarchy of representations, each HTM node represents recurrent spatial patterns in its inputs, as well as sequences of these spatial patterns. These spatial patterns are patterns of activity in the lower level nodes, that any node pools over (*i.e.* its *children*), that are coincident in time and which recurrently occur in the nodes bottom-up inputs.

According to the temporal proximity constraint, sequences of spatial patterns that consistently occur in sensory inputs are likely to represent a single/coherent cause in the world. Thus by encoding sequences of patterns in its inputs, every node maintains representations of causes that get progressively more complex and invariant as we move up the hierarchy. This is similar to the spatio-temporal properties of cortical representations and mirrors the statistical properties of causes in the world.

Inference

Inference in HTM is probabilistic and implemented using *Bayesian belief propagation* [10]. A key feature of Bayesian probabilistic inference is that it is not only based on estimates of the system's current inputs but uses *prior* information about the statistical properties of patterns in its inputs to compute *posterior probabilities* or *beliefs* over states and causes of the inputs. Probabilistic inference is a powerful way of dealing with the inherent ambiguity in sensory information. Moreover, work on perceptual inference and sensory-motor integration is increasingly suggesting that inference in the cortex is probabilistic [15].

The HTM Model as a Biologically Inspired Technology

Not only is the HTM model a neurobiological theory, that attempts to characterize a cortical algorithm which is hypothesized to best exploit the statistical properties of causes in the world, but it has been developed as a platform to be suitable for several real world applications [18]. The only assumptions in an HTM model are those regarding the nature of the spatio-temporal structure of causes, which are assumed to be invariant features of patterns in the real world, and hence can be applied to a wide range of problems.

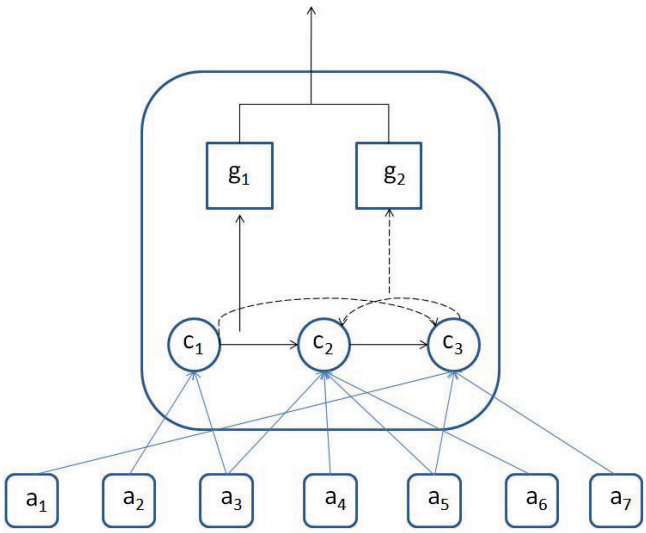


Fig. 1. The figure shows the internal structure of an arbitrary HTM node. The spatial patterns in the node are shown as circles and the sequences as square boxes. Each spatial pattern in the node represents coincident activity in a subset of the node's children and each of the sequences in the node represent sequences of these coincident patterns.

HTM has been previously tested on visual object recognition [10], the model has been extended and tested on sign language recognition [20], as well as for automated design evaluation [12]. HTM has also been tested against standard neural network architectures and has been shown to "produce recognition rates that exceed a standard neural network approach. Such results create a strong case for the further investigation and development of Hawkins' neocortically-inspired approach to building intelligent systems" [21]. In addition, HTM is already being commercially used by Vitamin D, EDSA and Forbes.com, and several other applications are being developed [2].

3 Methods

The current work aimed to investigate how inference using the HTM architecture could be implemented to recognize simple braille patterns using sensory inputs from an array of artificial whiskers.

Probabilistic inference and hierarchical representations are not new concepts, in either biology or pattern recognition. Cortical regions have been shown to be connected in a hierarchical manner [7] and several neurobiological models as well as statistical models of pattern recognition use hierarchical organization [17]. In addition, the idea that inference in the cortex is probabilistic has been popularized as the *Bayesian brain* hypothesis and belief propagation has generated

great interest as a biologically plausible inference strategy [5]. However, HTM is unique in that it uses spatio-temporal hierarchical representations for probabilistic inference, in contrast to most neurobiological and statistical models which either have spatial or temporal structure [17].

This spatio-temporal architecture was particularly well suited for the investigation of tactile perception since neither the spatial nor the temporal components in the sensory data can be ignored. The amount of information available from any single tactile sensor at any given instant is extremely sparse. Hence it is critical to pool over multiple sensors as well as consider how the inputs evolve over time to learn and infer about the properties of different stimuli.

Data Acquisition

The sensory input to the model was obtained from an array of three artificial whisker sensors that have been developed on the BIOTACT project [1]. The stimuli used were braille-like patterns (shown in Figure 2) and the size of the braille patterns was chosen to match the spatial resolution of the sensor array. The stimulus was swept over the whisker sensors using the XY positioning robot in a plane perpendicular to the shaft of the whiskers. The arrangement of the whiskers and the stimulus being swept over it are shown in Figure 2.

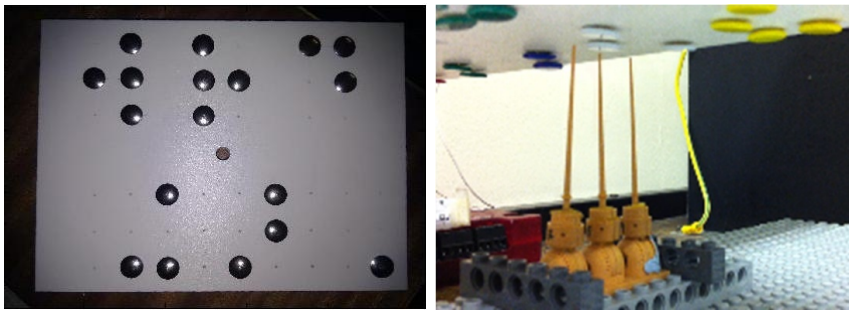


Fig. 2. The left panel shows an example of a braille pattern used as a stimulus. The right panel shows how the robot whiskers were set up and how the braille stimulus was swept over them for the current implementation.

The sensor, which has been previously described in [6], detects deflection in the whisker from its resting position on two orthogonal axes x and y and converts it into two voltage signals which are proportional to the whisker displacement on the two axes. For the current implementation, the displacement amplitude in the whiskers, when the stimulus was moved over them, was used. The signal from the sensors was segmented and an *activation* $a_m(t)$, for each sensor m at timestep t, was defined as a function of the displacement amplitude $d_m(t)$ over the tth segment,

$$a_m(t) = \frac{1}{1 + e^{-d_m(t)}} \tag{1}$$

where $d_m(t)$ is the deflection in the whisker sensor m at time t and $a_m(t)$ is the corresponding activation of the sensor. The whisker deflection signal was segmented and converted into an activation for each sensor to simplify the implementation of the model. The duration of the time window for each segment was defined to match the relative frequency with which non-zero patterns tend to occur in the inputs. Defining the activation function is not critical to the working of HTM, but since braille patterns are essentially binary, this transformation was used to make the *activation patterns* more representative of the characteristics of the stimuli (as shown in the lower panel in Figure 3).

The arrangement of the whisker sensors used in the current implementation is similar to the way in which the microvibrissae of the rat are used to explore patterns for shape recognition (see Figure 4). It has been shown in behavioral studies that the microvibrissae are critically involved in object/pattern recognition, as

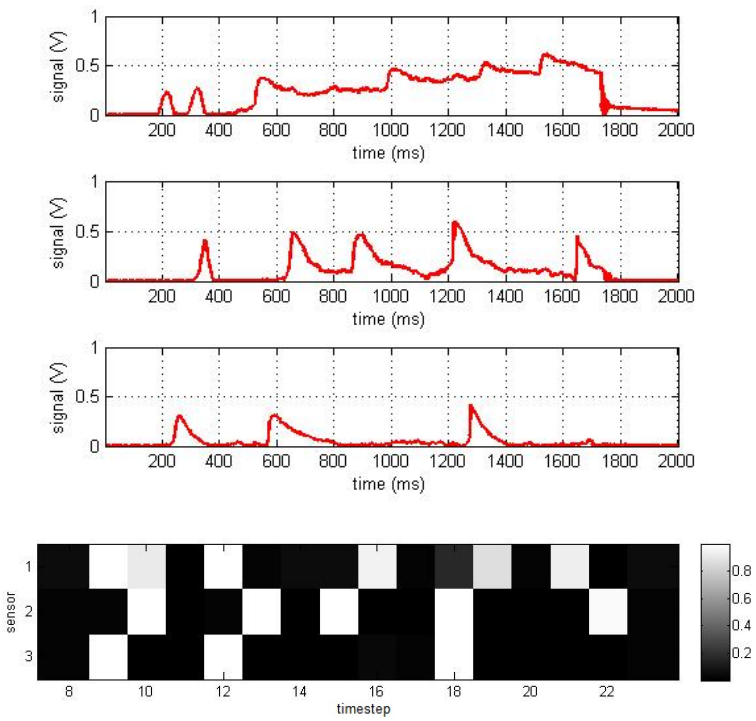


Fig. 3. The upper panels in the figure shows an example of a set of unprocessed signals from the whisker sensors. The top panel is an example of noisy sensory data, corrupt with a significantly large low frequency component and the second panel is an example of data which clearly reflects the features in the stimulus. These whisker deflection timeseries were filtered to remove low frequency components, segmented and the corresponding activation patterns for each sensor at timesteps corresponding to the segments of the timeseries are shown in the lower panel.

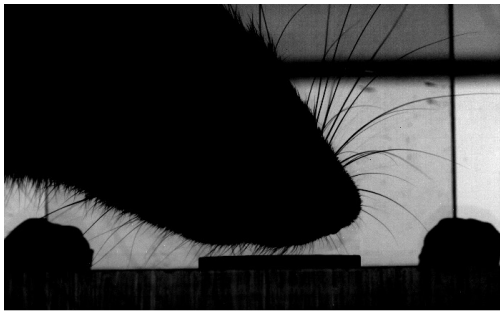


Fig. 4. The figure shows an image of a rat using its microvibrissae to explore a coin

opposed to the role of the macro-vibrissae in object localization [4]. And hence, the current implementation of HTM is also being suggested as a model of processing in the primary sensory cortex associated with the microvibrissae.

HTM Implementation

Inference in HTM nodes is with probabilistic message passing based on the Bayesian Belief Propagation algorithm [19] and the complete mathematical description of inference in HTM has been previously presented [10].

A key feature in Bayesian probabilistic inference is that it is based not only on current evidence, but also past evidence to resolve ambiguity. To this end, current evidence, of the form of a *likelihood* distribution, is combined, using the

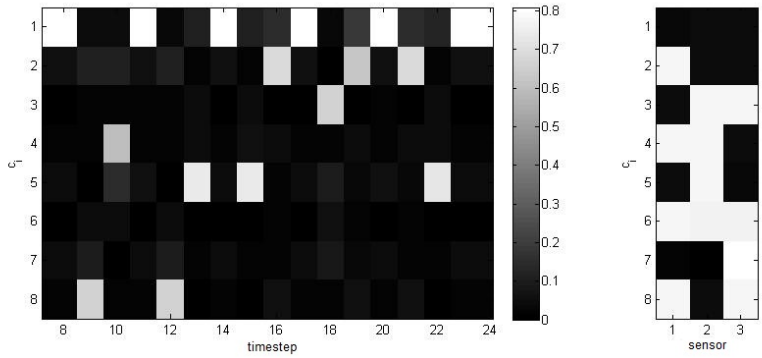


Fig. 5. The left panel in the figure shows the likelihood $y(i)$ which the node receives over its coincidence patterns. Each column in the image represents a probability distribution over spatial patterns c_i for each time step. The right panel shows the activation expected in each sensor corresponding to each of the spatial patterns c_i represented in the node.

Bayes Theorem, with a *prior* distribution, to obtain a *belief* distribution over states or events.

The feedforward input to the HTM node at each time step is the likelihood of the spatial pattern of activation in the whisker sensors that the node pools over. The activation patterns can be represented in m-dimensional space, where m is the number of sensors/children the spatial pattern is detected over. The likelihood y can then be defined by soft classification of data points in this space into clusters associated with the node's spatial patterns using any measure of degree belonging of data points to each of the clusters,

$$y_t(i) = P(A(t)|c_i) \tag{2}$$

where c_i represents coincident pattern i and $A(t) = a_1(t), a_2(t)...a_m(t)$. The coincident patterns represented in the node correspond to sub-characters in the braille pattern.

The likelihood y is combined with a prior distribution $\pi(c)$ to obtain a belief $Bel(c)$ over the node's spatial patterns.

$$Bel_t(c_i) = \frac{1}{Z} y_t(i)\pi_t(c_i) \tag{3}$$

where Z is the normalization constant.

Sequences represented in the node are assumed to be Markov chains, *i.e.* they are sequences of states in which the state of the system at any given time t is dependent on the previous states of the system, hence, the prior $\pi(c)$ and the likelihood $\lambda(g)$ are computed from the prior $\pi(g)$ and the likelihood y, respectively, using transition probabilities between the node's spatial patterns.

$$\pi_t(c_i) = \sum_{c_j} \sum_{g_r} P(c_i(t)|c_j(t-1), g_r)\pi(g_r), \tag{4}$$

and

$$\lambda_t(g_r) = P(A(0 \to t)|g_r) = \sum_{c_i} y(i) \sum_{c_j} P(c_i(t)|c_j(t-1), g_r) \tag{5}$$

where $P(c_i(t)|c_j(t-1), g_r)$ is the transition probability represented within the node. All the internal representations of the node were predefined in the current implementation.

This likelihood $\lambda_t(g)$ is the feedforward message that the node sends to the node above it in a hierarchy (*i.e* its *parents*) and the prior, which is sent to the node's children, is a function of the belief $Bel_t(c)$.

4 Results

For the current implementation the working of a single HTM node was instantiated. The coincidence patterns in the node represent single columns of Braille alphabets (see Figure 2), and the sequences represent entire Braille alphabets.

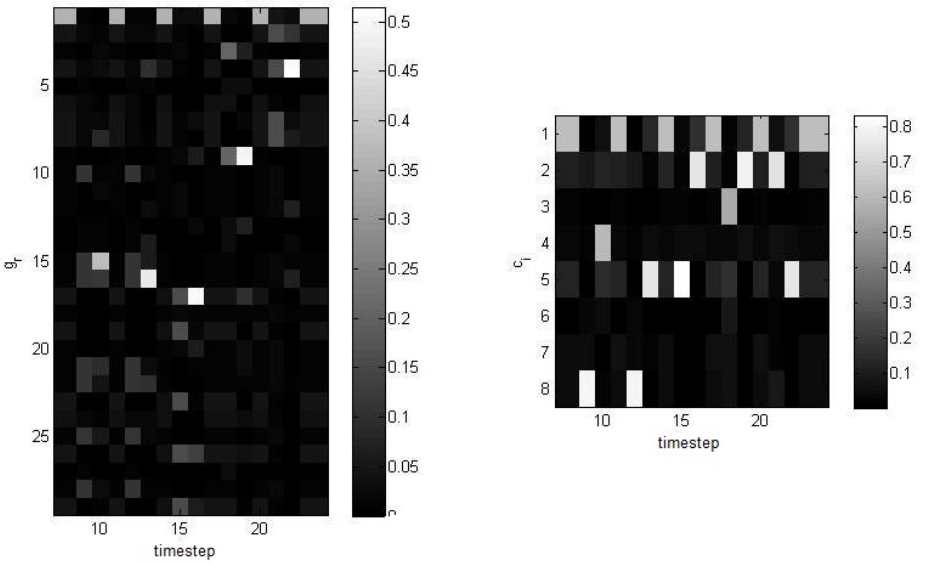


Fig. 6. The right panel in figure shows the node's belief over its coincidence patterns $Bel(c)$ and the left panel shows the feedforward likelihood over its sequences $\lambda(g)$. Each column in both images represent a probability distribution at that time step.

The results from the current implementation show how the probability distributions over the node's spatial patterns and sequences, *i.e.* the distributions over the nodes internal representations, change over time in response to the inputs from the whisker sensors (as shown in Figure 6).

Since the spatial patterns and sequences were predefined in the node, the accuracy with which the node inferred could be easily verified by matching the patterns in the node, over which the probability distributions peaked, with the patterns in the stimulus. For example, the first spatial pattern as well as the first sequence represented in the node correspond to blank patterns in the stimulus. The time steps at which the distributions peak on the first pattern indicate that the the node has inferred blank patterns in the stimulus at those time steps (see Figure 5). It is evident from number of time steps between blank patterns that sequences of non-zero patterns in the data are only two patterns long. This is a true of all braille characters which are formed by precisely arranged raised dots which occur only in six positions (two columns of three rows each) for each character (as shown in Figure 2). In other words, the patterns associated with the peaks in the belief distribution over the node's spatial patterns and the likelihood over its sequences match those in the stimulus, *i.e.* the patterns over which the node infers the highest degree of certainty match the actual patterns in the stimuli.

The results also show that the likelihood distribution over the sequences has a relatively high level of ambiguity, *i.e.* it is relatively broad, on the first time step when any non-zero sequence begins in the data. However, the distribution over the sequences sharpens over the next time step. Even in the general case, with sequences of any arbitrary length, it would be expected that the distribution over the node's sequences would become progressively sharper as more evidence is accumulated. In other words, since it is assumed that causes can share common sub-components, the degree of certainty in the state of a cause increases as more evidence becomes available.

5 Discussion

The sensor array, stimulus as well as the model architecture were all kept minimal in the current implementation to investigate thoroughly the inner workings of the HTM model. Although the data set of stimulus patterns used here was very simple, the spatio-temporal structure of braille patterns as a tactile stimuli is sufficiently rich to use them to test pattern recognition in HTM. The results show that the discussed architecture can successfully use spatio-temporal patterns in inputs to infer complex causes. This architecture can further be extended to receive sensory inputs from a larger array of sensors and by implementing a more elaborate hierarchy of nodes, to represent and infer with more complex spatio-temporal patterns. It is anticipated that insights from the current work are likely to be instrumental, within the general framework of developing robotic multiwhisker touch, for recognizing complex behaviorally relevant objects and patterns.

Key Challenges in Using HTM

For the current implementation, since the structure of the stimulus used was extremely simple, it was relatively straightforward to predefine the internal representations in the HTM node with which inference was implemented. However, a key challenge with more extended HTM hierarchies is that the nature of the node's internal representations become less intuitive as we go further up in the hierarchy. Hence, it becomes crucial that adequate learning mechanisms are defined for the model to extract these representations directly from its inputs.

The temporal proximity constraint is a powerful idea in learning invariant representations of stable causes in the world based on rapidly changing, ambiguous sensory inputs. Pooling over local spatial coincidences in a hierarchical arrangement allows representing progressively more complex spatial patterns from simpler components/features. By analogy, patterns of transitions between spatial patterns can be used to learn hierarchical representations of larger temporal patterns in inputs to represent stable/invariant features of causes in the world. However, learning complex sequences from transition probabilities between spatial patterns is not the same as learning complex spatial patterns from local coincidences. Information about inputs at previous time steps is not available to

the system from successions in the same sort of way as is the information about larger spatial patterns from local spatial coincidences. Hence, learning associations between complex sequences of patterns and higher level causes is still a key challenge.

Finally, it has been pointed out that, "if it were not for the fact that we can see, we might reasonably think it is impossible" [9]. This is true not only for vision but for several problems that nature seems to have found elegant solutions for and this becomes strikingly evident in trying to replicate these functions in artificial systems. Biomimetic robotics is a powerful tool to reverse engineer solutions available in biological systems, that have evolved to efficiently handle several hard problems, and to optimize these strategies for specific problems of interest.

Acknowledgments. The authors would like to thank Alex Cope for comments on the manuscript and for help on several occasions in the implementation of the presented work. They would also like to thank Stuart Wilson and Mat Evans of ATLAS (the Active Touch Laboratory at Sheffield) for their help with data collection, as well as, Javier Caballero and Darren Lincoln, at the university of Sheffield, for several helpful discussions. This work was supported by EU Framework project BIOTACT (ICT-215910).

References

1. Biotact consortium, `http://www.biotact.org`
2. Numenta customers, `http://www.numenta.com/about-numenta/customers.php`
3. Berkes, P., Wiskott, L.: Slow feature analysis yields a rich repertoire of complex cell properties. Journal of Vision 5(6) (2005)
4. Brecht, M., Preilowski, B., Merzenich, M.: Functional architecture of the mystacial vibrissae. Behavioural Brain Research 84(1-2), 81–97 (1997)
5. Doya, K.: Bayesian brain: Probabilistic approaches to neural coding. The MIT Press, Cambridge (2007)
6. Evans, M., Fox, C.W., Pearson, M.J., Lepora, N.F., Prescott, T.J.: Whisker-object contact speedaects radial distance estimation. In: IEEE International Conference on Robotics and Biomimetics (2010)
7. Felleman, D.J., Van Essen, D.C.: Distributed hierarchical processing in the primate cerebral cortex. Cerebral Cortex 1(1), 1 (1991)
8. Földiák, P.: Learning invariance from transformation sequences. Neural Computation 3(2), 194–200 (1991)
9. Frisby, J., Stone, J.: Seeing: the computational approach to biological vision. The MIT Press, Cambridge (2009)
10. George, D., Hawkins, J.: Towards a mathematical theory of cortical micro-circuits. PLoS Comput. Biol. 5(10), e1000532 (2009)
11. George, D.: How to make computers that work like the brain. In: Proceedings of the 46th Annual Design Automation Conference, DAC 2009, pp. 420–423. ACM, New York (2009)

12. Hartung, J., McCormack, J., Jacobus, F.: Support for the use of hierarchical temporal memory systems in automated design evaluation: A first experiment. In: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, CA, USA (2009)
13. Hinton, G.: Connectionist learning procedures. Artificial Intelligence 40(1-3), 185–234 (1989)
14. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology 160(1), 106 (1962)
15. Knill, D., Pouget, A.: The Bayesian brain: the role of uncertainty in neural coding and computation. Trends in Neurosciences 27(12), 712–719 (2004)
16. Mountcastle, V.: An organizing principle for cerebral function: The unit model and the distributed system. In: Edelman, G., Mountcastle, V. (eds.) The Mindful Brain. MIT Press, Cambridge (1978)
17. Numenta: Hierarchical temporal memory: Comparison with existing models. Tech. rep., Numenta (2007)
18. Numenta: Problems that fit htm. Tech. rep., Numenta (2007)
19. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
20. Rozado, D., Rodriguez, F., Varona, P.: Optimizing hierarchical temporal memory for multivariable time series. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6353, pp. 506–518. Springer, Heidelberg (2010)
21. Thornton, J., Gustafsson, T., Blumenstein, M., Hine, T.: Robust character recognition using a hierarchical Bayesian network. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1259–1264. Springer, Heidelberg (2006)

Scaling up a Boltzmann machine model of Hippocampus with visual features for mobile robots

Alan Saul, Tony Prescott, Charles Fox
Sheffield Centre for Robotics
University of Sheffield S10 2TN, UK
aca08ads@shef.ac.uk

***Abstract*— Previous papers [4], [5] have described a detailed mapping between biological hippocampal navigation and a temporal restricted Boltzmann machine [20] with unitary coherent particle filtering. These models have focused on the biological structures and used simplified microworlds in implemented examples. As a first step in scaling the model up towards practical bio-inspired robotic navigation, we present new results with the model applied to real world visual data, though still limited by a discretized configuration space. To extract useful features from visual input we apply the SURF transform followed by a new *lamellae*-based winner-take-all Dentate Gyrus. This new visual processing stream allows the navigation system to function without the need for a simplifying data assumption of the previous models, and brings the hippocampal model closer to being a practical robotic navigation system.**

I. Introduction

The hippocampus is thought to play a key role in associating an observed situation with similar memories [11], [8], [16], [9]. An alternative school of thought is that the hippocampus is a centre for navigation and map building [17], [1], [3]. Both approaches associate its Dentate Gyrus (DG) with the task of pattern separation [2], that is, creating non-overlapping representations of state which can be used to further discriminate between memories. All of these ideas could provide useful inspiration for localisation and mapping in mobile robotics, as explored in the RatSLAM system [12].

There are many models of the hippocampus which are gradually converging toward each other, but this paper will build on the 'unitary coherent particle filter hippocampus' (UCPF-HC) mapping of [4], [5] which begins with an explicitly Bayesian algorithm and works top-down towards the biology. The model has previously been presented primarily as a biological theory, but here we make initial steps towards using it as a practical robotic localisation and mapping method, by extending it to use more realistic sensory inputs. This requires machine vision features and some changes to the biological mapping.

This paper first gives a brief review of the hippocampus, the UCPF-HC model, and of the visual SURF features. It then presents a visual sensory extension to the model including a modified Dentate Gyrus and CA1 to handle the new sensors. We then show that the visual sensors allow for more realistic navigation – though still in an artificially discretized world – as a step towards bio-inspired mobile robotics applications.

A. Anatomy

The principal input structures of the hippocampus are the superficial layers of Entorhinal Cortex (ECs). ECs projects to Dentate Gyrus (DG) which is believed to sparsify the encoding of ECs. Both ECs and DG project to area CA3, which also receives strong recurrent connections that are disabled [7] by septal acetylcholine (ACh). CA3 and ECs project to area CA1, which in turn projects to the deep layers of Entorhinal cortex (ECd), closing a loop if ECd sends information back to ECs. ECs, CA1 and ECd outputs appear to share a coding scheme, as evidenced by one-to-one topographic projections. In contrast, DG and CA3 outputs are thought to work in other bases or latent spaces. In a second loop, ECs and CA1 both project to Subiculum (Sub), which projects to the midbrain Septum (Sep) via fornix. Septal ACh and GABA fibres project back to all parts of hippocampus.

B. UCPF-HC model

The UCPF-HC model [4], [5] mapped this hippocampal circuit onto a modified Temporal Restricted Boltzmann machine (TRBM, [20]), a machine learning algorithm. The TRBM is a Bayesian filter with Boolean observation vectors (including an always-on bias node), z'; Boolean hidden state vectors (also including an always-on bias node), x'; weight matrices $W_{x'z'}$ and $W_{x'x'}$. It specifies joint distributions,

$$P(x_t, x_{t-1}, z_t) = \frac{1}{Z} \exp \sum_t (-x'_t W_{x'x'} x'_{t-1} - x'_t W_{x'z'} z'_t). \tag{1}$$

Unlike the standard TRBM, the unitary coherent particle filter hippocampus mapping uses the following deterministic update to obtain maximum *a posteriori* estimates:

$$\begin{aligned} \hat{x}_t &\leftarrow \arg\max P(x_t | \hat{x}_{t-1}, z_t) \\ &= \{\hat{x}_t(i) = (P(x_t(i) | \hat{x}_{t-1}, z_t) > \frac{1}{2})\}_i \end{aligned} \tag{2}$$

which is the zero-temperature limit of an annealed sequential Gibbs sampler. A version of the wake-sleep algorithm is mapped onto particular phases of the hippocampal theta cycle in [4], conjecturing use of after-depolarization effects to reset the wake-sleep stages.

Noisy inputs $z_t = y_t + \epsilon_t$ are mapped to the combined ECs and DG, where the DG activations are functions of the ECs activations, $z_t = (ECs_t, DG_t(ECs_t))$. CA3 is mapped to

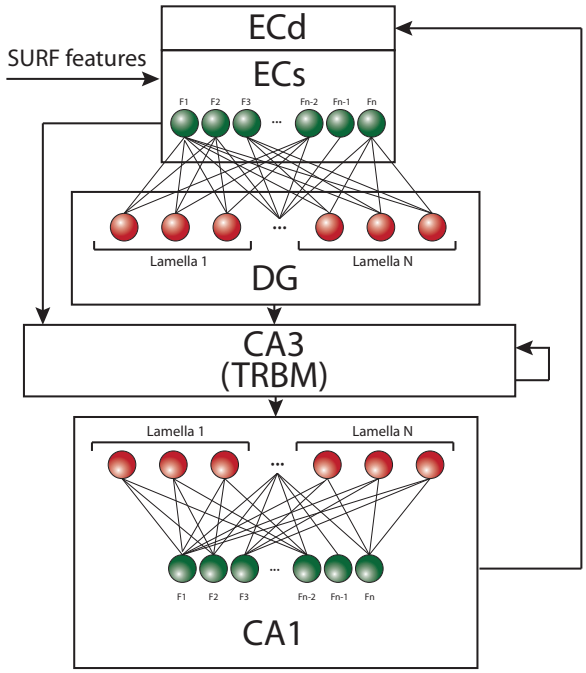


Fig. 1: Illustration of the hippocampus model showing data flows and hippocampus regions. SURF features are the new visual inputs. (Subiculum circuit not shown.)

the hidden state, x_t. CA1 performs a partial decoding into the DG basis. Finally the estimated de-noised output is mapped to ECd, $\hat{y}_t = ECd_t$. Each neural population is a Boolean vector at each discrete time step t.

A major problem with UCPF-HC is tracking loss, as it approximates whole posteriors with single samples. To recover from tracking loss, filter performance is monitored to heuristically detect its occurrence – by thresholding a moving average of discrepancy between observed and denoised sensors – then the priors are disabled when lostness is detected. In UCPF-HC, the Subiculum-Septum circuit performs this monitoring. Subiculum then compares the partially decoded CA1 information against the original ECs input, receiving one-to-one connections from both regions. If they differ for an extended period of time, this indicates loss of tracking, and the recurrent CA3 connections are disabled by Septal ACh.

The UCPF-HC model was shown in [5] to learn receptive fields of CA3 cells corresponding to a mixture of places and world features, as found in biology. This is in contrast to pure place-cell models which can learn to perform mapping and localisation (SLAM, e.g. [12]) but do not exhibit the world feature detectors found in the biological primate CA3 [11], [8], [16], [9].

C. Visual SURF features

The UCPF-HC model was originally intended as an explanatory theory of the biological hippocampus, though a simple proof of concept computational implementation was provided, running in a highly simplified microworld. This included highly abstract touch and vision senses, but the present paper will use a more advanced and realistic visual feature input, SURF.

Speeded-Up Robust Features (SURF) [6] are a state-of-the-art transform from images to a vector feature space, designed to be informative for recognising objects in machine vision. They have also been found useful in other navigation algorithms, and are an evolution of the older Scale Invariant Feature Transform [10]. The SURF transform begins by detecting interest points in an image, found as the local maxima of the scale pyramid of the image convolved with Haar wavelets. The dominant wavelet orientation is found at each interest point. Using the found interest point and orientation, a small localised grid is constructed, and sums of absolute and signed responses of vertical and horizontal Haar wavelets in its cells taken as a 64-element feature vector.

II. Methods

A. Environment and sensors

The microdomain used in the previous UCPF-HC models was a simulated plus-maze environment, containing 13 discrete locations consisting of four arms each containing 3 discrete locations, and a centre point. The robots state within this environment is encoded by one of 13 places ($place \in [0:13]$) and one of 4 head-directions corresponding to discrete compass headings ($hd \in \{N, E, S, W\}$). Touch sensors detect the presence of surrounding walls, and cells responding to coloured posters at the ends of arms simulated a crude form of colour detection.

Previously this simulated environment was completely theoretical and senses were crafted to allow for maximal separation between locations descriptions. To bring the model closer to a functional robotic application, a representation of vision has been introduced to the original model's array of senses. Each location direction pair, (p, hd), has a selection of associated views. Images correspond to real photographs taken from within the courtyard of Regent Court, University of Sheffield (Figure 2), whose paved paths form a real-world plus maze shape. Photographs from the real location are directly mapped to the simulated environment. Alternative views of the same location are used to simulate changes in the environment and incorrect odometry input upon revisiting a location. Alternative images were taken at different times of day, at slightly differing angles ($\pm 10°$) under different lighting and weather conditions. To overcome the effective noise introduced by using images under different lighting conditions, robust SURF features are used to encode the significant features of each image and detect similarities between new and previously seen views.

SURF features are extracted for every image, and are compared with a base set of SURF features. If the SURF feature is found to be a close enough match to any feature in the base set, the base set feature is determined to be present for that image. This allows a Boolean visual observation vector to be formed uniquely describing an image as a combination of present features. This visual observation vector is fed into the hippocampal model via the EC input

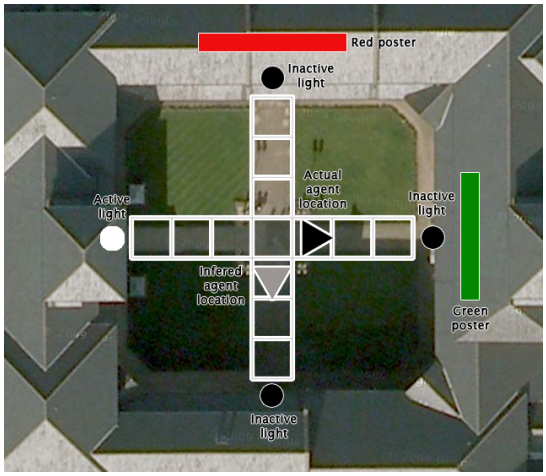


Fig. 2: An overhead view of Regent Court (Google Maps), showing the mapping of the artificial plus maze used in simulation to real life location.

and is processed alongside the existing odometry and sensory information throughout learning and inference.

B. Pre-processing visual features

The base set of features required for matching must be a range of features which are present in a range of images. In order to extract SURF features that are present in a range of images, the descriptions of individual SURF features need to be merged. SURF features produce a description of a feature in the form of 64 floating point numbers. The approach used to extract common SURF features that are as closely matched as possible by a range of images, was to use a combination of the *k*-Nearest-Neighbour algorithm and merging. The merged SURF features produced are similar to that of merging groups using *k*-Means-Clustering and taking the average SURF feature description for each group. The distance between a pair of SURF features $\mathbf{p}^i$ can be determined as the Euclidean distance between the two features,

$$d(\mathbf{p}^i) = \sum_{j}^{64} (\mathbf{p}_j^{i[1]} - \mathbf{p}_j^{i[2]})^2. \tag{3}$$

Algorithm 1 describes in depth the how this generalisation was made, accompanied by Figure 3. The Fast Library for Approximated Nearest Neighbour search [13] was used for find the nearest neighbour with k = 2.

Determining whether a feature is present within an image requires taking the Euclidean distance of Haar wavelet responses between a member of the merged SURF feature set, and a SURF feature extracted from the image being analysed (Equation 3). A feature is regarded present if this distance is small enough, presence is thus a function of distance between Haar wavelet responses,

$$c(\mathbf{fe_j}, \mathbf{fe_k}) = \begin{cases} 1 & \text{if } d(\mathbf{p}^i) < t' \text{ s.t } \binom{p^{i[1]}=\mathbf{fe_j}}{p^{i[2]}=\mathbf{fe_k}} \\ 0 & \text{otherwise} \end{cases} . \tag{4}$$

Algorithm 1 SURF feature generalisation

Extract most distinctive features from every image
Plot these features in 64 dimensional space using their descriptors
Use *k*NN to find nearest neighbour to every individual feature
while feature pair $\mathbf{p}$ exists s.t $d(\mathbf{p}) < t$ **do**
  for each pair: $\mathbf{p}^i$ in P s.t $d(\mathbf{p}^i) < t$ **do**
    Merge the two features Haar wavelet responses:
      $\mathbf{p}^{n+1} = \frac{\mathbf{p}^{i[1]}+\mathbf{p}^{i[2]}}{2}$
    Add $\mathbf{p^{n+1}}$ to feature set
    Remove $\mathbf{p}^{i[1]}$ and $\mathbf{p}^{i[2]}$ from feature set
  end for
end while

The maximum distance to be regarded as present, t', depends on the particular set of merged features and the features being analysed. It is the case, however, that the distance from the original features to the newly merged features is likely to be larger than the t used for merging (Algorithm 1), if the feature has been used in multiple merges. In order for even the same features that were used in the initial merging to be present, the similarity threshold between features required to indicate a 'match' needs to be relaxed to t' such that $t' > t$. Without merging features and relaxing the matching threshold t', images will share very few common features. Consequently a large Boolean observational vector would be required to ensure each image could be uniquely described by its combination of SURF feature matches.

C. DG encoding

The previous UCPF-HC models used a small number of simple touch and colour sensors, along with grid cells, as the ECs input vector. Because these sensors were well understood by the model authors, it was possible to handset weights $W_{EC \to DG}$ to make DG cells respond to particular combinations of inputs, such as touches and locations.

The new model retains these handset weights for the simple sensors, but as it introduces a large number (e.g. 80) of merged SURF features into the ECs input, it is no longer practical to choose and hand-set useful combinations of these additional features, and so an automated approach must be used instead.

The *lamellae hypothesis* [18], [14], [19] in neuroscience states that single cells in EC project exclusively to localised (2mm) regions of DG, known as *lamellae*. Following [14] we model the visual part (only) of our DG as a set of N lamellae, receiving input from subsets X of the merged SURF features only.[1]

The Hebbian, winner-take-all learning of Algorithm 2 is applied within each lamella, which forces it to sparsely

[1]In the full lammelae hypothesis, each lamella receives input from many types of EC cells such as touch, grid and vision. However we chose not to alter the structure of the old model's sensor processing, in order to keep the visual extensions as a removable/pluggable extension to the model.

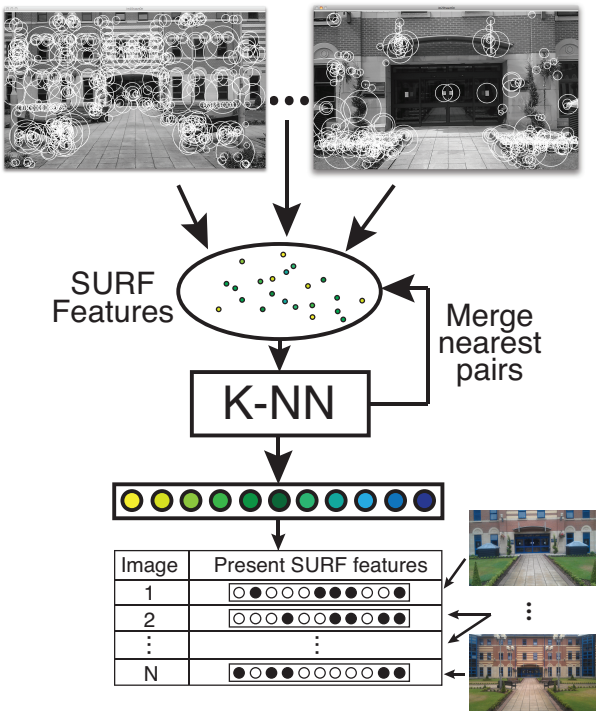


Fig. 3: Illustration of the process required to derive a set of SURF features common to multiple images, and computing SURF description of images.

encode its input. Consequently only one neuron in each lamella will be active at any time, and each lamella learns to represent a set of mutually exclusive conjunctions of its inputs. This ensures that the DG activity maintains sparseness, as seen in biology, with only N neurons being active in the entire DG subfield of $N \times X$ neurons.

D. CA3, CA1, ECd decoding

CA3 projects to CA1, which in turn projects back onto the deep layers of the EC (ECd). Several computational models [15], [4] view the CA1 as a translator, helping to map from the CA3 coding scheme back to the ECd representation. As in the previous UCPF-HC model [4], we assume that CA1 decodes the output of CA3 into the same coding scheme used by DG, then ECd decodes from this to the same scheme used by ECs.

As in the previous model, we do not attempt to learn the weights $W_{CA3 \to CA1}$ in a biologically plausible way[2]. Rather, we simply re-use the weights $W_{DG \to CA3}$ learned by the TRBM. A naïve decoding would be given by

$$CA1_{naive} = (sig(W^T_{DG \to CA3} CA3) > \frac{1}{2}), \qquad (5)$$

and ignoring the EC components of the resulting (EC,DG) vector. This decodes each CA1 cell individually using a threshold. However, a better decoding is possible because we have prior knowledge about the structure of DG's lamellae, which is reproduced in CA1. We know that due to winner-take-all encoding, exactly one cell of each lamella will be active. Hence we may apply the winner-take-all rule again in each CA1 lamella[3], which has the effect of denoising the CA1 in comparison to the naïve method.

The weights $W_{CA1 \to ECd}$ need to invert the previous sparse coding $W_{ECs \to DG}$ to return information to the original entorhinal input basis. In the new model we have been able to do this in quasi-biological way, using the classical perceptron learning rule of algorithm 3.

Algorithm 2 Competitive learning algorithm for N neural networks

for all lamella n in N **do**
  Choose X random feature indices
  Make fully connected neural network connecting EC[X] features to DG lamella n
  Randomise weights of all $W^n = W_{EC[X] \to DG[n]}$
end for
for P presentations of training data **do**
  for all image I in training set **do**
    for all lamella n in N **do**
      Calculate output firing:
        $O^n = I \cdot W^n$
      Find winning neuron in lamella n:
        $a = \max(O^n)$
      Update weights:
        $\Delta W^n_{ij} = \alpha O^n_j I_i$ if $j = a$ else 0
      Normalise weights:
        $\mathbf{W^n} = \frac{\mathbf{W^n}}{\sum_{ij} W^n_{ij}}$
    end for
  end for
end for

Algorithm 3 Perceptron learning rule

Initialise weights to zero.
for P presentations of training data **do**
  for all lamellae n in N **do**
    for all encoded patterns $ideal$ of input patterns I **do**
      Calculate actual output activity of CA1:
        $y_{Ideal} = [f(\mathbf{w}^n \cdot Ideal)]$
        where $f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$
      Update weights:
        $\Delta \mathbf{w^n} = \alpha(I - y_{Ideal}) ideal$
    end for
  end for
end for

III. Results

The present model extends the previous UCPF-HC model [4] by adding realistic visual input and new DG and CA1 processing of that visual input, after SURF preprocessing. It retains the discrete locations of the plus maze from the old model, but places them into real spaces around the Regent

[2]though see [4] for a discussion of how it could be achieved.

[3]in our code this is described as 'smart decoding'.

Court building as described in section II-A. One would expect the introduction of additional sensors and processing to improve the performance of the model.

We test the new model using the same protocol as in the previous paper [4]. 30,000 random walk steps are taken around the 13 discrete locations of the simulated Regent Court environment, and the walk was replayed though the learning algorithms until the weights converged. We used a number of input neurons feeding into each lamellae, $N = 7$ and a number of lamellae, $X = 45$, throughout all experiments. To simulate noise within the odometry and sensory input, the EC had 10% noise, ie. on average one in 10 of its Boolean senses flipped. Python code is again available as supplemental material from the authors.

The results show that the additional visual information produces dramatic improvement in the model's ability to maintain an estimate of its true location.

Figure 4 shows the amount of time that the agent is lost during the walk, (following the display format of [5], [4]) in runs with the new SURF extensions disabled and enabled. In a previous test of the learning model [5], a simplifying assumption was made[4] that the grid cells behave as 'noisy GPS' [4] units rather than accumulating odometry data – this was to make the learning problem easier while demonstrating the biological learning mapping. Figure 4 now shows the results of the original UCPF-HC model without this assumption, *No SURF No nGPS*, which are in fact very poor and comparable to *Random* weights[5]. The figure reproduces previous results [4] for performance with the learned model with the nGPS assumption made, *Learned with nGPS*.

The two bars on the right of figure 4 show the performance of the new model, with the new visual system enabled. Using SURF features (and odometry) only, and without the nGPS assumption, the model (*Learned with SURF only*) outperforms the old model, achieving 95% accuracy. When the old model's sensors are included in addition to SURF (*Learned with SURF*), the accuracy improves further to 97%.

Figure 5 shows the lostness probabilities with a larger, 20%, noise introduced to the input sensory information. This shows that extremely noisy data still cannot be handled by the present model thus there is room for improvement in these cases, for example by using further sensors as well as vision.

Figure 6 shows the real location of the robot (black line) against the estimated location by the hippocampal model within (blue line). These lines overlap throughout the whole simulation, showing that a highly accurate estimation is maintained. (See [5] for description of the complex display format.)

[4]documented in its accompanying source code.

[5]In fact they are slightly worse, as the random weights tend to produce very little change in the initial location. A stopped clock is accurate more often than a slow one!

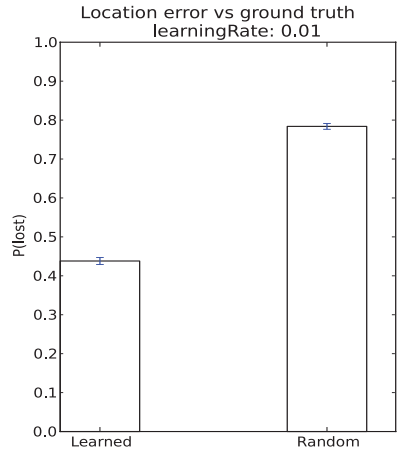


Fig. 5: Lostness probabilities working with 20% noisy data input to EC.

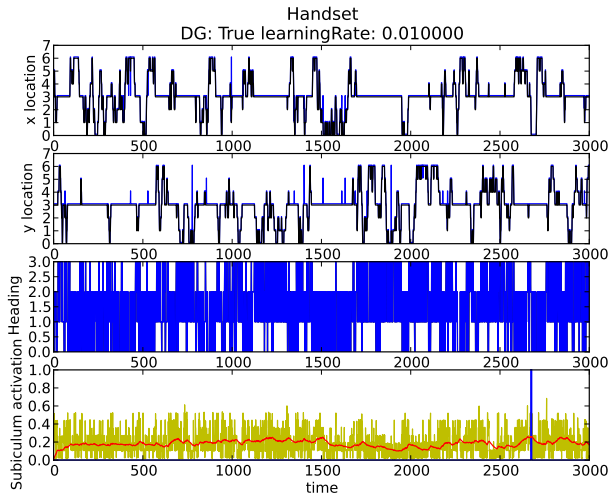


Fig. 6: Results showing almost perfect mapping between ground truths of location and belief of location illustrating extended models ability, with the septum intact.

IV. Discussion and Future Work

Previously the UCPF-HC model [4] used odometry and simplified abstract sensory information. This input included unrealistic 'noisy GPS' grid cell activity, whisker touch senses and extremely simplified colour senses, with handset receptive fields describing descriptive conjunctions of such features in the DG and CA1.

The present study has extended this model to: receive real-world visual SURF features as input; learn its own DG receptive field for these additional features using the biological hypothesis of DG lamellae; learn an improved CA3 representation using this information; and decode it back to ECd using a further lamellae based scheme in CA1.

The visual extensions allow navigation to be performed with high accuracy, even when the noisy GPS assumption is dropped, i.e. when the entorhinal grid cells are performing realistic odometric integration as in a real mobile robot, rather than acting as temporally independent location observations.

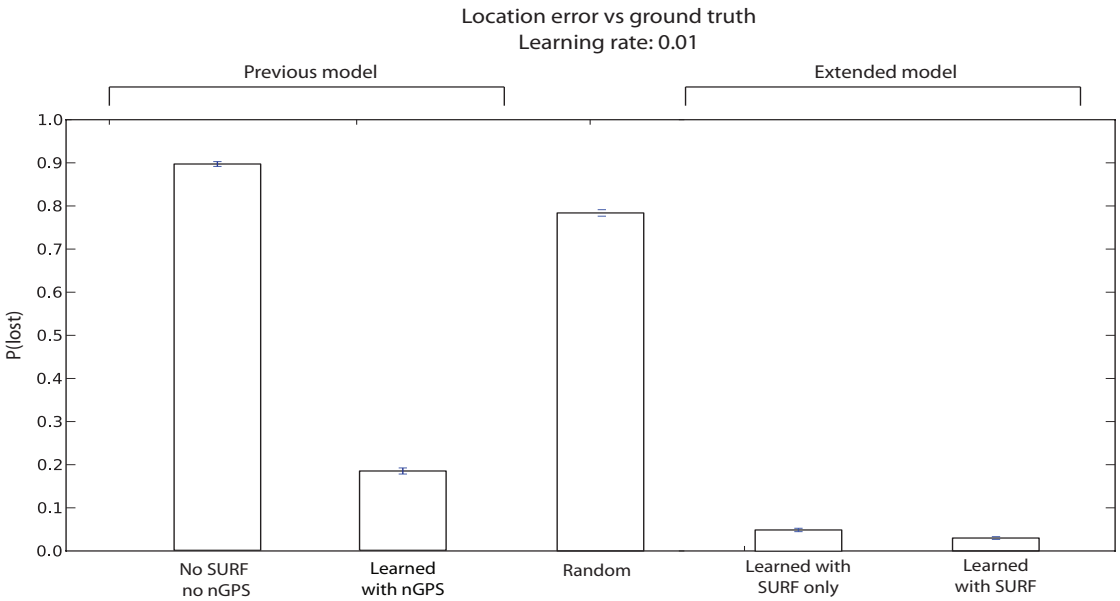


Fig. 4: Comparing lostness probabilities of model containing SURF features and model using only whisker senses, light senses and odometry information.

This level of accuracy suggests that visual SURF features coupled with the UCPF-HC model could form the basis of a future localisation and mapping (SLAM) system for real mobile robots.

However we have still retained the discrete location assumption in this work, which precludes real robot implementation at this stage. The next future work step would thus be to remove this, and allow a real robot using SURF features to learn its own place representations in a continuous world, as performed in RatSLAM [12] and similar architectures. The present work suggests that SURF vision features could be powerful enough to enable this research to take place. In contrast to RatSLAM-like architectures, UCPF-HC is derived from a top-down Bayesian machine learning model, which is able to represent more complex states of the world than place alone, and provides a probabilistic semantic interpretation of the hippocampal function.

REFERENCES

[1] A Arleo and W Gerstner. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299, 2000.

[2] A Bakker, C B Kirwan, M Miller, and C E L Stark. Pattern Separation in the Human Hippocampal CA3 and Dentate Gyrus. *Science*, 319(5870):1640–1642, March 2008.

[3] R Chavarriaga, T Strösslin, D Sheynikhovich, and W Gerstner. A computational model of parallel navigation systems in rodents. In *Neuroinformatics*, pages 223–241, Ecole Polytech Fed Lausanne, Sch Comp & Commun Sci, CH-1015 Lausanne, Switzerland, 2005. Ecole Polytech Fed Lausanne, Sch Comp & Commun Sci, CH-1015 Lausanne, Switzerland.

[4] C Fox and T Prescott. Hippocampus as unitary coherent particle filter. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, 2010.

[5] C Fox and T Prescott. Learning in a unitary coherent hippocampus. *Artificial Neural Networks–ICANN 2010*, 2010.

[6] Bay H, Ess A, Tuytelaars T, and Gool LV. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[7] ME Hasselmo, E Schnell, and E Barkai. Dynamics of Learning and Recall at Excitatory Recurrent Synapses and Cholinergic Modulation in Rat Hippocampal Region Ca3. *J. Neurosci*, 15(7):5249–5262, 1995.

[8] JJ Hopfield. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences*, 79(8):2554–2558, 1982.

[9] S Kali and P Dayan. The involvement of recurrent connections in area CA3 in establishing the properties of place fields: a model. *J. Neurosci*, 20(19):7463–7477, 2000.

[10] DG Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] D Marr. Simple Memory - Theory for Archicortex. *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences*, 262(841):23–81, 1971.

[12] MJ Milford, GF Wyeth, and D Prasser. RatSLAM: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, pages 403–408, 2004.

[13] M Muja. Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory And Applications*, pages 331–340, 2009.

[14] CE Myers and HE Scharfman. A Role for Hilar Cells in Pattern Separation in the Dentate Gyrus: A Computational Approach. *Hippocampus*, 19(4):321–337, 2009.

[15] RC O'Reilly. Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off. *Hippocampus, New York, Churchill Livingstone*, 4(6):661–682, 1994.

[16] ET Rolls, SM Stringer, and TP Trappenberg. A unified model of spatial and episodic memory. *Proceedings of the Royal Society of London Series B-Biological Sciences*, 269(1496):1087–1093, 2002.

[17] PE Sharp. Computer-Simulation of Hippocampal Place Cells. *Psychobiology*, 19(2):103–115, 1991.

[18] Lomo T. Excitability Changes Within Transverse Lamellae of Dentate Granule Cells and Their Longitudinal Spread Following Orthodromic or Antidromic Activation. *Hippocampus*, 19(7):633–648, 2009.

[19] N Tamamaki and Y Nojyo. Projection of the entorhinal layer II neurons in the rat as revealed by intracellular pressure-injection of neurobiotin. *Hippocampus*, 3:471–480, 1993.

[20] GW Taylor and GE Hinton. Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*, 19:1345–1352, 2007.

Inventory management with dynamic Bayesian network software systems

Mark Taylor[1] and Charles Fox[2]

[1] Management School
[2] Adaptive Behaviour Research Group
University of Sheffield, UK
charles.fox@sheffield.ac.uk

Abstract. Inventory management at a single or multiple levels of a supply chain is usually performed with computations such as Economic Order Quantity or Markov Decision Processes. The former makes many unrealistic assumptions and the later requires specialist Operations Research knowledge to implement. Dynamic Bayesian networks provide an alternative framework which is accessible to non-specialist managers through off-the-shelf graphical software systems. We show how such systems may be deployed to model a simple inventory problem, and learn an improved solution over EOQ. We discuss how these systems can allow managers to model additional risk factors throughout a supply chain through intuitive, incremental extensions to the Bayesian networks.

Keywords: inventory, supply chain, risk management, Bayesian networks, economic order quantity, Markov decision process

1 Introduction

Supply chain risk management (SCRM, [6]) involves modelling and optimising the flow of goods between suppliers, customers and warehouses. Broadly construed, the scope of SCRM includes factors such as transportation risk, supplier failure and customer order models, at all levels of a supply chain. Traditionally, inventory theory has been applied at the level of the single organisation, but can now be viewed as a component of larger multi-echelon supply chain models.

Economic Order Quantity (EOQ, [3]) is still a common approach to practical inventory management, determining the order quantity of stock for a particular item [12, p. 374], and is used to determine optimal ordering quantity given a large number of assumptions [8, p. 275]. It is these assumptions that give EOQ its limitations. EOQ's limitations have previously been addressed using mathematical formulations of Markov Decision Processes, known as Stochastic Inventory Theory [11, 4, 2, 15]. These approaches, while optimal, require complex modelling and dynamic programming mathematics specific to each particular case and are not available to non-specialist managers.

The present paper provides a new approach to practical inventory and supply chain risk management using off-the-shelf Bayesian network software. This approach is highly general and may relax most of the standard assumptions made by EOQ theory. It is highly extensible, as Bayesian networks allow and encourage incremental qualitative and quantitative modelling of details about the world in an intuitive and graphical manner by non-specialist managers. Our long-term goal is to model entire supply chains and detailed risk factors with these tools, however this paper presents our initial proof-of-concept results on a standard inventory problem, then discusses extensions to larger supply chain models. We show here that Bayesian networks can first replicate the standard EOQ case which is a familiar reference point for many mangers, then show how they can extend EOQ by relaxing most of its assumptions and finding dynamic policies using reinforcement learning. These policies solve the Markov Decision Process of Stochastic Inventory Control but – unlike in previous work – the learning is wrapped inside a software package leaving the manager free to specify increasingly detailed world models with intuitive visual modelling and historical or subjective probability data. Our aim is not to introduce new mathematics but to illustrate the beginnings of a software-based methodology for highly general, integrated inventory and supply chain risk management by non-specialists.

1.1 Standard EOQ

We make a distinction between the 'EOQ world' and the 'EOQ policy'. The *EOQ world* is the set of assumptions made in EOQ theory, and various inventory policies may be tested in simulations of this world. The *EOQ policy* is the particular policy found by EOQ theory. We will also introduce a *relaxed world* which relaxes most of the assumptions of the EOQ world. (As well as the EOQ policy, we will use a reinforcement learning software system to find optimal policies in the EOQ world and in the relaxed world.)

The *EOQ world* assumes [8, p. 275]: (1) Demand is certain, constant and continuous over time; (2) The quantity ordered is constant over ordering times; (3) Ordering cost is constant and independent of quantity ordered; (4) Lead time is fixed; (5) The cost of holding a unit of stock is independent from the quantity in stock; (6) The purchase price of the item is constant; (7) There are no limits on order size.

The *optimal EOQ policy* is defined algebraically by [8, p. 277]

$$EOQ = \sqrt{\frac{2ca}{h}}, \tag{1}$$

where c is the acquisition cost of each order (i.e. the administrative and overhead costs of the order, not the price of the goods themselves); a is the annual usage in units; and h is the holding cost per unit per year.

The following is an example application of EOQ which will be referenced throughout the rest of this study. Assume a demand of $a = 6000$ items per year; potential ordering points at the start of each month; a monthly holding

cost of $h = \$0.10$ per item and an ordering cost of $c = \$15$ per order. Using the EOQ formula above, the EOQ-optimal policy is to order 1341 items, with $4.47 = a/1341$ orders per year, which gives an order every $2.7 = 12/4.47$ months.

1.2 Bayesian networks

Bayesian networks provide a formalism for reasoning about partial beliefs under conditions of uncertainty. These parameters are combined and manipulated according to the rules of probability theory [10]. Let us consider n discrete random variables $x_1, x_2, ...x_n$, a directed acyclic graph with n nodes, and suppose the jth node of the graph is associated to the x_j variable. Then the graph is a Bayesian network, representing the variables $x_1, x_2, ...x_n$, if

$$P(x_1, x_2, ...x_n) = \prod_j P(x_j | parents(x_j)),$$

where $parents(x_j)$ denotes the set of all variables x_i such that there is an arc from node x_i to x_j in the graph. The probability terms in the product are described by Conditional Probability Tables (CPTs) which may be set by hand or learned from data. Standard algorithms such as junction trees [1] exist to perform inference on Bayesian networks.

Dynamic Bayesian Networks (DBNs) allow the modelling of entities in a changing environment where the values of variables change over time [1, 9]. Functionally, DBNs capture the process of variable values changing over time by representing multiple copies of network nodes with one copy for each time step [1]. Visually, they may be displayed using two copies of each recurrent node representing the current, t, and previous, $t-1$ states.

Decision Networks (DNs) [14] – also known as 'influence diagrams' [5] – are Bayesian Networks with the addition of two types of nodes: Decision Nodes and Utility Nodes. Inferences can be made about the best actions to take to maximise utility.

Utility nodes u_i – conventionally represented as diamonds in pictures of Bayesian networks – may have multiple parents and no children. Here they have finite, integer-valued state space, and are associated with Utility Tables specifying their values as a deterministic functions of the state of their parents. The utility of a whole network is given by the sum of all utility nodes, $\sum_i u_i$. When there is uncertainty in the network, the expected value of the total utility is used, $\langle \sum_i u_i \rangle$. When used in DBNs, utility nodes are discounted by the time value of utility γ to give an expected discounted present utility objective,

$$\langle U \rangle = \sum_{t=1}^{\infty} \gamma^t (\sum_{i=1}^{N} \langle u_i \rangle).$$

Action nodes – conventionally represented as squares in pictures of Bayesian networks – are here taken to be finite discrete variables, whose values are selected

dynamically by a policy, as a function of the states of their parent nodes[3]. The goal of policy optimisation is to find the policy that maximises the expected discounted present utility.

The networks in fig. 1 and 2 are dynamic Bayesian decision networks, including recurrent connections, action and utility nodes. There are many off-the-shelf software systems that allow Bayesian networks to be constructed graphically by end-users, for example BayesiaLab (www.bayesia.com), Netica (www.norsys.com) and Hugin (www.hugin.com). The examples in this paper are produced using BayesiaLab and show a novel application to inventory and supply chain risk management.

1.3 Reinforcement learning policy

To find optimal policies in the relaxed world – as opposed to EOQ policies – our software system uses internally a form of reinforcement learning [13]. It maintains a Q-value for each (action,state) pair, where the state ranges over the combined states of the parents of the action node. These values become estimates,

$$\langle U|s_t, a_t\rangle \approx Q(s_t, a_t) \approx \langle (\sum_i u_i)_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})\rangle, \tag{2}$$

when updated by

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha((\sum_i u_i)_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})). \tag{3}$$

If Q is known exactly, or a best estimate finalised from learning, then the optimal policy is to select actions,

$$a_t = \arg_{a_t} \max Q(s_t, a_t). \tag{4}$$

Learning can be performed by drawing actions from annealed distributions based on successive estimates of Q,

$$P(a_t) = \frac{1}{Z} Q(s_t, a_t)^{1/T}. \tag{5}$$

If T is reduced sufficiently slowly over many Monte Carlo Markov Chain samples from the DBN [9], then Q converges to a locally optimal policy, with the probability of reaching a global optimal increasing with the slowness. MCMC sampling and reinforcement learning are standard DBN algorithms which may be performed with off the shelf software systems and knowledge of their details is not required by the end-user.

We emphasise that in contrast to EOQ, which computes all order sizes and times in advance, DBNs can learn dynamic policies, where the ordering action at each time is a function of the state s_t of the network at that time. Such policies can for example increase the number of orders when a backlog state is large.

[3] The present paper considers only cases where the parents of action nodes are observed. For unobserved parents the actions must be function of the *distribution* of belief rather than the state and the task is known as a Partially Observable MDP.

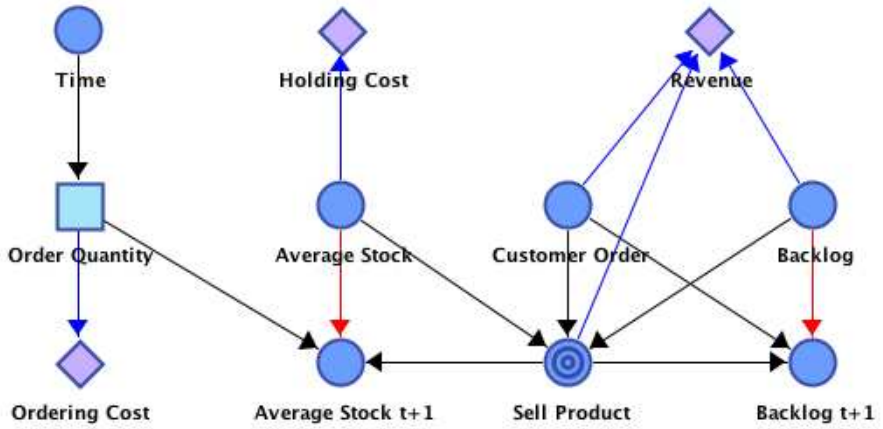


Fig. 1. Dynamic Bayesian network implementation of the EOQ world

2 Methods

We work with two Bayesian network models and two policy types. The first model is of the EOQ world, where customer demand is constant over time. The second is a relaxed world in which customer demand is a random variable conditioned on time of year. The first policy type is the EOQ policy described above. The second policy is the reinforcement learning policy, found automatically as described below. All models are created in BayesiaLab using a graphical interface to design the network, then deterministic equations or probabilities to specify the CPTs. Importantly, specialist knowledge of Bayesian inference and reinforcement learning is not required by the user, who needs only specify the world model using a graphical interface.

2.1 DBN model of the EOQ world

We first aim to demonstrate how a DBN can model the standard EOQ world. This will serve as a basis for the more complex relaxed model later. Fig. 1 shows the DBN structure. We use discrete time steps, of one month in length. Time-dependent CPTs are discretised into months, and stock quantity CPTs are discretised into integers between 1 and 20 'units' of stock, where one unit comprises 500 items.

As the demand is constant and known (6000 items per year), the *CustomerOrder* node is deterministic and constant, giving the number of customer orders, in units, at each time step to be

$$CustomerOrders = (6000/500)/12 = 1$$

which can be entered in the software as a specification of the CPT.

The number of units ordered for our own inventory each month is modelled by the *OrderQuantity*, an action node whose policy is to be learned.

There are two utility nodes modelling the ordering and holding costs, where the ordering cost is the administrative cost, \$15, of actually placing an order rather than the price of the units ordered, and it costs $-\$0.1 * 500 = -\$50$ to hold each unit,

$$OrderCost = -15$$

$$HoldingCost = -50 * AverageStock$$

The *Revenue* node models the reward for completing unit sales, being the difference between the purchase and sale price, which we set to be \$200 (this quantity is not required in standard EOQ), multiplied by the number of unit sales in the month,

$$Revenue = 200 * SellProduct$$

We model the size of the inventory and/or order backlog over time by introducing recurrent deterministic nodes,

$$AverageStock_{t+1} = AverageStock_t + OrderQuanity_{t+1}$$

$$Backlog_{t+1} = Backlog_t + CustomerOrder_{t+1}.$$

The quantity sold each month is the minimum of the stock level and the number of orders,

$$SellProduct = min(AverageStock, Backlog + CustomerOrders).$$

2.2 DBN model of a relaxed world

The relaxed world model in fig. 2 violates most of the EOQ assumptions. In particular, (1) the demand (*CustomerOrders*) are no longer constant each month but now depend stochastically on an underlying demand level which is a function of the time of year. This leads to an inherently dynamic model as this uncertainty propagates through the DBN; (2) Non-constant order quantities are now allowed, which may be functions of the backlog size; (3) the order cost utilities may now depend on order size; (4) Dynamic ordering policy allows non-fixed lead times to be modelled: if a component is not received when ordered, the backlog increases which may allow a re-order at the next step; (5) the holding cost utility may now be a function of the stock size (this is common in practice, for example filling a warehouse and needing to build a new one introduces a large non-linearity).[4] The variable costs used to illustrate (3) and (5) are listed in tables 1a and 1b.

[4] In the present network we retain (6) fixed purchase prices, although their dependency on time or random variability could easily be added to the DBN model. A new limitation introduced by the DBN formalism is that (7) we must use some upper limit on available order size, although in practice this can be made arbitrarily large, at the expense of computation time, until it becomes unimportant.

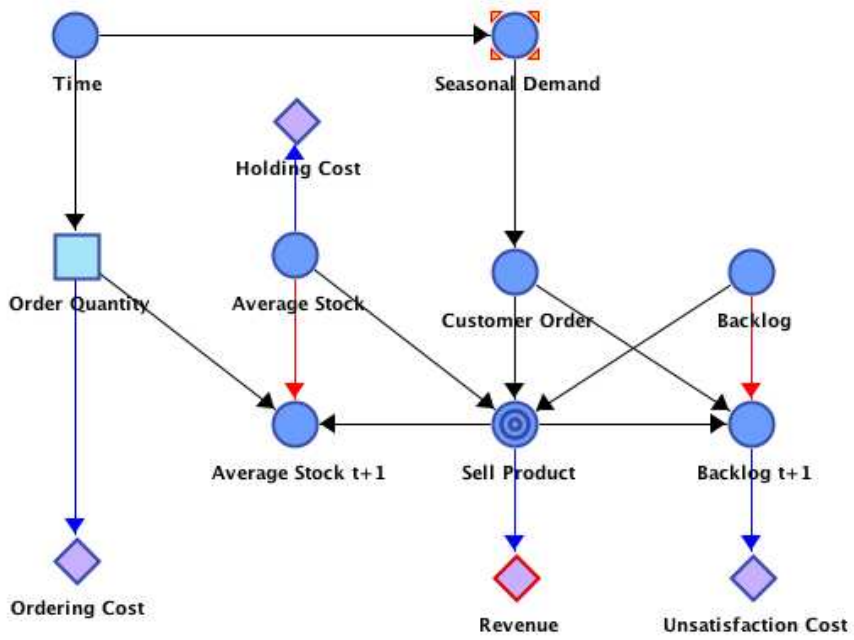


Fig. 2. Dynamic Bayesian network implementation of the relaxed world

Table 1. (a) Non-constant and dependent ordering cost; (b) Quantity dependent stock cost

Order quantity	Ordering cost
10	-15
9	-16
8	-17
7	-18
6	-19
5	-19
4	-19
3	-21
2	-22
1	-23
0	0

Average stock	Holding cost
10	-420
9	-320
8	-290
7	-260
6	-230
5	-200
4	-120
3	-90
2	-60
1	-30
0	0

3 Results

Having introduced time and backlogs into model, we first found that it was necessary to add a negative utility to model customer unsatisfaction if an order is late – this concept does not appear in the original EOQ world, and we found that it can lead to computational instabilities such as infinite backlogs if not included in the model. The *UnsatisfactionCost* node in figure 2 adds a $10 penalty for each unit-month of delay which stabilises the model.

3.1 Bayes nets can model EOQ world

To show that DBNs have the power to model the EOQ world, and to check for DBN model assumptions effects on solution optimality, we first run the standard EOQ policy on the EOQ DBN world model. If it produces similar results to the standard EOQ model then evidence is obtained that the model is good. In particular, the DBN introduces dynamic state over time, and discretizes time steps, both of which could lead to small deviations from the EOQ solution. Table 2 shows the results of the standard EOQ model against the DBN model of the EOQ world and EOQ policy. It can be seen that there is a small difference in the profit. To test whether this was due to the discretisation of time (which could be improved by using smaller time intervals in the DBN) or to other assumptions of the DBN, we computed similar results for a discretised version of standard EOQ, rounding the EOQ policy to integer months and the same order size bins as used in the DBN model. This result was very close to the original EOQ, suggesting that the rest of the discrepancy in the DBN model is due to other assumptions which do not appear in the EOQ model.

Table 2. Expected Average Profit

	Order freq.	size	cost	Hold. cost	Goods cost	Revenue	Profit
Standard EOQ model	4.47	1341.6	67.04	67.08	1200	2400	1065.88
DBN model of EOQ world	4	1500	60	100	1200	2400	1040
Discretised EOQ model	4	1500	60	75	1200	2400	1065

3.2 RL policy beats EOQ in relaxed world

Table 3 shows results from running the EOQ policy in the relaxed world, which includes stochastic customer demand, and variable holding and order costs. It also shows the results of a policy learned by reinforcement learning – built into our software system and usable by non-specialist modellers. Although the overall customer demand level is the same as the EOQ world, its fluctuations over time give rise to backlogs and unsatisfaction costs. The RL policy is able to adapt to

these fluctuations and make more profit than the rigid EOQ policy. (The overall profit is lower than in the EOQ *world* because the demand fluctuations make inventory management into a harder problem in general.)

Table 3. EOQ vs RL policy, in Relaxed world

	Holding cost	Unsatisfaction	Revenue	Profit
EOQ policy, on relaxed world	120.5	959.87	2399.63	61.28
RL policy, on relaxed world	116.15	825.60	2200.33	112.76

4 Discussion

The first set of results in table 2 show that the DBN formalism – recently made accessible to non-specialist managers through off-the-shelf software systems – has the power to model the standard EOQ world. The results show that the DBN simulation gives similar results to the discretised version of the EOQ policy, which is turn gives a close match to the performance of the continuous EOQ policy. In practice discretisation is often a real-world requirement, with stock being ordered in large standardised unit sizes. The discrepancy between the discrete EOQ math model and the DBN simulation is small, but should be explained.

We believe the discrepancy is due to latency factors relating to ordering of node updates. For example, demonstrating EOQ policy over a series of 12 time steps causes a cost discrepancy where a unit of stock is held until it can be sold fulfilling a customer order. The EOQ policy requires the ordering 3 units of stock four times per 12 time steps. Ordering of stock starts at time period 1. At time period 1 there are only two customer orders available (one customer order at time period 0, and one customer order at time period 1). This requires the third unit of stock to be held until another constant customer order is received at time period 2, at which point it is sold, and there are no units of stock left. This pattern therefore repeats every third time step and incurs additional holding cost over the EOQ model at each repetition. Latency factors could be made arbitrarily small by using smaller time steps in the model, at the expense of computation time.

Table 3 shows that EOQ policy is poor in the more realistic, relaxed world. Sticking to a predetermined, deterministic policy gives poor results when the customer orders and backlog are dynamic. In contrast, the profit is increased in this world when a dynamic, reinforcement learning policy is used, which can alter its ordering behaviour as a function of these variables.

Together, our results give a proof-of-concept example that shows how Dynamic Bayesian Network software systems can improve over standard EOQ inventory methods. Unlike previous Markov Decision Process approaches, such

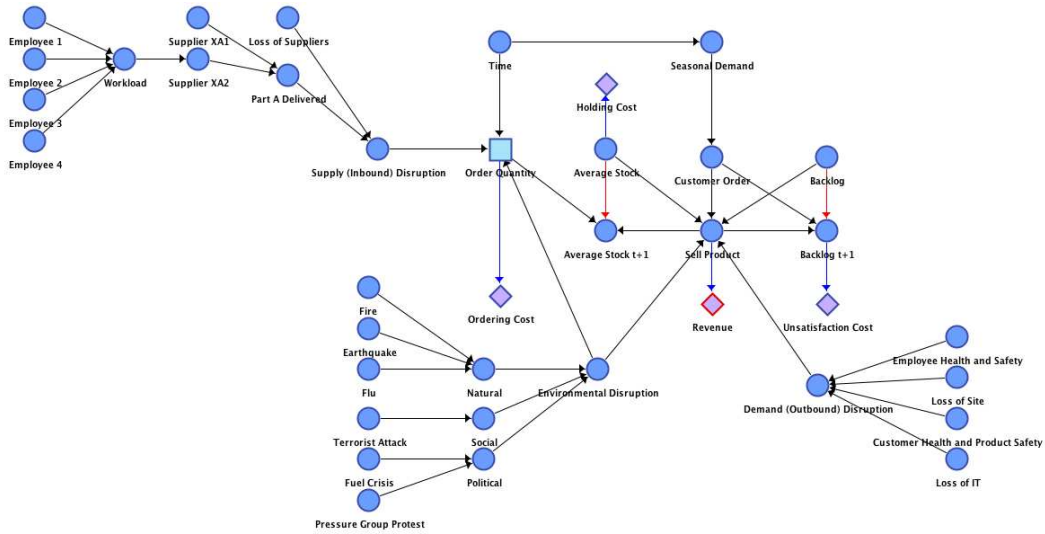


Fig. 3. Example of an extended SCRM Dynamic Bayesian network

software systems do not require the application of advanced math by end-users, rather they allow graphical representations of the Bayesian networks to be constructed by managers in an intuitive 'point and click' manner. This allows the manager to focus on constructing a realistic world model rather than on the details of inference and training algorithms. In contrast to recent fuzzy inventory approaches which solve a similar task, [7], DBNs have precise probabilistic semantics; in contrast to (s,S) inventory systems, DBNs are more general, allowing the policy to depend on factors other than the current stock level such as time of year.

We found that a problem with practical Bayesian network reinforcement learning is that it requires large amount of computation time to find good policies. The simple inventory policy used in table 3 took several hours to learn on an Apple iBook laptop, and the time typically grows exponentially with the number of parent nodes of the action nodes and the number of states in those parents. This is because the number of possible policies scales with these values and any learning algorithm needs to search amongst these policies for the best one. However for economically important decisions it is possible to buy supercomputer or cloud computing time to reduce the search time by arbitrary amounts.

A further advantage of dynamic Bayesian network software systems over existing MDP methods is the potential for extensibility of the models, both within the manager's organisation and beyond it into its supply chain partners. Bayesian networks allow incremental model refinement by replacing generic prior nodes with more detailed world models. As an example of this process, fig 3 shows an extended, supply chain risk version of the inventory model. Three sources of uncertainty have been modelled in detail: inbound, outbound, and environmental

factors. Inbound disruption can be caused by one or more suppliers failing to deliver one or more parts required to manufacture. In this example, supplier XA2 is a collaborating supply chain partner who has provided internal data about its individual employees workloads. By including this information into the Bayesian network we can made more detailed inferences about the supply side, and find better policies depending on them, such as distributing orders across multiple suppliers as a function of the dynamic internal states of those suppliers. On the outbound side, we can model potential problems with our client's state such as loss of site or IT facilities which would affect their ability to take our deliveries. Internal to our own manufacturing processes, we can model the effects of potential disruptions such as earthquakes, fuel crises or epidemics. Each of these nodes contains a CPT table, whose values can be estimated from historical data (e.g. the number of earthquakes in the location of our factory) or by experts (e.g. political analysts forecasts of terrorist risks). Unlike EOQ and standard MDP approaches to inventory, the Bayesian network formalism, as illustrated here, allows seamless integration of such detailed risk models with inventory policy, both within the organisation and across the supply chain partners. Bayesian network software systems thus provide a common language through which to integrate supply chain management, enterprise management and inventory policy, which can be spoken and used by non-specialist inventory managers through intuitive graphical tools.

Bibliography

[1] M. A. Arbib, editor. *The handbook of brain theory and neural networks*, chapter Bayesian Networks. MIT Press, 2003.

[2] S. Buffet. A Markov model for inventory level optimisation in supply chain management. *Eighteenth Canadian Conference on Artificial Intelligence*, pages 133–144, 2005.

[3] D. Erlenkotter. Ford Whittman Harris and the economic order quantity model. *Operations Research*, 38(6):937–946, 1990.

[4] I. Giannoccaro and P. Pontrandolfo. Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics*, 78:153–161, 2002.

[5] R. A. Howard and J. E. Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, 2005.

[6] D. Kayne. *Managing Risk and Resilience in the Supply Chain.* BSI British Standards Institutio, 2008.

[7] D. Kofjac, M. Kljajic, A. Skraba, and B. Rodic. Adaptive fuzzy inventory control algorithm for replenishment process optimization in an uncertain environment. In *Business Information Systems 10th International Conference, BIS 2007*, pages 536–548, 2007.

[8] K. Lysons and M. Gillingham. *Purchasing and Supply Chain Management.* Prentice Hall, 6th edition, 2003.

[9] K. P. Murphy. Dynamic Bayesian networks: Representation, inference and learning. Technical report, University of California, Berkeley, 2002.

[10] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers, 2nd edition, 1988.

[11] E. L. Porteus. *Foundations of stochastic inventory theory.* Standford University Press, 2002.

[12] N. Slack, S. Chambers, and R. Johnston. *Operations Management.* Prentice Hall, Financial Times, Harlow, Essex England, 5th edition, 2007.

[13] R. Sutton and A. Barto. *Reinforcement Learning.* MIT Press, 1998.

[14] N. L. Zhang, R. Qi, and D. Poole. A computational theory of decision networks. *International Journal of Approximate Reasoning*, 11:83–158, 1994.

[15] Q. Zhao, S. Chen, S. Leung, and K. Lai. Integration of inventory and transportation decisions in a logistics system. *Transportation Research Part E*, 46:913–925, 2010.

Mapping with sparse local sensors and strong hierarchical priors

Charles W. Fox and Tony J. Prescott

Active Touch Laboratory at Sheffield
University of Sheffield, UK.
charles.fox@sheffield.ac.uk

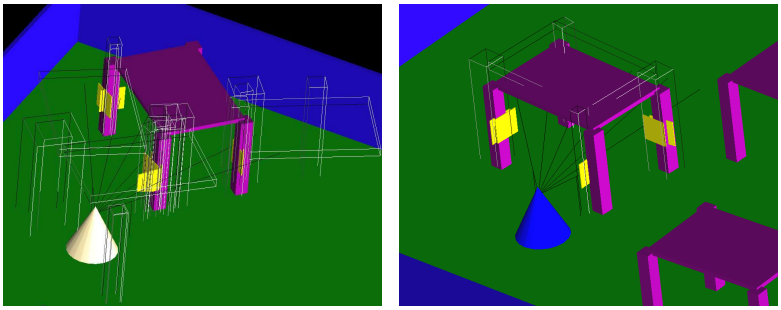
Abstract. The paradigm case for robotic mapping assumes large quantities of sensory information which allow the use of relatively weak priors. In contrast, the present study considers the mapping problem in environments where only sparse, local sensory information is available. To compensate for these weak likelihoods, we make use of strong hierarchical object priors. Hierarchical models were popular in classical blackboard systems but are here applied in a Bayesian setting and novelly deployed as a mapping algorithm. We give proof of concept results, intended to demonstrate the algorithm's applicability as a part of a tactile SLAM module for the whiskered SCRATCHbot mobile robot platform.

1 Introduction

The paradigm case for robotic mapping, as in Simultaneous Localisation and Mapping (SLAM) problems [27], considers a mobile robot with noisy odometry and laser scanners. Laser scanners provide large amounts of sensory information, and have effectively unlimited range in indoor environments. Such large quantities of input information allow the use of relatively weak priors, such as independent grid cell occupancy or flat priors over the belief of small feature sets [27].

In contrast, we consider the mapping problem in environments where only sparse, local sensory information is available. For example, a fire-fighting robot building up a map in a smoke-filled house cannot rely on vision or laser scanners functioning at all times, and could instead operate by feeling its way around with touch sensors. Proof that this type of navigation is possible is found in biology: electric fish make use of highly localised electric field sensors [13] and rats navigate through dark underground tunnels using their whiskers [4, 2], both having ranges of a few centimetres. In robotics, touch sensors are relatively cheap in both material and computational processing terms, and their use has previously been considered to enhance navigation in cheap household robots [17, 6]. (Related work on research robot platforms includes [24, 23, 15, 14, 8]).

As an example of this type of mapping, we consider the case of a mobile robot having six whiskers, able to report the (noisy) locations and orientations of contacts with surfaces. Such mechanical sensors and computational classifiers have previously been demonstrated in [17, 6, 7], and are able to report locations, orientations and textures of contact points (note that textures are especially difficult to report using other sensor modalities). The present mapping algorithm is intended to form part of a future



(a) (b)

Fig. 1. (a) Simulation screen-shot at high annealing temperature. Many hypothesised (wire-frame) tables and legs are on the blackboard, primed by the shapelets (yellow rectangles) contacted by the robot (cone)'s whisker sensors, in an arena containing a physical table (pink). (b) Simulation screen-shot at low annealing temperature. A single table hypothesis remains, aligned correctly with the physical table.

SLAM navigation module for the whiskered SCRATCHbot hardware platform [21], but here we give a proof of concept mapping-only algorithm in a simulated and simplified microworld. It is the first work to begin fusing whisker contact reports to perform mapping.

To compensate for the sparseness of the sensory information available from short-range touch sensors, we make use of strong, hierarchical priors about objects in the world. Hierarchical object recognition models were popular in classical, symbolic AI in the guise of blackboard systems [5, 19, 3] but have recently been recast in terms of dynamically constructed Bayesian networks [10, 18, 16, 26]. Here we provide a novel application of Bayesian blackboards to the robotic mapping problem.

Object based mapping models have recently appeared [28, 11, 25, 22] which use laser sensors to recognise and learn complex spatial models. However in the sparse local sensor case, this level of detail is unavailable, and only a few contact points may be present. Thus we go beyond the use of individual movable objects, to use strong hierarchical model priors. For example, on recognising a table leg, we may then infer the probable presence the rest of the table, including other leg objects, and edges and corners making up these legs, without ever sensing them directly. To construct hierarchical objects, we use hypothesis priming and pruning heuristics as in blackboard systems. However, following [10], we treat such heuristics as approximations to inference in a dynamically-constructed, Monte Carlo Markov Chain (MCMC) sampling Bayesian network, endowing them with probabilistic semantics.

2 Methods

Consider the task of building a map of an arena populated by four-legged table-like objects as in figs. 1(a) and 1(b). (Such objects could include chairs and desks for example). A mobile whiskered agent moves along a predetermined trajectory of location-angle

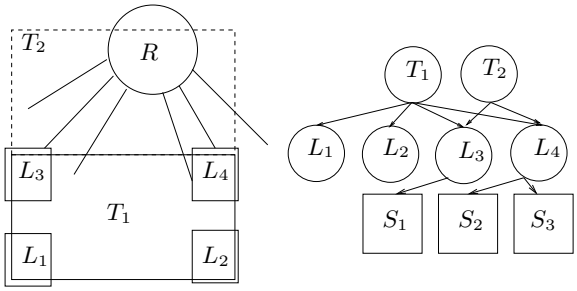


Fig. 2. Hierarchical object recognition. *Left:* Robot R (circle) with six whiskers (lines) makes tactile contact with legs L_j (squares) of a hypothesised table T_1 (rectangle). The two contact points ('shapelets') on the right are sufficient to infer the location of the corner of leg L_4. Coupled with prior knowledge about the shape and size of tables, and the third shapelet, this can be used to infer that there is a table either in the ground truth location or in a second configuration T_2 (dashed rectangle). *Right:* Bayesian network constructed to represent the same scenario. Square nodes are the shapelet observations.

poses, (x^t, y^t, θ^t), around the arena, over discrete time steps t. At each time step, its whiskers, $w \in 1:6$, report egocentric estimates of the radial distance r to, and surface normal ϕ and texture τ of, any contacts made,

$$\hat{r}^t_w = r^t_w + \varepsilon_r, \tag{1}$$

$$\hat{\phi}^t_w = \phi^t_w + \varepsilon_\phi, \tag{2}$$

$$\hat{\tau}^t_w = \tau^t_w + \varepsilon_\tau, \tag{3}$$

where ε are i.i.d. Gaussian noises having zero mean and standard deviations $\sigma^w_r, \sigma^w_\phi, \sigma^w_\tau$ respectively. (Such reports are currently available from the whiskered SCRATCHbot hardware platform, [21, 17, 6]). Assuming perfect robot localisation in the present study only, these estimates are converted into allocentric Cartesian coordinates to give *shapelet reports*, which are tuples $S(x_S, y_S, \phi_S, \tau_S)$.

2.1 Static structures: generative models

Tables, T, are parametrised by tuples, $T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)$, where x, y, θ is the pose, w^x_T and w^y_T are width and breadth, w^L is the width of the (square) legs, and $\tau \in (0, 1)$ is a texture parameter describing roughness or smoothness of the material. A generative model of tables is used. We assume a flat prior probability *density* generating tables in the world,

$$p(T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)|\emptyset) = c_T, \tag{4}$$

where c_T is a (non-normalising) constant.

If a table T exists, its presence causes (as in [20]) the presence of four leg objects,

$$L(x_L, y_L, \theta_L, w_L, \tau_L, T), \tag{5}$$

where w_L is the width of the square table leg; x_L, y_L, θ_L are its location and rotation, and τ_L is its texture, with probability density

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, T)|T(x_T, y_T, \theta_T, w_T^L, \tau_T)) \\ = \alpha_L \exp\left(-\Delta_{TL}\right), \tag{6}$$

where α is a (non-normalising) constant, and the distance measure is

$$\Delta_{TL} = \min_i \left(\frac{(x_T^i - x_L)^2 + (y_T^i - y_L)^2}{\sigma_r^2} \right) + \left(\frac{\theta_T - \theta_L}{\sigma_\theta} \right)^2 \\ + \left(\frac{w_T^L - w_L}{\sigma_w} \right)^2 + \left(\frac{\tau_T - \tau_L}{\sigma_\tau} \right)^2, \tag{7}$$

where $0 \leq i \leq 3$, and (x_T^i, y_T^i) are the coordinates of the table's four corners, and σ_w, σ_τ are parameters specifing standard deviations of the leg's w_L, τ_L values conditioned on the table's corresponding w_T^L, τ_T values. The inclusion of T in the parametrisation of L (eqn. 5) means that L is the hypothesis that the leg was caused *only* by table T rather than any other table or cause.

Shapelets are assumed to be generated by nearby legs,

$$p(S(x_S, y_S, \theta_S, \tau_S)|L(x_L, y_L, \theta_L, w_L, \tau_L, T)) = \alpha_S \exp\left(-\Delta_{LS}\right) \tag{8}$$

where

$$\Delta_{LS} = \left(\frac{r}{\sigma_r^S} \right)^2 + \left(\frac{f(\theta_L) - \theta_S}{\sigma_\theta^S} \right)^2 + \left(\frac{\tau_L - \tau_S}{\sigma_\tau^S} \right)^2, \tag{9}$$

and r is the shortest radial distance from the perimeter of the leg to (x_S, y_S), computed by basic geometry, $f(\theta_L) = \theta_L + m\pi/2$ picks the angle of the corresponding side m of the leg at this shortest-distance contact point, and $\sigma_r^S, \sigma_\theta^S, \sigma_\tau^S$ model sensor noise.

We also provide small *null priors* to allow legs and shapelets to exist in the absence of any generative parents. (These are required later, during construction on the blackboard, so that these objects can survive before their parents are constructed),

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, \emptyset)|\emptyset) = c_L, \tag{10}$$

$$p(S(x_S, y_S, \theta_S, \tau_S)|\emptyset) = c_S, \tag{11}$$

with constants such that the marginalised densities,

$$p(S(x_L, y_L, \theta_L)) < p(L(x_L, y_L, \theta_L)) < p(T(x_L, y_L, \theta_L)), \tag{12}$$

i.e. larger objects are more probable to exist without high-level causes than smaller objects are.

Unlike the parametrisation of L on T in eqn. 5, shapelets may be caused by *mixtures* of multiple leg hypotheses and by the null prior (eqn. 11). For example if there are two legs very close together then the density for observing shapelets in the area increases. We assume that multiple causal sources combine using noisy-OR semantics,

$$P(x_i|pa(x_i)) = 1 - \prod_{x_j \in pa(x_i)} \left(1 - P(x_i|x_j)\right). \tag{13}$$

where $pa(x_i)$ denotes the set of parents of generic node x_i. As we use probability *density* functions we require the continuous version of noisy-OR, proved in the Appendix:

$$p(x_i|pa(x_i)) = \sum_{x_j \in pa(x_i)} p(x_i|x_j). \tag{14}$$

We allow legs to be caused by a mixture of their *single* specified parent (i.e. the T parameter in eqn. 6) and null prior (eqn. 10), using a similar combination rule. Tables are caused by the null prior only (eqn. 4).

Taken together, the equations in this section define a Bayesian network for any given collection of tables, legs and shapelets as shown in fig. 2. However, in addition to the previous causal probabilities, we need to model the following constraints: (a) tables always have four legs; (b) each table leg is at a different corner of the table (we should not see two legs attached to the same corner); (c) two objects of the same type (table or leg) cannot overlap in physical space. Standard Bayesian networks cannot model such relations, as they are limited to joint distributions of the form

$$P(\{x_i\}_i) = \prod_i P(x_i|pa(x_i)), \tag{15}$$

To model these constrains, we extend the Bayesian network to the factor graph,

$$P(\{x_i\}_i) = \frac{1}{Z}\left(\prod_i P(x_i|pa(x_i))\right) \times \left(\prod_{ij} \phi_c(x_i,x_j)\phi_b(x_i,x_j)\right)\left(\prod_i \phi_a(x_i)\right), \tag{16}$$

where Z is a normalising constant, and ϕ_a, ϕ_b, ϕ_c are unnormalised penalty factors corresponding to the new constraints. Using superscripts for exponentiation, these are

$$\phi_a(x_i) = \epsilon_a^m, \tag{17}$$

$$\phi_b(x_i,x_j) = \epsilon_b^v, \tag{18}$$

$$\phi_c(x_i,x_j) = \epsilon_c^r, \tag{19}$$

where m is the number of missing legs iff x_i is a table, and $m = 0$ otherwise; v is a Boolean (0,1) value, true if hypotheses x_i and x_j are of the same type and overlap in physical space; and r is a Boolean, true if hypotheses x_i and x_j are legs and share the same parent (modelling this parent-sharing is why we parametrise L by T in eqn. 5).

2.2 Inference

For a given set of shapelet observations and a set of candidate hierarchical legs and tables, we may thus construct a factor graph. (We later describe how such a set of candidates is obtained automatically). Inference becomes highly complicated if the agent has an infinite memory for shapelets, so in the present study we use a working memory (queue) of the seven most recent shapelets, and discard all others.

At each t, new shapelets are read from the sensors, and inference is performed with the aim of obtaining the Maximum A Posterior (MAP) interpretation of their table causes, before the next time step begins,

$$\mathrm{MAP_t} = \arg_{\{T_j\}} \max P(\{T_j\}_j | \{S_k\}_k). \tag{20}$$

Thus we currently – naively – treat each time step as an independent inference problem. Limiting inference to the most recent shapelets also has the effect of working within a local 'fovea' of attention: if no recent shapelets are from distant areas, then only hypotheses around the agent's location will be considered.

Algorithm 1 Approximate Metropolis-hasting proposals generation.

```
for each time step t do
    update shapelet queue S by reading sensors
    for each annealing inverse temperature β do
        for each shapelet S_i ∈ S do
            propose and test parent H_i from Q(pa(S_i))
            if accepted, add H_i to hypothesis set B
        end for
        for each hypothesis H_i ∈ B do
            r ← rand(0, 1)
            if r < r_1 then
                propose death of H_i
                if accepted, remove H_i from B
            else
                if r < r_2 then
                    propose parent change for H_i
                    if accepted, replace H_i's parent parameter
                else
                    if r < r_3 then
                        propose child H_j from Q(ch(H_i)|H_i)
                        if accepted, add H_j to hypothesis set B
                    else
                        propose parent H_j from Q(pa(H_i)|H_i)
                        if accepted, add H_j to hypothesis set B
                    end if
                end if
            end if
        end for
        prune all hypotheses not linked to any shapelet directly or via a common ancestor.
    end for
end for
```

There is some subtlety in defining the meaning of MAP states in continuous parameter spaces. In the present study, we assume that discrete hypotheses $H_i(x, y, \theta, \Theta)$ (where $H \in \{S, L, T\}$) represent small but non-infinitesimal collections of possible

(x, y, θ) poses, with probability

$$P(H((x - \tfrac{\delta}{2}, x + \tfrac{\delta}{2}), (y - \tfrac{\delta}{2}, y + \tfrac{\delta}{2}), (\theta - \tfrac{\delta}{2}, \theta + \tfrac{\delta}{2}), \Theta))$$

$$= \delta^3 p(H(x, y, \theta, \Theta)), \tag{21}$$

where δ is a small but nonzero constant, Θ are the remaining parameters, and p is the density.

We use the annealed [1] approximate Metropolis-Hastings sampler of algorithm 1 to perform inference. Unlike standard inference problems, object-based mapping is a form of scene analysis task, i.e. the number of objects in the world – and therefore the number and type of nodes in the network – is unknown in advance. Algorithm 1 uses blackboard-like priming and pruning heuristics integrated with the sampling, to control the size of the network. Each hypothesis in the current 'blackboard' set B maintains (amongst other parameters), pose parameters x, y, θ and a current parent. The current parent may be another hypothesis, or may be null. Importantly, hypotheses that are not currently 'true' (according to the sampler) are never stored in B. The set B acts as a factor graph as detailed in the previous section, and may be thought of as the contents of a blackboard [5].

To obtain unbiased samples from the true joint distribution, Metropolis-Hastings sampling requires detailed technical conditions to be met, which are complicated by the jumps between factor graphs of different structures and sizes. Reversible jump methods [12] provide a rigorous theoretical basis from which to define acceptance probabilities based on reweighting proposals. Future work should incorporate such theory, for now we heuristically choose the Q distributions[1] and r_i thresholds; and use the annealed original P distribution from the factor graph as a simple Gibbs [1] acceptance probability,

$$P(\text{accept } H_i) = P^{\beta}(H_i | mb(H_i)), \tag{22}$$

where $mb(H_i)$ is the Markov blanket of H_i containing its parents, rivals $riv(H_i)$, and children $ch(H_i)$, β is inverse temperature. The Markov blanket conditional is

$$P(H_i|mb(H_i)) = P(H_i|pa(H_i), cop(H_i), ch(H_i), riv(H_i))$$

$$= \frac{1}{Z} \frac{\Phi_a \Phi_b \Phi_c P(H_i|pa) P(ch|H_i)}{P(ch|(H_i))P(H_i|pa) + P(ch|\neg H_i)P(\neg H_i|pa)}$$

$$= \frac{1}{Z} \frac{\Phi_a \Phi_b \Phi_c p(H_i|pa) p(ch|H_i)}{\delta^3 (p(ch|H_i)p(H_i|pa) + p(ch|\neg H_i)p(\neg H_i|pa))},$$

where Z normalizes the factors contribution $\Phi_a \Phi_b \Phi_c$ only; δ is the constant of eqn. 21; $\Phi_a = \phi_a(H_i) \prod_{j \in pa(i)} \phi_a(H_j)$ includes missing children of H_i and also the missing child penalty for each parent of H_i which would have a missing child in the case where H_i is false; $\Phi_b = \prod_{j \in mb(i)} \phi_b(H_i, H_j)$ and $\Phi_c = \prod_{j \in mb(i)} \phi_c(H_i, H_j)$. The update

[1] details can be found in the source code, however note that MH sampling can operate on *any* proposal Q so its precise form is unimportant. Better results are obtained as the approximate Q becomes close to the true P.

allows computation to proceed using density functions rather than probabilities, but depends on the choice of the small constant, δ.

Newly proposed nodes must be linked to existing ones, so it is necessary to locate all potential parents $pa(H_i)$. A threshold radius in pose space is used, which limits this set to candidates which are close enough to have non-negligible generating probabilities, i.e.

$$pa(H_i) := \{H_j : P(H_i|H_j) \gg 0\}. \tag{23}$$

For computational efficiency it is useful to implement a spatial hash-table to look up these nearby hypotheses. This hash-table may also be reused to look up overlapping hypotheses in the computation of ϕ_c.

3 Results

We have implemented a simple simulation of a whiskered robot in a world populated by six four-legged, table-like objects, in a simple mapping task. The simulation is coded in C++ using the ODE physics engine (`www.ode.org`) for collision detection. Source code is available on request, and contains handset values for all parameters described in this paper. The sensor noise levels are comparable to those found in strain template shapelet classifiers [9]. The agent follows a fixed sequences of poses around the world and runs algorithm 1 once at each pose. There are $10 \times 10 \times 4$ poses, from 10 discrete x and y positions and four compass θ angles, as shown in fig. 3. To further simplify the present simulation, tables and table hypotheses all have fixed identical w_T^x, w_T^y and τ_T parameters; and physical (but not hypothesis) tables and have fixed identical w_T^L parameters.

Steps in the inference are illustrated in the supplemental video material. The MAP hypothesis sets from all poses are collated and plotted onto a map of the arena in fig. 4. Comparing against the ground truth in fig. 3, the collated plot shows that table hypotheses are usually found in the correct locations, corresponding to the real tables. The average number of whiskers contacting tables at each pose having at least one table contact is 4.2$\pm$1.7. As we would expect from such a sparse amount of data, there are thus many incorrect hypotheses found in MAPs of the form shown in fig. 2. These are created from poses which do not provide enough information about the tables to resolve ambiguities, for example when the robot is close enough to touch two legs but no third leg as in fig. 2. Also of interest in the results are the many table hypotheses perceived around the edge of the arena. These are due to the agent observing shapelets from contact with the walls around the arena. The system does not (yet) have perceptual models of walls, so the best available explanations for such shapelets are those which postulate tables with legs at these shapelet locations. (This is a form of perceptual relativism: lacking a WALL concept, the system explains the data using its best available TABLE theories.) Similar plots for noiseless and highly noisy sensor cases are shown in figs. 5(a) and 5(b) for comparison. In both cases, the approximate locations of inferred tables are similar, though the accuracy of inferred table poses depends on the noise.

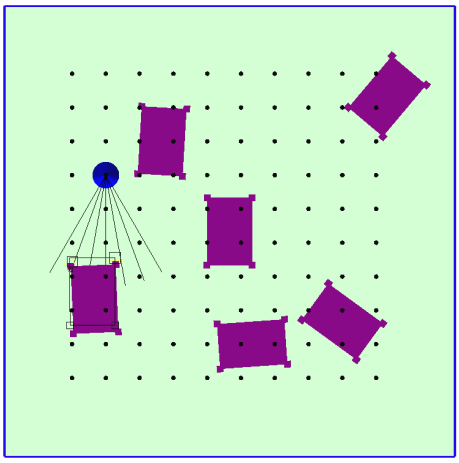


Fig. 3. Overhead view showing ground truth table configuration, and locations (black dots) of the discrete poses occupied by the robot. There are four angle poses at each location, facing in compass directions.

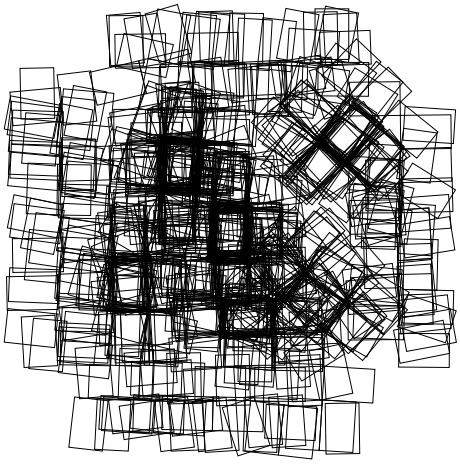
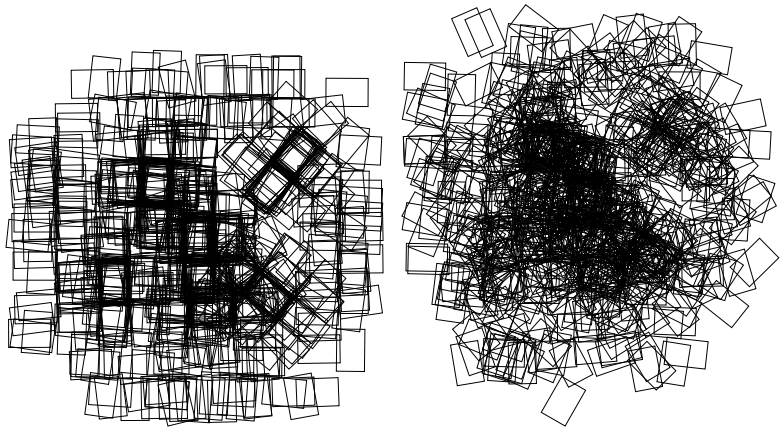


Fig. 4. Montage showing collection of inferred tables from each independent robot pose, for realistic [9] ($\sigma_r = 0.1, \sigma_\theta = \pi/32$) sensors.



(a) (b)

Fig. 5. (a) Montage showing the collection of inferred tables from each independent robot pose, for ideal, noiseless sensors. (b) Montage showing the collection of inferred tables from each independent robot pose, for very noisy ($\sigma_r = 0.5, \sigma_\theta = \pi/8$) sensors.

4 Discussion

We have presented a proof-of-concept implementation of a novel framework for hierarchical object-based mapping from sparse local sensors, such as whiskers and other types of touch; but also applicable to sensors such as low-power or covert short range scanners; or local field-based sensors as used by electric fish.

Many simplifications were made in this proof-of-concept, which future versions of the system should relax. The results presented here are simply the collation of many independent $\mathrm{MAP_t}$ inferences made from the different poses, and no information is shared between poses. Storing longer-term memories of shapelets and fusing them into the inferences would obviously allow a more refined map of the arena to be constructed: at present each table shown in the results has been inferred from typically 4.2 ± 1.7 shapelets only, which is extremely sparse. The present system makes no use of negative evidence, i.e. the observed absence of shapelets on non-contacting whiskers: this could be used to remove some of the ambiguous percepts. The heuristic threshold constants in the proposal distribution should be replaced with Reversible Jump MCMC reweightings to remove bias in the sampling distribution (although in practice the heuristic thresholds can work well, as ultimately only the annealed MAP is sought, rather than an approximation to the whole distribution).

Importantly, the proof-of-concept simulation operates in a world having only one size and texture of table (though tables may have different leg sizes). Enlarging the parameter space to range over tables sizes and textures will allow inference of more realistic four-legged objects such as different kinds of chairs and desks. Other types of objects could also be introduced, such as walls, kitchen units and radiators. The Bayesian blackboard architecture is able to automatically select between rival object models, treating them as rival hypotheses [10]. However, as the number of models and

parameters grows, sampling of course becomes less efficient. For example, it becomes less probable that a perfectly-fitting table will ever be proposed. (Even though once proposed, it will tend to remain accepted for having such a good fit.) We plan to investigate the use of 'smart proposals' which are classical heuristic object detectors (e.g. Hough transforms to find edges and corners) but re-purposed as Metropolis-Hastings proposals in the Bayesian Blackboard. When combined with RJ-MCMC acceptance probabilities, this gives a way to speed up the proposals but retain the probabilistic semantics.

We next hope to extend our implementation to recognise several types of object of varying size, and move from simulation to the SCRATCHbot platform [21], which is currently able to report shapelets of the form used in simulation. SCRATCHbot includes noisy odometry, so will require our mapping system to function as part of a SLAM system. New forms of loop-closure in SLAM may become possible by recognising different parts of the same hierarchical object.

Acknowledgements This work was supported by EU Framework projects BIOTACT (ICT-215910) and ICEA (IST-027819).

References

1. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines. Wiley (1988)
2. Ahl, A.: The role of vibrissae in behavior: a status review. Veterinary Research Communications 10(1), 245–268 (1986)
3. Binford, T., Levitt, T.: Evidential reasoning for object recognition. IEEE Tranactions on Pattern Analysis and Machine Intelligence (2003)
4. Carvell, G., Simons, D.: Biometric analyses of vibrissal tactile discrimination in the rat. J. Neurosci. 10(8), 2638 (1990)
5. Erman, L., Hayes-Roth, F., Lesser, V., Reddy, R.: The Hearsay-II system. ACM Computing Surveys 12(2) (1980)
6. Evans, M., Fox, C., Pearson, M., Prescott, T.: Spectral Template Based Classification of Robotic Whisker Sensor Signals. Proc. TAROS2009 (2009)
7. Evans, M., Fox, C., Prescott, T.: Tactile discrimination using template classifiers. From Animals to Animats SAB2010. (2010)
8. Fend, M.: Whisker-based texture discrimination on a mobile robot. Advances in Artificial Life pp. 302–311 (2005)
9. Fox, C., Pearson, M., Mitchinson, B., Pipe, T., Prescott, T.: Simple features for texture classification. Somatosensory and Motor Research 24(3), 139–162 (2007)
10. Fox, C.: ThomCat: A Bayesian blackboard model of hierarchical temporal perception. In: Proc. FLAIRS (2008)
11. Gallagher, G., Srinivasa, S.S., Bagnell, J.A., Ferguson., D.: Gatmo: A generalized approach to tracking movable objects. In: ICRA (2009)
12. Green, P.: Reversible jump Markov chain Monte Carlo computation. Biometrika 82(4), 711–732 (1995)
13. Heiligenberg, W.: Neural nets in electric fish. MIT (1991)
14. Kaneko, M., Kanayama, N., Tsuji, T.: Active antenna for contact sensing. IEEE Transactions on robotics and automation 14(2), 278–291 (1998)
15. Kim, D., Moller, R.: Biomimetic whiskers for shape recognition. Robotics and Autonomous Systems 55(3), 229–243 (2007)
16. Laskey, K.B., da Costa, P.C.: Of starships and klingons: Bayesian inference for the 23rd century. In: Proc. UAI (2005)

17. Lepora, N., Evans, M., Fox, C., Diamond, M., Gurney, K., Prescott, T.: Naive Bayes texture classification applied to whisker data from a moving robot. Proc. IEEE WCCI (2010)
18. Milch, B.: Probabilisitic Models with Unknown Objects. Ph.D. thesis, UC Berkeley (2006)
19. Mitchell, M.: Analogy-Making as Perception. MIT (1993)
20. Pearl, J.: Causality. Cambridge University Press (2000)
21. Pearson, M.J., Mitchinson, B., Welsby, J., Pipe, A.G., Prescott, T.J.: Scratchbot: Active tactile sensing in a whiskered mobile robot. From Animals to Animats SAB2010. (2010)
22. Petrovskaya, A., Khatib, O., Thrun, S., Ng, A.Y.: Touch based perception for object manipulation. In: ICRA (2007)
23. Schultz, A., Solomon, J., Peshkin, M., Hartmann, M.: Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. Proc. ICRA2005 (2005)
24. Seth, A., McKinstry, J., Edelman, G., Krichmar, J.: Texture discrimination by an autonomous mobile brain-based device with whiskers. In: Proc. IEEE ICRA (2004)
25. Srinivasa, S., Ferguson, D., Helfrich, C., Berenson, D., Romea, A.C., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., Vandeweghe, J.M.: Herb: a home exploring robotic butler. Autonomous Robots 28(1), 5–20 (January 2010)
26. Sutton, C., Burns, B., Morrison, C., Cohen, P.R.: Guided incremental construction of belief networks. In: Proc. Fifth Int. Symp. Intelligient Data Analysis (2003)
27. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT (2006)
28. Wang, C.C., Thorpe, C., Thrun, S., Herbert, M., Durrant-Whyte, H.: Simultaneous localization, mapping and moving object tracking. Int. J. Robotics Research 26, 889 (2007)

Appendix: Noisy-OR density combination

Let Y_i range over nodes $pa(X)$ in a continuous-valued Bayesian network with noisy-OR parent combinations,

$$P(X|\{Y_i\}_i) = 1 - \prod(1 - P_i). \tag{24}$$

with $P_i = P(X|Y_i)$. Consider the probability of a small range of hypotheses,

$$\delta^3 p(X|\{Y_i\}_i) = 1 - \prod(1 - \delta^3 p_i), \tag{25}$$

where p are probability densities and P are probabilities. Expansion terms with powers of δ that are > 3 vanish, so

$$\delta^3 p(X|\{Y_i\}_i) = \delta^3 \sum p_i. \tag{26}$$

The δ^3 terms cancel to yield

$$p(X|\{Y_i\}_i) = \sum p_i. \tag{27}$$

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220942084

A General Classifier of Whisker Data Using Stationary Naive Bayes: Application to BIOTACT Robots.

Conference Paper · January 2011

Source: DBLP

CITATION
1

READS
29

11 authors, including:

Nathan Lepora
University of Bristol
69 PUBLICATIONS **594** CITATIONS

SEE PROFILE

Mathew H Evans
The University of Manchester
25 PUBLICATIONS **239** CITATIONS

SEE PROFILE

J. Charlie Sullivan
Bristol Robotics Laboratory
22 PUBLICATIONS **181** CITATIONS

SEE PROFILE

Kevin Gurney
The University of Sheffield
183 PUBLICATIONS **5,753** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Green Brain View project

All content following this page was uploaded by Kevin Gurney on 13 August 2014.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

A General Classifier of Whisker Data using Stationary Naive Bayes: Application to BIOTACT Robots

Nathan F. Lepora[1], Charles Fox[1], Mat Evans[1], Ben Mitchinson[1], Asma Motiwala[1], J. Charlie Sullivan[2], Martin J. Pearson[2], Jason Welsby[2], Tony Pipe[2], Kevin Gurney[1], and Tony J. Prescott[1]

[1] Adaptive Behaviour Research Group, Department of Psychology, University of Sheffield, UK.
{n.lepora,c.fox,mat.evans,b.mitchinson,a.motiwala, k.gurney,t.j.prescott}@sheffield.ac.uk

[2] Bristol Robotics Laboratory, Bristol Business Park, Bristol, UK.
{charlie.sullivan,martin.pearson,jason.welsby,tony.pipe}@brl.ac.uk

Abstract. A general problem in robotics is how to best utilize sensors to classify the robot's environment. The BIOTACT project (BIOmimetic Technology for vibrissal Active Touch) is a collaboration between biologists and engineers that has led to many distinctive robots with artificial whisker sensing capabilities. One problem is to construct classifiers that can recognize a wide range of whisker sensations rather than constructing different classifiers for specific features. In this article, we demonstrate that a stationary naive Bayes classifier can perform such a general classification by applying it to various robot experiments. This classifier could be a key component of a robot able to learn autonomously about novel environments, where classifier properties are not known in advance.

Keywords: BIOTACT, Active touch, Whiskers, Bayes' rule, Classifier

1 Introduction

Robotics has much to learn from the ways that animals are constructed, control their bodies and utilize their sensory capabilities [1]. All of the robots described in this article result from a long-term collaboration between biologists and engineers that aims to further understanding of biological vibrissal (whisker) systems and determine the potential applications to engineered systems such as autonomous robots [2]. Research into these systems was undertaken as part of a European Framework 7 project termed BIOTACT (BIOmimetic Technology for vibrissal Active Touch) [3]. The vibrissal sensing technology used in these robots was inspired by many aspects of biological whiskers [4], including their morphology, control and sensory information processing.

This article focuses on how to process the sensory information from artificial vibrissal sensors to categorize and recognize the robot's nearby environment.

This classification problem relies on characterizing stimuli from the environment that are similar every time they are encountered, for example the speed of a contacting object is typified by the peak whisker deflection it produces [5]. Classifiers of sensory data from artificial vibrissae have generally focussed on distinctive features of the whisker signal produced by the stimulus. Examples include peak deflection amplitude and duration for contact speed and radial distance [5], the profile of the whisker deflection power spectrum for texture [6–8], and the change in peak deflection across many whiskers for surface shape [9]. These classifiers perform well in the situations they are designed for, but are not accurate outside their operating range. For example, peak whisker deflection may not accurately characterize texture.

While it is possible to choose *specific* classifiers for robots sensing artificial environments, in more natural environments it would be useful to employ *generic* classifiers that apply across a broad range of stimuli. Traditionally this goal has not been a focus of classifier design for whiskered robots, because the first priority was to find a set of classifiers that allows the robot to interact with test environments. However, now that a fairly comprehensive library of classifiers is becoming available [5–14], such questions about the choice and flexibility of classifiers are becoming relevant.

In recent work, we found that a classifier based on generic probabilistic methods could recognize surface texture accurately by utilizing the probability distribution (histogram) of whisker deflections over a time window of data [12]. In principle, such a method could also characterize other stimuli by utilizing the overall statistical properties of whisker deflections to determine the salient aspects of each stimulus. To explore the hypothesis that a probabilistic classifier could apply across general types of stimuli, we demonstrate here that one method, stationary naive Bayes, applies well across a broad range of stimuli encountered by several robot platforms constructed for the BIOTACT project.

2 Classification with stationary naive Bayes

The probabilistic classifier used here is based on Bayes' rule, but makes two assumptions for simple application to time series of whisker data. First, the sensor measurements are assumed independent, or naive, and second the measurement distributions are assumed stationary in time. Naive Bayes is usually considered across different data dimensions rather than over time series with the stationary assumption, and so we refer to the present method as stationary naive Bayes.

This method relies on calculating the probability distributions of measured time series from the empirical frequencies with which values occur in classes of training data. The probability of a single measurement x being from a class is

$$P(x|\mathrm{C}_l) = \frac{n_x(\mathrm{C}_l)}{\sum_x n_x(\mathrm{C}_l)}, \tag{1}$$

where $n_x(\mathrm{C}_l)$ is the total number of times that the value x occurs in the time series for class C_l. The conditional probability $P(x|\mathrm{C}_l)$ is commonly referred to as a likelihood of the sensor measurement x occurring.

Given a new set of test data, Bayes' rule states that the (posterior) probability $P(\mathrm{C}_l|x_1,\cdots,x_n)$ for a time-series of measurements $x_1,\cdots,x_n$ being drawn from the training data for class C_l is proportional to the likelihood of those measurements $P(x_1,\cdots,x_n|\mathrm{C}_l)$ estimated from that training data

$$P(\mathrm{C}_l|x_1,\cdots,x_n) = \frac{P(x_1,\cdots,x_n|\mathrm{C}_l)P(\mathrm{C}_l)}{P(x_1,\cdots,x_n)}, \tag{2}$$

where $P(\mathrm{C}_l)$ is the (prior) probability of the data being from class C_l and $P(x_1,\cdots,x_n)$ is the (marginal) probability of measuring $x_1,\cdots,x_n$ given no other information. A Bayesian classifier finds which class C has maximum *a posteriori* probability given the measurement time series

$$\mathrm{C} = \arg\max_{\mathrm{C}_l} P(\mathrm{C}_l|x_1,\cdots,x_n) = \arg\max_{\mathrm{C}_l} \frac{P(x_1,\cdots,x_n|\mathrm{C}_l)P(\mathrm{C}_l)}{P(x_1,\cdots,x_n)}, \tag{3}$$

where arg max refers to the argument (the class) that maximizes the posterior.

The following formalism uses that the marginals $P(x)$ are independent of the classes and can be ignored in the arg max operation. It will also be convenient to use the logarithm of the posterior probability to turn products of probabilities into sums. Then the classification in Eq. 3 is equivalent to finding

$$\mathrm{C} = \arg\max_{\mathrm{C}_l} \ \log P(x_1,\cdots,x_n|\mathrm{C}_l) + \log P(\mathrm{C}_l), \tag{4}$$

which is just the maximum over the log-likelihoods plus log priors of a series of measurements $x_1,\cdots,x_n$ being from each class of training data.

An important simplification occurs if the measurements are assumed statistically independent at each time across the data and the probability distributions from which these measurements are drawn are assumed stationary in time. Then the overall conditional probability of a series of measurements factorizes into a product of conditional probabilities for each individual measurement. Consequently, the classification can be rewritten as

$$\mathrm{C} = \arg\max_{\mathrm{C}_l} \ \sum_{i=1}^{n} \log P(x_i|\mathrm{C}_l) + \log P(\mathrm{C}_l), \tag{5}$$

Thus the most probable class is found from the maximum over the summed log-likelihoods for the time window, shifted by its log-prior. This equation defines the stationary naive Bayes classifier used in this paper, with 'naive' denoting the assumption of statistical independence over time and stationary denoting that the same likelihood distribution applies for all times.

The following examples are based around robot experiments in which similar amounts of data are collected for all classes. There is then no biasing towards any class from prior knowledge of its occurrence frequency. Therefore, we assume that the priors are equal and ignore them in the classification (Eq. 5). Although the classification is then equivalent to maximum likelihood, we retain the priors in the formalism to emphasize the Bayesian nature of the method in that in principle it applies also to situations with *a priori* class knowledge.

3 Experiment 1: Surface texture classification

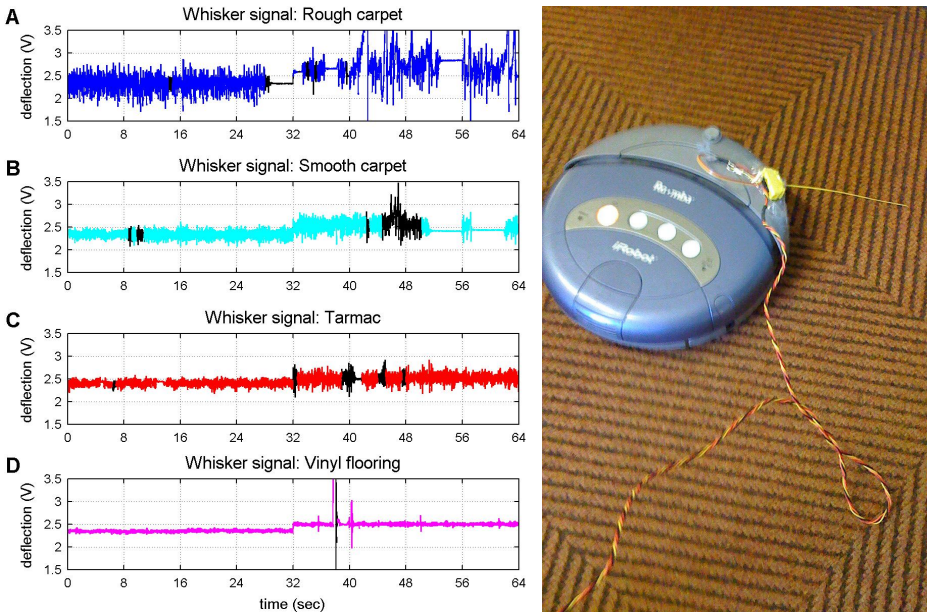


Fig. 1. Surface texture classification with the iRobot Roomba.
The Y-deflection validation data is shown for each of the four textures. On each trace, the first four half-trials of 8 seconds are for anticlockwise motion and the latter four half-trials are for clockwise motion. Hits are shown in color and misses shaded in black.

Methods: Surface texture data was collected from whisker attached to an iRobot Roomba mobile robotic platform [7].

The whisker was mounted on the robot at a 45 degree angle to the direction of forwards motion with a downwards elevation sufficient to contact the floor (Fig. 1). The whisker sensor consisted of a flexible plastic whisker shaft mounted into a short, polyurethane rubber filled tube called a follicle case. Overall dimensions were approximately five-times a rat whisker (200mm long, 2mm diameter), with material properties matched to scaled-up biological whiskers [15].

Four surfaces were chosen for classification: two carpets of different roughnesses, a tarmac surface and a vinyl surface. Deflections of the whisker shaft were measured as 2-dimensional (X,Y) displacements at the base using a Hall effect sensor sampled at 2KHz. In these experiments, the robot moved in a stereotyped rotating manner, spinning either anticlockwise or clockwise (4 trials each of 16 seconds). Here the classification applies over both types of motion.

Results: The data resemble noisy time series interspersed with dead zones, jumps and systematic changes in the mean or variance (Figs 1A-D). The initial 8 seconds of each trial was used for training data, leaving the final 8 seconds for

validation testing. The texture likelihoods were then found from the histograms of the empirical frequencies for measurement values of the four textures (Eq. 1). In previous work, we used likelihood distributions that assumed statistical independence between horizontal (X) and vertical (Y) deflections of the whisker [12]. Here, we build on that study by utilizing a two-dimensional joint-likelihood for classification. This study also aims to present texture classification in the wider context of identifying other aspects of the environment, such as contact speed and distance or shape classification, as discussed later.

For validation, the data was separated into discrete segments of fixed temporal window size 500ms over which the texture is determined. The stationary naive Bayes classifier then considers the log-likelihood values for the four candidate textures from the conditional probability distributions at the same measurement value in the training data. As discussed above, two-dimensional likelihoods over pairs of X and Y deflection values were considered, binning measurements over a grid of resolution 10mV by 10mV. The log-likelihood values over the segments of validation data were then summed for each of the four textures (Eq. 5), with the maximal value giving the classified texture.

The correct classifications, or hits, for each of the four textures is shown on Figs 1A-D by the colored segments of data and the misses by black segments. The classification for all four textures was highly accurate (mean hit rates 89%, 88%, 91% and 99%). These results are apparently little different from those assuming independence between the X and Y directions (*c.f.* [12]). Thus in this situation, a saving in computational complexity can be made by moving to a classifier that naively treats the X and Y values as independent; however, we expect that in general the dependence between separate streams of information is important, and the determination of when it is remains an open research question.

In general, the stationary naive Bayes classifier was robust to changes in robot motion, since an identical classifier was used for both clockwise and anticlockwise rotations. This is perhaps surprising considering that such changes cause large systematic effects in the whisker deflection data (Fig. 1 at 32 seconds). It is also curious that the misses occurred mainly near jumps in the whisker deflection, corresponding to the whisker spuriously catching the floor surface, rather than directly confusing textures on normal data. Thus the accuracy on data without such artifacts would improve upon the 92% mean hit rate found here.

4 Experiment 2: Speed and radial distance classification

Methods: An XY positioning robot moved a vertical bar onto a fixed, horizontal whisker to probe the whisker deflection dependency on contact speed and radial distance [5]. The data set consisted of four repeats of 2626 distinct contact speeds and distances along the whisker shaft.

The whisker sensor was similar to that in the preceding texture study, with slightly different dimensions (whisker length 185mm, tapering from 3mm to 0.5mm) and material properties again scaled up from biological rat whiskers [15]. Objects were moved onto the whisker and when contact was detected the robot

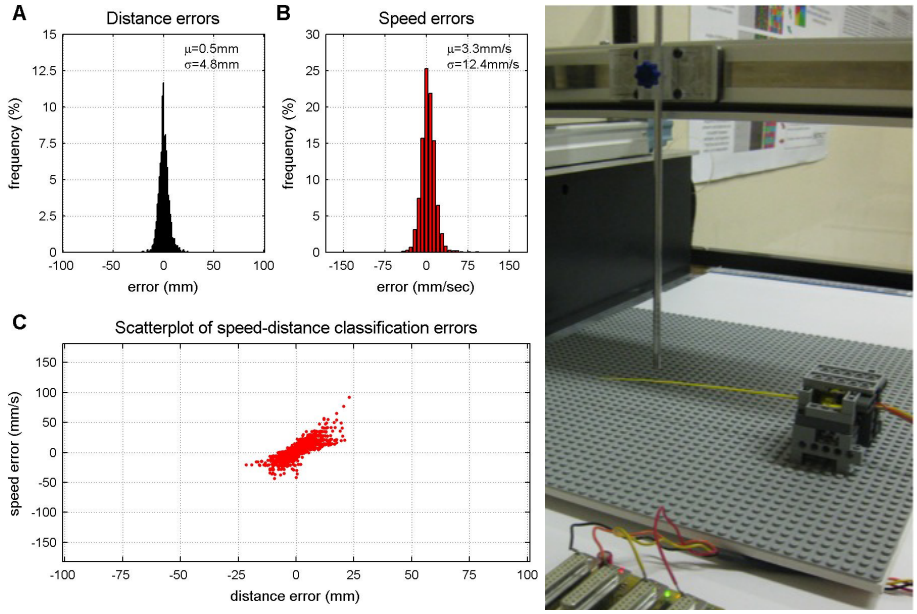


Fig. 2. Speed and radial distance classification with the XY table.
Panels A and B show the contact speed errors and radial distance errors between the target value for the validation trial and the classified value from the stationary naive Bayes classifier. Panel C shows a scatter plot of these errors.

retracted the bar at its maximum speed after a short 300ms delay. Thus the contacts were actively feedback-controlled, as are whisker contacts observed in animal behaviour studies. Radial contact distance along the whisker shaft away from the base was sampled in 1mm steps from 80mm to 180mm and contact speed in 7.2mm/sec steps from 36mm/sec to 216mm/sec, giving $101 \times 26 = 2626$ distinct contacts. For more details we refer to [5].

Results: Whisker signals had a characteristic profile in which their X-deflection increased smoothly from rest until reaching their peak after a few hundred milliseconds, and then quickly returned towards baseline followed by damped oscillatory ringing of period ∼50ms [5, Fig. 3]. As observed in the original analysis, deflections of a similar peak amplitude can be produced for many contact speed and distance combinations (by compensating the increase in speed with a decrease in radial distance).

Traces from 100ms before the initial deflection to 650ms after were used for analysis. Empirically, it was found that whisker velocity traces gave a better localization of contact speed and distance than whisker deflection traces. Thus, the positional deflections were converted to velocities for the analysis (by taking their numerical derivative and Gaussian smoothing to reduce noise). For each contact combination, three trials were used for training and the remaining trial saved for validation. These three trials were concatenated and the likelihoods

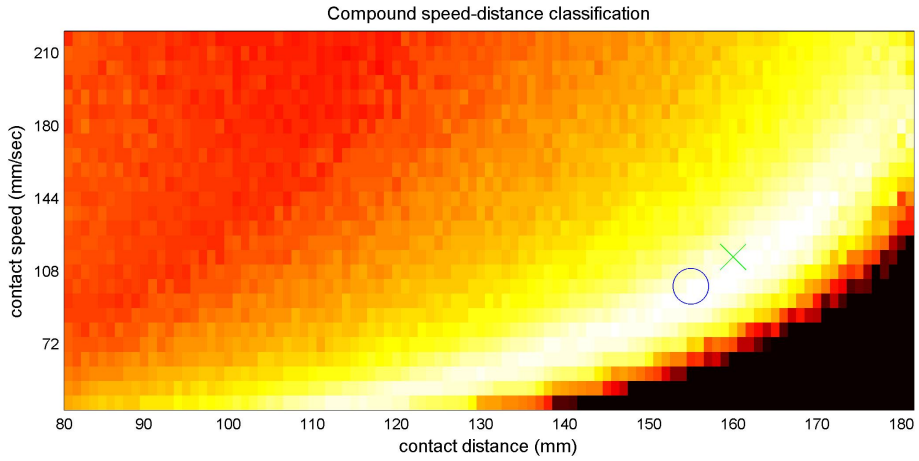


Fig. 3. Speed and radial distance classification with the XY table.
The brightness of each pixel represents the magnitude of the log-posterior for each combination of speed and distance marked on the axes. The green cross represents the target speed and distance of the test data and the blue circle the classified values according to the maximal log-posterior (at the brightest pixel).

found from the histograms of the empirical frequencies for the time series values binned into 10mV/sec intervals (Eq. 1). This procedure resulted in 2626 distinct likelihood distributions, one for each speed-distance combination.

For validation, each likelihood distribution was used to construct a log-posterior from the test trial data (Eq. 5). Generally, the distribution of log-posteriors for each trial had an extended curving segment of high values close to the target speed and distance of the test trial, surrounded by a drop-off of values further away from these targets (example shown in Fig. 3). The maximum log-posterior gave the classified speed and radial distance for that test trial.

Histograms of the errors between the classified and target speeds and distances were approximately Gaussian with little systematic bias of their centers from zero error (Figs 2A,B). A two-dimensional scatter-plot of these errors was consistent with these observations, and also revealed correlations between the speed and distance errors (Fig. 2C) consistent with the prevailing direction of the high value region in the log-posterior distribution (*c.f.* Fig. 3, bright region). This direction in contact speed and radial distance coincided with the region of constant peak amplitude examined in the original study of feature-based classifiers [5]. Hence, the stationary naive Bayes classifier appears highly dependent on peak deflection amplitude, but must also use other aspects of the whisker signal to localize the contact speed and distance close to the target values. Overall, the spread of distance errors was around 5mm compared with an overall range of 100mm, and the spread of speed errors was around 12mm/sec compared with an overall range of 180mm/sec. Thus the overall accuracy was around 93-95%.

5 Experiment 3: Shape and position classification

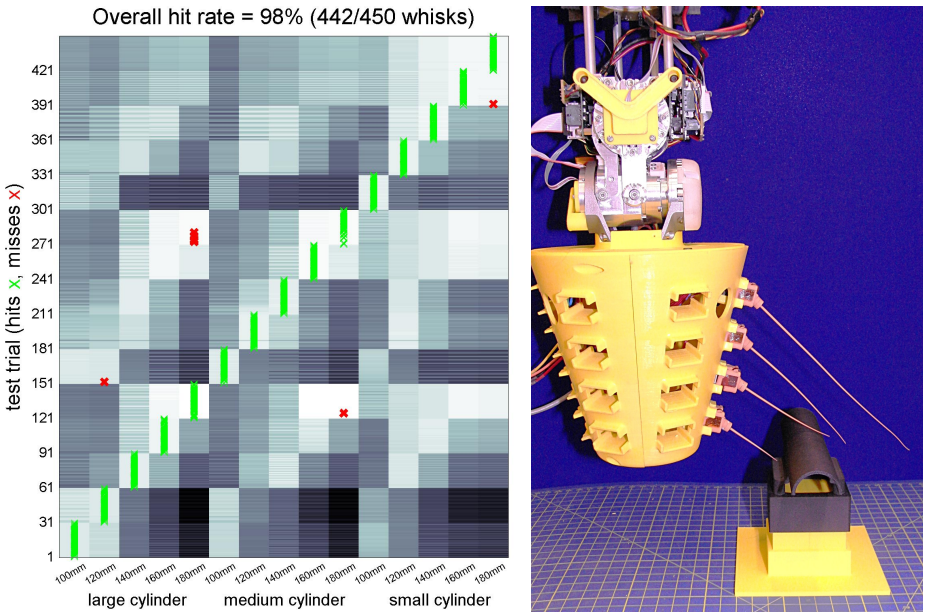


Fig. 4. Shape and position classification with the BIOTACT sensor.
The x-axis represents the cylinder configuration that is being tested for (5 positions and 3 curvatures). The y-axis denotes the test whisk, which were taken in 30 whisks trials in order of the configurations shown on the x-axis. Green crosses represent test trials that were correctly classified and red crosses represent incorrect classifications. The shade of pixels represents the value of the log-posterior according to the test cylinder configurations, such that brighter pixels denote larger log-posteriors (with the classified configuration being the brightest for each test trial).

Methods: The BIOTACT sensor (Fig. 4) was used to sense cylinders of various diameters placed horizontally at different positions along a platform and approached from a vertical direction.

The G1 (Generation 1) BIOTACT sensor consists of a truncated conical head that holds up to 24 whisker modules. The whisker modules are arranged in 6 radially symmetric rows of 4, oriented normally to the cone surface [16]. The head is mounted as the end-effector of a 7-degree-of-freedom robot arm (Fig. 4). For the present experiments, the head was fitted with a total of 4 whiskers in one row appropriate for sensing axially symmetric shapes such as cylinders aligned perpendicular to the whisker row. These whiskers were moved back-and-forth to repeatedly contact surfaces of interest akin to animal whisking behaviour [4], with individual whisker modules feedback modulated to make light, minimal impingement contacts on the surface [16]. Whiskers towards the front of the

head were shorter than at the back (lengths 50mm, 80mm, 115mm and 160mm), and were designed with a taper from a 1.5mm base to a 0.25mm tip.

Three rigid plastic hemi-cylinders with radii of curvature 35mm, 25mm and 15mm were used as test objects. They were mounted with their curved surfaces lying upwards on bases of height 15mm, 25mm and 35mm to ensure their vertical dimensions were equal. Five different positions for each cylinder were used, with central axes offset 100mm, 120mm, 140mm, 160mm and 180mm from the central axis of the head cone of the BIOTACT sensor. The orientation of the BIOTACT head was such that the four whiskers contacted the cylinders along the horizontal and perpendicular to the cylinder axis (*i.e.* across the cylinder). When contacting a horizontal surface, the whisker tips are approximately 30mm apart, spanning a distance from 100mm to 190mm perpendicular to the head axis. The depth of the contacts was arranged to be equal for all trials and only contacting the curved sections of the test objects. Each of the 15 distinct trials was repeated twice with 30 whisks at 1Hz to give training and validation data sets.

Results: The four whisker contacts across the cylinder surfaces had a distinctive pattern depending upon the curvature and position of the cylinder. For the largest cylinder in a central position, four whiskers could contact simultaneously, while for the smallest cylinder only one or two whiskers could make contact.

Single whisk traces from all 4 whiskers were used for analysis, with each trace starting and finishing when the whisker was maximally retracted from the object. The 30 whisks from each training trial were concatenated and the likelihoods found from the histograms of the empirical frequencies for the time series values binned into 10mV/sec intervals (Eq. 1). This procedure resulted in 15 likelihood distributions, one for each of the 3 cylinder curvatures and 5 positions.

For validation, the classification accuracy over single test whisks were considered. Over the 15 test trials, this gave a total of 450 distinct test whisks. The stationary naive Bayes classifier considered the log-posterior values for the 15 candidate configurations of 3 curvatures and 5 positions, calculated by summing the associated log-likelihood values for the whisker deflections over each single test whisk (Eq. 5). Correct classifications, or hits, correspond to when the maximum log-posterior coincides with the test cylinder configuration.

Overall, the hit rate for this experiment was above 98% using single test whisks contacting a surface (Fig. 4; hits shown in green and misses in red). The most common mistake was confusing the largest and medium cylinders at their most distant position from the robot head, where only one or two whiskers could contact the object; even then, though, there were just 6 mistakes over 60 test whisks. Therefore, using four whiskers dabbing onto a surface, the BIOTACT sensor could reliably determine both the curvature and horizontal positioning of test cylinders. This was achievable with single whisks lasting less than a second. If instead the full trial of 30 whisks were considered, then all test configurations could be determined with a perfect classification rate of 100%.

6 Discussion

The main point of this article is that a classifier based on probabilistic methods can be a general classifier for artificial whisker sensors because it determines the salient aspects of the data on which to base the discrimination. Such a classifier contrasts with the more traditionally considered specific classifiers that are based on pre-chosen features of the signals, such as peak deflection or profile of the power spectrum. We see these two types of classifiers as complementary in their applications and use. Specific classifiers are appropriate for controlled environments in which many details are known in advance, whereas general classifiers should just 'work out of the box' and apply across a range of situations.

For this initial study, we used the simplest Bayesian classifier available for time series analysis, by assuming stationarity and naive independence over sensor measurements. Further work could investigate how relaxing these assumptions affects classification accuracy. That being said, all applications of stationary naive Bayes considered in this article gave highly accurate hit rates of 90% or greater over diverse stimuli and robots. Furthermore, the efficiency savings from computationally simple classifiers, compared with more sophisticated methods for Bayesian inference, are important for sensing in autonomous robots. Hence we expect that the classifier used here will be appropriate for many situations of practical interest for whiskered robots. That being said, future study should clarify when the classification is limited by the simplifying assumptions, for example when correlations in the time series degrade the performance.

One problem requiring classifiers with the general performance found here is to autonomously explore and learn about novel environments. Because novel stimuli are not known in advance, only general classifiers would be sufficiently versatile for each new situation that arises. Fortunately, the probabilistic classifier in this article, stationary naive Bayes, can also detect novelty [17]. In another study with a BIOTACT robot, SCRATCHbot, novelty detection was examined by training the classifier to recognize the whisker signals from a blank floor and then examining the classifier output when it passed a novel texture. The classifier output, corresponding to the log-probability of the data being from the floor, displayed an anomalous change that can be used to identify novel events. One future project would be to combine the novelty detecting properties of the stationary naive Bayes classifier with its classification abilities to investigate the autonomous learning of stimuli in unfamiliar environments.

A related question for future study is how to extend the standard training/testing protocol of presenting many stereotyped examples of a contact onto a stimulus and then testing with the same examples. In natural environments, the robot might be expected to be more varied in its movements while contacting an object. Or perhaps the motion of the robot would need to become stereotyped to achieve the most reliable classification? These questions relate to a central theme of BIOTACT on active sensing, in which the appropriate sensation for the task is actively selected by adapting the sensor in response to information received about the environment, and will be a focus of future work.

Acknowledgments. The authors thank members of ATLAS (Active Touch Laboratory at Sheffield), the Bristol Robotics Laboratory, and the BIOTACT (BIOmimetic Technology for vibrissal ACtive Touch) consortium. This work was supported by EU Framework projects BIOTACT (ICT-215910), ICEA (IST-027819) and CSN (ICT-248986).

References

1. Y. Bar-Cohen. Biomimetics - using nature to inspire human innovation. *Bioinspiration & Biomimetics*, 1:P1, 2006.
2. T.J. Prescott, M.J. Pearson, B. Mitchinson, J.C.W. Sullivan, and A.G. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.
3. BIOTACT consortium. `http://www.biotact.org`.
4. T.J. Prescott. Vibrissal behavior and function. *Scholarpedia* (In press).
5. M. Evans, C.W. Fox, and T.J. Prescott. Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. *From Animals to Animats SAB 2010.*
6. C.W. Fox, B. Mitchinson, M.J. Pearson, A.G. Pipe, and T.J. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 26(4):223–239, 2009.
7. M. Evans, C.W. Fox, M.J. Pearson, and T.J. Prescott. Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task. *Proceedings TAROS 2009*, pages 19–24.
8. J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M.E. Diamond, and P. Konig. Texture signals in whisker vibrations. *Journal of neurophysiology*, 95(3):1792, 2006.
9. D.E. Kim and R. Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55(3):229–243, 2007.
10. M. Fend, S. Bovet, H. Yokoi, and R. Pfeifer. An active artificial whisker array for texture discrimination. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots and Systems IROS 2003*, volume 2.
11. M. Fend. Whisker-based texture discrimination on a mobile robot. *Advances in Artificial Life*, pages 302–311, 2005.
12. N.F. Lepora, M. Evans, C.W. Fox, M.E. Diamond, K. Gurney, and T.J. Prescott. Naive Bayes texture classification applied to whisker data from a moving robot. *Proc. IEEE World Congress on Comp. Int. WCCI 2010.*
13. J.H. Solomon and M.J. Hartmann. Biomechanics: Robotic whiskers used to sense features. *Nature*, 443(7111):525, 2006.
14. J.H. Solomon and M.J.Z. Hartmann. Artificial whiskers suitable for array implementation: Accounting for lateral slip and surface friction. *IEEE Transactions on Robotics*, 24(5):1157–1167, 2008.
15. M.J. Pearson, I. Gilhespy, C. Melhuish, B. Mitchinson, M. Nibouche, A.G. Pipe, and T.J. Prescott. A biomimetic haptic sensor. *International Journal of Advanced Robotic Systems*, 2(4):335–343, 2005.
16. J.C. Sullivan, B. Mitchinson, M.J. Pearson, M. Evans, N.F. Lepora, C.W. Fox, C. Melhuish, and T.J. Prescott. Tactile Discrimination using Active Whisker Sensors. *IEEE Sensors*, 99:1, 2011.
17. N.F. Lepora, M.J. Pearson, B. Mitchinson, M. Evans, C. Fox, A. Pipe, K. Gurney, and T.J. Prescott. Naive Bayes novelty detection for a moving robot with whiskers.

CrunchBot: a mobile whiskered robot platform

Charles W. Fox, Mathew H. Evans, Nathan F. Lepora,
Martin Pearson, Andy Ham and Tony J. Prescott

Active Touch Laboratory at Sheffield (ATL@S)
University of Sheffield, UK

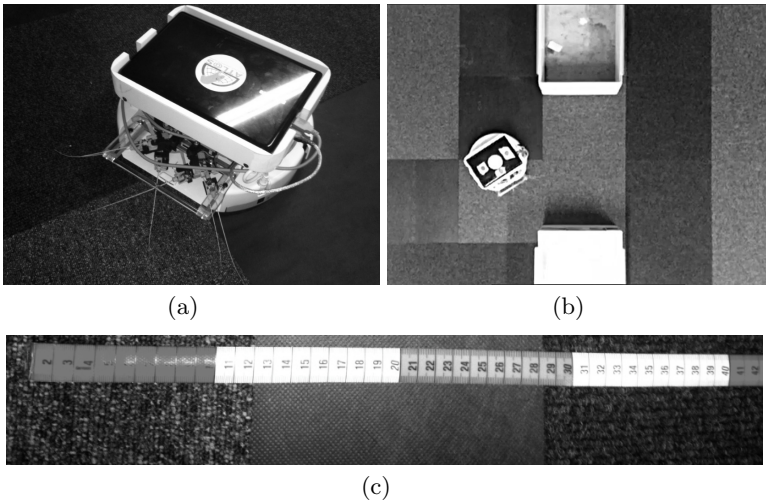
Abstract. CrunchBot is a robot platform for developing models of tactile perception and navigation. We present the architecture of CrunchBot, and show why tactile navigation is difficult. We give novel real-time performance results from components of a tactile navigation system and a description of how they may be integrated at a systems level. Components include floor surface classification, radial distance estimation and navigation. We show how tactile-only navigation differs fundamentally from navigation tasks using vision or laser sensors, in that the assumptions about the data preclude standard algorithms (such as extended Kalman Filters) and require brute-force methods.

1 Introduction

Touch-based navigation has two principal applications. Firstly, as a sole sensory system in environments where other types of sensors fail, such as smoky or dusty search-and-rescue sites, especially where covert (no signal emission) operation is required. Secondly, as a complement to other sensors such as vision, with which it can be fused or used as a 'last resort' during adverse conditions as in the sole sensor case.

This paper presents a new robot platform, CrunchBot (Fig. 1(a)) which draws together several strands of research towards systems for autonomous whisker-based tactile navigation. We and other authors have previously investigated individual components of such a system in isolation, including whiskered texture recognition [4],[11],[2], [5],[13], surface shape recognition [12],[9],[8],[3], and object recognition [7]. These components are often tested under ideal laboratory conditions or in individual mobile settings [16]. Here we present initial steps towards integrating them into a single platform, along with new results and observations on their performance 'in the wild' in a common arena environment. This integration report serves as a case study for other researchers wishing to work with similar platforms, listing the particular technologies used, the tools that link them together and providing new insights into the performance of such integrated systems.

Fig. 2 gives an overview of our general framework for perception and navigation with whiskers, which the present study works towards. When based on biological whiskers, whisker sensors have strain sensors at their base only. When



(a) (b)

(c)

Fig. 1. (a) Crunchbot. (b) Overhead view of Crunchbot in the arena environment. Different carpet tile textures can be seen on the floor along with square obstacles. (c) Textured floor tiles used in the navigation task. From left to right; smooth carpet, vinyl and rough carpet.

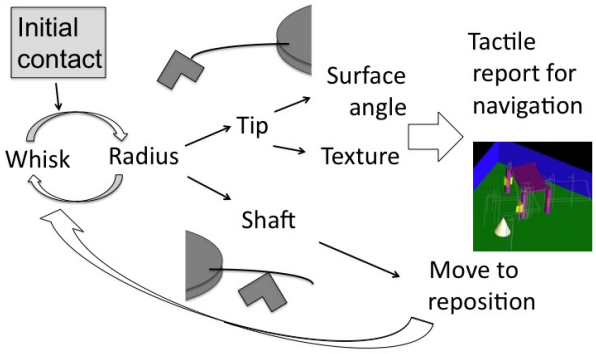


Fig. 2. Framework overview for whisker-based perception and navigation

a rat investigates an object it palpates the surface in a behaviour known as whisking [15],[1],[10]. It is thought that whisking is important for gathering the most reliable signals from whisker contacts. Whiskers can make two distinct types of contact with an object, contacting either at the tip or the shaft. Tip contacts are generally the most useful, because they provide a standardised, constrained setting (i.e. with the contact point at a known location at precisely the end of the whisker) from which surface properties such as orientation and texture can be identified [12],[4]. In contrast, shaft contacts are less informative; for exam-

ple, the radial distance of an object can confuse attempts to classify surface orientation and texture [5]. Shaft contacts are rare in practice, occurring only when small objects enter the field of multi-whisker arrays between the whisker tip points. In the scheme used here, a radial distance estimator [3] is first used to make a decision of whether the contact is at the tip or the shaft. If it is a shaft contact, then the robot should use the radial distance information to move to another location that is likely to yield a more useful tip contact. Following a tip contact, we can read surface angle and texture information (and possibly speed of object when there are moving objects in the world) and pass them as an observation to a navigation or mapping system.

The remainder of this paper outlines an initial implementation of each of these stages, which for the first time are integrated into a complete system. The present navigation method is driven by texture and distance observations only. An intermediate object recognition step could also be performed, as described in a companion paper [7].

2 Methods

2.1 Whiskers

CrunchBot's six whiskers measure 160mm in length, 1.45mm diameter at the base tapering linearly to 0.3mm at the tip. They are built from nanocure25 using an Evisiontec rapid prototyping machine. A magnet is bonded to the base of the whisker and held in place by a plug of polyurethane approximately 0.75 mm above a Melexix 90333 tri-axis Hall effect sensor IC [14]. This sensor generates two outputs representing the direction of the magnetic field (in two axes) with respect to its calibrated resting angle. These two 16-bit values are sampled by a local dsPIC33f802 micro-controller which, in turn, is collected using an FPGA configured as a bridge to a USB 2.0 interface. Up to 28 whiskers can be connected to this FPGA bridge at one time. Using the vendor provided software driver and API (Cesys GmbH), a user can request the data from all whiskers at minimum intervals of 500μs (a maximum sample rate of 2kHz).

2.2 Robot platform

The whiskers are mounted in the cargo bay of an iRobot Create base (www.irobot.com), being positioned on an adjustable metal bar and rapid prototyped ball joint mountings. These mountings allow adjustment of the whiskers, which is particularly important for obtaining good floor contacts. We have also extended the cargo bay mounting to accommodate a netbook PC, which is used for local control of the robot. The netbook runs Ubuntu 10.10 on a single-core Intel Atom processor. A circular buffer in shared memory is used to make data from the Cesys driver available to other processes. The netbook hosts a Player server (playerstage.sourceforge.net) which provides high-level, networked API interfacing to the Create's serial port commands. Processes such as texture and shape

recognition and basic motor control run on the netbook, reading the raw data from the fast circular shared memory buffer, and (usually, see Section 2.3 below) writing their results every 0.1s to a Python Pyro server (pyro.sourceforge.net) on the remote desktop. Differential and absolute odometry data from the Create is also sent to this server. Preliminary experiments in our lab show that the odometry of the Create, once loaded with the sensing and control hardware, is accurate to $< 5\%$ of any straight line or turn on the spot movements.

2.3 Navigation task

Fig. 1(b) shows the arena environment used in our tests. The arena is a 2.5m×2.5m square, surrounded by walls and paved with twenty five 0.5m×0.5m tiles. There are three types of tiles with different textures: vinyl, smooth carpet and rough carpet (see Fig. 1(c)). A few 0.5m×0.5m square obstacles are also placed over some carpet tiles. The current system implements a random exploratory behaviour of the arena, controlled by the netbook.

Previous work has shown that accurate object localisation with a whisker requires some measure of contact speed [3], or of the applied forces and bending moments at the base of the whisker [9],[12], values that are not always available in the mobile case as agent movement will affect these contacat properties. To address these points a 'body whisk' behaviour was included in the robot program. As the whiskers were not actuated the whole robot must rotate in a systematic way. Upon initial contact with an object the robot first reverses away a short distance before rotating at $\pi/24$ radians per second towards the object, then rotating at $\pi/24$ radians per second away from the object. This allows this whiskers to move over the surface of the contact object, collecting data about its shape. After the whisk the robot reverses again to clear the object, then rotates in a random direction and moves forward again.

The ultimate task is to infer the robot's location from noisy odometry and the whiskers. There is some subtlety in how odometry is reported. Inference running on the remote desktop may become computationally intensive, and its update cycle can fall behind the rate of reporting from the netbook. Markovian inference algorithms can simply discard any unused sensory observations, requiring only the latest observation, but the odometry must be *integrated* so that each inference step receives the total odometry occurring since the previous inference step. Thus multiple read outs may be required from the robot sensory systems, which were taken at 0.ls intervals.

2.4 Floor texture discrimination

The outer two of the six whiskers are angled downwards to make a light, brushing contact with the floor surface that CrunchBot is travelling over (Fig. 1). Classification software on the netbook then seeks to classify the whisker deflection signals into previously learnt classes (for example, vinyl or rough carpet), to infer which surface the robot is travelling over. In its current configuration, the signals from the two outer whiskers are classified individually, so that the

robot can determine whether its left and right sides are on the same or different textures.

A stationary naive Bayes algorithm is used to infer the surface texture from the whisker contacts (see also Lepora *et al* (2011) in this conference volume). This algorithm has been demonstrated to be a reliable classifier of surface texture from whisker contacts on mobile robot platforms [13]. The classifier uses stored texture log-likelihoods $\log P(x|\mathrm{T})$ that represent the log-probability distributions of contact deflections in training data from each possible texture. The overall log likelihood over a temporal window of whisker deflection data is then obtained by integrating these the log-likelihoods over the individual samples, assuming naive sample independence and stationarity over time,

$$\log P(x_{n_s}, \cdots, x_{n_f}|\mathrm{T}_l) = \sum_{i=n_s}^{n_f} \log P(x_i|\mathrm{T}_l). \tag{1}$$

In previous studies of this classifier [13], these likelihoods were fed into Bayes rule to give the posterior probabilities of the probabilities for each texture having generated the window of data. Instead, the present method feeds the likelihoods directly into the navigation system for the robot, to be used to infer navigation information.

Another principal difference from our previous applications of stationary naive Bayes, is that the present implementation requires that the classification be done in real-time on board the robot. Moreover, the texture that the robot is sensing can change during the motion, as the robot moves from one tile to the next. For these reasons, only the most recent 100ms (200 sample) window of texture data was classified, to give good classification reliability while minimising possible boundary crossing. Fortunately, the algorithmic complexity of the classifier is low, so that classification times were much less than the inter-classification intervals of 100ms.

2.5 Object localisation

To determine whether an object has made contact with the a whisker at the tip or the shaft, and to discriminate between contacts with the surfaces or corners of objects, object localisation was implemented. Previous work [3] has shown that peak deflection magnitude could be used as a feature for radial distance discrimination at a given speed. Whisker data was recorded during the 'body whisk' contact, and the maximum whisker deflection was measured. Deflection magnitude was taken as the Hall effect sensor output voltage at peak deflection, which is proportional to the bending moment. This feature f_1 can be defined as,

$$f_1 = \max_{\mathrm{t}} \theta(\mathrm{t}), \tag{2}$$

where $\theta(t)$ is the time-dependent deflection magnitude measured by the Hall effect sensor.

During the training phase a dataset was collected for each whisker, consisting of 5 contacts at each point along the whisker at 10mm intervals over a 50mm

range from the tip of the whisker. Though the whisker is 160mm long, only 140mm is external to the 'follicle'. A model was then generated of the relationship between the deflection magnitude and the corresponding radial distance to contact by fitting a linear equation to the training data in MATLAB. To find an estimate of radial distance r,

$$r = a_1 f_1 + a_0, \tag{3}$$

was fitted to the data with a linear-in-the-parameters regression on the line, giving a least-squares fit for (a_0, a_1) for each whisker.

2.6 Navigation

A key question to address in tactile navigation and mapping is to what extent methods developed for other sensory modalities, such as laser scanners, are applicable to tactile sensors. A naive view of tactile navigation holds that touch sensors are just like laser scanners but with a very short range, and therefore standard approaches could be used.

Standard laser-driven methods include Extended Kalman Filtering (EKF) and Iterative Closest Point matching. In ICP, large point clouds are aligned between steps to compute odometry estimates. In EKF, unique or rare features are extracted from point clouds, and used as discrete landmarks. In both of these methods, the pose likelihood function varies smoothly over poses, because the lasers have a long range, and their values change gradually with pose. In particular, the assumption of this smoothness allows the EKF to approximate the likelihood function by a Gaussian, with parameters given by the local curvature around its prior mean. It is unclear in the tactile case whether the prior mean is closely related to the posterior mean, as the relationship depends on the likelihood smoothness.

Foveal grid. When these assumptions are violated, brute force computation is required, such as grid based or particle filtering methods. To examine what approach best matches the data, we have implemented a simple hybrid grid and Gaussian likelihood model. By inspecting the shape of the likelihood functions in the grids we will see if smoothness and Gaussian assumptions are appropriate. To enable inference to run in real-time, we restrict the grid to a 'foveal' region centred on the robot's prior mode location, and oriented in egocentric coordinates. We denote likelihoods as $P(s|x, y, \theta)$ where (x, y, θ) each range over 11 discrete values in the grid, and s is the current sensory report. (The grid is 0.25m in diameter, with θ from $\pi/4$ to $+\pi/4$.)

Filtering. A slow method to filter the belief at each step, given sensory reports $s_{1:t}$ over time t would be to work entirely in the grid representation,

$$P(X_t|s_{1:t}) = P(s_t|X_t) \sum_{X_{t-1}} P(X_{t-1}|s_{1:t-1}) T(X_t|X_{t-1}, o_t) \tag{4}$$

where we write $X_t = (x_t, y_t, \theta_t)$, and T is a transition function describing motion probabilities as a function of observed odometry data o_t, and the sums range

over all possible states of the grids. This method is slow because it performs 3-dimensional convolutions. We speed up filtering by approximating the posterior and transitions with Gaussians, $N(\mu_t, \Sigma_t)$ and $N(\mu_{o_t}, \Sigma_{o_t})$, at each step as follows. First we compute the grid values,

$$P(X_t|s_{1:t}) = P(s_t|X_t)(N(\mu_{t-1}, \Sigma_{t-1}) * N(\mu_{o,t}, \Sigma_{o,t})) \tag{5}$$

for each discrete X_t (where $(o,t) = o_t$ to reduce subscripts). Here the computation only ranges over the present 3D grid, rather than the convolution over two such grids. The simpler convolution of Gaussians, $*$, is quickly computed analytically. We then fit a Gaussian to the resulting set of posterior grid points, $\{X_t\}$,

$$P(X_t|s_{1:t}) \approx N(\mu_t = \hat{X}_t, \Sigma_t = cov_{\hat{X}_t}(\{X_t\})) \tag{6}$$

by picking the mode and computing the covariance about the mode with the sum

$$\mathrm{cov}_{\hat{X}_t}(\{X_t\}) = \sum_{X_t} (X_t - \hat{X}_t)^T (X_t - \hat{X}_t) \tag{7}$$

This foveal filtering method is illustrated in Fig. 3. Here the agent has moved from a to b and has detected an object with its right whisker. Its previous posterior Gaussian, centred on point c, is used together with the Gaussian transition T to produce a prior Gaussian about point d. This Gaussian is discretised and fused with likelihoods at each grid point to give the posterior shown in the enlargement e. We would then fit a Gaussian to this posterior grid. (This method is similar to the EKF and Unscented Kalman Filters in using Gaussian posteriors, but differs by using a brute-force likelihood. Recall that we wish to examine these likelihoods to see how valid the approximations of EKF and UKF would be.)

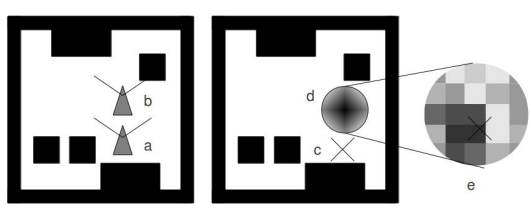


Fig. 3. Foveal grid based navigation.

Transition function. Exact transition functions for 2D mobile robots are 'banana shaped' under Gaussian uncertainty about the starting angle, because the direction of travel is uncertain. We use a heuristic Gaussian approximation to this shape, which preserves the mode and tries to approximately minimise the

KL divergence from the true distribution. It is also necessary to add a small additional component to each posterior covariance to compensate for quantisation error in the grid and the lack of computation about locations beyond the foveal grid.

Likelihood function. In the likelihood function, $s_t = (\tau_{i=1:2}, r_{i=1:4})_t$ is comprised of information from the two floor sweeping whiskers returning texture classes $\tau_{i=1:2} \in \{\text{VINYL}, \text{CARPET}, \text{ROUGH}\}$ and the four object localisation whiskers returning radial distances to contact, $r_{i=1:4}$ ranging from 1mm to 140mm at contact or null ($\emptyset$) for no contact. We assume these observations are conditionally independent given location and that the radial distance errors are Gaussian. We use the noise levels found from empirical data. We speed up the likelihoods by computing them offline for each grid square in the whole arena and saving them in a hash table. (This requires coordinates in the egocentric grid to be transformed to world-centred coordinates to perform the look-up).

3 Results

3.1 Floor texture discrimination

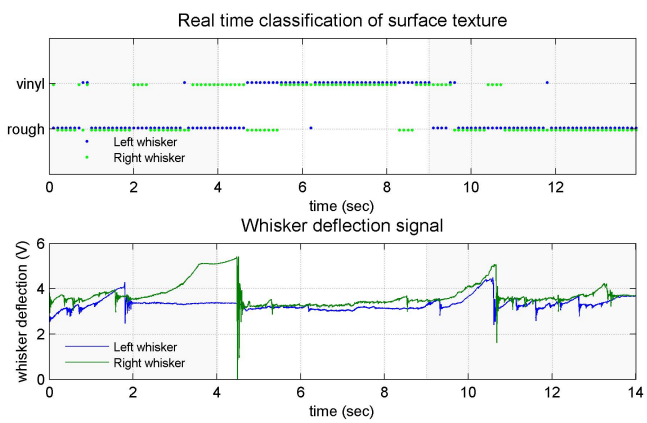


Fig. 4. Real-time classification of surface texture.
The top panel shows the classification results for the left (blue) and right (green) whiskers traversing a course over rough carpet (shaded regions), then vinyl (unshaded region) and back to rough carpet. The bottom panel shows the deflections associated with each whisker.

The robot's performance at real-time texture classification was assessed by having the robot follow a course in the arena where it traverses different textures. For an initial investigation, only two surface textures were considered, corresponding to a smooth vinyl and rough carpet. The classification algorithms

were trained by presenting the robot with just a single texture, from which its central controller determined the corresponding texture likelihoods for that surface. These were then stored for use in general classification.

A typical example of the classification performance is shown in Fig. 4. The course consisted of a tile of rough carpet, then a tile of smooth vinyl, and finally a tile of rough carpet. The approximate times when the robot was traversing each texture are marked on the figure, with rough carpet shaded and smooth vinyl unshaded. As is visible from the figure, both whiskers reliably reported the correct texture.

One general feature that we observed is that the texture classifier can produce sporadic results near texture boundaries, with the whiskers disagreeing about which texture is being encountered (Fig. 4 top panel, borders of shaded regions). Examining the whisker deflection traces, reveals that large deflections of the whisker can occur in these regions that last over a second. We diagnosed this issue as due to the whisker catching on the boundary between two tiles of different heights, and so this feature is actually a signature of a change in texture. In general, catching the whisker on the floor surface caused problems for the classification, because these are infrequent events that can last several hundred milliseconds or more. Hence, we sought to position the whisker to angle as far back as possible to minimise such events, which emphasises that the way in which the whisker contacts the floor can be important for how reliable a classifier can perform.

3.2 Object localisation

Peak deflection magnitude for each contact is shown in Fig. 5. Standard deviation of error for radial distance estimation is shown in the table below.

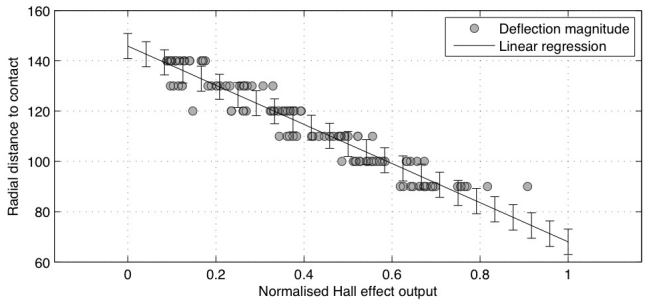


Fig. 5. Peak deflection magnitude for contacts along the shaft of the whisker. Standard error for the regression is 4.98mm

	Whisker 1	Whisker 2	Whisker 3	Whisker 4	Combined
Std error	5.68mm	2.78mm	1.82mm	4.37mm	4.98mm

Standard classification error is very low, typically less than 5mm over the 60mm range tested. For some whiskers classification error is even lower, below 2mm. These results compare favourably with previous work under highly controlled conditions where speed was variable. This indicates that the noise in the odometry is low enough to ensure a consistent contact force and speed.

3.3 Navigation

Likelihood function forms. Fig. 6(a) shows typical examples of forms of the likelihood function, from six poses in the arena. (Dimension stacking is used to display each 3D likelihood cube as a row of 2D slices.) It can be seen that these functions are highly non-Gaussian and non-smooth, consisting mostly of multiple straight edges and corners arising from the form of the floor and object layouts in the arena. They include many abrupt discontinuities, because the whiskers only sense a small local area, and neighbouring locations can have completely different textures or object contacts. This is unlike the case of vision or lasers whose likelihoods are locally smooth, and suggests that brute force methods such as the foveal grid (or particle filters) are more appropriate to tactile navigation than methods which assume that the likelihood is Gaussian or otherwise smooth.

Posterior fusion. Fig. 6(b) shows two typical examples of fusing priors (which are always Gaussian) and likelihood grids, when the prior mode is a short distance away from the ground truth, but still within the fovea radius. In the first example, the likelihood function is extremely focused – because only two points are compatible with the sensors – so it dominates the posterior to correct the belief. The second example shows a multi-peaked likelihood function knocking out the region around the prior mode, and producing a multi-peaked posterior. These two typical examples show that using the posterior mode as the next prior mean is an appropriate method, rather than using the posterior mean, or linearising the posterior about the old prior mean as in the EKF. They illustrate the advantage of fitting the Gaussian to the posterior grid, rather than Gaussianising the likelihood only and fusing it analytically with the prior Gaussian (which we attempted in earlier implementations with little success.)

Odometry. In simulations with low sensor noise, we found that computing the 'banana' approximation to odometry noise under uncertain pose angle noise models was unnecessary in practice, because the use of the foveal grid restricts the posterior Gaussian to the size of the fovea. As long as the likelihood functions are strong enough to move the mode around when fused with the prior, we found little difference in behaviour between using the full approximation and simply inflating every posterior covariance to equal the size of the fovea. The covariance cannot increase beyond this size because locations outside the fovea are never considered; while the full covariance could reduce in size on informative observations, such observations are rare in our tactile task, and also have little effect on the mode positions.

Full system integration. Fig. 6(c) shows a very early screenshot from the first test we have performed with the full system. It is the simplest possible test, moving over a single texture boundary and correcting the posterior belief.

4 Discussion

We have demonstrated an initial implementation of our framework for whiskered perception and navigation, showing how real-time signal processing, texture classification, distance estimation and navigation can be combined on an inexpensive mobile platform.

It is important to consider interactions between components when integrating systems. For example, we found it useful to disable both odometry reports and texture classification while performing mini-whisks. As noted in previous studies, contact location confounds surface property discrimination [5] and we are currently working to implement active sensing behaviours which position the robot so as to obtain standardised contact types to aid perception. Placing action at the heart of the perceptual process in this way is only possible with integrated mobile systems, rather than the statically mounted whiskers that have been used in previous single component laboratory tests [9],[17].

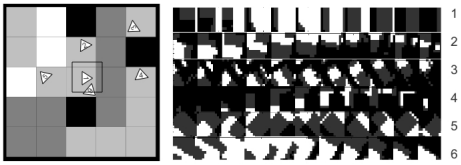
Although introduced purely for computational reasons, both the use of likelihood hash table and the mode-centred Gaussian posterior approximations are similar to, and loosely inspired by, biological models of hippocampal navigation [6]. However use of a single mode-centered Gaussian has lead to problems in pilot runs in which the true posterior becomes strongly multimodal, for example when a focussed likelihood appears far away from the prior mode. Future systems may address this using small particle filters – even just two particles would help. A larger scale solution to navigation is to perform an object recognition step using sensory information, then use the objects as landmarks – such an object recognition system is described in a companion paper [7].

Acknowledgements This work was supported by the EPSRC Doctoral Training Scheme and EU Framework projects BIOTACT (ICT-215910), ICEA (IST-027819) and CSN (ICT-248986).

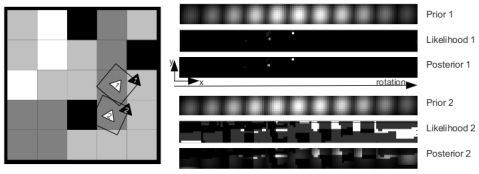
References

1. M. E. Diamond, M. von Heimendahl, P. M. Knutsen, D. Kleinfeld, and E. Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.
2. M. Evans, C. Fox, M. Pearson, and T. Prescott. Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In *Proc TAROS*, pages 19–24, 2009.
3. M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proc IEEE ROBIO*, 2010.
4. M. Fend. Whisker-based texture discrimination on a mobile robot. *Advances in Artificial Life*, pages 302–311, 2005.
5. C. Fox, B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 26(4):223–239, 2009.
6. C. Fox and T. Prescott. Hippocampus as unitary coherent particle filter. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2010.

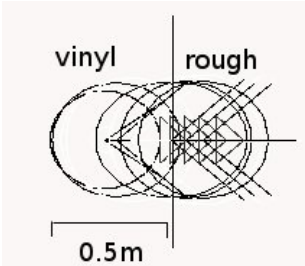
7. C. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. Submitted to TAROS 2011.
8. C. W. Fox, M. Evans, M. J. Pearson, and T. J. Prescott. Towards temporal inference for shape recognition from whiskers. In *Proc TAROS*, pages 226 – 233, 2008.
9. V. Gopal and M. J. Z. Hartmann. Using hardware models to quantify sensory data acquisition across the rat vibrissal array. *Bioinspir Biomim*, 2(4):S135–45, 2007.
10. R. A. Grant, B. Mitchinson, C. W. Fox, and T. J. Prescott. Active touch sensing in the rat: Anticipatory and regulatory control of whisker movements during surface exploration. *Journal of Neurophysiology*, 101:862–874, 2009.
11. J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. Diamond, and P. Konig. Texture signals in whisker vibrations. *Journal of neurophysiology*, 95(3):1792, 2006.
12. D. Kim and R. Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55(3):229–243, 2007.
13. N. Lepora, M. Evans, C. Fox, M. Diamond, K. Gurney, and T. Prescott. Naive Bayes texture classification applied to whisker data from a moving robot. *Proc. IEEE World Congress on Comp. Int. WCCI2010*, 2010.
14. Melexis. www.melexis.com/assets/mlx90333_datasheet_5276.aspx.
15. B. Mitchinson, C. J. Martin, R. A. Grant, and T. J. Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.
16. T. Prescott, M. Pearson, B. Mitchinson, J. Sullivan, and A. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.
17. R. Russell and J. Wijaya. Object location and recognition using whisker sensors. In *Australasian Conference on Robotics and Automation*. Citeseer, 2003.



(a)



(b)



(c)

Fig. 6. (a) Example of likelihoods (lighter=stronger) at six locations during an exploration. The white triangles on the map (left) show the fovea locations. At each step we show the likelihood function (right) as a row of 11×11 foveal location squares, over a third dimension (along the row) of orientation from $-\pi/4$ to $+\pi/4$. The black squares in the arena are solid objects; white, light and dark grey are VINYL, SMOOTH and ROUGH floor surfaces respectively. The square around pose (1) on the map shows the location and size of the fovea. Foveas are not shown at other locations for simplicity.(b) Two examples of non-smooth likelihoods fusing with Gaussian priors to give non-smooth posteriors. The location of the posterior mode can be highly dependent on the likelihood function, and approximating the likelihood with a Gaussian would change its location. (c) An early test of pose correction in the fully integrated system. (Screenshot from graphical display). The triangles show a sequence of five ground truth locations of the robot moving in a straight line, crossing the boundary from a vinyl tile to a rough tile. The circles show the covariance of the posterior beliefs for the same five steps. The belief is initialised to an incorrect location, 0.2m behing the ground truth. The output shows the beliefs jumping to be closer to the ground truths as the robot crosses the texture boundary.

Studio report: Linux audio for multi-speaker natural speech technology

Charles FOX, Heidi CHRISTENSEN and **Thomas HAIN**
Speech and Hearing
Department of Computer Science
University of Sheffield , UK

charles.fox@sheffield.ac.uk

Abstract

The Natural Speech Technology (NST) project is the UK's flagship research programme for speech recognition research in natural environments. NST is a collaboration between Edinburgh, Cambridge and Sheffield Universities; public sector institutions the BBC, NHS and GCHQ; and companies including Nuance, EADS, Cisco and Toshiba. In contrast to assumptions made by most current commercial speech recognisers, natural environments include situations such as multi-participant meetings, where participants may talk over one another, move around the meeting room, make non-speech vocalisations, and all in the presence of noises from office equipment and external sources such as traffic and people outside the room. To generate data for such cases, we have set up a meeting room / recording studio equipped to record 16 channels of audio from real-life meetings, as well as a large computing cluster for audio analysis. These systems run on free, Linux-based software and this paper gives details of their implementation as a case study for other users considering Linux audio for similar large projects.

Keywords

Studio report, case study, speech recognition, diarisation, multichannel

1 Introduction

The speech recognition community has evolved into a niche distinct from general computer audio and Linux audio in particular. It has its own large collection of tools, some of which have been developed continually for over 20 years such as the HTK Hidden Markov Model ToolKit [Young et al., 2006][1]. We believe there could be more crosstalk between the speech and Linux audio worlds, and to this end we present a report of our experiences in setting up a new Linux-based studio for dedicated natural speech research.

In contrast to assumptions made by current commercial speech recognisers such as Dragon Dictate, natural environments include situations such as multi-participant meetings [Hain et al., 2009], where participants may talk over one another, move around the meeting room, make non-sentence utterances, and all in the presence of noises from office equipment and external sources such as traffic and people outside the the room. The UK Natural Speech Technology project aims to explore these issues, and their applications to scenarios as diverse as automated TV programme subtitling; assistive technology for disabled and elderly health service users; automated business meeting transcription and retrieval, and homeland security.

The use of open source software is practically a prerequisite for exploratory research of this kind, as it is never known in advance which parts of existing systems will need to be opened up and edited in the course of research. The speech community generally works on offline statistical, large data-set based research. For example corpora of 1000 hours of audio are not uncommon and require the use of large compute clusters to process them. These clusters already run Linux and HTK, so it is natural to extend the use of Linux into the audio capture phase of research. As speech research progresses from clean to natural speech, and from offline to real-time processing, it is becoming more integrated with general sound processing [Wolfel and McDonough, 2009], for example developing tools to detect and classify sounds as precursors to recognition. The use of Bayesian techniques in particular emphasises the advantages of considering the sound process-

[1] Currently owned by Microsoft, source available *gratis* but not *libre*. Kaldi is a *libre* alternative currently under development, (kaldi.sourceforge.net).

ing and recognition as tightly coupled problems, and using tightly integrated computer systems. For example, it may be useful for Linux cluster machines running HTK in real-time to use high level language models to generate Bayesian prior beliefs for low-level sound processing occurring in Linux audio.

This paper provides a studio report of our initial experiences setting up a Linux based studio for NST research. Our studio is based on a typical meeting room, where participants give presentations and hold discussions. We hope that it will serve as a self-contained tutorial recipe for other speech researchers who are new to the Linux audio community (and have thus included detailed explanations of relatively simple Linux audio concepts). It also serves as an example of the audio requirements of the natural speech research community; and as a case study of a successful Linux audio deployment.

2 Research applications

The NST project aims to use a meeting room studio, networked home installations, and our analysis cluster to improve recognition rates in natural environments, with multiple, mobile speakers and noise sources. We give here some examples of algorithms relevant to natural speech, and their requirements for Linux audio.

Beamforming and ICA are microphone-array based techniques for separating sources of audio signals, such as extracting individual speakers from mixtures of multiple speakers and noise sources. ICA [Roberts and Everson, 2001] typically makes weak assumptions about the data, such as assuming that the sources are non Gaussian in order to find a mixing matrix M which minimises the Gaussian-ness over time t of the latent sources vector x_t, from the microphone array time series vectors y_t, in $y_t = Mx_t$.

ICA can be performed with as few microphones as there are sound sources, but gives improved results as the number of microphones increases. Beamforming [Trees, 2002] seeks a similar output, but can include stronger physical assumptions - for example known microphone and source locations. It then uses expected sound wave propagation and interference patterns to infer the source waves from the array data. Beamforming is a high-precision activity, requiring sample-synchronous accuracy between recorded channels, and often using up to 64 channels of simultaneous audio in microphone arrays (see for example the NIST Mark-III arrays [Brayda et al., 2005]).

Reverberation removal has been performed in various ways, using single and multi-channel data. In multi-channel settings, sample-synchronous audio is again used to find temporal correlations which can be used to separate the original sound from the echos. In the iPhone4 this is performed with two microphones but performance may increase with larger arrays [Watts, 2009].

Speaker tracking may use SLAM techniques from robotics, coupled with acoustic observation models, to infer positions of moving speakers in a room (eg. [Fox et al., 2012], [Christensen and Barker, 2010]). This can be used in conjunction with beamforming to attempt retrieval of individual speaker channels from natural meeting environments, and again relies on large microphone arrays and sample-accurate recording.

Part of the NST project called 'homeService' aims to provide a natural language interface to electrical and electronic devices, and digital services in people's homes. Users will mainly be disabled people with conditions affecting their ability to use more conventional means of access such as keyboard, computer mouse, remote control and power switches. The assistive technology (AT) domain presents many challenges; of particular consequence for NST research is the fact that users in need of AT typically have physical disabilities associated with motor control and such conditions (e.g. cerebral palsy) will also often affect the musculature surrounding the articulatory system resulting in slurred and less clear speech; known as *dysarthric* speech.

3 Meeting room studio setup

Our meeting room studio, shown in fig. 1, is used to collect natural speech and training data from real meetings. Is centred on a six-person table, with additional chairs around the walls for around a further 10 people. It has a whiteboard at the head of the table, and a presentation projector. Typical meetings involve participants speaking from their chairs but also getting up and walking around to present or to use the whiteboard. A 2×2.5m aluminium frame is suspended from the ceiling above the table and used for mounting au-

Figure 1: Meeting room recording setup. The boxes on the far wall are active-badge trackers. The frame on the ceiling and the black cylinder on the table each contain eight condenser microphones. Participants wear headsets and active badges.

dio equipment. Currently this consists of eight AKG C417/III vocal condenser microphones, arranged in an ellipse around the table perimeter. A further eight AKG C417/IIIs are embedded in a 100mm radius cylinder placed in the table centre to act similarly to eight-channel multi-directional tele-conferencing recorder. The table can also include three 7-channel DevAudio Microcones (www.dev-audio.com), which are commercial products performing a similar function. The Microcone is a 6-channel microphone array which comes with propriety drivers and an API. A further 7th audio channel contains a mix of the other 6 channels as well as voice activity detection and sound source localisation information annotation. Some noise reduction and speech enhancement capabilities are provided, although details of the exact processing are not made public.

There are four Sennheiser ew100 wireless headsets which may be mounted on selected participants to record their speech directly. The studio will soon also include a Radvision Scopia XT1000 videoconferencing system, comprised of a further source-tracking steered microphone and two 1.5m HD presentation screens. In the four upper corners of the meeting room are mounted Ubisense infrared active badge receivers, which may be used to track the 3D locations of 15 mobile badges worn by meeting participants. (The university also has a 24-channel surround sound diffusion system used in an MA Electroacoustic music course [Mooney, 2005], which may be useful for generating spatial audio test sets.)

Sixteen of the mics are currently routed through two MOTU 8Pre interfaces, which take eight XLR or line inputs each. Both currently run at 48kHz but can operate up to 96kHz. The first of these runs in A/D conversion mode and sends all 8 digitised channels via a single ADAT Lightpipe fibre optic cable to the second 8Pre. The second 8Pre receives this input, along with its own eight audio channels and outputs all 16 channels to the DAW by Firewire 400 (IEEE 1394, 400 bits/sec). (The two boxes must be configured to have (a) the same speed, (b) 1x ADAT protocol and (c) be in be in converter/interface mode respectively.) Further firewire devices will be added to the bus later to accommodate the rest of the microphones in the room.

4 Linux audio system review

Fig. 2 outlines the Linux audio system and highlights the parts of the many possible Linux audio stacks that are used for recording in the meeting room studio. The Linux audio architecture has grown quite complex in recent years, so is reviewed here in detail.

OSS (Open Sound System) was developed in the early 1990s, focused initially on Creative SoundBlaster cards then extending to others. It was a locking system allowing only one program at a time to access the sound card, and lacked support for features such as surround sound. It allowed low level access to the card, for example by `cataudio.wav > /dev/dsp0`. ALSA (Advanced Linux Sound Architecture) was designed to replace OSS, and is used on most current distributions including our Ubuntu Studio 11.10. PortAudio is an API with backends that abstract both OSS and ALSA, as well as sound systems of non-free platforms such as Win32 sound and Mac CoreAudio, created to allow portable audio programs to be written. Several software mixer systems were built to resolve the locking problem for

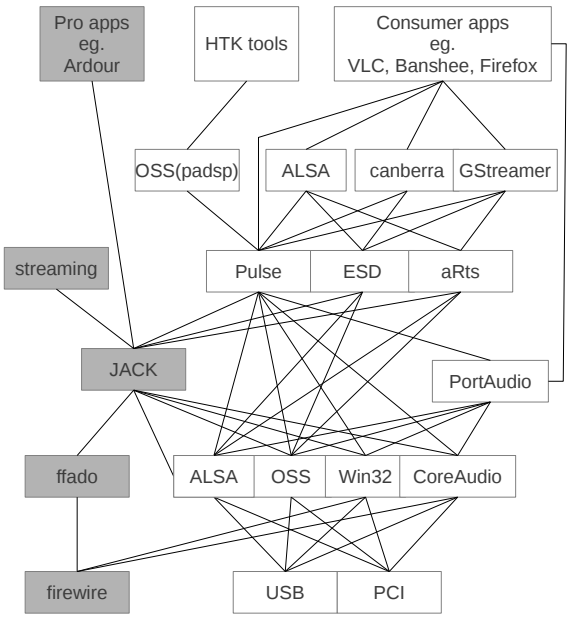


Figure 2: Audio system for recording.

consumer-audio applications, including PulseAudio, ESD and aRts. Some of these mixers grew to take advantage of and to control hardware mixing provided by sounds cards, and provided additional features such as network streaming. They provided their own APIs as well as emulation layers for older (or mixer-agnostic) OSS and ALSA applications. (To complicate matters further, recent versions of OSS4 and ALSA have now begun to provide their own software mixers, as well as emulation layers for each other.) Many current Linux distributions including Ubuntu 11.10 deploy PulseAudio running on ALSA, and also include an ALSA emulation layer on Pulse to allow multiple ALSA and Pulse applications to run together through the mixer. Media libraries such as GStreamer (which powers consumer-audio applications such as VLC, Skype and Flash) and libcanberra (the GNOME desktop sound system) have been developed closely with PulseAudio, increasing its popularity. However, Pulse is not designed for pro-audio work as such work requires very low latencies and minimal drop-outs.

The JACK system is an alternative software mixer for pro-audio work. Like the other soft mixers, JACK runs on many lower level platforms – usually ALSA on modern Linux machines. The bulk of pro-audio applications such as Ardour, zynAddSubFx and qSynth run on JACK. JACK also provides network streaming, and emulations/interfaces for other audio APIs including ALSA, OSS and PulseAudio. (Pulse-on-JACK is useful when using pro and consumer applications at the same time, such as when watching a YouTube tutorial about how to use a pro application. This re-configuration happens automatically when JACK is launched on a modern Pulse machine such as Ubuntu 11.10.)

5 Software setup

Our DAW is a relatively low-power Intel E8400 (Wolfdale) duo-core, 3GHz, 4Gb Ubuntu Studio 11.10-64-bit machine. Ubuntu studio was installed directly from CD – not added as packages to an existing Ubuntu installation – this gives a more minimalist installation than the latter approach. In particular the window manager defaults to the low-power XFCE, and resource-intensive programs such as Gnome-Network-Monitor (which periodically searches for new wifi networks in the background) are not installed. Ubuntu was chosen for compatibility and familiarity with our other desktop machines. (Several other audio distributions are available including PlanetCCRMA(Fedora), 64Studio(Debian), ArchProAudio(Arch)).

The standard ALSA and OSS provide interfaces to USB and PCI devices below, and to JACK above. However for firewire devices such as our Pre8, the `ffado` driver provides a *direct* interface to JACK from the hardware, bypassing ALSA or OSS. (Though the latest/development version provides an ALSA output layer as well.) Our DAW uses `ffado` with JACK2 (Ubuntu packages: `jack2d, jack2d-firewire, libffado, jackd, laditools`. JACK1 is the older but perhaps more stable single-processor implementation of the JACK API) and fig. 3 shows our JACK settings, in the `qjackctl` tool. The firewire backend driver (`ffado`) is selected rather than ALSA.

We found it useful to unlock memory for good JACK performance.[2] As well as ticking the un-

[2]By default, JACK locks all memory of its clients into RAM, ie. tells the kernel not to swap their pages to virtual memory on disc, see mlock(2). Unlock memory relaxes this slightly, allowing just the large GUI components of clients

lock memory option, the user must also be allowed to use it, eg. `adduser charles audio`. Also the file `/etc/security/limits.d/audio.conf` was edited (followed by a reboot) to include

```
@audio - rtprio 95
@audio - memlock unlimited
```

These settings can be checked by

```
ulimit -r -l.
```

The JACK sample rate was set to 48kHz, matching the Pre8s. (This is a good sample rate for speech research work as it is similar to CD quality but allows simple sub-sampling to power-of-two frequencies used in analysis.)

Fig. 4 shows the JACK connections (again in `qjackctl`) for our meeting room studio setup. The eight channels from the converter-mode Pre8 appear as ADAT optical inputs, and the eight channels from the interface-mode Pre8 appear as 'Analog' inputs, all within the firewire device. Ardour was used with two tracks of eight channel audio to record as shown in fig. 5.

5.1 Results

Using this setup we were able to record simultaneously from six overhead microphones, eight table-centre microphones, and two wireless headsets, as illustrated in fig. 5. We experienced no JACK xruns in a ten minute, 48kHz, 32-bit, 16-channel recording, and the reported JACK latency was 8ms. A one hour meeting recording with the same settings experienced only 11 xruns. Total CPU usage was below 25% at all times, with `top` listing the following typical total process CPU usages: `jack` 11%, `ardour` 8%, `jack.real` 3%, `pulseaudio` 3%.

However, we were unable to play audio back through the Pre8s, hearing distorted versions of the recording. For our speech recognition this is relatively unimportant, and can be worked around by streaming the output over the network with JACK and playing back on a second machine. The `ffado` driver's support for the Pre8 hardware is currently listed as 'experimental' so work is needed here to fix this problem.

The present two Pre8 system is limited to 16 audio channels, we plan to extend it with further firewire devices to record from more audio sources around the meeting room and from tele-

to perform swapping, leaving more RAM free for the audio parts of clients.

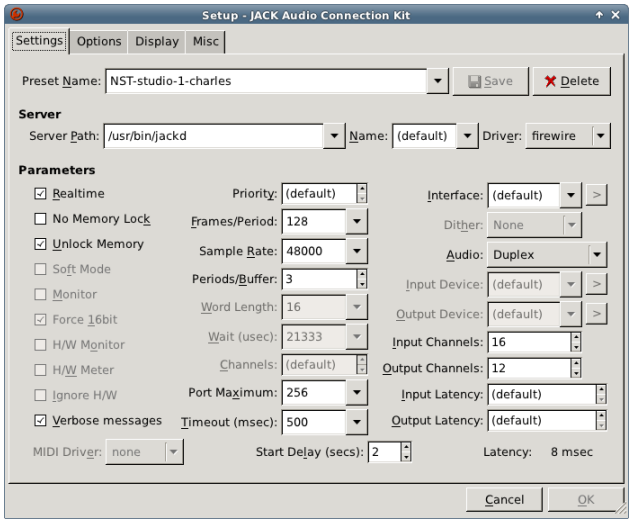


Figure 3: 16 channel recording JACK settings.

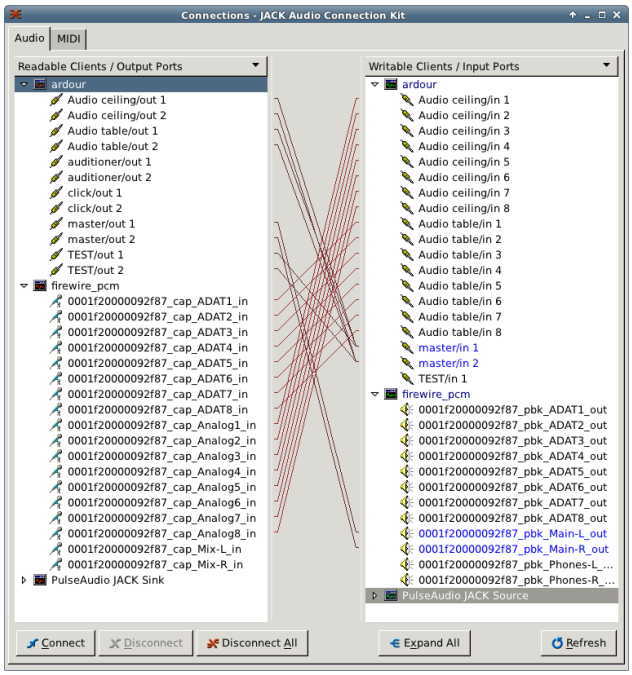


Figure 4: 16 channel recording JACK connections.

conferencing channels in future. We have not yet needed to make further speed optimisations, but we note that for future, more-channel systems, two speedups include disabling PulseAudio (adding `pulseaudio -kill` and `pulseaudio -start` to qjackctl's startup and shutdown option is a simple way to do this); and installing the real-time `rt-linux` kernel.

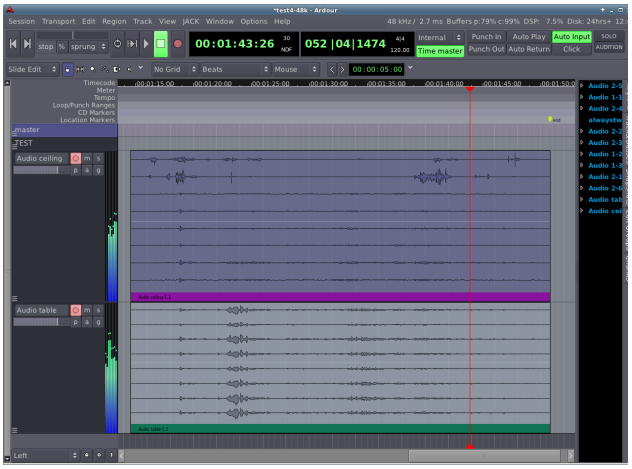


Figure 5: 16 channel meeting room recording in Ardour, using two 8-channel tracks.

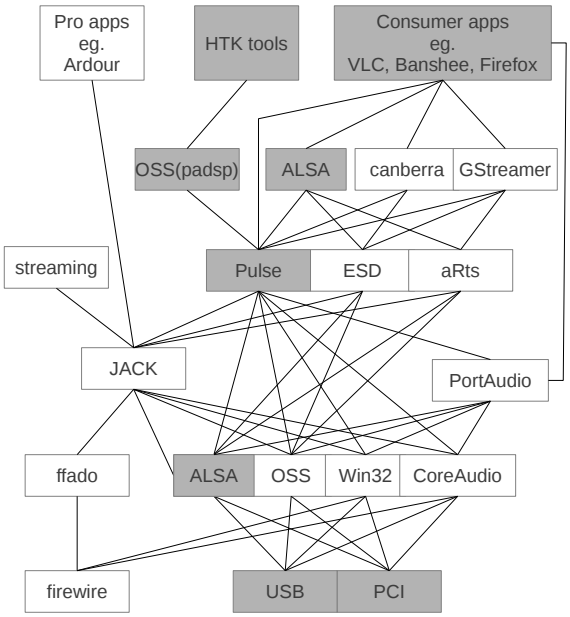


Figure 6: Audio system for data analysis.

5.1.1 Audio analysis

Analysis of our audio data is performed on a compute cluster of 20 Linux nodes with 96 processor cores in total, running the Oracle (formerly Sun) Grid Engine, the HTK Hidden Markov Model Tool Kit [Young et al., 2006] and the Juicer recognition engine [Moore et al., 2006]. During analysis, audio playback on desktop machines is useful and is done with the setup of fig. 6. For direct audio connections the HTK tools make use the OSS sound system, which may be emulated on PulseAudio on a modern machine, by installing the `padsp` tool (Ubuntu package `pulseaudio-utils_0.9.10-1ubuntu1_i386`) then prefixing all audio HTK commands with `padsp`. Similarly, Juicer allows the use of an OOS front, the implementation of a JACK plugin is in progress.

The cluster may be used both for online and offline processing. An example of online processing can be found in the description of an online transcription system for meetings [Garner et al., 2009]. Such systems distinguish between far and near-field audio sources and employ beamforming and echo cancellation procedures [Hain et al., 2009] for which either sample synchronicity or at least minimal latency between channels is of utmost importance. For both offline and online processing typically audio at 16kHz/16bit is required, to be converted into so-called feature streams (for example Perceptual Linear Predictive (PLP) features [Hermansky, 1990]) of much lower bit rate. Even higher sampling frequencies (up to 96kHz are commonplace) are often required for cross-correlation based sound source localisation algorithms to provide sufficient time resolution in order to detect changes in angles down to one degree. In offline mode the recognition of audio typically operates at 10 times real-time (i.e. 1 hour of audio in many channels takes 10 hours to process). However, the grid framework allows the latency of processing to drop to close to 1.5 times real-time using massive parallelisation.

6 Future directions

The ultimate goal of NST is to obtain transcriptions of what was said in natural environments. Traditionally, the source separation and denoising techniques of sec. 2 have been treated as a separate preprocessing step, before the cleaned audio is passed to a separate transcriber such as HTK. However for challenging environments this is suboptimal, as it reports only a single estimate of the denoised signal rather than Bayesian information about its uncertainty. Future integrated systems could fuse the predictions from transcription language models with inference and reporting of low-level audio data, for example by passing real-time probabilistic messages between HTK's transcrip-

tion inference (on a Linux computing cluster) and low-level audio processing (on desktop or embedded Linux close to the recording hardware.)

For training and testing speaker location tracking systems it is useful to build a database of known speaker position sequences, which need to be synchronised to the audio. Positions change at the order of seconds so it is wasteful to use audio channels to record them – however we note that JACK is able to record MIDI information alongside audio, and one possibility would be to encode position from our Ubisense active badges as MIDI messages, synchronous with the audio, and record them together for example in Ardour3. It could also be useful to – somehow – synchronise video of meetings to JACK audio.

The homeService system has two major parts, namely hardware components that are deployed in people's homes during trial (Android tablet, Microcone, and Linux box responsible for audio capture, network access, infrared and blue tooth communication), as well as our Linux computing cluster back at the university running the processes with particularly high demands in terms of processing power and memory usage. The main audio capturing will take place on a Linux box in the users' home, and we plan to develop a setup which will enable the Microcone and JACK to work together and provide – if needed – live streaming of audio over the network.

The main requirements of NST research for Linux audio are support for sample-synchronous, many (e.g. 128 or more) channel recording, and communication with cluster computing and speech tools such as HTK. As natural speech technology makes closer contact with signal processing research, we expect to see more speech researchers moving to Linux audio in the near future, and we hope that this paper has provided some guidance for those who wish to make this move, as well as a guide for the Linux audio community about what technologies are important to this field.

Acknowledgements

The research leading to these results was supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

References

L. Brayda, C. Bertotti, L Cristoforetti, M. , Omologo, and P. Svaizer. 2005. Modifications on NIST MarkIII array to improve coherence properties among input signals. In *Proc. of 118th Audio Engineering Society Conv.*

H. Christensen and J. Barker. 2010. Speaker turn tracking with mobile microphones: combining location and pitch information. In *Proc. of EUSIPCO.*

C. Fox, M. Evans, M. Pearson, and T. Prescott. 2012. Tactile SLAM with a biomimetic whiskered robot. In *Proc. ICRA.*

P. N. Garner, J. Dines, T. Hain, A. el Hannani, M. Karafiat, D. Korchagin, Mike L., V. Wan, and L. Zhang. 2009. Real-Time ASR from Meetings. In *Interspeech'09*, pages 2119–2122.

T. Hain, L. Burget, J. Dines, P. N. Garner, A. el Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan. 2009. The AMIDA 2009 meeting transcription system. In *Proc. Interspeech.*

H. Hermansky. 1990. Perceptual linear predictive analysis for speech. *J. Acoustic Society of America*, pages 1738–1752.

J. Mooney. 2005. *Sound Diffusion Systems for the Live Performance of Electroacoustic Music.* Ph.D. thesis, University of Sheffield.

D. Moore, J. Dines, M. Magimai Doss, J. Vepa, O. Cheng, and T. Hain. 2006. Juicer: A weighted finite state transducer speech decoder. In *Machine Learning for Multimodal Interaction*, pages 285–296. Springer-Verlag.

S. Roberts and R. Everson, editors. 2001. *Independent Components Analysis: Principles and Practice.* Cambridge.

H. L. Van Trees. 2002. *Optimum array processing.* Wiley.

L. Watts. 2009. Reverberation removal. In *United States Patent Number 7,508,948.*

M. Wolfel and J. McDonough. 2009. *Distant Speech Recognition.* Wiley.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, XA Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. 2006. The HTK book.

Review of Bayeian Brain, book edited by Doya, Pouget and Rao. MIT Press, 2007.
Review by Charles Fox, Adaptive Behavior Research Group, University of Sheffield, UK.

Computational Neuroscience and AI have diverged greatly over the past decade: what was once a coherent field which thrived on trading biological and engineering models has fragmented into independent camps, one seeking to model brains and the other building useful artifacts. Bayesian Brain gives an overview of the recent attempts by a core group of researchers to reunite these fields and bring back the cross-discipliniarity that once characterized them.

The book portrays two key ideas: first, that probabilistic Data Analysis methods are useful in analyzing neural data; and second – more speculatively -- that Bayesian computational methods are useful models of the functions of the brain itself. These goals are distinct but share similar, cross-fertilizable concepts. However it could be argued that the connection is somewhat tenuous – and the two parts are almost completely independent books.

After Doya and Ishii's basic probability and information theory tutorial, Fairhall gives an introduction to spike coding in the single neuron. The key questions in this field ask what is represented, and what granularity of coding is used. The differences in information about stimuli contained in different granularities and time-windows can be quantified, however a difficulty inherent in the brute-force, model-free, information-theoretic approach is the exponential-sized state space and data collection requirements to populate it. The discussion thus shifts to parametric models, introducing the the Spike-Triggered Average, cascade and Integrate-and-Fire models. Efficient parameter fitting in a related approach, the linear-nonlinear Poisson model, is detailed at length in a chapter by Pillow. Another alternative is to model the distribution of single spike times: Richmond and Weiner discuss the use of order statistics, which use combinatorics to give the temporal distribution of the kth spike out of n spikes. Bayes' theorem can then be used to invert this model and make inferences about the stimulus.

Moving suddenly from spike-level models to whole-brain interactions, Penny discusses the use of Dynamic Causal Modeling for temporal fMRI activity in macroscopic brain regions. DCM may be viewed as a generalization of linear Gaussian Bayesian Networks which drops their acyclicity constraint and so models the time sequence of the network state. With network topology constrained by anatomical data, the model weights can be learned by standard Expectation Maximization (EM) iteration.

The second half of the book examines probabilistic structures as models of brain function itself. Pouget and Zemel introduce various schemes for representing probability distributions using populations of neurons, as found in orientation-selective visual hypercolumns, place cells, and others. In contrast to the classical view of the maximum response simply coding the best solution, they consider the whole collection of responses as coding for the uncertainty during inference. Latham and Pouget give a heuristic, bottom-up architecture for computing in the style Bayesian network nodes, using population coding; Rao provides a principled neural implementation of the Pearl message-passing inference algorithm.

Undirected networks – often called Markov Random Fields – have been used in the computer vision community for preconceptual image segmentation. Lee and Yuille review an annealing and EM style method for simultaneous approximate inference of region boundaries and region content in the MRF framework. They then advance it has a theory of V1 activity, presenting biological evidence to subsume

mere orientation-selectivity into region and boundary responses.

Knill considers Bayesian inference as a normative theory of behavior, and uses visual psychophysics results to support this claim. In general, Bayesian inference is NP-hard so approximations must be used: the study of what deviations from the Bayesian ideal are made by the brain could shed light on what approximation methods are being used within this paradigm.

The remaining chapters shift from perception to action. Using the Martingale approach, Shadlen et al. review the sequential sampling problem: how much time and resources to expend on gathering evidence before making a decision. They go on to give evidence that this HMM-like process could be implemented in brain areas MT and LIP, representing current observations and accumulated beliefs respectively. (No mention is given to similar theories of Basal Ganglia function.) Tororov provides a highly mathematical tutorial on Kalman filters and control theory in the continuous-variable framework – an alternate view of Markov Decision Processes than the discrete approach often taken in Reinforcement Learning. These concepts are used in psychophysical experiments discussed by Kording and Wolpert, who like Knill, test human behavior against Bayesian optimal decisions. They also make some neurobiological speculations, including complementary roles for neurotransmitters ACh and NE in representing the almost Rumsfeldian expected uncertainty and unexpected uncertainty respectively.

Some notable omissions from the discussions: First, while the question of rates versus spike time representations appears often, there is no attempt to link it to what Machine Learning has to say about similar models. Variational and Gibbs Sampler methods are directly analogous to rates and spikes, and their pros and cons as approximate inference methods are well-known and could perhaps shed light on the biological debate. Second, the Bayesian paradigm is currently fashionable across many fields, and the book works firmly within it – but perhaps a discussion of how it relates to rival neural paradigms would be useful for the not-yet-converted? For example, many iterative Bayesian algorithms such as Variational Bayes and loopy Belief Propagation can be viewed as computing fixed points, in a similar way to the Dynamical Systems view of the brain. A final oddity is the amount of non-Bayesian statistics used throughout the book: estimators, Cramer-Rao bounds and General Linear Models all make appearances: while a comprehensive Bayesian tutorial is given in the first chapter, many classical concepts are simply assumed.

A common theme in the book is that optimal, exact Bayesian inference is computationally intractable, so both computers and brains must use heuristics to approximate it. The study of deviations from the ideal can give clues about what heuristics are used, both at the neural and psychophysical levels. However there is a danger of non-falsifiable science here: if the brain performs optimally then it is said to be a "Bayesian Brain"; but every deviation from this can be explained away as a heuristic. So how could we ever show that the brain is /not/ Bayesian?

Overall this is a much-needed and intriguing book, giving a birds-eye view of the whole research programme of rejoining Computational Neuroscience with the Machine Learning school of AI. The book showcases research from the many facets of this project, from information-theoretic spike models to neural implementations of belief propagation and Bayesian-normative psychophysics. There is a wealth of references and the book will reward many re-readings, both for overviews of the topics and as initial pointers into the literature, especially across disciplines. Many theses could be inspired by chapters of this book.

Maximum utility unitary coherent perception vs. the Bayesian brain

Charles Fox (charles.fox@sheffield.ac.uk)
Department of Computer Science, University of Sheffield, UK

Tom Stafford (t.stafford@sheffield.ac.uk)
Department of Psychology, University of Sheffield, UK

Abstract

Our subjective experience of the world is 'unitary coherent' (UC). 'Unitary' means we only perceive one interpretation at a time rather than a blur of multiple possible worlds. 'Coherent' means that we almost always perceive scenes that do not contain contradictory parts. While this form of first-person perceptual experience may seem obvious, it is in opposition to the requirements of optimal decision making, and to some forms of the 'Bayesian brain' hypothesis. We hypothesise that there are at least three types of 'Bayesian' action selection occurring in cognition, including a 'maximum utility (MU) percept' strategy that makes use of UC percepts. We give evidence from a video game experiment that is compatible with MU/UC perception and action selection, and is incompatible with optimal actions. Furthermore, it is compatible with the presence of utility bias in MU/UC perception: by changing the available actions we may be able to manipulate the subject's percept of a fixed ambiguous stimulus.

Keywords: Bayesian; psychophysics; utility; bias; perception

A paradox about perception

Our subjective experience of the world is 'unitary coherent'. *Unitary* means we only perceive one interpretation at a time (e.g. either a face *or* a vase in the Rubin Vase illusion) rather than a blur of multiple possible interpretations (never the face *and* vase together). *Coherent* means that we almost always perceive scenes that do not contain contradictory parts. (e.g we do not see part face and part vase). While the UC nature of perception may seem obvious from subjective experience, it is in opposition to the requirements of optimal decision making, which require consideration of *all* possible interpretations of sensory data (Bernardo & Smith, 2000). In particular, the 'Bayesian brain' hypothesis (Doya, Ishii, Pouget, & Rao, 2007) views perception as computing probabilities of many interpretations, and optimal actions would be found by integrating out the utility of all actions under all percepts. If both Bayesian brain research and optimal action theory (Körding & Wolpert, 2004) suggest that perception should operate using a distribution, or 'Bayesian blur' of possible percepts, why then is our subjective experience limited to a unitary coherent percept instead? And which unitary coherent percept do we perceive: the most probable one or the most useful one? This study argues that as full Bayesian perception and action selection is computationally hard, an approximation which we call 'maximum utility (MU)' perception is a useful surrogate. It then presents evidence in support of the maximum utility perception hypothesis using a video game style experiment.

Types of perception and action selection

When a unitary coherent percept is required in machine perception, such as the output of a machine vision (Felzenszwalb & Huttenlocher, 2005) or speech recognition system (Young et al., 2006), the maximum a posteriori (MAP) state is often used by system engineers,

$$s_{MAP} = \arg_s \max P(s|d), \tag{1}$$

where s are world states and d is the available data. However real-world agents are often required to make actions as well as – or instead of – reporting percepts. In these cases, perceiving the MAP state does not necessarily lead to the best action if the following naive action-selection rule is used as a separate stage following MAP perception,

$$a_{naive} = \arg_a \max U(a, s_{MAP}), \tag{2}$$

where U is utility, a are actions. Instead, optimal actions are obtained (Bernardo & Smith, 2000) by maximising *expected* utility (MEU), which requires integrating over the 'Bayesian blur' of possible worlds,

$$a_{MEU} = \arg_a \max \int_s U(a,s)P(s|d)ds. \tag{3}$$

MEU action selection has no role for unitary coherent percepts. Instead it must consider *every* interpretation s. Computational approximations to this integral might ignore some improbable interpretations (Spiegelhalter, Thomas, Best, & Gilks, 2008), but still sum over a *set* of possible world states s rather than privileging any particular unitary state.

In many cases humans have been claimed to make optimal actions (Griffiths & Tenenbaum, 2006). This may occur for low-level, rapid stimulus-response type actions, and for high-level cognitive decisions such as business and financial decisions. Much recent work in 'Bayesian Cognitive Science' proceeds by *assuming* an MEU framework, then reasoning backwards from observed actions to report human priors on various stimuli (Stone, Kerrigan, & Porrill, 2009).

So why then do we bother to perceive UC percepts? Do they have some functional significance as well as being correlates of subjective experience? If they do have a function, this would suggest that Bayesian Cognitive Science's assumption of optimal action is flawed, and could potentially invalidate some of its reported human priors.

Bayesian inference and hence MEU decision making is generally an NP-hard problem (Cooper, 1990) so is impractical for all but the most constrained percepts and actions.

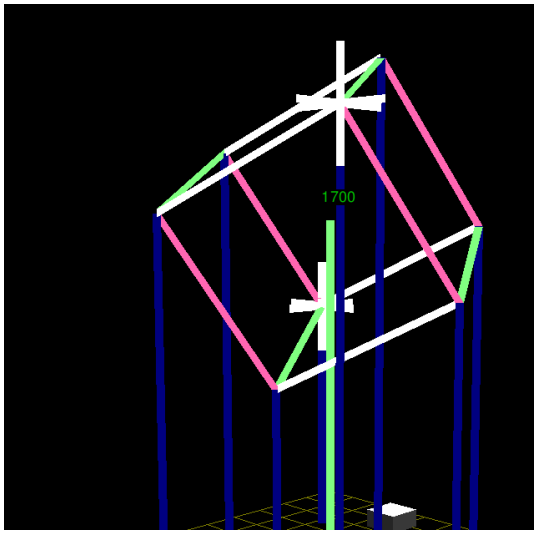


Figure 1: 3D environment used in the training phase of the game. A joystick moves the missile launcher around a 2D (x,z) plane on the ground. Pressing and holding the joystick button fires a missile (not shown) vertically upwards (y axis). Releasing the button detonates the missile. Points are scored for detonation close to the target(s) shown by white crosses. The training phase shown here includes colour, overlap, perspective and support cues to make the cube's configuration unambiguous. These cues are removed in the test phase to leave an ambiguous Necker cube, with ambiguous 3D target positions. Figure is best viewed in colour.

It has been suggested (Gigerenzer & Todd, 1999; Goldstein & Gigerenzer, 2002) that making actions based on a single 'best' percept (such as the 'take the best' heuristic and 'less if more; effect) could be a useful heuristic to speed up the decision making process at the expense of optimality. However the 'percepts' in these cases are high level logical states of the word rather than actual perceptual objects in three dimensional space.

We propose an alternative form of perception and action selection to MAP perception and MEU action selection, which we call *maximum utility perception (MU)*. In MU we choose a UC state and action together,

$$(s_{MU}, a_{MU}) = \arg_{s,a} \max U(a,s)P(s|d) \qquad (4)$$

which yields the best possible action assuming that only a single world state can be considered.

The MU Hypothesis

We hypothesise that humans have at least *three* kinds of decision making behaviour, moving from fast and simple to slow and accurate:

1. *Immediate stimulus-response (S-R)*. A fast association from input data directly to an action. Such mappings do not need to build a UC percept. The could be implemented neurally at the sub-cortical level, such as direct links from superior colliculus to basal ganglia (Redgrave, Prescott, & Gurney, 1999). For simple mappings, it is possible that simple computational structures such as small neural networks could learn to perform near-optimal action selection, as has been demonstrated by computational experiments (Ramamoorthy & Verguts, 2012). Near-optimal performance in fast, low-level tasks such as reaching and pointing *quickly* at spheres having different locations and utilities (Kording & Wolpert, 2007); and 'simple heuristics' (Kahneman, Slovic, & Tversky, 1982; Gigerenzer & Todd, 1999) would be candidates for this mechanism. For simple tasks this method would give MEU-like results but without explicitly performing the MEU integration.

2. *Fast MU/UC percept-response (P-R)* which achieves suboptimal MU action in reasonable computation time. UC perception could be implemented cortically, with high-level perceptual areas computing a single most *useful* percept of the world, jointly with action selection under utility bias. Evidence for UC perception is found in binocular rivalry experiments (Srinivasan & Nunez, 2006), and in computational models (Riesenhuber & Poggio, 1999) as well as in everyday subjective experience. This paper gives evidence for MU/UC perception.

3. *Full MEU* action selection, via conscious sequential consideration of many possible percepts and responses. This slow type of decision making would occur for example when making a business decision, where several minutes (or even hours or days) are set aside to consciously perceive one possible world at a time, and the effects of many possible actions in them are simulated, and the resulting utilities averaged over. Humans are well-known to be poor at this kind of computation (Kahneman, 2003), and real-life action selection of this type is often performed in the business world by specialised operations researchers making use of computers to calculate the expected utilities (Pourret, Naïm, & Marcot, 2008), rather than relying on their own cognitive faculties.

If multiple decision making systems exist, it seems likely that the basal ganglia system is used to switch between them, for example taking account of time pressures for the type of decision to be made (Lengyel & Dayan, 2008; Redgrave et al., 1999; Daw, Niv, & Dayan, 2005). Strong support for the existence of at least two systems comes from the Ebbinghouse illusion, which produces different perceptual reports in verbal and stimulus-response type actions. It has been shown (Goodale & Milner, 1992) that the motor actions are consistent with optimal MEU-like decisions in the *same* subjects that make incorrect verbal reports.

While research on near-Bayesian optimal decisions of the S-R and Full MEU types abounds, there has been comparatively little work on the role of unitary-coherent perception in decision making. While our subjective experience tells us very clearly that *something* in the brain is computing a UC percept (which is incidentally presented to our conscious experience), and researchers have modelled how MAP percepts could be computed in this way (Riesenhuber & Poggio, 1999)

there has been little study of how this type of perception could be used in action selection as a replacement for S-R and MEU behaviour. Our hypothesis is that MU perception and action is in fact the dominant mode of everyday, aware, perception and action – the type of cognition that occurs consciously but not deliberatively.

It is difficult to design experiments to isolate this middle, MU, level of perception, because as soon as subjects know their performance is being monitored they tend to start deliberating as in Full MEU, rather than performing 'everyday' perception and action selection. Conversely, if tasks are too low-level and fast-paced, they will use rapid S-R behaviour. Perhaps that is why few experiments have noticed MU effects before. To this end, we have carefully designed a simple 3D perception task, and examine two hypotheses:

Hypothesis H1 is that there are examples of human behaviour that are consistent with UC perception and inconsistent with both Full MEU (deliberative) and approximate MEU (S-R). A positive result here would stimulate further research into delimiting the circumstances in which the different behaviour types are employed.

Hypothesis H2 is that the particular kind of UC used in human perception is the MU percept. To find evidence for this stronger hypothesis, we will examine if it is possible to bias the percept from equally a priori probably percepts by altering the available action set, as predicted by MU.

Methods

A video game – loosely based on "space invaders" – was designed and implemented[1], having optimal MEU actions that require consideration of multiple scene interpretations, and having MU actions giving suboptimal rewards. If human behaviour in this (or any other) game could be shown to deviate from MEU behaviour and be consistent with MU, then evidence is provided for H1. Further, if the human behaviour is consistent with predictions made by MU selection, then evidence is provided for H2. An overview of the phases of the game is given here, followed by details of each phase.

In phase one of the game, shown in fig. 1, subjects were trained in several rounds to fire missiles from a launcher in a 2D plane, in an unambiguous simulated 3D environment. They received rewards according to how close to aerial targets (shown as white crosses in the figure) they get. After demonstrating that they understand the utility function and controls by passing a second, 'examination' phase, they are then tested (phase three) in an ambiguous bi-stable environment. A true MEU strategy would consider both interpretations of this environment, whereas a UC based strategy would use only one and lead to a different action. Phase one consists of several rounds which teach the subject about the game. The choice of tasks here is fairly arbitrary, as the logic of the experiment is that *if* subjects can pass the phase two exam *then* they have demonstrated an understanding of the rules sufficient to play the real test game in phase three. Phases one and two may be repeated in any order as many times as the subject desires, until the exam is passed successfully.

Phase 1: exploratory training round

The game environment consists of a visible 2D horizontal plane on which a missile base can move around, a wire-frame cube in the 3D space above the plane, and one or two targets located at vertices of the cube. The environment is drawn using very strong perspective[2], and the vertices of the cube are connected vertically to the plane by lines to make their 3D locations unambiguous. In addition, edges of the cube are drawn with think lines of different colours, producing additional disambiguation cues where one lines is seen to cross in front of another.

Subjects control the (x,z) position of a missile base using an analogue joystick (Logitech Extreme 3D Pro) and fire a missile by pressing and holding the joystick trigger. Once fired, the missile moves upwards (the y direction) until the trigger is released. The missile then explodes at position $\mathbf{m} = (m_x, m_y, m_z)$. $N \in 1,2$ targets are present in the environment at positions $\mathbf{t}^i = (t^i_x, t^i_y, t^i_z), i = 1:N$, and a Gaussian reward R is received and displayed centre-screen after the explosion,

$$R = \sum_{i=1}^{N} r_i, \tag{5}$$

$$r_i = 100 \times \exp - \frac{(\mathbf{m}-\mathbf{t})(\mathbf{m}\text{-}\mathbf{t})^T}{2\sigma^2}. \tag{6}$$

The spread, σ was fixed at a large enough value ($\sigma = 2\sqrt{3}/\sqrt{2\ln 2}$) so that if two targets are present, the score is always highest when firing at the point between them than when firing directly at one of the targets.

In the exploratory round, single targets are presented at different vertices of a fixed cube. Subjects have unlimited time to position the missile base and fire. They are then represented with a visual display of the reward, then the next target is presented. They are given 50 such targets to practice with, no cumulative score display, and are encouraged to experiment to learn about the rewards available at different distances from the target by an introductory message.

Phase 1: utility training round

In the exploratory round, subjects obtained high scores by firing as close to the target as possible. To help them learn about the shape of the Gaussian utility function, a series of rounds takes place in which fixed cubes are shown and the subject is asked to deliberately score *only* 50,70 or 90 points. Thus they are encouraged to try aiming at locations at different distances from the target.

Phase 1: double target training

This round is similar to the first exploratory round, but uses two targets presented together at each trial. By construction

[1] in Python, source code available on request.

[2] OpenGL: gluPerspective(45, 1.0*width/height, 0.1, 100.0); gluLookAt(7,0.05,7, 0,-1.25,0, 0,1,0). Cube faces are 2*2 units.

(choice of σ), the optimal action is now always to aim at the point midway between the targets.

Phase 1: blackout training rounds

Two further training rounds take place. In the first, the lower half ($x+z>0$) of the missile-launching area is 'blacked out'. It is coloured red, and the joystick is unable to move into the red area. In the second round, the situation is reversed at the top half of the grid ($x+z<0$) is blacked out. Subjects learn that that optimal strategy when faced with a target in the blacked out region is to fire from a position as close to it as possible that is on the centre line.

Phase 2: examination round

The purpose of the examination round is to demonstrate that the subject has learned the optimal actions for single and double targets, as well as possessing sufficient motor skills to control the game using the joystick. 20 trails are presented in rapid succession (one every 5s) and a cumulative score is maintained. If subjects fail to score 2170 points or more, they are sent back to the exploratory phases then made to repeat the examination (or allowed to leave the experiment). The qualifying score was chosen such that it can only be obtained by using the optimal strategy of aiming closer to the centre of each pair of double targets than to either individual target. Thus by passing the examination phase, subjects demonstrate knowledge of this strategy.

Phase 3: Ambiguous test round

In trials within this round, a bi-stable ambiguous (Necker) cube is presented to the subject very quickly, at random orientations. In some (80% of) trials there is one target at a random vertex. In others there are two targets, which may be at opposite (10% of trails) or non-opposite (10% of trials) vertices. The percept is made ambiguous by switching the projection from perspective to orthographic, dropping the vertex-to-plane cues, and drawing all edges in white to remove overlap cues. To motivate subjects, they are told that that their cumulative score from all rounds of phase three will be their reported result, and that the subject with the highest reported result will receive twenty UK pounds in cash. (Subjects were undergraduate psychology students and were not paid otherwise; but received credit towards their degree for participating.)

Phase 3: Blackout test rounds

The ambiguous test round is repeated twice more, with blacked out near and far regions as in the learning phase.

Debriefing

It is crucially important that subjects do not become aware of the ambiguity, because this could allow high-level (Full MEU) reasoning to aim in the centre, and destroy any UC-revealing behavioural effects. For this reason, after phase 3, subjects were told about Necker cubes and asked if they were aware of the Necker ambiguity.

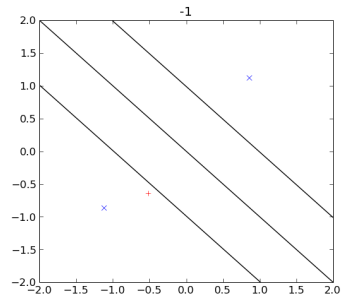


Figure 2: Example of a single-target trail, (x,z) plane. The viewer's position is in top right corner. Blue crosses show the two ambiguous target positions resulting from a single target vertex on a Necker cube. The red cross shows the firing position. Black lines show the centre line and the two classification boundaries, dividing the launching area into near (top-right), centre and far (bottom-left) firing regions.

Processing

25 subjects were tested. Of these, 20 completed the exam and proceeded to generate data in the test phases. In debriefing, no subjects reported awareness of the ambiguity in the Necker cubes. For each trial, the 3D positions of both ambiguous locations of the target or targets were computed. This was achieved by transforming the x,z joystick co-ordinates into a horizontal and depth pairs, (h,d), then flipping the depth coordinate,

$$\begin{bmatrix} h \\ d \end{bmatrix} = H \begin{bmatrix} x \\ z \end{bmatrix}, \tag{7}$$

where H is the Hadamard matrix,

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{8}$$

The complementary ambiguous location is thus

$$\begin{bmatrix} x' \\ z' \end{bmatrix} = H^{-1} \begin{bmatrix} h \\ -d \end{bmatrix}. \tag{9}$$

Furthermore, the height coordinate was transformed by $y' = y - cd$, where c is a constant ($c = 0.9$) which compensates for the choice of viewing angle in the projection images.

All shots were classified into three regions (fig 2), according to whether their (x,z) firing locations were closest to the near ambiguous target location, the far ambiguous target location, or the centre line.

Results

Non-blackout trials with single targets

In these trials, the MEU action by construction (i.e. choice of σ as described in phase 1) is to fire at the point on the centre line between the two possible ambiguous locations. (fig. 2). In contrast, the MU action is to fire directly at a

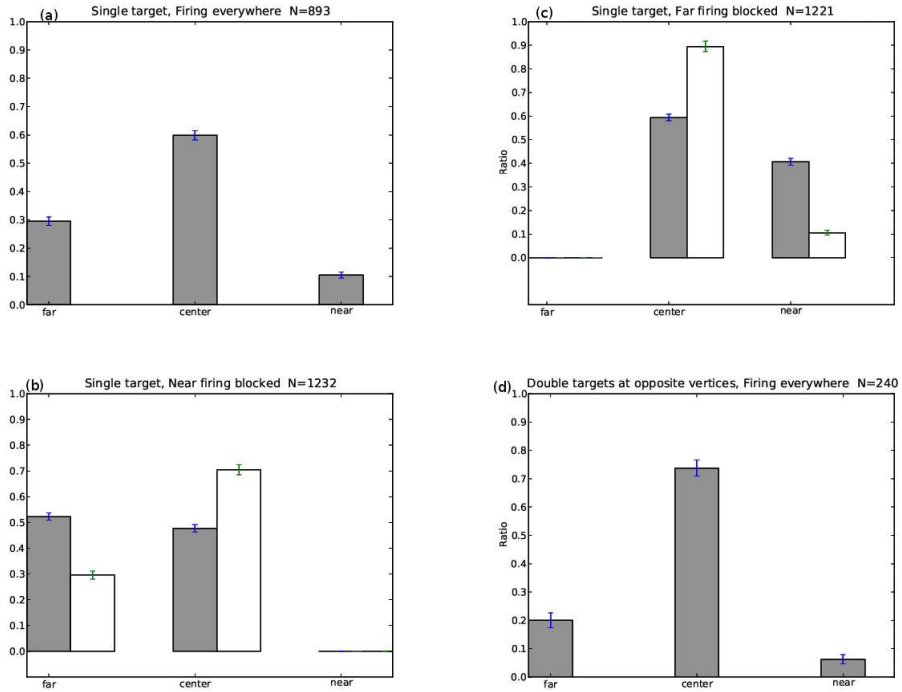


Figure 3: Results. Grey bars: observed frequency ratios, with Beta posterior, one-standard-deviation confidence intervals. White: predictions for blocked firings under the null hypothesis. The null hypothesis is that the unblocked area is unchanged from the 'single target, firing everywhere" case, while the centre is the sum of the centre and blocked firings from that case. All error bars assume IID observations and ignore which observation came from which subject.

randomly-chosen single one of those locations. This is because the MEU action averages over the two possible states of the world, which gives the same calculation as choosing where to fire in an unambiguous double target case; whereas the MU action picks just one interpretation of the target location, then fires directly at it.

Fig. 3a shows the distribution into region classes, over all subjects and trials of this type. Treating each action by each subject as an independent observation (i.e. ignoring subject-specific effects), and beginning with a flat Beta prior over the ratio of shots in each region, we infer posterior ratios, along with uncertainties. The figure shows the mean and one standard deviation error bars inferred about the population ratio of shots fired of each type. Signal detection theory can be used to obtain the p values, but broadly two ratios are significantly different if pairs of error bars do not overlap.

The results of these trials are surprising but inconclusive. Although target locations are perfectly ambiguous between near and far positions, subjects show a preference for the far target over the near one. That is, they are already interpreting the Necker cubes percept in a biased way, to favour interpretations with the target at the back of the scene.

If the MEU strategy was followed perfectly, we would see all shots fired in the centre and none in the near or far regions. If the MU strategy was followed perfectly, we would see all shots in the near and far regions and none in the centre. Unfortunately, we see shots in all three regions. Whilst this is incompatible with a pure MEU strategy, it can weakly be explained from a MU perspective: A large number of shots are fired from the centre line, which may be due to subjects losing all depth perception (i.e. not perceiving the cube as 3D at all) and hedging by firing in the centre; they may also be due to limited depth perception resulting in a stable cube percept but an inaccurate joystick placement. (Some subjects commented on the lack of training in the absence of the vertical supports, and consequent loss of skill at pointing to the depth of the targets.) The near and far shots would be correct MU actions, and the centre shots due to a problem with the experiment, requiring a better communication of the depth to the subjects in future versions whilst retaining the ambiguity.

Blackout trials with single targets

In these trials, the pure MEU action is *still* to fire at the point on the centre line between the two possible ambiguous locations. Points on the centre line are still available during a blackout, so the optimal strategy is unchanged.

Fig. 3b shows the results when the near-side is blacked out. The majority of shots are now fired in the far region. This is consistent with the MU strategy: actually *perceiving* and acting on the Necker cube in the configuration which enables the target to be reached; an optimism bias. If we assume UC perception and action, these new results then show MU-like bias occurring within in. For comparison, we show in white the prediction of a null hypothesis. This is obtained taking

the Single target histogram of fig. 3a and moving its near mass to its centre mass, as would occur if UC percepts were unchanged by the utility bias, and near shots were substituted by firing on the centre line as close to blacked-out near targets as possible. MEU theory is unable to explain why the observed frequencies are so different from this null hypothesis. MU theory explains it easily: there is no utility in *perceiving* near targets; but if they are re-perceived as far targets, then an increased utility can be obtained by firing at them.

Similarly, fig. 3c shows the results when the far-side is blacked out. Again compared to a null hypothesis (white bars) which moves the mass from the far region to the centre region from fig. 3a, we see a significant difference, again suggesting the MU-like change in both percept and action towards the obtainable (non-blacked-out) target position.

Fig. 3d is shown as a control. It is the distribution from the non-blacked-out trials having two targets at opposite cube vertices. In these cases, the MEU and MU strategies are the same – fire in the dead centre of the grid, and the results show a significantly increased rate of firing in the centre region over that of fig. 3a. This again supports the presence of MU over MEU, because MEU would give identical results in 3a and 3d but MU would give an increase in the centre region in 3d over 3a, which we do see here.

Discussion

In informal discussions, we often argue that "perception is obviously UC from subjective experience". Operationalists challenge this statement and would prefer us to cite an experiment to demonstrate the claim in the third person. While MEU actions are known to occur at both low-level psychophysical tasks and at high-level cognitive reasoning tasks, we have here presented evidence for the existence of a largely unexplored middle ground in which action selection is consistent with MU, and inconsistent with both Full deliberative and S-R approximate MEU, as in Hypothesis H1.

MU, but not MEU, can explain the deviations from the null hypotheses seen in figs. 3b and 3c, and also the difference between figs. 3a and 3d. The match of the data to MU behaviour in these cases gives some support for Hypothesis H2.

It was disappointing for the MU theory that fig. 3a did not present conclusive evidence by itself of MU over MEU, as pure MU would predict all shots to be fired in near and far regions, not in the centre. One way to explain the data away here is that the stimuli used were insufficiently informative to give subjects a sense of space, so they fire in the centre by default in the absence of any meaningful percept. Future work should try to refine the experiment to see if such hypothetical null percepts can be replaced by true percepts, for example by using different input and display systems while retaining the ambiguity in the Necker cube itself. Finally, the assumption that all shots by all subjects are mutually independent is strong, and future work should employ more subjects so that the assumption of independence of shots belonging to each subject can be dropped in the analysis.

References

Bernardo, J., & Smith, A. (2000). Bayesian Theory. Wiley.

Cooper, G. F. (1990). The computational complexity of probabilistic inference. *Artif. Intell.*, *42*, 393–405.

Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nat Neurosci*, *8*(12), 1704–1711.

Doya, K., Ishii, S., Pouget, A., & Rao, R. P. N. (2007). *Bayesian brain*. MIT.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55-79.

Gigerenzer, G., & Todd, P. (1999). *Simple heuristics that make us smart*. OUP.

Goldstein, D., & Gigerenzer, G. (2002). Models of ecological rationality. *Psych. review*, *109*(1), 75.

Goodale, M., & Milner, A. (1992). Separate visual pathways for perception and action. *TINS*, *15*(1), 20–25.

Griffiths, T. L., & Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psych. Science*, *17*, 767–773.

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *American Economic Review*, *93*(5), 1449–1475.

Kahneman, D., Slovic, P., & Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases*. Cambridge.

Körding, K., & Wolpert, D. (2004). Bayesian integration in sensorimotor learning. *Nature*, *427*(6971), 244–247.

Kording, K., & Wolpert, D. (2007). *Bayesian statistics and utility functions in sensorimotor control, in Doya et al.*

Lengyel, M., & Dayan, P. (2008). Hippocampal contributions to control: the third way. In *NIPS*. MIT.

Pourret, O., Naïm, P., & Marcot, B. (2008). *Bayesian networks: a practical guide to applications*. Wiley.

Ramamoorthy, A., & Verguts, T. (2012). A computational model of instruction following. *Brain Research, pre-press, doi:10.1016/j.brainres.2011.12.025*.

Redgrave, P., Prescott, T. J., & Gurney, K. (1999). The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, *89*(4), 1009–1023.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Spiegelhalter, D., Thomas, A., Best, N., & Gilks, W. (2008). *Bugs: Bayesian inference using Gibbs sampling*. Available from `http://www.mrc-bsu.cam.ac.uk/bugs/`

Srinivasan, R., & Nunez, P. (2006). *Brain networks with distinct spatial and temporal structure*. Abstr. in Assoc. Scientific Study of Consciousness, Oxford.

Stone, J., Kerrigan, I., & Porrill, J. (2009). Where is the light? Bayesian perceptual priors for lighting direction. *Proc. Royal Soc. B*, *276*(1663), 1797–1804.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., et al. (2006). The HTK book.

2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Whiskered texture classification with uncertain contact pose geometry

Mathew H. Evans, Martin J. Pearson, Nathan F. Lepora, Tony J. Prescott and Charles W. Fox

***Abstract*— Tactile sensing can be an important source of information for robots, and texture discrimination in particular is useful in object recognition and terrain identification. Whisker based tactile sensing has recently been shown to be a promising approach for mobile robots, using simple sensors and many classification approaches. However these approaches have often been tested in limited environments, and have not been compared against one another in a controlled way. A wide range of whisker-object contact poses are possible on a mobile robot, and the effect such contact variability has on sensing has not been properly investigated. We present a novel, carefully controlled study of simple surface texture classifiers on a large set of varied pose conditions that mimic those encountered by mobile robots. Namely, single brief whisker contacts with textured surfaces at a range of surface orientations and contact speeds. Results show that different classifiers are appropriate for different settings, with spectral template and feature based approaches performing best in surface texture, and contact speed estimation, respectively. The results may be used to inform selection of classifiers in tasks such as tactile SLAM.**

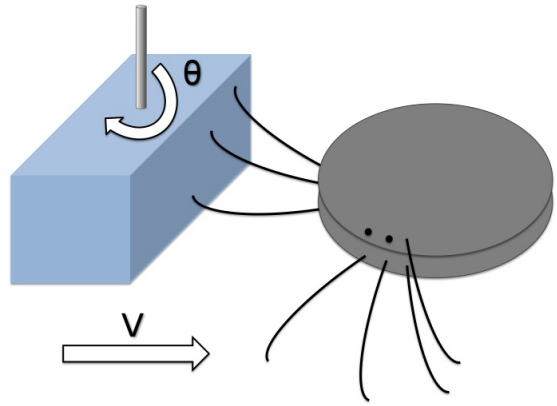


Fig. 1. Whisker pose geometry. A textured surface (blue) is presented to the whiskered agent (grey). Contact speed V and surface angle θ are unknown, but affect texture discrimination.

I. Introduction

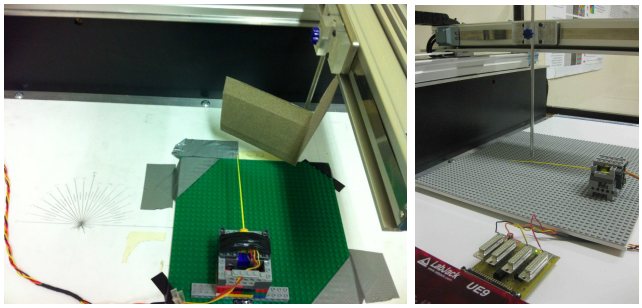
Tactile information can provide an agent with reports of object location and identity. Whisker sensors have some unique advantages over more complex sensors such as fingertips, including mechanical simplicity, and cost [21], and rodents provide proof that excellent tactile sensing is possible using only whiskers [3]. Simple shape recognition from whiskers is possible with simple features [5], [11] or static beam equations [18]. Tactile surface identification remains an active research topic, broadly focussing on the two tasks of terrain identification – where the surface is continuous [15],[19], and object discrimination – where each contact with the surface is discrete (such as in whisking robots [23]). Texture identification for object discrimination would be useful in robot tasks such as tactile SLAM [4], [12]. Many machine learning approaches have been used to establish some benchmarks for texture discrimination in constrained, idealised poses, with both artificial fingertips [2] and whisker-like sensors [15]. However these have been performed in isolation in unique whisker-object contact settings, and have not been compared against one another in an independent controlled study for more realistic poses.

We define a pose as the angle with which a whisker makes contact with a textured surface, and the speed that whisker is moving as contact it made. The way a whisker contacts a surface affects the deflection signals. It has been shown previously that texture discrimination is more difficult when whisker pose is unknown or variable [9] as is the case for contacts made by mobile robots in unknown environments. Surface angle is known to broadly affect texture discrimination [13], but has not been investigated systematically or classified from single whisker contacts (though multi-whisker template classifiers have successfully discriminated surface angles on a mobile robot [12]).

Contact speed is uncertain in many cases, including uncertain robot body velocity from odometry noise; uncertain angular whisker velocity during active whisking (the oscillatory whisker movements generated by rats to explore environments, and control the duration and force of contacts [3]); and the possibility of contact with moving objects in the environment. While other studies [5],[22] have considered contact distance along the whisker shaft as a further variable, we have found in practice [12] that almost all real-world contacts for mobile robots occur at the tip of the whisker, and the few that do not can be used to reposition the robot to obtain a tip contact [11].

To address this problem of discriminating textures with brief contacts at an uncertain whisker pose we have developed an artificial whisker and XY positioning robot system to generate large datasets and comprehensively explore the whisker-object contact parameter space. This system allows us to understand how different contact parameters that vary with mobile robot whisker pose interact to affect whisker deflections. Using this system we compare three existing

M.H. Evans, N.F. Lepora, T.J. Prescott and C.W.Fox are with the Sheffield Centre for Robotics (SCentRo), University of Sheffield, Sheffield, S10 2TN, U.K. (e-mail: mat.evans@shef.ac.uk; n.lepora@sheffield.ac.uk; t.j.prescott@shef.ac.uk; charles.fox@gmail.com). M.J. Pearson is with the Bristol Robotics Laboratory, Bristol, BS16 1QD, U.K. (e-mail: martin.pearson@brl.ac.uk). The authors would like to thank the EPSRC Doctoral Training Scheme and EU FP-7 grants BIOTACT (ICT-215910) and EFAA (ICT-270490) for funding the work.



(a) (b)

Fig. 2. The XY positioning robot (a) from above, to show the range of movement available. Textured surfaces were fixed to the angled corner stimulus and presented to the whisker at different angles and speeds. (b) The robot from the side.

texture, whisker pose and contact speed classification methods: template based classification [6], spectral template based classification [7] and feature based classification [16], [13], [5]. These methods were chosen for their recent track record in classifying whisker data, and accuracy when only a small training set is available. The remainder of this paper describes the XY positioning robot data collection procedure, then specifies the classifiers. Results for each classifier are detailed before being discussed with a view to interpretation and mobile robot applications.

II. Methods

A. Artificial whisker and XY positioning robot

A tapered, flexible plastic whisker [20],[7] ∼5 times scale models of a rat whisker, was mounted into an inflexible rubber-filled 'follicle' case. A tri-axis hall effect sensor mounted in the follicle case measures the deflection of a magnet fixed to the base of the whisker shaft. The hall effect sensor IC was programmed to generate two voltages, corresponding to the magnitude of the whisker base deflection in two directions, *x* and *y*. An XY-table (Yamaha-PXYX, Yamaha Robotics) was used (Figure 2), having a movement range of 350 × 650mm, and able to move up to 720mm/s. Repeatability of the robot is ±0.01mm, and the maximum load it can carry is 1.5kg. Objects are carried by the robot into the artificial whisker, which is fixed to the table as this allows us to control the contact as carefully as possible. Moving the whisker into an object would cause the whisker to oscillate unpredictably during movement between contacts, and as a result each contact would be slightly different. A controller (Yamaha RCX 222, 2-axis robot controller) takes instructions from a PC through an RS232 cable, and the controller interprets the instructions, completes path integration, and drives the motors. Instructions for the robot are generated inside a MATLAB (`www.mathworks.com`) loop, and can be easily updated during robot operation, depending on the whisker input.

B. Data collection for angle, speed and texture estimation

A right angled corner stimulus was designed for presenting textures to the whisker on the XY positioning robot. This textured object is shown in Figure 2 (a). The stimulus consists of an angled strip of aluminium suspended from the XY positioning robot on an pole. Textured sandpapers were affixed to the aluminium surface using double sided adhesive tape. A range of contact speeds was chosen to limit any damage that could occur to the whisker from the large stimulus during high velocity impacts. 6mm/s was the lower bound on speed, with an upper bound of ≈106mm/s, with an interval of ≈7mm/s, providing 11 different speeds. Surface angle ranged from 10°–80°, in increments of 10°. When the stimulus was angled at 0° or 90° the contact was equivalent to a point contact along the shaft so was not considered a tip–surface contact for this experiment. Four textures were chosen, three grades of sandpaper (P80, P180, and P600) and a smooth aluminium surface. Eleven speeds, eight angles and four textures results in 352 different combinations. Contact combinations were randomly interleaved during data acquisition to limit any effects of changing whisker properties during the trial. For each contact combination the whisker was deflected by the robot in both a clockwise and anticlockwise directions (-ve and +ve in *x*), ensuring that the whisker did not bend over time through repeated unilateral deflections. The experiment was performed twice to generate sufficient data for classification.

A simple biomimetic robot control algorithm was implemented to restrict whisker object contacts to brief, light touches. The robot was programmed to move an object into the whisker at a given speed until a deflection threshold was crossed, at which point the robot retracted the object as fast as possible (720mm/s). Temporal latency for the loop is ≈300ms from initial contact due to the controller duty cycle. Whisker deflections were processed at 4kHz. Data from each trial was stored separately. Deflections from the clockwise robot movement trials (-ve in *x*) were converted so all data samples were equivalent. Trials were ordered into arrays by robot movement direction, contact speed, surface angle and texture. Each trial was aligned to peak deflection, and shortened to only the 6000 samples either side of peak deflection (1.5s).

C. Classifier specification

Data was separated into training and test sets that were each complete data sets of 8 angles, 11 speeds, and 4 textures (a total of 2,112,000 samples per set). Signals were placed in the training or test sets at random from the original data. In each case classifiers were developed on the training sets, and performance was determined on the test set.

1) Time–domain template based classification for angle, speed and texture discrimination: An input signal is stored as a template, then compared to new data from the test set. During the test phase, trials were taken at random from the test set as inputs to the classifier. An element-wise sum of squared errors calculation was made between the input I and each template T_i,

$$e(I, T_i) = \sum_{t=1}^{n} (I(t) - T_i(t))^2, \tag{1}$$

where n is the length of the template in samples. The template with the lowest sum of squared error was determined the winner, and a recording was made in an output array of the estimated speed and radial distance to contact of the input trial.

2) Spectral template based classification for angle, speed and texture discrimination: A fast Fourier transform (FFT) was performed on the filtered data in MATLAB, The MATLAB FFT function returns the discrete Fourier transform (DFT) of the input signal (x) of length N, computed with a fast Fourier transform (FFT) algorithm,

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}, \tag{2}$$

where $\omega_N = e^{(-2\pi i)/N}$ is the Nth root of unity.
The absolute of the DFT is then stored as a template for comparison with DFTs of incoming test data as in the template classifier described above.

3) Feature based classification for angle, speed and texture discrimination: Inspection of the data showed that peak deflection magnitude could be used as a feature for speed discrimination under conditions of varying contact angle and texture. Deflection magnitude was taken as the Hall effect sensor output voltage at peak deflection, which is proportional to the bending moment M. Feature f_1 can be defined as,

$$f_1 = \max_{\mathrm{t}} \boldsymbol{M}(t), \tag{3}$$

where $M(t)$ is the deflection magnitude varying with time, measured by the Hall effect sensor in volts. Note that $t(f_1) = t(\max_{\mathrm{t}} \boldsymbol{M}(t))$

In previous preliminary work [8] it was shown that the latency to peak or slope of the initial whisker deflection could be used as a measure of surface angle. To test this idea comprehensively we implemented the same classifier here. Taking the amplitude of deflection f_1, and the time of peak deflection $t(f_1)$, we can find feature f_2 the slope of deflection,

$$f_2 = \frac{f_1}{t(f_1)}. \tag{4}$$

To perform a feature based classification for texture discrimination, centroid energy features such as those described in [16] and [13] were extracted from the signal. After the data had been IIR (infinite impulse response) notch filtered to remove the resonant frequency of the modulation centroid, f_3, was taken as the frequency with most energy in the DFT (X, see equation 2) of the signal x after $X_{1-5\mathrm{Hz}}$ is set to zero,

$$f_3 = \arg\max\ \mathrm{abs}(\mathrm{X}_{[6\mathrm{Hz}-1\mathrm{kHz}]}). \tag{5}$$

The modulation energy, f_4, was defined as the energy of the DFT at the modulation centroid (the magnitude of energy at f_3),

$$f_4 = \max\ \mathrm{abs}(\mathrm{X}_{[6\mathrm{Hz}-1\mathrm{kHz}]}). \tag{6}$$

Finally a total power feature, f_5, was defined as the total energy in the FFT below 1kHz,

$$f_5 = \sum_{6\mathrm{Hz}}^{1\mathrm{kHz}} |X|, \tag{7}$$

where FFT(1:5Hz) were set to 0.

In both the contact pose geometry and frequency case a model was generated of the relationship between each feature and the corresponding contact property with regression, (using the `polyfit` toolbox in MATLAB `http://bit.ly/polyfitn`). Using linear least squares a model is generated that can be used to classify new data. Three arguments are required for generating the model, a vector of independent variable values, a vector of dependent variable values, and a model specification, namely the degree of the polynomial. A second degree polynomial was chosen as preliminary studies showed it provided good results. The independent variables for contact pose geometry were features f_1 for contact speed, f_2 for surface angle, f_3, f_4 and f_5 for surface texture.

An additional combined feature classifier was built to see if combining the texture features f_3, f_4 and f_5 into a single model could improve classification. Polynomial regression was used to develop a model comprising all three texture features, as above. A fourth degree polynomial was chosen as preliminary studies showed it provided good results. After the angle, speed and texture of a test data file is estimated with the model the output is rounded to the nearest integer to make a classification.

4) A note on the analysis: Results are presented for each classifier in turn for surface angle, speed and texture estimation. The performance is compared by calculating the accuracy and precision of angle, speed and texture classification by the reporting of mean and standard errors for classification. Confusion matrices are given for each classifier, as imaged arrays only to give a visual indication of classifier performance.

Cohen's Kappa κ [1] is given as a summary statistic, providing a measure of classification accuracy that is scaled for the number of classes involved. Formally κ is defined in terms of total accuracy (hit rate or P_O, O for observed value), and chance performance P_E (E for expected value) by,

$$\kappa = \frac{P_O - P_E}{1 - P_E}. \tag{8}$$

In this way performance can be compared for each of the contact parameters even though they are over different classification ranges [10]. The value will fall between 0 and 1, with 1 indicating perfect classification and 0 indicating chance performance. Scores below 0.2 are deemed poor classification. In a 2 choice discrimination, a κ of 0.5 would be equivalent to a hit rate of 75% (50% = chance).

III. Results

Typical whisker deflections for each condition are shown in Figure 3.

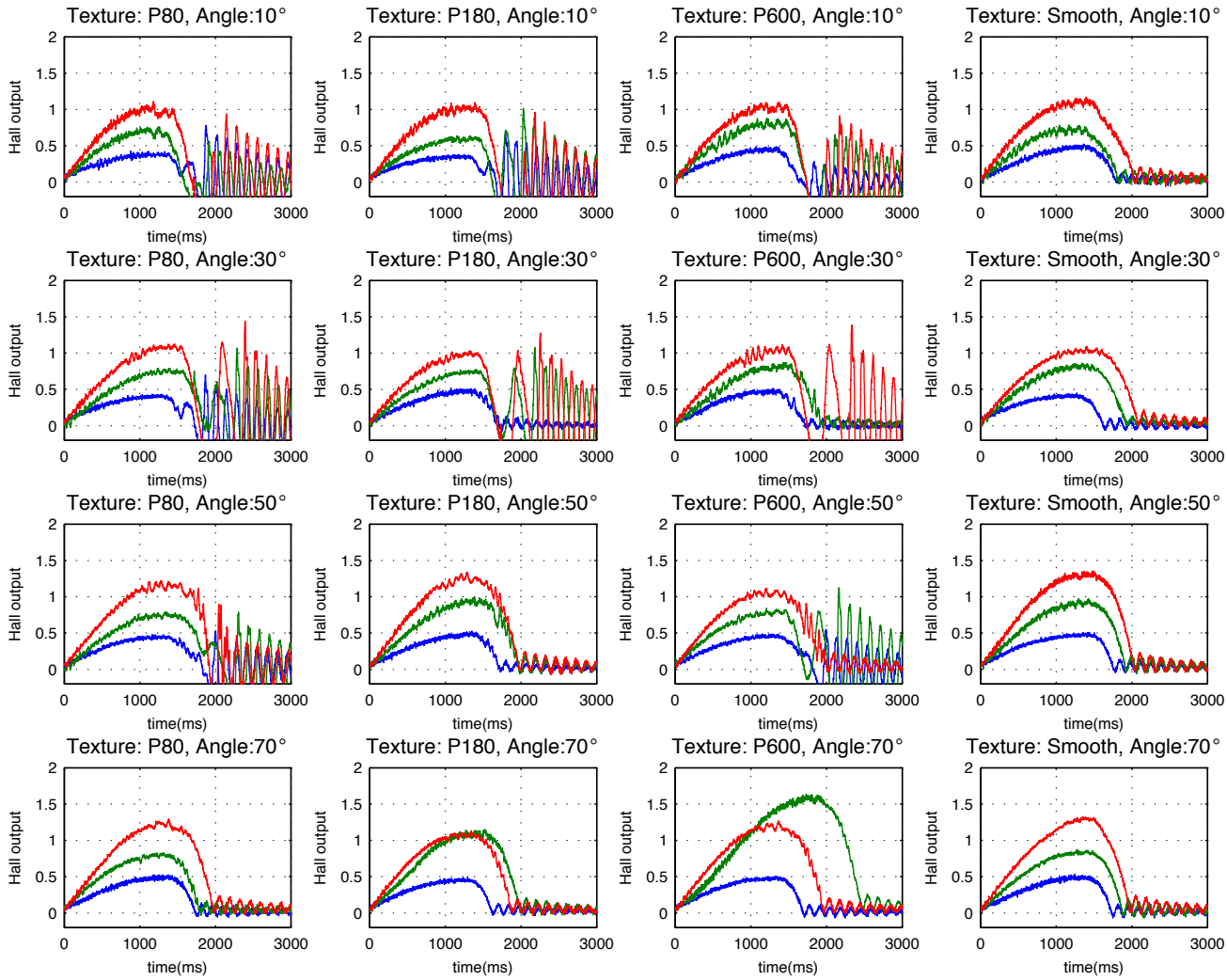


Fig. 3. Typical whisker deflections from the artificial whisker, for different angles, speeds and textures. Texture varies by column, angle varies by row. Colour indicates contact speed, blue = 36mm/s, green = 72mm/s, red = 108mm/s. Note large differences in the amount of oscillatory ringing in some trials.

A. Template based classification of angle, speed and texture

Figure 4 shows the confusion matrices for simultaneous angle, speed and texture classification using a template classifier on raw data. Cohen's κ for each parameter: Angle = 0.27, Speed = 0.25, Texture = 0.24. Cohen's κ for texture independent of contact pose = 0.24. Mean and standard errors for classification, when classified simultaneously: Angle, mean (μ) = 0.71°, standard error (σ) = 15.6°; Speed, μ = -2.7mm/s, σ = 14.22mm/s; Texture, μ = -0.02, σ = 1.23.

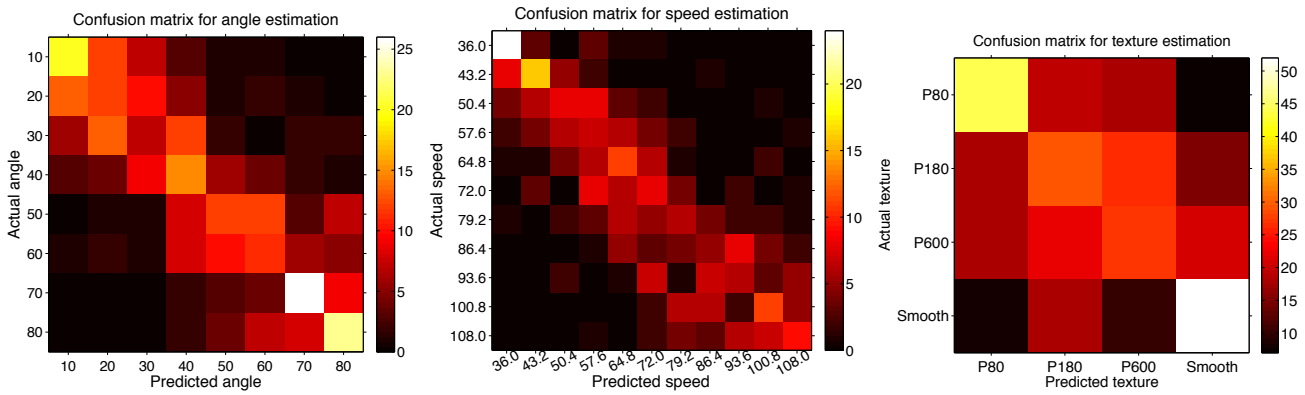
B. Spectral template based classifier results for angle, speed and texture discrimination

Figure 5 shows the confusion matrices for simultaneous angle, speed and texture classification using a spectral template classifier. Cohen's κ for each parameter: Angle = 0.44, Speed = 0.14, Texture = 0.46. Cohen's κ for texture independent of contact pose = 0.42. Mean and standard errors for classification, when classified simultaneously: Angle, mean (μ) = -1.08°, standard error (σ) = 13.8°; Speed, μ = -2.78mm/s, σ = 22.4mm/s; Texture, μ = 0.14, σ = 1.00.

C. Feature based classification of angle, speed and texture

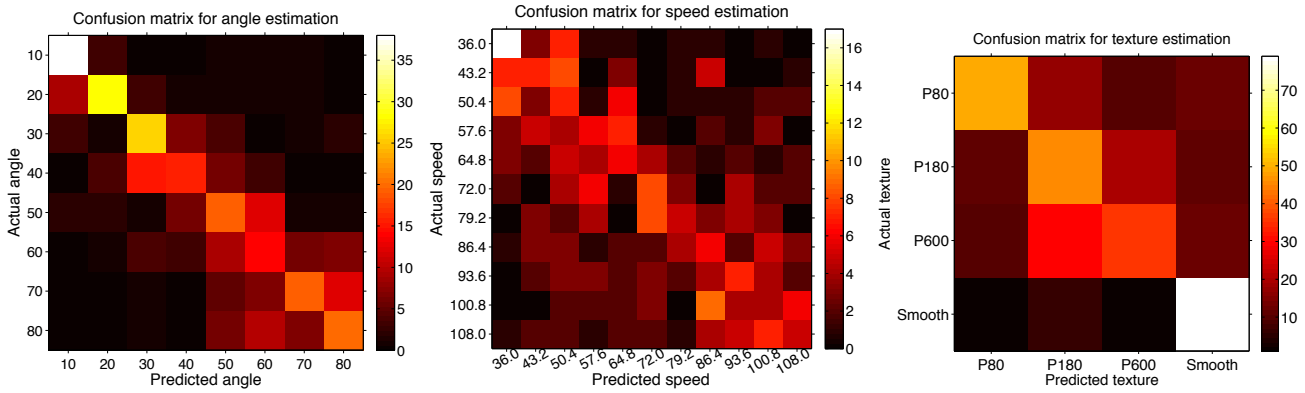
Figure 6 shows the confusion matrices for feature based speed and angle classification using features f_1 (magnitude of deflection), f_2 (slope of deflection), and texture classification using features f_3 (modulation centroid), f_4 (centroid energy) and f_5 (total energy). Cohen's κ for each parameter; angle = 0.11, speed = 0.43, texture (f_3) = 0.06, texture (f_4) = 0.03, texture (f_5) = 0.15. Cohen's κ for each parameter independent of each other; angle = 0.3, speed = 0.25, texture (f_3) = 0.08, texture (f_4) = 0.21, texture (f_5) = 0.27. Cohen's κ for texture discrimination by combining features with multinomial regression was 0.12.

Table I shows the mean and standard error for classification with each feature, when classified simultaneously.



(a) (b) (c)

Fig. 4. Confusion matrices for classification of angle (a), speed (b), and texture (c) with the raw signal template classifier over all contacts. Scale on the right indicates correct classifications. Brightness indicates larger numbers, and better classification



(a) (b) (c)

Fig. 5. Confusion matrix for classification of texture with a spectral template based classification. Scale on the right indicates correct classifications. Brightness indicates larger numbers, and better classification.

TABLE I

MEAN AND STANDARD DEVIATIONS OF CLASSIFICATION ERROR FOR ALL FIVE FEATURES.

	f_1	f_2	f_3	f_4	f_5	Combined
Mean	1.80mm/s	0.17°	0.43	-0.07	0.05	-0.09
Std Err	12.12mm/s	18.6°	1.10	1.22	1.09	0.97

D. Comparing the classifiers

Figure 8 compares Cohen's κ for the best version of each method, namely the low pass filtered template, the frequency template, a 'best case' feature classifier (f_1 for speed, f_2 for angle and f_5 for texture). Mean and standard errors for each method are compared in Figure 7.

IV. DISCUSSION

We have presented the first results for whiskered texture classification in a controlled classifier comparison study, under varying contact pose (varying surface angle and speed). These variations occur in real mobile robot settings, where the classifications would be useful for tasks such as object detection for tactile SLAM [12]. We found that classification results vary a great deal across the different methods, with classifiers performing better for some parameters than for others. No method performed well across all conditions. This was not apparent in the original presentations of the classifiers [16], [13], [7], [6]. Figure 8 compares the results for each classifier using Cohen's κ. The best classifier for angle estimation was spectral templates. The same spectral template classifier was also most successful in texture discrimination. Spectral templates have been successful at whisker based texture discrimination in the past [7], therefore it is no surprise that it is successful here. Surface angle was also successfully classified by the spectral template classifier, which may be unexpected. Speed classification was most accurate with the feature based classifier, with all other methods performing poorly. Again this may be expected in light of the success had with feature based speed discrimination [5]. This split in classifier success between different methods for different parameters may indicate that the salient features in the signals for texture and angle vary together, while speed discrimination is somehow orthogonal. More work needs to be done to develop a clearer understanding of the

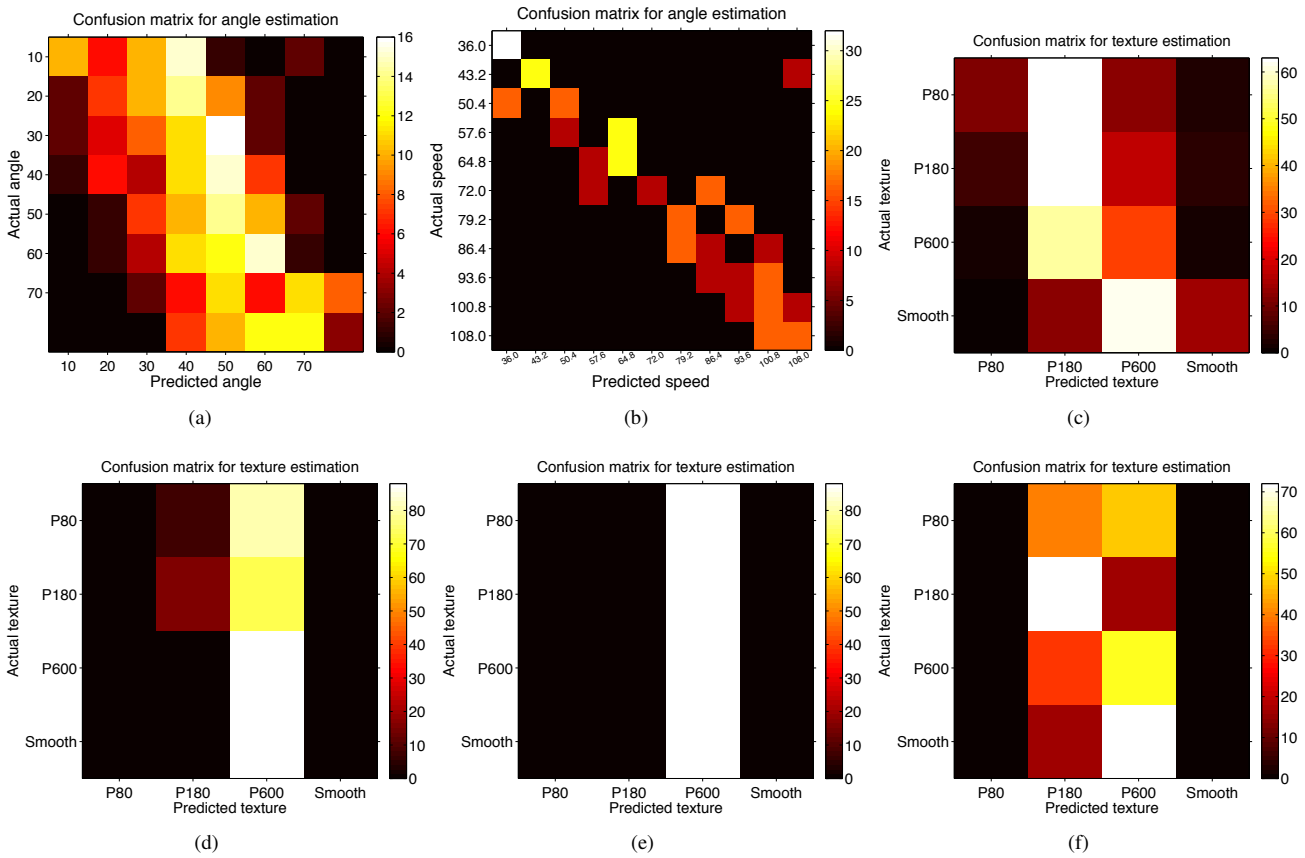


Fig. 6. Confusion matrices for classification with the feature based classifier; of angle with f_2 (a), speed with f_1 (b), and texture with f_3 (d), f_4 (e) and f_5 (f). (c) Confusion matrix for classification of texture with the multinomial feature based classifier, using features f_3, f_4 and f_5 over all contacts. Brightness indicates larger numbers, and better classification

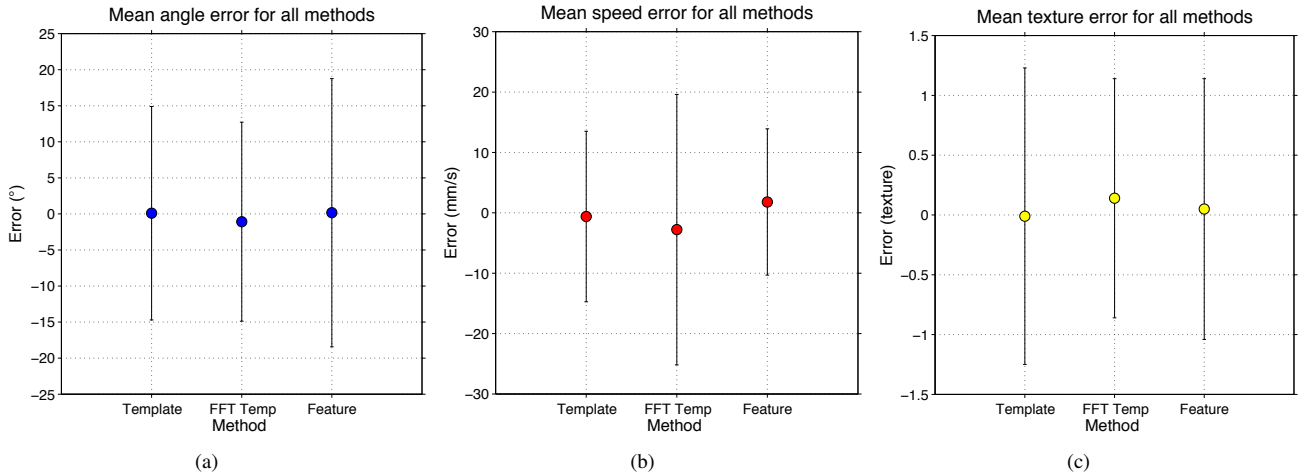


Fig. 7. Mean and standard error of classification for the best version of each method for angle (a), speed (b) and texture (c). Template: raw data template; FFT Temp: spectral template; Feat: a 'best case' feature classifier (f_1 for speed, f_2 for angle and f_5 for texture)

way surface angles and textures, and contact speeds affect whisker deflections and how these effects interact. The XY positioning robot system will be central to these efforts.

There appeared to be no advantage to classifying texture independently of the contact pose parameters, and for some classifications Cohen's κ remains below 0.2, and is a poor classification (0 is chance performance), indicating that many of the features and classifiers presented here are not succeeding in classifying the data. In previous studies it has been shown that information about the whisker–object

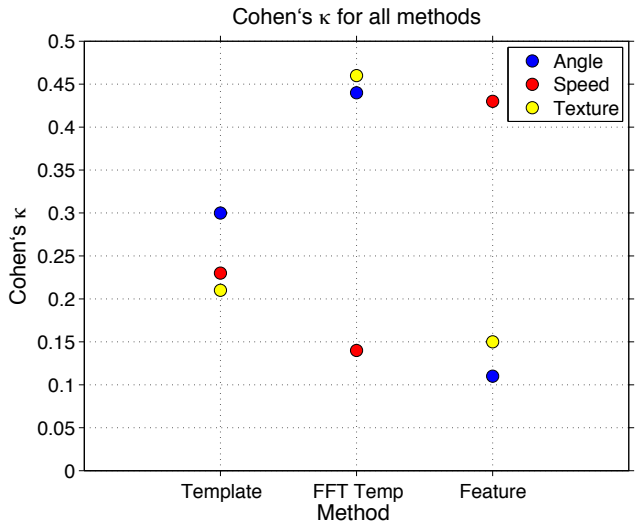


Fig. 8. Cohen's κ for each method for classification of surface angle (blue), object speed (red) and surface texture (yellow). Template: raw data template; FFT Temp: spectral template; Feat: a 'best case' feature classifier (f_1 for speed, f_2 for angle and f_5 for texture). Note contrasting classifier performance across parameters.

contact pose geometry can improve whisker sensing [9],[13]. It is difficult to determine why this has not been the case here. It may be that better classifiers could be built, or more successful features could be extracted, for example using a machine learning feature extraction approach such as PCA. In previous studies a Gaussian classifier was used for feature based texture discrimination [13]. This kind of classifier may be more effective than a polynomial at combining information from a number of features, or for discriminating properties that interact.In addition, larger data sets may be useful for training classifiers to be robust to small within–trial variations. Inspection of the raw signals in Figure 3 shows oscillatory ringing after contact on some trials but not others. This ringing does not seem to be associated with a particular contact parameter, but may be due to the whisker tip interacting with a surface uniquely on different trials. It may be that trying to classify so many parameters effectively and at the same time may to too difficult a task for one whisker making a single brief contact. This may be the limit of single whisker sensing, and improvements would no doubt be gained through pooling of information across whiskers. For example using multiple-whisker classifiers, as used previously for surface angle discrimination on a mobile robot [12].

It may be possible to combine classifiers such as those presented here in a mixture of experts [17], or with boosting. Probabilistic methods, some of which have been successfully applied to whisker sensing such as Bayesian blackboards [14] and sequential analysis [19], are ideally suited to optimally combining information from a number of sources. In this way many low-resolution reports of touch events can be combined across whiskers and over time to make high-level inferences about objects and surfaces in the environment.

REFERENCES

[1] J. Cohen et al. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[2] S. Decherchi, P. Gastaldo, R.S. Dahiya, M. Valle, and R. Zunino. Tactile-data classification of contact materials using computational intelligence. *IEEE Transactions on Robotics*, 27(3):635–639, 2011.

[3] M.E. Diamond, M. von Heimendahl, P.M. Knutsen, D. Kleinfeld, and E. Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008.

[4] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[5] M. H. Evans, C. W. Fox, N.F. Lepora, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proceedings IEEE ROBIO*, 2010.

[6] M. H. Evans, C. W. Fox, M. J. Pearson, and T. J Prescott. Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. In *Proceedings of the 11th International Conference on Simulation of Adaptive Behaviour. (SAB 2010) From Animals to Animats.*, 2010.

[7] M. H. Evans, C. W. Fox, and T. J. Pearson, M. J.and Prescott. Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In T Kyriacou, U. Nehmzow, C. Melhuish, and M. Witkowski, editors, *Proceedings of Towards Autonomous Robotic Systems (TAROS 2009)*, pages 19–24, 2009.

[8] M. H. Evans, C.W. Fox, M. J. Pearson, and T. J. Prescott. Object location, orientation, and velocity extraction from artificial vibrissal signals. In *Society for Neuroscience Abstracts. Society for Neuroscience (Program No. 174.8/ Z12)*, 2009.

[9] M. Fend. Whisker-based texture discrimination on a mobile robot. *Advances in Artificial Life*, pages 302–311, 2005.

[10] A.D. Forbes. Classification-algorithm evaluation: Five performance measures based onconfusion matrices. *Journal of Clinical Monitoring and Computing*, 11(3):189–206, 1995.

[11] C. W. Fox, M. H. Evans, N. F. Lepora, M. J. Pearson, A. Ham, and T. J. Prescott. Crunchbot: a mobile whiskered robot platform. In *Proceedings of Towards Autonomous Robots, Springer.*, 2011.

[12] C. W. Fox, M. H. Evans, M. J. Pearson, and T. J. Prescott. Tactile SLAM with a whiskered robot. Proceedings ICRA 2012, 2012.

[13] C. W. Fox, B. Mitchinson, M. J. Pearson, A. G. Pipe, and T. J. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.

[14] C. W. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. In *Proceedings of Towards Autonomous Robotic Systems (Springer)*, 2011.

[15] P. Giguere and G. Dudek. A simple tactile probe for surface identification by mobile robots. *Robotics, IEEE Transactions on*, 27(3):534–544, 2011.

[16] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, and P. Konig. Texture signals in whisker vibrations. *J Neurophysiol*, 95(3):1792–1799, 2006.

[17] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[18] Makoto Kaneko. Active antenna. *Robotics and Automation, IEEE Transactions on*, pages 2665–2671, 1994.

[19] N.F. Lepora, C.W. Fox, M.H. Evans, M.E. Diamond, K. Gurney, and T.J. Prescott. Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination. *Journal of The Royal Society Interface*, 2012.

[20] M. J. Pearson, B. Mitchinson, J. Welsby, A. G. Pipe, and T. J. Prescott. Scratchbot: Active tactile sensing in a whiskered mobile robot. *From Animals to Animats SAB2010.*, 2010.

[21] T.J. Prescott, M.J. Pearson, B. Mitchinson, J.C.W. Sullivan, and A.G. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.

[22] J.H. Solomon and M.J.Z. Hartmann. Radial distance determination in the rat vibrissal system and the effects of weber's law. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581):3049–3057, 2011.

[23] J.C. Sullivan, B. Mitchinson, M.J. Pearson, M. Evans, N.F. Lepora, C.W. Fox, C. Melhuish, and T.J. Prescott. Tactile discrimination using active whisker sensors. *Sensors Journal, IEEE*, 12(2):350 –362, February 2012.

Towards hierarchical blackboard mapping on a whiskered robot

C.W. Fox[a,*], M. H. Evans[a,*], M. J. Pearson[b], T.J. Prescott[a]

[a]*Sheffield Centre for Robotics, University of Sheffield, Western Bank, Sheffield, S10 2TF, UK*
[b]*Bristol Robotics Laboratory, Bristol, UK*

Abstract

The paradigm case for robotic mapping assumes large quantities of sensory information which allow the use of relatively weak priors. In contrast, the present study considers the mapping problem for a mobile robot, CrunchBot, where only sparse, local tactile information from whisker sensors is available. To compensate for such weak likelihood information, we make use of low-level signal processing and strong hierarchical object priors. Hierarchical models were popular in classical blackboard systems but are here applied in a Bayesian setting as a mapping algorithm. The hierarchical models require reports of whisker distance to contact and of surface orientation at contact, and we demonstrate that this information can be retrieved by classifiers from strain data collected by CrunchBot's physical whiskers. We then provide a demonstration in simulation of how this information can be used to build maps (but not yet full SLAM) in an zero-odometry-noise environment containing walls and table-like hierarchical objects.

1. Introduction

Touch-based mapping has two principal applications. Firstly, as a sole sensory system in environments where other types of sensors fail, such as smoky or dusty search-and-rescue sites, especially where covert (no signal emission) operation is required. Secondly, as a complement to other sensors such as vision, with which it can be fused or used as a 'last resort' during adverse conditions as in the sole sensor case.

However, the paradigm case for robotic mapping, as in Simultaneous Localisation and Mapping (SLAM) problems [50], instead considers a mobile robot with noisy odometry and vision or laser scanners. Vision and laser scanners provide large amounts of sensory information, and have effectively unlimited range in indoor environments. Such large quantities of input information allow the use of relatively weak priors, such as independent grid cell occupancy or flat priors over the belief of small feature sets [50].

This study considers the touch-based mapping problem in which only sparse, local sensory information is available. Proof that navigation from such sensors is possible is readily found in biology: rats navigate through dark underground tunnels using their whiskers [7, 2], having ranges of only a few centimetres. In robotics, whisker sensors are relatively cheap in both material and computational processing terms, and their use has previously been considered in constrained tasks [44, 43, 30, 29, 15]. The previous robotic attempts at mapping from sparse local sensors have either used the extremely strong generic prior that the whole world is made entirely of north-south and east-west straight edges [53] or have used relatively long range but sparse ray sensors integrated over multiscans [3].

We will demonstrate touch-based mapping using a mobile robot, CrunchBot, having six whisker touch sensors only. First, it is shown that CrunchBot's whiskers are able to recover approximate position and orientation reports about contacts with surfaces. Then it is shown how these reports can be fused with strong priors to recognise hierarchical objects such as tables and chairs, as a step in building a map of the environment.

Fig. 1 gives an overview of the general framework for perception and navigation with whiskers within which this study operates. When biomimetically inspired by rodents, whisker sensors have strain sensors at their base only. When a rat investigates an object it palpates the surface in a back and forth oscillatory sweeping behaviour known as 'whisking' [52],[9]. It is thought that whisking is important for gathering the most reliable signals from whisker contacts [36]. Straight whiskers can make two distinct types of contact with an object,

*These authors contributed equally and are joint first authors
Email addresses: charles.fox@sheffield.ac.uk (C.W. Fox), mat.evans.sheffield.ac.uk (M. H. Evans), martin.pearson@brl.ac.uk (M. J. Pearson), t.j.prescott@sheffield.ac.uk (T.J. Prescott)

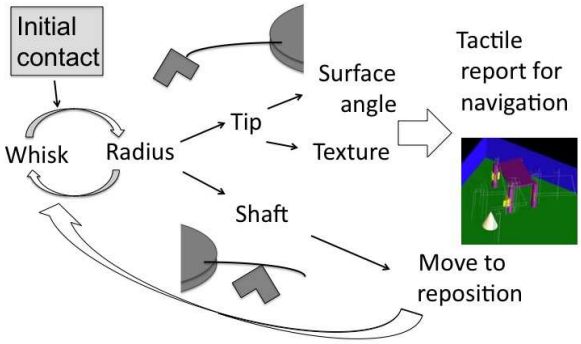


Figure 1: A new framework for extracting contact parameters. After initial contact a whisk behaviour allows the discrimination of object location. If contact is made along the whisker shaft the agent must move to reposition the whisker for subsequent contacts. If contact is made at the whisker tip a robust discrimination of surface properties can be made. Reports of surface properties can be used by other systems, such as for navigation or to construct complex object models as in the present study.

contacting it either at their tip or their shaft. Tip contacts are generally the most useful, because they provide a standardised, constrained setting (i.e. with the contact point at a known location at precisely the end of the whisker) from which surface properties such as orientation and texture can be identified [30],[15]. In contrast, shaft contacts are less informative. For example, an unknown distance to an object along the shaft can confuse attempts to classify surface orientation and texture [17]. Shaft contacts are rare in practice in both rodents and mobile robots, occurring only when small objects enter the field of multi-whisker arrays between the whisker tip points. In the scheme used here, a feature based radial distance estimator [13] is first used to make a decision of whether the contact is at the tip or the shaft. If it is a shaft contact, then the robot should use the radial distance information to move to another location that is likely to yield a more useful tip contact. Following a tip contact, we can read surface orientation and texture information (and possibly speed of object when there are moving objects in the world) and pass them as an observation to a navigation or mapping system.

This study provide an implementation of the distance and angle stages of this framework on CrunchBot (Fig.2(a)). Individual components of such a system have previously been investigated in isolation, including whiskered texture recognition [15],[26],[11], [17],[32], surface shape recognition [30],[24],[21],[13], and object recognition [19]. These components have previously been tested under ideal laboratory conditions or in individual mobile settings [41]; here we present steps integrating them into a single platform for hierarchical object recognition, along with results and observations on their performance 'in the wild' in a common arena environment.

To compensate for the sparseness of the sensory information available from these distance-orientation reports, we fuse them with strong hierarchical priors about objects in the world. Hierarchical object recognition models were popular in classical AI in the guise of 'blackboard systems' [10, 35, 5] but have recently been recast in terms of dynamically constructed Bayesian networks [16, 34, 31, 49]. Here we provide an application of Bayesian blackboards to robotic mapping. We do not consider the full SLAM problem here, but instead work in a simulation of CrunchBot having zero odometry noise to avoid the localisation problem and focus on mapping only. Related object-based mapping models have recently appeared [51, 23, 46, 40] using laser sensors to recognise and learn complex but non-hierarchical spatial models. However as data available through whiskers to CrunchBot is much sparser than that from laser scanners, the required level of sensor detail is unavailable, therefore we compensate with the new mapping technique of fusing contact reports into *hierarchical* models. For example, on recognising a single table leg, we may infer the probable presence the rest of the table, including other leg objects, and edges and corners making up these legs, without ever sensing them directly. To construct hierarchical objects, we use hypothesis priming and pruning heuristics as in classical blackboard systems. However, following [16], we treat such heuristics as approximations to inference in a dynamically-constructed, Monte Carlo Markov Chain (MCMC) sampling Bayesian network, whose observations are the distance-orientation reports from the whiskers.

2. Methods

2.1. Whiskers.

CrunchBot's six whiskers measure 160mm in length, 1.45mm diameter at the base tapering linearly to 0.3mm at the tip. They are built from nanocure25 using an Envisiontec rapid prototyping machine. A magnet is bonded to the base of the whisker and held in place by a plug of polyurethane approximately 0.75 mm above a Melexis 90333 tri-axis Hall effect sensor IC [33]. This sensor generates two outputs representing the direction of the magnetic field (in two axes) with respect to its calibrated resting angle. These two 16-bit values are sampled by

Figure 2: CrunchBot, a whiskered mobile robot platform.

a local dsPIC33f802 micro-controller which, in turn, is collected using an FPGA configured as a bridge to a USB 2.0 interface. Up to 28 whiskers can be connected to this FPGA bridge at one time. Using the vendor provided software driver and API (Cesys GmbH), a user can request the data from all whiskers at minimum intervals of 500μs (a maximum sample rate of 2kHz).

2.2. Robot platform.

CrunchBot is based on the iRobot Create base (`www.irobot.com`) platform, with the whiskers mounted in the cargo bay, being positioned on an adjustable metal bar and rapid prototyped ball joint mountings. These mountings allow adjustment of the whiskers. For data collection experiments in the present study, only four whiskers are used, configured in the horizontal plane to detect objects in an arena (the other two whiskers scrape along the floor and are used in other experiments, such as for texture discrimination in our previous study, [20]). We have also extended the cargo bay mounting to accommodate a netbook PC, which is used for local control of the robot. The netbook runs Ubuntu 10.10 on a single-core Intel Atom processor. A circular buffer in shared memory is used to make data from the Cesys driver available to other processes. The netbook hosts a Player server (`playerstage.sourceforge.net`) which provides high-level, networked API interfacing to the Create's serial port commands. Processes such as texture and shape recognition and basic motor control run on the netbook, reading the raw data from the fast circular shared memory buffer, and writing their results every 0.1s to a Python Pyro server (`pyro.sourceforge.net`) on the remote desktop which runs hierarchical object recognition and mapping.

2.3. Robot movement

Previous work has shown that accurate object localisation with a whisker requires some measure of contact speed [13], or of the applied forces and bending moments at the base of the whisker [24],[30], values that are not always available in the mobile case as agent movement will affect these contact properties. To address these points a 'body whisk' behaviour was included in the robot program. As the whiskers were not actuated the whole robot must rotate in a systematic way. Upon initial contact with an object the robot first reverses away a short distance before rotating at $\pi/24$ radians per second towards the object, then rotating at $\pi/24$ radians per second away from the object. This allows this whiskers to move over the surface of the contact object, collecting data about its location and orientation. After the whisk the robot reverses again to clear the object, then rotates in a random direction and moves forward again.

2.4. Radial distance reporting

To determine whether an object has made contact with any of CrunchBot's whiskers at the tip or the shaft, and to discriminate between contacts with the surfaces or corners of objects, object localisation was implemented. Previous work [13] has shown that peak deflection magnitude could be used as a feature for radial distance discrimination at a given speed. Whisker data was recorded during the 'body whisk' contact, and the maximum whisker deflection was measured. Deflection magnitude was taken as the Hall effect sensor output voltage at peak deflection, which is proportional to the bending moment. This feature f_1 can be defined as,

$$f_1 = \max_{\mathrm{t}} \theta(\mathrm{t}), \tag{1}$$

where $\theta(t)$ is the time-dependent deflection magnitude measured by the Hall effect sensor.

During the training phase a dataset was collected for each whisker, consisting of 5 contacts at each point along the whisker at 10mm intervals over a 50mm range from the tip of the whisker. Though the whisker is 160mm long, only 140mm is external to the 'follicle'. A model was then generated of the relationship between the deflection magnitude and the corresponding radial distance to contact by fitting a linear equation to the training data in MATLAB. To find an estimate of radial distance r,

$$r = a_1 f_1 + a_0, \tag{2}$$

was fitted to the data with a linear-in-the-parameters regression on the line, giving a least-squares fit for (a_0, a_1) for each whisker.

2.5. *Surface orientation reporting*

A complementary data-driven approach to feature extraction methods is to store instances of time series as a set of templates for comparison to novel data. It has been shown that simple k-means style templates on strain time series from *individual* whiskers can be used for discriminating contact *distance* classes in physical simulation [21], and stationary robot hardware [12]. In the present study we have access to four whiskers together, so we can train templates corresponding to surface orientation classes from the 8-dimensional time series from the whole *multi-whisker* set (four whiskers, each with vertical and horizontal strain channels). Orientation reports could be used to inform complex object models as in sec. 2.6. The rationale for this particular approach is that a template method can utilise bulk data from all whiskers to find surface orientations, without any of the geometric assumptions required when splining individual radial distance reports [30].

Offline training data was collected by programming the robot to drive into a wall at fifteen different angles (20°:160° in 10° intervals) four times. Data was aligned to initial contacts, low pass filtered (17Hz) to remove oscillations caused by robot body movement, recorded for 2s, and smoothed with a five-point moving average. Templates were generated by averaging across three of the four sets for each angle. Templates for each angle comprised data of all eight channels from the four whiskers to allow multi-whisker information to inform classification.

During testing the fourth data set was used compared to the averaged template using a sum of squared error measure. The average squared error, e for each template, T_i is computed over the N logged data points,

$$e(T_i) = \frac{1}{N}\sum_{t=1}^{n}(I(t) - T_i(t))^2. \qquad (3)$$

The template with the lowest sum of squared errors was determined the winner, and the orientation of this template recorded. This process was repeated four times, each time using a different data set as the test set, and a template constructed from the remaining three sets. In this manner it is possible to generate a robust estimate of mean classifier performance, while preserving individual trial differences for inspection in the results.

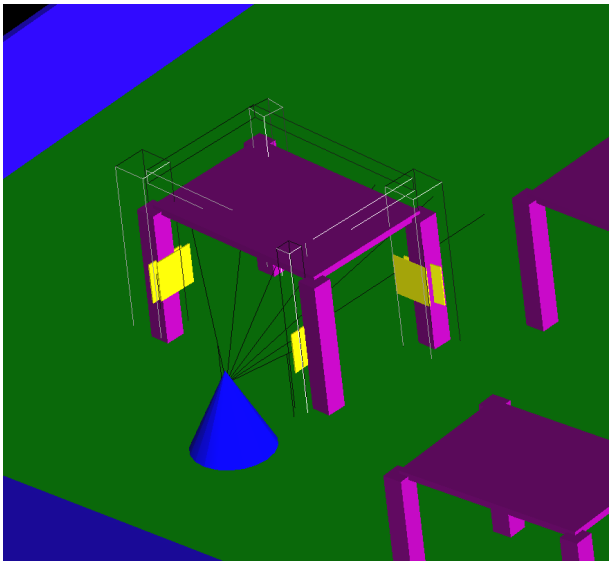


Figure 3: Simulation screen-shot at low annealing temperature. A single table hypothesis remains, aligned correctly with the physical table.

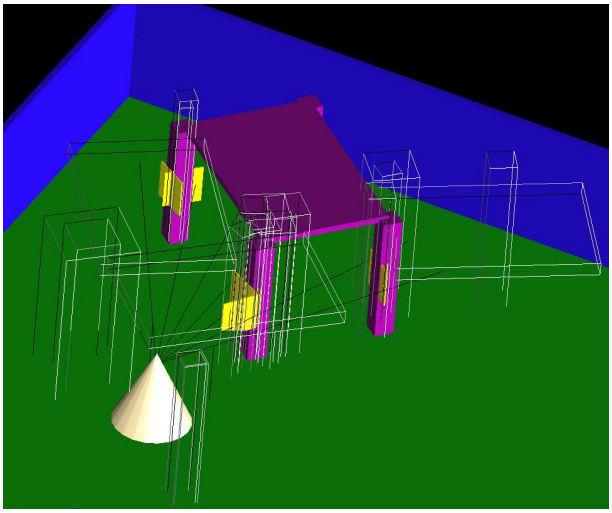


Figure 4: Simulation screen-shot at high annealing temperature. Many hypothesised (wire-frame) tables and legs are on the blackboard, primed by the shapelets (yellow rectangles) contacted by the robot (cone)'s whisker sensors, in an arena containing a physical table (pink).

2.6. Hierarchical Object Models

CrunchBot's task is to build a map of an arena populated by four-legged table-like objects as in figs. 4 and 3. Such objects could include chairs and desks in a home or office environment for example. In our object recognition experiment, CrunchBot moves along a *predetermined* trajectory of location-angle poses, (x^t, y^t, θ^t) around the arena (though we discuss possibilities for autonomous exploration in sec. 2.9), over discrete time steps t. We assume that at each discrete time step t, CrunchBot's whiskers, $w \in 1:6$, each report egocentric estimates of the radial distance r to, and surface orientation ϕ, using the methods of previous sections, and of contact texture[1] τ of, any contacts made,

$$\hat{r}^t_w = r^t_w + \varepsilon_r, \tag{4}$$

$$\hat{\phi}^t_w = \phi^t_w + \varepsilon_\phi, \tag{5}$$

$$\hat{\tau}^t_w = \tau^t_w + \varepsilon_\tau, \tag{6}$$

where ε are i.i.d. Gaussian noises having zero mean and standard deviations $\sigma^w_r, \sigma^w_\phi, \sigma^w_\tau$ respectively. Assuming perfect odometric localisation in the present study, these estimates may be converted into allocentric Cartesian coordinates to give tuples $S(x_S, y_S, \phi_S, \tau_S)$ which we call *shapelets* and which will be treated as observations in graphical models.

Tables, T, are parametrised by tuples, $T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)$, where x, y, θ is the pose, w^x_T and w^y_T are width and breadth, w^L is the width of the (square) legs, and $\tau \in (0, 1)$ is a texture parameter describing roughness or smoothness of the material. A generative model of tables is used. CrunchBot assumes a flat prior probability *density* generating tables in the world,

$$p(T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)|\emptyset) = c_T, \tag{7}$$

where c_T is a (non-normalising) constant.

If a table T exists, its presence causes (in the sense of [38]) the presence of four leg objects,

$$L(x_L, y_L, \theta_L, w_L, \tau_L, T), \tag{8}$$

where w_L is the width of the square table leg; x_L, y_L, θ_L are its location and rotation, and τ_L is its texture, with probability density

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, T)|T(x_T, y_T, \theta_T, w^L_T, \tau_T))$$

[1]Texture reports are not yet implemented on the physical CrunchBot, but can already be handled by our hierarchical framework in simulation so are included for completeness. In the present simulations we assume all tables and reports have the same texture, $\tau = 1$.

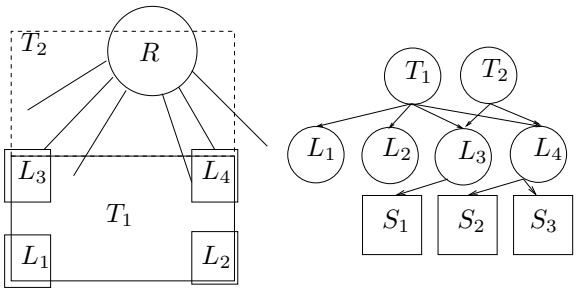


Figure 5: Hierarchical object recognition. *Left:* Robot R (circle) with six whiskers (lines) makes tactile contact with legs L_j (squares) of a hypothesised table T_1 (rectangle). The two contact points ('shapelets') on the right are sufficient to infer the location of the corner of leg L_4. Coupled with prior knowledge about the shape and size of tables, and the third shapelet, this can be used to infer that there is a table either in the ground truth location or in a second configuration T_2 (dashed rectangle). *Right:* Bayesian network constructed to represent the same scenario. Square nodes are the shapelet observations.

$$= \alpha_L \exp\left(-\Delta_{TL}\right), \tag{9}$$

where α is a (non-normalising) constant, and the distance measure is

$$\Delta_{TL} = \min_i \left(\frac{(x^i_T - x_L)^2 + (y^i_T - y_L)^2}{\sigma^2_r}\right) + \left(\frac{\theta_T - \theta_L}{\sigma_\theta}\right)^2 + \left(\frac{w^L_T - w_L}{\sigma_w}\right)^2 + \left(\frac{\tau_T - \tau_L}{\sigma_\tau}\right)^2, \tag{10}$$

where $0 \leq i \leq 3$, and (x^i_T, y^i_T) are the coordinates of the table's four corners, and σ_w, σ_τ are parameters specifing standard deviations of the leg's w_L, τ_L values conditioned on the table's corresponding w^L_T, τ_T values. The inclusion of T in the parametrisation of L (eqn. 8) means that L is the hypothesis that the leg was caused *only* by table T rather than any other table or cause.

Shapelets are assumed to be generated by nearby legs,

$$p(S(x_S, y_S, \theta_S, \tau_S)|L(x_L, y_L, \theta_L, w_L, \tau_L, T)) = \alpha_S \exp\left(-\Delta_{LS}\right) \tag{11}$$

where

$$\Delta_{LS} = \left(\frac{r}{\sigma^S_r}\right)^2 + \left(\frac{f(\theta_L) - \theta_S}{\sigma^S_\theta}\right)^2 + \left(\frac{\tau_L - \tau_S}{\sigma^S_\tau}\right)^2, \tag{12}$$

and r is the shortest radial distance from the perimeter of the leg to (x_S, y_S), computed by basic geometry, $f(\theta_L) = \theta_L + m\pi/2$ picks the angle of the corresponding side m of the leg at this shortest-distance contact point, and $\sigma^S_r, \sigma^S_\theta, \sigma^S_\tau$ model sensor noise.

We also provide small *null priors* to allow legs and shapelets to exist in the absence of any generative parents. (These are required later, during construction on

the blackboard, so that these objects can survive before their parents are constructed),

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, \emptyset)|\emptyset) = c_L, \quad (13)$$

$$p(S(x_S, y_S, \theta_S, \tau_S)|\emptyset) = c_S, \quad (14)$$

with constants such that the marginalised densities,

$$p(S(x_L, y_L, \theta_L)) < p(L(x_L, y_L, \theta_L)) < p(T(x_L, y_L, \theta_L)), \quad (15)$$

i.e. larger objects are more probable to exist without high-level causes than smaller objects are.

Unlike the parametrisation of L on T in eqn. 8, shapelets may be caused by *mixtures* of multiple leg hypotheses and by the null prior (eqn. 14). For example if there are two legs very close together then the density for observing shapelets in the area increases. We assume that multiple causal sources combine using noisy-OR semantics,

$$P(x_i|pa(x_i)) = 1 - \prod_{x_j \in pa(x_i)} (1 - P(x_i|x_j)). \quad (16)$$

where $pa(x_i)$ denotes the set of parents of generic node x_i. As we use probability *density* functions we require the continuous version of noisy-OR, proved below,

Theorem

$$p(x_i|pa(x_i)) = \sum_{x_j \in pa(x_i)} p(x_i|x_j). \quad (17)$$

Proof

Let Y_i range over nodes $pa(X)$ in a continuous-valued Bayesian network with noisy-OR parent combinations,

$$P(X|\{Y_i\}_i) = 1 - \prod(1 - P_i). \quad (18)$$

with $P_i = P(X|Y_i)$. Consider the probability of a small range of hypotheses,

$$\delta^3 p(X|\{Y_i\}_i) = 1 - \prod(1 - \delta^3 p_i), \quad (19)$$

where p are probability densities and P are probabilities. Expansion terms with powers of δ that are > 3 vanish, so

$$\delta^3 p(X|\{Y_i\}_i) = \delta^3 \sum p_i. \quad (20)$$

The δ^3 terms cancel to yield

$$p(X|\{Y_i\}_i) = \sum p_i \quad (21)$$

as required.

We allow legs to be caused by a mixture of their *single* specified parent (i.e. the T parameter in eqn. 9) and null prior (eqn. 13), using a similar combination rule. Tables are caused by the null prior only (eqn. 7).

Taken together, the equations in this section define a Bayesian network for any given collection of tables, legs and shapelets as shown in fig. 5. However, in addition to the previous causal probabilities, we need to model the following constraints: (a) tables always have four legs; (b) each table leg is at a different corner of the table (we should not see two legs attached to the same corner); (c) two objects of the same type (table or leg) cannot overlap in physical space. Standard Bayesian networks cannot model such relations, as they are limited to joint distributions of the form

$$P(\{x_i\}_i) = \prod_i P(x_i|pa(x_i)). \quad (22)$$

To model these additional constrains, we extend the Bayesian network to the factor graph,

$$P(\{x_i\}_i) = \frac{1}{Z}\left(\prod_i P(x_i|pa(x_i))\right) \times \left(\prod_{ij} \phi_c(x_i, x_j)\phi_b(x_i, x_j)\right)\left(\prod_i \phi_a(x_i)\right), \quad (23)$$

where Z is a normalising constant, and ϕ_a, ϕ_b, ϕ_c are unnormalised penalty factors corresponding to the three new constraints. Using superscripts for exponentiation, these are

$$\phi_a(x_i) = \epsilon_a^m, \quad (24)$$

$$\phi_b(x_i, x_j) = \epsilon_b^v, \quad (25)$$

$$\phi_c(x_i, x_j) = \epsilon_c^r, \quad (26)$$

where m is the number of missing legs iff x_i is a table, and $m = 0$ otherwise; v is a Boolean (0,1) value, true if hypotheses x_i and x_j are of the same type and overlap in physical space; and r is a Boolean, true if hypotheses x_i and x_j are legs and share the same parent (modelling this parent-sharing is why we parametrise L by T in eqn. 8).

2.7. Hierarchical Object Inference

For a given set of shapelet observations and a set of candidate hierarchical legs and tables, CrunchBot may thus construct a factor graph. (We later describe how such a set of candidates is obtained automatically). Inference would become highly complicated if CrunchBot

Algorithm 1 Blackboard-inspired approximate Metropolis-hasting proposals generation.

```
for each time step t do
  update shapelet queue S by reading sensors
  for each annealing inverse temperature β do
    for each shapelet S_i ∈ S do
      propose and test parent H_i from Q(pa(S_i))
      if accepted, add H_i to hypothesis set B
    end for
    for each hypothesis H_i ∈ B do
      r ← rand(0, 1)
      if r < r_1 then
        propose death of H_i
        if accepted, remove H_i from B
      else
        if r < r_2 then
          propose parent change for H_i
          if accepted, replace H_i's parent parameter
        else
          if r < r_3 then
            propose child H_j from Q(ch(H_i)|H_i)
            if accepted, add H_j to hypothesis set B
          else
            propose parent H_j from Q(pa(H_i)|H_i)
            if accepted, add H_j to hypothesis set B
          end if
        end if
      end if
    end for
    prune all hypotheses not linked to any shapelet directly or via a common ancestor.
  end for
end for
```

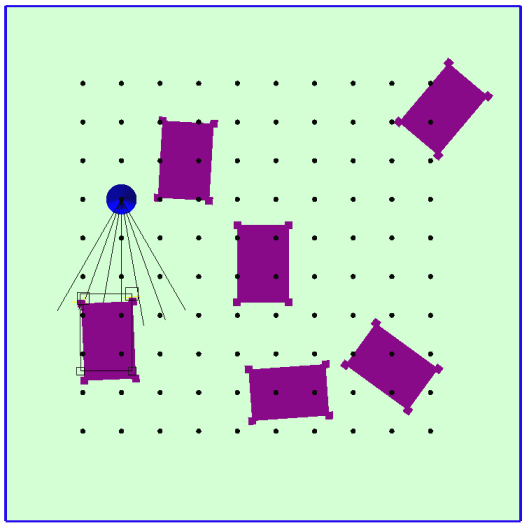


Figure 6: Overhead view showing ground truth table configuration, and locations (black dots) of the discrete poses occupied by the robot. There are four angle poses at each location, facing in compass directions.

had an infinite memory for shapelets, so in the present study we use a working memory (queue) of the seven most recent shapelets, and discard all others. At each t, new shapelets are read from the sensors, and inference is performed with the aim of obtaining the Maximum A Posterior (MAP) interpretation of their table causes, before the next time step begins,

$$\mathrm{MAP_t} = \arg_{\{T_j\}} \max P(\{T_j\}_j | \{S_k\}_k). \tag{27}$$

Thus CrunchBot currently – naively – treats each time step as an independent inference problem. Limiting inference to the most recent shapelets also has the effect of working within a local 'fovea' of attention: if no recent shapelets are from distant areas, then only hypotheses around CrunchBot's location will be considered.

There is some subtlety in defining the meaning of MAP states in continuous parameter spaces. In the present study, we assume that discrete hypotheses $H_i(x, y, \theta, \Theta)$ (where $H \in \{S, L, T\}$) represent small but non-infinitesimal collections of possible (x, y, θ) poses, with probability

$$P(H((x-\tfrac{\delta}{2}, x+\tfrac{\delta}{2}), (y-\tfrac{\delta}{2}, y+\tfrac{\delta}{2}), (\theta-\tfrac{\delta}{2}, \theta+\tfrac{\delta}{2}), \Theta))$$
$$= \delta^3 p(H(x, y, \theta, \Theta)), \tag{28}$$

where δ is a small but nonzero constant, Θ are the remaining parameters, and p is the density.

CrunchBot uses the annealed [1] approximate Metropolis-Hastings sampler of algorithm 1 to perform inference. Unlike standard inference problems, object-based mapping is a form of scene analysis task, i.e. the number of objects in the world – and therefore the number and type of nodes in the network – is unknown in advance. Algorithm 1 uses blackboard-like priming and pruning heuristics integrated with the sampling, to control the size of the network. Each hypothesis in the current 'blackboard' set B maintains (amongst other parameters), pose parameters x, y, θ and a current parent. The current parent may be another hypothesis, or may be null. Importantly, hypotheses that are not currently 'true' (according to the sampler) are never stored in B. The set B acts as a factor graph as detailed in the previous section, and may be thought of as the contents of a blackboard [10].

To obtain unbiased samples from the true joint distribution, Metropolis-Hastings sampling requires detailed technical conditions to be met, which are complicated by the jumps between factor graphs of different structures and sizes. Reversible jump methods [25] provide a rigorous theoretical basis from which to define acceptance probabilities based on reweighting proposals. Future work should incorporate such theory, for now we heuristically choose the Q distributions[2] and r_i thresholds; and use the annealed original P distribution from the factor graph as a simple Gibbs [1] acceptance probability,

$$P(\text{accept } H_i) = P^{\beta}(H_i|mb(H_i)), \tag{29}$$

where $mb(H_i)$ is the Markov blanket of H_i containing its parents, rivals $riv(H_i)$, and children $ch(H_i)$, β is inverse temperature. The Markov blanket conditional is

$$\begin{aligned} P(H_i|mb(H_i)) &= P(H_i|pa(H_i), cop(H_i), ch(H_i), riv(H_i)) \\ &= \frac{1}{Z}\frac{\Phi_a\Phi_b\Phi_c P(H_i|pa)P(ch|H_i)}{P(ch|(H_i))P(H_i|pa) + P(ch|\neg H_i)P(\neg H_i|pa)} \\ &= \frac{1}{Z}\frac{\Phi_a\Phi_b\Phi_c p(H_i|pa)p(ch|H_i)}{\delta^3(p(ch|H_i)p(H_i|pa) + p(ch|\neg H_i)p(\neg H_i|pa))}, \end{aligned}$$

where Z normalizes the factors contribution $\Phi_a\Phi_b\Phi_c$ only; δ is the constant of eqn. 28; $\Phi_a = \phi_a(H_i)\prod_{j\in pa(i)}\phi_a(H_j)$ includes missing children of H_i and also the missing child penalty for each parent of H_i which would have a missing child in the case where H_i is false; $\Phi_b = \prod_{j\in mb(i)}\phi_b(H_i, H_j)$ and $\Phi_c = \prod_{j\in mb(i)}\phi_c(H_i, H_j)$. The update allows computation to proceed using density functions rather than probabilities, but depends on the choice of the small constant, δ.

Newly proposed nodes must be linked to existing ones, so it is necessary to locate all potential parents $pa(H_i)$. A threshold radius in pose space is used, which limits this set to candidates which are close enough to have non-negligible generating probabilities, i.e.

$$pa(H_i) := \{H_j : P(H_i|H_j) \gg 0\}. \tag{30}$$

For computational efficiency it is useful to implement a spatial hash-table to look up these nearby hypotheses. This hash-table may also be reused to look up overlapping hypotheses in the computation of ϕ_c.

2.8. Mapping task

To remove the complexities of noisy odometry localisation during mapping, a noiseless-odometry simulation of Crunchbot in a world populated by six four-legged, table-like objects was implemented. The simulation is coded in C++ using the ODE physics engine (`www.ode.org`) for whisker contact detection. Source code is available in the supplemental material. The sensor noise levels are comparable to those found in the physical classifiers. The agent follows a fixed sequences of poses around the world and runs algorithm 1 once at each pose. There are $10 \times 10 \times 4$ poses, from 10 discrete x and y positions and four compass θ angles, as shown in fig. 6. To further simplify the present simulation, tables and table hypotheses all have fixed identical w_T^x, w_T^y and τ_T parameters; and physical (but not hypothesis) tables and have fixed identical w_T^L parameters.

2.9. Entropy based exploration

While the mapping experiment uses a fixed sequence of CrunchBot locations, we performed a further experiment to investigate a potential method for autonomous exploration. Preliminary experiments suggested that a common scenario which could enable exploration is the presence of ambiguity between rival high level table percepts, as illustrated in fig. 3.4. In these cases, CrunchBot's whiskers have seen a set of shapelets which enable the presence of one or two legs to be inferred, but the configuration of the rival tables remains ambiguous. In fig. 3.4, the Gibbs sampler is switching between tables $T1$ and $T2$ which are equally valid explanations of legs $L1$ and $L2$. We note that the presence of $T1$ and $T2$ are strongly corellated with the presence of legs $(L3, L4)$ and $(L5, L6)$ respectively. Therefore,

[2] details can be found in the source code, however note that MH sampling can operate on *any* proposal Q so its precise form is unimportant. Better results are obtained as the approximate Q becomes close to the true P.

observing the presence or absence of shaplets caused by any of these legs could resolve the high-level ambiguity about the tables.

This idea can be made precise by using a novel hierachical version of well-known entropy-based exploration methods (reviewed in [42]), integrated into the Bayesian blackboard architecture. During mapping, CrunchBot's goal is to maximize knowledge about table locations (i.e. construct a map showing the tables.) We can write this goal as maximixing entropy of the distribution over sets (maps) of tables $\{T_i\}_i$,

$$\max H(\{T_i\}_i) = \max_{\{T_i\}_i} (-\langle \log P(\{T_i\}_i)\rangle_{T_i}) \quad (31)$$

Future work could explore ways to compute this entropy exactly within the Bayesian blackboard framework; in the present study we simplify the computation by quantising a local region of space around CrunchBot into a square occupancy grid, and working with the probabilities that each grid cell is occupised by a leg or a table instead of the full distribution over sets of tables. As is common in occupancy grid methods, we assume (strongly, but falsely) that cell probabilities are mutually independent. Under these assumptions the local map probability is,

$$\prod_{x,y} P(T(x,y)) \quad (32)$$

where $P(T(x,y))$ is the table occupancy probability of the cell at location (x,y). From this we further approximate the entropy of the distribution over sets of tables, by the sum of the grid cell entropies,

$$H(\{T_i\}_i) \approx \sum_{x,y} H(T(x,y)), \quad (33)$$

which is the new goal to minimize. We quanitise CrunchBot's next possible (greedy) actions as movements to the same set of grid cells, and assume that visiting a cell will always find any legs in that cell (as a result of the body-whisks, and radial distance and orientation reports). We ignore any evidence that may be collected during the path to reach the action cell in our approximiation. Let the action of moving to and observing the cell (x_a, y_a) be $a(x_a, y_a)$.

Performing action $a(x_a, y_a)$ will (by assumption) answer with certainty the Boolean question of whether or not there is a leg $L(x_a, y_a)$ in cell (x_a, y_a). It may also reduce the entropy of the tables, by restricting possible percepts to those matching the presence or absence of this leg, as in fig. 3.4. We can compute this potential entropy change in advance, from the *current* location,

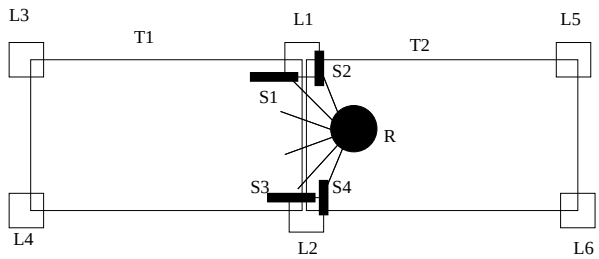


Figure 7: Typical ambiguous table scenario. Here the robot R has reported shapelets $S1 - 4$ and inferred legs $L1, 2$ unambiguously. But these legs are equally compatible with tables $T1, 2$ having legs $L3, 4$ and $L5, 6$. Exploring any of these four legs would remove the ambiguity about the tables.

before the action is performed, by computing the distribution of tables conditioned on the possible Boolean leg state b,

$$\{P(T(x,y)|L(x_a, y_a) = b)\}_{x,y}, \quad (34)$$

then computing $\Delta H =$

$$H(\{T(x,y)\}_{x,y}|L(x_a, y_a)) - H(\{T(x,y)\}_{x,y}|\neg L(x_a, y_a)) \quad (35)$$

$$= \sum_{x,y} H(T(x,y)|L(x_a, y_a)) - H(T(x,y)|\neg L(x_a, y_a)) \quad (36)$$

for each $a(x_a, y_a)$. The action with the largest difference in entropy is the most informative about the table distribution and is thus could be a useful candidate to explore next.

In practice, we need a way to approximate equation 34. In the present study, we experimented by using samples of table sets $\{T_i\}_i$ drawn from the existing Metropolis-Hastings sampler. For each sample of tables, the Boolean cell occpancies are computed (by drawing the tables onto a grid using a graphics library, www.cairographics.org), then normalized occupancy frequencies summed over samples used as approximations to occupancy probabilities $P(T(x,y))$. To avoid estimation bias due to the changing annealing temperature, we extended the annealing cycle with a fixed, high temperature phase at $\beta = 1/7.5$ for $N = 20$ steps before beginning to reduce the temperate for the MAP optimisation annlealing phase. The high temperature was used to allow the sampler to jump often between minima, as at $\beta = 1$ there is little probability of the rivals and missing children – which are necessary to transition to alternative ambiguous table percepts – ever occuring.

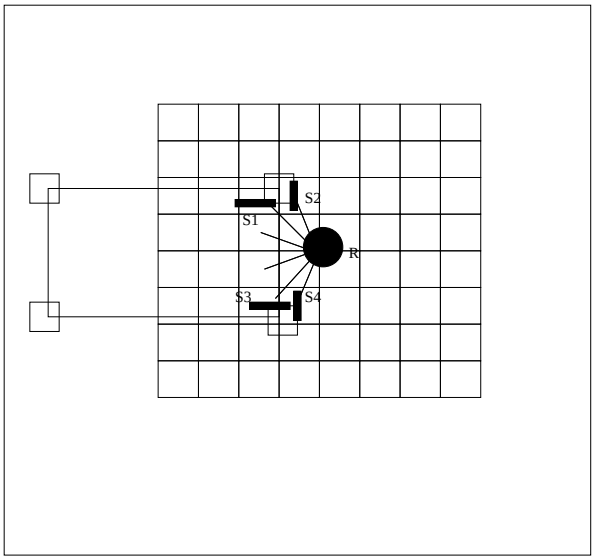


Figure 8: Illustration of the foveal grid used in entropy mapping. The grid covers a small local region around the robot, within the arena. Table and leg occupancies are recorded in each cell and used to find regions of interest, such as the ambiguity-resolving legs of fig. 3.4.

3. Results

3.1. Radial distance reports

Peak deflection magnitude for each contact is shown in Fig. 9. Standard deviation of error for radial distance estimation is shown in Table 1.

Standard classification error is very low, typically less than 5mm over the 60mm range tested. For some whiskers classification error is even lower, below 2mm. These results compare favourably with previous work under highly controlled conditions where speed was variable. This indicates that the noise in the odometry is low enough to ensure a consistent contact force and speed.

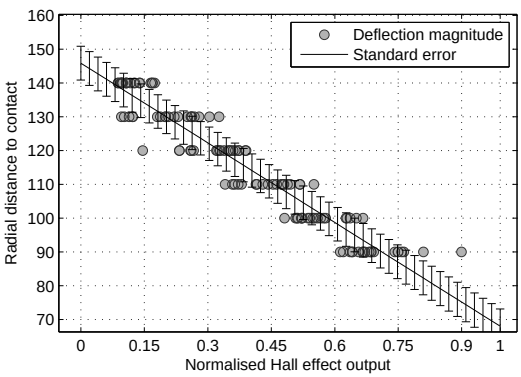


Figure 9: Peak deflection magnitude for contacts along the shaft of the whisker. Standard error for the regression is 4.98mm

W	1	2	3	4	Combo
σ	2.78mm	1.82mm	4.37mm	5.68mm	4.98mm

Table 1: Standard classification error for radial distance estimation on the CrunchBot mobile robot. Results are given for each whisker in turn, and or all the whiskers together. W = whisker, σ = standard error of classification

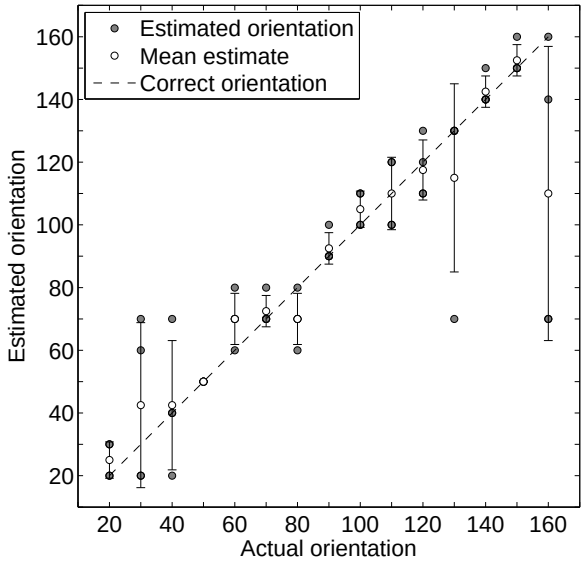


Figure 10: Surface orientation discrimination error with a template based classifier. Mean classification error is -2.2°, standard error is 21.2°

3.2. Surface orientation reports

Fig. 10 displays surface orientation estimation performance for the template based classifier. Mean classification error is -2.2°, standard error is 21.2°. Classification performance is best for orientations near 90°, with larger errors being made for large and small orientations. This may be due to fewer whiskers making contact with the surface at the extreme orientations, providing less information to the classifier with which to base a classification.

3.3. Hierarchical object mapping

Based in part on the physical classifier results, the parameters used were: $c_T = 0.4, c_L = 0.1, c_S = 0.05, \epsilon_a = 0.3\epsilon_b = 0.1, \epsilon_c = 0.1, \sigma_r = 0.2, \sigma_\theta = \pi/32, \delta^3 = 0.4$. The annealing schedule was $\beta_i = \exp\{7.5 - 0.5i\}$. Steps in the inference are illustrated in the supplemental video material. The MAP hypothesis

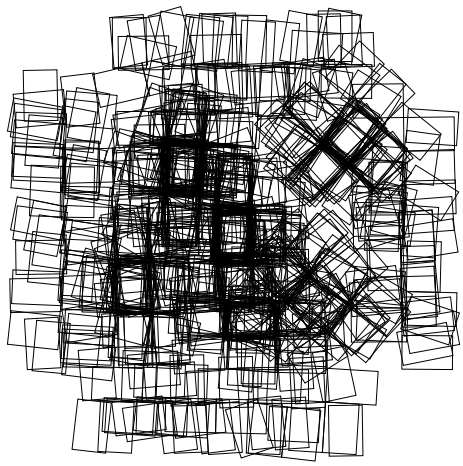


Figure 11: Montage showing collection of inferred tables from each independent robot pose, for realistic [18] ($\sigma_r = 0.1, \sigma_\theta = \pi/32$) sensors.

sets from all poses are collated and plotted onto a map of the arena in fig. 11. Comparing against the ground truth in fig. 6, the collated plot shows that table hypotheses are usually found in the correct locations, corresponding to the real tables. The average number of whiskers contacting tables at each pose having at least one table contact is 4.2$\pm$1.7. As we would expect from such a sparse amount of data, there are thus many incorrect hypotheses found in MAPs of the form shown in fig. 5. These are created from poses which do not provide enough information about the tables to resolve ambiguities, for example when the robot is close enough to touch two legs but no third leg as in fig. 5. Also of interest in the results are the many table hypotheses perceived around the edge of the arena. These are due to the agent observing shapelets from contact with the walls around the arena. The system does not (yet) have perceptual models of walls, so the best available explanations for such shapelets are those which postulate tables with legs at these shapelet locations. (This is a form of perceptual relativism: lacking a WALL concept, the system explains the data using its best available TABLE theories.) Similar plots for noiseless and highly noisy sensor cases are shown in figs. 12 and 13 for comparison. In both cases, the approximate locations of inferred tables are similar, though the accuracy of inferred table poses depends on the noise.

3.4. Entropy based exploration

While CrunchBot's annealed Bayesian Blackboard was sucessful in finding tables in the arena, it showed less success in finding good exploration locations. We

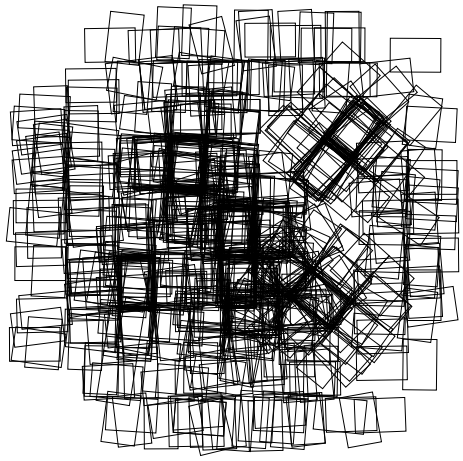


Figure 12: Montage showing the collection of inferred tables from each independent robot pose, for ideal, noiseless sensors.

Figure 13: Montage showing the collection of inferred tables from each independent robot pose, for very noisy ($\sigma_r = 0.5, \sigma_\theta = \pi/8$) sensors.

have showed mathematically how to set up exploration with hierarchical objects as a conditional entropy minimisation task, however we noted that computation of eqn. 34 requires approximation and chose to use occupancy frequencies from the high-temperature sampler to approximate the occupancy probabilities. The resulting behaviour of the simulated CrunchBot was indistinguishable from random movements, and CrunchBot did not appear to explore disambiguating legs as in fig. 7. . Analysis of the entropy maps gives some idea of the failure of the approximation of eqn. 34, and examples are shown in fig. 14. The problem is that the entropy differences are dominated by the probabilities of the empty space around the tables. Beginning with a flat prior on occupancy, and fusing in table percepts conditioned on leg states, the problem is that some leg states are visited more times than others. So as well as adding to the table distributions, the observations of empty space also deepen the probability of non-occupancy there. As we do not have access to an infinite number of observations, we do not reach the true occupancy distributions, but instead move towards them in proportion the number of observations. But the number of observations differs according to the leg state, meaning that the probability of non-occupancy for the background is higher for leg states that are visited more by the sampler. Thus the differences in background probability between common and uncommon leg states can become large, and contribute more to ΔH than the actual table distribution. This experiment shows that while the mathematical model and initial approximations may be sound, the frequency based approximation to the conditional occupancy probabilities is poor in this setting, and further work should be done to find better approximations. (This is potentially a large area of research, for example a recent entire PhD thesis was devoted to similar problems arising from a much more simplified grid environment, [42]).

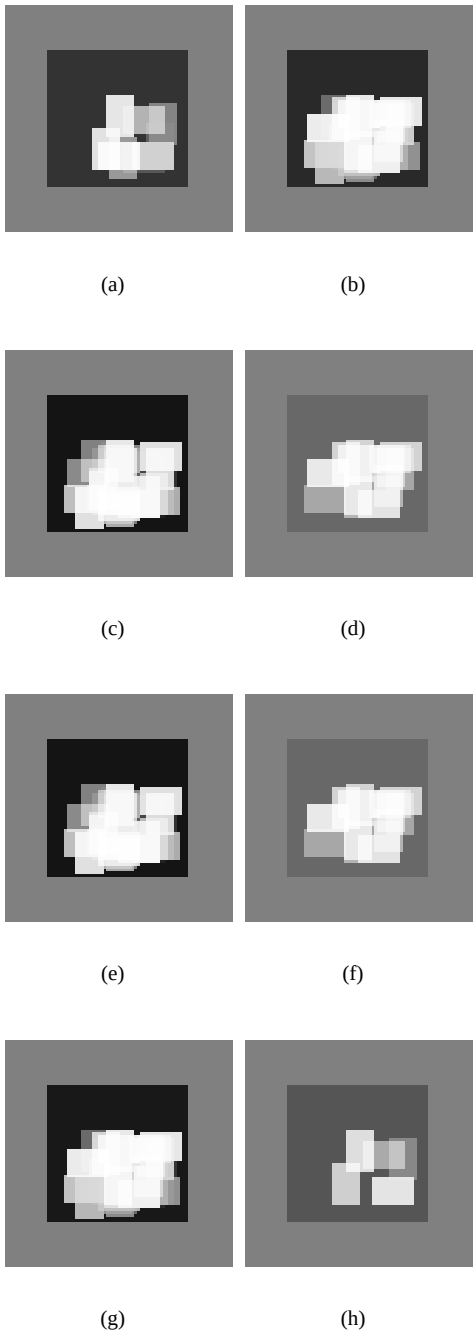


Figure 14: Examples of pairs (rows) of entropy maps. Each pair shows the table grid cell occpancy probabilities, conditioned on the presence (left) or absence (right) of a leg at some location in the arena. It can be seen that the entropy change is dominated by the shading of the background rather than the table distribution proper.

4. Discussion

Map building with only CrunchBot's whisker sensors is a difficult task, and our previous paper [20] gave some indication of the problems faced by conventional particle filtering and EKF SLAM style approaches to the mapping and navigation problems. In contrast, the present paper has shown how to combine signal processing for extraction of information about distance and surface orientation from physical whiskers with strong hierarchical priors about objects to compensate for the poverty and locality of the initial touch information.

After demonstrating that practical extraction of distance and orientation are possible on the real world, mobile CrunchBot platform (unlike previous work which has performed similar demonstrations in highly constrained, fixed-base environments, we then showed how these reports can be fused together using a Bayesian Blackboard to perceive high level objects such as tables that caused the reports.

Many simplifications were made in the present blackboard implementation, which future versions of the system should relax. The 'maps' presented here are simply the collation of many independent $\mathrm{MAP_t}$ inferences made from the different poses, and no information is shared between poses. Storing longer-term memories of shapelets and fusing them into the inferences would obviously allow a more refined map of the arena to be constructed: at present each table shown in the results has been inferred from typically 4.2 ± 1.7 shapelets only, which is extremely sparse. To avoid combinatorial explosion from handling many historical shapelets, one approach would be to discard very old shapelets memories but preserve only the locations of recognised tables and other high level objects, similar to the approach used in [16]. Such an approach raise interesting questions and analogies about the biological split between perception of the immediate local present (thought to occur in cortex), and perception of distinct locations and the past (thought to occur in hippocampus [8]). The present system makes no use of negative evidence, i.e. the observed absence of shapelets on non-contacting whiskers: this could be used to remove some of the ambiguous percepts. The heuristic threshold constants in the proposal distribution should be replaced with Reversible Jump MCMC reweightings to remove bias in the sampling distribution (although in practice the heuristic thresholds can work well, as ultimately only the annealed MAP is sought, rather than an approximation to the whole distribution).

The template classifier was able to discriminate the orientation of a surface but was not trained to discriminate other sorts of contacts, for example with the corners of objects. In principle it is possible to train a template classifier on every possible contact in the arena. However collecting such a data set would be impractical, and the computations involved in comparing incoming data to templates for every possible contact could be cumbersome. An alternative approach is to extract features from the tactile data, as was done radial distance estimation in this paper, has been done in the field of haptic touch [48],[45] and is commonly used in vision [28], and audition [4]. It has been proposed that cells in the thalamus and cortex of the rat are encoding features [39],[27] in this way. Though reliable features can be extracted for radial distance estimation in this paper, and contact speed on a stationary robot [13], it is unclear what other features can be extracted from whisker deflection signals for discriminating different kinds of object properties. In our own lab we are developing features for whisker based tactile sensing of contact geometry [14] and texture [17]. In future we hope to be able to combine features for diverse tactile properties in rich environments into a coherent system onboard a mobile robot, which in turn would provide reports that could be used as inputs to hierarchical object models as presented in this study.

We showed how to frame the exploration question for hierarchical objects in terms of entropy, in a related but novel approach from standard entropy grid based mapping (known as Active SLAM, eg. [6][3], but found an initial computational approximation to entropy to be lacking for the implementation purpose. It is possible that links to work of [42] may be useful to produce better approximations here in future work.

Importantly, the present system operates in a world having only one size and texture of table (though tables may have different leg sizes). Enlarging the parameter space to range over tables sizes and textures will allow inference of more realistic four-legged objects such as different kinds of chairs and desks. Other types of objects could also be introduced, such as walls, kitchen units and radiators. The Bayesian blackboard architecture is able to automatically select between rival object models, treating them as rival hypotheses [16]. However, as the number of models and parameters grows, sampling of course becomes less efficient. For example, it becomes less probable that a perfectly-fitting table will ever be proposed. (Even though once proposed, it will tend to remain accepted for having such a good fit.) Future work should investigate the use of 'smart proposals' which are classical heuristic object detectors (e.g. Hough transforms to find edges and corners) but re-purposed as Metropolis-Hastings proposals in the Bayesian Blackboard. When combined with RJ-MCMC acceptance probabilities, this gives a way to speed up the proposals but retain the probabilistic semantics. Further research should also extend the system to recognise several types of object of varying sizes, incorporate our previous research on CrunchBot's texture recognising abilities [20] and move CrunchBot's hierarchical mapping components from simulation to its physical plat-

[3]This paper, together with [22, 47], also gives ideas for how future CrunchBot versions could recover from getting lost during failed inference, by monitoring uncertainty about location.

form. The latter will involve handling the full SLAM problem rather than just mapping – however once large objects such as tables are recognised it should be possible to treat them as features in a standard EKF type approach. New forms of loop-closure in SLAM may become possible by recognising different parts of the same hierarchical object, for example CrunchBot may be able to close a loop by recognising a previously unseen leg of a previously seen table. Tracking of moving hierarchical objects may become possible by fusing CrunchBot's hierarchical models with the SLAP algorithm [37].

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, 1988.

[2] A. Ahl. The role of vibrissae in behavior: a status review. *Veterinary Research Communications*, 10(1):245–268, 1986.

[3] K. Beevers and W. Huang. Slam with sparse sensing. In *ICRA*, 2006.

[4] J. B. Bello and M. Sandler. Techniques for automated music transcription. *International Symposium on Music Information Retrieval*, 2000.

[5] T. Binford and T. Levitt. Evidential reasoning for object recognition. *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, 2003.

[6] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri. An application of kullback leiber divergence to active slam and exploration with particle filters. In *IROS*, 2010.

[7] G. Carvell and D. Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *J. Neurosci.*, 10(8):2638, 1990.

[8] C.Fox and T.Prescott. Hippocampus as unitary coherent particle filter. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2010.

[9] M. E. Diamond, M. von Heimendahl, P. M. Knutsen, D. Kleinfeld, and E. Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[10] L. Erman, F. Hayes-Roth, V. Lesser, and R. Reddy. The Hearsay-II system. *ACM Computing Surveys*, 12(2), 1980.

[11] M. Evans, C. Fox, M. Pearson, and T. Prescott. Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task. *Proceedings TAROS2009*, pages 19–24, 2009.

[12] M. Evans, C. Fox, and T. Prescott. Tactile discrimination using template classifiers. *From Animals to Animats SAB2010.*, 2010.

[13] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proc IEEE ROBIO*, 2010.

[14] M. H. Evans, C. Fox, M. J. Pearson, and T. J. Prescott. Object location, orientation, and velocity extraction from artificial vibrissal signals. In *Society for Neuroscience Abstracts. Society for Neuroscience (Program No. 174.8/ Z12)*, 2009.

[15] M. Fend. Whisker-based texture discrimination on a mobile robot. *Advances in Artificial Life*, pages 302–311, 2005.

[16] C. Fox. ThomCat: A Bayesian blackboard model of hierarchical temporal perception. In *Proc. FLAIRS*, 2008.

[17] C. Fox, B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 26(4):223–239, 2009.

[18] C. Fox, M. Pearson, B. Mitchinson, T. Pipe, and T. Prescott. Simple features for texture classification from robot whisker strains. *Somatosensory and Motor Research*, 24(3):139–162, 2007.

[19] C. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. TAROS 2011 (this volume).

[20] C. W. Fox, M. Evans, N. Lepora, A. Ham, and T. J. Prescott. Crunchbot: a mobile whiskered robot platform. In *Proceedings of Towards Autonomous Robots (TAROS)*, 2011.

[21] C. W. Fox, M. Evans, M. J. Pearson, and T. J. Prescott. Towards temporal inference for shape recognition from whiskers. In *Proc TAROS*, 2008.

[22] D. Fox. Adapting the sample size in particle filters through kld-sampling. *International Jounral of Robotics Researc*, 12, 2003.

[23] G. Gallagher, S. S. Srinivasa, J. A. Bagnell, and D. Ferguson. Gatmo: A generalized approach to tracking movable objects. In *ICRA*, 2009.

[24] V. Gopal and M. J. Z. Hartmann. Using hardware models to quantify sensory data acquisition across the rat vibrissal array. *Bioinspir Biomim*, 2(4):S135–45, 2007.

[25] P. Green. Reversible jump Markov chain Monte Carlo computation. *Biometrika*, 82(4):711–732, 1995.

[26] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. Diamond, and P. Konig. Texture signals in whisker vibrations. *Journal of neurophysiology*, 95(3):1792, 2006.

[27] S. Jadhav, J. Wolfe, and D. Feldman. Sparse temporal coding of elementary tactile features during active whisker sensation. *Nat Neurosci*, 12:792–800, 2009.

[28] L. Juan and O. Gwun. A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(5), 2010.

[29] M. Kaneko, N. Kanayama, and T. Tsuji. Active antenna for contact sensing. *IEEE Transactions on robotics and automation*, 14(2):278–291, 1998.

[30] D. Kim and R. Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55(3):229–243, 2007.

[31] K. B. Laskey and P. C. da Costa. Of starships and klingons: Bayesian inference for the 23rd century. In *Proc. UAI*, 2005.

[32] N. Lepora, M. Evans, C. Fox, M. Diamond, K. Gurney, and T. Prescott. Naive Bayes texture classification applied to whisker data from a moving robot. *Proc. IEEE World Congress on Comp. Int. WCCI2010*, 2010.

[33] Melexis. www.melexis.com/assets/mlx90333_datasheet_5276.aspx.

[34] B. Milch. *Probabilisitic Models with Unknown Objects*. PhD thesis, UC Berkeley, 2006.

[35] M. Mitchell. *Analogy-Making as Perception*. MIT, 1993.

[36] B. Mitchinson, C. J. Martin, R. A. Grant, and T. J. Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.

[37] M. Montemerlo, W. Whittaker, and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002. ICRA.

[38] J. Pearl. *Causality*. Cambridge University Press, 2000.

[39] R. Petersen, M. Brambilla, M. Bale, A. Alenda, S. Panzeri, M. Montemurro, and M. Maravall. Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron*, 60(5):890–903, 2008.

[40] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng. Touch based perception for object manipulation. In *ICRA*, 2007.

[41] T. Prescott, M. Pearson, B. Mitchinson, J. Sullivan, and A. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.

[42] Z. Saigol. *Automated Planning for Hydrothermal Vent Prospeting using AUVS*. PhD thesis, University of Birmingham, UK,

2010.

[43] A. Schultz, J. Solomon, M. Peshkin, and M. Hartmann. Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. *Proc. ICRA2005*, pages 2588–2593, 2005.

[44] A. Seth, J. McKinstry, G. Edelman, and J. Krichmar. Texture discrimination by an autonomous mobile brain-based device with whiskers. In *Proc. IEEE Int. Conf. Robot. Autom. ICRA2004*, pages 4925–4930, 2004.

[45] J. Sinclair, J. Kuo, and H. Burton. Effects on discrimination performance of selective attention to tactile features. *Somatosensory & Motor Research*, 17(2):145–157, 2000.

[46] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. C. Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and J. M. Vandeweghe. Herb: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, January 2010.

[47] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *In RSS*, pages 65–72, 2005.

[48] S. Stansfield. Primitives, features, and exploratory procedures: Building a robot tactile perception system. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1274–1279. IEEE, 1986.

[49] C. Sutton, B. Burns, C. Morrison, and P. R. Cohen. Guided incremental construction of belief networks. In *Proceedings of the Fifth International Symposium on Intelligient Data Analysis*, 2003.

[50] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT, 2006.

[51] C.-C. Wang, C. Thorpe, S. Thrun, M. Herbert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *International Jounral of Robotics Research*, 26:889, 2007.

[52] W. Welker. Analysis of sniffing of the albino rat. *Behaviour*, pages 223–244, 1964.

[53] Y. Zhang, J. Liu, G. Hoffmann, M. Quilling, K. Payne, P. Bose, and A. Andrews Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *Mobile Adhoc and Sensor Systems (MASS), IEEE 7th International Conference on*, 2010.

Tactile SLAM with a biomimetic whiskered robot

Charles Fox, Mat Evans, Martin Pearson and Tony Prescott

***Abstract*— Future robots may need to navigate where visual sensors fail. Touch sensors provide an alternative modality, largely unexplored in the context of robotic map building. We present the first results in grid based simultaneous localisation and mapping (SLAM) with biomimetic whisker sensors, and show how multi-whisker features coupled with priors about straight edges in the world can boost its performance. Our results are from a simple, small environment but are intended as a first baseline to measure future algorithms against.**

I. Introduction

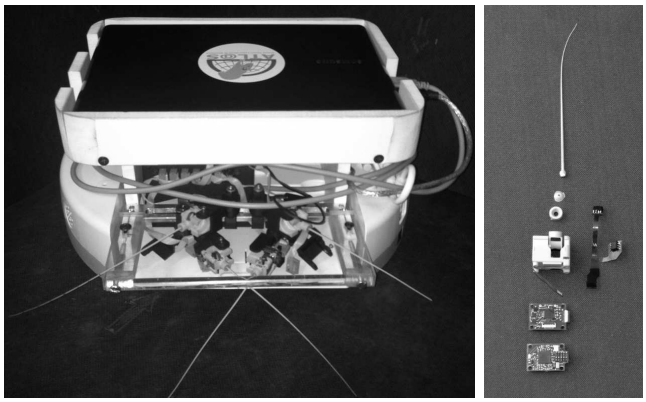
Tactile sensing allows rodents to excel in environments where many other sensory modalities are impaired. Rodents often operate in the dark or in complex underground tunnels, many are diurnal or nocturnal, and consequently cannot rely on their eyes to navigate, hunt and explore. Instead they are excellent at tactile sensing and use their whiskers (known anatomically as 'vibrissae') to gather information about the world. Unlike distal sensors such as lasers and vision, whiskers make direct contact with the world around the location of the agent only. There are environments where distal sensors are inappropriate in robotics, for example smoke-filled search and rescue operations, or adversarial environments where covert (emission-less) sensing is required. Biomimetic touch sensors such as whiskers have been proposed as useful in such environments [14].

The task of determining the nature of a surface a whisker has made contact with is a difficult one. Both the location in space, and the identity of an object must be determined while taking the agent's location uncertainty into account. A number of systems have been constructed for whisker-based tactile discrimination: using the geometry of whisker-object contact [22]; measurement of bending moments at the whisker base [18], [4]; or through the extraction of features from the whisker deflection signal for texture [15],[12] and radial distance estimation [7]. Integrating sensory systems onto an autonomous mobile robot presents additional challenges. Though a number of whiskered mobile robots have been built, [21] few if any perform mapping and navigation, beyond simple reactive behaviours such as wall-following and obstacle avoidance.

On first consideration, it may appear that touch-based SLAM is doomed to failure due to the sparsity of the likelihood functions. A previous study [10] gave examples of whisker-based likelihood functions, and showed them to have high spatial variances as well as many discontinuities and non-Gaussian shapes. The high spatial variances occur because many locations in the world look the same to touch sensors. In related – but different – modalities, short range infrared [2], [1] and sonar sensors [27] have enabled SLAM through multiscans, landmark constraints and inferences over pose histories; and infrared sensors with the (very) strong prior assumption that the world is made of rectilinear structures only [26]. (Rectilinear assumptions have also been made for covering tasks [5].)

Supported by FP7 BIOTACT (ICT-215910) and EFAA (ICT-270490).

Fox, Evans and Prescott are with the Sheffield Center for Robotics, UK. Pearson is with Bristol Robotics Laboratory, UK.



(a) (b)

Fig. 1. (a) The robot platform. (b) A single whisker unit. See Section II-A for a description of the components

This paper presents an alternative approach to strengthen mapping, using properties of whisker sensors without multiscan or rectilinear assumptions. Unlike infra-red or sonar sensors, which collect only *distance* information from a nearby *point*, whiskers can recover information about the local *orientation* of the surface from contacts [9]. We present three methods for whiskered tactile SLAM. First, using contact locations only; second, using geometric contact information from groups of whiskers to recover surface orientation (these two methods would also be applicable to the other types of short range sensors mentioned above) and third, using whisker-specific strain time series to recover surface orientation. In all methods we also exploit prior knowledge about the world structure: all three methods assume that neighbouring points are correlated; the two surface orientation methods assume that the environment is made mostly of straight edges, but of any orientation, unlike the rectilinear assumption.

II. Methods

A. Whiskers.

Experiments were performed using four artificial whiskers of 140mm in length, 1.45mm diameter at the base tapering

linearly to 0.3mm at the tip, and designed to mimic properties of rodent whiskers, at a scale of ≈5:1. They are built from nanocure25 using an Evisiontec rapid prototyping machine. Like rat whiskers, sensing takes place only at the base (or 'follicle'), and measures the local strain there [6]. Magnets are bonded to the bases of the whiskers and held in place by plugs of polyurethane approximately 0.75 mm above Melexix 90333 tri-axis Hall effect sensor ICs [19]. These sensors each generate two outputs representing the magnetic field direction (in two axes) with respect to its calibrated resting angle. These two 16-bit values are sampled by a local dsPIC33f802 micro-controller which is collected using an FPGA configured as a bridge to a USB 2.0 interface. Up to 28 whiskers can be connected to this FPGA bridge at one time. Using a software driver and API (Cesys GmbH), users can request horizontal and vertical strain data from all whiskers at a sample rate of 2kHz.

B. Robot platform.

Four whiskers are mounted in the cargo bay of an iRobot Create base (`www.irobot.com`), positioned on a rapid prototyped ball joint mountings which allow adjustment of the whiskers. We have also extended the cargo bay mounting to accommodate a netbook PC, which is used for local control of the robot and runs Ubuntu 10.10 on a single-core Intel Atom processor. A circular buffer in shared memory is used to make data from the Cesys driver available to other processes. The netbook hosts a Player server (`playerstage.sourceforge.net`) providing high-level, networked API interfacing to the Create's serial port commands. Low-level processes such as texture and shape recognition and basic motor control can run on the netbook, reading the raw data from the circular buffer. These processes send their results to a desktop machine which handles mapping. Communication is via the C++ Thrift RPC protocol. Differential and absolute odometry data from the Create is also sent to the mapping server. Preliminary experiments showed that the odometry of the Create, once loaded with the sensing and control hardware, is accurate to $< 5\%$ of any straight line or turn on the spot movements. It was useful to cache commonly used trigonometric quantities describing the whisker geometry to enable fast lookup during navigation.

C. Environment and behaviour

The robot is placed in the 1.25m × 1.25m square arena shown in Fig. 2, containing several square objects. Its movements are controlled by a finite state machine (FSM),

$$\begin{tikzcd}
FWD_1 \arrow[r] \arrow[dr] & HIT_1 \arrow[r] & FWD_2 \arrow[dl] \arrow[d] \\
TURN \arrow[u] \arrow[r] & ABORT \arrow[d] & HIT_2 \arrow[dl] \\
 & BACK \arrow[ul] & \\
\end{tikzcd}$$

Fig. 2. Arena used in map building experiments

which moves the robot forward in a straight line (FWD_1) *repeatedly* until its whiskers hit something ($HIT1$). It then moves the robot forward again at a slower speed (FWD_2) until either a second whisker makes contact (HIT_2), or the strain in any whisker exceeds a safety threshold ($ABORT$). The robot then reverses ($BACK$) and turns on the spot ($TURN$). Turns angles are draw from a mixture of two Gaussians, one of which has a small mean and variance (0.14π, 0) to encourage wall following and the other a large mean and variance (0.3π,0.25π) to encourage movement away from walls to explore other parts of the arena. Each FWD_1, FWD_2 and $BACK$ step of the FSM lasts for 0.5s, the FWD_1 and $BACK$ state moves at 0.05m/s and the FWD_2 state moves at 0.02m/s for safety. Turning is at 0.3rad/s. Within each 0.5s FSM step, the whisker strains are monitored regularly for strains exceeding the safety threshold – if this occurs, the motion is terminated early and the FSM switches to the $ABORT$ state then $BACK$ to escape. Under these behaviours, the robot tends to move anticlockwise overall around the arena, interspersed with periods of wall following and exploration, and typically makes around 3 or 4 circuits (and hopefully loop closures) in a 6 minute run.

Combined odometry and *radial distance* (i.e. the distance along each whisker shaft to any contact, or lack of any contact) reports are sent after every FSM step with the exception of $BACK$ states, which revisit recently visited poses and would double-count recent observations there if their likelihoods were fused into localisation and mapping.

D. Localisation

Localisation is performed using the standard [25] particle filter of algorithm 1. 100 particles $\{s^i\}_{i=1:100}$ are maintained, each of which carries a continuous-valued pose (2D location and orientation) and a grid map, $m[x, y]$, of the environment. We use a 50×50 grid map covering a 2.5m×2.5m space (double the dimensions of the arena to allow for overspill; 50mm cells). Updates occur at each FSM step.

Algorithm 1 Sequential importance particle filter

for each time step t **do**
  for $s = 1 : N$ **do**
    sample $s^I[t] \sim P(s^i|s^i[t-1], \delta_{trans}, \delta_{rot})$
  end for
  for $s = 1 : N$ **do**
    $\lambda^i[t] \leftarrow \frac{1}{Z} P(o[t]|s^i[t])$
  end for
  resample $s^i[t] \leftarrow s^j[t-1]$ with $P(j) = \lambda^j[t]$
end for

The observation vector o consists of radial distance estimates $\{r_i\}_{i=1:4}$ from the four whiskers, along with translational (δ_{trans}) and rotational (δ_{rot}) odometry estimates from the Create platform (and wall surface angle estimates ϕ, not used for localisation). The resampling step draws samples according to the likelihood (importance) weights of the previous set.

Radial distances from whisker sensors may be estimated using methods such as inverse beam theory [4], [18] and feature extraction [7] which could later be plugged in here – but in the present localisation (not mapping) module we simply set a strain threshold and declare a contact at a standard, fixed radius (130mm) when that threshold is exceeded (and the FSM moves to HIT_1 at this point).

Each particle's grid map contains occupancy probabilities which are used to compute the likelihood function λ as the product of the individual whisker likelihoods,

$$\lambda = \prod_{w=1}^{4} \lambda^w. \tag{1}$$

The likelihood of the ith whisker is a function of its radial contact distance, r_i, and the particle pose and particle map m. We discretise the whisker shaft into 5 segments, and assume the segments are represented by equally spaced points, $G \in 1:5$ along the length of the shaft. The observed radial distance r is discretised to the nearest segment number R. The location L of each segment point in the grid map is found, $[x, y] = L(G)$. The whisker likelihood is then given by the product of the map's probability of contact at the reported location, $m(L(R))$ and all the probabilities of no contacts at the segments between the base of the shaft and the observed contact,

$$\lambda^w = m(L(R)) \prod_{G=0}^{R} (1 - m(L(G))). \tag{2}$$

Software speed is important in particle filtering, so we pre-allocate memory for two populations of particles, then copy and update values between them in the manner of double-buffering, to avoid computational overheads of construction and destruction of objects. The two population buffers are held in shared memory, which allows monitors such as GUI displays to run as separate processes on the dual-core desktop SLAM machine with minimal communications overhead.

E. Blob-based mapping

The simplest whiskered mapping method would treat each contact as an observation of a single cellof the grid map at the contact location, and assume independence between cells. Preliminary experiments showed this is impractical, as there are many cells and only a small number of contacts (e.g. 30) during a run (of 6 minutes). A simple extension of this idea is to assume a local correlation between cells, as in a Markov Random Field. Under this assumption, a single contact observation gives rise to a small local Gaussian Δm likelihood to be fused into the grid map m,

$$\Delta m[x, y] = \Delta[x_c, y_c] \exp\{-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\}, \tag{3}$$

where (x_c, y_c) are the coordinates of the contact cell, σ is set to make the resulting blob affect a radius of about two pixels, and $\Delta m[x_c, y_c]$ is the likelihood of the original contact cell occupancy given the current particle and observation (set to a constant > 0.5).

Whiskers that do not make contact also carry likelihoods that cells along their lengths are empty. We approximate this by a single Gaussian as in eqn. 3, but with $\Delta m[x_c, y_c] < 0.5$, and (x_x, y_c) in the center of the whisker shaft. Furthermore, we know that the region occupied by the robot's current body position cannot be occupied by another object, so we can also fuse a similar negative evidence Gaussian centred on the robot body location.

F. Angle-based maps with multi-whisker contact geometry

A more sophisticated mapping strategy is to exploit prior knowledge about the structure of the world, coupled with using features from multiple whiskers together. Previous work [26] made the strong assumption that all objects in the environment have straight edges aligned along Cartesian axes, exploiting the fact that many man-made environments are based on square grids. Strong hierarchical object priors were used by [13] to constrain the interpretation of contacts as known 3D object forms. Here we use a prior whose strength lies somewhere between these strong prior approaches and the weak prior blob method of sec. II-E. We assume that the environment is made up mostly of long, straight edges, but do not impose a Cartesian grid on their poses or make assumptions about the 3D forms of objects. So rather than placing Gaussian blobs at contact points, we place a blur of long, oriented edges.

We and others have previously investigated the recovery of surface normal information from *individual* whisker strains [18],[9]. A new simple approach for *multi*-whiskered robots is to locate two contact points on the same surface with two different whiskers, then compute the angle between them. Assuming that edges in the world are locally straight at this scale, then this angle gives the angle of the surface. The two contacts can be read during the FSM states HIT_1 and HIT_2 as described in section II-C. (In some cases the FWD_2 state terminates to $ABORT$ without a second contact due to a strain safety threshold being exceeded. In these cases, we

revert to mapping a single Gaussian blob at the first contact point only as in sec. II-E.)

When an oriented surface is found in this way, we fuse a blur of long oriented edges into the map,

$$\Delta m[x,y] = \Delta[x_c, y_c] \exp\{-\frac{R^2}{2\sigma_R^2} - frac(\theta - \theta_c)^2 2\sigma_\theta^2\}, \tag{4}$$

where (R, θ) are radial coordinates centered on the contact midpoint (x_c, y_c) and estimated surface angle θ_c. We use $\sigma_R = 0.25$ and $\sigma_\theta = \pi/12$, values which were tuned by hand to produce a *long* (0.25m) blurred edge in the map around the contact point, giving the best maps for this environment.

G. Angle-based mapping with multi-whisker templates

It has been shown that simple k-means style templates on strain time series from *individual* whiskers can be used for discriminating contact *distance* classes in physical simulation [11], and stationary robot hardware [8]. In the present study we have access to four whiskers together, so we can train templates corresponding to contact *angle* classes from the 8-dimensional time series from the whole *multi-whisker* set (four whiskers, each with vertical and horizontal strain channels). The rationale for this approach is that the geometric multi-whisker method of sec. II-F must assume that the estimated contact locations are accurate – which is not necessarily true – and is restricted to utilising data from two contact whisker locations only. In contrast, a template method can utilities bulk data from all whiskers to find similar surface normals, and without any geometric assumptions. It is a purely data-driven method. Oriented edges as in eqn. 4 may again be added to the map once surface normals are found using templates.

Offline training data was collected by programming the robot to drive into a wall at fifteen different angles (20°:160° in 10° intervals) four times. Data was aligned to initial contacts (at HIT_1 occurrence), low pass filtered (17Hz) to remove oscillations caused by robot body movement, recorded for 2s, and smoothed with a five-point moving average. Templates were generated by averaging across the four sets for each angle. Templates for each angle comprised data of all eight channels from the four whiskers to allow multi-whisker information to inform classification.

During online SLAM, strain time-series data was logged from immediately after each HIT_1 to the following HIT_2 then sent to the classifier at HIT_2. The average squared error, e for each template, T_i is computed over the N logged data points,

$$e(T_i) = \frac{1}{N} \sum_{t=1}^{n} (I(t) - T_i(t))^2. \tag{5}$$

The template with the lowest sum of squared errors was determined the winner, and its surface normal used in eqn. 4 to fuse an oriented long edge into the map.

III. Results

Results are presented for mapping using the three methods: blobs, geometric multi-whisker and template multi-whisker.

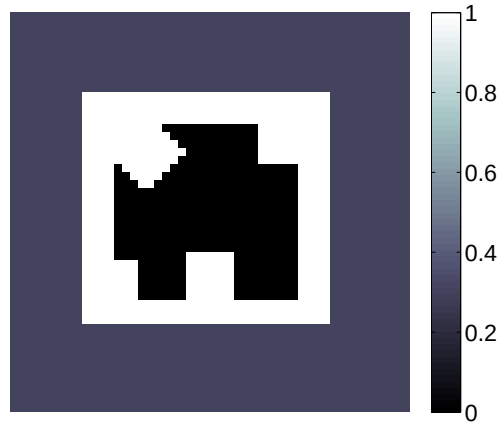


Fig. 3. Ground truth grid map based on the arena used for comparison with generated maps. Brightness indicates occupancy. (cf. Fig. 2.)

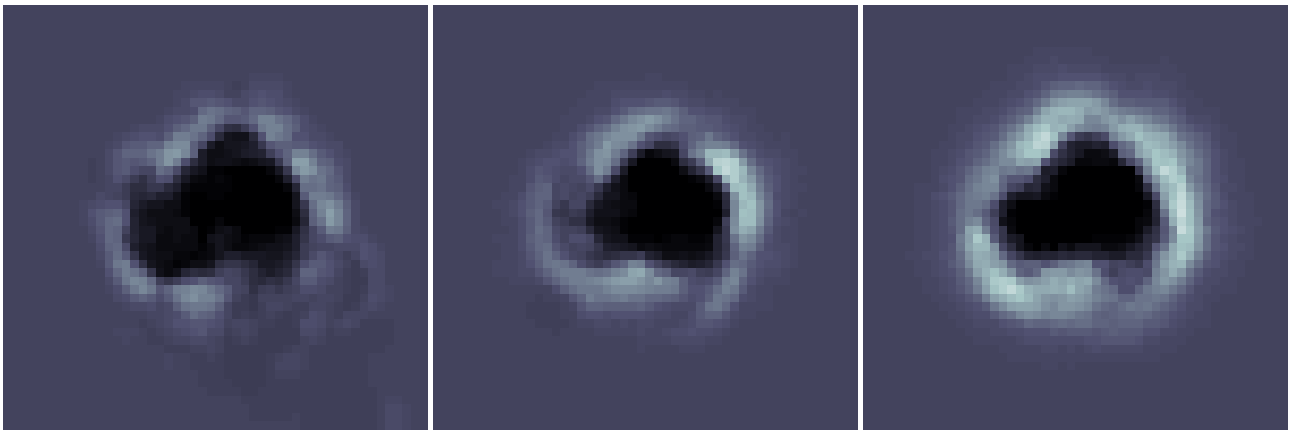
In each instance the robot was programmed to run for twenty trials of six minutes (6 hours' data in total). The resulting grid maps are compared to a ground truth grid map (Fig. 3; as built by a human observer – which is then smoothed with a 5 × 5 cell Gaussian filter, standard deviation 2.5). Occupancy in the grid map is represented by a 1 (object present) or a 0 (object not present). Unexplored areas are marked with 0.3 as this is approximately the mean occupancy of the arena. Grid maps, $m[x,y]$, are compared by an element-wise sum of absolute errors calculation to the ground truth map, $gt[x,y]$,

$$\frac{1}{N} \sum_{t=1}^{n} \frac{1}{50} \sum_{x=1}^{50} \frac{1}{50} \sum_{y=1}^{50} |m[x,y] - gt[x,y]|. \tag{6}$$

The mean error per cell is reported for each map. For baseline comparison, error for a random map populated from a Bernoulli distribution ($p = 0.5$) is 0.47 (unitless difference of probability).

Fig. 4(a) shows the average map generated from running the robot with the blob based mapping system. Mean occupancy error was 0.40. Fig. 4(b) shows the average map generated from running with the geometric multi-whisker based mapping system. Mean occupancy error was 0.39. Fig. 4 shows the average map generated from running with the template based mapping system. Mean occupancy error was 0.37.

So get some idea of failure cases, we also tested a larger 2.5m square arena in simulation only, using Player/Stage and the geometric angle method (physical strains for templates being unavailable in simulation), which allowed us to explore different odometry noise levels. For 2% noise in δ_{trans} and δ_{rot}, the inferred map and ground truth arena are shown in Fig. 6. Correct location tracking was maintained during the 200 steps used to build this map. We then ran a simulation with 5% odometry noise, but loop closures failed in this case.



(a) (b) (c)

Fig. 4. (a) Average grid map generated when using blob based mapping over twenty trials. Note grid occupancy outside of the area of the arena and low mean grid occupancy (dark occupied regions). (b) Average grid map generated when using multi-whisker angle based mapping over twenty trials. Note grid occupancy restricted to the area of the arena and high mean grid occupancy (brighter occupied regions). (c) Average grid map generated when using template based mapping over twenty trials. Note grid occupancy restricted to the area of the arena and high mean grid occupancy (brightest occupied regions). Brightness indicates occupancy, all maps are drawn on the same occupancy scale (0:1)

IV. Discussion

All three mapping methods perform well. The template method is the best and blob-based is the worst under the metric of eqn. 6. This seems to be because errors in localisation meant the robot became lost and occupied the map with objects outside of the area of the arena (as can be seen when comparing the lower right quadrants of Fig.3 and Fig. 4(a). Geometric multi-whisker angle mapping was an improvement on blob based mapping, with grid occupation being restricted to the area of the arena, and surface contours are recovered partially (white patches in Fig.4(b)).

The best performance comes from the template method. Mapping is restricted to the area of the arena and large sections of surface contours are recovered (prominent white patches in Fig.4(c)). Templates are especially useful as they provide strong predictions of surface angle even when only single whisker contacts are made – unlike the geometric method. Templates therefore can extract more information from impoverished whisker data, informing stronger predictions of object contours, leading to a greater occupancy in the grid maps.

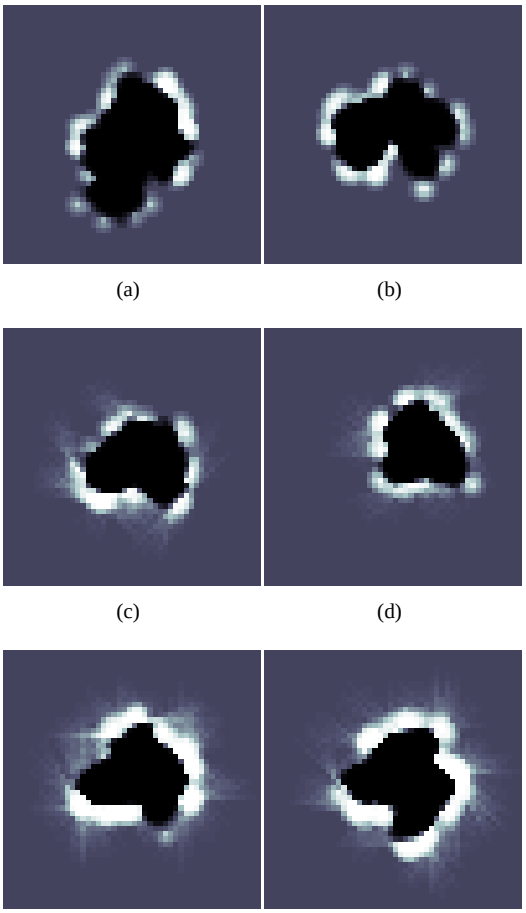
Differences between the mapping performance of the three methods can be seen more clearly in the maps generated on individual trials. Fig.5 shows typical maps generated in each of the three conditions. Blob based mapping (Fig.5 (a) and (b)) results in sparse object location reports, and unreliable localisation resulting in mapping outside of the area of the arena. Geometric multi-whisker based mapping (Fig.5 (c) and (d)) generates predictions of object contours, and these improve localisation to restrict mapping to the bounds of the arena. Template based mapping (Fig.5 (e) and (f)) generates more, better predictions of object contours (white areas in the grid maps), improving localisation. Object features such as sharp corners can also be seen in the template based grid maps (lower region of Fig.5 (a) and (b)).

The template classifier was able to discriminate the orientation of a surface but was not trained to discriminate other sorts of contacts, for example with the corners of objects. In principle it is possible to train a template classifier on every possible contact in the arena. However collecting such a data set would be impractical, and the computations involved in comparing incoming data to templates for every possible contact could be cumbersome. An alternative approach is to extract features from the tactile data, as has been done in the field of haptic touch [24],[23] and is commonly used in vision [17], and audition [3]. It has been proposed that biological cells in the thalamus and cortex of the rat are encoding features [20],[16] in this way. In our own lab we are developing features for whisker based tactile sensing of contact geometry [7] and texture of different materials [12]. In future we hope to be able to combine such features for diverse tactile properties in rich environments into a coherent system onboard a mobile robot.

V. Conclusion

We previously [10] showed the Create platform performing basic localisation from whiskers and a given map. This study has addressed the other aspect of SLAM – mapping – showing it is feasible to map *small* arenas using whisker sensors by exploiting timing information between contacts and prior knowledge about edges in the environment.

The results presented here are from a simple small arena only, and are of course not competitive with current SLAM systems using visual or laser sensors, which can run in large outdoor environments. However we have presented the first results of whiskered grid mapping, which may serve as a baseline for improved whiskered SLAM systems. For example, [13] presents more complex whiskered object recognition using even stronger priors, which could in future form part of a landmark-SLAM method for larger environments, and it would be useful to compare its results – and



(a) (b) (c) (d) (e) (f)

Fig. 5. Grid maps generated on individual trials for blog based mapping ((a) and (b)), multi whisker geometry based mapping ((d) and (e)), and template based mapping ((e) and (f)). Brightness indicates occupancy.

other algorithms such as multiscans and landmark constraints – with the grid-SLAM baseline presented here.

REFERENCES

[1] K. R. Beevers and W. H. Huang. Inferring and enforcing relative constraints in slam. In *WAFR*, 2006.

[2] K.R. Beevers and W.H. Huang. SLAM with sparse sensing. In *ICRA*, pages 2285–2290. IEEE, 2006.

[3] J. B. Bello and M. Sandler. Techniques for automated music transcription. *ISMIR*, 2000.

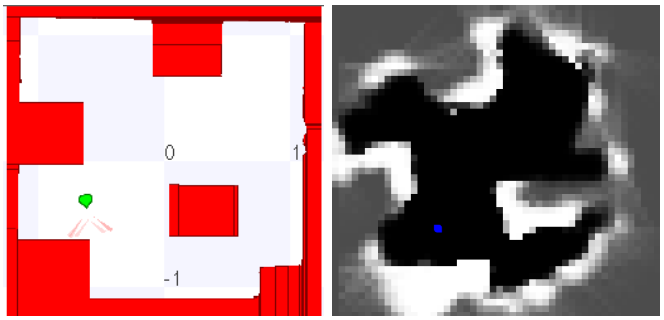
[4] J.A. Birdwell, J.H. Solomon, M. Thajchayapong, M.A. Taylor, M. Cheely, R.B. Towal, J. Conradt, and M.J.Z. Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *J Neurophysiol*, 98(4):2439–2455, 2007.

[5] Z. J. Butler, A. A. Rizzi, and R. L. Hollis. Contact sensor-based coverage of rectilinear environments. In *Proc. of IEEE Int'l Symposium on Intelligent Control*, 1999.

[6] Mathew E Diamond, Moritz von Heimendahl, Per Magne Knutsen, David Kleinfeld, and Ehud Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[7] M. H.. Evans, C. W. Fox, N.F. Lepora, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proceedings IEEE ROBIO*, 2010.

[8] M. H. Evans, C. W. Fox, M. J. Pearson, and T. J Prescott. Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. In *SAB*, 2010.



(a) (b)

Fig. 6. (a) Stage simulation of a larger arena, running geometric multi-whisker mapping. (b) Grid map generated in Stage simulation, with 2% odometry noise.

[9] M. H. Evans, C.W. Fox, M. J. Pearson, and T. J. Prescott. Object location, orientation, and velocity extraction from artificial vibrissal signals. In *SFN Abs. 174.8/Z12*, 2009.

[10] C. W. Fox, M. H. Evans, N. F. Lepora, M. J. Pearson, A. Ham, and T. J. Prescott. A mobile whiskered robot platform. In *TAROS*, 2011.

[11] C. W. Fox, M. H. Evans, M. J. Pearson, and T. J. Prescott. Temporal inference for shape recognition from whiskers. In *TAROS*, 2008.

[12] C. W. Fox, B. Mitchinson, M. J. Pearson, A. G. Pipe, and T. J. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.

[13] C. W. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. In *TAROS*, 2011.

[14] V.V. Hafner. Cognitive maps in rats and robots. *Adaptive Behavior*, 13(2):87, 2005.

[15] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, and P. Konig. Texture signals in whisker vibrations. *J Neurophysiol*, 95(3):1792–1799, 2006.

[16] S.P. Jadhav, J. Wolfe, and D.E. Feldman. Sparse temporal coding of elementary tactile features during active whisker sensation. *Nat Neurosci*, 12:792–800, 2009.

[17] L. Juan and O. Gwun. A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(5), 2010.

[18] D.E. Kim and R. Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55:229–243, 2007.

[19] Melexis. www.melexis.com/assets/mlx90333_datasheet_5276.aspx.

[20] R.S. Petersen, M. Brambilla, M.R. Bale, A. Alenda, S. Panzeri, M.A. Montemurro, and M. Maravall. Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron*, 60(5):890–903, 2008.

[21] T.J Prescott, M.J. Pearson, B Mitchinson, J.C.W. Sullivan, and A.G. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.

[22] R.A. Russell and J.A. Wijaya. Object location and recognition using whisker sensors. In *ACRA*, 2003.

[23] J. Sinclair, J. J. Kuo, and H. R. Burton. Effects on discrimination performance of selective attention to tactile features. *Somatosensory & Motor Research*, 17(2):145–157, 2000.

[24] S. Stansfield. Primitives, features, and exploratory procedures: Building a robot tactile perception system. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1274–1279. IEEE, 1986.

[25] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT, 2006.

[26] Y. Zhang, J. Liu, G. Hoffmann, M. Quilling, K. Payne, P. Bose, and A. Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *IEEE MASS*, 2010.

[27] G. Zunino and H.I. Christensen. Navigation in realistic environments. In *9th Int. Symp. on Intelligent Robotic Systems, SIRS*. Citeseer, 2001.

A comparative study of adaptive, automatic recognition of disordered speech

Heidi Christensen[1], *Stuart Cunningham*[2], *Charles Fox*[1],
Phil Green[1], *Thomas Hain*[1]

[1]Computer Science, University of Sheffield, Sheffield, United Kingdom
[2]Human Communication Sciences, University of Sheffield, Sheffield, United Kingdom
h.christensen@dcs.shef.ac.uk,s.cunningham@sheffield.ac.uk,charles.fox@dcs.shef.ac.uk,
p.green@dcs.shef.ac.uk, t.hain@dcs.shef.ac.uk

Abstract

Speech-driven assistive technology can be an attractive alternative to conventional interfaces for people with physical disabilities. However, often the lack of motor-control of the speech articulators results in disordered speech, as condition known as *dysarthria*. Dysarthric speakers can generally not obtain satisfactory performances with off-the-shelf automatic speech recognition (ASR) products and disordered speech ASR is an increasingly active research area. Sparseness of suitable data is a big challenge. The experiments described here use UAspeech, one of the largest dysarthric databases available, which is still easily an order of magnitude smaller than typical speech databases. This study investigates how far fundamental training and adaptation techniques developed in the LVCSR community can take us. A variety of ASR systems using maximum likelihood and MAP adaptation strategies are established with all speakers obtaining significant improvements compared to the baseline system regardless of the severity of their condition. The best systems show on average 34% relative improvement on known published results. An analysis of the correlation between intelligibility of the speaker and the type of system which would represent an optimal operating point in terms of performance shows that for severely dysarthric speakers, the exact choice of system configuration is more critical than for speakers with less disordered speech.

Index Terms: dysarthric speech, speech recognition, speaker adaptation

1. Introduction

Dysarthria is the blanket term for a range of disorders which arise from a loss of control of the speech articulators. It is the most common speech disorder affecting 170 per 100,000 of the population [**?**] There are a number of underlying causes: congenital conditions such as cerebral palsy or acquired neurological conditions as a result of stroke or traumatic brain injury. There is a taxonomy of dysarthrias [**?**] and there are established assessment procedures for speech and language therapists [**?**]. People with severe dysarthria can be close to unintelligible to unfamiliar listeners, though they can generally communicate successfully with family and friends. Dysarthria often co-occurs with physical disability, and the inability to use conventional keyboard-and-mouse interfaces or operate assistive technology makes speech control an attractive alternative, even though the speech is degraded.

There have been a number of small-scale studies of ASR for dysarthric speech using conventional techniques (see [**?**, **?**] and [**?**] for reviews), with patchy results. Some success has been achieved for mild-to-moderate dysarthria but there is an inverse relationship between the degree of impairment and the accuracy of 'off-the-shelf' speech recognition. Perhaps the best performance with more severe conditions has been reported by [**?**], which was based on small-vocabulary, speaker-dependent (SD) whole-word recognisers built from limited amounts of training data.

Speech recognition technology has increasingly relied on large corpora, but until recently only the Nemours database [**?**] was commonly available for research into dysarthric ASR. It contains a total of 814 sentences from 11 male speakers with mild-to-moderate impairments. In the last five years, however, this has changed with the appearance of the UAspeech database [**?**], which is used here, and TORGO [**?**]. These corpora are still small by modern LVCSR standards: the baseline recogniser used in this paper was trained on just over 170 hours of normal speech whereas UAspeech has around 18 hours[1] and the TORGO recordings amount to 23 hours, not all of which is disordered speech. Nevertheless UAspeech and TORGO make it possible to apply at least some of the modern training and adaptation techniques to dysarthric speech, and thus to arrive at phone-level context-dependent models.

The problem of very limited training data reflects the situation researchers and clinical scientists face when deploying ASR systems 'in-the-wild' for dysarthric users. People who would like to use speech-driven assistive technologies, but who have physical disabilities which also affect their speech often find it difficult to contribute a sufficient amount of data; for some dysarthric speakers supplying even a couple of minutes can be very tiring and also lead to distress, especially for people with degenerative illnesses such as Parkinson's disease.

The question is then how to best make use of such limited data. Adaptation techniques, notably maximum likelihood linear regression (MLLR) [**?**] and maximum a posteriori (MAP) [**?**] are used in LVCSR to tune speaker independent (SI) recognisers to the speech of an individual, resulting in an SD system, with a relatively small amount of adaptation data. There is, however, an assumption in these procedures that the target speech is not a gross mismatch to that used to train the SI models. The viability of this assumption for dysarthric data can be expected to depend on the severity of the condition. In [**?**], MAP adaptation was used on UAspeech data with some success and in section **??**, we compare the results of this study to ours. [**?**] report encouraging results on TORGO. Their baseline was a SI monophone model recogniser trained on a mixture of normal and dysarthric data. Acoustic model adaptation by MLLR

[1]After leading and trailing silence is cut off as described later in the paper.

resulted in a 16% absolute reduction in error rate, which was somewhat improved by speaker-dependent pronunciation models.

'In-the-field' work with ASR can be challenging at the best of times, but assistive technology users have a very high demand for reliability. If a system exhibits poor performance it may lead to the user losing confidence in the approach and he/she starting to question the strategy behind his or her assistive technology. It can take a long time for such a user to want to engage and spend time and effort with another attempt at using ASR in their home. It is therefore crucial to have informed ways of establishing - from a small amount of initial enrolment data and possibly assessments from health professionals - a suitable best 'operating point' : a task which will be useful for the user and for which the ASR will perform well.

The work described here is a two-fold step towards this goal investigating: i) to what degree does the 'optimal' system vary with the speaker, and ii) how to leverage art ASR algorithms developed and refined on typical speech. Section **??** gives a description of the data and acoustic modelling, section **??** presents results and section **??** presents conclusions.

2. Experimental setup

2.1. Data

The UAspeech database contains synchronised audio and visual streams and the publicly available part of the recordings includes speech from 15 speakers (4 female and 11 male). The speakers were asked to repeat single words from 5 groups: 10 digits, 29 Nato alphabet letters, 19 command words ('delete', 'enter' etc.), 100 common words ('the','will' etc.), and 300 uncommon words chosen to be phonetically rich and complimentary to the remaining words ('Copenhagen','chambermaid' etc.). The speakers came into the lab to complete the recordings in three blocks, and at each block, all words were repeated once, except the uncommon words where each block contained 100 unique words. In total, each speaker has produced around 70 minutes of speech. Full details of the corpora can be found in [**?**].

The speakers all have a type of disordered, dysarthric speech, and accompanying the database are percent intelligibly scores as obtained from listening tests with unfamiliar listeners. These range from 4% to 95%. Further meta-data in the form of intelligibility classes ('very low', 'low', 'medium' and 'high') are also supplied as well as broad diagnostic classes ('spastic','athetoid','mixed' and 'not diagnosed' dysarthria).

2.2. Data pre-processing

For the acoustic modelling, the data is encoded in 12-dimensional PLP features with c_0, and with added first and second order time derivatives, giving a 39 dimensional feature vector in total. Due to the way the UAspeech data was recorded (subjects sitting in front of a laptop on which prompts were presented at regular intervals), the original audio files in the database distribution contain silence, and hence this data was aligned and resegmented allowing for a 0.2 sec silence boundary around each word. This reduced the overall amount of data from around 60 hours to around 18 hours. In a real system, a good voice activity detector could provide a similar advantage.

The UAspeech database was recorded using a 7-channel microphone array. Not all channels have been supplied for each speaker in the publicly available database; for the work described here, we have chosen to use all available channels for each speaker. Following previously published work using the UAspeech for ASR (e.g. [**?**]) the data was divided into training and test data with a 2:1 split, using blocks 1 and 3 for training and block 2 for testing. Table **??** shows the amount of data available before and after alignment and resegmentation as well as the total number of word segments in each word category.

	Duration		*Number of words*				
	Org.	Rseg.	D	L	C	CW	UW
Train	39.44	12.44	1,725	4,498	3,307	17,395	17,350
Test	19.34	5.97	878	2,262	1,646	8,765	8,724

Table 1: *Duration of data [hours] and number of words in the training and test partitions after alignment. 'Org.' is the originally distributed files, 'Rseg.' is the resegmented files post alignment. The word categories are: D: digits, L:Nato alphabet, C: command words, CW: common words, UW: uncommon words.*

2.3. Acoustic modelling

All Hidden Markov Models (HMMs) were trained using the maximum likelihood (ML) criterion. State-clustered, triphones having Gaussian mixture models with 16 components per state were used.

2.4. Decoding

As the database consists of single words, it was decided to restrict the decoding so that only one word could be recognised per utterance. A uniform language model was used, as well as a word grammar network containing silence models at the start and end, and all possible test words in parallel in the middle.

Some initial testing was carried out with a decoding strategy which enabled more than one word to be output in the transcript. The results of not controlling the insertions and deletions was a fall in absolute, mean accuracy over all systems of between 2 and 4%. Deletion rates stayed <0.4% but the insertion rates were high at up towards 10%. For dysarthric speakers it is possible that the prevalence of false starts can lead to higher insertion rates. Also, it was noted that one of the speakers in UAspeech has a stammer as well as having dysarthric speech.

3. Results

Our initial work with the UAspeech database concentrated on establishing the performance of a number of baseline systems such as a speaker dependent (SD) and a speaker independent (SI) system. The speaker independent systems are trained using a 'round robin' style where in turn, the data from each speaker is held out of the training data, a SI1 model is tested on data for the speaker who was missed out. A second type of SI system was established; they differ in that this system was trained and tested on all speakers and subsequently tested with the same *known* speakers. This system is called SI2 in the following.

To compare with this, we also decoded the UAspeech test set with a typical speech model set trained on more than 177 hours of spontaneous meeting data covering an array of accented, but purely non-disordered speech; called Mtg. The Mtg models are ML trained models without CMN/CVN and are trained on the ihmtrain09 data set[**?**].

The accuracy of these four baseline systems (Mtg, SD, SI1 and SI2) are presented in columns 1 to 4 in table **??**. The results

Intellibibility	Speaker	Baseline				MAP			Domain		MAPviaDomain	
		1 Mtg	2 SI1	3 SI2	4 SD	5 mapMtg	6 mapSI1	7 mapSI2	8 domSI1	9 domSI2	10 mvdSI1	11 mvdSI2
Very low	M04 (2%)	2.1	4.1	6.1	4.9	1.4	6.1	8.3	3.2	3.8	3.1	3.5
	F03 (6%)	1.4	5.5	18.2	17.5	8.6	21.4	23.0	2.9	10.1	15.8	13.7
	M12 (7%)	0.6	3.2	8.2	9.0	5.8	12.4	11.7	4.1	8.5	10.2	9.9
	M01 (17%)	2.1	12.9	23.7	18.9	13.3	29.0	29.8	7.3	16.6	20.7	19.6
Low	M07 (28%)	3.3	20.3	62.7	66.4	34.8	68.2	66.9	10.5	40.3	43.0	40.8
	F02 (29%)	3.5	7.0	30.7	29.6	24.6	36.1	36.9	6.4	23.4	30.7	30.1
	M16 (43%)	13.0	22.2	51.1	53.6	27.8	50.1	49.3	21.9	39.8	29.8	37.3
Mid	M05 (58%)	4.9	20.2	45.4	56.4	20.6	49.9	53.4	11.2	30.6	28.5	26.9
	M11 (62%)	10.2	30.3	47.4	48.2	23.0	53.9	53.0	17.8	33.1	31.5	29.5
	F04 (62%)	22.5	30.8	61.6	53.7	43.3	62.0	65.6	30.8	51.6	49.2	53.5
High	M09 (86%)	33.6	50.2	79.5	79.1	71.5	82.4	81.5	44.6	70.0	74.7	75.1
	M14 (90%)	49.6	58.1	73.6	74.9	74.9	76.6	74.9	61.6	73.1	75.6	77.8
	M10 (93%)	62.1	68.5	83.2	81.2	86.3	87.6	86.2	70.0	85.7	87.2	87.3
	M08 (95%)	57.2	64.9	81.2	85.0	76.8	83.6	81.8	68.5	80.6	78.9	78.4
	F05 (95%)	70.2	46.6	85.9	85.6	89.1	89.1	89.6	65.0	89.6	90.2	89.2
Average		22.4	29.7	50.6	50.9	40.1	53.9	54.1	28.4	43.8	44.8	44.6

Table 2: *Word accuracy rates from baseline and MAP adaptation systems by speakers. Speakers are ordered according to their intelligibility, presented in parentheses by the speaker id. All systems are tested with the UAspeech test set; the accuracy presented is the per speaker accuracy. The coloured cells indicates the system with the highest performance for that speaker. System name descriptions:'Mtg': Typical speech meeting models, 'SI1': speaker independent models; 'SI2' : speaker independent models tested with known speakers, i.e. training data from the test speaker is present in the training data; 'SD': speaker dependent models; 'mapMtg': MAP adaptation of typical speech meeting models; 'mapSI': MAP adaptation of SI models; 'domSI': domain adaptation mtg models to SI data; 'mvdSI': MAP adaptation via SI domain models.*

for each speaker are presented row-wise and ordered according to the intelligibility rating (given in parentheses); the final row of the table gives the average accuracy across all speakers. Of the four baseline systems, the Mtg models are clearly not well matched (average accuracy of 22.4 %) as no dysarthric speech has been used during training. The SI1 system shows better performance at 29.7%, but not until any speaker specific speech is used in the training, does the average accuracy rise significantly. The SI2 and SD systems have average accuracies of 50.6% and 50.9% respectively. Two observations can be made: the SI2 and SD have used vastly different amounts of data (∼12 hours vs. ∼1.2 hours), and if the accuracies are compared per speaker the best-performing model type varies.

The next step was to investigate ways of using MAP adaptation with the available data; initially through a simple MAP adaptation from the Mtg, SI1, and SI2 models. Those results are presented in table **??**; columns 5 (mapMtg), 6 (mapSI1), and 7 (mapSI2)) with average accuracies of 40.1%, 53.9% and 54.1% respectively. This represents relative increases in accuracies of around 80% for both the Mtg and SI1 - a very clear benefit of adding speaker specific data. In contrast, the SI2 to mapSI2 relative improvement is only 7%.

Comparing individual speaker performances for the three map systems (columns 5, 6, and 7) shows that – especially for less severely dysarthric speakers – good, >70% accuracies can be achieved using both typical speech and purely dysarthric speech as starting points. The idea that perhaps for certain speakers there would be an advantage gained from harnessing the statistics present in a large LVCSR database led us to our final set of experimental systems. For these systems, the starting point was the meeting data models ('Mtg'); these models were then adapted to the general domain of dysarthric speech using the UAspeech data (domSI1/domSI2). The final step was

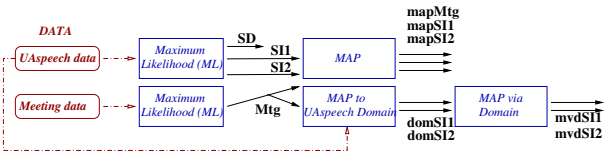


Figure 1: *Illustration of training strategies. 'ML': training using maximum likelihood criterion, 'MAP': using MAP adaptation to get SD systems, 'Domain': using MAP adaptation to get SI domain model, 'mvdSI': MAP adaptation to SD model via domain model.*

a standard MAP adaptation via this domain models resulting in an SD model (mvdSI1/mvdSI2). This MAP-via-domain training and adaptation strategy is illustrated in figure **??**.

In table **??** the results of testing each stage in the MAP-via-domain path is presented in columns 8-11. Overall, the average recognition accuracies do not compare to the mapSI systems' performance, but the average masks a very high degree of speaker variability. The colour coding in table **??** indicates the best system for each speaker; it is evident that the best choice of system varies hugely.

This is analysed in more detail in figure **??**, which shows a grey-scale visualisation of the dependance between severity of dysarthria and the accuracies from table **??**. It gives an impression of the degree to which there exists an optimal system for each speaker. Speakers are ordered by increasing intelligibility down the y-axis, and systems are ordered by increasing average, *normalised* accuracy along the x-axis. If we assume that all speakers were to exhibit a linear dependance between average system accuracy and their individual performance, the

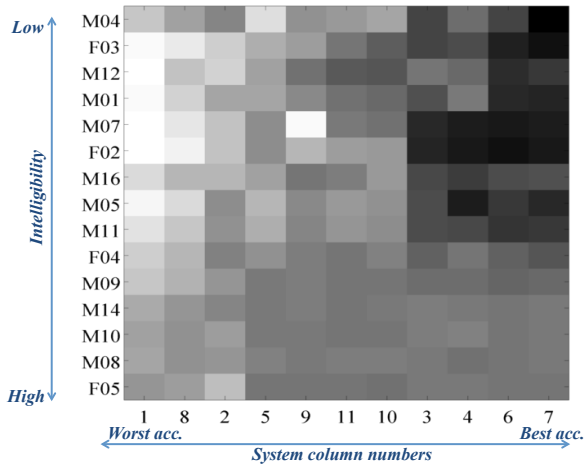


Figure 2: *Visualising the correlation between severity of dysarthria by axis and the performance of systems; image of normalised accuracies by speakerIDs and system columns from Table* **??**. *Speakers are ordered by increasing intelligibility and systems are ordered by increasing average accuracy.*

image would have increasingly darker grey colours from left to right. However, this is far from the situation. Take the case of less severe dysarthric speakers first (bottom rows). It can be seen on the limited colour variation, that their performances tend to vary little across systems. In contrast, speakers with severely disordered speech, display a strong bias towards a few systems (seen by the darker top, right-hand corner).

4. Discussion and conclusions

This paper has explored the extent to which core LVCSR training and adaptation algorithms can deal with dysarthric speech. Through a comparative study using the UAspeech database very competitive word accuracy rates have been achieved compared to results published elsewhere. [**?**] report results on a subset of 7 UAspeech speakers of which there is a 6 speaker overlap with the released data used in this study. Their MAP adaptation results are lower than ours in all cases bar for speaker F02. The results in this paper improve with an average, relative rate of 34.5% on theirs, ranging between -0.2% (F02) and 97.6% (M04). Some notable differences which may account for that difference are the cropping of extra silence from the data prior to training, and the use of highly optimised clustering, training and adaptation scripts arising from many years of LVCSR research on *typical* speech.

This leads us to conclude, in answer to our opening question, that porting LVCSR evolved on typical speech to the domain of disordered speech is a viable way of achieving good results despite the inherent differences. Although the mismatch in domains is large – exemplified by the poor performance achieved on the typical speech baseline – MAP estimation can deal with it to a large extent. For some speakers though, opting for a pure SD is a better option. In general, the study has shown how there is no 'one solution to fit all'. Particular for more severely dysarthric speakers. In future work, we plan to investigate systematic ways of arriving at the best operating point for a particular speaker in terms of system configuration. We will also investigate the effect of doing MLLR adaptation and discriminative training. This research will be supported by a longitudinal study into the use of speech-driven assistive technology for disabled and elderly users, in the homeService project, part of the UK EPSRC Programme Grant NST[**?**].

5. Acknowledgements

The research leading to these results was supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

JOURNAL OF THE ROYAL SOCIETY Interface

J. R. Soc. Interface (2012) **9**, 1517–1528
doi:10.1098/rsif.2011.0750
Published online 25 January 2012

Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination

Nathan F. Lepora[1,*], Charles W. Fox[1], Mathew H. Evans[1], Mathew E. Diamond[2], Kevin Gurney[1] and Tony J. Prescott[1]

[1]*Department of Psychology, University of Sheffield, Sheffield S10 2TP, UK*
[2]*International School for Advanced Studies (SISSA), Trieste, Italy*

Texture perception is studied here in a physical model of the rat whisker system consisting of a robot equipped with a biomimetic vibrissal sensor. Investigations of whisker motion in rodents have led to several explanations for texture discrimination, such as resonance or stick-slips. Meanwhile, electrophysiological studies of decision-making in monkeys have suggested a neural mechanism of evidence accumulation to threshold for competing percepts, described by a probabilistic model of Bayesian sequential analysis. For our robot whisker data, we find that variable reaction-time decision-making with sequential analysis performs better than the fixed response-time maximum-likelihood estimation. These probabilistic classifiers also use whatever available features of the whisker signals aid the discrimination, giving improved performance over a single-feature strategy, such as matching the peak power spectra of whisker vibrations. These results cast new light on how the various proposals for texture discrimination in rodents depend on the whisker contact mechanics and suggest the possibility of a common account of decision-making across mammalian species.

Keywords: biomimetics; perception; decision-making; Bayes' rule; whiskers; texture

1. INTRODUCTION

The last 15 years has seen major advancement in the understanding of human and animal perception as statistically optimal inference from noisy and ambiguous sensations. This statistical approach is based on using Bayes' rule to calculate the conditional probability distribution over possible percepts given sensory data, knowing aspects of the world such as the likelihood of sensory data for various percepts and their prior probabilities of occurring [1,2]. In neuroscience, there has been parallel progress in understanding perceptual decision-making as evidence accumulation for competing hypotheses. One notable line of experiments considers neuronal activity in parietal cortex as monkeys make perceptual judgements about the direction of motion for a group of random dots, and finds individual neurons that noisily ramp-up their firing rates until reaching a threshold when a decision is made [3,4]. Theoretically, these processes appear well described by the statistical approach of sequential analysis [5,6], which applies Bayes' rule to accumulate evidence for competing perceptual hypotheses over time series of sensory data until reaching a preset threshold [7].

This article aims to help develop a paradigm in rodents for testing this Bayesian approach to perceptual decision-making from the dual perspectives of guiding biological experimentation and testing biological hypotheses in robots with biomimetic sensors. Our particular focus is on texture discrimination using vibrissae, which is both a well-developed approach for examining decision-making in rats [8–16] and a task for which the state-of-the-art sensors based on rat whiskers are under continued development [17–24]. Furthermore, in biological systems, there are several proposals for which features of whisker motion vary according to texture, for example the resonance hypothesis [16] and the kinetic signature hypothesis [15]. However, it is not known how these proposals relate to theories of perceptual decision-making via evidence accumulation and how they would function in practice when embodied in a biologically inspired robot.

Our overall hypothesis is that Bayesian sequential analysis can also account for texture discrimination in rats, and thus offer a common account of decision-making in different mammalian species. The first part of this hypothesis is that the brain makes perceptual decisions by applying Bayesian inference to time series of sensory data over the motion of the whisker contacts, by comparing posterior probabilities (or functions thereof) with predefined thresholds. The second part of this hypothesis is that the brain uses simplifying assumptions about the sensory data to reduce the complexity of the neural computation. In particular, the principal assumption underlying sequential analysis is that the data samples are independently distributed over time and drawn randomly from sampling distributions associated with the likelihood functions.

*Author for correspondence (n.lepora@sheffield.ac.uk).

Received 1 November 2011
Accepted 4 January 2012

This journal is © 2012 The Royal Society

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

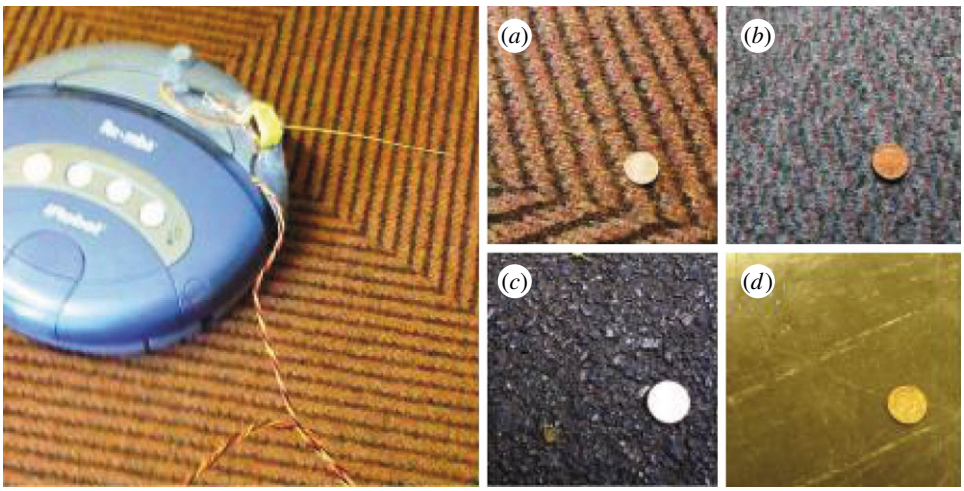


Figure 1. The robot with whisker attachment. A Roomba robot was used as a platform for the experiments. The whisker was mounted in the front of the robot, angled down to make constant contact with the floor. The panels show the various textured surfaces: (*a*) rough carpet; (*b*) smooth carpet; (*c*) tarmac and (*d*) vinyl. (Online version in colour.)

In computational neuroscience, it is commonly assumed that these sampling distributions are Gaussian [5], but we find that this assumption is too restrictive for a whiskered robot sensing texture. Instead, we assume that both the robot and animal are able to encode more general probability distributions of sensed data.

These proposals for rodent decision-making are investigated in a physical model of the rat whisker system consisting of a robot equipped with a biomimetic vibrissa sensor. Our particular platform is a Roomba robot (iRobot, Boston, MA, USA) with attached whisker module (figure 1), an ideal device for performing whisker-based experiments on surface texture because the robot can either move autonomously or be guided [25]. This investigation reveals how biomimetic principles can guide the development of new technologies and in turn provide a greater understanding of the biological systems. In particular, we conclude that decision-making with variable reaction times based on Bayesian sequential analysis out-competes the existing alternative methods, including both non-Bayesian and maximum-likelihood (fixed-response time) classifiers. A second observation is that probabilistic perception uses whatever aspects of the contact dynamics help discriminate the alternatives, so that the various proposals for how whisker motion varies according to texture could all give reliable classification depending upon the contact mechanics of the whisker and the surface.

Some partial results have been published in a conference paper on feature-based classifiers [25], and a preliminary version of the probabilistic analysis [26] based on a similar method to the maximum-likelihood classifier discussed here (but referred to as 'naive' Bayes).

2. TEXTURE SENSING WITH BIOMIMETIC WHISKERS

Touch sensors inspired by mammalian vibrissae have been in development since the 1980s [17]. Recently, biomimetic whisker sensors have been engineered to have shape and material properties similar to those of rat whiskers, while scaled to larger sizes appropriate for autonomous robots [21,27]. Another recent innovation is that the whisker deflections are transduced into sensory signals with a miniature Hall-effect sensor mounted at the base of the artificial whisker shaft [28]. Arrays of these whiskers have been employed in robots based on the rat whisker system [27,28] and could serve a variety of functions on mobile robots [17,19,22].

The biomimetic whisker was mounted in front of the Roomba robot (figure 1) at 45° azimuth from the forward direction of travel with a slight downward elevation sufficient to make constant contact with the floor during movement. Outputs from the whisker sensor included two voltages, x and y, with magnitudes linearly proportional to the tangential component of the two orthogonal displacement angles of the magnet from its resting position (with the x-component parallel to the contact surface and the y-component normal to it). As the robot moved, the whiskers were swept across the floor. The deflections of the whisker were acquired at a rate of 2 kHz and sent to a computer through the BRAHMS execution framework [29] for analysis in MATLAB (Mathworks, Natick, MA, USA).

Four surfaces were chosen for classification: two carpets of different roughnesses, a tarmac surface and a vinyl surface (figure 1). These surfaces were chosen because they were appropriately generic for a real-world experiment and they provided a range of surface types that were sufficiently similar to make classification non-trivial. Two primary behavioural conditions were also chosen: where the robot moved in a stereotyped manner, rotating either anticlockwise only, or clockwise only (four trials with a motion of 16 s each) and where the robot moved autonomously using its motion guidance, consisting of externally unpredictable clockwise, anticlockwise and forward motions (four trials of 16 s[1]). Data from the four floor surfaces are shown in

[1]Trials 13–16 of Evans *et al.* [25]. Trials 9–12 were rejected because of a systematic change in whisker baseline position during trial 12 owing to the whisker module becoming detached from the robot (see also [26], figure 7).

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

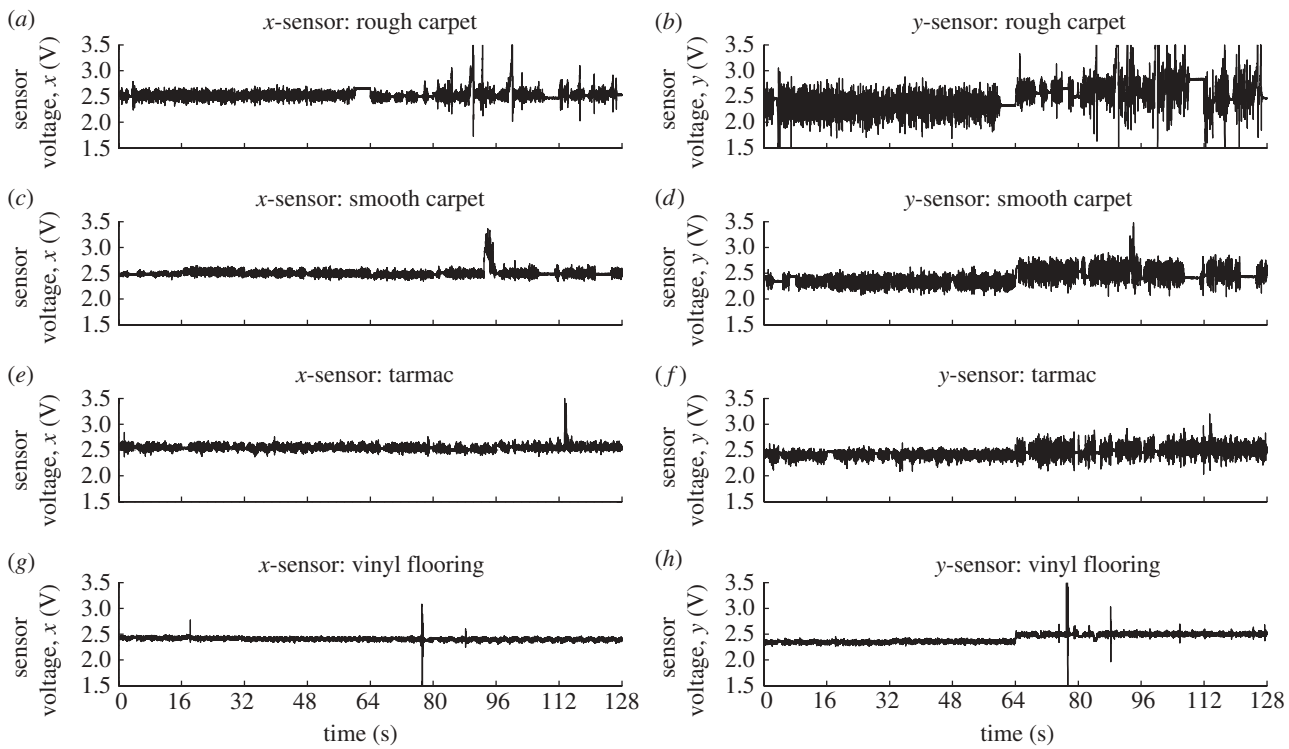


Figure 2. Stereotyped motion data. Data of the four floor surface textures were collected in eight trials each of length 16 s. The first four trials were for anticlockwise rotating motion and the last four trials were for clockwise motion.

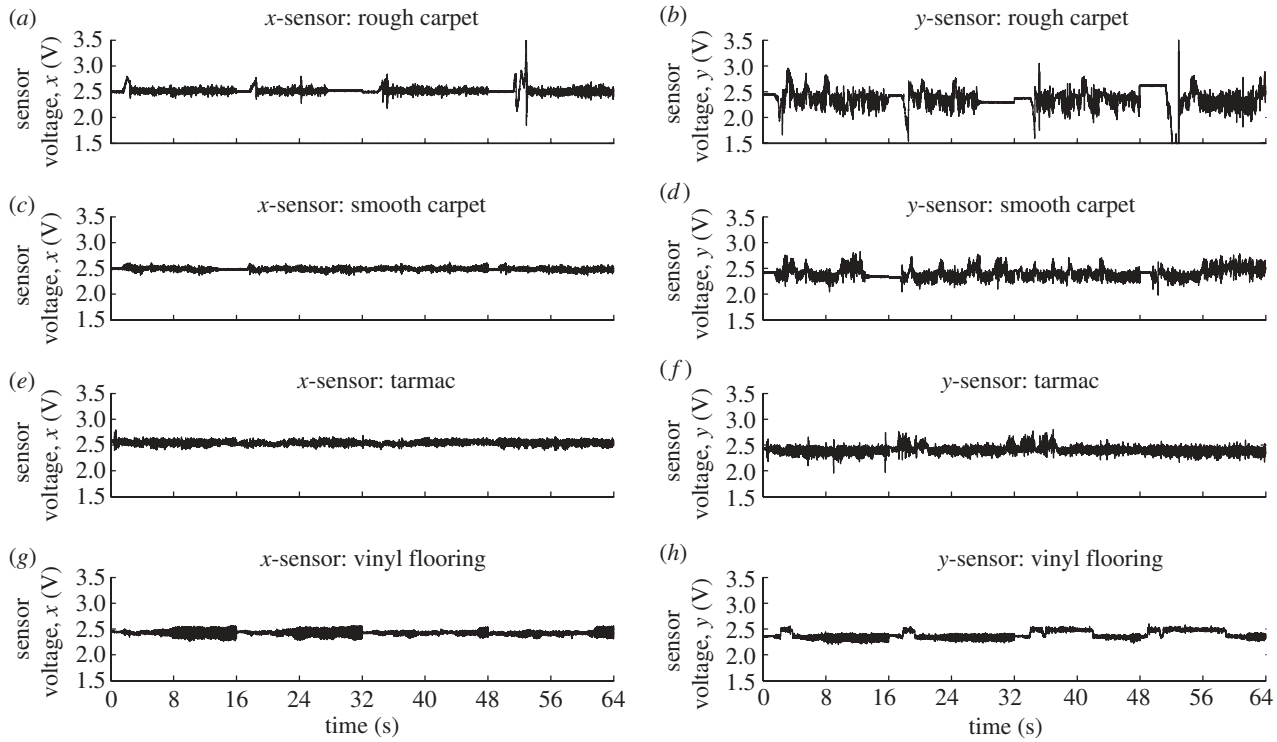


Figure 3. Autonomous motion data. Data collection as for figure 2, but with the robot moving autonomously in a series of externally unpredictable anticlockwise, clockwise and forward motions.

figure 2 for the stereotyped motion and figure 3 for the autonomous motion.

3. PROBABILISTIC CLASSIFICATION AND DECISION-MAKING

To introduce the relation between decision-making in neuroscience and inference over sequentially sampled sensory data, we recall work by Gold & Shadlen [5,30] on the decision-making of two alternative forced choices. They argued that electrophysiological recordings from lateral intraparietal association cortex in awake behaving monkeys [4] match well with hypothesis testing by the sequential probability ratio test. The log-likelihood ratio (log LR), is central to this interpretation because it indicates whether alternative H_1 or H_2

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

is supported by a sample of sensory data s, as follows from using Bayes' rule $p(H_k|\,s) = p(s\,|\,H_k)p(H_k)\,/\,p(s)$ to derive:

$$\left.\begin{aligned} \log \mathrm{PR}(s) &= \log \mathrm{LR}(s) + \log \mathrm{HR}, \\ \log \mathrm{PR}(s) &= \log \frac{p(H_1|s)}{p(H_2|s)}, \\ \log \mathrm{LR}(s) &= \log \frac{p(s|H_1)}{p(s|H_2)} \\ \text{and} \qquad \log \mathrm{HR} &= \log \frac{p(H_1)}{p(H_2)}, \end{aligned}\right\} \tag{3.1}$$

where log PR is the log posterior ratio and log HR is the log prior ratio. For a known likelihood ratio function of the sampling distributions, the decision of whether hypothesis H_k is supported to a given (log) reliability $\boldsymbol{\Theta}_k$ is determined by threshold-crossing, such that if $\log \mathrm{PR}(s) \gtrless \boldsymbol{\Theta}_k$, then H_k is supported (corresponding to upward and downward threshold crossing, as depicted in the study of Gold & Shadlen [5], figure 2*b*). Considering many such independent identically distributed samples, the log LR becomes a sum of individual terms with the threshold-crossing rule determining when there is sufficient evidence to make a decision:

$$\log \mathrm{LR}(s_1, \ldots, s_n) = \sum_{i=1}^{n} \log \mathrm{LR}(s_i) \gtrless \boldsymbol{\Theta}_k - \log \mathrm{HR}. \tag{3.2}$$

The similarity of this decision process to the observed accumulation of neural activity to threshold provides strong motivation that the relation between the log LR and the log PR gives a natural way of 'trading off sensory information, prior probability and expected value to form a perceptual decision' [30]. In particular, decision-making with the sequential probability ratio test optimizes the cost of making errors plus the cost per sample time of delaying the decision [7]. Thus, for a given accuracy, it gives the fastest reaction time [5,6].

How should this Bayesian approach to decision-making be applied to multiple competing alternatives? Denoting the training data from the four choices as $\mathrm{T}_1, \ldots, \mathrm{T}_4$ (for rough carpet, smooth carpet, tarmac, and vinyl flooring, respectively), the sampling distributions for single samples s are:

$$P(s|\mathrm{T}_k) \equiv P(q(s)|\mathrm{T}_k) = \frac{n_q(\mathrm{T}_k)}{\sum_{q=1}^{N} n_q(\mathrm{T}_k)}, \tag{3.3}$$

where $n_q(\mathrm{T}_k)$ is the total number of times that the binned sample value $q(s)$ occurs over the time series and N is the number of bins. We consider pairs $s = (x,y)$ of sampled voltage data corresponding to the two-dimensional whisker deflections, which for simplicity are assumed independent so that $P(x,y\,|\,\mathrm{T}_k) = P(x\,|\,\mathrm{T}_k)P(\,y\,|\,\mathrm{T}_k)$. The likelihoods associated with these sampling distributions can then be used to define a probabilistic classifier to discriminate the four choices of textures.

Given some test data to be classified, the log posterior probability that its samples are drawn from the training data for choice T_k is found from the logarithm of Bayes' rule:

$$\begin{aligned} \log P(\mathrm{T}_k|s_1, \ldots, s_n) &= \log P(s_1, \ldots, s_n|\mathrm{T}_k) \\ &\quad + \log P(\mathrm{T}_k) - \log P(s_1, \ldots, s_n), \end{aligned} \tag{3.4}$$

where $P(\mathrm{T}_k)$ is the prior probability of the data being from texture T_k. Here, $\log\, P(s_1, \ldots, s_n)$ is a normalization term that ensures the posteriors' sum to unity, and is found by summing the likelihoods and priors over all textures:

$$\log P(s_1, \ldots, s_n) = \log \left[\sum_{k=1}^{4} P(s_1, \ldots, s_n|\mathrm{T}_k) P(\mathrm{T}_k) \right]. \tag{3.5}$$

Considering conditionally independent and identically distributed samples for each choice of texture, an estimator for the log-likelihood is a sum of individual terms with:

$$\log P(s_1, \ldots, s_n|\mathrm{T}_k) = \frac{1}{m} \sum_{i=1}^{n} \log P(s_i|\mathrm{T}_k), \tag{3.6}$$

where $m > 0$ is a normalization (see below). The log posteriors are then found by evaluating equations (3.4)–(3.6) over n samples of training data.

Note that, in practice, we have evaluated the log posteriors from the estimated log likelihoods in discrete steps (here every 20 samples, or 10 ms). Mathematically, the likelihood to the right of equation (3.6) should be normalized by the number of combinations that could give the particular histogram of measurement values $\{n_q\}$ in equation (3.3), otherwise the small values of the estimated likelihoods could present numerical floating point issues when calculating the posteriors (equation (3.5)). All log-likelihoods were thus accumulated with a normalization factor $m = 20$ in equation (3.6), equivalent to using the average log likelihood as an estimator [31].

This study compares and contrasts two types of probabilistic classifiers based on the accumulated log likelihoods (3.6). Both classifiers assume that there is no biasing from prior knowledge of the occurrence frequency of the textures, so that the prior probabilities $P(\mathrm{T}_k)$=1/4 are equal.

(i) *Maximum-likelihood classifier*. The first probabilistic classifier considers the number of samples n as a constant set in advance of the decision-making. Then the most probable choice of texture T_k to have produced the test data has the largest log posterior probability, corresponding to the maximal *a posteriori* estimate:

$$\begin{aligned} \mathrm{T} &= \underset{\mathrm{T}_k}{\operatorname{argmax}}\; P(\mathrm{T}_k|s_1, \ldots, s_n) \\ &= \underset{\mathrm{T}_k}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log P(s_i|\mathrm{T}_k) \right], \end{aligned} \tag{3.7}$$

where we have used that both the normalizing term and prior are texture-independent and can be ignored in the argmax operation. The probabilistic classification is then maximum-likelihood estimation over a fixed window of test data.

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

(ii) *Sequential Bayes' classifier*. By analogy with the sequential probability ratio test, the decision of when a choice of texture is supported to a given log reliability θ_k is determined by when a log posterior crosses its threshold:

$$\log P(\mathrm{T}_k|s_1,\ldots,s_n) > \theta_k, \tag{3.8}$$

which determines the number of samples n used for the classification. We then take the most probable choice of texture T to have generated the test data as the above maximal *a posteriori* estimate (equation (3.7)).

For two choices, Bayesian sequential analysis (equation (3.8)) is formally equivalent to the sequential probability ratio test (equation (3.2)) with $\boldsymbol{\Theta}_k = \theta_k/(1-\theta_k)$, as follows by rearranging the threshold conditions. For given thresholds, it thus gives the fastest decision times [7]. For more than two choices, there exist more complicated sequential probability ratio methods that are asymptotically optimal [32], in that they optimize the speed-accuracy trade-off as the number of samples approaches infinity. In this study, we are concerned with decision-making over finite sample numbers on data that may invalidate optimality assumptions such as sample independence. Therefore, for simplicity, we confine our treatment to the fixed duration (maximum likelihood) and probability threshold crossing (sequential analysis) decision rules, and examine their comparative performance on naturally generated whisker sensor data.

4. PROBABILISTIC TEXTURE DISCRIMINATION

An initial visual inspection indicates that the data look like noisy time series with means and variances that vary from texture to texture. All data show dead-zones and jumps where the whisker either became static (e.g. lost contact with the surface) or encountered an irregularity. For the rotating robot motion, there are systematic differences between clockwise and anticlockwise motion (first and last four trials in figure 2) because of the angle of the whisker to the robot body. Meanwhile, the autonomous motion consists of random transitions between these two states, interspersed with a third state of forward motion. All of these effects present challenges for classifying the textural data, but those that must be overcome if the classifier is to function robustly for an animal or autonomous robot in a natural environment.

4.1. Training and sampling distributions

The initial 8 s of each trial was used for training data, leaving the final 8 s of each trial for later validation of the classification algorithms. The data were then pooled under four choices of robot motion: (i) stereotyped anticlockwise motion trials 1–4; (ii) stereotyped clockwise motion trials 5–8; (iii) either stereotyped motion trials 1–8; and (iv) autonomous motion trials 1–4.

The sampling distributions associated with the texture likelihoods of the measured x- and y-sensor voltages were found for the four textures under these four robot motions (figure 4). These probabilities were calculated from binning the range of sensor voltages into 10 mV intervals and totalling the number of values in each bin. These totals represented the empirical frequencies of the samples, which when normalized by the sample numbers gave the probability distributions of the sensor values.

The sampling x-distributions (figure 4*a,c,e,g*) looked approximately Gaussian, whereas the y-distributions (figure 4*b,d,f,h*) were often non-Gaussian with a range of differences across the textures. The y-distributions also tended to be wider than the x-distributions, apparently owing to greater vertical motion of the whisker as it bumped along surface features compared with its horizontal motion relative to the robot.

In practice, it was necessary to smooth the sampling distributions to correct for bias in the training data. Without this smoothing, the few samples in the tails of the distributions led to errors in estimating the probabilities, which diminished the performance of the classifier. All sampled probabilities were thus convolved with a Gaussian smoother of width $\sigma = 100$ mV (10 intervals), which improved the classification performance while smoothing on a relatively small scale when compared with the overall spread of data.

4.2. Maximum-likelihood classification

Given data from an unknown texture, the sampling distributions plotted in figure 4 were first used to construct a texture classifier based on maximum-likelihood estimation over fixed temporal window duration [26,33]. For validation, the data were separated into discrete segments over which the texture was determined. The classifier then identified the maximum of the log-likelihood values accumulated over these sampling windows (equation (3.7)) as described in §3.

The proportion of correct classifications, or hits, was calculated for each of the four textures (figure 5*a–d*) under the four types of robot motion considered here, namely stereotyped clockwise, anticlockwise and either rotating motion, or autonomous motion under the robot's guidance system. In a previous study [26], using both x- and y-sensor voltage data was found to give the best classification (with no consistency for classification using only single x- or y-sensor data); hence, the present study considered only classification using both sensor directions. Then a feature common across all types of motion and all textures was that the reliability of the classification depended on the window duration, so that as more evidence was used, the decision became better. Roughly speaking, the reliability improved greatly as the window duration increased up to about 50–100 ms of data, and then modest gains occurred thereafter.

Comparing the hit rates for the individual stereotyped rotating motions (figure 5*a,b*), the maximum-likelihood classifier performed accurately over both anticlockwise or clockwise rotating motion. Mean hit rates were 91 and 85 per cent for a 200 ms window. Furthermore, when the classifier was agnostic to the direction of rotation (figure 5*c*), it still performed at a similar accuracy, achieving a mean hit rate of 89 per cent over a 200 ms window. Thus, by combining the likelihoods for the clockwise and anticlockwise motion, the classifier was able to perform well on either type of motion, even though the individual likelihoods

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

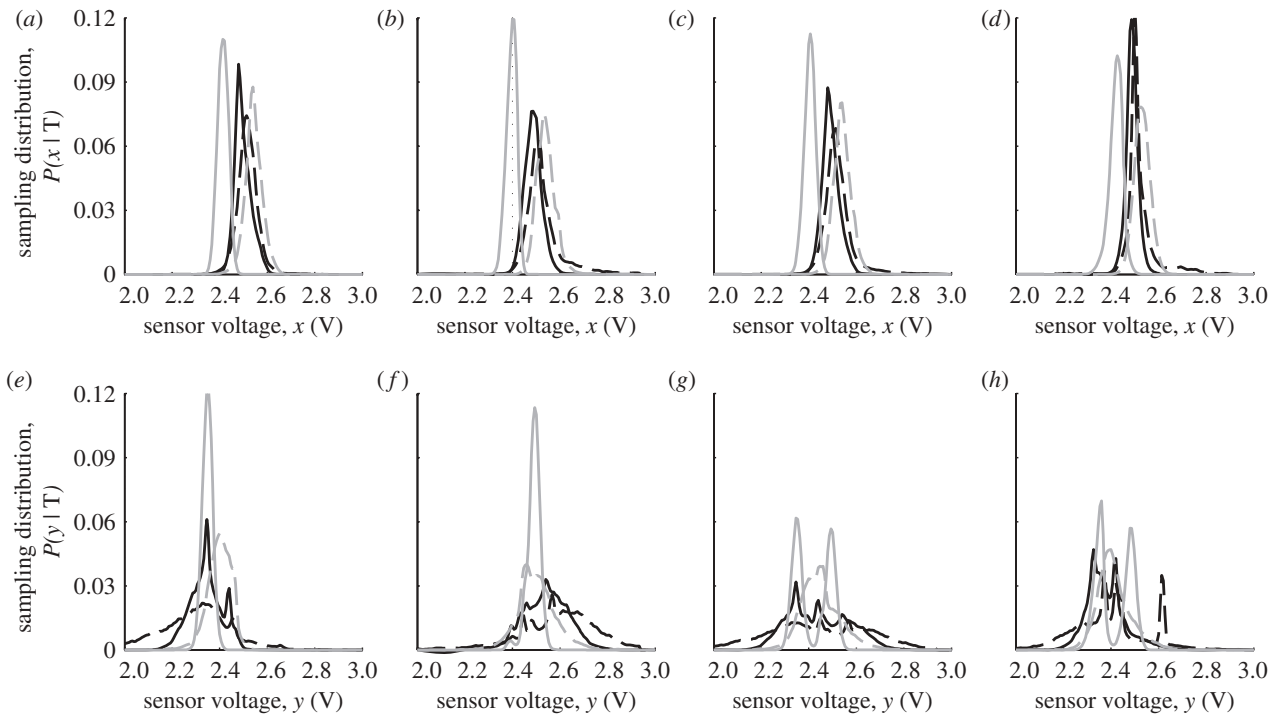


Figure 4. Texture sampling distributions. The sampling distributions associated with texture likelihoods were calculated from the empirical frequencies with which the samples occurred in training data for the four textures. (*a*–*d*) Distributions from the sensor *x*-component and (*e*–*h*) *y*-component. The horizontal groups of panels are ordered by robot motion for anticlockwise, clockwise, either rotating and autonomous motion. Black dashed line, rough carpet; black solid line, smooth carpet; grey dashed line, tarmac; grey solid line, vinyl flooring.

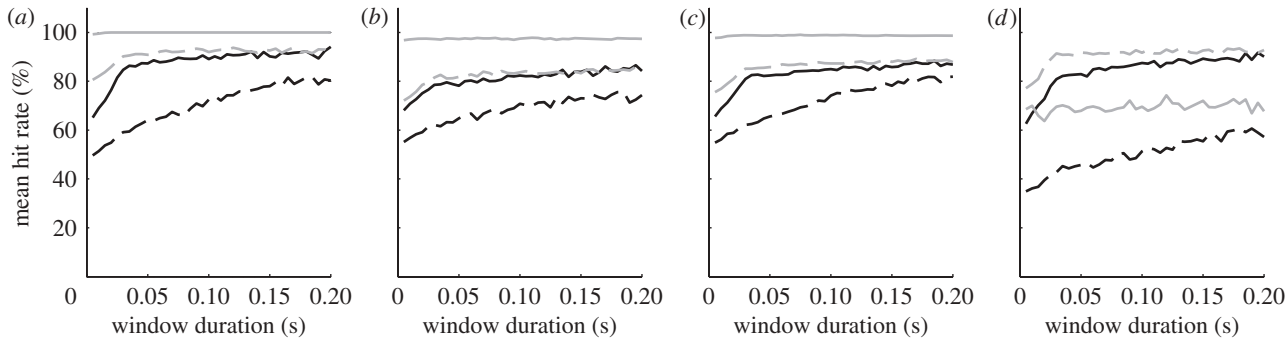


Figure 5. Hit rates for maximum-likelihood classifier. The percentages of correct classification were evaluated over validation data from the four textures for the maximum-likelihood classifier with fixed window durations. The four types of robot motion ((*a*) stereotyped anticlockwise, (*b*) clockwise, (*c*) either rotation, and (*d*) autonomous motion) were considered separately. Black dashed line, rough carpet; black solid line, smooth carpet; grey dashed line, tarmac; grey solid line, vinyl flooring.

were very different (cf. figure 4*e*, *f*). This ability to generalize by combining likelihood information was a key aspect of all probabilistic classifiers.

For the autonomous motion (figure 5*d*), the hit rates on two of the four textures (smooth carpet and tarmac) were comparable with the accuracy achieved for stereotyped motion. However, the hit rates on the two other textures (rough carpet and vinyl) were degraded to about around 60 per cent, which brought the overall mean hit rate on autonomous motion down to 77 per cent for a 200 ms window. Even so, given the considerable complexity of the autonomous motion, consisting of random anticlockwise, clockwise and forward motions, the maximum-likelihood classifier demonstrated that it can successfully generalize over these motions by combining the probability distributions.

4.3. Sequential Bayes' classification

A probabilistic classifier based on Bayesian sequential analysis was then considered. Log posterior probabilities for each of the four textures were evaluated over an increasing duration of test data, with the class given by the maximal value at the time of passing a preset (log) probability threshold (equation (3.8)). For validation, the sequential Bayes' classifier ran until it made a decision (example shown in figure 7), after which the posterior probabilities were reset to

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

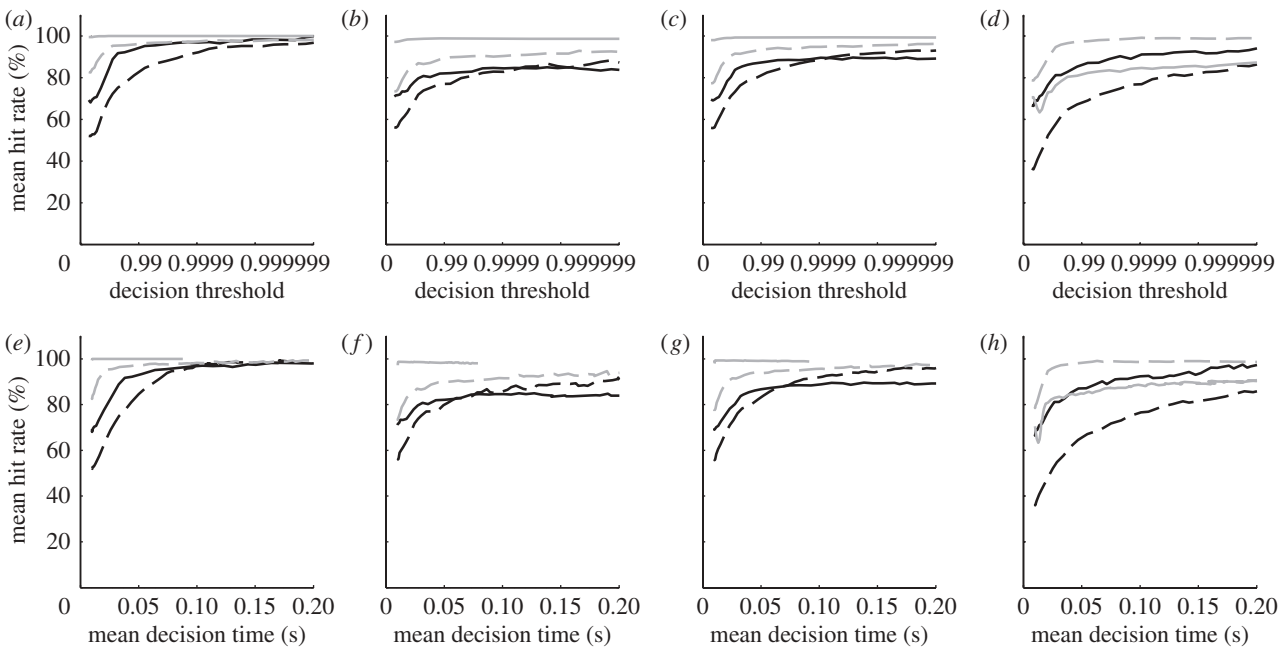


Figure 6. Hit rates for sequential Bayes' classifier. The percentages of correct classification were evaluated over validation data from the four textures for the sequential Bayes' classifier. The four types of robot motion ((*a*,*e*) stereotyped anticlockwise, (*b*,*f*) clockwise, (*c*,*g*) either rotation, and (*d*,*h*) autonomous motion) were considered separately. (*a*–*d*) Mean hit rate plotted against decision threshold and (*e*–*h*) mean hit rate against the mean decision time calculated at the same decision threshold. Black dashed line, rough carpet; black solid line, smooth carpet; grey dashed line, tarmac; grey solid line, vinyl flooring.

their flat prior values of one-quarter and the sequential classification began again. Note that this process is reminiscent of the time course of neuronal activity recorded from the parietal cortex of monkeys during a decision-making task (e.g. [5], figure 5*c*).

The hit rates were calculated for each of the four textures under the four types of robot motion (figure 6*a*–*d*; top row) for probability thresholds ranging from 0.5 to 0.99999999. Similar to maximum-likelihood classification, the accuracy improved by increasing the decision parameter, which here was the probability threshold (whereas maximum-likelihood classification varied the decision time directly). For each texture, the decision times had a distribution depending upon the chosen probability threshold (examples in figure 8), from which we plotted the mean decision time against the mean hit rate parametrized with the same probability threshold (figure 6*e*–*h*; bottom row). Unlike the maximum-likelihood classifier, these timing distributions also depended upon the test texture. For example, vinyl was relatively easy to discriminate and had quick decisions (figure 8*d*), whereas smooth carpet was more difficult and took longer (figure 8*b*). Note that these reaction time distributions are not unlike those found in humans and animals (e.g. [6], figure 1*a*).

The most striking aspect of the hit rates for the sequential Bayes' classifier was that they were substantially better than those for the maximum-likelihood classifier (cf. figures 5 and 6*e*–*h*). In particular, mean hit rates over all textures were 6–16% greater than the maximum-likelihood method for mean decision times close to 200 ms (tables 1 and 2). This is consistent because the sequential Bayes' classifier chooses the appropriate data duration to classify (increasing or reducing the sample number with ambiguity or clarity), whereas the maximum-likelihood classifier is restricted to a fixed duration.

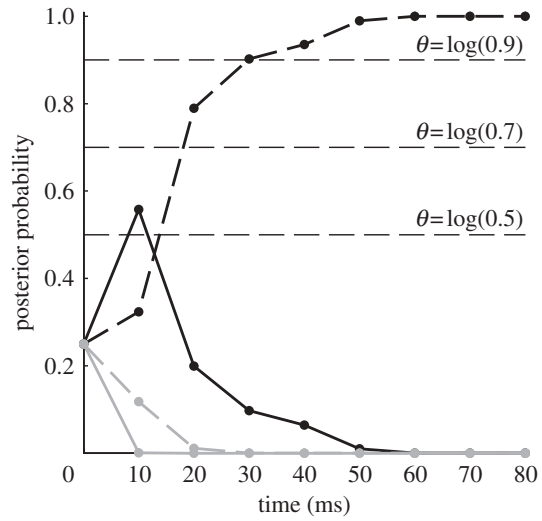


Figure 7. Example of texture discrimination for the sequential Bayes' classifier. The posterior probabilities are plotted against increasing the window duration of sampled test data (for rough carpet; texture classes denoted with the line styles from figures 4–7). Probability thresholds of 0.5, 0.7 and 0.9 are also shown, giving decision times of 10, 20 and 30 ms, respectively.

The overall accuracy of the sequential Bayes' classifier was generally as good as could be reasonably expected on noisy data with artefacts such as jumps and dead zones. For less than 200 ms of data, the average hit rates over all textures were well above 90 per cent, which is considerably better than other classification methods applied to texture data from artificial vibrissa, as discussed later. The classifier was also able to generalize over robot motions by combining likelihood information. For mean decision times of approximately 200 ms, there

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

Table 1. Maximum-likelihood classifier; 200 ms temporal window (400 samples).

robot motion	rough carpet (%)	smooth carpet (%)	tarmac (%)	vinyl flooring (%)	mean hit rate (%)
anticlockwise	80	94	91	100	91
clockwise	74	84	84	97	85
rotating	82	87	88	99	89
autonomous	57	90	93	68	77

Table 2. Sequential Bayes' classifier. (Decisions taking close to 200 ms (400 samples).)

robot motion	rough carpet (%)	smooth carpet (%)	tarmac (%)	vinyl flooring (%)	mean hit rate (%)
anticlockwise	99	98	99	100	99
clockwise	91	84	94	98	92
rotating	96	89	97	99	95
autonomous	85	97	99	91	93

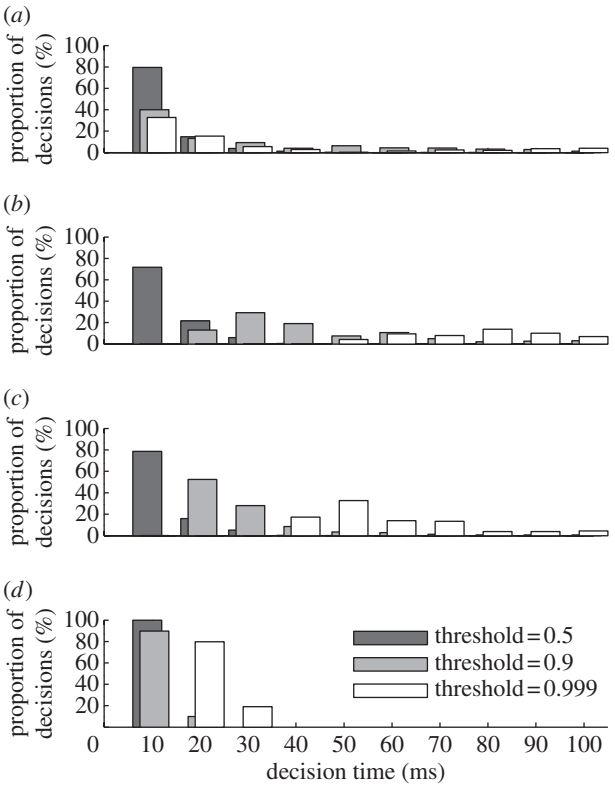


Figure 8. Example of decision times for the sequential Bayes' classifier. Histograms of the decision times are shown for each of the four textures. Data were taken from the stereotyped rotating motion, with probability thresholds 0.5, 0.9 and 0.999. (*a*) Rough carpet; (*b*) smooth carpet; (*c*) tarmac; (*d*) vinyl.

was an overall 95 per cent reliability over both anticlockwise and clockwise rotations (table 2). Meanwhile, on autonomous motion, the sequential Bayes' classifier had a 93 per cent mean hit rate (compared with 77% for the maximum-likelihood classifier).

5. COMPARISON WITH NON-PROBABILISTIC TEXTURE DISCRIMINATION

Previous studies of texture classification with artificial whisker sensors have used the frequency spectrum of the whisker signals [23–25,34–36]. The main idea is that contacts with various textured surfaces will cause whiskers to vibrate at distinct frequencies and amplitudes that are characteristic of the contacted surfaces. Here, we consider two methods of classifying the surfaces from their power spectra that have been applied previously to whisker data from mobile robots: spectral template matching [24,25] and spectral pattern recognition with a neural network [23,34,36].

5.1. Spectral template classification

Similar to §4.1, training data were taken from the initial 8 s of each trial and validation data from the final 8 s, with the selection of training and validation data identical for all classifiers considered in this article. The data were separated into 400 ms (800 sample) segments, giving 20 training and 20 validation sets from each of the 12 trials for each of the four textures. (Note that this uses at least twice the data of the probabilistic classifiers, but not favouring the spectral methods with less data led to an overly poor resolution of the power spectrum.) These 400 ms segments were then transformed to frequency space by numerical calculation of the discrete fast-Fourier transform using the Cooley–Tukey algorithm. The power spectrum was then found from the square of the absolute value of the Fourier transform value at each frequency (figure 9).

Templates for texture classification were constructed from the mean power spectra over these textures in the training data (figure 9). These power spectra peaked in the 30–40 Hz range with amplitude characteristic of the texture. Classification was then achieved by calculating the total root-mean-square error from these mean power spectrum templates for each of the four textures, with the least error specifying the chosen class [24,25]. Discriminators included that tarmac and rough carpet had large surface features (figure 1), which led to significant low-frequency power for a whisker tip moving at a few centimetres per second over centimetre-scale features. Another discriminator was the amplitude of the resonant peak near 30 Hz, which showed a clear dependence on surface type (figure 9), such as that from the friction of the whisker tip against the surface.

Results of the spectral template method on validation data were of moderate accuracy, with mean hit

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

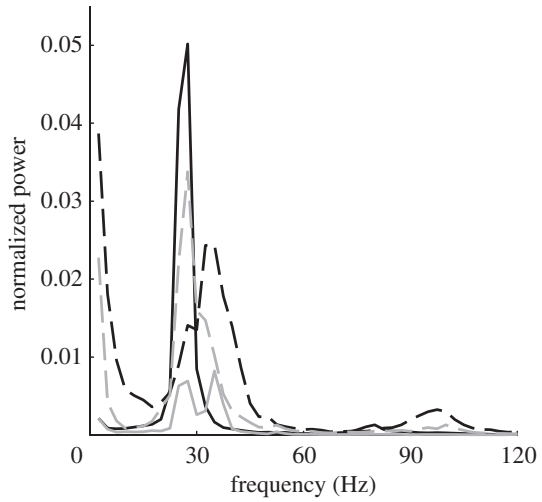


Figure 9. Power spectra of whisker signals. Whisker deflection power spectra are shown for trials 1 to 4 (anticlockwise rotating motion), which were taken over 400 ms windows and then averaged. These spectra are normalized by the mean total power for all textures. Black dashed line, rough carpet; black solid line, smooth carpet; grey dashed line, tarmac; grey solid line, vinyl flooring.

rates around 50–70% (table 3). These hit rates were about 20–40% poorer than the maximum-likelihood classifier and 30–50% poorer than the sequential Bayes' classifier.

5.2. Spectral pattern recognition with neural networks

Another method for classifying texture data from their frequency spectra is to employ pattern recognition techniques via multi-layer neural networks [23,34,36]. For proper comparison with the other classifiers, we used the same training and validation datasets employed elsewhere in this article (with 400 ms segments of whisker data sampled at 2 kHz, again favouring these methods over the probabilistic classifiers). The power spectrum was determined similarly to the spectral template method in §5.1. However, in order to reduce the number of inputs to the neural network, only the spectrum of up to 120 Hz was considered, as the power above this region is close to zero (figure 9).

It is well known that feed-forward neural networks can recognize patterns, for which they are commonly termed multi-layer perceptrons (MLPs). Here, we used an MLP with 120 inputs, 20 neurons in the hidden layer and four output neurons (one for each texture). Each neuron in the hidden and output layers had a log-sigmoid transfer function that generated outputs between 0 and 1. The hidden network weights were learnt using backpropogation and overall optimization achieved via scaled conjugate gradient descent with cost function of the root mean square error on the network output. The classification of new test data was then given by the highest output on the MLP outputs, and the hit rates equal to the proportion of correct classification to the overall number of test trials. In the following results, the network was trained 10 times and only used if it converged into a reliable classifier (all hit rates greater than 25%).

Results of the neural network classifier on validation data were of good accuracy, with mean hit rates on all types of motion around 70–80% (table 4; mean result over 10 training sessions). These hit rates were considerably higher than the spectral template matching method from §5.1. The better performance of the neural network method compared with the template matching is not surprising in hindsight, because MLPs are capable of capturing quite complicated relations between data. This does come at a significant expense of computational cost, with the template matching method taking only a fraction of a second to train and the neural network method taking many tens of seconds (on a standard 2 GHz PC with 4 Gb of RAM).

In comparison with the maximum likelihood and sequential Bayes' classifiers, the neural network classifier was about 7–17% and 14–24% less accurate, respectively.

6. DISCUSSION

We have shown that a sequential Bayes' classifier related to leading proposals for perception in animals outperforms the existing alternative accounts of texture discrimination in our physical model of the rat whisker system. In Bayesian sequential analysis, the evidence for competing hypotheses (here percepts of texture) was accumulated over time until reaching a preset threshold when a decision was made. We found that alternative, non-probabilistic classification methods considered previously [24,25,34] were generally around 20–50% poorer in accuracy than the sequential analysis, while a probabilistic method based on (fixed-response time) maximum-likelihood estimation [26,33] was about 10 per cent poorer.

The better performance of sequential analysis over maximum-likelihood estimation originated in using variable reaction times from a probability threshold for decision-making. Both of these probabilistic classifiers used an evidence accumulation framework based on log-likelihood integration assuming sampling independence over time. The maximum-likelihood method was restricted to a predefined decision time, whereas the sequential Bayes' classifier dynamically made the decision when at least one inferred posterior probability reached a preset probability threshold. Given that neuronal recordings in monkeys' making perceptual decisions suggest a mechanism of probabilistic threshold crossing [3–5] and that here we have demonstrated clear benefits for texture discrimination with artificial whiskers using a similar decision rule, we suggest that rats may also use this discrimination strategy to perceive texture.

We also found that the performance of the sequential Bayes' classifier improved dramatically as the resulting decision times increased up to 100 ms, but improved slowly thereafter. Hence, the optimal integration time in the trade-off between speed and accuracy would be around 100 ms. It is intriguing that in behaving rats, texture discrimination is achieved over a similar time frame, consisting typically of three to four contacts of about 50 ms duration each [37]. In other words, the rat may collect and analyse three to four samples of whisker vibration in order to make a fast and accurate

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

Table 3. Template classifier of power spectrum. (Fourier transform over 400 ms window (800 samples).)

robot motion	rough carpet (%)	smooth carpet (%)	tarmac (%)	vinyl flooring (%)	mean hit rate (%)
anticlockwise	65	64	54	100	71
clockwise	36	53	3	96	47
rotating	32	60	49	97	48
autonomous	6	43	83	79	53

Table 4. Neural network classifier of power spectrum. (Fourier transform over 400 ms window (800 samples).)

robot motion	rough carpet (%)	smooth carpet (%)	tarmac (%)	vinyl flooring (%)	mean hit rate (%)
anticlockwise	68	84	84	99	84
clockwise	48	56	72	94	68
rotating	68	81	79	97	81
autonomous	62	57	91	96	76

decision. Thus, the time scale is similar in optimizing biological and robot texture classification.

These results inform about sensory encoding in biological systems, specifically rodents, in their relation to various hypotheses about which features of whisker motion are important for texture discrimination. One idea is that the entire set of whiskers functions analogously to the cochlea in performing frequency analysis [16] owing to differences in vibrassae properties across the face [38], resulting in differential resonant frequencies. A related idea is that a single whisker could also use resonance as a texture cue, by considering the variation in oscillation amplitude [35] or mean velocity [13] from the resonant interaction of surface features with the intrinsic whisker dynamics. Instead, another hypothesis stresses the kinematic conversion of surface shape into trains of discrete motion events (stick-slips) by individual whiskers [15], such that those with high velocity (and high acceleration) can encode textures by their occurrence, number and possibly timing [9–12,39,40]. Meanwhile, another related view is that the mean speed of whisker micro-motion over the contact could also cue for texture [8,14,15].

In our robot study of the rat whisker system, the artificial whisker resonated when contacting natural surfaces, as was evident from both the fine detail of the whisker contact (figure 10) and the peak near 30 Hz in the frequency spectra (figure 9). The amplitude of this oscillation varied systematically with surface roughness to give a discriminator for the probabilistic classifiers from the variance of the sampling distributions associated with each likelihood (figure 4). Therefore, in the present robot study, the probabilistic classifiers used changes in resonance amplitude to discriminate texture. However, recent experimental evidence found that the power spectra of rat whisker movements during voluntary palpation of various sandpaper surfaces have little systematic variation with texture ([10], figure 6*c,d*). Taken on face value, this result suggests that the rat does not extract the same kinematic features as the robot with a biomimetic whisker and probabilistic classifier over the time series of whisker deflections.

However, closer examination does reveal some subtleties when directly comparing biological studies with the present robot treatment, which are informative about both how the biological system is configured for tactile sensing and how robots could better use artificial whisker sensors.

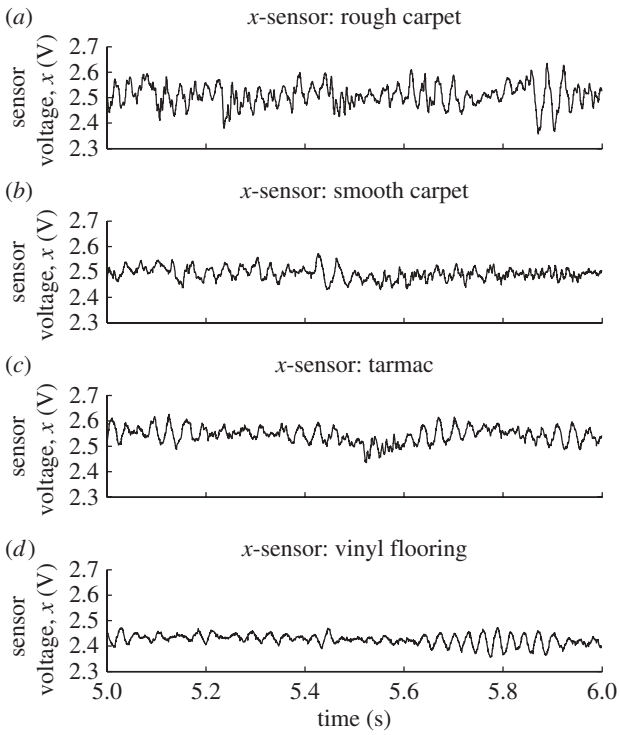


Figure 10. Whisker deflection profiles for four textures. Whisker deflection profiles are shown for trial 1 (from 5 to 6 s).

An argument against the resonance hypothesis was that the power of the whisker vibrations at the resonance frequency does not vary substantially across textures, and therefore cannot discriminate surfaces [10]. However, a visual inspection of the power spectra from the biomimetic whisker in the present study (figure 9) revealed only a small amount of systematic variation between several textures, and yet the surfaces can be reliably identified to about 70–80% with a neural network classifier of the power spectrum (table 4). Therefore, what may appear to be small differences in the overall shape of the power

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

spectra can be sufficient for reasonable classification. In our view, the most convincing way to demonstrate that the power spectra are not sufficient for reliable classification is to apply an appropriate classifier to the whisker signals, such as the neural network method considered here, which was not attempted in the original study [10].

Furthermore, the probabilistic classifiers (sequential analysis and maximum likelihood) must also use features other than the resonance amplitude to achieve better performance than classifiers based solely on the power spectra. A distinctive feature is that mean whisker positional deflection (evident in figure 4) varied systematically with surface roughness, as was also evident in the whisker deflection profiles (figure 10). Physically, as the friction between the surfaces and the whisker increased, the whisker was dragged back more strongly, which led to greater mean deflection. This signal component is not usually considered in the biological literature, even though the animal has information about whisker position at the thalamic relay to sensory cortex [41]. The contribution of this effect to the perception of texture could be investigated in behavioural experiments, for example, by checking whether rodents can discriminate surfaces of equal roughness but differing friction or by presenting demeaned artificial whisker vibrations.

In general, discrimination based on the probabilistic classifiers will use all the available evidence in the likelihood function to give the best decision. In the present robot experiment with fixed, passive biomimetic whiskers, this evidence related to the amplitude of the whisker resonance and its mean deflection, as represented in the mean and variance of the sampling distributions associated with the likelihoods. However, in rodent experiments such as those described above [10,11,40], the whiskers were actively dabbed against surfaces rather than passively brushed along them; moreover, the scale of the whiskers was such that the resonance shifted to higher frequencies (around 100 Hz rather than 30 Hz). In such circumstances, it appears from the biology that stick-slip events overtake the resonance amplitude as the dominant signal component for reliable texture discrimination [9–12,39,40].

To investigate these proposals further in a robot implementation would require a follow-up study in which the whiskers are actuated rather than mounted in a fixed position, such as that with the BIOTACT sensor which consists of a whisking head mounted on a robot arm [28]. The head of the BIOTACT sensor has multiple whiskers of differing lengths akin to rodent vibrissae, also allowing investigation of multi-whisker integration and the effect of whisker dynamics on discrimination. (For example, shortening whiskers to reduce interference between the resonance and micro-features of the surface has been reported to improve texture discrimination [23].) Direct control of the head and whiskers would also allow study of reaction-time and perception in an artificial device; for example, by manipulating the costs of making errors and delaying decisions, which are fundamental aspects of animal perception [5].

The value of this approach for understanding biological sensing is that the experimenter can design the robot to test theories motivated by the biology, whereas biology is principally an empirical science based on systems that are given. Combining both biological and engineering approaches through biomimetics, can in principle answer questions that are not answerable by biology alone [42]. The results presented here cast light on how the various previous proposals for rodent texture discrimination are dependent on the whisker contact mechanics, and suggest a number of potential lines of enquiry for future neurobiological studies. We would also anticipate that extending our biomimetic approach to robotic systems that incorporate additional aspects of biological vibrissal sensing, such as active control of whisker movement, should help resolve these issues in tactile sensing and contribute to a common account of decision-making across mammalian species.

The authors thank members of ATLAS (Active Touch Laboratory At Sheffield), the Bristol Robotics Laboratory, the BIOTACT (BIOmimetic Technology for vibrissal ACtive Touch) consortium and the anonymous reviewers. This work was supported by EU Framework projects BIOTACT (ICT-215910) and EFAA (ICT-270490), by the Human Frontier Science Programme (contract RG0041/2009-C), the Compagnia San Paolo and the Italian Institute of Technology through the BMI Project.

REFERENCES

1 Knill, D. C. & Pouget, A. 2004 The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci.* **27**, 712–719. (doi:10.1016/j.tins.2004.10.007)
2 Kersten, D., Mamassian, P. & Yuille, A. 2004 Object perception as Bayesian inference. *Annu. Rev. Psychol.* **55**, 271–304. (doi:10.1146/annurev.psych.55.090902.142005)
3 Huk, A. C. & Shadlen, M. N. 2005 Neural activity in macaque parietal cortex reflects temporal integration of visual motion signals during perceptual decision making. *J. Neurosci.* **25**, 10 420. (doi:10.1523/JNEUROSCI.4684-04.2005)
4 Platt, M. L. & Glimcher, P. W. 1999 Neural correlates of decision variables in parietal cortex. *Nature* **400**, 233–238. (doi:10.1038/22268)
5 Gold, J. I. & Shadlen, M. N. 2007 The neural basis of decision making. *Annu. Rev. Neurosci.* **30**, 535–574. (doi:10.1146/annurev.neuro.29.051605.113038)
6 Bogacz, R., Brown, E., Moehlis, J., Holmes, P. & Cohen, J. D. 2006 The physics of optimal decision making: a formal analysis of models of performance in two alternative forced-choice tasks. *Psychol. Rev.* **113**, 700. (doi:10.1037/0033-295X.113.4.700)
7 Wald, A. & Wolfowitz, J. 1948 Optimum character of the sequential probability ratio test. *Ann. Math. Stat.* **19**, 326–339. (doi:10.1214/aoms/1177730197)
8 Morita, T., Kang, H., Wolfe, J., Jadhav, S. P. & Feldman, D. E. 2011 Psychometric curve and behavioral strategies for whisker-based texture discrimination in rats. *PLoS ONE* **6**, e20437. (doi:10.1371/journal.pone.0020437)
9 Diamond, M. E. 2010 Texture sensation through the fingertips and the whiskers. *Curr. Opin. Neurobiol.* **20**, 319–327. (doi:10.1016/j.conb.2010.03.004)
10 Wolfe, J., Hill, D. N., Pahlavan, S., Drew, P. J., Kleinfeld, D. & Feldman, D. E. 2008 Texture coding in the rat whisker system: slip-stick versus differential resonance. *PLoS Biol.* **6**, e215. (doi:10.1371/journal.pbio.0060215)
11 Diamond, M. E., Von Heimendahl, M. & Arabzadeh, E. 2008 Whisker-mediated texture discrimination. *PLoS Biol.* **6**, e220. (doi:10.1371/journal.pbio.0060220)

Downloaded from http://rsif.royalsocietypublishing.org/ on November 29, 2016

12 Diamond, M. E., von Heimendahl, M., Knutsen, P. M., Kleinfeld, D. & Ahissar, E. 2008 'Where' and 'what' in the whisker sensorimotor system. *Nat. Rev. Neurosci.* **9**, 601–612. (doi:10.1038/nrn2411)

13 Ritt, J. T., Andermann, M. L. & Moore, C. I. 2008 Embodied information processing: vibrissa mechanics and texture features shape micromotions in actively sensing rats. *Neuron* **57**, 599–613. (doi:10.1016/j.neuron.2007.12.024)

14 von Heimendahl, M., Itskov, P. M., Arabzadeh, E. & Diamond, M. E. 2007 Neuronal activity in rat barrel cortex underlying texture discrimination. *PLoS Biol.* **5**, e305. (doi:10.1371/journal.pbio.0050305)

15 Arabzadeh, E., Zorzin, E. & Diamond, M. E. 2005 Neuronal encoding of texture in the whisker sensory pathway. *PLoS Biol.* **3**, e17. (doi:10.1371/journal.pbio.0030017)

16 Neimark, M. A., Andermann, M. L., Hopfield, J. J. & Moore, C. I. 2003 Vibrissa resonance as a transduction mechanism for tactile encoding. *J. Neurosci.* **23**, 6499–6509.

17 Prescott, T. J., Pearson, M. J., Mitchinson, B., Sullivan, J. C. W. & Pipe, A. G. 2009 Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robot. Autom. Mag.* **16**, 42–50. (doi:10.1109/MRA.2009.933624)

18 Solomon, J. H. & Hartmann, M. J. Z. 2008 Artificial whiskers suitable for array implementation: accounting for lateral slip and surface friction. *IEEE Trans. Robot.* **24**, 1157–1167. (doi:10.1109/TRO.2008.2002562)

19 Kim, D. E. & Moller, R. 2007 Biomimetic whiskers for shape recognition. *Robot. Auton. Syst.* **55**, 229–243. (doi:10.1016/j.robot.2006.08.001)

20 Solomon, J. H. & Hartmann, M. J. 2006 Biomechanics: robotic whiskers used to sense features. *Nature* **443**, 525. (doi:10.1038/443525a)

21 Pearson, M. J., Gilhespy, I., Melhuish, C., Mitchinson, B., Nibouche, M., Pipe, A. G. & Prescott, T. J. 2005 A biomimetic haptic sensor. *Int. J. Adv. Robot. Syst.* **2**, 335–343.

22 Schultz, A. E., Solomon, J. H., Peshkin, M. A. & Hartmann, M. J. 2005 Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. In *Proc. IEEE Int. Conf. Robot. Autom.* (*ICRA 2005*) *Barcelona, Spain, 18–22 April 2005*, pp. 2588–2593. Piscataway, NJ: IEEE Press. (doi:1109/ROBOT.2005.1570503)

23 Kim, D. E. & Moeller, R. 2004 A biomimetic whisker for texture discrimination and distance estimation. In *From animals to animats 8* (eds E. Sahin & W. M. Spears), p. 140. Cambridge, MA: The MIT Press.

24 Fend, M., Bovet, S., Yokoi, H. & Pfeifer, R. 2003 An active artificial whisker array for texture discrimination. In *Proc. IEEE Int. Conf. Intelligent Robots and Systems* (*IROS 2003*) *Las Vegas, NV, USA, 27–31 April 2003*, vol. 2, pp. 1044–1049. Piscataway, NJ: IEEE Press.

25 Evans, M., Fox, C. W., Pearson, M. J. & Prescott, T. J. 2009 Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In *Proc. towards Autonomous Robotic Systems* (*TAROS 2009*) *Ulster, UK, 31 August–2 September 2009* (eds T. Kyriacou, U. Nehmzow, C. Melhuish & M. Witkowski), pp. 19–24. Ulster, UK: Ulster University Press.

26 Lepora, N. F., Evans, M., Fox, C. W., Diamond, M. E., Gurney, K. & Prescott, T. J. 2010 Naive Bayes texture classification applied to whisker data from a moving robot. *The IEEE 2010 Int. Joint Conf. on Neural Networks* (*IJCNN*) *Barcelona, Spain, 18–23 July 2010*, pp. 1–8. Piscataway, NJ: IEEE Press. (doi:10.1109/IJCNN.2010.5596360)

27 Pearson, M. J., Mitchinson, B., Sullivan, J. C., Pipe, A. G. & Prescott, T. J. 2011 Biomimetic vibrissal sensing for robots. *Phil. Trans. R. Soc. B* **366**, 3085–3096. (doi:10.1098/rstb.2011.0164)

28 Sullivan, J., Mitchinson, B., Pearson, M., Evans, M., Lepora, N., Fox, C., Melhuish, C. & Prescott, T. 2011 Tactile discrimination using active whisker sensors. *IEEE Sensors* **12**, 350–362. (doi:10.1109/JSEN.2011.2148114)

29 Mitchinson, B., Chan, T. S., Chambers, J., Pearson, M., Humphries, M., Fox, C., Gurney, K. & Prescott, T. J. 2010 BRAHMS: novel middleware for integrated systems computation. *Adv. Eng. Inform.* **24**, 49–61. (doi:10.1016/j.aei.2009.08.002)

30 Gold, J. I. & Shadlen, M. N. 2001 Neural computations that underlie decisions about sensory stimuli. *Trends Cogn. Sci.* **5**, 10–16. (doi:10.1016/S1364-6613(00)01567-9)

31 Wooldridge, J. M. 2009 *Introductory econometrics: a modern approach*. OH, USA: South-Western Publishers.

32 Dragalin, V. P., Tartakovsky, A. G. & Veeravalli, V. V. 1999 Multihypothesis sequential probability ratio tests. I. Asymptotic optimality. *IEEE Trans. Information Theory* **45**, 2448–2461. (doi:10.1109/18.796383)

33 Lepora, N. F. *et al.* 2011 A general classifier of whisker data using stationary naive Bayes: application to BIOTACT robots. *Towards autonomous robotic systems. Sheffield, UK, 31 August–2 September 2011*. Lecture Notes in Computer Science, vol. 6856 (eds R. Gross, L. Alboul, C. Melhuish, M. Witkowski, T. Prescott & J. Penders). Heidelberg, Germany: Springer. (doi:10.1007/978-3-642-23232-9_2)

34 Fend, M. 2005 Whisker-based texture discrimination on a mobile robot. In *8th European Conf. ECAL 2005, Canterbury, UK, 5–9 September 2005*, vol. 3630 (eds M. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson & J. Timmis), pp. 302–311. Heidelberg, Germany: Springer

35 Hipp, J., Arabzadeh, E., Zorzin, E., Conradt, J., Kayser, C., Diamond, M. E. & Konig, P. 2006 Texture signals in whisker vibrations. *J. Neurophysiol.* **95**, 1792–1799. (doi:10.1152/jn.01104.2005)

36 Fox, C. W., Mitchinson, B., Pearson, M. J., Pipe, A. G. & Prescott, T. J. 2009 Contact type dependency of texture classification in a whiskered mobile robot. *Auton. Robots* **26**, 223–239. (doi:10.1007/s10514-009-9109-z)

37 Zuo, Y., Perkon, I. & Diamond, M. E. 2011 Whisking and whisker kinematics during a texture classification task. *Phil. Trans. R. Soc. B* **366**, 3058–3069. (doi:10.1098/rstb.2011.0161)

38 Brecht, M., Preilowski, B. & Merzenich, M. M. 1997 Functional architecture of the mystacial vibrissae. *Behav. Brain Res.* **84**, 81–97. (doi:10.1016/S0166-4328(97)83328-1)

39 Lottem, E. & Azouz, R. 2008 Dynamic translation of surface coarseness into whisker vibrations. *J. Neurophysiol.* **100**, 2852. (doi:10.1152/jn.90302.2008)

40 Jadhav, S. P., Wolfe, J. & Feldman, D. E. 2009 Sparse temporal coding of elementary tactile features during active whisker sensation. *Nat. Neurosci.* **12**, 792–800. (doi:10.1038/nn.2328)

41 Petersen, R. S., Brambilla, M., Bale, M. R., Alenda, A., Panzeri, S., Montemurro, M. A. & Maravall, M. 2008 Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron* **60**, 890–903. (doi:10.1016/j.neuron.2008.09.041)

42 Mitchinson, B., Pearson, M., Pipe, T. & Prescott, T. J. 2011 A view from the whisker tip. In *Biomimetic robots as scientific models*. (eds I. L. Krichmar & H. Wagatsuma). Cambridge, UK: Cambridge University Press.

Tactile Discrimination Using Active Whisker Sensors

J. Charles Sullivan, Ben Mitchinson, Martin J. Pearson, Mat Evans, Nathan F. Lepora, Charles W. Fox, Chris Melhuish, and Tony J. Prescott

***Abstract*—We describe a novel, biomimetic tactile sensing system modeled on the facial whiskers (vibrissae) of animals such as rats and mice. The "BIOTACT Sensor" consists of a conical array of modular, actuated hair-like elements, each instrumented at the base to accurately detect deflections of the shaft by whisker-surface contacts. A notable characteristic of this array is that, like the biological sensory system it mimics, the whiskers are moved back-and-forth ("whisked") so as to make repeated, brief contacts with surfaces of interest. Furthermore, these movements are feedback-modulated in a manner intended to emulate some of the "active sensing" control strategies observed in whiskered animals. We show that accurate classification of surface texture using data obtained from whisking against three different surfaces is achievable using classifiers based on either naive Bayes or template methods. Notably, the performance of both these approaches to classify textures after training on as few as one or two surface contacts was improved when the whisking motion was controlled using a sensory feedback mechanism. We conclude that active vibrissal sensing could likewise be a useful sensory capacity for autonomous robots.**

***Index Terms*—Bayesian methods, biomimetics, classification algorithms, robotics, sensor systems and applications, tactile sensors.**

I. Introduction

An important characteristic of many biological sensory systems is that they are active, where "active" here means purposively controlled and information-seeking in the sense described by Bajcsy [1], rather than operating by emitting energy (which is an alternative meaning for the phrase "active sensing"). An example of biological active sensing, in the realm of touch, comes from the way we control the movement of our hands and fingers depending on the type of information about a surface or object of interest that is being sought. For instance, we typically stroke a surface with our fingertips to determine texture, palpate at different locations to judge shape, and press or squeeze to assess hardness [2]. In this paper, we discuss biomimetic tactile sensing in the context of a different, but perhaps equally important, active sensory system – the facial whiskers (or vibrissae) of mammals. Nearly all mammals have facial whiskers but the specialists in vibrissal sensing, such as rats and mice, tend to be creatures of small body-size living nocturnally, or in environments with limited natural light, who will use their whiskers in support of locomotion, navigation, prey catching, and even social behavior [3]. Vibrissal sensing is of interest to engineers as artificial vibrissae could provide proximal sensors that can determine distance to nearby objects alongside surface characteristics such as shape and texture [4]). A useful property is that vibrissal sensors could be employed in circumstances where the efficacy of other types of proximal sensing could be seriously compromised – for instance, in environments containing smoke or dust, or in darkness where there is a need to operate covertly.

The biomimetic vibrissal sensing system described in this paper has been developed as the result of a long-term collaboration between biologists and engineers aimed at advancing the understanding of biological vibrissal systems and determining to what extent, and in what forms, vibrissal sensing could be useful in engineered systems such as autonomous robots. Most recently, this collaboration has taken the form of a European Framework 7 project termed BIOTACT (BIOmimetic Technology for vibrissal Active Touch), and in this paper, we describe the design of a vibrissa-based sensory array – the BIOTACT sensor – that has been one of the chief focuses of that project. The BIOTACT sensor is the latest in a line of artificial vibrissal systems (see [4] and [5]) modeled most directly on that of the common rat. We have taken inspiration from this animal model in various ways, investigating and attempting to mimic aspects of: (i) vibrissal morphology including the physical characteristics of individual whiskers and their organization into a spatial-distributed array of different length elements; (ii) vibrissal control including the manner in which the whiskers are moved and positioned both through actuation of individual whiskers and through the ability to orient the full array; and (iii) vibrissal sensory processing including analysis of the types of tasks in which vibrissal sensing is employed by animals and the nature of the information they are thought to extract from vibrissal signals. In this paper, we consider each of these different areas of potential biomimicry. In each case, we briefly describe some of the salient characteristics of biological vibrissal systems, discuss how these have influenced the design of our artificial sensor, and show how the current sensor prototype implements key properties of interest. We also describe a number of experiments with this prototype showing, in particular, how we have implemented biomimetic vibrissal control, and how properties, such as surface texture and radial distance to contact, can be extracted from sensor signals. In the discussion, we consider the impact of active control on

Manuscript received November 19, 2010; revised March 31, 2011; accepted April 08, 2011. Date of publication April 29, 2011; date of current version December 07, 2011. This work was supported by the European Union FP7 BIOTACT Project (ICT-215910). The associate editor coordinating the review of this paper and approving it for publication was Prof. Raul Martin-Palma.

J. C. Sullivan, M. J. Pearson, and C. Melhuish are with the Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QD, U.K. (e-mail: charlie.sullivan@brl.ac.uk; http://www.brl.ac.uk; martin.pearson@brl.ac.uk; chris.melhuish@brl.ac.uk).

B. Mitchinson, M. Evans, N. F. Lepora, C. W. Fox, and T. J. Prescott are with the Department of Psychology, University of Sheffield, Sheffield, S10 2TP, U.K. (e-mail: b.mitchinson@sheffield.ac.uk; Mat.Evans@sheffield.ac.uk; n.lepora@sheffield.ac.uk; charles.fox@gmail.com; t.j.prescott@sheffield.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2011.2148114

1530-437X/$26.00 © 2011 IEEE

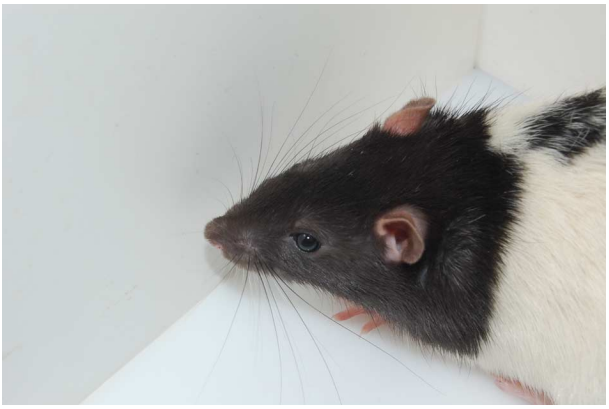


Fig. 1. Rat facial whiskers (*vibrissae*).

the information-gathering capacities of our novel sensor in the context of our twin goals of furthering the understanding of biological vibrissal sensing and developing useful artificial vibrissal sensors for robotics. Finally, we outline some of the limitations of our current sensor design and describe how these might be overcome in the future.

II. The Biotact Sensor

A. The Whisking System in Rat

Long facial whiskers, or *macrovibrissae*, are found in many mammalian species, projecting outwards and forwards from the snout of the animal to form a tactile sensory array that surrounds the head (see [6], and Fig. 1). In rats, the macrovibrissae form a two-dimensional grid of five rows on each side of the snout, each row containing between five and nine whiskers ranging between $\sim 15\ \mathrm{mm}$ and $\sim 50\ \mathrm{mm}$ in length. Each whisker is mounted in a specialized and mechanically complex hair follicle (see [7] for a description). These, in turn, sit within an area of dense muscular tissue strongly innervated by nerve fibers called the mystacial pad. From an engineering perspective, facial whiskers can be thought of as tapered elastic beams that deform easily upon contact with objects, and possibly also as the result of movement through air. Since the whisker shaft is composed of dead hair cells, the transduction of whisker deformations into neural activity begins inside the follicle where multiple populations of mechanoreceptors surround the base of the shaft and respond with high acuity when it is deflected. Mechanoreceptor responses are then transduced into patterns of neural firing in approximately 200 primary afferent fibers (per whisker) within the facial nerve. This transduction apparatus has been found to respond, in different ways, to rotation of the follicle by its muscles and deflection of the whisker shaft by external contacts. Thus, activity in facial nerve neurons is thought to encode information about the direction, velocity, and duration of whisker displacements.

When the rat is moving and exploring its surroundings the long facial whiskers are swept back and forth, at typical rates of seven to ten "whisks" per second [8]. The forward and backwards motion of each vibrissa is partly determined by its own intrinsic muscle, and partly by a set of extrinsic muscles that anchor the mystacial pad to the bones of the skull. Thus, although each whisker has some capacity for independent movement, the whiskers on each side of the snout generally move together, coupled by linkages within the pad and by the action of extrinsic muscles.

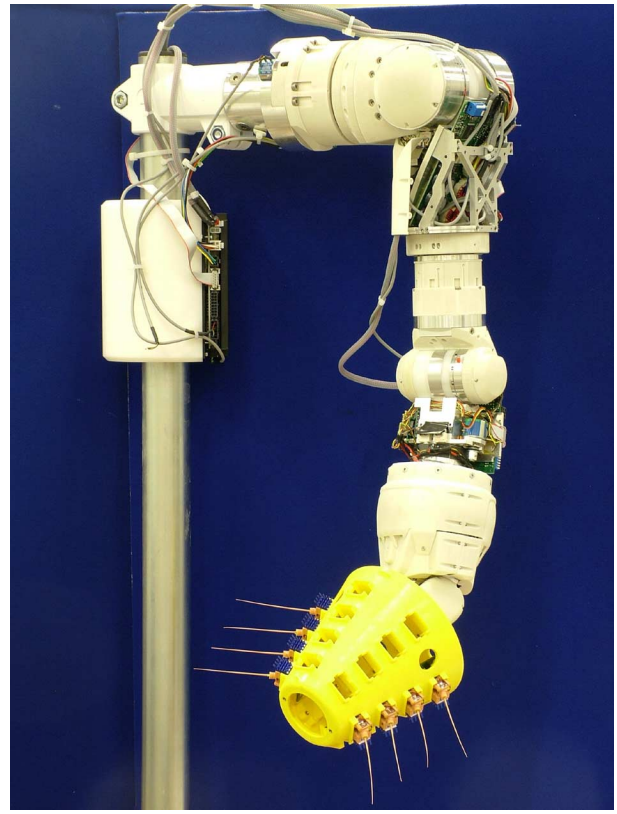


Fig. 2. The BIOTACT G1 sensor head and robot arm.

In addition to the macrovibrissae, rats also have a large number of shorter, more densely packed, and non-actuated microvibrissae on the chin and lips. These arrays of short whiskers are thought to form the "fovea" of the vibrissal system [9] being brought to bear on objects and surfaces by head and body orienting movements triggered when the longer whiskers make unexpected contacts.

B. Sensor Design

The Generation 1 (G1) BIOTACT sensor consists of a truncated conical "head" made from ABS plastic, which holds up to 24 whisker modules. The whisker modules are arranged in six radially symmetric rows of four and are oriented at outward-facing normals to the cone surface. For the purposes of our experiments and to enable accurate and repeatable location and movement of the sensor, it is mounted as the end-effector of a 7 degree of freedom robot arm (Fig. 2 shows the G1 sensor fitted with eight whiskers). Also, for the experiments reported here, the head was not fully populated with whiskers. Two rows were fitted with three whiskers each to create a bilaterally symmetric configuration.

C. Design of Active Whisker Sensor Module

The individual whisker sensors are completely modular in design, incorporating their own actuation mechanism and control electronics. Actuation is by means of miniature brushless DC motors (Faulhaber 1307BH), which incorporate an 11:1 gearbox, controlled by custom-built three-phase bridge drivers,

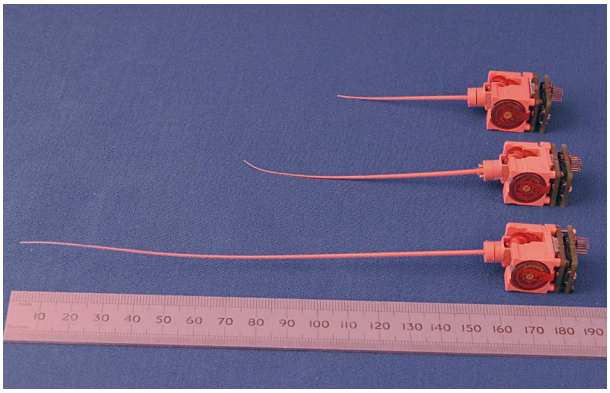


Fig. 3. Whisker modules.

with closed-loop PD control provided by on-board microcontrollers (Microchip dsPIC33FJ128GP802). The modules are 20 mm by 15 mm by 15 mm in size (see Fig. 3) and weigh 8 g including the controller boards. Typical power consumption during whisking is 0.9 W. They are capable of whisking at frequencies of up to 10 Hz, which is roughly comparable to the whisking rate of real rat mystacial whiskers [10].

Software to process the signals from the whisker sensors and to control the whisking patterns is written in C++ and executed under the BRAHMS Modular Execution Framework [11] on a PC which connects to the sensor head via a USB2 connection, running in high speed bulk transfer mode. This is able to transmit data to and from the sensor (half-duplex) synchronously at 500 μs intervals, enabling data transfer and feedback updates at 2 kHz.

The whisker sensor is based on the principle of Hall-effect detection of movement of a small magnet fixed to the base of the whisker. Monolithic triaxis Hall-effect sensor ICs by Melexis (MLX90333) are used and these incorporate two orthogonal pairs of Hall plates which, together with on-chip signal processing, enable the three orthogonal components of magnetic field strength, to be resolved with 14 bit resolution. These data, together with the protraction angle (θ) of the whisker relative to the head are sent in synchronous serial form using an SPI bus to the local microcontroller. Data from the microntrollers are collected via six separate SPI busses which are marshalled by an field programmable gate array (FPGA) (Xilinx Spartan 3e) and then sent to the first-in–first-out (FIFO) buffer of the USB controller (Cypress FX2). The FPGA acts as the bus master for the SPI busses and data transfers in packets of 512 bytes are synchronized with the USB2 host controller on the PC resulting in very low latency and jitter.

The whiskers are made using a Data Light Processing (DLP) rapid-prototyping machine (Envisiontec Perfactory) in Nanocure RC25, a composite material consisting of a UV-curing photo-polymer matrix and aluminium oxide nanoparticle filler. The combination of the DLP process and the composite material enable us to make whiskers which taper to a fine point (0.25 mm diameter) and yet retain sufficient strength and toughness to withstand repeated impact against the discriminanda. Two important benefits of whisker taper, and in particular the fine tip diameter were highlighted in [12]: a small tip should improve resolution of fine surface features (here, we are attempting to discriminate surface textures with mean particle diameters of 25 μm) and tip breakages (which are common for real rat whiskers) have a smaller effect on natural frequencies of vibration. The artificial whiskers have a taper ratio (hub/tip) of 8, whereas rat whisker taper ratios are typically 10–20 with tip diameters of ~ 10 μm. Rat whiskers are both tapered and curved but we have chosen not to mimic the curvature of the natural whiskers in these experiments. Future work will examine the effect that curvature has on the ability to discriminate texture.

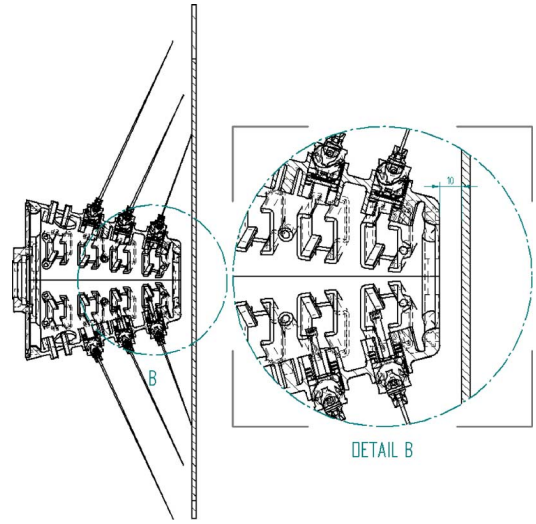


Fig. 4. G1 sensor in test configuration (section).

Note also that the whisker lengths in the test configuration were not all the same: they increased from front to rear in order that all whiskers could touch a plane surface orthogonal to the axis of the cone (see Fig. 4). This feature was also motivated by the observation that the whiskers within each row of the mystacial pad of rats and other whisking mammals closely follow an exponential increase in length from the front to the rear: the actual ratio of length between adjacent whiskers is species and row specific, varying from 1.2 to 1.6 [9]. In our experiments, we used whiskers of length 80, 112, and 158 mm, corresponding to a ratio of 1.4. Our whiskers were longer than those of whisking mammals, which in rats are typically between 15 and 50 mm long [3]. The lowest static natural frequencies of vibration of the artificial whiskers were measured *in situ*, using Fourier spectral analysis of response after release from an imposed deflection parallel to the motor axis, as 86, 50, and 21 Hz, respectively. Frequencies of corresponding whiskers on each side were within 1 Hz. Studies of the mechanical characteristics of rat whiskers have revealed that they have natural frequencies in the same range. Hartmann *et al.* [13], for example, measured frequencies between 27 and 260 Hz, noting that in all cases whisker resonant frequencies were significantly higher than the whisking frequency.

III. Motor Control

A. Motor Control in Rats

Although early research emphasized the regularity and symmetry of whisker motion (e.g., [14]), more recent work by the current authors, and in other laboratories, has shown that biological whisker motion is often strongly modulated by the nature of the task [15], by the intentions of the animal [16], and by

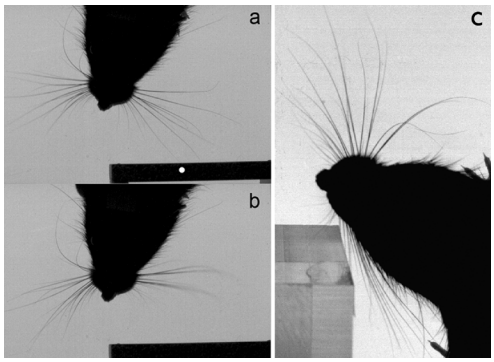


Fig. 5. Snapshots from high-speed videography recordings of rat whisker movement. (a) and (b) An example of rapid cessation of protraction (RCP). In (a), the left and right whisker fields are protracting synchronously when one of the forward whiskers in the ipsilateral (left-hand) field makes contact (indicated by the white dot) with a vertical surface. Protraction on the ipsilateral side ceased 12 ms later while protraction continued contralaterally (adapted from [17]). In (b), 32 ms after the contact, the ipsilateral whiskers are already considerably retracted when the contralateral field reaches maximum protraction. (c) An example of contact-induced asymmetry (CIA). Here, the whiskers ipsilateral (nearest) to the Perspex block can be seen to be considerably less protracted than those contralateral (image from the Active Touch Laboratory).

the immediate environment [17], [18]. It has been hypothesized (for example, see [17]) that some of these modulations are functional, and more specifically, that they are examples of "active sensing" control, serving to improve the quality and/or quantity of sensory information collected. The BIOTACT sensor is an ideal platform for testing these hypotheses: its great number of degrees of freedom allow the comparison of the impact of different motor control strategies on data collection. In the current paper, we focus on two candidate active sensing strategies, termed rapid cessation of protraction (RCP), and contact-induced asymmetry (CIA).

The forward and backward phases of the periodic "whisking" motions, observed in rats and mice, are usually denoted protraction and retraction, respectively. RCP is the observation that when the animal's whiskers encounter a novel object they often do so with a relatively "light touch"; that is, the forward protraction of the whiskers stops very soon (12–14 milliseconds) after the initial contact preventing strong bending of the whiskers against the surface. This was first observed by Mitchinson *et al.* [17] in the situation where the animal encounters an object on one side of the snout only (Fig. 5). Here, whisker motion on the ipsilateral (i.e., contacting) side of the snout can be seen to cease shortly after the contact, while protraction on the contralateral (opposite first contact) side continues seemingly unaffected by the surface impact. Grant *et al.* [18] went on to show that RCP takes place even when whiskers on both sides of the snout contact the surface and that it occurs in the second whisk cycle following the initial contact as well as in the first (and probably also in subsequent whisk cycles where there is continued contact). Interestingly, one effect of RCP can be to break the synchrony of the whisker movements in the left and right fields, potentially leading to extended periods where the whiskers on the two sides of the face move asynchronously (unpublished observations). RCP is an example of feedback control, since the sensory signals resulting from whisker-environment contact are used to regulate the same contact.

CIA is the related observation that when an animal has made contact with a nearby vertical surface the whiskers on the two sides of the snout are observed to move asymmetrically in subsequent whisks in advance of any further contact. Specifically, whiskers in the ipsilateral (nearest to surface) field tend to protract less, while the contralateral whiskers (furthest away) are seen to protract more strongly (Fig. 5). The result is, again, on the ipsilateral side, to allow the whiskers to make light contacts with the surface avoiding strong bending, while on the contralateral side the increased protraction tends to lead to more whisker contacts with the surface. CIA, in contrast to RCP, is an example of feed-forward control, since the sensory signals resulting from contact during one whisk are used to regulate contacts occurring during subsequent whisks.

Observationally, whisking behavior thus appears to be consistent with a 'minimal impingement, maximal contact' active control strategy [17], where "impingement" describes the extent to which the whiskers could be imagined to penetrate through the contacted surface if it were to offer no resistance. A further observed modulation of whisking, consistent with the maximal contact element of this hypothesis, has been described by Grant *et al.* [18]. This is the ability of the animal to differentially control the velocity of the forward and rear whiskers, and thus the angular separation or "spread" of the whiskers, in order to bring the rear whiskers further forward and (again) increase the number of contacts with the surface of interest.

In the following sections, we describe first our implementation of bilateral whisking pattern generation for the BIOTACT sensor using two coupled oscillators. We then describe how this is modulated by vibrissal signals to generate RCP and CIA that is functionally similar to that observed in the animal (N.B., we make no strong claim concerning the similarity of the underlying mechanisms). Although a number of active sensing strategies have been demonstrated on our earlier whiskered robot platforms [19]–[22], we have not previously investigated or measured the consequences of this active control for sensory discrimination. Our first efforts to do this using the BIOTACT sensor are described in Section IV.

B. Pattern Generation Model

Whisking, as described above, consists most simply of periodic forward and backward motion, and this is what we reproduce first using the BIOTACT Sensor. In the animal, it has been shown that the motion is generated by a Central Pattern Generator (CPG), an intrinsic oscillator that does not require connection to the environment to operate [14]. Since it is known that periodic whisking that is out-of-phase on the two sides of the animal may be expressed, at least two such oscillators must be present. However, whisking tends towards being in phase, so these oscillators must be coupled. In addition, the oscillators may be perturbed by extrinsic events.

We model two individual oscillators, each as a two-dimensional dynamical system with a time-dependent state Ψ_i. The state can be alternatively written as follows (time dependence is not explicitly shown in most of the equations below, for notational simplicity)

$$\Psi_i = \begin{bmatrix} \psi_{i,1} \\ \psi_{i,2} \end{bmatrix} = \begin{bmatrix} r_i \cos(\phi_i) \\ r_i \sin(\phi_i) \end{bmatrix}. \tag{1}$$

The subscript i is chosen from 1, 2 and indexes into the pair of oscillators (also denoted "left" and "right"). The update equation is given by

$$\frac{d\Phi_i}{dt} = D_i + C_i + P_i \tag{2}$$

where the three terms on the right-hand side represent the forcings due to, respectively, the oscillator characteristic, the coupling between the two oscillators, and any perturbation of the oscillator caused by extrinsic events. Integration is forward Euler with an integration period T and, for physical plausibility, all ψ states are hard limited to fall between plus and minus $\psi_{\text{lim}} = 1.5$ after the computation of each timestep.

The update representing the dynamics is given in polar space for simplicity, as follows.

$$D_i^{\text{polar}} = \begin{bmatrix} \frac{(1-r_i)}{\tau_r} \\ 2\pi f_W \end{bmatrix}. \tag{3}$$

This represents a system with a limit cycle on the unit circle and a period of $1/f_W$ seconds. The whisking frequency is set to $f_W = 2$ throughout this report; we also define the whisk period, $T_W = 1/f_W$. Also throughout, we use $\tau_r = 2T_W$. The coupling forcing is given in Cartesian space, as follows:

$$C_i = \begin{bmatrix} \frac{(\psi_{j,1}-\psi_{i,1})}{\tau_{11}} + \frac{(\psi_{j,2}-\psi_{i,1})}{\tau_{12}} \\ \frac{(\psi_{j,1}-\psi_{i,2})}{\tau_{21}} + \frac{(\psi_{j,2}-\psi_{i,2})}{\tau_{22}} \end{bmatrix} \tag{4}$$

where $j = 3 - i$ is the index of the opposite oscillator and τ_{pq} is the time constant with which state $\psi_{i,p}$ is forced towards state $\psi_{j,q}$ of the opposite oscillator. Note that a negative τ_{pq} will tend to force states apart. In this report, only τ_{11} is not infinite (that is, only the first coupling term is nonzero), and it always takes a positive value of $2T_W$, which brings the oscillators back into phase with a time constant of one whisking period.

In this report, the only perturbation is a noise signal, given by the following:

$$P_i = \begin{bmatrix} \sigma_\eta H_{50}(\eta_{i,1}) \\ \sigma_\eta H_{50}(\eta_{i,2}) \end{bmatrix}. \tag{5}$$

$\eta_{i,d} \sim N(0,1)$ is White Gaussian Noise, H_{50} is a third-order low-pass Butterworth filter with 50 Hz cutoff frequency, and $\sigma_\eta = 0.005$ is a small noise amplitude. The first state of each oscillator, $\psi_{i,1}$, is associated with whisker protraction on that side by deriving a "muscle drive" signal from it as follows:

$$W_i = \frac{\tanh(2\psi_{i,1}) + 1}{2}. \tag{6}$$

The other state $\psi_{i,2}$ can be viewed as the recovery state of the oscillator, and its value does not affect any output. W_i is used to drive a mechanical model of the motion of the whiskers, as follows:

$$m\frac{d^2\theta_{i,w}}{dt^2} = q_{i,w}W_i - k\theta_{i,w} - \nu\frac{d\theta_{i,w}}{dt} \tag{7}$$

$m = f_W^2/100$ is a mass, $k = 1$ is a spring constant, and $\nu = 4\sqrt{km}$ a damping constant. Hence, $\theta_{i,w}$, the protraction angle of the wth whisker on the ith side, displays second-order dynamics in response to the muscle drive W_i. $q_{i,w}$ is the muscle gain (coefficient between drive and force) associated with the wth whisker on the ith side, and is given by the following equation:

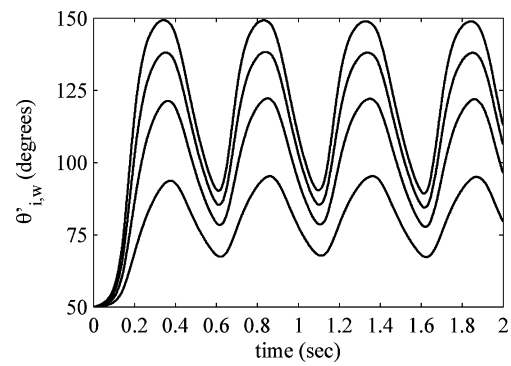


Fig. 6. $\theta'_{1,1}$ during free (unperturbed) whisking with four different values of $g_{1,1}$ (higher values generate higher values of $\theta'_{i,w}$).

$$q_{i,w} = 1.45(1 - z_i)g_{i,w}e^{-\theta_{i,w}^2} \tag{8}$$

$z_i \in [0,1]$ is a gating variable associated with the ith side. It takes a nominal value of zero (no suppression), and can range up to one (total suppression), modeling suppression of muscle drive to all whiskers on that side. $g_{i,w} \in [0,2]$ is a gating variable associated with the wth whisker on the ith side, which controls how excitable the individual whisker is – i.e., how much the whisker responds to the drive from the oscillator. It takes a nominal value of one, and can range both positively and negatively, modeling excitation or suppression for each whisker individually. The constant term scales the overall gain so that with $z_i = 0$ and $g_{i,w} = 2$, θ approaches, but does not exceed, unity (see below). The final (exponential) term is a simple model of the reduction in force produced by a muscle as its length decreases (reaching 37% as θ reaches its maximum value of unity). The overall effect is that the values of z_i and $g_{i,w}$ control the response of each side and each whisker, affecting both amplitude of oscillation and mean protraction angle, although it should be noted that there is no direct independent control over these latter variables.

An example of the effect of varying the gain is given in Fig. 6. The output, the angle of the first whisker on the left side, is given for $z_1 = 0$ and $g_{1,1} \in \{0.5, 1.0, 1.5, 2.0\}$. $\theta' = 100\theta + 50$ is plotted, which is the conversion of θ (which ranges in zero to one) into an angle in degrees to pass to the motor hardware.

C. Rapid Cessation of Protraction

A substantial component of whisker motor control that has been quantified is the *Rapid Cessation of Protraction* (RCP) that often follows contact between the whiskers and the environment, and which is hypothesized to limit the degree to which the whiskers bend during contact [17], [18]. We wish to reproduce this aspect because controlling the degree of bending could potentially improve the quality of data collected.

A contact signal is derived on each side from the x-deflection of the whiskers on that side (contact usually elicits strong x-deflection since the sensory x axis is parallel to the actuated axis).

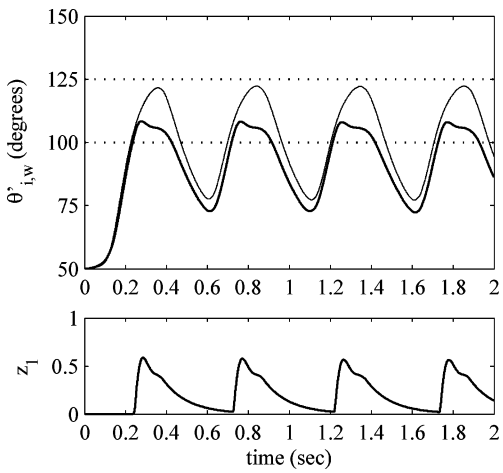


Fig. 7. Whisking with RCP modulation. When the left whiskers (thick line) are driven past a simulated obstacle (lower dotted line), a contact signal results which temporarily excites the gating variable z_1 and suppresses protraction ipsilaterally. The right whiskers (thin line) do not reach the obstruction on that side (upper dotted line), and are not modulated (thus, they show the expected movement of the left whiskers if the modulation were absent).

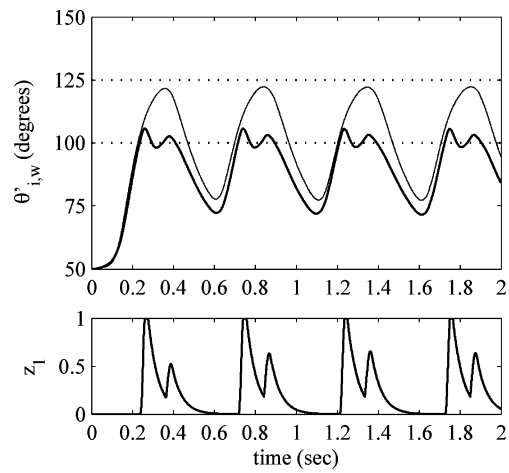


Fig. 8. Whisking with strong and fast RCP modulation. Left whiskers (thick line) quickly detach from the simulated obstacle (lower dotted line), and protraction reignites within a single whisk, leading to a biphasic whisk profile and two distinct contacts with the obstacle. The movement of the right whiskers (thin line) shows the expected movement of the left whiskers if the modulation were absent.

A third-order low-pass Butterworth filter with 50 Hz cutoff frequency H_{50} is first used to smooth the noisy whisker sensory signals. The maximum absolute value of this signal, across all whiskers on a side, forms the contact signal, π_i

$$\pi_i(n) = \max_{w\in[1,12]} \left|H_{50}(x_{i,w}(n))\right|. \tag{9}$$

This signal is used to suppress protraction ipsilaterally, through the gating variable z_i, without affecting the oscillator dynamics. The gating variable dynamics are given by

$$z_i(n) = \left\langle \max\left(\left(1-\frac{T}{\tau_z}\right)z_i(n-1), \sigma_z\pi_i(n)\right)\right\rangle_0^1 \tag{10}$$

where σ_z is a gain associated with the motor modulation, and $\langle\,.\,\rangle_a^b$ is a limit operation (transforming values not in the interval $[a,b]$ onto the nearest value that is in the interval). Contact on any whisker, thus, suppresses protraction ipsilaterally, an effect which decays with time constant τ_z. We can simply simulate contact signals from the whisker sensors using the following equation:

$$x_{i,w} = \left\langle \sigma_x \max\left(0, \theta'_{i,w} - \theta'_{i,\lim}\right)\right\rangle_{-1}^{+1}. \tag{11}$$

We simulate an obstruction at a different location on each side by setting $\theta'_{1,\lim} = 100$ and $\theta'_{2,\lim} = 125$, using $\sigma_x = 0.1$. Setting $\sigma_z = 0.7$ and $\tau_z = T_W/4$ generates the results graphed in Fig. 7. Protraction on the contacting (left) side is suppressed during contact, such that the depth of contact is controlled. The oscillator dynamics are unaffected, so the phase behavior is unchanged. Furthermore, the gating variable z_1 has all but recovered by the time of the next contact, so there is no inter-whisk modulation; that is, this aspect of control is entirely *reactive*.

We can simulate stronger and faster RCP control by setting $\sigma_z = 2$ and $\tau_z = T_W/10$; the results are shown in Fig. 8. In this case, protraction is more strongly suppressed such that the $\theta_{1,w}$ quickly fall below $\theta_{1,\lim}$ and the contact signal π_i becomes zero. Thus, the gating variable z_1 quickly recovers, and protraction reignites within the same whisk period, resulting in a biphasic whisk profile. We hypothesize that this control loop may be the origin of the biphasic whisks that have been observed in whisking animals [23].

Note that in the experiments presented in Section IV, we have set τ_z to $T_W/4$ throughout and the maximum value of σ_z was 0.9 so that this biphasic whisking was not encountered.

D. Contact-Induced Asymmetry

Another aspect of whisker motor control that has been measured is *Contact-Induced Asymmetry* (CIA), in which contact with a unilateral obstruction has been shown to reduce/increase whisker protraction on the ipsilateral/contralateral side [17], [18]. Here, we model CIA by introducing global excitation of protraction in all whiskers following contact, through the gating variables $g_{i,w}$. CIA results from the interaction of this global excitation with the reduction of protraction on the ipsilateral side due to the RCP mechanism described above.

The dynamics of $g_{i,w}$ are governed by the equations below. The results of using this modulation with $\sigma_g = 0.5$ and $\tau_g = T_W/4$ ($\sigma_z = 0.7$, $\tau_z = T_W/4$, as for Fig. 7) are shown in Fig. 9. The whiskers on the right side are now brought forward and "find" the obstruction

$$g_{i,w}(n) = \left\langle \max\left(g^*_{i,w}(n), \sigma_g\zeta(n)\right)\right\rangle_0^2 \tag{12}$$

$$g^*_{i,w}(n) = g_{i,w}(n-1) + \frac{T}{\tau_g(1-g_{i,w}(n-1))} \tag{13}$$

$$\zeta = \max(\pi_1, \pi_2). \tag{14}$$

E. Summary

In summary, the motor control strategies described in this section may help the robot, in emulation of the rat, to take best advantage of its sensors (whiskers) in the face of the changing geometry of its near environment (Figs. 5 and 9). Complete quantification of the impact of these strategies in different sensory tasks will require extensive testing. In the following section, we

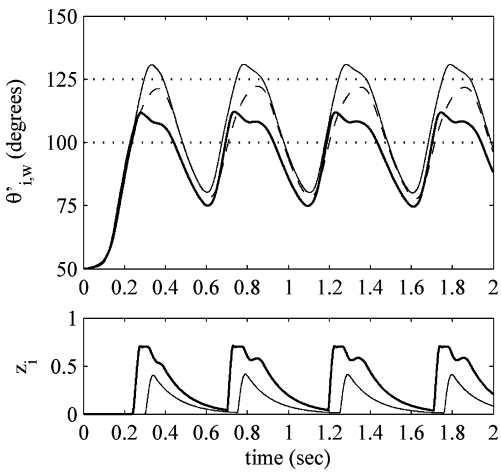


Fig. 9. As Fig. 7, but with CIA modulation enabled as well as RCP modulation. The left whiskers (thick solid line) contact an obstruction (lower dotted line) first, and cause excitation of the gating variables $g_{i,w}$, leading to global excitation of protraction. Thus, the right whiskers (thin solid line) are pushed forward and "find" the obstruction on the right side (upper dotted line) – compare with Fig. 7, where the obstruction on the right side is not found. Since both sides are modulated, here, we show also (dashed gray line) the expected movement of both sides were the modulations absent.

present our first assessment of the impact of implementing RCP, measuring the effect on performance in a texture discrimination task.

IV. Sensory Discrimination

A. Sensory Discrimination in Rats

Experiments with rats (reviewed in [3] and [24]), and with other tactile specialists such as the Etruscan shrew [25], have shown that whiskered animals are able to use their facial vibrissae to localize objects in space and to discriminate object properties such as shape and texture. For instance, in localization tasks, rats have been shown to use vibrissal information in gap measurement and gap jumping, and to be able to measure angular position of a contact along the sweep of the whisker, and the axial distance to a point of contact on a single whisker. Ahissar *et al.* [26] have proposed that the timing and extent of bending of the whiskers together provide sufficient information to localize a point of contact in three-dimensional space (elevation, azimuth, and axial distance, respectively). Several studies have shown that rats can use their macrovibrissae to reliably discriminate between surfaces with different texture, showing a similar level of acuity to human fingertips [10], [27], [28]. Movement of an individual whisker across a texture has been shown to give rise to a characteristic whisker vibration or "kinetic signature" which is thought to form the basis on which sensory discriminations are made [24], [29], [30], although the precise process that underlies such discriminations is not well understood.

A number of studies with artificial whiskers have looked at classification or feature detection tasks that require tactile discrimination abilities similar to those seen in whiskered animals. Previous work is this area has been reviewed by Fox *et al.* [5] and Prescott *et al.* [4]. In the current study, we sought to examine the capacity of the BIOTACT sensor to simultaneously extract two different stimulation parameters: the horizontal (axial) distance to the contacted surface and the texture of that surface, with and without whisking modulation using RCP. To evaluate the effectiveness of the BIOTACT sensor in this discrimination task, we tested two different classification methods that we have developed for use with artificial vibrissal signals, one template-based [5], [31], [32]), the other using "naïve" Bayes [33], [34]). Here, we describe the experimental setup used to generate a $3 \times 3 \times 3$ (axial distance, texture, RCP) training, and test (validation) datasets. We then briefly describe the two classification algorithms and the results of applying these to the BIOTACT sensor data.

TABLE I
Textured (Abrasive) Surfaces

ISO Grade	average particle diameter (μm)
P240	58.5
P600	25.8

B. Texture Classification Experiments

A set of textured surfaces were fixed in a vertical plane and rigidly attached to the robot arm's support post. The arm trajectory was preprogrammed to bring the front of the sensor to within a specified horizontal axial distance from the surface (5, 10, or 15 mm) with the two rows of whiskers aligned in a horizontal plane. Fig. 4 shows a horizontal section through the sensor in its test configuration. All experiments were carried out with a whisk frequency (f_W) of 2 Hz. This is somewhat slower than the whisking frequency of rats (dominant frequency 8 Hz [10]), but this is not unreasonable given the greater length of our whiskers (see Section II-B). The choice of the 2 Hz frequency is fairly arbitrary, however, and limited testing at other whisking frequencies has not revealed that this parameter is critical in achieving successful texture classification.

Three different textures have been used as discriminanda, a smooth plastic surface and two grades of commercial abrasive (3M silicon carbide "Wetordry Tri-M-ite") see Table I. The latter surfaces were chosen to give a direct comparison with the whisking mobile robot "Whiskerbot" texture classification reported by Fox *et al.* [5].

The combination of three distances and three textures gave a total of nine distinct classes to be discriminated between.

We also present results comparing three different whisking strategies.

— **Unmodulated**.
The whisker pattern generator output, unperturbed by feedback from sensors.

— **RCP7**.
Motor drive signal modulated with RCP, using a moderate gain factor [$\sigma_z = 0.7$, see (10)].

— **RCP9**.
Motor drive signal modulated with RCP, using a stronger gain factor [$\sigma_z = 0.9$, see (10)].

The purpose of this experiment was to determine whether RCP modulation was effective in improving the accuracy and robustness of classification and not to determine optimal values of

these parameters. As such, we chose values which we knew would be large enough to have a clearly visible effect on the output waveforms and small enough to prevent the biphasic whisking effect seen in simulation in Fig. 8.

C. Stationary Naïve Bayesian Classification

Bayesian classification involves recording the likelihoods of measurements from example sensory data. Given new test data, these likelihoods are used with Bayes' rule to calculate the posterior probability of the test data being drawn from each trained class. The class with the largest posterior is most probable.

Here, a stationary naïve Bayes classifier was used that had previously been demonstrated as an effective classifier of texture data [33], [34]. A naïve version of Bayes' rule, in which the measurements are assumed to be statistically independent is used, along with an assumption of identical likelihoods over time, to calculate the posterior probability for each class of training data.

Briefly, the training data is used to estimate the log-probability distributions $\log P_w(x_i|\mathrm{C}_l)$ (log-likelihoods) of each sensor measurement x_i occurring. Measurements were quantized into one hundred equal-width intervals that spanned the entire data range of each whisker, then the resulting histogram of sensor measurements smoothed with a Gaussian of width 5 intervals to correct for sampling errors and normalizing by the number of samples to give the likelihood. This led to $n_w \times n_C$ log-likelihood functions, for each of the $n_w = 12$ whisker inputs (two sensor directions for each of the six whiskers) and $n_C = 9$ texture/distance classes.

Then, for each validation trial, the classification was achieved using naïve Bayes rule to calculate the nine posterior probabilities for each of the texture/distance classes, representing that the measurements were drawn from the 12 likelihoods for each of these possible classes. These posterior probabilities were calculated from the stationary naïve Bayes' rule

$$\log P(\mathrm{C}_l|x_1,\ldots,x_n) = \sum_{w=1}^{12}\sum_{i=1}^{n}\log P_w(x_i|\mathrm{C}_l). \tag{15}$$

for flat priors, a stationary likelihood and the "naïve" assumption of statistical independence. The classification is given by the largest log-posterior for each validation trial,

$$\mathrm{T} = \arg\max_{\mathrm{T}_l} \log P(\mathrm{C}_l|x_1,\ldots,x_n). \tag{16}$$

The utility of the stationary naïve Bayes rule is that it greatly simplifies the process of calculating the posteriors to an algorithm that is linear sum over log-likelihoods. Then, the classification is similar in algorithmic complexity to a linear perceptron. Hence, the classification is extremely fast, with real-time performance easily achievable on a standard PC.

D. Sum-of-Squares Template-Based Classification

Template-based classification involves recording example sensory data as templates during a training phase, and comparing the stored templates to new data during the test phase. A template-based classifier has been shown to be successful in discriminating whisker contacts of varying radial distance and speed [31]. Template-based classification involves recording example sensory data as templates during a training phase, and comparing the stored templates to novel data during the test phase. By systematically comparing the novel data to signals encountered previously, a classification can be made by declaring which of the stored templates the novel signal is most similar to. This approach is straightforward to implement, and requires very little computation or preprocessing of the signal.

It has been shown previously that texture discrimination is dependent on whisker movement [35], and surface position [5]. In the present study, each template corresponds to a texture-distance pair. Classification based on these templates is, therefore, simultaneous classification of both the texture and distance to contact. From the training dataset, an array of templates were generated by storing the average of the signals for each whisker in each class in the training set. This gave rise to 9 sets of 12 templates. During the test phase, trials were taken at random from the test set as inputs to the classifier. An element-wise sum of squared errors calculation was made between the input I and each template T_i

$$e(T_i) = \sum_{t=1}^{n}(I(t) - T_i(t))^2 \tag{17}$$

where n is the length of the template, in samples. A sum of errors from the 12 templates in each set were taken, and compared across the 9 classes. The class with the lowest total sum of squared errors was determined the winner, and a recording was made in an output array of the estimated texture and distance to contact of the input trial.

E. Classification Performance

1) Overall Classification Performance (Unmodulated Whisking Dataset): The hit rates over multiple validation trials were first calculated for the unmodulated whisking dataset. From one to eight training whisks were used. Because a total of 12 (training and validation) whisks were measured, from 11×9 to 4×9 single whisks remained for validation over all nine texture/distance classes. The hit rate for each whisk number was then plotted against the number of training whisks [Figs. 10(a) and 11(a), solid black lines]. Both classifiers show classification accuracy increasing with numbers of training whisks and reaching a maximum after 4 or 5 training whisks. Thereafter, increasing the number of whisks leads to similar performance.

2) Comparative Performance of Unmodulated Whisking, RCP7 and RCP9 Datasets: The unmodulated whisking dataset discussed in the preceding study was compared with results from the RCP datasets. Hit rates for each whisk number are plotted in Figs. 10(a) and 11(a), dashed and dotted lines.

For both classifiers, the RCP modulated whiskers give higher hit rates after one or two training whisks than the unmodulated, with RCP7 giving the highest accuracy. However, the two classifiers behave differently as the number of training whisks increases; using the stationary Bayesian method RCP7 continues increasing, reaching 100% accuracy after seven training whisks,

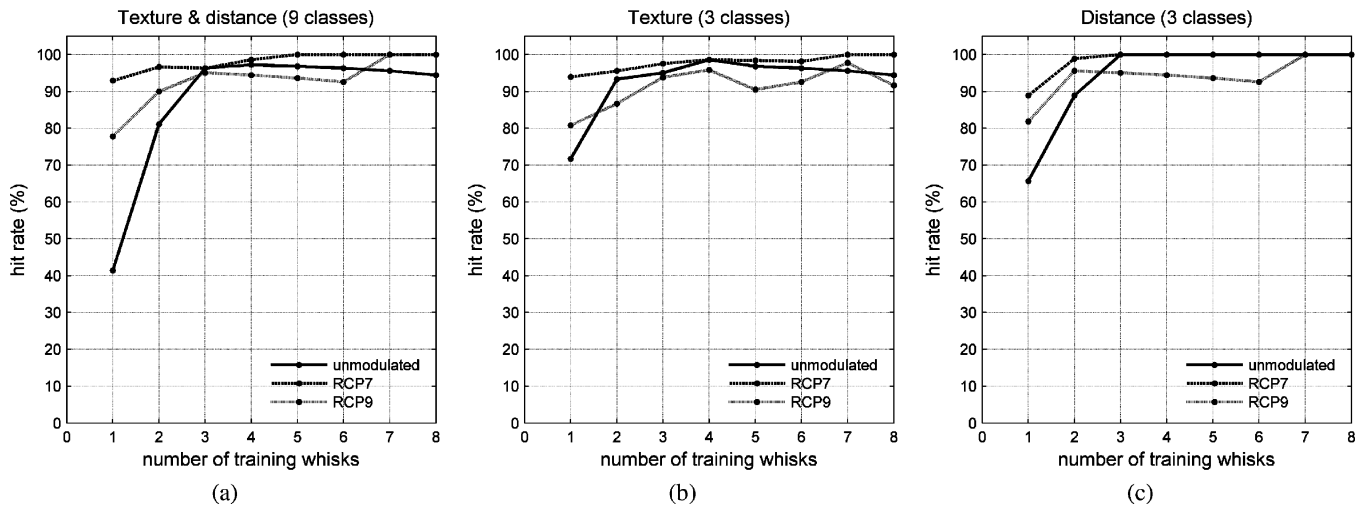


Fig. 10. Classification performance using naïve Bayes. Panel A compares the performance of the unmodulated, RCP7, and RCP9 datasets for classifying both texture and distance together. Panels B and C are analogous to panel A, but for a single classification of either texture or distance. Classification accuracy is measured by hit rate, which is the ratio of successful classifications to validation whisks. (a) Combined texture and distance. (b) Texture classification. (c) Distance classification.

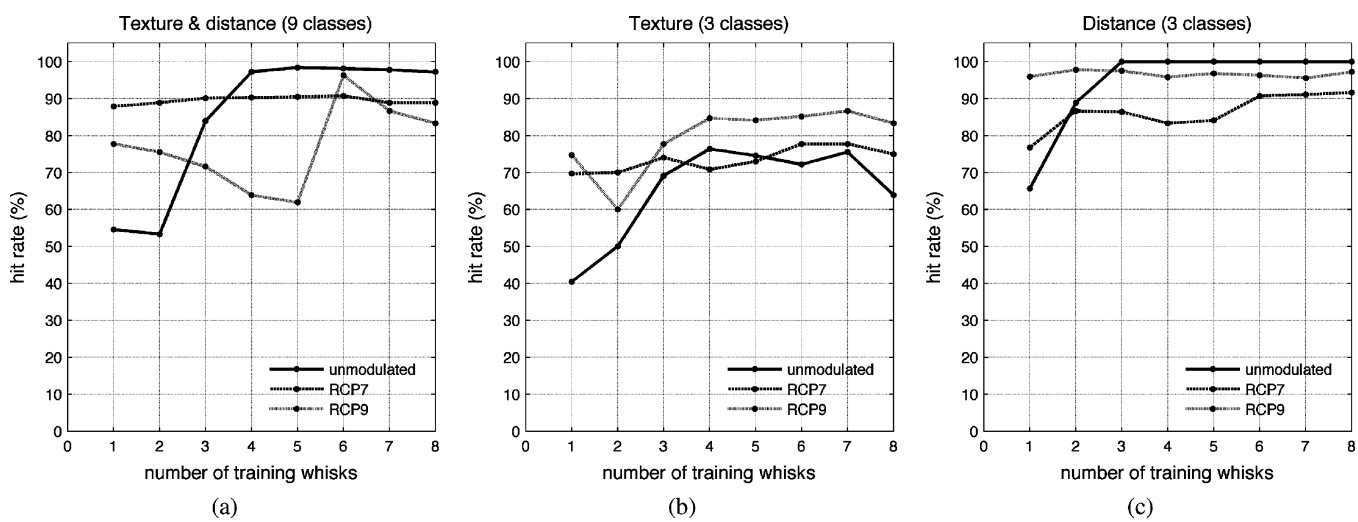


Fig. 11. Classification performance using template based classifier. Classification accuracy is measured by hit rate, which is the ratio of successful classifications to validation whisks. (a) Combined texture and distance. (b) Texture classification. (c) Distance classification.

whereas with the template classifier, RCP7 shows little improvement and RCP9 begins to lose accuracy with increased training but suddenly improves after six training whisks.

3) Classification Performance for Just Textures or Distances (Unmodulated Whisking Dataset): Classifiers for texture or distance (three classes of each) were then constructed by concatenating all training data for each distance or each texture. (Equivalently, the likelihoods could be averaged.) Once again, the number of training whisks was varied from one to eight and the hit rates plotted against whisk number [Figs. 10(b) and (c) and 11(b) and (c) solid lines].

Using the stationary Bayesian method, classification for both features again reached high accuracy after about three/four training whisks. Roughly speaking, the hit rate for the single classifier texture/distance classifier was about equal to multiplying the hit rates for the individual classifiers, so there was no advantage apparent from considering either the single nine-class method or the double three-class methods to distinguish both texture and distance. It should be noted that the double method is more efficient computationally, because a total of six classes are considered rather that nine.

The template-based method performed similarly on the distance classification task but only reached 75% hit rate on the texture classification.

4) Comparative Performance for Just Textures or Distances of Unmodulated Whisking, RCP7 and RCP9 Datasets: Finally, the single-classifiers of texture or distance were applied to the RCP7 and RCP9 datasets and the hit rate plotted against training whisk number [Figs. 10(b) and (c) and 11(b) and (c) dashed and dotted lines].

With the stationary Bayesian method, the same pattern of RCP7 being the best classifier, followed by RCP9 and then unmodulated whisking was again found, with the classification performance reaching its maximum at four whisks.

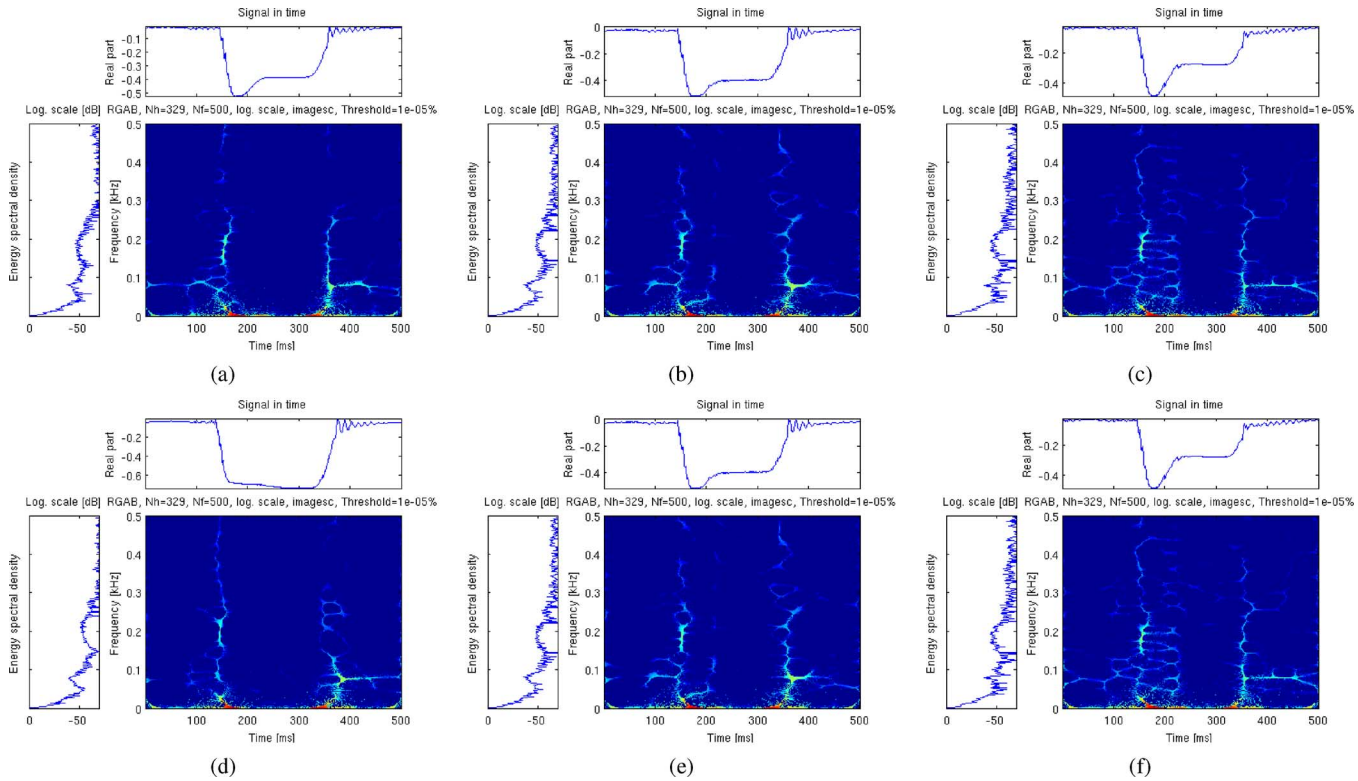


Fig. 12. Spectral analysis comparing response of smooth and textured (p240) surfaces with and without RCP modulation. Spectrograms computed using reassigned Gabor time-frequency representation [Time Frequency Toolbox (http://tftb.nongnu.org) *tfrrgab* function]. (a) Unmodulated, smooth. (b) RCP 0.7, smooth. (c) RCP 0.9, smooth. (d) Unmodulated, p240 abrasive. (e) RCP 0.7, p240 abrasive. (f) RCP 0.9, p240 abrasive.

With the template-based method, RCP9 gave higher classification accuracy in texture classification. This was also the case in distance classification with less than three training whisks, but RCP modulated whisking resulted in lower accuracy than unmodulated with more training contacts.

V. Discussion

The design of tactile sensor systems for robots has taken inspiration from mammalian whiskers (see [4]), and from insect antennae (see, e.g., [36] and [37]), since at least the 1980s. However, most work in this field has not specifically addressed the contribution of sensor motion to the problem of tactile discrimination. Earlier work by the current authors identified characteristic features of active sensing control in the whisker movement of rats [17], [18] and showed that these could be incorporated into artificial vibrissal systems for robots [20], [21]. In the current study, we extended this work to directly examine the impact of active control on a two-dimensional (texture $\times$ distance) sensory discrimination task. Based on these early results, we appear to see an advantage for active control (rapid cessation of protraction – RCP) for classification after fewer training whisk cycles, particularly when using the stationary naïve Bayes method. This outcome is perhaps surprising in that RCP tends to reduce rather than increase the duration of contact between the whisker and the surface. It therefore suggests that the benefit of RCP may be to increase the repeatability of whisker-surface contacts. The advantage is not clearly demonstrated when using the template-based method and more than two training whisks; under which circumstances classification of distance was actually less accurate when RCP modulation was applied. Further experiments both with artificial vibrissae and perhaps, using *in vivo* electrophysiology in animals, could usefully investigate the specific consequence of this aspect of control that might lead to improved discriminative capabilities. One of the advantages of the BIOTACT Sensor, over our earlier work on artificial vibrissal systems, is that the whiskers are individually actuated; therefore, sensor movement can potentially be modulated by sensory feedback on a per-whisker basis. Although we have yet to explore this possibility, our experimental work with rats suggests that whisker velocity can be differentially controlled for more anterior and posterior whiskers in a manner that affects the angular spread of the whiskers [18]. We also found a difference of up to 5 milliseconds in the timing of RCP, with posterior whiskers beginning to retract later than more anterior ones [18]. The artificial whisker results presented above suggest that such differences, while subtle, could be impacting on the discrimination capabilities of the whiskers. Since the experiments described here used only symmetric contacts with a vertical surface, we have also yet to investigate or quantify the impact of other aspects of active whisking control (particularly contact-induced asymmetry) on sensory processing.

It has been shown previously that whisker-based texture discrimination is critically dependent on whisker-object contact geometry [5]. In the current study, inspection of the raw signals for each class reveals greater within class variation for contacts with a given texture than at a given distance. This is because contacts at different distances result in stereotypical macroscopic changes in whisker deflection amplitude [32], whereas texture-dependent features involve smaller and more variable deflections of the whisker (either by their direct contact with random

particles on the rough surface or by an oscillatory "ringing" of the whisker as it draws away from the surface). This variation could perhaps explain why the template-based classifier considered here did not generalize well across conditions when attempting to classify texture, compared to the stationary Naïve Bayes classifier [compare Figs. 10(b) and 11(b)).

Fig. 12 shows single-whisk time histories of the signals from one of the smaller whiskers contacting the smooth and p240 textures at the same distance, together with spectrograms computed using the reassigned Gabor time-frequency representation [38]. Although the effect of RCP can be clearly seen in the time history, the differences between textures are more subtle.

Further support that the time-independence of a classifier can lend its robustness in certain circumstances is given by an equivalence between the template classifier and a form of Naïve Bayes classifier. Because the considered template classifier uses a sum-of-squares error function, it is equivalent to a log-product of independent noise terms that satisfy the Naïve Bayes definition

$$\sum_{t=1}^{N}(I(t)-T_i(t))^2 = -\log\prod_{t=1}^{N}\frac{1}{Z_i}\exp-\frac{(I(t)-T_i(t))^2}{2\sigma^2} - c \tag{18}$$

for constants $\sigma = 1/2$, $c = \sum_i \log Z_i$, and Z_i area-terms canceling with c. We can then consider the differences between these two forms of Naïve Bayes. The above Naïve Bayes formulation of the template classifier assumes varying noise distributions over time t and can thus be interpreted as a time-dependent Naïve Bayes classifier. In contrast, the Naïve Bayes considered earlier in this paper was stationary, supporting that time-dependence/independence is the principle difference between the two classifiers considered here. That being said, in some circumstances the simplifying assumption of time-independence may be a hindrance, for example, when the phases of the oscillations are crucial to the classification. Future work could investigate new forms of Naïve Bayes rules combining useful properties of both methods. The success of Naïve Bayes classification is encouraging and suggests that this relatively simple technique could potentially be a useful tool for other discrimination tasks involving time series data.

One of the goals in designing this first generation BIOTACT Sensor was to produce a modular unit that could be flexibly assembled into different configurations. The first configuration investigated was chosen to be a simple, radially symmetric cone with a small number of whiskers. This configuration clearly does not match the vibrissal system of the rat in a number of ways, one important consequence of which is that the sensory "surface" formed by the whiskers tips at full protraction does not form a broad field in front of the sensing device in the manner observed in animals (compare Figs. 1 and 2, for example). A second generation system, currently under construction, is being developed to specifically address this limitation. It will also incorporate an array of shorter, more closely spaced, unactuated *microvibrissae*, again based on the morphology of the rat which has a similar array on the frontal part of the snout and around its mouth and which have been shown to play a key role in object discrimination [9]. In addition to a more biomimetic sensor morphology, we also plan to improve aspects of the sensory processing. For instance, we have shown that, in the task of detecting whisker-surface contacts, sensor noise, due to the movement of the whiskers alone, can generate spurious false positives [39]. To address this problem, we previously developed an adaptive filtering algorithm that uses a copy of the motor control signal to predict movement-related sensory noise, such that cleaned-up signals (from which the predicted noise is subtracted) avoid the problem of spurious contacts. The same filter will likely prove useful in discrimination tasks such as the distance/texture problem considered above. Interestingly, the algorithm employed for adaptive filtering was modeled on the microcircuitry of the mammalian cerebellum, which, in rats, is known to be heavily involved in processing vibrissal signals.

A long-term ambition of our research is to identify the computations that are carried out in the rat brain when the animal is engaged in tasks that involve tactile discrimination. The classification methods, described above, are not specifically derived from considerations of how vibrissal processing mechanisms in the rat brain may operate. However, if we are able to devise effective classifiers whose capabilities match those of the rat, these can, in the future, be used to inform hypotheses about the biological system that may be testable using neurophysiological data.

Most tactile sensors for robots are binary devices that do little more than detect whether or not a contact has happened. Biology shows us that this is a modality with considerably greater potential. In order to make the most of tactile sensing, we believe that the biomimetic route, focusing particularly on the role of active control, can inspire technologies that will provide future robots with a sense of touch to parallel, or even surpass, that of animals. The BIOTACT Sensor described in this paper is we hope a useful step towards this longer term goal.

REFERENCES

[1] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, pp. 996–1005, 1988.

[2] S. J. Lederman and R. L. Klatzky, "Extracting object properties through haptic exploration," *Acta Psychol. (Amst)*, vol. 84, no. 1, pp. 29–40, 1993.

[3] T. Prescott, B. Mitchinson, and R. Grant, "Vibrissal behavior and function," *Scholarpedia*, 2011. [Online]. Available: http://www.scholarpedia.org/article/Vibrissal_behavior_and_function, to be published

[4] T. Prescott, M. Pearson, B. Mitchinson, J. Sullivan, and A. Pipe, "Whisking with robots from rat vibrissae to biomimetic technology for active touch," *IEEE Robot. Autom. Mag.*, vol. 16, no. 3, pp. 42–50, 2009.

[5] C. Fox, B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott, "Contact type dependency of texture classification in a whiskered mobile robot," *Autonomous Robots*, vol. 26, pp. 223–239, 2009.

[6] A. S. Ahl, "The role of vibrissae in behavior – A status review," *Vet. Res. Commun.*, vol. 10, no. 4, pp. 245–268, 1986.

[7] B. Mitchinson, K. N. Gurney, P. Redgrave, C. Melhuish, A. G. Pipe, M. Pearson, I. Gilhespy, and T. J. Prescott, "Empirically inspired simulated electro-mechanical model of the rat mystacial follicle-sinus complex," *Proc. R Soc. London B Biol. Sci.*, vol. 271, no. 1556, pp. 2509–2516, Dec. 2004.

[8] D. N. Hill, R. Bermejo, H. P. Zeigler, and D. Kleinfeld, "Biomechanics of the vibrissa motor plant in rat: Rhythmic whisking consists of triphasic neuromuscular activity," *J. Neurosci.*, vol. 28, no. 13, pp. 3438–3455, 2008.

[9] M. Brecht, B. Preilowski, and M. M. Merzenich, "Functional architecture of the mystacial vibrissae," *Behav. Brain Res.*, vol. 84, no. 1-2, pp. 81–97, 1997.

[10] G. Carvell and D. Simons, "Biometric analyses of vibrissal tactile discrimination in the rat," *J. Neurosci.*, vol. 10, no. 8, pp. 2638–2648, 1990.

[11] B. Mitchinson, T.-S. Chan, J. Chambers, M. Pearson, M. Humphries, C. Fox, K. Gurney, and T. J. Prescott, "BRAHMS: Novel middleware for integrated systems computation," *Adv. Eng. Inform.*, vol. 24, no. 1, pp. 49–61, 2010.

[12] C. M. Williams and E. M. Kramer, "The advantages of a tapered whisker," *PLoS ONE*, vol. 5, no. 1, p. e8806, Jan. 2010.

[13] M. J. Hartmann, N. J. Johnson, R. B. Towal, and C. Assad, "Mechanical characteristics of rat vibrissae: Resonant frequencies and damping in isolated whiskers and in the awake behaving animal," *J. Neurosci.*, vol. 23, no. 16, pp. 6510–6519, Jul. 2003.

[14] P. Gao, R. Bermejo, and H. P. Zeigler, "Whisker deafferentation and rodent whisking patterns: Behavioral evidence for a central pattern generator," *J. Neurosci.*, vol. 21, no. 14, pp. 5374–5380, 2001.

[15] R. W. Berg and D. Kleinfeld, "Rhythmic whisking by rat: Retraction as well as protraction of the vibrissae is under active muscular control," *J Neurophysiol.*, vol. 89, no. 1, pp. 104–117, Jan. 2003.

[16] R. B. Towal and M. J. Hartmann, "Right-left asymmetries in the whisking behavior of rats anticipate head movements," *J Neurosci.*, vol. 26, no. 34, pp. 8838–46, 2006.

[17] B. Mitchinson, C. J. Martin, R. A. Grant, and T. J. Prescott, "Feedback control in active sensing: Rat exploratory whisking is modulated by environmental contact," *Royal Soc. Proc. B*, vol. 274, no. 1613, pp. 1035–1041, 2007.

[18] R. A. Grant, B. Mitchinson, C. W. Fox, and T. J. Prescott, "Active touch sensing in the rat: Anticipatory and regulatory control of whisker movements during surface exploration," *J. Neurophysiology*, vol. 101, pp. 862–874, 2009.

[19] B. Mitchinson, M. Pearson, C. Melhuish, and T. J. Prescott, "A model of sensorimotor co-ordination in the rat whisker system," in *Proc. 9th Int. Conf. Simulation of Adaptive Behaviour, From Animals to Animats 9*, 2006, pp. 77–88.

[20] M. J. Pearson, A. G. Pipe, C. Melhuish, B. Mitchinson, and T. J. Prescott, "Whiskerbot: A robotic active touch system modeled on the rat whisker sensory system," *Adaptive Beh.*, vol. 15, no. 3, pp. 223–240, 2007.

[21] M. J. Pearson, B. Mitchinson, J. Welsby, T. Pipe, and T. Prescott, "Scratchbot: Active tactile sensing in a whiskered mobile robot," in *From Animals to Animats 11*, S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, and J.-B. Mouret, Eds. Berlin, Germany: Springer, 2010, vol. 6226, Lecture Notes in Computer Science, pp. 93–103.

[22] B. Mitchinson, M. J. Pearson, A. G. Pipe, and T. J. Prescott, "Biomimetic robots as scientific models: A view from the whisker tip," in *Neuromorphic and Brain-Based Robots*, J. Krichmar and H. Wagatsuma, Eds. Cambridge, MA: MIT Press, 2011, to be published.

[23] R. B. Towal and M. J. Z. Hartmann, "Variability in velocity profiles during free-air whisking behavior of unrestrained rats," *J. Neurophysiol.*, vol. 100, no. 2, pp. 740–752, 2008.

[24] M. Diamond, M. von Heimendahl, and E. Arabzadeh, "Whisker-mediated texture discrimination," *PLoS Biol.*, vol. 6, no. 8, 2008.

[25] F. Anjum, H. Turni, P. G. Mulder, J. van der Burg, and M. Brecht, "Tactile guidance of prey capture in Etruscan shrews," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 103, no. 44, pp. 16 544–16 549, 2006.

[26] E. Ahissar and P. M. Knutsen, "Object localization with whiskers," *Biol. Cybern.*, vol. 98, no. 6, pp. 449–458, 2008.

[27] E. Guic-robles, G. Guajardo, and C. Valdivieso, "Rats can learn a roughness discrimination using only their vibrissal system," *Behavioral Brain Res.*, vol. 31, pp. 285–289, 1989.

[28] E. Guic-robles, W. M. Jenkins, and H. Bravo, "Vibrissal roughness discrimination is barrelcortex-dependent," *Behavioral Brain Res.*, vol. 48, no. 2, pp. 145–152, 1992.

[29] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, and P. Konig, "Texture signals in whisker vibrations," *J. Neurophysiol.*, vol. 95, no. 3, pp. 1792–9, 2006.

[30] J. Wolfe, D. N. Hill, S. Pahlavan, P. J. Drew, D. Kleinfeld, and D. E. Feldman, "Texture coding in the rat whisker system: Slip-stick versus differential resonance," *PLoS Biol.*, vol. 6, no. 8, p. e215, 2008.

[31] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott, "Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems," in *Proc. 11th Int. Conf. Simulation of Adaptive Behavior*, 2010, pp. 178–187.

[32] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott, "Whisker-object contact speed affects radial distance estimation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2010, pp. 720–725.

[33] N. F. Lepora, M. Evans, C. W. Fox, M. E. Diamond, K. N. Gurney, and T. J. Prescott, "Naive Bayes texture classification applied to whisker data from a moving robot," in *Proc. IEEE IJCNN*, Apr. 27, 2010, pp. 1–8.

[34] N. F. Lepora, M. Pearson, B. Mitchinson, M. Evans, C. Fox, T. Pipe, K. Gurney, and T. Prescott, "Naive Bayes novelty detection for a moving robot with whiskers," in *Proc. IEEE ROBIO'10*, 2010, pp. 131–136.

[35] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott, T. Kyriacou, U. Nehmzow, C. Melhuish, and M. Witkowski, Eds., "Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task," in *Proc. Towards Autonomous Robot. Syst., TAROS'09*, 2009, pp. 19–24.

[36] M. Kaneko, N. Kanayama, and T. Tsuji, "Active antenna for contact sensing," *IEEE Trans. Robot. Autom.*, vol. 14, no. 2, pp. 278–291, 1998.

[37] A. Lamperski, O. Loh, B. Kutscher, and N. Cowan, "Dynamical wall following for a wheeled robot using a passive tactile sensor," in *Proc. IEEE Int. Conf. Robot. Autom., ICRA'05*, 2005, pp. 3838–3843.

[38] P. Flandrin, F. Auger, and E. Chassande-Mottin, "Time-frequency reassignment: From principles to algorithms," in *Applications in Time Frequency Signal Processing*, A. Papreou-Suppappola, Ed. Boca Raton, FL: CRC Press, 2003, pp. 179–203.

[39] S. A. Anderson, M. J. Pearson, A. G. Pipe, T. J. Prescott, P. Dean, and J. Porrill, "Adaptive cancellation of self-generated sensory signals in a whisking robot," *IEEE Trans. Robotics*, pp. 1–12, Aug. 2010.

J. Charles Sullivan received the Ph.D. degree from the University of the West of England, Bristol, U.K., in 2001.

Currently, he is a Senior Lecturer at the University of the West of England and a member of the Bristol Robotics Laboratory. His research interests include dynamics modeling and analysis, finite-element analysis, evolutionary algorithms and optimization, biomechanics, and biomimetic design.

Ben Mitchinson received the Ph.D. degree in nonlinear learning machines from the University of Sheffield, Sheffiend, U.K., in 2002.

Currently, he is an ATLS Senior Postdoctoral Researcher. His research interests include the vibrissal sensory system, and bioinspired information processing and robotics.

Martin J. Pearson received the B.Eng. degree in electrical and electronics engineering from the University of Manchester, Manchester, U.K., the MSc. degree in advanced technologies in electronics and the Ph.D. degree in 2008 from the University of the West of England, Bristol, U.K.

Currently, he is a Research Fellow at the Bristol Robotics Laboratory with research interests in bio-inspired robotics, neural control, and reconfigurable computing.

Mat Evans received the B.Sc. degree in psychology and the M.Sc. degree in computational and systems neuroscience from the University of Sheffield, Sheffield, U.K. Currently, he is working towards the Ph.D. degree in whisker-based tactile perception with the Adaptive Behaviour Research Group, Department of Psychology, University of Sheffield.

His research interests include active sensing, encoding of (tactile) stimuli, sensory-motor interaction, and embedded models thereof.

Nathan F. Lepora received the B.A. degree in mathematics and the Ph.D. degree in theoretical physics from the University of Cambridge, Cambridge, U.K.

He was a Research Fellow with Kings College, University of Cambridge for four years. Currently, he is a Research Associate with the Adaptive Behavior Research Group, Department of Psychology, University of Sheffield. His research interests include various topics between the interface of the physical and life sciences.

Charles W. Fox received the M.A. degree in computer science from the University of Cambridge, Cambridge, U.K., the M.Sc. degree in cognitive science from the University of Edinburgh, Edinburgh, U.K., and the D.Phil. degree in engineering from the University of Oxford, Oxford, U.K.

He was a Quantitative Hedge Fund Analyst at Algometrics Ltd. and Vantage Investment Advisory Ltd. for two years. Currently, he is a Research Associate with the Adaptive Behavior Research Group, Department of Psychology, University of Sheffield, and a co-investigator on the EU BIOTACT and EFAA projects. His research interests include applying machine learning techniques to neuroscience.

Chris Melhuish is Director of the Bristol Robotics Laboratory, a partnership between the University of Bristol and the University of the West of England, Bristol, U.K. His research interests include collective mobile robotics, robot-human interaction, neuro-inspired systems, and energy autonomy in robots. In addition to his Chair at the University of the West of England in Intelligent Autonomous Systems, he also holds the Chair in Robotics and Autonomous Systems, University of Bristol.

Prof. Melhuish is a Fellow of the British Computer Society, Fellow of the Institution of Engineering Technology, and is a Chartered Engineer.

Tony J. Prescott received the the M.A. degree in psychology from the University of Edinburgh, Edinburgh, U.K., the M.Sc. degree in applied A.I. from the University of Aberdeen, Aberdeen, U.K., and the Ph.D. degree in machine learning from the University of Sheffield, Sheffield, U.K.

Currently, he is a Professor of Cognitive Neuroscience at the University of Sheffield, where he directs the Active Touch Laboratory and co-directs the Adaptive Behavior Research Group. He is also the coordinator of the EU Framework 7 BIOTACT project. His research interests are focused on understanding the functional architecture of the vertebrate brain using techniques in computational neuroscience, robotics, and neuroethology.

Noname manuscript No.
(will be inserted by the editor)

A Tutorial on Variational Bayesian Inference

Charles Fox · Stephen Roberts

Received: date / Accepted: date

Abstract This tutorial describes the mean-field variational Bayesian approximation to inference in graphical models, using modern machine learning terminology rather than statistical physics concepts. It begins by seeking to find an approximate mean-field distribution close to the target joint in the KL-divergence sense. It then derives local node updates and reviews the recent Variational Message Passing framework.

Keywords Variational Bayes · mean-field · tutorial

1 Introduction

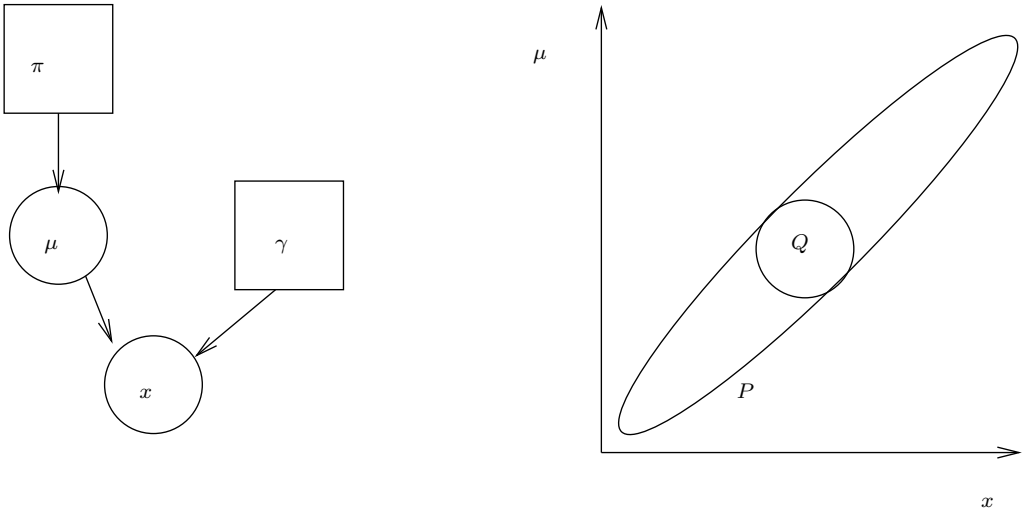
Variational methods have recently become popular in the context of inference problems, [1], [4]. *Variational Bayes* is a particular variational method which aims to find some approximate joint distribution $Q(x;\theta)$ over hidden variables x to approximate the true joint $P(x)$, and defines 'closeness' as the KL divergence $KL[Q(x;\theta)||P(x)]$. The mean-field form of VB assumes that Q factorises into single-variable factors, $Q(x) = \prod_i Q_i(x_i|\theta_i)$. The asymetric $KL[Q||P]$ is chosen in VB principally to yield useful computational simplifications, but can be viewed as preferring approximation where areas of high Q are accurate, rather than areas of high P. This is often useful because if we were to draw samples from, or integrate over, Q, then the areas used will be largely accurate (though they may well miss out areas of high P).

1.1 Problem statement

We wish to find a set of distributions $\{Q_i(x_i;\theta_i)\}$ to minimise the KL-divergence:

Charles W. Fox
Adaptive Behaviour Research Group
University of Sheffield, UK
charles.fox@sheffield.ac.uk

Stephen Roberts
Pattern Analysis and Machine Learning Research Group
Department of Engineering Science
University of Oxford, UK



(a) (b)

Fig. 1 (a) Graphical model for a population mean problem. Square nodes indicate observed variables. (b) True joint P and VB approximation Q.

$$KL[Q(x)||P(x|D)] = \int dx.Q(x) \ln \frac{Q(x)}{P(x|D)}$$

where D is the observed data and x are the unobserved variables and

$$Q(x) = \prod_i Q_i(x_i|\theta_i)$$

We sometimes omit the notational dependence of Q_i on θ_i for clarity. As the Q_i are approximate beliefs, they are subject to the normalisation constraints:

$$\forall i. \int dx_i Q_i(x_i) = 1.$$

1.2 VB approximates joints, not marginals

Q approximates the joint, but the individual $Q_i(x_i)$ are poor approximations to the true marginals $P_i(x_i)$. The $Q_i(x_i)$ components should not be expected to resemble – even remotely – the true marginals, for example when inspecting the computational states of network nodes. This makes VB considerably harder to debug than algorithms whose node states do have some local interpretation: the optimal VB components can be highly counter-intuitive and make sense only in the global context.

Fig. 1(a) shows a graphical model of an extreme example of mean-field VB breaking down, as a warning for the algorithms that follow. Suppose a factory produces screws of unknown diameter μ. We know the machines are very accurate so the precision $\gamma = 1/\sigma^2$ of the diameters is high. However no-one has told us what the mean diameter

is, except for a drunken engineer in a pub who said is was 5mm. We might have a prior belief π that μ is say 5±4mm (the 4mm deviation reflecting our low confidence in the report.) Suppose we are given a sealed box containing one screw. What is our belief in its diameter x? The exact *marginal* belief in x should be almost identical to our belief in μ, i.e. 5±4mm. Fig. 1(b) shows one standard deviation of the true Gaussian joint $P(\mu, x)$ and the best mean-field Gaussian approximate joint $Q(\mu, x)$. As discussed above, this Q is chosen to minimise error *in its own domain* so it appears as a tight Gaussian. Importantly, the marginal $Q_x(x)$ is now very small, and not at all equal to the marginal $P(x)$. We emphasise such cases because we found them to be the major cause of debugging time during development.

We also want to find the model likelihood, $P(D|M)$ for model comparison. This quantity will appear naturally as we try to minimise the KL distance. There are many ways to think about the following derivations, which we will see involves a balance of three terms called lower bound, energy and entropy. The derivation presented here begins by aiming to minimise the above KL distance. For example, other equivalent derivations may begin by aiming to maximise the lower bound.

1.3 Rewriting KL optimisation as an easier problem

We will rewrite the KL equation in terms that are more tractable. First we flip the numerator and denominator, and flip the sign:

$$KL[Q(x)||P(x|D)] = \int dx.Q(x) \ln \frac{Q(x)}{P(x|D)} = -\int dx.Q(x) \ln \frac{P(x|D)}{Q(x)}$$

Next, replace the conditional $P(x|D$ with a joint $P(x, D)$ and a prior $P(D)$. The reason for making this rewrite is that for Bayesian networks with exponential family nodes, the $\log P(x, D)$ term will be a be a very simple sum of node energy terms, whereas $\log P(x|D)$ is more complicated. This will simplify later computations.

$$P(x, D) = P(x|D)P(D)$$

$$\Rightarrow \ln P(x|D) = \ln P(x, D) - \ln P(D)$$

$$KL[Q(x)||P(x|D)] = -\int dx \left(Q(x) \ln \frac{P(x, D)}{Q(x)} + \ln P(D) \right)$$

The $\log P(D)$ term does not involve Q so we can ignore it for the purposes of our minimisation.

Finally, define

$$L[Q(x)] = \int dx \left(Q(x) \ln \frac{P(x, D)}{Q(x)} \right)$$

So that

$$KL[Q(x)||P(x|D)] = -L + \ln P(D)$$

So to minimize the KL divergence, we must *maximize* L.

The maximisation is still subject to the normalisation constraints:

$$\forall i. \int dx_i Q_i(x_i) = 1$$

$L[Q(x)]$ are *lower bounds* on the model log-likelihood, $P(D) = P(D|M)$ (where we generally drop the M notation as we are working with a single model only). The best bound is thus achieved when $L[Q(x)]$ is maximised over Q. The reason for L being a lower bound is seen by rearranging as:

$$\ln P(D) = L[Q(x)] + KL[Q(x|D)||P(x)]$$

Thus when the KL-divergence is zero (a perfect fit), L is equal to the model log-likelihood. When the fit is not perfect, the KL-divergence is always positive and so $L[Q(x)] < \ln P(D)$.

Another rearrangement gives

$$L[Q(x)] = \ln P(D) - KL[Q(x|D)||P(x)]$$

showing that the KL divergence is the error between L and $\ln P(D)$.

1.4 Solution of free energy optimisation

We now wish to find Q to maximise the lower bound, subject to normalisation constraints,

$$L[Q(x)] = \int dx Q(x) \ln \frac{P(x,D)}{Q(x)}$$

$$= \int dx Q(x) \log P(x,D) - \int dx Q(x) \ln Q(x)$$

$$= \langle E(x,D) \rangle_{Q(x)} + H[Q(x)]$$

where we define *energy* as $E = \ln P$ and *entropy*[1] as $H[Q(x)] = -\int dx Q(x) \ln Q(x)$. For exponential models, this will become a convenient sum of linear functions. By the mean field assumption:

$$L[Q(x)] = \int dx \left(\prod_i Q_i(x_i) \right) E(x,D) - \int dx \left(\prod_k Q_k(x_k) \right) \sum_i \ln Q_i(x_i) \quad (1)$$

Consider the entropy (rightmost) term. We can bring out the sum:

$$\sum_i \int dx \left(\prod_k Q_k(x_k) \right) \ln Q_i(x_i)$$

Consider the partitions $x = \{x_i, \bar{x}_i\}$ where $\bar{x}_i = x \backslash x_i$.

[1] This is Shannon entropy, used by convention.

$$= \sum_i \int dx_i d\bar{x}_i Q_i(\bar{x}_i) Q_i(x_i) \ln Q_i(x_i)$$

$$= \sum_i \langle \int dx_i Q_i(x_i) \ln Q_i(x_i) \rangle_{Q(\bar{x}_i)}$$

$$= \sum_i \int dx_i Q_i(x_i) \ln Q_i(x_i)$$

Substituting this into the right term of equation 1:

$$L[Q(x)] = \int dx \left(\prod_i Q(x_i) \right) E(x, D) - \sum_i \int dx_i . Q(x_i) \ln Q(x_i) \tag{2}$$

Now look at and rearrange the *energy* (left) term of equation 2, again separating out one variable:

$$\int dx \left(\prod_i Q_i(x_i) \right) E(x, D) = \int dx_i Q_i(x_i) \int d\bar{x}_i . Q(\bar{x}_i) E(x, D)$$

$$= \int dx_i Q_i(x_i) \langle E(x, D) \rangle_{Q(\bar{x}_i)}$$

$$= \int dx_i Q_i(x_i) \ln \exp \langle E(x, D) \rangle_{Q(\bar{x}_i)}$$

$$= \int dx_i Q_i(x_i) \ln Q_i^*(x_i) + \ln Z$$

where we have defined $Q_i^*(x_i) = \frac{1}{Z} \exp \langle E(x, D) \rangle_{Q(\bar{x}_i)}$ and Z normalises $Q_i^*(x_i)$. Substituting this new form of the energy back into equation 2 yields

$$L[Q(x)] = \int dx_i Q_i(x_i) \ln Q_i^*(x_i) - \sum_i \int dx_i Q_i(x_i) \ln Q_i(x_i) + \ln Z$$

Separate out the entropy for $H_i = H[Q_i(x_i)]$ from the rest of the entropy sum:

$$L[Q(x)] = \left\{ \int dx_i Q_i(x_i) \ln Q_i^*(x_i) - \int dx_i Q_i(x_i) \ln Q_i(x_i) \right\} + H[(Q(\bar{x}_i)] + \ln Z$$

Consider the terms in the brackets:

$$\int dx_i Q_i(x_i) \ln Q_i^*(x_i) - \int dx_i Q_i(x_i) \ln Q_i(x_i) = \int dx_i Q_i(x_i) \ln \frac{Q_i^*(x_i)}{Q_i(x_i)} = -KL[Q_i(x_i) || Q_i^*(x_i)]$$

What a lucky co-incidence! Though we started by trying to minimise the KL-divergence between large joint distributions (which is hard), we have converted the problem to that of minimising KL-divergences between *individual 1D* distributions (which is easier). Write:

$$L[Q(x)] = -KL[Q_i(x_i) || Q^*(x_i)] + H[Q_i(\bar{x}_i)] + \ln Z$$

Thus L depends on each individual Q_i only through the KL term. We wish to maximise L with respect to each Q_i, subject to the constraint that all Q_i are normalized to unity. This could be achieved by Lagrange multipliers and functional differentiation:

$$\frac{\delta}{\delta Q_i(x_i)} \left\{ -KL[Q_i(x_i)||Q^*(x_i)] - \lambda_i(\int_{s_i} Q_i(x_i)dx_i - 1) \right\} := 0$$

A long algebraic derivation would then eventually lead to a Gibbs distribution. However, thanks to the KL form rearrangement we do not need to perform any of this, because we can see immediately that L will be maximised when the KL divergence is zero, hence when

$$Q(x_i) = Q^*(x_i)$$

(The normalisation constraint on Q_i is satisfied thanks to the inclusion of Z in the previous definition of Q_i^*). Expanding back the definition gives the optimal Q_i to be

$$Q(x_i) = \frac{1}{Z} \exp\langle E(x_i, \bar{x}_i, D)\rangle_{Q(\bar{x}_i)}$$

where $E(x_i, \bar{x}_i, D) = \log P(x_i, \bar{x}_i, D)$ is the energy.

1.5 Solution as iterative update equations

Converting the equation above into an iterative update equation gives:

$$Q(x_i) \leftarrow \frac{1}{Z} \exp\langle E(x_i, \bar{x}_i, D\rangle_{Q(\bar{x}_i)}$$

where x_j is a hidden node to be updated; D are the observed data, and $\bar{x_i} = (x \backslash x_i)$ are the other hidden nodes. These updates give us an EM-like algorithm, optimising one node at a time in the hope of converging to a local minimum. Graphical models define conditional independence relationships between nodes, so for models having such structure there exists a *Markov blanket* set of nodes $mb(x_i)$ for each node x_i such that $P(x_i|\bar{x_i}) = P(x_i|mb(x_i))$. For undirected graphical model links, the Markov blanket is the net of neighbouring nodes of x_i; for directed Bayesian networks it is the neighbours of x_i augmented with the set of co-parents of x_i. By the definition of Markov blankets, we may write

$$\ln Q(x_i) \leftarrow \langle \ln P(x_i, mb(x_i), D\rangle_{Q(mb(x_i))}$$

where $mb(x_i)$ is the Markov blanket of the node x_i of interest.

1.6 Example: variational Bayesian mean update

Until very recently, variational implementations have consisted of calculating the iterative update equations by hand for each specific network model. These calculations required much algebra and were error-prone. We consider the simple graphical model shown in fig. 2: the task is to infer the mean μ component of the posterior joint of a Gaussian population of unknown precision γ given a set of M observations $D = \{D_i\}_{i=1}^M$ drawn from this population.

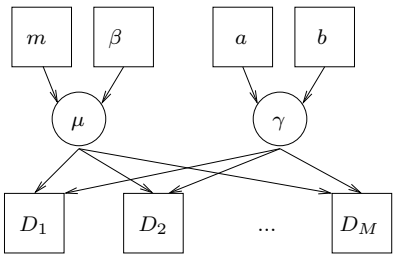


Fig. 2 Graphical model for variational example. Square nodes are observed.

Making the mean-field approximation and assuming conjugate exponential priors we have:

$$Q(x) = Q(\mu)Q(\gamma) = N(\mu; m, \beta^{-1})\Gamma(\gamma; a, b)$$

where m, β, a and b are constants specifying conjugate priors on the population parameters.

The update equation is

$$\ln Q(x_i) \leftarrow \langle \ln P(x_i, mb(x_i), D\rangle_{Q(mb(x_i))}$$

Substituting the variables for the mean update, $x_i = \mu$:

$$\ln Q(\mu) \leftarrow \int d\gamma \Gamma(\gamma; a, b) \ln P(\mu, \gamma, \{D_i\})$$

$$= \int d\gamma \Gamma(\gamma; a, b) \ln N(\mu; m, \beta^{-1})\Gamma(\gamma; a, b) \prod_i N(D_i|\mu, \gamma)$$

$$= \int d\gamma \Gamma(\gamma; a, b) \left\{ \ln N(\mu; m, \beta^{-1}) + \ln \Gamma(\gamma; a, b) + \sum_i \ln N(D_i|\mu, \gamma) \right\}$$

$$= \int d\gamma \Gamma(\gamma; a, b) \ln N(\mu; m, \beta^{-1}) + \int d\gamma \Gamma(\gamma; a, b) \ln \Gamma(\gamma; a, b)] + \sum_i \int d\gamma \Gamma(\gamma; a, b) \ln N(D_i|\mu, \gamma)$$

This simplifies greatly. First, integrands not dependent on μ can be discarded and replaced by a normalising Z, since we are only interested in the PDF for $Q(\mu)$:

$$\ln Q(\mu) = \int d\gamma \Gamma(\gamma; a, b) \ln N(\mu; m, \beta^{-1}) + \sum_i \int d\gamma \Gamma(\gamma; a, b) \ln N(D_i|\mu, \gamma) + \ln \frac{1}{Z}$$

The left term does depend on μ but not on γ, so the integral over the normalised $Q(\gamma)$ has no effect. Its integral is simply its integrand, so we can write

$$\ln Q(\mu) = \ln N(\mu; m, \beta^{-1}) + \sum_i \int d\gamma \Gamma(\gamma; a, b) \ln N(D_i|\mu, \gamma) + \ln \frac{1}{Z}$$

We next consider the integral containing data D_i. Writing out its log Gaussian energy expression in full, this term becomes

$$\int d\gamma \Gamma(\gamma; a, b) \ln N(D_i|\mu, \gamma) = \int d\gamma \Gamma(\gamma; a, b)[c + (D_i - \mu)\gamma(D_i - \mu)]$$

where c is its normalising constant. Rewriting up to normalisation gives

$$\propto (D - \mu) \left[\int d\gamma \Gamma(\gamma; a, b)\gamma \right] (D - \mu)$$

The required integral is now just the expectation (first moment) of a Gamma distribution, which can be found in standard statistics tables:

$$\langle \gamma \rangle = E[\Gamma(\gamma; a, b)] = ab^{-1}$$

(We notate pre- and post-multiplications by $(D-\mu)$ rather than a single multiplication by $(D - \mu)^2$ so that the multivariate Wishart case follows the same derivation and notation.) The term is now

$$\propto (D - \mu)\langle \gamma \rangle (D - \mu)$$

Substituting this back into the equation for $\ln Q(\mu)$ gives

$$\ln Q(\mu) = \ln N(\mu; m, \beta^{-1}) + \sum_i (D_i - \mu)\langle \gamma \rangle (D_i - \mu) + \ln \frac{1}{Z}$$

$$Q(\mu) = \frac{1}{Z} N(\mu; m, \beta^{-1}) \exp \left\{ \sum_i (D_i - \mu)\langle \gamma \rangle (D_i - \mu) \right\}$$

$$Q(\mu) = \frac{1}{Z} N(\mu; m, \beta^{-1}) \prod_i N(D_i|\mu, \langle \gamma \rangle)$$

Switching round the dependent variable we obtain the following. (For μ in Gaussian distribution, the flipped result is still Gaussian. For other distributions, it will be conjugate.)

$$Q(\mu) = \frac{1}{Z} N(\mu; m, \beta^{-1}) \prod_i N(\mu|D_i, \langle \gamma \rangle)$$

Finally we can use the standard equation for product of Gaussians to give

$$Q(\mu) = N(\mu|m', \beta'^{-1})$$

with

$$\beta' = \beta + M\langle \gamma \rangle$$

$$m' = \beta'^{-1}(\beta m + \langle \gamma \rangle \sum_{i=1}^{M} D_i)$$

The above illustrates how the general VB updates can be transformed into particular updates for graphical models. The $Q(\mu)$ component is meaningless by itself as VB aims to approximate the full joint rather than local marginals, so a more useful analysis would repeat the above to obtain the $Q(\gamma)$ updates as well – requiring even more algebra. We see that this process is time-consuming and error prone even for very simple networks such as the Gaussian population used here.

1.7 Variational Bayes with message passing

The previous method of by-hand derivation was time-consuming and tedious, though until recently was state-of-the-art. However the recent variational message passing (VMP) algorithm ([3], [4]) has shown how to automate these derivations in the case of conjugate-exponential networks. For such networks, the updates all have a standard form, involving the sufficient statistics and natural parameters of the particular node types. For unavoidable non-conjugate-exponential nodes (such as mixture models in particular) it is possible to make further approximations to bring them into the standard form. For conjugate-exponential networks, VMP should now be the standard method for variational Bayesian inference, replacing derivations by hand. For non-conjugate-exponential networks, VMP may still be useful if fast approximations are required at the expense of accuracy.

1.8 VMP details

A standard theorem [2] about exponential family distributions shows that the expectation of sufficient statistics are given simply by:

$$\langle u(x_i)\rangle = \bigtriangledown_\phi g(\phi)|_\phi$$

where the Dell symbol ($\bigtriangledown$) means we form a vector whose rth component is the derivative of g with respect to the rth component of the vector ϕ.

We will write x_i's set of parents as $pa(x_i)$; its set of children as $ch(x_i)$; its set of co-parents with respect to particular child ch as $cop(x_i; ch)$; and its set of co-parents over all children as $cop(x_i)$. We wish to compute the variational Bayesian update as in the previous section:

$$Q_i(x_i) \leftarrow \langle \ln P(x_i, mb(x_i), D\rangle_{Q(mb(x_i))}$$

Assuming the effects of D are already in the Markov blanket nodes, and separating, this becomes

$$\langle \ln P(pa(x_i)) + \ln P(cop(x_i)) + \ln P(x_i|pa(x_i)) + \ln P(ch(x_i)|x_i, cop(x_i))\rangle_{Q(mb(x_i))}$$

Simplifying and dropping constant parent and co-parent terms:

$$= \langle \ln P(x_i|pa(x_i))\rangle_{Q(pa(x_i))} + \langle \ln P(ch(x_i)|x_i, cop(x_i))\rangle_{Q(ch(x_i),cop(x_i))}$$

The children separate:

$$= \langle \ln P(x_i|pa(x_i))\rangle_{Q(pa(x_i))} + \sum_{ch\in ch(x_i)} \langle \ln P(ch|x_i, cop(x_i; ch)\rangle_{Q(ch,cop(x_i;ch))}$$

We will consider the two parts one at a time.

1.8.1 Messages from parents

A conjugate-exponential node x_i is parametrised by a natural parameter vector ϕ_i. By the definition of such nodes,

$$\begin{aligned}\langle \ln P(x_i|pa(x_i))\rangle_{Q(pa(x_i))} &= \langle \phi_i u(x_i) + f_i(x_i) + g_i(\phi_i)\rangle_{Q(pa(x_i))} \\ &= \langle \phi_i\rangle_{Q(pa(x_i))} u_i(x_i) + f_i(x_i) + \langle g(\phi_i)\rangle_{Q(pa(x_i))}\end{aligned}$$

As ϕ and g are multi-linear functions of the parent sufficient statistics (by construction), and using the mean field assumption, we can simply take their formulae (defined as conditional on single values of the parents) and substitute the expectations for the sufficient statistics, to get the expectation of the whole expression as required. So parents of x_i need only to send their sufficient statistic expectations to x_i as messages. (A concrete example is shown in section 1.9.)

1.8.2 Messages to parents

A key property of the exponential family is that we can multiply (fuse) similar distributions by adding their natural parameter vectors ϕ:

$$\begin{aligned}&\exp\{\phi_1 u(x_i) + f(x_i) + g(\phi_1)\}\exp\{\phi_1 u(x_i) + f(x_i) + g(\phi_1)\} \\ &= \exp\{(\phi_1+\phi_2)u(x_i) + f(x_i) + g(\phi_1+\phi_2)\}\end{aligned}$$

A second property is that by conjugacy, ϕ and g are also multi-linear in parental sufficient statistics. So we can always rearrange the formula by finding functions ϕ_{ij}, f_j, g_{ij} to make it look like a function of a parent $x_j \in pa(x_i)$:

$$\langle \ln P(x_i|pa(x_i))\rangle_{Q(pa(x_i))} = \langle \phi_{ij} u_j(x_j) + f_{ij}(x_j) + g_{ij}(\phi_{ij})\rangle_{Q(pa(x_i))}$$

As before, we may handle the expectation by using the multi-linear property to push all the expectations tight around the sufficient statistics. So from the point of view of the parent, this is written in terms of the sufficient statistic expectations of its child and co-parents. We can thus pass a likelihood message consisting of

$$\phi_{ij}\left(\langle u(x_i)\rangle, \{\langle u(cop)\rangle\}_{cop\in cop(x_j;x_i)}\right)$$

The parent may then simply add these to its prior parameters, by the first property.

Observed data nodes D may be treated as Delta distributions which send standard messages to their parents and children.

1.9 Example: mean, precision and data using VMP

To demonstrate the power of the VMP formalism, we consider the same scenario as used to hand-derive the VB updates in section 1.6. Using VMP we can quickly substitute the particular Gaussian and Gamma distributions into the VMP update equations and quickly obtain *all* network messages for mean, precision, and data nodes as follows. (Messages to D are not required in this particular example, but are useful in general for making inferences about unobserved data.) The exponential forms used here are standard [2].

1.9.1 Mean node (with known prior parameters)

Beginning with the conjugate-exponential form of the Gaussian distribution:

$$\ln P(\mu|m,\beta) = \begin{bmatrix} m\beta \\ -\beta/2 \end{bmatrix} \cdot \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} - g(m,\beta)$$

with

$$g(m,\beta) = \frac{1}{2}(\ln\beta - \beta m^2 - \ln 2\pi)$$

The message to child D is the expectation of sufficient statistics:

$$\langle \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} \rangle = \nabla_\phi g(\phi)|_\phi = \begin{bmatrix} \mu \\ \mu^2 + \beta^{-1} \end{bmatrix}$$

1.10 Computing the log-likelihood bound using VMP

VMP also makes computation of the log-likelihood bound simple. Recall that

$$\ln P(D|M) \geq L[Q(x)]$$

$$L[Q(x)] = \int dx Q(x) \log \frac{P(x,D)}{Q(x)}$$

$$= \langle \log P(x,D) \rangle_{Q(x)} - \langle \log Q(x) \rangle_{Q(x)}$$

Writing as the sum of individual node contributions from the universe of all (data and hidden) nodes $\Omega = x \cup D$,

$$= \sum_{\omega \in \Omega} \langle \log P(\omega|pa(\omega)) \rangle_{Q(\omega,pa(\omega))} - \langle \log Q(\omega) \rangle_{Q(\omega)} = \sum_{\omega \in \Omega} L_\omega$$

with

$$L_\omega = \langle \log P(\omega|pa(\omega)) \rangle_{Q(\omega,pa(\omega))} - \langle \log Q(\omega) \rangle_{Q(\omega)}$$

$$= \langle \phi^\pi_\omega u(\omega) + f(\omega) + g(\phi^\pi_\omega) \rangle_{Q(\omega,pa(\omega))} - \langle \phi^*_\omega u(\omega) + f(\omega) + g(\phi^*_\omega) \rangle_{Q(\omega)}$$

where ϕ^π are the *prior* natural parameters conditioned only on the parents, and ϕ^* are the posterior parameters, after child message are fused. $Q(\omega)$ uses the posterior parameters. By multi-linearity we can push in the expectations and simplify to obtain:

$$L_\omega = (\langle \phi^\pi_\omega \rangle_{Q(pa(\omega))} + \phi^*_\omega) \langle u(\omega) \rangle_{Q(\omega)} + \langle g(\phi^\pi_\omega) \rangle_{Q(pa(\omega))} + g(\phi^*_\omega)$$

which is simple to compute locally, from each node's received messages. (The term $\langle \phi^\pi_\omega \rangle_{Q(pa(\omega))}$ is computed by substituting in the received parent sufficient statistics expectations into the conjugate-exponential formula for ϕ_ω. By multi-linearity, the expectation can be pushed into the sufficient statistics.) The global model log-likelihood bound is computed by summing the contributions from all nodes, both hidden and observed.

1.11 Summary

In this tutorial we have seen how variational methods may be used to approximate joint posteriors with mean-field distributions. An long-hand calculation of variational inference was shown, then the more general variational message passing framework introduced, which greatly simplifies calculations for conjugate-exponential networks. Variational methods are not appropriate when the marginals or the correlations structure of the joint are required, but are useful in model comparison tasks when the joint is integrated out.

References

1. Hagai Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*. MIT Press, 2000.
2. J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley, 2000.
3. C.M Bishop, J.M.Winn, and D. Spiegelhalter. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems*, 2002.
4. J. Winn and C. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

LIGHTLY SUPERVISED LEARNING FROM A DAMAGED NATURAL SPEECH CORPUS

Charles Fox, Thomas Hain

Department of Computer Science, University of Sheffield, UK

ABSTRACT

Large corpora of transcribed speech are rare and expensive to acquire, but valuable for ASR systems. Of current research interest are corpora of natural speech, i.e. far-field recordings of multiple speakers in noisy environments. In the big data era there are many speech transcriptions collected for purposes other than ASR, which omit features required by typical ASR systems such as timing information. If we could recover training data from such 'found' corpora this would open up large new resources for ASR research. We present a case study for this type of data recovery – becoming known as 'lightly supervised learning' – for a highly damaged corpus called Family Life. We use a novel comparison of a parallel decode and forced audio alignment to iteratively select and grow good data. Family Life also has unusual data mislabelling problems which can be addressed by an integrated *tfidf* approach. These methods reduce WER on the corpus from 83.0 to 57.2. We also discuss a probabilistic loose string alignment approach which removes untranscribed 'icebreaker' speech.

1. INTRODUCTION

The UK Economic and Social Data Service (ESDS) contains around 5,000 data collections from social science studies from the 1960s to the present day, including raw audio interviews and transcripts. We work with one called Family Life[13] as a case study, showing how timing information can be recovered from a corpus where damaged and permuted text is wide-spread and long audio recordings do not come with any timing annotations. This opens up Family Life and similar corpora for use in social-, market- or other corpus-based analysis as well as turning them into resources for speech technology development - for example automatic speech recognition (ASR). Our aim is to retrieve good timing annotations on a word level for as much data as possible. We take a bootstrapping, iterative approach which alternates between forced audio alignments and training to grow the amount of data that can be aligned and improve trained models in the process. This paper tests a sequence of variations of increasing complexity on this strategy: first a basic form; then extending to handle Family Life's tape labelling permutations and an approach combining audio alignment with parallel biased decoding to select only good aligns for training.

Our method is closely related to other recent experiments in lightly supervised training. The field tends to use the same set of components – forced and loose alignments; text and audio alignments; scoring and iteration – in various architectures to approach the problem in different ways. [2] ran biased decodes, forced alignment between the decodes and transcripts, then loose audio alignment to the segmented decodes. [6] use manual annotation as a seed, then iteratively run biased decodes and trains directly from the decoder output. [8] run a biased decode then forced-align the audio to the decode. [15] use a loose audio alignment. Systems have mostly been developed on clean audio – lectures, broadcast news and audio-books – which lack the range of noise and tape labelling permutations of Family Life, so yielding better performance (lower word error rates) than we expect for Family Life. However our aim is not to recognise the audio especially well, but to achieve confidence in as many aligned hours as possible. In contrast to these systems, we deploy parallel forced audio aligns and decode, then compare them to select training data, and iterate. (Less related approaches have also used techniques such as SVMs[10], factor automata [7] and machine translation [5].)

2. THE ESDS FAMILY LIFE CORPUS

Family Life is a set of one-on-one interviews with elderly people describing life in the UK between 1900-1918, made in 1967[14, 13]. There were 452 interview subjects, and interviews last for around 3 hours. There total audio amounts to 1,354 hours, almost all is speech, all interviews are transcribed. Multiple transcribers used different conventions for fragments, dialect words, and non-speech annotations. The recordings were made with mono microphones on analogue tapes of various lengths, most commonly 45, 30, 15 and 60 minutes[1]. Not all tapes are filled with data: Some interviews use up whole tapes and continue immediately on the next one; others terminate the interview part way through a tape and begin on a new one, possibly on a different day. Interviews typically comprise 3 or 4 tapes (10 maximum), digitised to 44.1kHz stereo wav files.

Interviews are loosely structured over a set sequence of topic themes, such as school, religion, work. Questions are not prescribed but the corpus comes with an interviewer guide-sheet, giving an order in which topics are likely to occur. There are 44 named interviewers, conducting 8.2 ± 7.2 interviews each. A further 84 interviews are conducted by unknown interviewers. The 452 subjects are from 11 UK regions, with 40.8 ± 17.0 subjects per region. 230 subjects are female, 222 male, of ages 72 ± 7 years. An A-G social class is provided for each subject, with 64.5 ± 35.9 subjects in each. Other data provided includes rural/urban environment; marital status; job description and category; interview location. Being recorded in the 1960s with elderly subjects, the diversity of accents is notably greater than in modern recordings. Most subjects have strong regional accents, usually related to the interview region, where many have lived all their lives; the interviewers are 1960s sociologists, many having strong 'received pronunciation"(RP) accents. A few interviews are with non-native English speakers, sometimes in their native language (e.g. Welsh) but transcribed in English.

2.1. Data issues

Many tapes begin with 30s-2mins of non-transcribed 'icebreaker' conversations, in which the interviewer tests the tape recorder, sets the volume, and chats casually with the subject. Similarly, there are

[1] We use 'tape' to mean audio from one side of a two-sided cassette.

many untranscribed utterances that we call 'interjections', including prompts and repetitions by interviewers, and short remarks irrelevant to the interview topic by the interviewee. Speech is far-field, the tape recorder usually being placed on a table between interviewer and subject; common background noises include traffic; ticking clocks; furniture creaks. Most seriously, errors have been made in the tape labelling and/or digitisation. For many interviews, one or more tapes is either missing, permuted within the same interview, or permuted with tapes across interviews. Around 20 interviews are missing all transcripts or audio. As transcriptions were not made for ASR work, they contain *no timing information* and do *not even show the locations of the tape boundaries*. In places, often near starts of tapes, digitisation errors have resulted in tape noises and out-of-order audio, for example playing the tape for 30s then rewinding or fast forwarding it – putting the cued audio into the digitiser.

2.2. Processing

After text normalisation, the corpus contains 560598 script-style, alternating lines between interviewer and subject. There are 11.4M word tokens (59Mb text file), of 51,000 distinct words. Perplexity of the corpus under a generic language model [1] is 144. 49% of words only occur once, possibly due to many place names. An indication of topics is given by the top nouns: YOU, ME, MOTHER, SCHOOL, FATHER, PEOPLE, WORK, HOUSE, DAYS, CHILDREN, THINGS, SUNDAY, FAMILY, PARENTS, CHURCH, FRIENDS, JOB, MONEY, STREET, CLASS, BED, ROOM, BOYS, SISTER, TEA, CLOTHES, WAR, LIFE, SHOP. Common non-standard words include Scottish contractions (DIDNAE, COULDNAE) and terms and symbols for old English money.

Audio was down-sampled to 16kHz, 16bit and encoded as 12 PLP coefficients and c_0 [3]. First and second order derivatives were added forming a 39 dimensional feature vector. No further normalisation was applied. Using the meta-data, a test pool of 30 interviews was selected automatically, having similar region, age, and gender distribution to the full corpus. Two test sets consisting of different 5 minute extracts from each test pool member were constructed. They were manually checked to remove permuted or otherwise damaged interviews, and manually annotated with ground truth start and end times. The remaining interviews form the training pool. Each step of our process consists of creating a new time-annotated data set then training new acoustic models. Standard 3-state left-to-right hidden Markov Models (HMMs) are state-clustered phonetic decision tree ties state with 16-component GMMs to model output probabilities. Training follows a standard HTK mixup procedure [17]. Word error rates (WERs) are obtained with NIST sclite [9]. Decodes are based on HTK HDecode with per-interview biased language models built from the transcripts with the SRI language model toolkit[11]. Each alignment referred to is followed by a resegmentation step, cutting into short utterances using silence detection.

3. BASELINE ALIGNMENT AND TRAINING

Initial experiments were performed using a basic iterative bootstrapping approach as is commonly used to obtain acoustic models in a new domain. This approach serves as baseline and illustrates the complexity of the task at hand. Initially we have no knowledge of timing information, and thus are unable to split the data into manageable chunks (the length of interviews is up to 3 hours!). Thus a rough alignment of the entire corpus is performed using an initial model set, M0, trained on 177 hours of timing-annotated general meetings audio, recorded with close-talking microphones [1]. Once a rough alignment was obtained, a first Family Life model, M1, was trained. M1 is then used for alignment again, and the new alignment is used to train a model M2.

In all of these iterations, we attempt to align all 452 interviews using no prior segmentation – each is a single, 3 hour long segment – and alignment often fails due to aforementined data issues. Results are shown in table 1. Successful alignment, even with tight pruning, does not imply that the aligned times are actually correct. A slightly wider beam (400) than usual was chosen to allow alignments to recover from gross mismatches between labels and acoustics. Such settings are normal for a standard corpus. It is also important to note that there is no guarantee that the same data aligning in iteration 1 does align again in the second round, though the size of of the training sets tends to grow overall tables columns ex,ai,ah). As shown in the table, the better matching in-domain models allow this basic scheme was able to grow the aligned data at each iteration, to 217 for M2. However it was clear from a 94.4 test WERs that the training is going wrong: incorrectly aligned interviews are being admitted to training, and encouraging more incorrect alignments. Despite more aligns, the test scores are worse than the 83.0 of M0, showing that this basic iteration scheme is insufficient for this corpus.

4. HANDLING TAPE PERMUTATIONS AND BAD ALIGNS

An unusual cause of WERs in Family Life is its tape labelling errors. Given the size of the corpus a fully automated method for correcting such errors is desirable. There are three types of labelling error. First, two or more tapes within the same interview may be transposed or otherwise permuted (imagine the interviewer writing 'subject 453 side 1' and 'subject 453 side 4' on the wrong cassette sides, or the digitisers making similar errors). Second, tapes *between* interviews may be permuted (e.g. swapping 'subject 68 side 3' with 'subject 96 side 2'). Third, sections of transcripts and audio are missing. A more generic WER cause is failure of the basic iteration scheme to converge on correct alignments, resulting in training on bad data. We present methods for handling the most common, intra-interview, permutation case, and for filtering out bad alignments.

4.1. Tape label permutations

We assume audio and transcripts are complete and free of spurious data, i.e. there exists a permutation of the tapes resulting in good alignment. No timing information is available as initial forced alignments are unreliable, so the location of the tape boundaries in the transcript is also unknown and needs to be estimated automatically.

For each permutation of the N tapes, the transcript is split into N chunks corresponding to the tape lengths in such a way that they have approximately the same lengths as the corresponding tapes under that permutation. This is done by ensuring they have the same ratios of number of words as the tape sides have of audio duration. Each permutation of the transcript is then scored against a decoding of the whole interview, here using the M0 model. To score, each tape's decoding is treated a query, q, in a term-frequency/inverse-document-frequency [4] model,

$$tfidf(q, i, \{doc_j\}_j) = \sum_{w \in q} \frac{f(w,q) f(w, doc_i)}{f(w, \{doc_j\}_j)}, \qquad (1)$$

where $f(w, d)$ is the frequency of word w in document or set of documents d, and doc_i is the ith segment of the hypothesised permuted

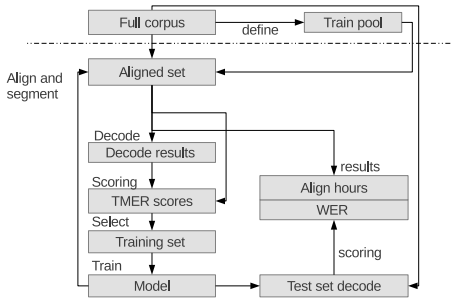


Fig. 1. System architecture for showing iterations of parallel alignment and decoding; and training. Each iteration creates an enlarged training set and better model.

transcript. A score sc is then assigned to a complete permutation,

$$sc(perm) = \sum_i tfidf(q_i, perm(i), \{doc_j\}_j), \qquad (2)$$

where $perm(i)$ is the permutation index for the ith segment of the permutation, i.e. we sum scores for each decode segment in the permutation. The best permutation for each interview is retained.

4.2. Filtering of mismatched audio

Another source of error occurs when alignments are very poor and HMM training learns from inconsistent or mismatched data. We saw that M1 and M2 align increasing quantities of data but with poor WERs suggesting that training on either aligns or decodes from these models – as in previous systems – is counterproductive. Under normal circumstances HMM training is known to recover well from such situations, but if the amount of badly aligned data is too large, significant modelling errors occur and compound. From a data selection perspective the outcome of each interview's alignment is binary: it either survives or it fails. However, we may look for subregions *within an interview* that appear to be well- or badly aligned. For those that align mismatched data can be filtered using output from recognition of the training set. A full decode of all interviews is obtained alongside the forced alignment. This is followed by standard DP alignment between the two time-segmented sequences. Rather than using word error rates of whole segments, we define a smoothed, per-word *temporal matching error rate, TMER*, by

$$TMER[t] = \frac{1}{N} \sum_{t'=t-N}^{t} (C[t'] - S[t'] - D[t'] - I[t']), \qquad (3)$$

where C,S,D and I are Boolean functions indicating correct, substitutions, deletions and insertions, and t ranges over the union of decoded and transcript words from the scoring. On inspection, as in fig. 2, the TMERs typically fall into two regimes – aligning and not-aligning – for most interviews, with a typical interview beginning with successful alignment, then losing tracking for a while, then perhaps regaining it again. The TMER in the non-aligned sections is expected to be roughly constant, reflecting the error rate of matching random words to audio, while any parts that align will be significantly better than this, reflecting the local word error rate. It was found to be sufficient to simply filter segments on the basis of thresholds on TMER values. Segments are assumed to be correctly aligned when $TMER[t] < -0.75$, with $N = 100$.

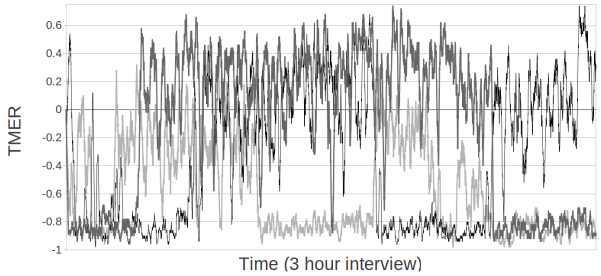


Fig. 2. Typical TMER scores for three interviews, each plotted over three hours.

A new new training set can be constructed based on TMER filtering, discarding all bad permutations and badly aligned segments. This leads to an extended version of the iterative bootstrap training cycle used in the previous section, and show in fig. 1. Note that at each iteration the best model so far is used to both forced-align and decode the current best training set. The outputs of these processes are synchronised by TMER to define a new, correct-align training set, which then is used to train the iteration acoustic model.

4.3. Joint icebreaker removal and permutation repair

The tdidf approach to permutation detection makes a big approximation in estimating the tape boundaries in the reference text: it assumes that the ratios of transcript words are equal to the ratios of the audio tape lengths. This is unfounded when, for example, full tapes are not used in interviews and there are many minutes of silence at tape ends, or when long icebreakers or interjections occur in some tapes. Icebreakers and interjections also reduce alignment quality, as they are forced to align against transcripts that do not include them.

A more complex approach to permutation repair avoiding tape boundary approximation, which could also handle icebreakers and interjections as a side effect, is given by considering a further combination of elements found in the reviewed previous systems: decoding, loose alignment, and string alignment. Here we first run a full decoding of each interview, then align the decode text to the transcript text to produce training data. String alignment is much faster than audio alignment, and can be performed with loose string HMMs that model permutation, icebreakers and interjections.

For each current transcription position, beginning at the transcript start, we built a loose HMM based on the transitions though the reference text from that position, but with probabilities of word insertions, deletion and substitution taken directly from previous NIST scoring runs. Incorrect word observation distributions were modelled by the first order corpus word distribution. The first state of each HMM is modelled as an icebreaker state, having a large self-transition (using an exponential prior with mean one minute icebreaker length) and generating all words from the corpus distribution. We then ran Viterbi alignment on each remaining tape decode, and selected the best tape and alignment. This provides a greedy $O(N^2)$ search over permutations, aligning one tape decode at a time and finding precise tape boundary in the reference text at each step. Some example aligned text from the model is shown in fig. 3.

4.4. Experiments

Interviews with more than six tapes and various data glitches were removed due to the computational load of $O(N!)$ tdidf search, so

we ran on 287 interviews. First we tested the accuracy of the tdidf algorithm itself. 74 interviews were reported to have incorrect permutations (25%), 21 of which were human checked. Of these, 5 were inter-interview permutations where the algorithm gave correct permutations; 1 was an inter-interview permutation reported as an incorrect permutation; 8 were between-interview swaps; 2 were interviews in Welsh language but transcribed in English. 5 were false positives having high levels of tape and environmental noise and/or strong dialects. (The algorithm may fail more when when there are many tapes: if we further exclude the six-tape interviews we have 60 of 272 problems, or 22%). Of another human checked 21 interviews reported with no permutation problems, 2 were false negatives (one in Welsh and the other missing 10% of the audio at the end).

Applying *tfidf* permutation removal and TMER alignment filtering in three iterations produced models M3-M5 whose results are shown in table 1. These indicate that by removing both bad permutations and badly aligned segments we can construct improved models (M5 WER of 63.6, vs M0 WER of 83.0) and alignments (165h from M5 vs 104h from M0) of more of the corpus. This is in contrast to the basic M1,M2 iterations, which aligned more (217h) data but apparently converging incorrectly as evidenced by their large WERs. However the iterative process appears to quickly reach a new local minimum state where little further gains are available from iterations (WER stuck at around 63), so a new technique is needed at this stage. To escape from this local minimum, we tried to restore, rather than discard, the badly permuted interviews. The best $tfidf$ score comes with an estimated (as the tape boundaries in the reference text are only approximate) permutation needed to re-order the tapes correctly. We created a new, larger training set including such repaired permutation, and performed two further iterations to produce the M6 and M7 models whose results are shown in table 1. Perhaps due to the crude approximation of tape boundaries, M6 does not align a larger quantity of data than its predecessor M5 422h vs 424h), though the aligns are of higher quality as evidenced by the increase in post-TMER data size (tsh of 273 vs 269) and improved WER (57.2 vs 63.6).

Our present implementation of string based alignment removes only icebreakers, not interjections, and has an implicit flat prior over possible permutations. Beginning with decodes and NIST probabilities from the M5 model (to enable comparison with the M6 tfidf method) we aligned 426 hours of the training set and obtained a WER of 76.1, a comparable performance to the M5 model on which its decodes were based. This shows that the algorithm is working but not yet very useful. Inspection of the aligns suggests that the main failure case occurs for very damaged interviews: if string align tracking is lost just once then both alignment and permutation estimation are irrecoverably destroyed, leading to worse training data that the tdidf method. This could perhaps be corrected by using an informative prior on permutations, so that when aligns give weak likelihoods, the default ordering would be used (or the interviews removed altogether, as for failed audio aligns in the tfidf/TMER version.)

5. CONCLUSIONS

We have demonstrated two new methods to repair damaged transcriptions having tape label permutations and no timing information, and produced good alignments for a useful subset (273 hours) of the natural speech Family Life corpus. We saw that simple forced audio align and train iterations are not sufficient for highly damaged corpora such as Family Life, but that by introducing a TMER-based comparison between parallel decodes and forced-aligns along with tfidf based permutation restoration, progress can be made.

Once a good model has been created from this process, the audio alignment may be dispensed with and a loose string alignment between decodes and transcripts used for lightly supervised training. We have showed an implementation of this with a WER comparable to the best audio/TMER approach, and seen that it shows potential for further improvements by removing interjections and using informative permutation priors.

Our methods use similar building blocks to other lightly supervised systems: mixtures of forced and loose alignment; string and audio alignments; data selection; in new combinations. The field has not yet matured to running comparison studies between methods and corpora but we suggest this as an important future topic.

Table 1. *Results. m=model; ai=number of aligned interviews; ex=number of exits (interviews that previously aligned and now fail); ah=aligned hours; adh=hours successfully aligned and decoded; tsh=TMER-selected hours; I,D=insertion,deletion rates.*

m	ai	ex	ah	adh	tsh	WER	I	D
M0	104	-	286	-	(95)	83.0	9.9	15.5
M1	89	15	285	-	-	95.7	5.7	23.7
M2	217	1	559	-	-	94.4	3.9	24.1
M3	133	9	333	206	150	64.0	15.3	12.6
M4	153	5	385	231	158	63.1	17.0	11.0
M5	165	0	424	263	269	63.6	17.5	10.6
M6	167	0	422	406	273	59.3	9.7	18.5
M7	-	-	-	-	-	57.2	18.6	9.9

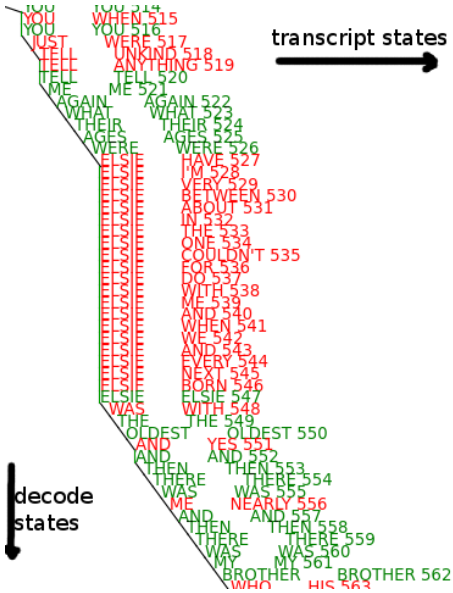


Fig. 3. String alignment. Green=correct, red=incorrect matches. The vertical line in an interjection or icebreaker, which could be removed on detection to match the transcript.

6. ACKNOWLEDGEMENTS

Thanks to the ESDS hosted by the University of Essex in Colchester, UK for access to the data. We benefitted from work by Vincent Wan for initial investigations. Funded by EPSRC EP/I031022/1.

7. REFERENCES

[1] Thomas Hain, Lukas Burget, John Dines, Philip N Garner, Asmaa el Hannani, Marijn Huijbregts, Martin Karafiat, Mike Lincoln, and Vincent Wan. The AMIDA 2009 Meeting Transcription System. In *Interspeech'10*, pages 358–361, 2010.

[2] Timothy J. Hazen. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proceedings of Interspeech*, 2006.

[3] Hynek Hermansky. Perceptual linear prediction (PLP) analysis of speech. 87(4):1738–1752, April 1990.

[4] Karen Spark Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[5] Tatsuya Kawahara, Masato Mimura, and Yuya Akita. Language model transformation applied to lightly supervised training of acoustic model for congress meetings. In *Proc. ICASSP*, 2009.

[6] Jean-Luc Gauvain Lori Lamel and Gilles Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech and Language*, 16:115–129, 2002.

[7] Pedro J. Moreno and Christopher Alberti. A factor automaton approach for the forced alignment of long speech recordings. In *Proceedings of ICASSP*, 2009.

[8] Sabine Buchholz Norbert Braunschweiler, M.J.F. Gales. Lightly supervised recognition for automatic alignment of large coherent speech recordings. In *Proceedings of Interspeech*, 2010.

[9] National Institute of Standards and Technology (NIST). *Speech Recognition Scoring Toolkit (SCTK) Version 2.4.0.* web resource: http://www.itl.nist.gov/iad/mig/tools,, 2010.

[10] Kengo Ohta, Masatoshi Tsuchiya, and Seiichi Nakagawa. Detection of Precisely Transcribed Parts from Inexact Transcribed Corpus. In *Proc. ASRU*, 2011.

[11] Andreas Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, 2002.

[12] Annika Hamalainen Antonio Calado Miguel Sales Dias Daniela Braga Thomas Pellegrini, Isabel Trancoso. Impact of age in asr for the elderly: Preliminary experiments in european portuguese. In *Advances in Speech and Language Technologies for Iberian Languages Communications in Computer and Information Science, pp 139-147*, 2012.

[13] P. Thompson and T Lummis. Family life and work experience before 1918, 1870-1973. *Economic and Social Data Service*, SN2000, 2002.

[14] Paul Richard Thompson. *The Edwardians: the remaking of British society*. Psychology Press, 1992.

[15] Anand Venkataraman, Andreas Stolcke, Wen Wang, Dimitra Vergyri, Jing Zheng, and Venkata Ramana Rao Gadde. An efficient repair procedure for quick transcriptions. In *Proc. Interspeech*. ISCA, 2004.

[16] Ravichander Vipperla. *Automatic speech recognition for ageing voices*. PhD Thesis, Edinburgh University, 2011.

[17] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, XA Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. The HTK book. 2006.

Formalising robot ethical reasoning as decision heuristics

Charles Fox
Sheffield Centre for Robotics, UK, charles.fox@sheffield.ac.uk

Abstract: Autonomous robotics ultimately seeks to replicate human functioning in all possible cognitive domains, and this must include the domain of ethics. Well-functioning ethical reasoning should enable rapid decisions to be made in difficult situations, such as where the interests of multiple parties are at stake. Decision making in autonomous systems is currently almost always construed in terms of Bayesian utility maximisation [9]. But how does ethical reasoning fit into this maximisation? Many concepts of ethics involve adherence to sets of specific rules. Again, how can such rule sit alongside utility maximisation decisions in automated systems? We consider classical philosophical concepts for ethics from the perspective of an engineer attempting to integrate them into the standard utility based decision approach.

Keywords: robot, military, ethics, utility, rules, heuristics

Most research concerning robotics and ethics has focused on ethical codes for humans, about when and where they should deploy their robots. But truly autonomous robotics ultimately seeks to replicate human functioning in all possible cognitive domains, and this must include the domain of ethics itself. For example, a truly autonomous military robot operating in theatre would need to make similarly complex ethical judgements to those made by human soldiers, which are learned though years of classroom study as well-instilled though important ceremonies and tradition. Well-functioning ethical reasoning should enable rapid decisions to to be made in difficult situations, such as where the interests of multiple parties are at stake. They could also enable an agent to know when to break explicit rules and orders if they are unethical.

To automate ethical reasoning and begin its implementation for robotics, we must define a conceptual framework to formalise ethical reasoning as an inference scheme. Decision making in autonomous systems is currently almost always construed in terms of Bayesian utility maximisation [9]. But how does ethical reasoning fit into this maximisation? Is any kind of altruistic ethics by definition in violation of standard utility maximisation? Or can altruisms be construed as particular forms of utility functions? Many concepts of ethics, especially religious systems, involve adherence to sets of specific rules. Again, how can such rules be formalised to sit alongside utility maximisation decisions in automated systems? This paper will consider classical philosophical concepts for ethics from the perspective of a robotics engineer attempting to integrate them into the standard utility based decision approach of the robotics field.

1 Utility maximisation

Current autonomous systems can be categorised according to whether they follow predetermined rules or make decisions. Examples of rule-following robots include mobile robots programmed to follow walls, solve mazes by turning left at every junction, or operate factory equipment using machine vision and if-then rules to generate actions based on the percepts. Decision making agents do not follow such pre-programmed behaviours, but internally simulate the effects of various possible actions in order to choose the best one. For example, classic AI game-playing agents search trees of board games for the best move; Monte Carlo supply simulations search for best actions to operate business supply chains [13] and financial trades [17]; and active SLAM [2] simulates mobile robot explorations to find best movements for building maps. 'Best' is always specified by some function $U^a(s)$ for the agent a, which maps world states s to real valued utilities, and the agent's sole objective is to maximise this utility function. If the world states are modelled as complex structures, then multiple parts of these structures $s_i \subset s$ may contribute independently and often additively ($U^a(s) = \sum_i s_i$) to the world state utility. For example, an active SLAM robot could assign additive utilities to a word state based both on how much map-building is achieved (positive utilities) and how much time and power has been expended on motion (negative utilities).

Predetermined rule-following agents' rules can usually be viewed as approximate heuristics to optimise a utility, whilst saving on the computation required for simulation of actions and world states. For example the simple 'always turn left' rule for maze-solving will maximise a utility function which assigns 1 to

escaping the maze and 0 to all other states; and pre-programmed rules for car-building robots optimise a utility function which assigns 1 to correctly built cars and 0 otherwise. Most AI players for real-time strategy games (e.g. as seen in the source code of `boswars.org`) do not attempt to simulate the future but use hand programmed rules which have been found empirically to yield behaviour which performs well at reaching the desired utilities (such as 'if the enemy has more than five tanks, build a defence turret'). A current research area in machine learning concerns the relationship between computationally complex inference problems and fast heuristics. For example, it is known that discriminative linear classification [12] gives identical classifications to full Bayesian inference of two generative distributions sharing a spherical covariance matrix. More recently research has examined how hierarchical Bayesian networks can be compiled into fast heuristics classifiers, such as the feed-forward systems of Helmholtz machines [16] and Deep Belief Networks [8] and kernelised Bayesian networks [22]. The field of action selection has seen similar approaches to 'compile' slow computational decisions into fast heuristic stimulus-response action selections. Reinforcement learning in particular [20] gives actions of this type, and psychological and machine learning research has shown how it can be trained as a secondary decision system from a primary full computational decision system [21, 11].

2 Consequentialism vs. absolutism

One way to categorise philosophical positions on ethics is to consider consequentialist views vs. absolutist views. Consequentialism means that only the final outcome state of a decision is considered in making ethical judgements, rather than the path taken to achieve that state. Absolutism means the opposite, that ethical judgements are taken to concern the actions taken rather than the outcome state. For example, various consequentialists have argued that it is good to take money (taxationists), property (communists) or life (eugenicists) from individuals by force on the grounds that the final state of society is made better off. Various absolutists have argued that these actions themselves are bad on the grounds of fundamental rights to property (the US constitution), life (human rights acts) or to freedom from all force (libertarians).

Absolutist views split into duty and virtue approaches. Duty ethics encodes judgements about actions in terms of the specifics of the actions themselves, for example 'Do not steal' and 'Do not kill' (Exodus 20), 'Do not use force except in defence' (Rand), 'put the strength of our group before any individual' (Hitler and Marx), 'treat others as you would like to be treated yourself' (Kant and Jesus). Asimov's laws are of this form[A].

In contrast, virtue ethics encodes judgements on actions according to what those actions reveal about postulated underlying latent variables of one's character. For example a robotic agent might be constructed to have characteristics concern for others: 80%, bravery: 50%, truth-telling:90%, keeping-promises: 95% Maximisation of the application of chosen traits - or sometimes the externally perceived application of them - becomes the basis for ethical decision making. From a robotics perspective we note that any trait maximisation can be re-written as a duty rule, so virtue and duty ethics for robots can be viewed together as absolutism. For example, we could construct duty rules such as 'Thou shalt maximise truth telling' or 'Thou shalt maximise displays of bravery'. [B]

3 Heuristic utility maximising agents

If we consider a robot with a given utility function (leaving aside for a moment what that function should be), we can see how the philosophical distinction between consequentialist and absolutist ethics maps onto the distinction between full probabilistic computation and the use of heuristics for decision making. Consequentialism means that all consequences (utilities) of an action in the future are taken into account and the best (highest utility) action selected. Consequences of most actions are probabilistic so very large computation is required to compute consequentialist utility accurately, by

$$(1)\ U^a(\alpha, s_0)=\sum_{t=0}^{\infty}\sum_{st+1} p(s_{t+1}|s_t, \alpha)U^a(s_{t+1})$$

where the sum is over all future times t and all future states s_t at each time that could occur as a result of performing action α in the initial state s_0. For example, the decision to buy a meat sandwich will affect the existence and livelihood of many animals in the future, as well as the agent's own hunger, the retailer's profits, the retailer's landlord's profits, tax revenue received by the government and so on. It is generally impractical for a human or machine to compute all consequences of real-world actions, let alone estimate all their probabilities and integrate over them to produce an expected utility.

Instead, humans rely on some form of heuristics to make most (and probably all) decisions. For split second and relatively unimportant decisions, the 'simple heuristics' of [7] and [3] provide rules of thumb such as 'buy any product whose brand is familiar' that achieve reasonably good (approximately optimal) utilities without the need to sit and compute for hours. Such rules can be acquired by various means. In the reinforcement learning literature,

stimulus-response rules can be acquired through repeated experience of actions and outcomes [14], or by repeated simulation of these same actions and outcomes (some authors [19] have suggested that simulations of and learning from events requiring fast decisions is a function of dreaming in humans.) Acquiring such experiences is at least computationally time consuming; if real world experiences are used it can be dangerous and wasteful, and if simulations are used it can be inaccurate as simulations never behave exactly as the real world does.

So an alternative way to learn heuristics is to be taught them. In chess, simply being told 'don't get your king out early' saves many thousands of lost exploratory games and analysis that would otherwise be required to induce this simple but powerful rule. In life, 'don't kill', 'sell in May, go away' and 'take a rest on Sunday' are similarly simple but powerful rules which have been found useful by various people over time, and passed from person to person. Heuristics in chess and life are learned by years or millennia of experience, and are thus extremely valuable cultural artefacts, with books of them painstakingly hand-copied by scribes and handed down generations.[C]

Importantly, an agent does not need to know *why* such rules work, only that they do work. To understand them may indeed require as much experience and/or computation as to induce them from scratch. Thus in teaching such rules to humans or agents we do not necessarily need to provide any cognitive understanding, rather we must provide training to make the rules part of the agent's behaviour. For example, to train the rule 'share with your neighbours' into an agent: rather than perform or simulate thousands of actions and learn the rule by trial and error, we might design specific world states and actions for them to perform and experience so that the rule is induced more quickly, such as ritual sharing coupled with a reward.[D] Such ritualised states and actions might bear little resemblance to the real world situations in which the induced rules will be applied, but still serve to help the agent learn and apply the rule in those real world situations later.

In humans [3], rats [14] and robots [6] it is often useful to use a mixture of consequentialist and rule-based approaches for utility maximisation. For example, classic AI game players perform a consequentialist tree-search of all possible games to some level of computational complexity, then evaluate each resulting state (e.g. 4 moves ahead) using heuristic rules. Human grandmasters appear to use a similar process, though using fewer moves ahead but better rules [4]. Daw and Dayan [14] designed experiments to show explicitly when rats switch from computation to heuristics. Most readers will have had their own experiences of using simple rules for relatively small decisions (selecting a known brand in a supermarket) and of deliberately and slowly cognating over bigger decisions (making spreadsheets to decide which house to buy).

We can thus view absolutist ethical rules as computational heuristics for optimising utility functions. Robots have the advantage that rules may be programmed in directly rather than learned though classrooms or rituals.

4 Ethical utility functions

Many of the absolutist rules discussed so far, such as rules for winning at chess, do not seem especially 'ethical' in character, while others such as 'do not kill' and 'put our group before any individual' do. Why is this? Ethical rules are those that take account of real-world effects on other agents, whereas non-ethical rules do not. Ethical rules can still be construed as heuristics to maximise an agent's utility, but in cases where the agent's own utility is affected by utilities of other agents. For example, an agent could conceivably have an interest in helping other agents in its team or army, or in helping its creator, or its creator's company, family, country, genetic group or species. In military cases, it could have interests in hindering members of similar groups on the other side. In such cases, the agent would ideally include weighted copies of these other agent's utilities in its own utility function, such as

(2) $U^a(s) = w_a{}^a U_a{}^a(s) + \sum_g w_g{}^a U^g(s)$,

where U_a is the agent's purely selfish utility (concerning the state of its own body, own resources etc), U_g are the utilities of other agents or groups, and w are weights.

We should distinguish this form from rules which affect others' utilities only as a side-effect of maximising the agent's own utility. For example, 'be honest in business' in a free market society is purely a way to maximise one's own long term utility though reputation building, rather than any attempt to help others for its own sake. In such cases, the utilities of others are not considered, they are treated simply as parts of the external world, and the problem reduces to a purely private optimisation, $U^a(s) = U_a{}^a(s)$. [E]

Let us consider some aspects involved in building robotic systems having ethical utility functions of this form. For example, consider a military robot operating in theatre, alongside human soldiers, civilians and robots from its own side, from the enemy side, and - as in a typical modern conflict - from many additional allies of both sides having varying degrees of trust and kinship with the principal actors. The recent killing of

Osama bin Laden in Pakistani territory provides an interesting case of such a third party: US troops were presumably acting to maximise the interests of their colleagues first, then their commanders and country, also to minimise the interests of Al Qaeda, but with some weighted but not total respect for Pakistan's government and local population, and international human rights and national reputations. Had the attack been carried out by a fully autonomous drone then these factors would all have to be modelled in its ethical utility function.

In addition to specifying weights on other parties' utilities in eqn. 1, a robot designer would also need a way to estimate what the other parties' utilities are. Write $U_g^a(s) \approx U^g(s)$ for agent a's estimate of g's utility for state s. This estimation can be highly non trivial, especially in political cases such as the Pakistan example where agents may conceal their true objectives. [F]

The situation becomes even more complex when the other affected agents' utilities are also ethical, i.e. including models of yet further agents' utilities, perhaps leading to an endless recursion. One might have for example,

$$(3) \quad U^a(s) = w_a U_a^a(s) + w_b^a U_b^a(s)$$

$$U^b(s) = w_b U_b^b(s) + w_a^b U_a^b(s),$$

which is reminiscent of the story of the two friends, one of whom sells his hair to buy a watchstrap gift for the other, who sells her watch to buy a hair brush for the first. Here both ethical utilities are dependent primarily on the other's, leading to a paradoxical regress in deciding whether they are happy or not. In theatre we see similar cases, where agents in a group often value each other's lives more than their own.

Further modelling complications arise from the relationship of utility to risk. While many current robotics systems assume that the total utility of an action is the average (expected) utility over possible future states, in the real world humans generally assign additional utility to actions which have the lowest outcome uncertainty *ceteris paribus*. In finance this is especially clear: equities tend to give bigger returns because those returns are less certain than those of bonds, and finance has a well developed theory of the utility of risk leading to models such as

$$(4) \quad U^a(\alpha, s_0) = \sum_{t=0}^{\infty} \sum_{st+1} p(s_{t+1}|s_t, \alpha) U^a(s_{t+1}) + \lambda S[U^a(s_{t+1})]_{p(st+1|st, \alpha)},$$

where S is the standard deviation of the future utility and λ is a risk appetite parameter. In theatre there are similar preferences - if an action is highly likely to lead to one outcome, it is *ceteris paribus* preferable to another with uncertain outcomes, because the second would require twice as much work to go into planning for the future scenarios. So utility should be augmented with risk appetite λ to account for the cost of this work.

Similar to assigning utility by risk appetite is the issue of distributing utility across groups of agents. If an agent has to choose between actions leading to $1000 of utility each for 1000 soldiers, or to $1,000,000 for a single soldier[G], which should it chose? This is the classic problem of utilitarianism *vs*. egalitarianism in robotic decision form, and as with risk premia there is no obvious solution but again a parameter ρ similar to the price of risk could be set to model whatever trade-off is preferred by its creators,

$$(5) \quad U_g^a(s) = \sum_{i \in g} U_i^a(s) + \rho S(U_i^a(s))_{p(i|g)}.$$

Where utilities of group (team, family, nation etc) are concerned, an agent also needs a function that assigns a degree of membership of individuals to these groups. Who exactly is a 'soldier', 'civilian', 'enemy combatant', 'Jew' or 'Taliban'? Do these categories depend on individuals' behaviour, clothing, contracts, beliefs or genetics? Some of these features can be measured in theatre, others cannot.

Computing non-ethical expected utilities $U^a(\alpha, s)$ is already computationally hard, but when these additional factors are included for ethical utilities it appears that all hope must be abandoned of computing exact consequentialist utilities, especially in real time such as in theatre. Hence the use of absolutist rules for ethical decision making is a much more practical approach to practical problems. Ethical utility-maximising robots must, like their human counterparts, rely on the centuries of wisdom that have been compiled into simple rules rather than trying to estimate and simulate their own and *others*' future utilities from scratch at every decision. [H]

The practical robot designer must therefore be concerned with *which* absolutist rules to implement. Should we build libertarian robots that maximise their non-ethical utilities but vote for free markets; national socialist or communist robots that kill anyone from other nations or classes and protect members of their own nation or class? Old Testament robots that 'Do not spare him, but kill men and women, children and infants' (1 Samuel 15:3)? New Testament robots that treat their enemies as their friends? None of these rules, nor Asimov's laws, seem appropriate for a modern military robot.

The standard criticism of absolutist systems, many of which contradict each other, is 'why these particular

rules?' And the standard answer is 'because they work at this particular time and place'. Having arrived at the need for absolutist rules from the need to compile down complex utility computations into fast action selections, we can give a more precise answer, 'because they are, empirically, the most accurate computationally efficient approximations to optimal action selection under our desired utility function that we currently know'. Does this lead to relativism? No, arbitrary rules cannot be used. It is pragmatism: given a desired complex utility function, we search empirically for fast rules that can be used to find actions that are useful. The question of what goes into this given target utility is not one that concerns engineers. If they are designing for a company, it will be provided by the customers; if for a democratic nation, by the voters; and if for a dictatorship, by the dictator. In practice, we will find that human soldiers already have very well developed rules taught to them though classrooms, exercises and rituals that serve approximately this purpose and which could be implemented similarly in their robot counterparts. They might not be perfect or complete — neither are our laws of mathematics or physics — but they are the best we have.

5 Public ethics *vs*. private ethics

We may encounter scenarios where what an agent communicates *about* ethics is as important as its actual ethical rule set. Any agent that has to build trust with others - such as soldiers, eBay traders, and teachers - is concerned with other agents' perceptions of its own rule-set. It might be argued that this is what 'trust' in an agent means - being able to predict its future actions. This form of ethical perception is a central aspect of many virtue and duty absolute ethical systems. But it is not always the case that what an agent should (in the sense of maximising its utility) reveal about itself is identical to its actual rule-set. Machiavellian ethics recommends that an agent promotes altruistic ethics to its associates — so that they will include the agent's own utility in their utility functions - whilst simultaneously excluding any altruism from one's own private utility. Such strategies must often be balanced against the negative utility of being exposed - hypocrisy - and consequent loss of trust by others. Automating this type of 'second-order' ethical reasoning would be an additional research area after implementing ethical systems for individual agents. It is possible that it could draw from game-theoretic research such as automated Poker playing [11] which focuses on the inference and concealment of strategies between players. Taken to extremes, this kind of game playing between many members of whole societies is called 'politics'.

6 Conclusion

Robots with increasing levels of autonomy are progressing rapidly from labs to companies to active military service, and many autonomous robot engineers will have been involved in collaborations such as BAE Systems autonomous Wildcat [25] or QinetiQ's UAVs [10]. While most robots in critical real world situations still have a human 'in the loop', [15] it is becoming more important to automate more and more behaviour in the military and civilian worlds, for reasons of speed, accuracy and even cost efficiency. The humane dream of pure robot-on-robot warfare is still far away, and with both humans and robots now in theatre together we are rapidly reaching a point where some human decisions concerning ethics need to be automated.

We have examined how to formalise absolutist, rule-based ethical reasoning as a heuristic approximation to consequentialist utility maximisation as a step towards a framework for implementation on robotic agents. The latter is the ideal form of reasoning but is usually computationally intractable, especially for real-time split-second decisions in military theatre. An agent's utility function, and heuristic rules, can be viewed as ethical when they take the utilities of other agents into account. Ethical utility functions become even more complex and have even greater need for compilation into absolutist rules. Ethical utility functions have several parameters which must be specified. These include weights for immediate self-utility vs. utilities of other agents of various groups; models specifying degrees of membership of agents into such groups; models of other agents' own utility functions (which may be recursive); utilities of the distribution of other agents' utilities over possible outcomes (risk appetite) and over populations (egalitarianism).

We noted that the underlying utility function for an automated agent should be specified by its designer to reflect the utility of the customer or voter requesting and paying for the work. Customers and voters may have utilities that prioritise themselves, their families, communities, ethnic groups, nations, cultures or species, but the nature of their specifications is none of the business of a robot engineer[I]. These requirements may change over time, so for example, absolutist rules for military robots may need to be 'recompiled' from new utility functions if the commissioning government changes its policies or if a new government is elected. Rules may be searched for by specifying an explicit utility function, running many simulations or real experiences, and testing candidate heuristics on them against the utility (cf. [18]). Or, more simply, they could be obtained by asking humans who already know useful rules to share with robots and their designers.

References

[1] C. Rohlfs. The government's valuation of military life-saving in war: a cost-minimization approach. Papers and Proceedings of the American Economic Association:96(2):39-44, 2006.

[2] L. Carlone, Jingjing Du, M.K. Ng, B. Bona, M. Indri. An application of Kullback-Leibler divergence to active SLAM and exploration with Particle Filters. Intelligent Robots and Systems (IROS), 287 -293, 2010.

[3] D. Kahneman. Thinking Fast and Slow. Macmillan, 2011.

[4] D.H. Holding. The psychology of chess skill. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985.

[5] F.M. Kamm. Harming Some to Save Others. Philosophical Studies, 57:227-60, 1989.

[6] C.W. Fox, M. Humphries, B. Mitchinson, T. Kiss, Z. Somogyvari, T.J. Prescott. Technical Integration of Hippocampus, Basal Ganglia and Physical Models for Spatial Navigation. Frontiers in Neuroinformatics, 3(6), 2009.

[7] G. Gigerenzer, P.M. Todd, ABC Group. Simple Heuristics That Make Us Smart. Oxford University Press, 2000.

[8] Geoffrey E. Hinton, Simon Osindero. A fast learning algorithm for deep belief nets. Neural Computation, 18:2006, 2006.

[9] J. M. Bernardo, A. F. M. Smith. Bayesian Theory. John Wiley & Sons, 2000.

[10] Graham S. Horn Jeremy W. Baxter. Fly-by-agent: Controlling a pool of UAVs via a multi-agent system. Knowledge-Based Systems:21:232-237, 2008.

[11] Kevin Korb, Ann Nicholson, Nathalie Jitnah. Bayesian Poker. Proc. Uncertainty in AI 343—350, 1999.

[12] Jean-Luc Gauvain Lori Lamel, Gilles Adda. PCA vs LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23:228-2339, 2001.

[13] M. Taylor, C. Fox. Inventory management with dynamic Bayesian network software systems. Proc. Int. Conf. Business Information Systems, Springer LNBIP, 2011.

[14] N.D. Daw, Y. Niv, P. Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatum systems for behavioral control. Nature Neuroscience, 8:1704-1711, 2005.

[15] N.E. Sharkey. Automating Warfare: lessons learned from the drones. Journal of Law, Information and Science, 21(2), 2012.

[16] P. Dayan, G.E. Hinton, R.M. Neal, R.S. Zemel. The Helmholtz Machine. 1995.

[17] P. Glasserman. Monte Carlo Methods in Financial Engineering. Springer, 2003.

[18] R. Axelrod. The Evolution of Cooperation (Revised ed.). Perseus Books Group, 2006.

[19] Revonsuo A. The reinterpretation of dreams: an evolutionary hypothesis of the function of dreaming. Behav Brain Sci., 23(8):904-1121, 2000.

[20] Richard S. Sutton, Andrew G. Barto. Reinforcement Learning: An Introduction. IEEE Trans. Neural Networks, 9(5):1054-1054, 1998.

[21] Steven M LaValle. Planning Algorithms. 2004.

[22] Tommi Jaakkola, David Haussler. Exploiting Generative Models in Discriminative Classifiers. NIPS:487-493, 1998.

Notes

[A] 1. A robot may not injure a human being or, through inaction, allow a human being to come to harm. 2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law. 3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws. 0. A robot may not harm humanity, or, by inaction, allow humanity to come to harm.

[B] For conscious humans the duty/virtue distinction is arguably more important, as philosophers such as Confucius and Plato argued that the experiential qualia of being a good character is a pleasurable end in itself, but we assume that robots constructed with current technology do not experience qualia.

[C] Eventually lists of rules may become so long that it becomes useful to construct abstract theoretical concepts to unify them, as a memory aid. Again, it is not necessarily the case that such concepts must correspond to the actual underlying generative process, for example postulating Chi forces helps to unify many martial arts principles and allows new useful moves to be derived, even if there are no noumenal Chi forces involved in the actual generative process of fighting.

[D] Ritual sharing of food may be especially useful as eating food is a reward in itself, reinforcing the desired sharing aspect.

[E]Whilst is it true that benefits to others are merely side-effects of individual trade actions I such a system, the act of its members voting to enforce the existence of that system is an ethical action which does take account of such side effects. When voting for freedom, agents can and do take account of the utilities of others in their society by choosing that will benefit them in this way.

[F]The difficultly of estimating others' utility functions leads to a justification for libertarianism over central planning: that the state can never know the details of these different functions, e.g. When assuming that everyone wants a grey car.

[G]estimated by [1] as the implicit modern value placed on a WWII soldier's life by the US military.

[H] Evidence for human use of both and conflicting consequentialist and absolutist ethical systems is provided by some of the trolley problems of [5], which as in Daw and Dayan's [14] illustration of rats switching between computation and heuristic decisions, similarly show how human ethical judgements can be made in both ways, and try to determine in which situations each system tends to dominate. For example a cognitive illusion tends to occur in the case of asking a human whether to push a button that would kill one person if the absence of a press would kill five people. Here consequentialist ethics says to push the button but most people's absolutist rules say not to. It seems likely that as in Daw and Dayan, humans will switch to using absolutist rules more as the problem complexity increases, and/or the time allowed for decision making decreases. Such lab based scenarios are interestingprecisely because they are simple enough to allow full consequentialist reasoning, but in the real world most problems are too complex and so absolutist rules become the norm.

[I]While the author respects the right of individuals to vote for the interests of their ethnic and other groups, he also notes that history shows that those groups are usually best served in the long term by acting to help their neighbours too, and hopes that voters and their representatives are smart enough to take this into account in their decisions. He respects the right of individuals to choose whether or not to be professional engineers on projects conflicting with their personal utility functions. Such functions are personal beyond the scope of the profession and this paper.

Thanks to Anand Ramamoorthy and Andrew Bower for comments.

Where wall-following works: case study of simple heuristics vs. optimal exploratory behaviour

Charles Fox

Sheffield Centre for Robotics, University of Sheffield, UK

Abstract. Behaviours in autonomous agents – animals and robots – can be categorised according to whether they follow predetermined stimulus-response rules or make decisions based on explicit models of the world. Probability theory shows that optimal utility actions can in general be made only by considering all possible future states of world models. What is the relationship between rule-following and optimal utility modelling agents? We consider a particular case of an active mapping agent using two strategies: a simple wall-following rule and full Bayesian utility maximisation via entropy-based exploration. We show that for a class of environments generated by Ising models which include parameters modelling typical robotic mazes, the rule-following strategy tends to approximate optimal action selection but requires far less computation.

1 Introduction

Autonomous behaviours can be categorised according to whether they follow predetermined rules or make decisions. Examples of rule-following robots include mobile robots programmed to follow walls, solve mazes by turning left at every junction, or operate factory equipment using machine vision and if-then rules to generate actions based on the percepts. Decision making agents do not follow such pre-programmed behaviours, but internally simulate the effects of various possible actions in order to choose the best one. For example, classic AI game-playing agents search trees of board games for the best move; Monte Carlo supply simulations search for best actions to operate business supply chains [17] and financial trades [8]; and active SLAM [2] simulates mobile robot explorations to find best movements for building maps. 'Best' is always specified by some utility function, and the agent's sole objective is to maximise this utility function. For example, an active SLAM robot could assign additive utilities to a world state based both on how much map-building is achieved (positive utilities) and how much time and power has been expended on motion (negative utilities).

During our experiments on Active SLAM [3], [6], we noticed a tendency for wall-following behaviour to emerge from the full Bayesian decision computations. This was surprising, because wall following can be obtained so easily from simple rules as well as by these large, expensive computations. This suggested that in at least some environments, full Bayesian computation may be unnecessary and well-approximated by the simple wall-following rule. In the present paper we

perform controlled experiments in a simplified class of environments to examine how general this approximation might be.

In purely theoretical decision making, the utility maximisation approach is always the 'correct' one, as it is by definition mathematically optimal. However, practical deployed systems tend to use simple rules instead. Such simple-rule based systems were popular in the Artificial Life research of the 1990s, before the advent of Bayesian machine learning, which has shifted the focus to utility maximisation. In Artificial Life, inspiration was drawn from natural systems such as insects following simple rules such as 'turn left if about to hit something' in order to form apparently complex behaviours [12] which were formalised into robotic systems such as Braitenberg vehicles [1]. Rats are known to perform much of their exploration by wall-following behaviour [10], the implementation of which has been modelled as emerging simply from maximum contact minimum impingement (MCMI) behaviour in conjunction with other biomimetic control systems [14]. State of the art robotic vacuum cleaners still use similar simple rules, such as 'follow walls' or 'if hit something, turn a random angle and move in a straight line' [16].

What is the relationship between these rules and the ideal utility maximisations? From a Bayesian perspective, we consider that an agent's objective when in some state s_0 is to maximise some utility function,

$$U(\{a_t\}_{\forall t}, s_0) = \sum_{t=1}^{\infty} \sum_{s_t} p(s_t|s_0, a_{0:t}) R(s_t), \tag{1}$$

where s are states of the world (which may include the agent's knowledge state), t is discretised time, a are actions and $R(s_t)$ are (discounted present value) rewards for being in states s_t. In general full planning, the agent must consider and optimise over all of its future action sequences to select a plan or sequence of actions $\{a_t\}_{\forall t}$,

This optimisation is generally computationally infeasible, especially for real-time systems, being exponential compute time in the number of time steps when the infinite sum is truncated. Even if a myopic approach is assumed as an extreme approximation to full planning, and we assume that all utility contribution occurs at the next greedy step,

$$U(a_t, s_t) \approx \sum_{s_{t+1}} (s_{t+1}|s_t, a_t) R(s_{t+1}), \tag{2}$$

or if a Q-learning temporal difference approach is taken (and ignoring the issue of how to compute Q),

$$U(a_t, s_t) \approx \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t) Q(s_{t+1}), \tag{3}$$

with

$$Q(s_\tau) \approx R_\tau + \max_{\{a_t\}_{t=\tau:\infty}} \sum_{t=\tau+1}^{\infty} \sum_{s_t} p(s_t|s_\tau, a_{\tau:t}) R(s_t), \tag{4}$$

we can still be left with an exponentially hard problem to compute this and select an action because the state s_t can itself be a joint variable $s = (s^1, s^2, \cdots s^S)$ having a state space with size exponential in S which must be summed over (e.g. the joint states of many cells in a map). Furthermore, it is sometimes the case that for *each* s, $R(s)$ may be similarly exponentially hard to compute (as will be the case in the present study). Therefore it is imperative that a real-time[1] agent uses some fast approximation to this computation instead of sitting still doing computation. The psychological literature has identified many 'heuristics and biases' [11] and 'simple heuristics' [7] showing cases where humans behave as if following simple rules rather than computing utilities. Daw and Dayan [4] have shown similar heuristic behaviour in rats, and further shown the existence of complexity boundaries in families of tasks which cause the rats to switch from full utility computation to the use of heuristics.

As a case study of this general problem of understanding how simple rules relate to full Bayesian utility computation, we will study in this paper a simple form of greedy active mapping – a component of Active SLAM [2] – problem, and its relationship to the popular wall-following heuristic used in many exploring robots. In what environments does wall-following tend to work so well in practice?

1.1 Active mapping

Active SLAM is the Bayesian formalisation of the problem of how a robot should act to explore and localise itself to build a map of an environment and its location in it. Active SLAM is defined in terms of maximising the expected reduction in the entropy of an agent's belief about its map of the world including its own location in it, or informally, 'where to look next'. Active *mapping* is a simplified case of this problem which considers only the map of the world, and assumes the agent has a perfect location sensor at all times.

We assume the physical world w is made of discrete cell locations w^i which are either occupied or not, $w^i \in \{0, 1\}$. A map of the world at time t, m_t, is similarly made of discrete cells m_t^i, and an agent's belief about what the true map is, is represented by a distribution $P(m_t|o_{1:t})$ given all available observations up to time t, $o_{1:t}$.

While this belief may be an inseparable joint distribution over all the cells, its entropy can be (crudely but standardly [2]) approximated by assuming independent cells and summing the marginal entropies,

$$H[P(m_t|o_{1:t})] \approx \sum_i H[P(m_t^i|o_{1:t})], \tag{5}$$

with

$$H[P(m_t^i|o_{1:t})] = P(m_t^i|o_{1:t}) \log P(m_t^i|o_{1:t}) + P(\neg m_t^i|o_{1:t}) \log P(\neg m_t^i|o_{1:t}), \tag{6}$$

[1] Where 'real-time' here means having a large negative utility for slow decision making.

where $P(m_t^i|o_{1:t})$ is the occupancy probability for a cell, and $P(\neg m_t^i|o_{1:t}) = 1 - P(m_t^i|o_{1:t})$ is the probability of non-occupancy.

At each time t, greedy active mapping seeks to choose the best action $\hat{a}_t$ which reduces the entropy of the map belief as much as possible,

$$\hat{a}_t|o_{1:t} = \arg_{a_t} \max U(a_t, o_{1:t}) \tag{7}$$

$$= \arg_{a_t} \max \left(H[P(m_t|o_{1:t})] - H[P(m_{t+1}|a_t, o_{1:t+1})]\right) \tag{8}$$

$$= \arg_{a_t} \min H[P(m_{t+1}|a_t, o_{1:t+1})]. \tag{9}$$

At decision time we do not know what the future observations o_{t+1} will be, so greedy active mapping marginalises the decision over their expectation,

$$\hat{a}_t \approx \arg_{a_t} \min \langle H[P(m_{t+1}|a_t, o_{1:t})]\rangle_{P(o_{t+1}|m_t, a_t)}. \tag{10}$$

2 Methods

To isolate the phenomenon of interest in our experiments, we made as many simplifying assumptions as possible to produce a micro-world model of more general robotic exploration. We ignore the localisation part of the Active SLAM task and assume perfect (GPS-like) agent knowledge of its location at all times. Rather than model directional distal sensors such as vision or laser range finders, we assume 360 degree proximal sensors (such as touch sensors all around its body). We discretise the world into cells, and use a hexagonal grid as shown in fig. 1. (This has the modelling advantage of each cell having six well-defined, equally important neighbours rather than having to handle diagonal contacts in a square grid).

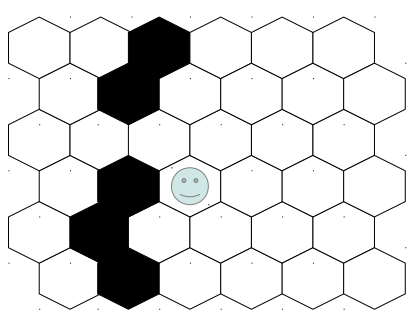


Fig. 1. A physical map. Black=occupied, white=unoccupied. The world is discretised into hexagonal cell locations. The agent position is shown by the face.

In order to test for advantageous behaviour from wall-following, relative to the Bayesian optimal solution, we wish to compute approximate average entropy changes over all possible physical worlds. To approximate this expectation we draw large numbers of samples of worlds. A simple way to model natural environments with the hex grid is to use the Ising [13] model,

$$P(\{w^i\}_{\forall i}) = \frac{1}{Z} \exp\left(\sum_i \pi(w^i) + \sum_{i,j:neigh(i,j)} \phi(w^i, w^j) \right), \tag{11}$$

with factors

$$\pi(w^i) = (\pi)^{w^i} (1-\pi)^{1-w^i} \tag{12}$$

$$\phi(w^i, w^j) = (\phi)^{w^i w^j}, \tag{13}$$

where $neigh(i,j)$ means that cells i and j are neighbours in the hexagon grid topology and Z is a normaliser. Fig. 2 shows sample worlds drawn from this model with various parameters. The model assigns probabilistic factors π to individual cells and ϕ to neighbouring pairs of cells. The effect of π is to increase the overall number of occupied cells. The effect of ϕ is to increase the contiguity of occupation, i.e. make it more probable that a cell is on if its neighbours are on. Intuitively this models the fact that physical 'stuff' in the world tends to actively clump together into large objects such as walls and furniture, but empty space does not, except as a result of the stuff clustering. By manual parameter search we found that $\pi = 0.05, \phi = 2$ gives similar environmental characteristics to our real robot mazes such as in [3]. Note that we use all positive factors rather than negative energy terms as in statistical physics[13]; the Boolean exponents in the factors act as on-off switches; and the pairwise factors equal one if it is not the case that both neighbours are occupied. To draw samples from eqn. 11 we use standard Gibbs sampling [13], with a burn-in of 100 samples initialised with independent cell draws from the π factors only.

To simplify the simulation further, we sample each decision instance in a new environment, rather than a previous partially mapped one. In a typical exploration of a *completely new* part of an environment, an agent will always have arrived from some direction, so have no (greedy) need to consider moving back in that direction. With proximal sensors, it will sense which cells around it are occupied. Fig 3 shows an example of the agent's mental world corresponding to the physical world of fig. 1, including its previous location that it has moved from. So for each decision, we sample a new world from the Ising occupancy model, place the agent at a random location, and consider the five possible moves it could take, excluding a randomly chosen simulated arrival direction. As we only consider one movement decision per world, we can use very small, 36-cell worlds as in the figures, and allow their edges to wrap around to simplify the mathematics.[2]

[2] This does however have an effect on the characteristics of sampled worlds such as island sizes, however we have chosen parameters to give realistic characteristics after the wrap around was included.

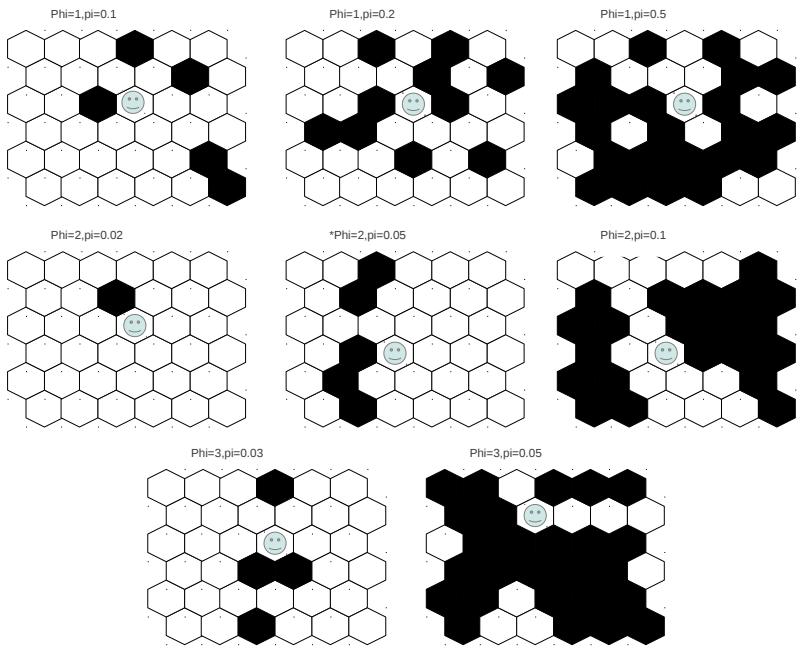


Fig. 2. Physical world samples with various parameters. Parameters around $\phi = 2, \pi = 0.05$ tend to generate worlds most like typical man-made environments, being quite sparse but locally correlated so as to produce structures resembling walls.

As we are investigating only the question of whether wall following is useful, the decisions that interest us are only those that occur when the agent begins next to at least one wall cell. If our sample world and random robot location do not meet this criterion then we discard the simulation and draw a new sample world.

At each movement decision, we consider moving to each neighbouring unoccupied cell. We compute (our best approximation to) the Bayesian expected entropy reduction given that move. We also test whether or not the move is a wall-following move (defined as a move in which the pre-move and post-move states share a neighbouring occupied cell). We store records of expected entropy reductions for every wall-following and non-wall-following move considered so that we may compare their averages to determine whether wall-following is significantly more useful than non-wall-following.

For each possible action, the ideal correct sensor values were obtained by consulting the physical world model. This is a further approximation to the expectation in eqn. 10, ideally we would use further samples from the agent's mental model here to integrate over possible physical worlds, but this approximation again speeds up the computation.

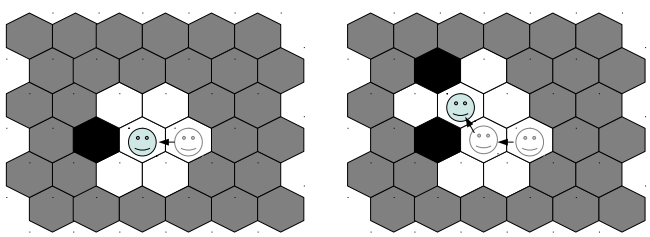


Fig. 3. Mental maps before and after a movement. Grey=uncertain. Outline robots show previously visited locations.

2.1 Approximation of entropies

For both the pre-move and post-move states we must compute eqn. 5. We make use of the standard independence assumption in the entropy sum, but we do want to retain the joint dependencies,

$$P(m_t|o_{1:t}) \neq \prod_i P(m_t^i|o_{1:t}). \tag{14}$$

This is because we are interested in how the correlations in the world induced by occupancy clumping in natural environments might affect the utility of the wall-following heuristics.

In order to compute eqn. 5 under these assumptions, we must therefore draw samples from the *joint* distribution of map cells, i.e. from the space of possible mental worlds given the observations. This can be done using a 'wake' [9] version of the Gibbs sampler used to generate physical worlds from eqn. 11, i.e. by clamping the observed cells to their observed values and Gibbs sampling from all remaining cells.

The pre-move and post-move mental map joint distributions were each approximated by drawing 100 wake samples from the unobserved cells, using 100 random Gibbs updates between each sample to approximate burn-in. The marginal entropy terms in eqn. 6 were then estimated from Good-Turing ('add-one') adjusted occupancy frequencies [13]. (It has been found empirically that correlated, short approximate burn-ins often work surprisingly well, e.g.. the use of single updates in contrastive divergence learning [9]. A full analysis of this assumption is beyond the scope of the present paper.)

3 Results

Setting $\phi = 2, \pi = 0.05$ which gives typical robot-arena-like environments, we ran the simulation for 100,000 world samples with about 3.5 possible actions per sample. The total computation time using these approximations was around 10 hours on a 3GHz Pentium Duo PC.

We obtained a sample mean expected entropy per cell reduction of 0.001933 for wall-following moves, and -0.001687 for non-wall-following moves.[3] The sample standard deviations were 0.0127 and 0.0122, and the numbers of observations were 49,241 and 64,039. To avoid any possible bias from simulating multiple actions from the same world sample, we will conservatively use N=10,000 rather than these counts. Assuming both populations standard deviations to be 0.015, a classical t-test gives a significant difference in favour of the wall-following moves. (t=46.7, N=10,000, df=10,000, $p \leq 0.001$).

Having demonstrated that this number of world samples lead to a significant result for our manual-parametrised world, we then ran a further 70 repeats of the experiments (700 hours computation time) for a range of (ϕ, π) parameters, to test how general the wall-following rule's usefulness is. The differences in expected entropy reduction means are plotted in fig. 4.

4 Discussion

Where do animals [10] and practical robots [6] benefit from using simple wall-following behaviour in place of full Bayesian utility maximation? The results show that for proximal sensing agents exploring previously unvisited regions in

[3] The negative entropy may seem strange but these are *per cell* statistics. The number of uncertain cells changes after a move when new cells are observed which may explain the apparent increase in uncertainty *per cell* rather than total map uncertainty. It is only the difference between the two means that is of interest.

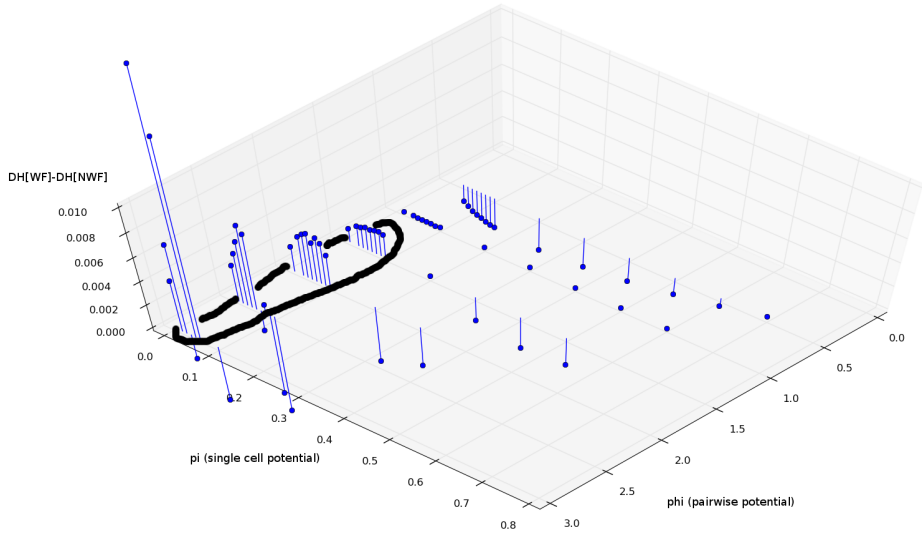


Fig. 4. Results. x and y axes are Ising parameter values. z axis is the sample mean difference between entropy change (DH) given wall following (WF) moves, and entropy change given non-wall-following (NWF) moves. The lasso shows the region of parameter space where wall following gives a significant advantage over non-wall-following.

a realistic environment class having sparse occupancy but strong local correlation, under simplifying assumptions, that wall-following actions do have significantly better outcomes than non-wall-following actions. This suggests that wall-following provides a fast approximation to optimal entropy exploratory behaviour in this type of environment.

Fig. 4 suggests that there is a 'sweet spot' in the parameter space where wall following works better than non-wall-following. This has been indicated by the hand-drawn lasso. It is interesting to see that the centre of this lasso corresponds roughly to the parameters $(\pi = 0.05, \phi = 2)$ that were originally chosen by hand to give real-environment-like characteristics. Fig. 2 shows samples of physical worlds with various parameters, corresponding to parameter coordinates inside and outside the lasso. The $(\pi = 0.05, \phi = 2)$ sample in the centre perhaps looks more like a typical room environment than any of the others in terms of density and connectivity. (Note that the hex grid wraps around, so correlations around the edges count like any others.)

Our simulations used the assumption that all exploration was in some 'new' area, i.e. that only one previously-visited cell is nearby, that it is a neighbour, and that we have just moved to the present location from it. This assumption is valid for agents just beginning to explore a new region of a map, but not for agents building up a detailed map over a longer time and revisiting areas. A pure wall-follower starting at the wall would never discover an object in the centre of a room for example. Revisiting also often occurs as an optimal action

in full Active SLAM systems, where the agent must re-localise from time to time as well as explore. Our simulations do not include tasks where particular locations must be visited (e.g. vacuuming a room, finding a power charger), rather their utility is only in mapping itself. Further work would be needed to test the utility of wall following in these more complex situations. An intriguing idea would be to test for similar effectiveness of other simple rules in these tasks, against full utilities, for example attempting to compare the Roomba's default combination of wall-following and random-straight-line wall-bouncing against an Active-SLAM approach to room coverage. The greedy active mapping utility used in our Bayesian utility maximisation is itself an approximation to non-greedy full planning, the latter would be expected to behave differently from the former in long-term exploration, but is so computationally hard as to be infeasible to simulate to significance even in our micro-world.

The hex-grid worlds used have the nice features that all actions are cleanly classified into wall-following or non-wall-following, and there are typically similar numbers of each that get sampled by our simulation. (49,241 and 64,039 in the main example above.) It is possible that other simple strategies could exist that are more selective that both wall-following and non-wall-following, and provide an even better approximation to optimal behaviour, though such strategies are beyond the scope of the present study.

In using the same (π, ϕ) parameters in both world generation and mental map sampling, we have implicitly assumed that the agent knows the Ising statistics of the class of environments it will be placed in. Such parameters could be learned from experience, but it would be interesting to investigate the effect of mismatches between the physical and mental parameters.

We have made use of large compute power sampling approximations in our results, and it is possible that for some simple worlds such as the Ising hex worlds, there exist analytic solutions for the expected entropy reduction. Further theoretical work could analyse the models used here to search for closed form solutions which may provide further insights into why the rule works. The 700 hour compute time used was necessary to obtain significance in the results (a 6×6 hex world has 2^{36} physical states, and far more mental states, to sample from), and gives some indication of how computationally hard optimal exploration can be, even for very simple worlds (cf. [15], which uses similarly tiny micro-worlds and approximations to compute entropy based explorations.)

While we have demonstrated a particular simple rule approximating optimal utility-based action selection, this work is also intended as an example of the more general case [5]. Can we analyse other well-known heuristics in this way? It is becoming clear that even with large supercomputers many decisions are intractable, and therefore a relevant research agenda for psychology and robotics is to seek out and try to understand useful rules that approximate such decisions. In what classes of environment are they valid and when do they break down? We could try to invent new rules from intuition or data-mining, or look at rules traditionally used by humans to find test candidates.

Bibliography

[1] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* Bradford Bks. MIT Press, 1986.

[2] L. Carlone, Jingjing Du, M.K. Ng, B. Bona, and M. Indri. An application of Kullback-Leibler divergence to active SLAM and exploration with particle filters. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 287 –293, oct. 2010.

[3] C.Fox, M.Evans, M.J.Pearson, and T.J.Prescott. Towards hierarchical blackboard mapping on a whiskered robot. *Robotics and Autonomous Systems*, 60:1356–1366, 2012.

[4] N.D. Daw, Y. Niv, and P. Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatum systems for behavioral control. *Nature Neuroscience*, 8:1704–1711, 2005.

[5] C. Fox. Formalising robotic ethical reasoning as decision heuristics. In *Proc. First UK Workshop on Robot Ethics, Sheffield*, 2013.

[6] C. W. Fox, M. H. Evans, M. J. Pearson, and T. J. Prescott. Tactile SLAM with a biomimetic whiskered robot. In *ICRA*, 2012.

[7] G. Gigerenzer, P.M. Todd, and ABC Group. *Simple Heuristics That Make Us Smart.* Oxford University Press, 2000.

[8] P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2003.

[9] G. E. Hinton, S. Osinderoi, and Y-W Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[10] G. Hosterrer and G. Thomas. Evaluation of enhanced thigmotaxis as a condition of impaired maze learning by rats with hippocampal lesions. *Journal of Comparative and Physiological Psychology*, 63(1):105–110, 1967.

[11] D. Kahneman. *Thinking Fast and Slow.* Macmillan, 2011.

[12] C. G. Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1–3):120–149, 1986.

[13] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[14] B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott. The emergence of action sequences from spatial attention: Insight from rodent-like robots. In *Proc. Biomimetic and Biohybrid Systems*, 2012.

[15] Z. Saigol. *Automated Planning for Hydrothermal Vent Prospeting using AUVS.* PhD thesis, University of Birmingham, UK, 2010.

[16] P. Strauss. Roombas make dazzling time-lapse light paintings. *http://technabob.com/blog/2009/09/29/roomba-time-lapse-art/*, URL.

[17] M. Taylor and C. Fox. Inventory management with dynamic bayesian network software systems. In *Proc. Int. Conf. Business Information Systems, Springer LNBIP*, 2011.

The Sheffield Wargames Corpus

Charles Fox[1], Yulan Liu[1], Erich Zwyssig[2,3], Thomas Hain[1]

[1]Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK
[2]Centre for Speech Technology Research, University of Edinburgh, Edinburgh, EH8 9AB, UK
[3]EADS IW, Appleton Tower, Edinburgh, EH8 9LE, UK

Abstract

Recognition of speech in natural environments is a challenging task, even more so if this involves conversations between several speakers. Work on meeting recognition has addressed some of the significant challenges, mostly targeting formal, business style meetings where people are mostly in a static position in a room. Only limited data is available that contains high quality near and far field data from real interactions between participants. In this paper we present a new corpus for research on speech recognition, speaker tracking and diarisation, based on recordings of native speakers of English playing a table-top wargame. The Sheffield Wargames Corpus comprises 7 hours of data from 10 recording sessions, obtained from 96 microphones, 3 video cameras and, most importantly, 3D location data provided by a sensor tracking system. The corpus represents a unique resource, that provides for the first time location tracks (1.3Hz) of speakers that are constantly moving and talking. The corpus is available for research purposes, and includes annotated development and evaluation test sets. Baseline results for close-talking and far field sets are included in this paper.

1. Introduction

Automatic Speech Recognition of clean speech has matured to provide practical commercial applications such as word processor dictation and phone voice menu selection. However recognising speech in natural environments such as business meetings and casual conversations is still an unsolved problem. A useful application of meeting recognition in conjunction with language processing methods would be the automated production of minutes for meetings. Meetings have several challenging properties for ASR: speakers are typically seated at several places around a table, microphones can generally only be placed at a few fixed locations away from the seating locations so recognition must be by farfield methods; speakers may speak simultaneously during heated discussions; and – most seriously for current beamforming methods – speakers may sometimes move around while speaking, such as walking to a whiteboard to present ideas to the group. The latter breaks the beamforming and source-separation assumptions of stationary sources used in many systems, and the development of algorithms to handle this case is an active research area. However there has been a lack of realistic data to train and test such algorithms, for two reasons. First, real business meetings have been unwilling to make public their contents as they often concern sensitive information. Meetings have been simulated by actors such as in the AMI corpus [3] but it is unknown how they compare to real meetings. Second, within available recorded meetings, the amount of data containing moving speakers is usually small, as the bulk of most meetings has the speakers seated and stationary. It is difficult to obtain large amounts of realistic data for moving speakers in meetings, despite the need for recognition of this type of data, which often contains the most valuable content of the meetings (a speaker standing to present is often more informative than one sitting during a general discussion).

To aid research in this area we have collected, and are making public as part of this publication, a 7 hour recording of the natural speech in a realistic environment where the speakers are almost constantly moving and often talking over one another. To avoid the above problems of meeting recordings we examined several surrogate scenarios with the requirement that the speech be from a real task (not reading text, not artificial tasks or by actors) which encourages movement, and by native English speakers. After examining several types of teamwork and game scenarios, we found that tabletop wargames provide an ideal surrogate. This paper describes the Sheffield Wargames Corpus (SWC) collected from 7 hours of tabletop gaming as a surrogate for mobile speakers during business meetings. We have pre-processed the corpus and defined evaluation sets and development sets for future users, and we present the initial baseline result using standard farfield ASR methods.

1.1. Related work

The AMI corpus [3] provides 100 hours of meeting recordings, a mixture of real and acted meetings, and includes basic speaker movement data for some parts. The non-acted meetings are for an artificial design task which does not yield the same level of interactivity as the wargames task; there is also no music in the background and particuants do not move around as much as when playing a tabletop game. AMI particpants often do not know each other and interact formally, in SWC, in several cases players have played together for up to 10 years and speak very casually and naturally, for example completing each others' sentences. The ICSI Meeting Corpus [8] provides natural meeting data from up to 12 participants wearing lapel microphones, and from four desk mounted microphones. The NIST RT09 challenge included meeting speech recognition sets from several sources (NIST,CMU etc.) but equivalently meetings were mostly not equivalent to the AMI style scenarios, athough some included board game playing.

The COSINE corpus [11] provides 145 hours (26.7 transcribed speech) of noisy indoor and output conversational fluent (though some non-native) English speech by groups of 2-7 paid volunteers equipped with head, shoulder, throat and chest-array microphones. Subjects were requested to have natural conversations with eath other and a list of topics provided. Unlike COSINE, the Sheffield Wargames Corpus does not require natural speech for its own sake, which may still be somewhat artificial in a recording envionment, but records natural speech of players in a real meeting-like task where the speech is essential for the task. Our subjects did not appear to be affected by the presence

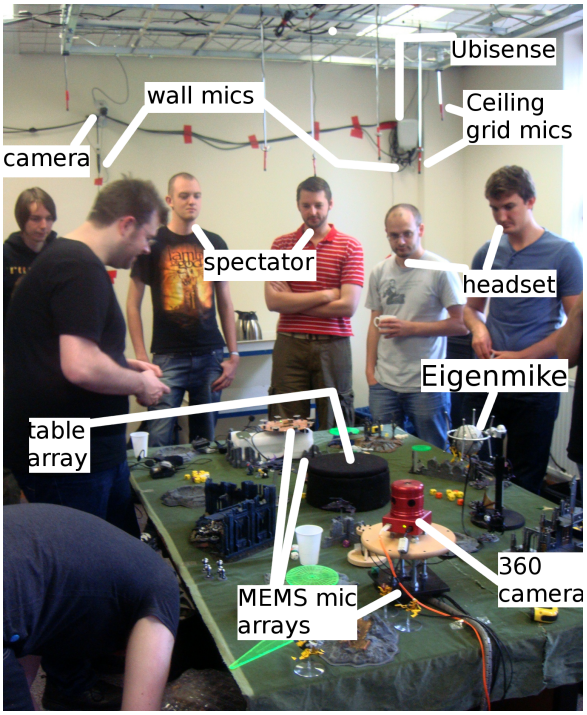


Figure 1: Wargame players during the recording. Mics can be seen hanging from the ceiling grid, on the back wall, on headsets, and in arrays on the table incorporated into the game scenery.

of recording equipment in any way, and many commented that they felt normal wearing their headsets. Many players were also online gamers who wear similar headsets online. COSINE does not include location data.

Corpora such as [10] and [9] exist containing natural speech in the specific noisy environment inside cars. Natural conversations over telephones are available in [6] and other corpora. The CHIME corpus [4] provides read speech in from a stationary location with natural speech and domestic background noise. The CAVA [2] corpus provides binaural recordings of moving speakers. None of these corpora however include location data or moving speakers relative to fixed microphones.

2. Corpus Description

2.1. Wargame Recording

The corpus recorded the speech of players of a tabletop game named Warhammer 40,000 [12]. In each game, four players (in two teams of two players) wear headsets to which the Ubisense[1] tracking tags adhere. Each game lasts for around 1.5 hours and comprises several (typically four) turns. Each turn consists of three formal phases in which different activities are carried out (moving, shooting and fighting with miniature soldiers). Team members must talk to one another; measure distances with a tape-measure throughout their actions to decide what to do; roll dice; and talk with the opposing team to agree on the effects of their actions upon them and vice versa. Fig. 1 shows a typical moment from a game. To maximise players' movement, the room was cleared of seating so players stand and walk around.

[1]Ubisense: `http://www.ubisense.net/en/about-us`

Game	Ses.	Dur.	#Utt.	Notes
1	1	38:00	441	–
	2	44:59	323	–
2	3	33:54	352	–
	4	34:49	352	–
	5	38:00	172	–
3	6	47:79	407	–
	7	32:10	237	–
	8	42:59	295	Dev set 3 pizza party
4	9	35:50	301	Background rock music.
	10	-	-	Excluded from corpus.
	11	41:49	404	Background light music.

Table 1: Wargame recording summary, showing game number, duration and number of utterances in each session.

They were also asked to customise the rules of their games to create as much motion as possible (e.g. banning fixed artillery but allowing fast vehicles which require players to physically move them long distances around the table.) At various times, additional spectators enter and leave the room without any head microphones, but they may also speak.

The whole corpus consists of 4 games in 10 release sessions (Table 1). The first 8 sessions of the first 3 games are mainly recorded without background noise, while the remaining sessions include background music (rock music and light music). In part of Session 8, Game 3, a 'party' was held by inviting all the day's players and some recording assistants to eat and chat while a game was in progress. A total of nine male native British English speakers participated in the recording. Some of the sessions include between-game speech such as setting up, packing away, and chatting. We originally recorded 11 hours of data but have cropped to 7 hours to remove recording problems such as incorrectly-worn head microphones and equipment failures.

There are 96 audio channels in total. Twenty-four sample-synchronous microphones recordings are composed of 4 groups: 4 headsets on the player, 4 microphones adhered to the wall, one circular microphone array of 8 on the table, one microphone array hanging on the grid to the ceiling (figure 2) in the shape of two overlapping squares. All distant microphones are hyper-cardioid AKG C417/III vocal condenser microphones, while the headsets are all Sennheiser ew100 wireless headsets of cardioid directivity. These 24 channels are recorded sample-synchronous using an all-Linux setup. 48kHz, 16 bit A/D conversion is by MOTO 8Pre's, linked by firewire 400 and FFADO drivers to JACK middleware on an Ubuntu Studio desktop PC, streaming audio data direct to hard disc. Full details of the sample-synchronous recording setup can be found in [5].

Additionally, but not sample-synchronised to the above channels, we made further recordings using an omnidirectional 32-channel Eigenmike© array, five 8-channel microphone arrays using analogue and digital MEMS microphones and the PointGrey Ladybug2 camera. A detailed description of this setup is available from the homepage of the 2012_MMA corpus [2].

There are two cameras hanging overhead (figure 3) and a 360 degree panoramic camera on the table recording simultaneously with the microphones. Four Ubisense 3D location sensors are installed at corners of the ceiling to collect signals from

[2]`http://www.cstr.inf.ed.ac.uk/research/#corpora`

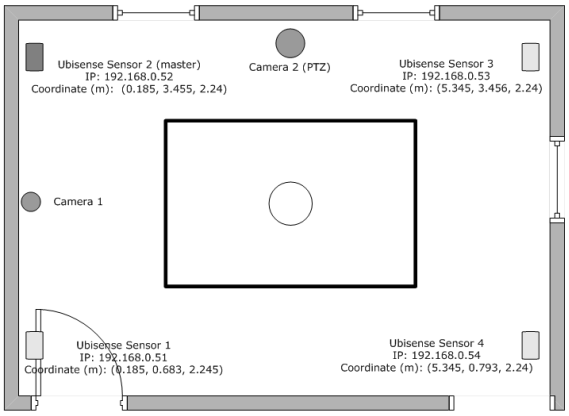


Figure 2: Audio recording configuration

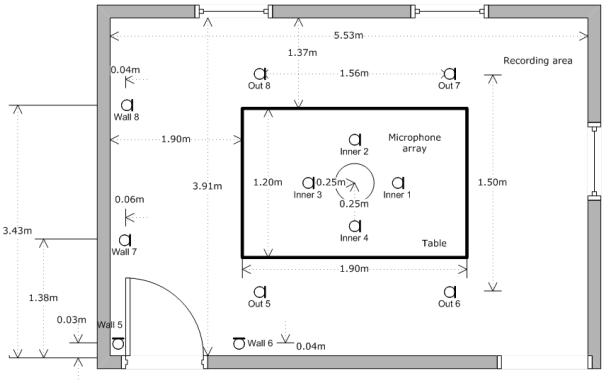


Figure 3: Video and location tracking system configuration

Ubisense tags attached to players' headsets, i.e. to track their head locations.

The Eigenmike and MEMS mics, video and Ubisense were synchronised manually to the 24 sample-synchronous audio channels by aligning a Ubisense-tracked clapper-board's clap at the start of each session.

2.2. Ubisense Location Tracking

In the public release data we provide both the raw data from the Ubisense location tracking system and a smoothed version. The manufacturer's stated precision of the raw recordings is 15cm and the effective location update rate is about 1.3Hz. To compensate for location noise, simple exponential smoothing is performed by selecting a exponential factor to compromise between smoothing error and the smoothness with the *Least Normalised Distance* (LND) method, where the *surge*, (third order derivative) of the smoothed data is used as a metric of smoothness. The curve between the smoothing error and the surge has a cross point with each coordinate axis respectively, then both axis are scaled and normalised so that the cross point is unit 1. The corresponding exponential smoothing factor to the point with least distance from the origin on the curve is selected to smooth the raw location data (Figure 4).

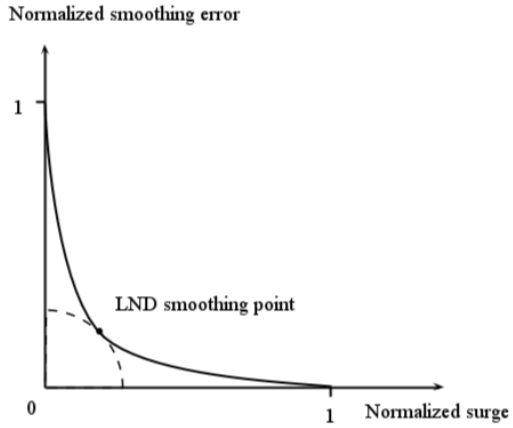


Figure 4: Exponential smoothing factor determination with LND method

2.3. Pre-processing and Set Selection

In each session, 8 and 7 minutes of data are selected for the evaluation and development sets respectively. The evaluation set is selected from the middle of a session when players are talking most seriously and when there are only few artefacts in the recording channel. Development sets are selected before and after the evaluation sets similarly by avoiding heavy laughs, breaths or recording artefacts as much as possible.

Segmentation, annotation and transcription were performed with *XTrans*[3]. For the evaluation sets, manual segmentation, annotation and transcription are generated in a first round of annotation based on headset recordings. Then the transcription is revised by native English speakers and the format is corrected to be compatible with rules as used in the AMI corpus[4] and published in NIST STM format. Currently the development sets have been manually segmented and annotated based on headset recording. We also publish meta-data comprising subjects' English nativeness and accent region, gender, age and height.

85% of utterances overlap with at least one other (1.11 hours of 1.35hours) in the transcribed evaluation set. The perplexity of the evaluation set against a standard AMI 3-gram language model [7] was 167, with an OOV rate of 1.3% (=203 of 15430 words). The reason for the relatively high perplexity and low OOV seems to be that many entities in the game have known English words as names but they are used in unusual ways, for example soldiers called 'space wolves'. There are 3255 utterances in the test set, 26% contain a LAUGH or BREATH token (7% LAUGH, 18% BREATH).

2.4. Segment Tagging

Based on raw data from the Ubisense tracking system, the evaluation data was grouped according to the location and the moving speed. Four spatial criteria are used and all are based on segmental average of location, distance and moving speed. This is to enable standard comparisons of algorithms such as beamformers whose performance may depend on location. *Spatial Splitting Circular* (SSC) is based on the segmental average distance from the speaker active in that headset channel segment to

[3]LDC tools XTrans: `http://www.ldc.upenn.edu/tools/XTrans`

[4]AMI transcription: `http://corpus.amiproject.org/documentations/transcription`

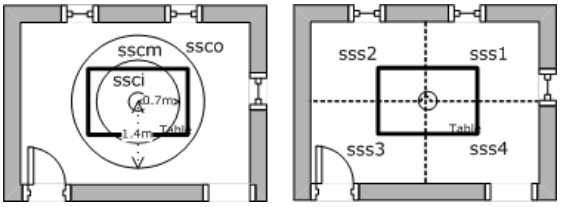


Figure 5: Video and location tracking system configuration

total	ssci	sscm	ssco
2548	281	1660	607
–	11.03 %	65.16 %	23.82 %

Table 2: Statistics of segment tagging: SSC

the centre of the circular microphone array on the table in horizontal plane (figure 5, left). If the distance is less than 0.7m, it's tagged as "ssci" where "i" stands for "inner circle". If the distance is larger than 1.4m, it's tagged as "ssco" where "o" stands for "outside circle". Otherwise the segment is tagged as "sscm" where "m" stands for "middle circle". Table 2 shows the statistics of the splitting based on the first criterion. Note that the segments with only breath and laugh are excluded here. *Spatial Splitting Square* (SSS) is based on the segmental average location in the room. The room is split into four square parts equally in horizontal plane. As shown in figure 5 (right), the four subareas are named respectively "sss1", "sss2", "sss3" and "sss4", in the same order of mathematical coordinate quartiles. Table 3 gives the statistic details in this case. *Spatial Splitting Height* (SSH) is based on the segmental average vertical height. According to the height of the speakers and the relative position of tags to human head, if the segmental average height is smaller than 1.476m, the segment is tagged with "sshl" where "l" stands for "low" since there is a high possibility the person leans down in that segment. Otherwise it's tagged with "sshn" where "n" stands for "normal". Table 4 gives the statistic details in this case. *SPeeD splitting* (SPD) is based on the segmental average moving speed in horizontal plane. If it's smaller than 0.2m/s, the segment is tagged with "spds" where "s" stands for "slow". Otherwise the segment is tagged with "spdn" where "n" stands for "normal". Table 5 gives the statistic details in this case.

In addition to spatial tags, we also provide tags indicating the session number (which contains information about background music) and whether transcribed utterances overlap.

3. Baseline speech recognition results

For the head microphones and a single distant microphone (SDM, grid channel 0), recognition was performed on individual channels. Acoustic models and language models trained on the AMI corpus [3] and from the AMI RT'09 meeting transcription system [7] were used in the experiments. For the latter both individual head microphone (IHM) data and multiple distance microphone (MDM) models are available, used in the first

total	sss1	sss2	sss3	sss4
2548	512	673	754	609
–	20.09 %	26.41 %	29.59 %	23.90 %

Table 3: Statistics of segment tagging: SSS

total	sshl	sshn
2548	848	1700
–	33.28 %	66.72 %

Table 4: Statistics of segment tagging: SSH

total	spds	spdn
2548	1325	1223
–	52.00 %	48.00 %

Table 5: Statistics of segment tagging: SPD

passes of the RT'09 systems for recognition of IHM and MDM conditions. For the sample-synchronous table-mics-only (table) and table+grid+all (tgw) sets we applied the Beamformit beamformer (BF) [1] with default parameters and no noise reduction pre-processing, to extract a continuous single channel. We used a variety of standard decoders and input features [13] to report a range of results.

Scoring was performed with NIST tools although for the far field conditions prior knowledge about the word to channel association was assumed. The results are shown in table 6. We found no significant differences in performance between the music and non-music sessions, and between overlappiing and non-overlapping speech, for the head-mic baselines. For the SDM case only we performed a run on a subset of the data tagged as non-overlapping speech (noOverlap).

4. Discussion

Speech recognition research is moving to more natural situations, but the community has lacked a natural, distant, native English speech corpus of highly mobile speakers with location data and we are publishing the Sheffield Wargames Corpus to fill this need. Our baselines show this is a highly challenging ASR task – harder than previous location corpora using actors – even with state of the art adapted CMVN/HLDA/CMLLR systems with well trained acoustic and language models achieving only 65.3 WER from the head mics and 85.8 from a basic default beamformer. These baselines will provide a challenge both for our and others' research on natural distant speech and the publication of 96 audio channels and location data should aid development of more powerful natural speech technology.

5. Acknowledgements

Thanks to Sheffield University Wargames Society for their great support, planning and cooperation; Ghada Al Harbi, Davide Marino, Sarah Al Shareef, Raymond Ng and Madina Hasan for recording help. Funded by EPSRC Natural Speech Technology Programme Grant EP/I031022/1 and EADS.

Data	Models	P	WER	S	D	I
IHM	AMI	1	73.7	49.0	18.3	6.4
IHM	AMI	2	70.7	43.6	23.5	3.6
IHM	RT'09 IHM	2	65.3	41.5	19.3	4.5
SDM-o1	RT09 MDM	2	87.3	44.2	40.7	2.4
table-o1	RT09 MDM	2	86.8	40.6	43.8	2.4
gtw-o1	RT09 MDM	2	85.7	39.8	43.0	2.9

Table 6: Baseline word error rate results on the SWC evaluation test set. P = number of passes.

6. References

[1] Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech and Language Processing*, (7):2011–2023, 2007.

[2] Elise Arnaud, Heidi Christensen, Yan-Chen Lu, Jon Barker, Vasil Khalidov, Miles Hansard, Bertrand Holveck, Hervé Mathieu, Ramya Narasimha, Elise Taillant, Florence Forbes, and Radu P. Horaud. The CAVA corpus: synchronised stereoscopic and binaural datasets with head movements. In *ACM/IEEE International Conference on Multimodal Interfaces (ICMI'08)*, October 2008.

[3] J.C. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction, Second International Workshop, 11-13 July 2005, Edinburgh, UK. pp. 28-39. Lecture Notes in Computer Science 3869. Springer Verlag*, 2006.

[4] Heidi Christensen, Jon Barker, Ning Ma, and Phil Green. The CHiME corpus: a resource and a challenge for computational hearing in multisource environments. In *Proc. Interspeech*, 2010.

[5] C. Fox, H. Christensen, and T. Hain. Linux audio for multi-speaker natural speech technology. In *Proc. Linux Audio Conference*, 2012.

[6] J. Godfrey, E. Holliman, and J. McDaniel. SWITCHBOARD: telephone speech corpus for research and development. In *Proc. ICASSP*, 1992.

[7] Thomas Hain, Lukas Burget, John Dines, Philip N Garner, Asmaa el Hannani, Marijn Huijbregts, Martin Karafiat, Mike Lincoln, and Vincent Wan. The AMIDA 2009 Meeting Transcription System. In *Interspeech'10*, pages 358–361, 2010.

[8] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proc. ICASSP*, 2003.

[9] N. Kawaguchi, S. Matsubara, H. Iwa, S. Kajita, K. Takeda, F. Itakura, and Y. Inagak. Construction of speech corpus in moving car environment. In *Proc. ICASSP*, 2000.

[10] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. Huang. Avicar: audio-visual speech corpus in a car environment. In *Proc. ICSLP*, 2004.

[11] Alex Stupakov, Evan Hanusa, Jeff Bilmes, and Dieter Fox. Cosine - a corpus of multi-party conversational speech in noisy environments. In *Proc. ICASSP*, 2009.

[12] Games Workshop. *Warhammer 40,000 Rulebook, 5th Ed.* Games Workshop Ltd. ISBN 978-1841548753, 2008.

[13] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.

frontiers in
NEUROROBOTICS

ORIGINAL RESEARCH ARTICLE
published: 02 January 2013
doi: 10.3389/fnbot.2012.00012

The effect of whisker movement on radial distance estimation: a case study in comparative robotics

Mathew H. Evans[1]*, Charles W. Fox[1], Nathan F. Lepora[1], Martin J. Pearson[2], J. Charles Sullivan[2] and Tony J. Prescott[1]

[1] Department of Psychology, Sheffield Centre for Robotics, University of Sheffield, Sheffield, UK
[2] Bristol Robotics Laboratory, University of the West of England, Bristol, UK

Edited by:
Blythe Towal, California Institute of Technology, USA

Reviewed by:
Karen Moxon, Drexel University, USA
Joseph H. Solomon, Northwestern University, USA

***Correspondence:**
Mathew H. Evans, Department of Psychology, University of Sheffield, Western Bank, Sheffield, S10 2TN, South Yorkshire, UK.
e-mail: mat.evans@shef.ac.uk

Whisker movement has been shown to be under active control in certain specialist animals such as rats and mice. Though this whisker movement is well characterized, the role and effect of this movement on subsequent sensing is poorly understood. One method for investigating this phenomena is to generate artificial whisker deflections with robotic hardware under different movement conditions. A limitation of this approach is that assumptions must be made in the design of any artificial whisker actuators, which will impose certain restrictions on the whisker-object interaction. In this paper we present three robotic whisker platforms, each with different mechanical whisker properties and actuation mechanisms. A feature-based classifier is used to simultaneously discriminate radial distance to contact and contact speed for the first time. We show that whisker-object contact speed predictably affects deflection magnitudes, invariant of whisker material or whisker movement trajectory. We propose that rodent whisker control allows the animal to improve sensing accuracy by regulating contact speed induced touch-to-touch variability.

Keywords: active sensing, touch, whisker, robot, biomimetic, comparative, perception, classification

1. INTRODUCTION

Many robots have been developed for understanding whisker sensing (Prescott et al., 2009; Evans et al., 2012). Though each has expanded our understanding in certain ways, it is difficult to apply the results in a general way to other paradigms or wider applications. Choices made in the development of robotic hardware specify the kind of questions that can be answered by that platform. For example, robots with high degrees of freedom allow research into the effects of whisker movement on sensing, but this movement is not precise enough to expose the underlying mechanisms of whisker sensing.

A complementary robotics approach, similar to approaches used in the biological sciences (Kappers et al., 1936; Kardong, 2006), allows the development of robots where the results from one platform can inform the experiments on another. Great progress can be made, both by performing experiments on appropriate platforms and ensuring that results inform general conclusions. Results from robotics in turn may provide insights for neuroscience. A key example where such an approach may be fruitful is in understanding the effect whisker movement has on sensing.

In this paper we will briefly introduce rodent whisker movement control, and whiskered robots that model these systems will be reviewed. A comparative robotics approach is described, outlining a path for addressing some of the questions from biology in a more explicit and effective manner. Specifically, what is the effect of whisker movement on radial distance estimation?

1.1. ACTIVE WHISKER TOUCH SENSING IN RODENTS

Whiskers are found in almost all terrestrial mammals, *Homo Sapiens* excepted, and some marine mammals (Ahl, 1986). Although whiskers are hairs, their structure is highly specialized, with regards to their surface structure and mechanical properties, in transferring contact information to the hair follicle for tactile sensing (Chernova and Kulikov, 2011). For example, whiskers vary in length, thickness, shape, and stiffness between species depending on animal size or how the whiskers are used (Sarko et al., 2011).

Rats typically have around 30 prominent whiskers on each cheek (or mystacial pad), arranged in a regular grid of rows and columns (Ahl, 1986). These large *macro*-vibrissae vary in length and width across the whisker pad, from the largest [2–40 mm in length (Diamond et al., 2008)] in the most caudal column down to the smallest in the rostral column. A dense array of 40–70 smaller *micro*-vibrissae (a few mm in length) are located around the lips (Brecht et al., 1997). Physical differences between the whiskers affect their mechanical properties, such as their bending and damping characteristics (Hartmann et al., 2003), which could have repercussions for sensing, a critical consideration when building artificial whiskers.

Whiskers can only encode information about objects when they make contact with them. To gather information about the world, rodents sweep their whiskers through the air, and bring them on to surfaces in the environment. This back and forth sweeping movement of the whiskers [called "whisking" (Welker, 1964)] has been the subject of a great deal of research. A single

"whisk" is defined as one cycle of whisker protraction (forward movement) and retraction (backward movement), and without perturbation rats typically whisk in short bouts of ≈10 cycles, at around 5–8 Hz (Carvell and Simons, 1990).

Though initially thought to be very regular (Semba and Komisaruk, 1984), recent studies using optoelectronic monitoring techniques (Bermejo et al., 2002) and high speed videography (Sachdev et al., 2002; Towal and Hartmann, 2006) has revealed that rat whisking can be highly irregular and complex. It is full of asynchronies, where different whiskers are protracted by different amounts (Sachdev et al., 2002) and asymmetries, where the whiskers on either side of the head are moved out of phase with one another (Towal and Hartmann, 2006). These irregular movements are thought to be the result of active sensing strategies (Hartmann, 2001; Berg and Kleinfeld, 2003; Mitchinson et al., 2007).

Among other parameters (Towal and Hartmann, 2008), rats control the spread and contact force of whiskers to ensure even, light contacts across the whisker array. Specifically, rats seem to use particular strategies for sensing, such as the rapid cessation of protraction upon initial contact with a surface, and contact induced asymmetry in the whisker movements, where a whisker contact on one side of the rat's head causes an increase in the protraction of whiskers on the side contra-lateral to contact (Grant et al., 2009). Together these efforts are grouped into a strategy described as minimal impingement (hereafter MI), maximal contact (Mitchinson et al., 2007).

In addition, head movement greatly effects the velocity of whisker contacts (Grant et al., 2009), and whisker movement is controlled to sweep space in anticipation of head movement (Towal and Hartmann, 2006). Though some have been identified (Grant et al., 2009), it remains unclear which components of whisker movement are actively controlled by the rat, which are artefacts arising from limitations of biological systems, and which if any are important for sensing. For example, do rats change their whisking frequency to improve the discrimination of particular surfaces, or because their muscles cannot maintain high frequencies of whisking for prolonged periods of time?

1.2. ACTIVE WHISKER TOUCH SENSING IN ROBOTS

A number of software and hardware models have been developed to better understand whisker sensing. There are many reasons why modeling a system is an important step toward understanding, and why synthetic models (models built in software or hardware) in particular are so useful (Rosenblueth and Wiener, 1945; Mitchinson et al., 2010). For example, in a model whisker movement can be precisely controlled to determine the effects any changes have on whisker deflections and subsequent analysis.

Whiskers have been modeled simply as elastic beams (Salisbury, 1984; Young et al., 2003). Though progress has been made very recently in more precise computational modeling of whiskers (Quist and Hartmann, 2012), their small size make accurate simulation difficult. A more straightforward method is to build artificial whiskers and mount them on robots. Whiskered robots have been broadly reviewed recently in Prescott et al. (2009). Specifically focusing on whisker actuation mechanisms, and the effect these have on sensing, early models were static and provided binary contact vs. no contact reports (Schiebel et al., 1986; Jung and Zelinsky, 1996). Hinged whiskers were used to infer the location of tip contact through potentiometer readings (Russell, 1992). Emulating earlier modeling work, elastic beam equations have been used by a number of researchers to infer the location of contact along an artificial whisker, and in turn the curvature of a surface with whiskers mounted on robots (Russell and Wijaya, 2003, 2005), rotational DC motors (Kim and Moller, 2004, 2007), or a set screw (Solomon and Hartmann, 2006). In a unique design Wilson and Chen (1995) used a pair of pressurized tubes laid end to end as a whisking mechanism, and a closed loop control system to infer whisker tip contact location in space.

In more biomimetic (Vincent et al., 2006) robots (such as in, Seth et al., 2004; Fend et al., 2005; Meyer et al., 2005; Pearson et al., 2007; Lepora et al., 2012b) multiple degrees of freedom are included as whiskers are often actuated, as well as being mounted on mobile robot platforms. This increased whisker movement makes texture discrimination difficult (Fend et al., 2003), especially in conditions where whisker motion varies from trial to trial (Fox et al., 2009). To address this point further more complex whiskered robots, with individually actuated whiskers have been developed in recent years for investigating biomimetic whisker control strategies (such as MI discussed earlier, Pearson et al., 2011), and how these strategies may improve texture discrimination (Lepora et al., 2010b; Sullivan et al., 2012).

1.3. WHISKER MATERIALS

The material a whisker is made from has a critical influence on the way a whisker interacts with a surface, and as a result the nature of the deflections created at the whisker base (Hartmann et al., 2003). Whiskers are specialized sensory elements for aiding tactile sensing in the hair follicle, differing in structure from other mammalian hairs to ensure strength and stiffness (Chernova and Kulikov, 2011; Sarko et al., 2011). Rat whiskers have evolved to have excellent mechanical properties for transferring tactile information to the follicle during sensory exploration (Chernova and Kulikov, 2011). Specifically, rat whiskers are stiff when moved in air but bend in contact and are highly damped with damping ratios ζ of 0.11:0.19 and Young's modulus E of ≈3–4 GPa (Hartmann et al., 2003). This ensures that the whiskers do not oscillate when whisked in air, which can add noise to the deflection signal and make contacts difficult to detect. This damping also increases when the whisker is in the animal, as observed in that contact induced oscillations are smaller in whisking rats than isolated whiskers (Hartmann et al., 2003). Whiskers are tapered, which has certain advantages [some are described in detail by Williams and Kramer (2010)], so it is important that artificial whiskers taper if they are to appropriately mimic the biological system. In artificial systems whisker material and morphology have also been shown to be important for texture discrimination (Lungarella et al., 2002; Fend et al., 2006).

1.4. RADIAL DISTANCE TO CONTACT ESTIMATION

Estimating the radial distance to contact (in this paper, from the base of the whisker) allows an agent to determine whether an object has made contact with a whisker at the tip or the shaft, which is important for texture discrimination, and to

discriminate between contacts with the surfaces or corners of objects. Measuring the location in space of multiple contacts over time allows an agent to reconstruct the contours of an object or perimeters of the environment (Fox et al., 2012). Radial distance estimation has been demonstrated in rodents (Szwed et al., 2006), and approached by many researchers. Theoretically, radial distance to contact estimation along a beam is a solved problem (Solomon and Hartmann, 2006, 2011), as long as whisker movement is precisely controlled and the physical properties such as size, taper, and elasticity are known. In an applied robotic setting these parameters are not always known precisely, therefore a more data-driven approach is appropriate (Evans et al., 2010a; Lepora et al., 2010a). In this paper we use a feature-based radial distance estimation method, essentially extracting analogous information to the bending moment at peak protraction, but using regression to determine the relationship between this value and radial distance to contact.

Feature-based classification involves finding invariant features in the data that correspond to parameters in the real world. For example, using scale invariant feature transformation (SIFT) algorithms in vision (Lowe, 1999; Juan and Gwun, 2010). Feature extraction has also been demonstrated in biological sensing systems. Frog prey capture is based on the principle of feature detection, with responses elicited for any object matching the size and angular velocity of a fly (Lettvin et al., 1959). In the rat whisker system some researchers have reported cells that respond to "kinetic features" in whisker deflections (Petersen et al., 2008). An advantage of this approach is that it reveals how different whisker movement patterns affect the extracted features, and may allow the measurement of numerous features to classify a range of whisker-object contact parameters simultaneously in future.

1.5. A COMPARATIVE ROBOTICS APPROACH

As robots become more complex, they become more difficult to control. As this progression continues it may be difficult to conduct experiments that address fundamental questions about whisker-object interactions. In this paper we present a complementary robotics approach. Here, a group of different robots are used to address the same problem of radial distance to contact estimation, allowing a direct comparison of whisker materials and actuation methods. This approach may help in understanding radial distance to contact estimation more generally, invariant of whisker material or actuation method.

Three robots were used for comparison. Firstly, an XY positioning robot moves objects onto an artificial whisker sensor in an accurate and highly repeatable manner, allowing the collection of large amounts of whisker deflection data. This approach provides the opportunity for a better understanding of the nature of whisker-object contacts (expanding on previous preliminary work in Evans et al., 2010a,b). Data collected on the XY positioning robot is used to systematically train and test a classifier under a range of contact conditions, and extract features for radial distance to contact and contact speed estimation. Secondly, SCRATCHbot (an acronym of Spatial Cognition and Representation through Active TouCH, Pearson et al., 2010) is a mobile whiskered robot which approximates the degrees of movement of an exploring rat. Actuated whiskers are mounted on an articulated "neck," which is in turn fixed to a mobile base. SCRATCHbot is used here in a "head-fixed" protocol to show how the classifier and features developed on the XY positioning robot can be applied to data from a less restricted whisking robot. Thirdly, CrunchBot is a mobile whiskered robot with stationary whiskers (Fox et al., 2011). This robot has fewer degrees of freedom than SCRATCHbot, allowing more straightforward robot control and data collection in a mobile setting. Testing the feature-based classifier on CrunchBot evaluates whether this approach can be robustly implemented on a mobile robot. In addition to separate actuation methods, all three robot platforms utilize different whisker materials. This change allows for the comparison of whisker materials in the same radial distance estimation task, and tests the classifier's robustness to this change. A great deal of effort was made to find materials for artificial whiskers that would match the mechanical properties of real whiskers at different scales, such as stiffness and bending characteristics. Additional concerns are toughness and the ability to appropriately shape the whisker.

2. MATERIALS AND METHODS

2.1. ROBOT PLATFORMS

2.1.1. XY positioning robot platform

A Cartesian robot (see **Figure 1**) was chosen as it is capable of a wide range of movement, is very accurate and can move at speeds which approximate scaled rat-whisk velocities. Deflections for the whisker are streamed to a PC, and can be processed in real time to control subsequent movement of the positioning robot. The robot (Yamaha-PXYX, Yamaha Robotics) has a movement range of 350 × 650 mm, and can move up to 720 mm/s. Repeatability of the robot is ±0.01 mm, and the maximum load it can carry is 1.5 kg. Objects are carried by the robot into an artificial whisker fixed to the table, as this allows us to control the contact as carefully as possible. Moving the whisker into an object would subject the sensor to trajectory-dependent accelerations that would cause more complex effects such as whisker oscillations. Subsequent robots described in this paper allow for exploring these trial to trail variations and their effect on sensing. A controller (Yamaha RCX 222, 2-axis robot controller) takes instructions from a PC through an RS232 cable, and the controller interprets the instructions, completes path integration, and drives the motors. Instructions for the robot are generated inside a MATLAB (mathworks.com) loop, and can be easily updated during robot operation, depending on the whisker input.

2.1.2. SCRATCHbot robot platform

The SCRATCHbot robot platform (**Figure 2A**) consists of a head-mounted whisker array, a mobile body housing computing means, motors and power supply, and an articulated neck allowing free movement of the head independent from the body. For this experiment we focus only on the head. Six independent columns of three whiskers, arranged in two arrays of nine whiskers either side of the head, are independently driven by a DC motor and gearbox. Whiskers in a column are mechanically coupled, but columns themselves are capable of independent rotational (anterior–posterior) whisk-like movement. Movements and data collection are coordinated by independent

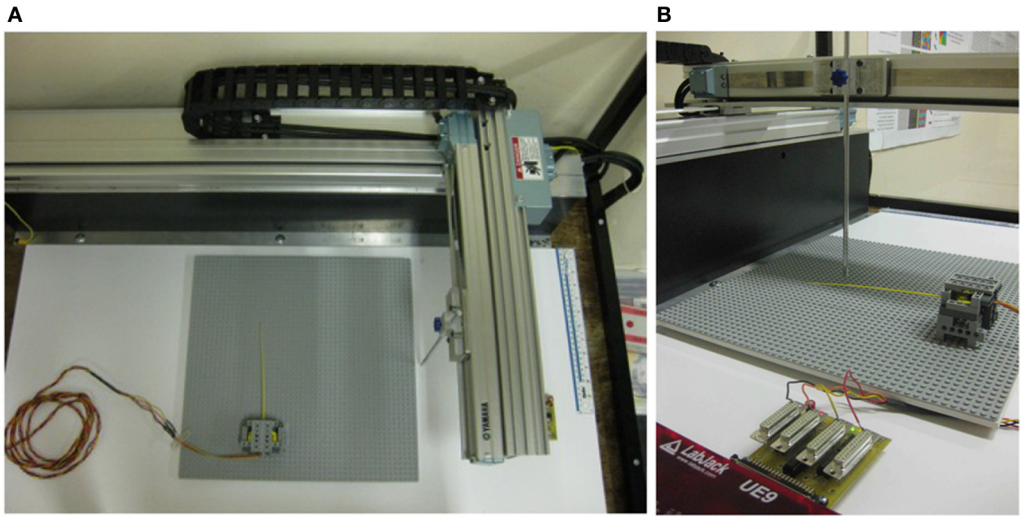


FIGURE 1 | The XY positioning robot. (A) From above, to show the range of movement available. **(B)** From the side. A narrow, rigid aluminium bar was moved into the whisker perpendicularly, from a clockwise or anticlockwise direction.

micro-controllers. A central PC-104+ reconfigurable computing platform, including a closely coupled array of FPGAs and a single board computer, handled all sensor and motor coordination.

2.1.3. CrunchBot robot platform

CrunchBot (**Figure 2B**) consists of an iRobot Create base (irobot.com) with an extended cargo bay to accommodate a netbook PC. This netbook is used for autonomous control of the robot, running Ubuntu 10.10 on a single-core Intel Atom processor. The netbook hosts a Player server (playerstage.sourceforge.net) providing high-level, networked API interfacing to the Create's serial port commands. Rapid prototyped ball joint mountings fixed to an adjustable metal bar individually hold six static artificial whiskers. The whiskers are positioned at angles to fan out across the width of the robot while covering any blind spots. Radial distance estimation and basic motor control can run in real time on the netbook, reading the raw data from the circular buffer.

2.2. ARTIFICIAL WHISKERS

Three different materials were used in the fabrication of whiskers for the three robot platforms. Each whisker is made on an Envisiontec Perfactory rapid prototyping machine (envisiontec.de). The XY positioning robot whisker was made from flexible Acrylonitrile butadiene styrene (ABS) plastic ($E \approx 1.63$ GPa, $\zeta \approx 0.07$), 185 mm long, 2 mm diameter at the base, 0.5 mm at the tip. SCRATCHbot whiskers are identical in shape and size to those on the XY positioning robot, but are made from the fiberglass material ($E \approx 25$ GPa, $\zeta \approx 0.5$). CrunchBot whiskers were made from Nanocure RC25 ($E \approx 4.89$ GPa, $\zeta \approx 0.2$) and were smaller in size, 160 mm in length, 1.45 mm diameter at the base tapering linearly to 0.3 mm at the tip. All whiskers are straight, but may curve slightly due to gravity perpendicular to the plane of movement, and are linearly tapered. Each whisker was mounted at the base into a short, polyurethane rubber (Poly 74-20 RTV from Polytec, synergyrm.co.il) filled, inflexible tube called a follicle case (see **Figure 3**).

A magnet was bonded to the base of the whisker shaft in such a way that when the follicle case/whisker shaft assembly was located into the whisker mount (see **Figure 3**), the magnet was positioned directly above a tri-axis Hall effect sensor integrated circuit (IC, Melexis MLX90333 www.melexis.com). Hall effect sensors measure the change in voltage across a conductor in response to changes in the strength of a nearby magnetic field. The tri-axis Hall effect sensor used here can measure the voltage changes in three orthogonal axes, i.e., x and y across the plane of the sensor, and z upwards toward the whisker. As forces are applied to the whisker shaft, the moment experienced at the base will rotate the magnet around a pivot point, nominally in the center of the polyurethane bearing. The sensor output voltage provides information about the magnitude of whisker deflection whether the whisker is moving or not, therefore the information is useful for static as well as dynamic classification approaches. When the whisker is deflected the movement of the magnet is proportional to whisker bending.

2.3. DATA COLLECTION

2.3.1. XY positioning robot data collection

Deflections of the whisker were transmitted through the Hall effect sensors to a LabJack UE9 USB data acquisition card (labjack.com) at a rate of 1 kHz for each of the x and y directions. Each trial lasted 4 s. This data was sent to a computer through the BRAHMS middleware (brahms.sourceforge.net) for analysis in MATLAB.

A MI control policy (observed in rats and discussed in Section 1.1) was implemented. In contrast to passive deflections, this policy keeps the amplitude and duration of whisker deflection within a limited range, and also keeps whisker ringing after contact to a minimum. An additional benefit is that the forces acting

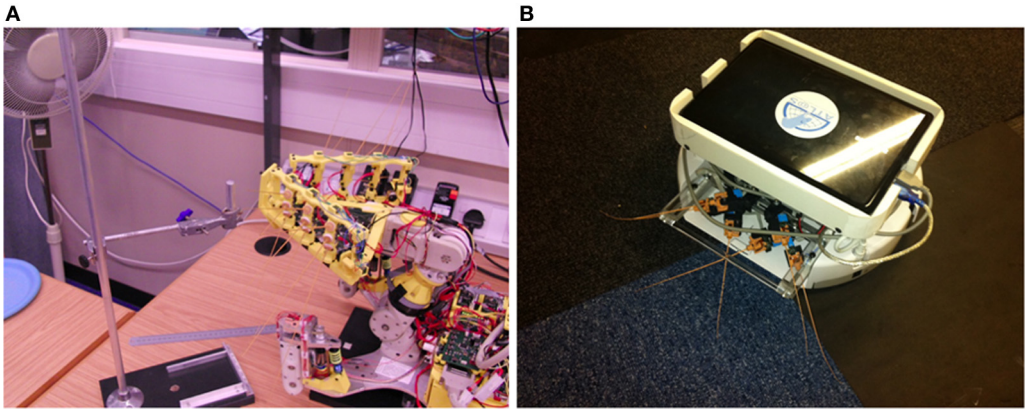


FIGURE 2 | (A) The SCRATCHbot whiskered mobile robot. To collect data for this experiment the robot platform was kept stationary while it whisked into a pole at varying radial distances to contact, and whisk speed. **(B)** The CrunchBot mobile whiskered robot.

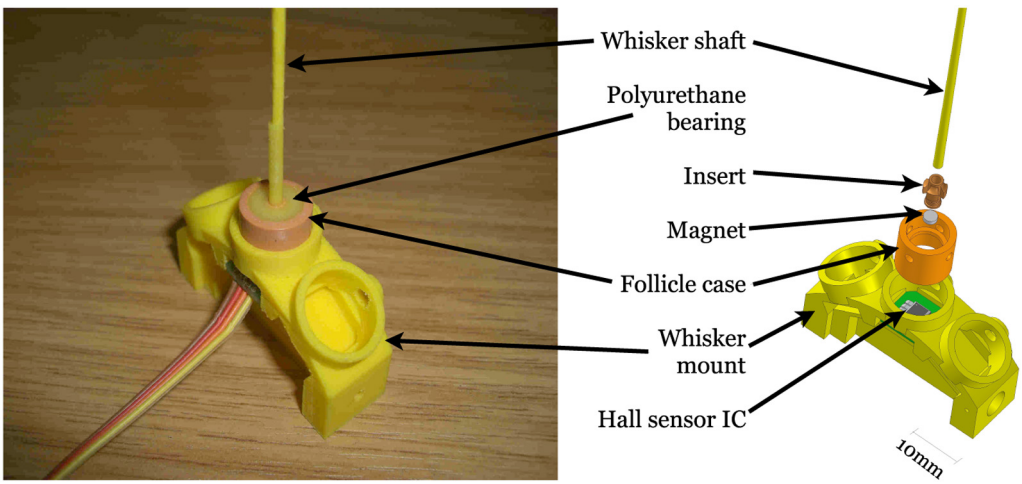


FIGURE 3 | Diagram of whisker follicle sensor construction. CrunchBot whisker follicles differ slightly in shape, but operate in exactly the same way.

on the whisker are much smaller, meaning whisker breakage is less likely, even in high speed collisions.

MI was implemented by instructing the robot to move an object (here a narrow, rigid, cylindrical bar) into the whisker at a given speed until a deflection magnitude threshold (0.05 V) is crossed, at which point the robot retracts the object as fast as possible (720 mm/s). Temporal latency for the loop is ≈300 ms from initial contact due to the controller duty cycle.

Preliminary investigations showed that contacts could be made over a radial distance range of 80–180 mm without saturating the Hall effect sensor, or the bar slipping past the whisker tip before a retraction. Object speed ranged from 36–216 mm/s. Contacts were sampled at radial distance intervals of 1 mm, and speed intervals of ≈7 mm/s over the previously described ranges, respectively. In total 101 radial distances and 26 speeds were sampled, giving 2626 different radial distance and speed combinations. Contact combinations were randomly interleaved during data collection to limit any order effects, such as changing whisker properties across trials. For each contact combination, the whisker was deflected by the robot in both a clockwise and anticlockwise directions (−ve and +ve in x, see **Figure 1**), ensuring that the whisker did not undergo plastic deformations. The experiment was performed twice (two runs of clockwise and anticlockwise, generating four separate sets in total) to generate sufficient data for training the classifier. Data from each trial was stored separately. Deflections from the clockwise robot movement trials (−ve in x) were multiplied by −1, so data from all trials were directly comparable. Trials were ordered into arrays by speed and radial distance to contact. Each trial was aligned to peak deflection, and cut down to only 325 ms either side of the peak deflection.

2.3.2. SCRATCHbot data collection

A single column of whiskers from a SCRATCHbot head was used for this experiment. The upper and lower whiskers were removed, and the dorsal–ventral axis of the whisker was set

to 90° (horizontal). A vertical aluminium bar (13 mm cross-sectional diameter) was positioned at three different radial distances (70, 100, and 130 mm) at an azimuthal angle of 135° (180° = dead-ahead). The whisker was driven around the azimuthal axis (anterior–posterior) using a sinusoidal whisking pattern, with a retraction-protraction range of 60° (from 90° to 150°). The frequency of whisking was set at 2, 4, and 6 Hz, giving nine conditions in total. Eight contacts were made in each condition, four were used for training the classifier and four for testing (36 contacts for both training and testing). Data was streamed to the onboard computer through BRAHMS, and stored for later analysis. The whisker drive controller received no sensory feedback from the whisker sensor itself, only using the absolute measurement of theta to close a PID controller.

2.3.3. CrunchBot data collection

Data from the six whiskers was collected using an FPGA configured as a bridge to a USB 2.0 interface. Up to 28 whiskers can be connected to this FPGA bridge at one time. Using the vendor provided software driver and API (Cesys GmbH http://www.cesys.com/en/home.html), a user can request the data from all whiskers at minimum intervals of 500 μs (a sample rate of 2 kHz).

A "body whisk" behavior was included in the robot program to ensure consistent contact forces and speed. As the whiskers were not actuated the whole robot must rotate in a systematic way to simulate the whisking behavior of rats. Upon initial contact with an object the robot first reverses away a short distance before rotating at 15° per second toward the object for 1 s, then rotating at 15° per second away from the object for 1 s. This allows this whiskers to move over the surface of the contact object, collecting data about the radial distance (or in other experiments the orientation and texture of the surface). After the whisk the robot reverses again to clear the object, then rotates in a random direction and moves forward again. The whisker sweep during the contact phase is similar to a sinusoidal whisk.

For the verification of radial distance estimation a square cornered object was used. The robot was set in motion on a trajectory that would ensure the corner of the object would make contact with a particular whisker at a specific radial distance. The robot would then perform the body whisk movement, and the data would be stored. A dataset was collected for each whisker, consisting of five contacts at each of six points along the whisker (10 mm intervals over a 50 mm range) from the tip of the whisker. Though the whisker is 160 mm long, only 140 mm is external to the "follicle."

2.4. FEATURE-BASED RADIAL DISTANCE ESTIMATION WITH UNCERTAIN CONTACT SPEEDS

To successfully implement a feature-based classifier, appropriate features must first be found and extracted. Inspection of the whisker data showed that Hall effect sensor output voltage at peak deflection (proportional to bending moment M) could be used as a feature for radial distance discrimination at a given speed.

Feature f_1 can be defined as

$$f_1 = \max_t M(t), \tag{1}$$

where $M(t)$ is the deflection magnitude varying with time, measured by the Hall effect sensor in volts. Note that $t(f_1)$ is the time at $\max_t M(t)$.

Similarly, contact speed could be discriminated using deflection duration. Deflection duration was taken as the width of the deflection peak (prominent initial deflection in each trace of **Figure 4**). Deflection duration was measured using a threshold crossing on the sensor output. When Hall effect sensor output exceeded $\gamma = 0.05$ V a timer was initiated (t_1), and when Hall output subsequently fell below this threshold the timer was stopped (t_2). Feature f_2 (measured in ms) can thus be defined as,

$$t_1 = \min\{t : M(t) \geq \gamma\}, \tag{2}$$

$$t_2 = \min\{t : M(t) \leq \gamma, t_2 > t_1\}, \tag{3}$$

$$f_2 = t_2 - t_1, \tag{4}$$

where γ is the threshold. Colored arrows in **Figure 4** give examples of these measurements.

A model was generated of the relationship between each pair of features and the corresponding contact properties with polynomial regression (using polyfitn in MATLAB, bit.ly/polyfitN). Using linear least squares a model is generated that can be used to classify new data. Three arguments are required for the model, an array of independent variable values, an array of dependent variable values, and a model specification, namely the degree of the polynomial. A fifth degree polynomial was chosen as preliminary studies showed it provided good results. The independent variables in this instance were features f_1 and f_2. To find both radial distance and speed, two models were developed, with dependent variables of radial distance and speed, respectively.

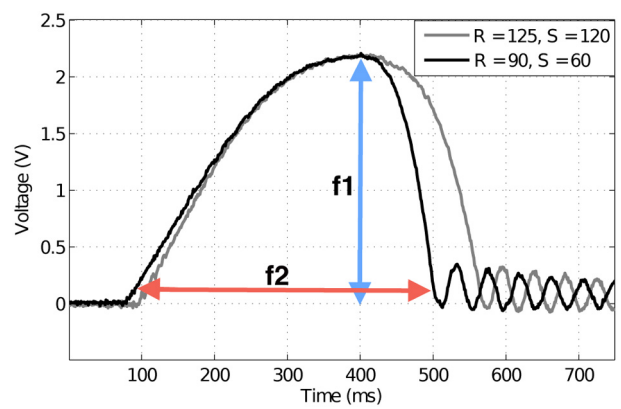


FIGURE 4 | Example deflection signals from the artificial whisker. Magnitude of deflection, or force, has been used previously as a discriminator of radial distance to contact. Here the two traces are at different radial distances (measured in mm), but create the same magnitude of deflection. Speed measured in mm/s. Colored arrows indicate how the extracted features for classification are measured. Peak deflection magnitude f_1 (blue arrow) and contact duration f_2 (red arrow) are used to discriminate radial distance to contact and contact speed, respectively.

The same polynomial regression operation was repeated on data from SCRATCHbot, though fewer summary statistics were generated as the dataset was smaller than that generated on the XY positioning robot.

On CrunchBot a linear regression was used. As robot motion is controlled, whisker contact speed variability is low between trials and a linear regression is sufficient for classification in this instance. To find an estimate of radial distance r,

$$r = a_1 f_1 + a_0, \tag{5}$$

was fitted to the data with a linear-in-the-parameters regression on the line, giving a least-squares fit for (a_0, a_1) for each whisker. Due to the small dataset size a "leave one out" protocol was used for classifier testing of CrunchBot. Four out of five contacts at each radial distance was used to train the classifier, with the remaining contact used for testing. This process was repeated using a different test contact each time. For each robot a mean absolute error statistic is given, which is more informative than mean error alone.

3. RESULTS

3.1. XY POSITIONING ROBOT

Figure 5 shows histograms of classification errors for both radial distance (**A**) and speed (**B**), and a scatterplot of the errors for each sample in the test set. Mean μ and standard deviation σ for radial distance and speed estimation errors was 1.2, 7.9 mm and 3.3, 25.8 mm/s, respectively. The mean absolute error was 6.2 mm and 20.4 mm/s for radial distance and speed, respectively. **Figure 6** shows mean classification error for radial distance error, with respect to true radial distance (**A**), and for contact speed, with respect to true contact speed (**B**).

Figure 7 shows a contour plot of the extracted features f_1 and f_2. While deflection magnitude is proportional to radial distance to contact for a given speed (**Figure 7A**), the precise degree of deflection is ambiguous without a separate measure of contact speed. On data generated on the XY positioning robot the duration of contact can provide this additional measure. This figure is examined further in Section 4.

3.2. SCRATCHbot

Results from SCRATCHbot show that the features and classifier developed on the XY positioning robot also apply to data collected from a whisking robot. As **Figure 8** shows, classification performance is almost perfect, with only one mis-classification of speed in the 36 contact test-dataset. **Figure 9** demonstrates that a key difference between data from SCRATCHbot and the XY positioning robot is the way whisker speed affects contact duration. Contact duration on the XY positioning robot increases as object speed increases, as object retraction is controlled by a feedback loop of a fixed duration. The faster the object moves, the further the whisker is deflected before a retraction is initiated. This increases contact duration in proportion to an increase in speed. Since SCRATCHbot is performing active whisking onto a static object, increased whisk speed results in a shorter contact duration. However, though the direction of the relationship is reversed, whisk speed still predictably affects contact duration. As in the XY positioning robot data, whisking at the same speed but different radial distances affects peak deflection magnitude (as can be seen in **Figure 9B**). Taking contact duration into account with a feature-based classifier allows accurate radial distance estimation at different whisk speeds.

3.3. CRUNCHBOT

Typical whisker deflections from CrunchBot are shown in **Figure 10**. Peak deflection magnitude for each contact is shown in **Figure 11**. Mean absolute error for radial distance estimation is shown in Table below.

	Whisker 1 (mm)	Whisker 2 (mm)	Whisker 3 (mm)	Whisker 4 (mm)	Combined (mm)
μ abs. Err	4.11	1.89	1.28	3.30	2.65

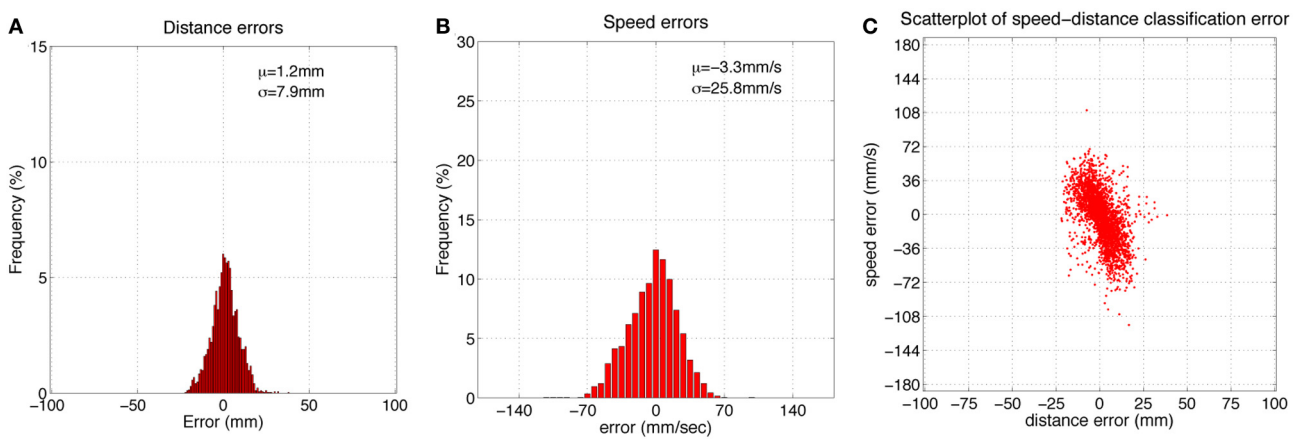


FIGURE 5 | (A,B) Histograms of radial distance and speed classification errors using the feature-based classifier. **(C)** Scatterplot of these errors for each point in the dataset. μ = mean, σ = standard deviation, sample size = 2626.

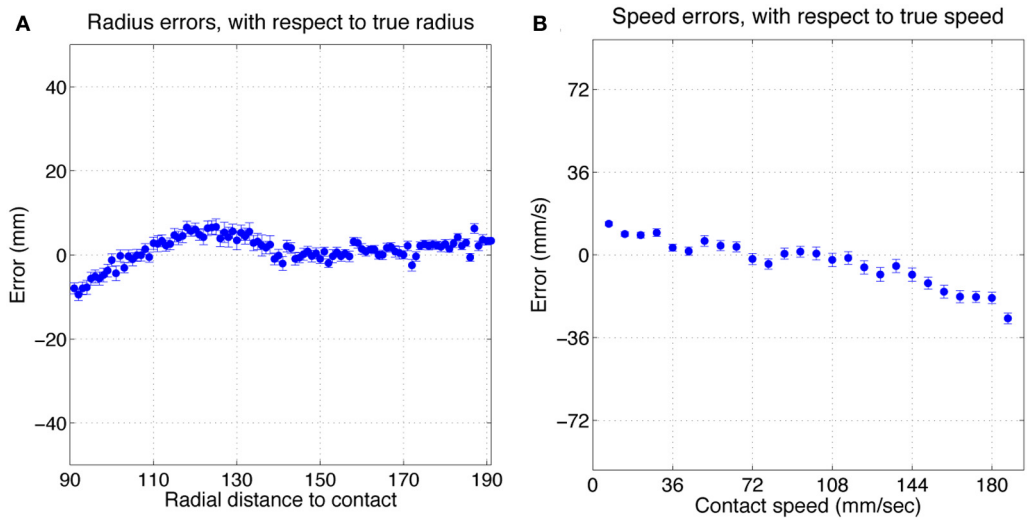


FIGURE 6 | Mean classification error for radial distance error, with respect to true radial distance (A), and for contact speed, with respect to true contact speed (B). Errorbars show standard error.

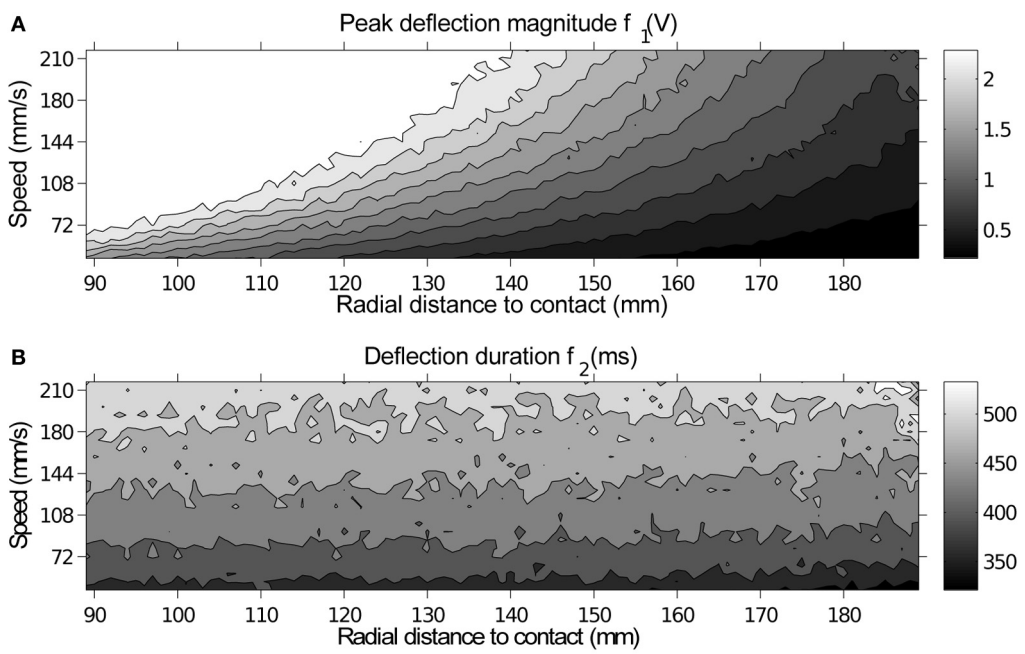


FIGURE 7 | A contour plot of peak deflection magnitude and duration for each contact showing how each feature varies with respect to contact parameters. Each point in the image corresponds to a location in the speed-radial distance space, which is equivalent in both plots. **(A)** Peak deflection magnitude f_1, brightness indicates higher deflection magnitude, measured in volts. All 10 contours are evenly spaced across the voltage range. **(B)** Deflection duration f_2, brightness indicates greater duration (measured in ms). All 6 contours are evenly spaced across the duration range.

Mean absolute error is very low, typically less than 5 mm over the 50 mm range tested. For some whiskers classification error is even lower, below 2 mm. These results compare favorably with results from controlled conditions on the XY positioning robot (Section 3.1) where speed was variable. This indicates that the noise in the odometry is low enough to ensure a consistent contact force and speed on this mobile robot.

4. DISCUSSION

Deflection magnitude is proportional to radial distance to contact, a relationship that is preserved across robot platforms, regardless of whisker material or actuation method. Whisker-object contact speed also affects deflection magnitude in a predictable manner on both the XY positioning robot and whisking SCRATCHbot. Controlling the whisker movement allows a very

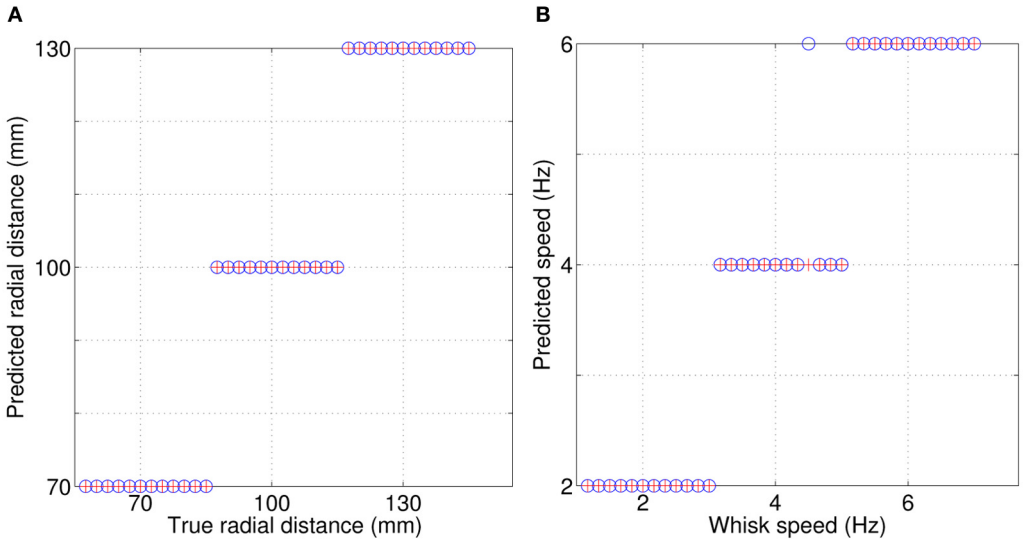


FIGURE 8 | Simultaneous classification of radial distance to contact (A), and whisk speed (B) on SCRATCHbot. Red crosses = true values, blue circles = classifications. Only one contact is miss-classified in the 36 contact test-dataset.

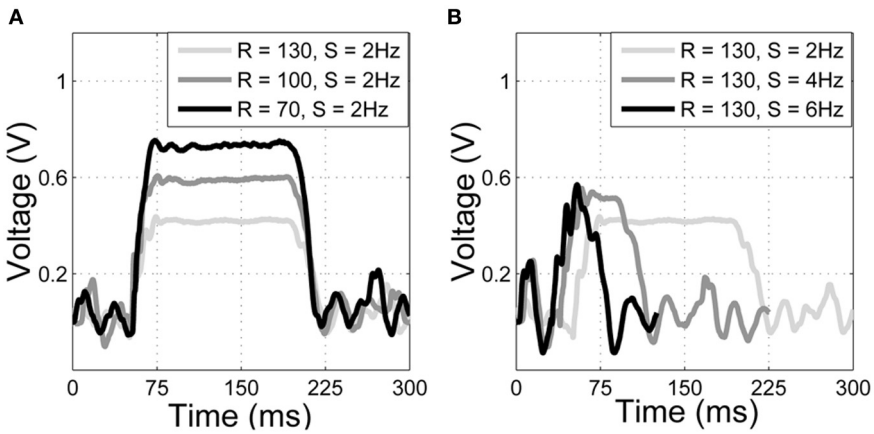


FIGURE 9 | Raw data from SCRATCHbot. Properties of the deflections match closely to those from the XY positioning robot (compare with **Figure 4**). **(A)** three deflections at different radial distances (R, in mm), but the same speed (S, in Hz). Peak deflection height varies predictably with radial distance. **(B)** three deflections at the same radial distance but at different speeds. Contact duration varies predictably with speed. Contact latencies are for clarity of presentation.

simple linear regression-based radial distance estimation method to be successfully implemented on CrunchBot, a mobile robot.

4.1. COMPARISON AND SYNTHESIS ACROSS ROBOT PLATFORMS

It may have been assumed that the relationship between whisker deflection, for a given radial distance to contact, and contact speed would be linear. However, though the relationship may be linear for a certain radial distance to contact, that linear relationship does not hold for all contact locations along the whisker. This can be seen on data from the XY positioning robot in **Figure 7**.

Inspection of **Figure 5C** reveals that classification errors on XY positioning robot data are not completely random. There is an interaction between the parameters, which can be seen as a skewing in alignment of the errors: positive errors in radial distance estimation occur more often with negative errors in speed estimation, and vice versa. A more detailed look at these effects can be seen in **Figure 6**. This figure shows how classification errors vary across each parameter range. The systematic trends reflect the fact both speed and radial distance are classified simultaneously. Both contact speed and radial distance estimation is best (error is lowest) in the middle of each range. Large radial distances are over estimated (negative errors on the left side of **Figure 6A**) and small radial distances are underestimated (positive errors on the right side of **Figure 6A**). The opposite effect is seen for contact speed (**Figure 6B**), where large contact speeds are underestimated and low contact speeds are overestimated.

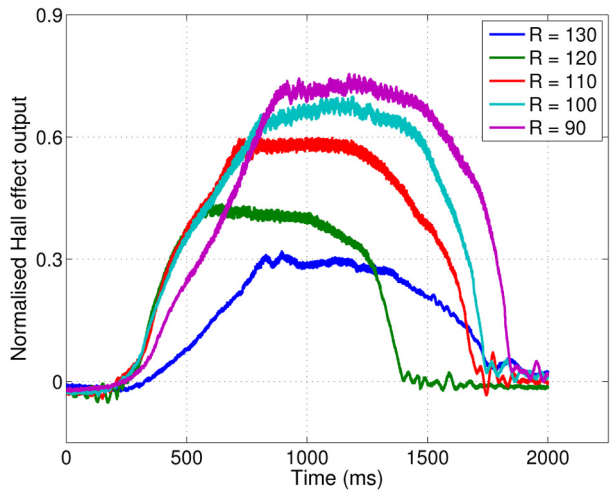


FIGURE 10 | Radial distance to contact affects deflection magnitude on the mobile CrunchBot robot. Five deflections at different radial distances (R, in mm), but the same speed. Peak deflection height varies predictably with radial distance.

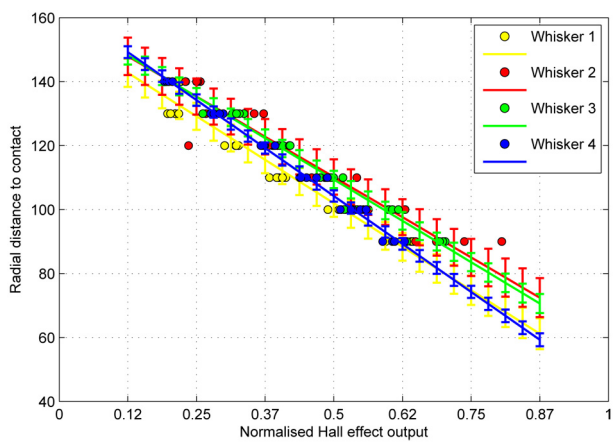


FIGURE 11 | Radial distance to contact for a given magnitude of whisker deflection (dots), and estimates of the standard deviation of the error in predicting future observations (errorbars) for each whisker. Sample size = 30 contacts per whisker, 120 in total. Radius measured in mm.

These effects can be explained by looking at how the extracted features change across the parameter space in **Figure 7**. A positive classification error of speed i.e., a jump from a dark to light area in **Figure 7B**, would result in a corresponding negative classification error of radial distance i.e., a jump from a dark to light area in **Figure 7A**. More transparently an increase in contact speed results in an increase in contact duration (f_2), while an increase in radial distance to contact results in a reduction in deflection magnitude. Misclassifications as the deflection "grows" in both height and duration would result in an over-estimation of speed and an underestimation of radial distance and vice versa. A prediction of this work for biological whisker systems is that rats would overestimate the radial distance to contact when contact speed is lower that expected, for example, as an object moves away from the rat.

All three robotic platforms presented here use different whisker materials and control strategies. These differences affect the temporal pattern of whisker deflections, which can be seen in **Figures 4**, **9**, and **10**. Whisker mechanical properties affect the initial rate of deflection change, and contact induced oscillations. Stiff fiberglass SCRATCHbot whiskers (**Figure 9**) result in a sharp initial increase in deflection and larger oscillations in between contacts. On the XY positioning robot and CrunchBot (**Figures 4** and **10**, respectively) robot movement speed changes throughout the contact, slowing down as peak deflection is approached, resulting in differences in gross deflection shape. While previously work has shown that whisker movement affects texture discrimination (Evans et al., 2009; Lepora et al., 2012a; Sullivan et al., 2012), our results show that such changes do not affect the key features extracted for radial distance estimation with the feature-based method presented here. Successful classification of radial distance on a particular robot platform does require the classifier to be trained on data from that robot, but the underlying principles are invariant for whisker material and robot movement. Specifically, that radial distance to contact affects the magnitude of peak deflection, and this is modulated predictably by contact speed.

4.2. RELATION TO OTHER STUDIES OF RADIAL DISTANCE ESTIMATION

These are the first published results of whisker-based contact speed estimation. As rats carefully control whisker motion, and as consequence contact speed, it may not be immediately apparent why this discrimination is important. Objects in the environment sometimes move and, for example, when a shrew is hunting crickets it needs to determine both the location and movement of that prey animal to execute an accurate fatal attack (Anjum et al., 2006). Another consideration is that rats have no spindles in their whisking muscles, and therefore do not have accurate proprioception of their whiskers (Diamond et al., 2008; Mameli et al., 2010). Since accurate radial distance estimation is dependent on well characterized contact speed, a signal-based method, such as the contact duration feature approach presented here, is of potentially great importance. From a robotics perspective this kind of tactile movement tracking may be useful for other tasks, such as in tactile manipulation.

It is difficult to determine how much better or worse the feature-based approach presented here is over previous whisker-based radial distance estimation methods. These are the first results where contact speed is both variable and unknown. On data collected on the XY positioning robot we report a mean absolute discrimination error of 6.17 mm with a 185 mm whisker. This is a normalized accuracy of 3.4% of whisker length. In a real-time application on board a mobile robot we report an average mean absolute discrimination error of 2.65 mm with a 160 mm whisker. This is a normalized accuracy of 1.65% of whisker length.

With fixed contact speed and a static beam equation-based method (Solomon and Hartmann, 2010) report contact localization accuracy between 0.3 and 0.88 mm on different surfaces with a 50 mm whisker. This is a normalized accuracy of ≈1% of

whisker length. This approach is more accurate than the results presented in the present paper, but reported under very different conditions (contact speed was very carefully controlled, slow, and not variable across trials). Further research is required to determine whether a similar approach would be successful in a more applied mobile robot setting, or under conditions of variable contact speed.

Rats have demonstrated radial distance estimation up to an accuracy of 2.5 mm (Krupa et al., 2001), which, with a 50–60 mm whisker, is a normalized accuracy of ≈4–5% of whisker length. These findings indicate that the feature-based approach presented here compares favorably with the performance of rats, even in the strict conditions of single whiskers making single object contacts and variable contact speed.

The results compare less favorably with typical range finding methods in robotics, such as laser range finders, or a Microsoft Kinect camera, which are both capable of sub-millimeter accuracy over short ranges (Khoshelham, 2011). Therefore the key contribution of whisker sensors for robotics is unlikely to be contact localization for its own sake, but in a wide range of other applied settings. Firstly, whiskers are useful in environments where other localization methods are impaired, for example, in smoky and dusty environments or underwater. Secondly, whiskers are small, low powered and can be manufactured cheaply, making them ideal for implementation as arrays on mobile robots. Finally, accurate characterization of contact properties such as localization and speed are essential for subsequent surface discriminations, as previous results have shown that whisker-based texture discrimination is improved when contact location and whisker movement are taken into account (Fend, 2005; Evans et al., 2009; Fox et al., 2009). A feature-based approach, as demonstrated here, could in principle provide a texture classifier with the necessary contact localization and speed information for improved discrimination. The integration of multiple texture reports over time into a local map of an object would also be dependent on accurate contact localization. This would be an extension of the tactile SLAM work published previously (Fox et al., 2012), and an area we hope to pursue in the future.

Hall effect sensors have some advantages over other sensing methods. Hall effect sensors are robust to damage, especially when housed in a rubber filled follicle, which is an important consideration when measuring whisker deflections as they are constantly striking objects in the environment. Hall effect sensors are also relatively inexpensive, and can be made quite small which makes them ideal for application to large arrays of whiskers. It has been proposed that rats determine radial distance to contact by encoding the bending of whiskers through moments at the base (Szwed et al., 2006). The Hall effect sensor is not a direct model of the rat follicle sinus, and does not report pure moments or forces at the whisker base but a combination of these properties along with whisker rotation angle about a pivot. Using a completely hard follicle rubber would remove the angular component of the deflection, but the sensor would no longer be able to measure bending. A whisker sensor could feasibly be designed that more closely models the physical structure of the rat follicle sinus, but it would remain an approximation. The artificial whiskers presented here capture the important aspects of contact induced deflections at both high and low frequencies, which is sufficient for understanding the abstracted principles of whisker sensing.

4.3. IMPLICATIONS FOR UNDERSTANDING BIOLOGICAL WHISKER SYSTEMS

Exploring active whisker movement in artificial systems highlights that active whisker control may be similar to aspects of eye movement control in active vision (Aloimonos et al., 1988). The field of active vision explores how the eyes may be moved to efficiently search an environment. The difference between active vision and active touch is in the scale of the movements with respect to the environment. In active vision the sensors can be moved to search a whole environment, for tasks such as scene identification of mapping (Davison and Murray, 2002). Active whisker touch can only be used over a very local region of the environment, therefore active whisker control may be thought of as analogous to micro-saccades for gaze stabilization (Collewijn and Kowler, 2008) or pupil diameter and lens focus, for luminance and depth of field control in the eye (Koss and Wang, 1972; Takehiko and Haruo, 1991). In vision optic flow and retinal slip can be used as an error signal for corrective eye movements for smooth pursuit (De Brouwer et al., 2001). Similarly, a measure of contact duration may be used by the rat as an error signal to correct whisker movements.

5. CONCLUSION

In this paper we have shown that a similar complementary approach can be successfully pursued in robotics. Certain experiments are much easier to perform on robots with fewer degrees of freedom, such as the XY positioning robot. The results from these experiments can save a great deal of time when implementing classifiers onboard mobile robots such as CrunchBot, or robots with high degrees of freedom such as SCRATCHbot. Mobile robot experiments can then generate predictions for biological systems, or drive further XY positioning robot research. This approach, and indeed these robots, could be used to answer a broad array of questions about active touch in the future.

We have shown that in each of the robots presented here, regardless of whisker material or actuation method, the radial distance to contact can be determined from peak deflection magnitude. In addition the speed of contact also predictably affects the amplitude of whisker deflection in each of these robots. By taking the speed of contact into account, radial distance estimation can be accurately performed in a range of settings. We predict that if whiskered mammals are using deflection amplitude (or degree of bending) to determine the radial distance to contact, the contact induced signal will change if the animal whisks at a different speed or force, and that this must be taken into account for accurate discriminations.

ACKNOWLEDGMENTS

The authors would like to thank the EPSRC doctoral training scheme, a University of Sheffield/EPSRC Doctoral Prize Fellowship, and European Framework 7 Grants BIOTACT (ICT-215910), and EFAA (ICT-270490) for funding this research. The authors would also like to thank the reviewers for their constructive appraisals.

REFERENCES

Ahl, A. (1986). The role of vibrissae in behavior: a status review. *Vet. Res. Commun.* 10, 245–268.

Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1988). Active vision. *Int. J. Comput. Vis.* 1, 333–356.

Anjum, F., Turni, H., Mulder, P., van der Burg, J., and Brecht, M. (2006). Tactile guidance of prey capture in Etruscan shrews. *Proc. Natl. Acad. Sci. U.S.A.* 103, 16544–16549.

Berg, R., and Kleinfeld, D. (2003). Rhythmic whisking by rat: retraction as well as protraction of the vibrissae is under active muscular control. *J. Neurophysiol.* 89, 104–117.

Bermejo, R., Vyas, A., and Zeigler, H. (2002). Topography of rodent whisking–i. Two-dimensional monitoring of whisker movements. *Somatosens. Mot. Res.* 19, 341–346.

Brecht, M., Preilowski, B., and Merzenich, M. (1997). Functional architecture of the mystacial vibrissae. *Behav. Brain Res.* 84, 81–97.

Carvell, G., and Simons, D. (1990). Biometric analyses of vibrissal tactile discrimination in the rat. *J. Neurosci.* 10, 2638–2648.

Chernova, O. F., and Kulikov, V. F. (2011). Structural differences between the shafts of mammalian vibrissae and hairs and their causes. *Dokl. Biol. Sci.* 438, 182–185.

Collewijn, H., and Kowler, E. (2008). The significance of microsaccades for vision and oculomotor control. *J. Vis.* 8, 1–21.

Davison, A., and Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 865–880.

De Brouwer, S., Missal, M., and Lefèvre, P. (2001). Role of retinal slip in the prediction of target motion during smooth and saccadic pursuit. *J. Neurophysiol.* 86, 550–558.

Diamond, M., von Heimendahl, M., Knutsen, P., Kleinfeld, D., and Ahissar, E. (2008). 'where' and 'what' in the whisker sensorimotor system. *Nat. Rev. Neurosci.* 9, 601–612.

Evans, M., Fox, C., Pearson, M., Lepora, N., and Prescott, T. (2010a). "Whisker-object contact speed affects radial distance estimation," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference* (Tianjin, China), 720–725.

Evans, M. H., Fox, C. W., Pearson, M. J., and Prescott, T. J. (2010b). "Tactile discrimination using template classifiers: towards a model of feature extraction in mammalian vibrissal systems," in *Proceedings of the 11th International Conference on Simulation of Adaptive Behaviour (SAB). FROM ANIMALS TO ANIMATS 11. Lecture Notes in Computer Science, Vol. 6226/2010* (Paris, France), 178–187.

Evans, M. H., Fox, C. W., Pearson, M. J., and Prescott, T. J. (2009). "Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task," in *Proceedings of Towards Autonomous Robotic Systems (TAROS 2009)*, eds T. Kyriacou, U. Nehmzow, C. Melhuish, and M. Witkowski (Derry, Northern Ireland), 19–24.

Evans, M., Pearson, M., Lepora, N., Prescott, T., and Fox, C. (2012). "Whiskered texture classification with uncertain contact pose geometry," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference* (Vilamoura, Algarve, Portugal), 7–13.

Fend, M. (2005). "Whisker-based texture discrimination on a mobile robot," in *Advances in Artificial Life* (Canterbury, UK), 302–311.

Fend, M., Bovet, S., and Hafner, V. (2005). The artificial mouse-a robot with whiskers and vision. *Signal [V]* 5, 6.

Fend, M., Bovet, S., and Pfeifer, R. (2006). On the influence of morphology of tactile sensors for behavior and control. *Rob. Auton. Syst.* 54, 686–695.

Fend, M., Bovet, S., Yokoi, H., and Pfeifer, R. (2003). "An active artificial whisker array for texture discrimination," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2003, Vol. 2* (Las Vegas, NV).

Fox, C., Evans, M., Lepora, N., Pearson, M., Ham, A., and Prescott, T. (2011). "Crunchbot: a mobile whiskered robot platform," in *TAROS'11 Proceedings of the 12th Annual Conference on Towards Autonomous Robotic Systems* (Sheffield, UK), 102–113.

Fox, C., Evans, M., Pearson, M., and Prescott, T. (2012). "Tactile slam with a biomimetic whiskered robot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference* (IEEE), 4925–4930.

Fox, C., Mitchinson, B., Pearson, M., Pipe, A., and Prescott, T. (2009). Contact type dependency of texture classification in a whiskered mobile robot. *Auton. Robots* 26, 223–239.

Grant, R., Mitchinson, B., Fox, C., and Prescott, T. (2009). Active touch sensing in the rat: anticipatory and regulatory control of whisker movements during surface exploration. *J. Neurophysiol.* 101, 862–874.

Hartmann, M. (2001). Active sensing capabilities of the rat whisker system. *Auton. Robots* 11, 249–254.

Hartmann, M., Johnson, N., Towal, R., and Assad, C. (2003). Mechanical characteristics of rat vibrissae: resonant frequencies and damping in isolated whiskers and in the awake behaving animal. *J. Neurosci.* 23, 6510–6519.

Juan, L., and Gwun, O. (2010). A comparison of SIFT, PCA-SIFT and SURF. *Int. J. Image Process.* 3, 143–152.

Jung, D., and Zelinsky, A. (1996). "Whisker based mobile robot navigation," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference, Vol. 2* (IEEE) (Beijing, China), 497–504.

Kappers, C., Huber, G., and Crosby, E. (1936). The comparative anatomy of the nervous system of vertebrates including man. *J. Nerv. Ment. Dis.* 84, 709–711.

Kardong, K. (2006). *Vertebrates: Comparative Anatomy, Function, Evolution*. Boston, MA: McGraw-Hill Higher Education.

Khoshelham, K. (2011). "Accuracy analysis of kinect depth data," in *ISPRS Workshop Laser Scanning, Vol. 38* (Calgary, AB), 1.

Kim, D., and Moller, R. (2004). "A biomimetic whisker for texture discrimination and distance estimation," in *From Animals to Animats 8, Proceedings of the International Conference on the Simulation of Adaptive Behaviour*, eds S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, (MIT Press), 140–149.

Kim, D., and Moller, R. (2007). Biomimetic whiskers for shape recognition. *Rob. Auton. Syst.* 55, 229–243.

Koss, M., and Wang, S. (1972). Brainstem loci for sympathetic activation of the nictitating membrane and pupil in the cat. *Am. J. Physiol.* 222, 900.

Krupa, D., Matell, M., Brisben, A., Oliveira, L., and Nicolelis, M. (2001). Behavioral properties of the trigeminal somatosensory system in rats performing whisker-dependent tactile discriminations. *J. Neurosci.* 21, 5752–5763.

Lepora, N., Evans, M., Fox, C., Diamond, M., Gurney, K., and Prescott, T. (2010a). "Naive bayes texture classification applied to whisker data from a moving robot," in *Neural Networks (IJCNN), The 2010 International Joint Conference* (Barcelona, Spain), 1–8.

Lepora, N. F., Pearson, M. J., Mitchinson, B., Evans, M. H., Fox, C. W., Pipe, A. G., et al. (2010b). "Naive Bayes novelty detection for a moving robot with whiskers," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference* (IEEE) (Tianjin, China), 131–136.

Lepora, N., Fox, C., Evans, M., Diamond, M., Gurney, K., and Prescott, T. (2012a). Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination. *J. R. Soc. Interface* 9, 1517–1528.

Lepora, N., Sullivan, J., Mitchinson, B., Pearson, M., Gurney, K., and Prescott, T. (2012b). "Brain-inspired bayesian perception for biomimetic robot touch," in *Robotics and Automation (ICRA), 2012 IEEE International Conference* (IEEE) (St. Paul, MN), 5111–5116.

Lettvin, J., Maturana, H., McCulloch, W., and Pitts, W. (1959). What the frog's eye tells the frog's brain. *Proc. IRE* 47, 1940–1951.

Lowe, D. (1999). "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference, Vol. 2* (IEEE) (Kerkyra, Greece), 1150–1157.

Lungarella, M., Hafner, V., Pfeifer, R., and Yokoi, H. (2002). "An artificial whisker sensor for robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Lausanne, Switzerland), 2931–2936.

Mameli, O., Stanzani, S., Mulliri, G., Pellitteri, R., Caria, M., Russo, A., et al. (2010). Role of the trigeminal mesencephalic nucleus in rat whisker pad proprioception. *Behav. Brain Funct.* 6:69. doi: 10.1186/1744-9081-6-69.

Meyer, J., Guillot, A., Girard, B., Khamassi, M., Pirim, P., and Berthoz, A. (2005). The psikharpax project: towards building an artificial rat. *Rob. Auton. Syst.* 50, 211–223.

Mitchinson, B., Martin, C., Grant, R., and Prescott, T. (2007). Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc. Biol. Sci.* 274, 1035–1041.

Mitchinson, B., Pearson, M., Pipe, A., and Prescott, T. (2010). "Biomimetic robots as scientific models: a view from the whisker tip," in *Neuromorphic and*

Brain-based Robots, ed J. Krichmar (Cambridge, UK: MIT Press).

Pearson, M., Mitchinson, B., Pipe, A., and Prescott, T. (2011). Biomimetic vibrissal sensing for robots. *Philos. Trans. R. Soc. B*. 366, 3085–3096.

Pearson, M., Pipe, A., Melhuish, C., Mitchinson, B., and Prescott, T. (2007). Whiskerbot: a robotic active touch system modeled on the rat whisker sensory system. *Adapt. Behav.* 15, 223.

Pearson, M. J., Mitchinson, B., Welsby, J., Pipe, A. G., and Prescott, T. J. (2010). "SCRATCHbot: active tactile sensing in a whiskered mobile robot," in *Proceedings of the 11th International Conference on Simulation of Adaptive Behaviour (SAB). FROM ANIMALS TO ANIMATS 11. Lecture Notes in Computer Science, Vol. 6226/2010* (Paris, France), 93–103.

Petersen, R., Brambilla, M., Bale, M., Alenda, A., Panzeri, S., Montemurro, M., et al. (2008). Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron* 60, 890–903.

Prescott, T., Pearson, M., Mitchinson, B., Sullivan, J., and Pipe, A. (2009). Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Rob. Autom. Mag.* 16, 42–50.

Quist, B., and Hartmann, M. (2012). Mechanical signals at the base of a rat vibrissa: the effect of intrinsic vibrissa curvature and implications for tactile exploration. *J. Neurophysiol.* 107, 2298–2312.

Rosenblueth, A., and Wiener, N. (1945). The role of models in science. *Philos. Sci.* 12, 316–321.

Russell, R. (1992). "Using tactile whiskers to measure surface contours," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Nice, France), 2, 1295–1299.

Russell, R., and Wijaya, J. (2003). "Object location and recognition using whisker sensors," in *Australasian Conference on Robotics and Automation* (Citeseer), 761–768.

Russell, R., and Wijaya, J. (2005). Recognising and manipulating objects using data from a whisker sensor array. *Robotica* 23, 653–664.

Sachdev, R., Sato, T., and Ebner, F. (2002). Divergent movement of adjacent whiskers. *J. Neurophysiol.* 87, 1440–1448.

Salisbury Jr., J. (1984). Interpretation of contact geometries from force measurements. *Proc. IEEE Int. Conf. Robot. Automat.* 1, 240–247.

Sarko, D., Rice, F., and Reep, R. (2011). Mammalian tactile hair: divergence from a limited distribution. *Ann. N.Y. Acad. Sci.* 1225, 90–100.

Schiebel, E., Busby, H., and Waldron, K. (1986). "Design of a mechanical proximity sensor," in *Robotics and Automation Proceedings 1986 IEEE International Conference* (San Francisco, CA).

Semba, K., and Komisaruk, B. R. (1984). Neural substrates of two different rhythmical vibrissal movements in the rat. *Neuroscience* 12, 761–774.

Seth, A., McKinstry, J., Edelman, G., and Krichmar, J. (2004). "Texture discrimination by an autonomous mobile brain-based device with whiskers," in *Proceedings of the IEEE International Conference on Robotics and Automation ICRA2004* (New Orleans, LA), 4925–4930.

Solomon, J., and Hartmann, M. (2006). Biomechanics: robotic whiskers used to sense features. *Nature* 443:525. doi: 10.1038/443525a

Solomon, J., and Hartmann, M. (2010). Extracting object contours with the sweep of a robotic whisker using torque information. *Int. J. Robot. Res.* 29, 1233–1245.

Solomon, J., and Hartmann, M. (2011). Radial distance determination in the rat vibrissal system and the effects of weber's law. *Philos. Trans. R. Soc. B Biol. Sci.* 366, 3049–3057.

Sullivan, J., Mitchinson, B., Pearson, M., Evans, M., Lepora, N., Fox, C., et al. (2012). Tactile discrimination using active whisker sensors. *IEEE Sens. J.* 12, 350–362.

Szwed, M., Bagdasarian, K., Blumenfeld, B., Barak, O., Derdikman, D., and Ahissar, E. (2006). Responses of trigeminal ganglion neurons to the radial distance of contact during active vibrissal touch. *J. Neurophysiol.* 95, 791–802.

Takehiko, B., and Haruo, T. (1991). Cerebral cortical and brainstem areas related to the central control of lens accommodation in cat and monkey. *Comp. Biochem. Physiol. C* 98, 229–237.

Towal, R., and Hartmann, M. (2006). Right-left asymmetries in the whisking behavior of rats anticipate head movements. *J. Neurosci.* 26, 8838–8846.

Towal, R. B., and Hartmann, M. J. Z. (2008). Variability in velocity profiles during free-air whisking behavior of unrestrained rats. *J. Neurophysiol.* 100, 740–752.

Vincent, J., Bogatyreva, O., Bogatyrev, N., Bowyer, A., and Pahl, A. (2006). Biomimetics: its practice and theory. *J. R. Soc. Interface* 3, 471.

Welker, W. (1964). Analysis of sniffing of the albino rat. *Behaviour* 22, 223–244.

Williams, C., and Kramer, E. (2010). The advantages of a tapered whisker. *PLoS ONE* 5:e8806. doi: 10.1371/journal.pone.0008806

Wilson, J., and Chen, Z. (1995). A whisker probe system for shape perception of solids. *J. Dyn. Syst. Meas. Control* 117, 104.

Young, W., Budynas, R., and Roark, R. (2003). *Roark's Formulas for Stress and Strain*. 7th Edn. (Beijing, China: Tsinghua University Press.).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 19 March 2012; accepted: 10 December 2012; published online: 02 January 2013.

Citation: Evans MH, Fox CW, Lepora NF, Pearson MJ, Sullivan JC, and Prescott TJ (2013) The effect of whisker movement on radial distance estimation: a case study in comparative robotics. Front. Neurorobot. ***6****:12. doi: 10.3389/fnbot.2012.00012*

Copyright © 2013 Evans, Fox, Lepora, Pearson, Sullivan, and Prescott. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in other forums, provided the original authors and source are credited and subject to any copyright notices concerning any third-party graphics etc.

Simultaneous localisation and mapping on a multi-degree of freedom biomimetic whiskered robot

Martin J. Pearson[1], Charles Fox[2], J. Charles Sullivan[1], Tony J. Prescott[2], Tony Pipe[1] and Ben Mitchinson[2]

***Abstract*— A biomimetic mobile robot called *"Shrewbot"* has been built as part of a neuroethological study of the mammalian facial whisker sensory system. This platform has been used to further evaluate the problem space of whisker based tactile Simultaneous Localisation And Mapping (tSLAM). Shrewbot uses a biomorphic 3-dimensional array of active whiskers and a model of action selection based on tactile sensory attention to explore a circular walled arena sparsely populated with simple geometric shapes. Datasets taken during this exploration have been used to parameterise an approach to localisation and mapping based on probabilistic occupancy grids. We present the results of this work and conclude that simultaneous localisation and mapping is possible given only noisy odometry and tactile information from a 3-dimensional array of active biomimetic whiskers and no prior information of features in the environment.**

***Index Terms*— Biomimetic, Tactile SLAM, Whiskered robot**

I. INTRODUCTION

Many small mammals explore their environment and navigate using the sense of touch through their array of active facial whiskers[1], [2]. This enables such animals to survive and forage effectively in habitats where visual cues are sparse. Similarly, the ability to explore and localise using whisker based touch would allow future autonomous vehicles to operate more effectively in environments where current sensory systems, such as laser, acoustic or machine vision, perform poorly. Confined, dusty or smoke-filled spaces such as collapsed buildings, unstructured, perhaps arboreal, environments and even turbid water, are examples of operational domains where a sense of touch could be beneficial for autonomous robot navigation[3]. A recent study of rats has found that tactile encoded locations of reward are represented in hippocampus[4], a structure within the brain that is associated with the consolidation of event memory[5]. Intriguingly, the hippocampal formation has also been demonstrated as instrumental in localisation and navigation within familiar (i.e., previously explored) environments[6]. It has been proposed that the *'place cells'* of hippocampus region CA1&3[7], in conjunction with the *'grid cells'* of the entorhinal cortex[8] and the granule cell mossy fibres of the dentate gyrus, encode a metric map of the immediate environment. It has also been suggested that through the integration of self-motion, head orientation and multi-modal sensory cues, the hippocampal formation maintains an estimate of the animal's location and orientation with respect to an allocentric reference frame (e.g. [9], [10]). Such a systems level description of hippocampal function has been directly compared to navigation algorithms developed for mobile robotic platforms in the past[11], [12], [13], [14], [15] and is particularly relevant when addressing the problem of SLAM[16]. The evidence that whisker-touch stimuli is associated with memory of spatial location is a biological observation that invites further investigation into the role that active tactile sensing could provide to mobile robotics. Here, we build upon our previous work demonstrating the principles of whisker based tactile Simultaneous Localisation and Mapping, or *tSLAM*[17]. This article summarises new results integrating this system with a multi-degree of freedom mobile robotic platform, a tactile sensory attention model of action selection to guide exploration, and an array of *active* (i.e. motile) biomimetic whiskers.

This work was supported by FET Proactive BIOTACT (ICT-215910) and EPSRC Grant number EP/I032533/1

[1]Bristol Robotics Laboratory, Universities of Bristol and of the West of England, Bristol, UK [2]Active Touch Laboratory, Sheffield University, Sheffield, UK

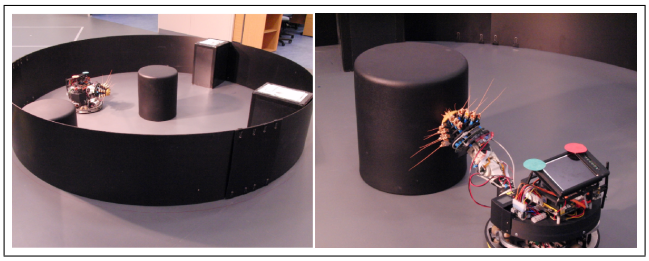


Fig. 1. Photographs of Shrewbot within the test arena (left) and exploring an object using an array of 18 active biomimetic whisker sensors (right)

II. METHOD

A. Robot hardware

Shrewbot consists of a omni-drive Robotino® (www.festo-didactic.com) *'body'*, an Elumotion (www.elumotion.com) 3 DoF *'neck'* and a custom built end-effector, referred to as the *'head'* (see photographs in Figure 1). The head was built around 3 disks of decreasing radii (60, 40 & 20mm) mounted at 20mm spacings along a 120mm long central column. Around the periphery of each disk are 6 mounts each holding a *whisker module*. Briefly, a whisker module consists of an instrumented composite whisker assembly mounted onto the shaft of a brushless dc motor and gearbox (see[18] for more detail). The motor allows $\pm 50°$ of rotation around the whisker resting angle to emulate the active touch *'whisking'* behaviour observed in mammals[19], one of the functional advantages of which is to increase the spatial coverage of the tactile array[3]. The most forward mounted whiskers were 60mm long, whilst those on the middle and rearward measured 115 & 160mm respectively. Each whisker module

has an embedded micro-controller for marshalling sensory data and updating the closed loop position control of the whisker angle. Desired angular positions of the whisker (as set by rotations of the motor shaft) are relayed to the micro-controller via an SPI bus which returns the sensory state of the whisker (2-dimensional deflections of the whisker shaft measured at the base w_x, w_y, and actual whisker angle w_θ). The SPI buses to all whisker modules (18 in total) are connected to the FPGA based bridge electronics which, in turn, is connected (via USB) to a MiniITX PC mounted on Shrewbot's body. Therefore, the nth sample of whisker sensory information from the entire array is denoted as: $W_n = \{\overrightarrow{w_1}, \overrightarrow{w_2}, \ldots, \overrightarrow{w_{18}}\}_n$, where $\overrightarrow{w_i}$ refers to the whisker sensory state (w_x, w_y, w_θ) of whisker i. Communication between the on-board PC and the position controllers for each actuator of the neck is also implemented via USB, whilst the motor drives of the Robotino are controlled across Ethernet.

B. Robot behaviour

The whisker sensory information, W_n, is geometrically transformed into a 3-dimensional array representing an ego-centric map of the volume surrounding the head of Shrewbot. Whereby, when a whisker makes contact with an obstruction in the environment the resulting increase in $||w_x, w_y||$ results in a corresponding increase in the *'tactile salience'* of that point in ego-centric space. Using a winner-takes-all arbitration mechanism across the map reveals the current most salient point, and it is here that the nose of the head of the robot (which we call the *fovea*) is directed. The arbitration mechanism is synchronised with the whisk rate of the robot (the rate at which the whiskers are moved back and forth), such that the target for foveation is selected at the end of each whisk. The ego-centric target is transformed through a forward model of the robot kinematics and a motor-recruitment based trajectory planner into a sequence of head and body motor actions to bring the fovea to this point. The *orient* is also coordinated with the whisker motion such that the whiskers will be in mid-protraction at the end of the head placement. When coupled with a non-uniform noise distribution across the saliency map and a simple inhibition-of-return mechanism, this iterative series of orients results in the emergence of more complex behaviour patterns such as exploration, wall following and novelty seeking [20]. This tactile attention model of action selection was used to control the robot during the data capture trials described below.

C. Data collection

Shrewbot was allowed to freely explore the arena (shown in Figure 1) through a series of short bouts of two or three minutes length. Separation into bouts allowed hardware maintenance when necessary and eased the management of the large time series that were collected. When a bout is started, the neck *'un-parks'*, and the whiskers are auto-calibrated. Occasionally, the robot halted in a position such that some whiskers would be in contact with a surface after un-parking, which would have interfered with calibration; on these occasions, the robot was moved a small amount before being restarted to avoid this interference. Post-hoc, these datasets were deleted or trimmed so that time periods including incorrect hardware operation (such as whisker sensor failure) were eliminated. The remaining trimmed bouts were then stitched together to form the dataset used here, which represents approximately 62 minutes of exploration, or $N_{\mathrm{W}} = 2857$ whisks (samples). In addition to the odometry and whisker sensory data logged on the robot, video footage was also recorded from a camera located 3 metres above the centre of the arena. This was also processed post-hoc to extract the ground truth pose of the robot used to qualify the performance of our localisation and mapping algorithm. A time lapsed sequence of such footage is presented in the supplementary video associated with this article.

D. Robot movement estimates

The robot pose was defined as the location and orientation of the head projected onto the 2D plane, and is denoted $\Psi_n = [x_n, y_n, \phi_n]^T$, where $\phi_n = 0$ indicates that the robot head is pointing in the same direction as the +ve y-axis, with x and y being the coordinates of the world frame (and superscript T the transpose operator). The change in this pose between whisks (samples) is denoted $\Delta\Psi_n = [\Delta u_n, \Delta v_n, \Delta\phi_n]^T$, where u and v are the coordinates of the robot head frame, wherein the robot head always points along the +ve v-axis. Then, these two series are related by $\Psi_n = \Psi_{n-1} \oplus \Delta\Psi_n$, where $\oplus$ is the motion composition operator as defined by [21]. Two estimates of $\Delta\Psi_n$ were measured. The first, denoted $\Delta\Psi_n^{\mathrm{GT}}$ (noiseless ground truth) was recovered from the overhead camera tracking the robot[1]. The second, denoted $\Delta\Psi_n^{\mathrm{OD}'}$ was recovered from the robot odometry. The initial position of the robot was recovered from the camera, and is denoted Ψ_1^{GT}.

E. Odometry model

The odometry estimate recovered from the robot, $\Delta\Psi_n^{\mathrm{OD}'}$, was systematically in error in comparison with the true movements of the robot measured by the overhead camera. Therefore, we used simple linear regression (least squares) to fit a linear/Gaussian model between the odometry and ground truth, given by $\Delta\Psi_n^{\mathrm{GT}} = G^{\mathrm{OD}}\Delta\Psi_n^{\mathrm{OD}'} + \Gamma_n^{\mathrm{OD}}$, where G^{OD} is a diagonal gain matrix with on-diagonal terms g_u, g_v and g_ϕ, and $\Gamma_n^{\mathrm{OD}} \sim [N(0,\sigma_u), N(0,\sigma_v), N(0,\sigma_\phi)]^T$ is a Gaussian vector process. Thus, each channel (x, y and ϕ) has an associated gain and standard deviation. The 'corrected odometry' estimate is then defined as $\Delta\Psi_n^{\mathrm{OD}} = G^{\mathrm{OD}}\Delta\Psi_n^{\mathrm{OD}'}$, and has the noise statistic Γ^{OD}. The values of the noise parameters correspond to 9%, 7% and 4% noise on the three channels, respectively. Finally, we define an estimate of $\Delta\Psi_n$ with an adjustable noise magnitude as

$$\Delta\Psi_n^{\mathrm{IN}} = \lambda\Delta\Psi_n^{\mathrm{OD}} + (1-\lambda)\Delta\Psi_n^{\mathrm{GT}} \tag{1}$$

[1]This estimate is, in fact, not noiseless but, since the noise includes no integral drift and is in any case small compared with the drift present in the odometry, we treat it as noiseless.

where λ is a value between 0 and 1. It is this estimate, and its associated noise statistic $[\lambda\sigma_u, \lambda\sigma_v, \lambda\sigma_\phi]$, that is used as the input to the particle filter based localisation algorithm.

F. Localisation algorithm

We used a standard particle filter approach to represent the posterior probability of pose within a 2-D grid based occupancy map as described in [17] and in more detail in [22]. The filter consists of a population of N_{P} particles, each maintaining a current estimate of pose of the robot head $\hat{\Psi}_n^{P_i}, i \in (1, N_{\mathrm{P}})$ within a local map $M_n^{P_i}$. A map represents $4m^2$ of space, with each cell representing 12.5mm x 12.5mm, i.e., 102400 cells per map[2]. The state of the particle filter was updated at the end of each whisk period, i.e., at approximately 1 second intervals, with the change in odometry $\Delta\Psi_n^{\mathrm{IN}}$ (derived from Eqn. 1) and the odometry noise statistic used to update the estimate of pose in each particle. A summary of sensory information from the whisker array gathered during the whisk was used to update each map and to determine the likelihood of each particle. The sensory information taken from the whisker array consists of a discrete representation of the space through which each whisker had travelled during the whisk, projected onto the 2-D plane. If no contact was made by a particular whisker during that whisk then all 10 points representing that whisk arc will be cast onto each map as regions with a lower likelihood of occupancy. If a contact did occur during that whisk then the location of the tip of the whisker at the point of contact will be fused into each map with an increased likelihood of occupancy (placement of a simple Gaussian shaped 'blob', similar to description in[17]). A grid cell in each M^{P_i} represents the probability of a contact being sensed if a whisker tip were located at that cell. Thus, the state of each map encodes sensor noise as well as the geometry of the world. Consequently, as the robot explores the environment each particle will develop a map that can be used to determine the likelihood of its current estimate of pose, i.e., $\Lambda_n^{P_i} \rightarrow \frac{1}{Z} P(M^{P_i} | \hat{\Psi}_n^{P_i})$. $\Lambda_n^{P_i}$ being the likelihood of the current pose estimate of particle i, based upon the probability of occupancy within the local map, given that it represents a history of all sensory observations including the most recent from the whisker array ($\frac{1}{Z}$ is a normalising function). For computational efficiency we adopted the *low-variance re-sampling strategy* as described in [22]. Ultimately the output from the filter is the pose estimate from the particle that currently has the highest posterior, which we denote $\Psi_n^{\mathrm{PF}} = P(\Psi_1^{\mathrm{GT}}, \{\Delta\Psi_j^{\mathrm{IN}}, W_n : j \in [2, n]\})$, where $P(.)$ represents the particle filter and W_n is the nth sample of (whisker) sensory data.

G. Performance analysis

Even where learned maps were apparently of high quality (reflected against ground truth), an unpredictable transform between these maps and the geometry of the world was a clear feature of our results. This makes direct quantitative comparison between the absolute pose estimates Ψ^{GT}, Ψ^{IN} and Ψ^{PF} impossible, a problem which has been highlighted recently for SLAM in general [23]. Kümmerle et al. outlined an approach to comparison between estimates by focusing on changes in each pose estimate, rather than the estimates themselves, noting that the choice of changes to compare must be problem-specific. They described their methodology for manual identification of particular pose estimates; owing to our large trials (thousands of realisations) we could not use a manual approach. Instead, we developed two automatic approaches to performance analysis, as follows, both of which gave similar results.

Pose change analysis: We analysed change in pose location according to each estimate over a range of intervals. Pose change for an estimate X at an interval of T_ζ is defined as $\Delta'\Psi_n^{\mathrm{X}}(T_\zeta) = \Psi_{n+L}^{\mathrm{X}} \ominus \Psi_n^{\mathrm{X}}$, where $\ominus$ is the inverse of $\oplus$ and $L = T_\zeta / T$, with T the average sample period across the dataset. The pose change performance metric is then defined as follows,

$$\zeta(T_\zeta) = \sum_{n \in [1, N_{\mathrm{W}} - L]} q_n^{\mathrm{PF}}(T_\zeta) / q_n^{\mathrm{IN}}(T_\zeta) \tag{2}$$

$$q_n^{\mathrm{X}}(T_\zeta) = Q\left(\Delta'\Psi_n^{\mathrm{X}}(T_\zeta) - \Delta'\Psi_n^{\mathrm{GT}}(T_\zeta)\right) \tag{3}$$

where $Q(\Delta'\Psi_n) = |x_n, y_n|$ is the operation that measures the magnitude of the location vector in a pose difference, $\Delta'\Psi_n$. Thus, $\zeta(T_\zeta)$ is a measure of the error in the pose change estimate due to the particle filter as a fraction of that due to the odometry, over an interval of T_ζ. Where performance is good, this metric is expected to decrease as T_ζ increases and odometry drift increases correspondingly.

Absolute pose analysis: The second approach is to attempt to identify the transform between the learned map and the ground truth geometry. We cannot, in general, use correlation between the maps to this end, since some of the learned maps are incomplete. Instead, we identify the rotation/translation $H^{\mathrm{PF}}(.)$ that minimizes the mean square error between $\Psi_n^{\mathrm{PF}^*} = H^{\mathrm{PF}}(\Psi_n^{\mathrm{PF}})$ and Ψ_n^{GT}, for $n \in [1, N_{\mathrm{W}}]$. Since this process will (undesirably for this analysis) reduce the impact of any integral drift, as well, we choose a similar transform for the odometry estimate, to generate $\Psi_n^{\mathrm{IN}^*} = H^{\mathrm{IN}}(\Psi_n^{\mathrm{IN}})$. We can then directly compare these transformed estimates as follows,

$$e_n^{\mathrm{X}^*} = Q\left(\Psi_n^{\mathrm{X}^*} - \Psi_n^{\mathrm{GT}}\right) \tag{4}$$

$$E^{\mathrm{X}^*} = \mathrm{median}\left(\{e_n^{\mathrm{X}^*} : n \in [1, N_{\mathrm{W}}]\}\right) \tag{5}$$

$$\omega_n = e_n^{\mathrm{PF}^*} / e_n^{\mathrm{IN}^*} \tag{6}$$

$$\Omega = E^{\mathrm{PF}^*} / E^{\mathrm{IN}^*} \tag{7}$$

with $Q(.)$ defined as above. That is, $e^{\mathrm{X}_{\mathrm{n}}^*} / E^{\mathrm{X}^*}$ are measures of absolute position error in the estimate X^* (at sample n, or over the whole dataset), whilst ω_n / Ω are measures of position error due to tSLAM as a fraction of that due to the odometry (per sample, per dataset). Since integral drift is expected to be large in the estimate Ψ_n^{IN}, these metrics will tend to be biased against tSLAM when it performs

[2]Note that all maps presented herein show only the central 3.2m square region, since the arena itself has a radius of only 1.5m.

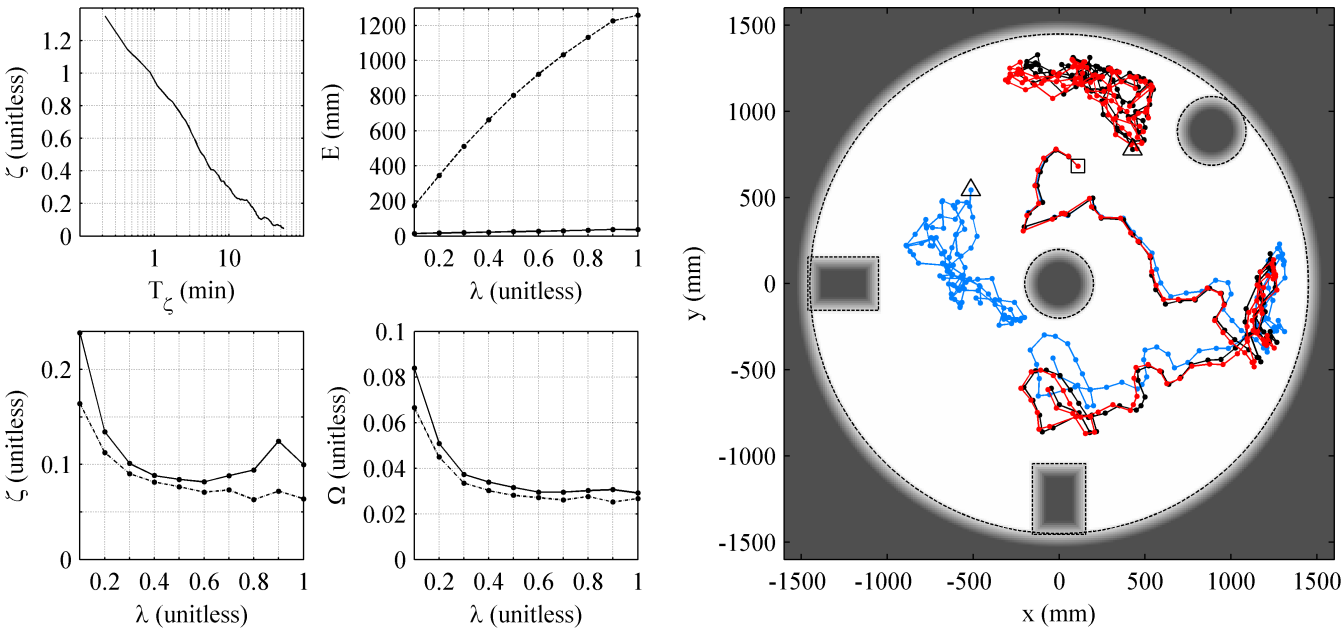


Fig. 2. Localisation only. (Top Left) Filtering improves performance (as measured through ζ) at measurement intervals (T_ζ) longer than about one minute, with much of the benefit measurable at an interval of around 30 minutes. (Bottom Left) Improvement due to filtering, as measured by ζ (T_ζ = 30mins.), increases as noise level, λ, rises. Mean (solid) and minimum (dashed) over 10 realisations. (Top Middle) Absolute error before (E^{IN}, dashed) and after (E^{PF}, solid) particle filtering, mean over 10 realisations. (Bottom Middle) Improvement due to filtering, as measured by Ω, increases as noise level, λ, rises. Mean (solid) and minimum (dashed) over 10 realisations. (Large panel, Right) The ground truth map, with example pose estimates for $\lambda = 1.0$ overlaid (example shown is that realisation returning median Ω). The contact probability for each map cell is shown from white to dark grey. Ground truth geometry is shown by dashed black lines. Location part of pose estimates Ψ_n^{GT}, Ψ_n^{PF} , and Ψ_n^{IN} are shown, respectively, in black, red and blue, for the first and last 100 samples of $N_{\mathrm{W}} = 2857$ in all. At the starting point (square) all three estimates agree ($\Psi_1^{\mathrm{GT}} = \Psi_1^{\mathrm{PF}} = \Psi_1^{\mathrm{IN}}$). Near the end (triangles) Ψ_n^{GT} and Ψ_n^{PF} (black and red) are in close agreement, but Ψ_n^{IN} (blue) is quite different ($n = N_{\mathrm{W}} - 100$).

well. However, identifying these transforms allows us to plot all pose estimates, and the map itself, in register, which is convenient to our presentation.

III. RESULTS

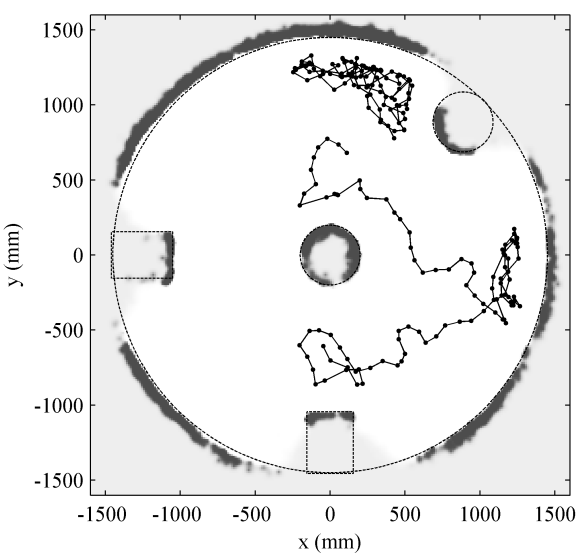


Fig. 3. Mapping only. As Figure 2, but map is that due to particle with maximum posterior after all samples have been presented, and only one pose trace is shown since all three estimates (PF, IN and GT) are identical.

A. Localisation

Initially we assume a map that reflects the true layout of the arena during the experiment, disable mapping, and focus on localisation only. As described above, the value at each cell of the map represents the probability that a whisker tip located here at peak protraction will report a contact. Thus, in building a 'ground truth' map for this analysis, it is necessary to represent the uncertainty in whisker tip location, so the objects in the map have 'feathered' rather than sharp edges. The map resolution we chose of 320 cells per edge was more than sufficient for this simple arena; the width of the feathering ($\pm$50mm) was chosen manually and approximately to maximize performance on the localisation task. Localisation performance over 10 realisations, as well as the map used and location traces from one example realisation, are shown in Figure 2. The summary result is that the system is successful in maintaining tracking in all 100 trials, with the median localisation error at the output of the particle filter, E^{PF^*}, rising only slowly (15mm to 37mm) as the input noise level (λ) is increased, whilst the mean localisation error at its input, E^{IN^*}, rises from 170mm to 1260mm. Both summary error metrics, ζ and Ω, similarly reflect the benefit resulting from filtering.

B. Mapping

Setting $\lambda = 0$ we can observe the generation of the map using the ground truth estimate of robot movement. An example map produced is shown in Figure 3: surfaces are

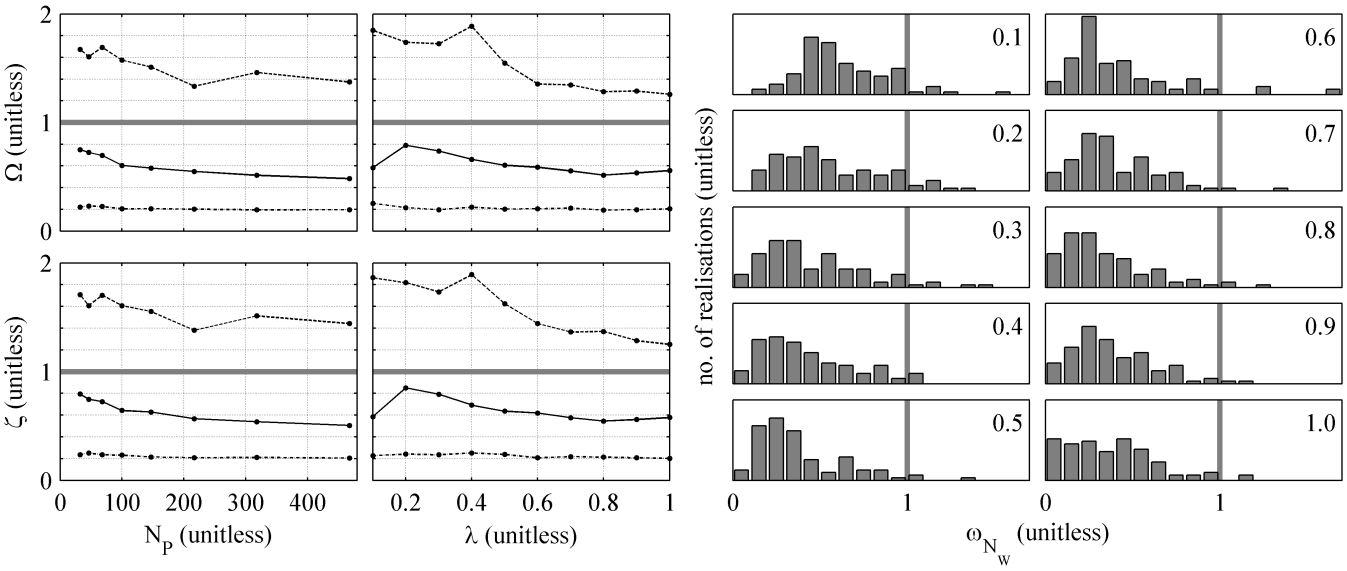


Fig. 4. SLAM localisation. (Curves, left) Error metrics Ω and ζ (T_ζ = 30mins.) against number of particles, N_P, or noise level, λ. Each curve is minimum (chained), median (solid) or maximum (dashed) over 100 realisations. When plotted against N_P (λ), curve shown is mean over λ (N_P). (Histograms, right) Each panel shows the distribution of relative final PF location error (in the last sample), ω_{N_W} , for N_P = 468 and a value of λ as annotated. (All) Unity is marked on all plots as thick grey line: values of all three error metrics below unity indicate an improvement due to filtering.

mostly well represented, but sharp concave corners have not been explored (the robots movement strategy means that it does not generally explore into these). Observation of such maps was used to inform the values chosen for some of the model parameters; specifically, the radius of the Gaussian shaped 'blobs' used to populate the map $r_\mathrm{BLOB} = 25mm$, and the number of such blobs to fuse into the map in the event of a positive contact $n_\mathrm{BLOB} = 4$.

C. Parameter search

The particle filter model (see Methods) has a number of parameters that are unknown and cannot be derived easily from the available data. Some of these (see above) were set manually informed by observation of the localisation-only and mapping-only conditions, reducing the parameter space substantially. Nonetheless, the parameter space remains large, so we used a Monte Carlo grid-based approach to find appropriate values for some further parameters when performing tSLAM. We varied N_P (between 100 and 1000), and the prior probability of occupancy across the maps p_PRIOR (between 0.025 and 0.1), across five different noise levels ($\lambda \in [0.1, 0.5]$) and five realisations. We reviewed the results to find the parameter point at each of the noise levels that gave the minimum-mean and minimum-maximum Ω over the five realisations (ten parameter points in total). These parameter points were not all the same, but we found that larger N_P was helpful, at least up to about 300, and that $p_\mathrm{PRIOR} = 0.1$ was common in 7 of 10 cases. Hence, we chose the single parameter point given by $p_\mathrm{PRIOR} = 0.1$ at which to perform further analysis, and restricted the particle count to a few hundred.

D. Tracking performance

We tested the performance of the tSLAM implementation *in toto* using the parameters determined above, varying only $N_\mathrm{P} \in [32, 468]$ and $\lambda \in [0.1, 1]$, with 100 realisations at each parameter point. The results are summarised in Figure 4. Across all noise levels and all values of N_P, the minimum (across realisations) value of both error metrics (Ω and ζ, curves in Figure) was similar and varied little, at around 0.2, whilst the maximum value of both was substantially over unity, indicating that filtering probably degraded the location estimate in some realisations. Increasing the number of particles tended to reduce the median and maximum of the error metrics, as expected, with these relationships apparently beginning to asymptote in the upper end of the tested range of N_P. None of the curves indicate that the highest noise level tested (raw odometry from the robot) was high enough to render this approach to filtering impossible. Measures of relative estimate improvement due to filtering at the end of the hour (ω_{N_W}, histograms in Figure) told a similar story: the noise level λ had no discernible impact on this error metric. Histogram analysis of ζ (not shown) showed essentially the same picture: 98 of 100 trials at λ = 1.0 showed an improvement in localisation ($\zeta < 1, T_\zeta$ = 30min) due to filtering.

Figure 5 displays examples of the maps that were learned at different noise levels; without reviewing all 100 maps for each case, these examples suggest the following. At a low noise level ($\lambda = 0.2$), all realisations produced useful maps (maps that appear to be moving towards a representation of the known geometry). At a medium noise level ($\lambda = 0.5$), most—if not all—realisations produced useful maps; it is not clear whether the maps in some realisations (such as the 'worst' one presented in the Figure) would be stable in

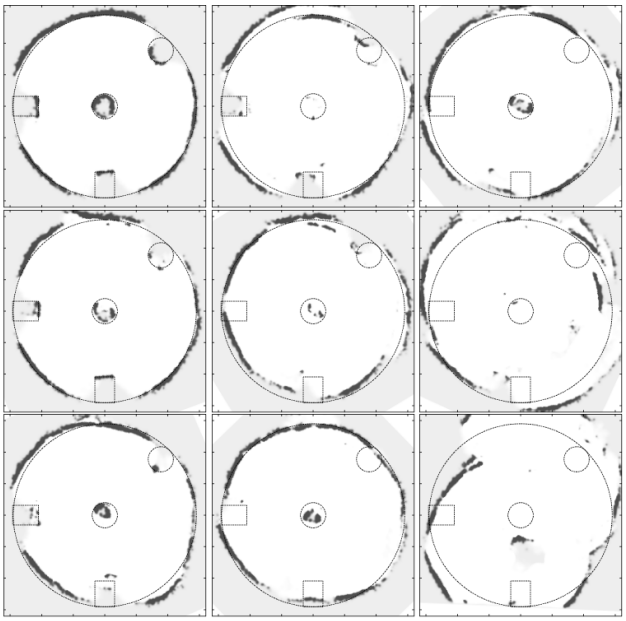


Fig. 5. tSLAM example maps. Each panel shows the final map produced by tSLAM after all N_{W} samples have been filtered, with N_{P} = 468. Top/middle/bottom is for $\lambda = 0.2, \lambda = 0.5$ and $\lambda = 0.8$, respectively. Left/middle/right is from the realisation at each of those parameter points that scored the lowest, median and highest value of Ω. All maps have been transformed through $H^{\mathrm{PF}}(.)$, for ease of comparison (see Methods).

the long-term. In the high noise case ($\lambda = 0.8$), it appears that in some realisations, at least, the map is not showing signs of converging on truth after one hour. In summary, at all noise levels, the expected outcome of using the system is a coherent map and thus, presumably, an improvement in location estimate, but the reliability of this outcome appears to suffer increasingly (as expected) as λ increases. The time lapsed graphical reconstruction of map development shown in the supplementary video to this article was generated using the medium noise level, i.e., $\lambda = 0.5$.

IV. DISCUSSION

A. Quantification

Given the relatively short data set, meaningful quantification of performance was challenging, as previously noted [23]. Since our implementation generally resulted in improved location estimates, our main error metric, Ω, is likely to be biased against tSLAM (see Methods), though an alternative error metric following the methodology of [23] generated a consistent analysis. Nonetheless, our quantitative results must be treated cautiously. Our main result is qualitative: it was possible to maintain tracking in the majority of cases. The degree of possible improvement in location estimates depends more on the nature of the problem than the characteristics of tSLAM. The current (one hour) dataset was about sufficient for one reasonably thorough exploration of this small arena (the robots exploratory behaviour is unhurried), so that we would expect quantitative benefits to be realised increasingly over longer time periods. One particular characteristic of our results was that the proportion of possible tracking failures ($\omega_{N_{\mathrm{W}}} > 1.0$) did not appear to increase with the noise level, λ, suggesting that any tracking failures that did occur may have been due to imprecision of our physical model rather than to a level of noise too high for the system to bootstrap.

B. Characteristics of tSLAM

A marked characteristic of our results was a small fixed error term (around $\Omega = 0.2, \zeta = 0.2$) that was not eliminated at any noise level nor in any realisation (Figure 4). This is characteristic of tSLAM (at least, as described) in that there may be regions of the map wherein the senses provide no information for localisation and odometry must be relied upon exclusively, for a time. An example would be a region of featureless floor. Such regions of 'sensory deprivation' need not exist in implementations based on vision or range-finding. We expect that it is this characteristic that is reflected in the lack of any examples below a threshold from 8,000 realisations contributing to Figure 4. It is worth noting, then, that the size of such regions, along with the time spent in them and the uncertainty in the odometry, probably constitutes a key limiting factor of a tSLAM implementation such as that presented. It seems likely that, in a majority of applications, large time periods of sensory deprivation could be engineered away by behavioural control. For instance, in a novel environment, actively employing a behavioural strategy such as wall-following until a sufficiently developed map is available might tend to reduce the incidence of erroneous correspondences between locations. Rats exploring novel environments do express thigmotaxis [24], as do blindfolded humans [25], interestingly, rats also make repeated returns to a home-base whilst gradually increasing the length of excursions which could play a significant role in correcting path integration errors[26].

C. Computational requirements

The computation times of this localisation solution are reasonable. With $N_{\mathrm{P}} = 100$ ($N_{\mathrm{P}} = 400$) the implementation described computes at 40ms (160ms) per cycle on a single domestic processor (2.4GHz). Using a more efficient re-sampling algorithm (not reported, but returns similar results) gave times of 14ms (55ms). These values compare well with the approximately one second between whisks in the robot, easily allowing online use with these parameters and off-the-shelf hardware. Since the computations are local to the robot's location, larger grid-based maps would not be expected to impact either processing time or memory bus load. Larger maps would probably benefit from storage optimisation (such as sharing of common data between particles), however, doubling the map extent to cover 8m by 8m at $N_{\mathrm{P}} = 100$ gives an unoptimised memory footprint approaching 160MB. Since the short-term error uncorrected by using tSLAM is larger than the 12.5mm map cell we used, the map resolution could also probably be reduced.

V. CONCLUSION AND FUTURE WORK

In our previous work we introduced a framework for whisker based tactile SLAM using a simple differential drive

robot and 6 static whiskers in a 1.5m square arena[17]. Here we have extended that work in three ways; Firstly, demonstrating the principle on a mobile robot that has many more degrees of freedom, including an 18 whisker array mounted as the end-effector of a 3 DoF 'neck' that it can be positioned within a 3D peri-personal workspace which, in turn, is mounted on an omni-drive mobile platform. Secondly, the whiskers are individually actuated to maximise the coverage of sensory exploration; Thirdly, we have shown, using a more comprehensive statistical analysis, that the tSLAM algorithm can provide a useful improvement in localisation performance even over a relatively short dataset. When a ground truth map estimate was provided, localisation provided very useful improvements in location estimate (and 100% reliability over 100 trials). When such a map was not provided (full tSLAM), an improvement in localisation was evident across realisations in reduced location error both over the course of one hour and at the end of one hour, versus the estimate due to the raw odometry. Furthermore, a large majority of realisations produced maps that reflected the geometry of the arena and in an improved location estimate over using odometry alone; accurate maps were, naturally, more common at lower odometry noise levels.

One obvious avenue for further investigation will be to scale up the problem, moving towards quantities appropriate for practical deployment, with larger maps and datasets taken over longer periods of time. Some parameters of the solution, such as map resolution and particle count, may be approximately sufficient as reported. This scaling-up will facilitate the exploration of the practicalities of the representation of usefully-sized regions in tactile maps. A second direction for extension will be to represent more than just occupancy or probability of contact, moving towards the use of feature labels such as texture or shape, as well as pre-assigned landmarks [27]. Moving in both of these directions will pave the way for a practical test of tSLAM as a navigation solution in a realistic environment where electromagnetic-based SLAM systems are unreliable, such as in smoke-filled buildings.

REFERENCES

[1] S. Vincent, "The function of vibrissae in the behaviour of the white rat," *Behavioural monograph*, vol. 1, pp. 1–81, 1912.

[2] B. Mitchinson, R. Grant, K. Arkley, V. Rankov, I. Perkon, and T. Prescott, "Active vibrissal sensing in rodents and marsupials," *Phil. Trans. R. Soc. B*, vol. 366, pp. 3037–3048, 2011.

[3] M. J. Pearson, B. Mitchinson, J. Sullivan, A. Pipe, and T. Prescott, "Biomimetic vibrissal sensing for robots," *Phil. Trans. R. Soc. B*, vol. 366, pp. 3085–3096, 2011.

[4] P. Itskov, E. Vinnik, and M. Diamond, "Hippocampal representation of touch-guided behavior in rats: Persistent and independent traces of stimulus and reward location," *PLoS ONE*, vol. 6, no. 1, January 2011.

[5] E. Kandel, I. Kupfermann, and S. Iversen, *Principals of neuroscience*, 4th ed. McGraw-Hill, 2000, ch. Learning and memory, pp. 1227–1246.

[6] R. Morris, "Spatial localization does not require the presence of local cues," *Learning and Motivation*, vol. 12, no. 2, pp. 239–260, 1981.

[7] J. O'Keefe, "Place units in the hippocampus of the freely moving rat," *Exp. Neurol.*, vol. 51, no. 2, pp. 78–109, 1976.

[8] T. Hafting, M. Fyhnand S. Molden, M. Moser, and E. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 11, pp. 801–806, 2005.

[9] J. Knierim, H. Kudrimoti, and B. McNaughton, "Place cells, head direction cells, and the learning of landmark stability," *The Journal of Neuroscience*, vol. 15, no. 3, pp. 1648–1659, 1995.

[10] D. Touretzky and A. Redish, "Theory of rodent navigation based on interacting representations of space," *Hippocampus*, vol. 6, pp. 247–270, 1996.

[11] C. Fox and T. Prescott, "Learning in a unitary coherent hippocampus," in *Artificial Neural Networks ICANN 2010*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6352, pp. 388–394.

[12] T. Strösslin, D. Sheynikhovich, R. Chavarriaga, and W. Gerstner, "Robust self-localisation and navigation based on hippocampal place cells," *Neural Networks*, vol. 18, pp. 1125–1140, 2005.

[13] V. Hafner, "Cognitive maps in rats and robots," *Adaptive Behavior*, vol. 13, no. 2, pp. 87–96, 2005.

[14] M. Milford, G. Wyeth, and D. Prasser, "RatSLAM : a hippocampal model for simultaneous localization and mapping," in *International Conference on Robotics and Automation*. IEEE, 2004, pp. 403–408.

[15] M. Mataric, "Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation," in *Animals to animats: Simulation of Adaptive Behavior*, 1990, pp. 169–175.

[16] J. Leonard and H. Durrant-whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems*. IEEE/RSJ, 1991, pp. 1442–1447.

[17] C. Fox, M. Evans, M. Pearson, and T. Prescott, "Tactile SLAM with a biomimetic whiskered robot," in *International Conference on Robotics and Automation*. IEEE, 2012, pp. 4925–4930.

[18] J. Sullivan, B. Mitchinson, M. Pearson, M. Evans, N. Lepora, C. Fox, C. Melhuish, and T. Prescott, "Tactile discrimination using active whisker sensors," *IEEE Sensors*, vol. 12, no. 2, pp. 350–362, 2012.

[19] P. Gao, R. Bermejo, and H. P. Zeigler, "Whisker deafferentation and rodent whisking patterns: Behavioral evidence for a central pattern generator." *J Neurosci*, vol. 21, no. 14, pp. 5374–5380, 2001.

[20] B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott, "The emergence of action sequences from spatial attention: Insight from rodent-like robots," in *Biomimetic and Biohybrid Systems*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, vol. 7375, pp. 168–179.

[21] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2006.

[23] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of slam algorithms," *Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009.

[24] S. Barnett, *The rat: A study in behaviour*. Aldine, 2007.

[25] O. Yaski, J. Portugali, and D. Eilam, "The dynamic process of cognitive mapping in the absence of visual cues: human data compared with animal studies," *J. Experimental Biology*, vol. 212, no. 16, pp. 2619–2626, 2009.

[26] O. Tchernichovski, Y. Benjamini, and I. Golani, "The dynamics of long-term exploration in the rat. part I: A phase-plane analysis of the relationship between location and velocity," *Biol. Cybern.*, vol. 78, no. 6, pp. 423–432, 1998.

[27] C. Fox, M. Evans, M. Pearson, and T. Prescott, "Towards hierarchical blackboard mapping on a whiskered robot," *Robotics and Autonomous Systems*, vol. 60, no. 11, pp. 1356–1366, 2012.

Extending Limabeam with discrimination and coarse gradients

Charles Fox, Thomas Hain

Speech and Hearing group
Department of Computer Science
University of Sheffield, UK
charles.fox@sheffield.ac.uk

Abstract

Limabeam is an approach to multi-microphone array processing for ASR which makes minimal assumptions about system geometry, instead searching for filters to maximise output likelihoods under a speech model. The first results of Limabeam on the AMI meeting corpus are given, then two extensions of the algorithm for this corpus. First, it is shown that the original local gradient following sticks in local minima, and a coarser gradient is used. Second, a new discriminative objective function is provided to handle mis-matched silence models. The extensions are based on examination of 2D receptive fields and 2D likelihood maps which are novel near-field analogs of radial beamformer response patterns, but do not show radial symmetry and have many local minima. The extended Limabeam improves WER on TDOA baselines on the AMI corpus, by 1% rel. when both are adapted with decodes and by 19% rel. when both adapted with ground truth.

Index Terms: ASR, beamforming, discriminative

1. Introduction

ASR in noisy, reverberant, distant-talking environments such as meeting rooms remains a difficult task [1, 2] as additive noise from overlapping speakers and non-speech sources and convolutional noise from reverberation degrade the signal. However in some cases, instrumentation with multi-microphone arrays may be possible, either via static installations or ad-hoc networks using, say, the participants' mobile phones. Such array signals can be processed with weighted delay and sum transforms, $\{w_{ij}\}$ of the multiple input channels $x_i[t]$ to an output channel $y[t]$ for ASR,

$$y[t] = \sum_i \sum_j w_{ij} x_i[t-j]. \tag{1}$$

Traditional beamformers [3] have used geometric assumptions to choose $\{w_{ij}\}$ to optimise criteria. For example, Time Delay Of Arrival (TDOA, [4]) is optimal for a single source in the presence of diffuse white noise; Maximum Variance Distortionless Response (MVDR, [5]) is optimal assuming the target source has the widest variance of any combination of sources. The Multiple Inverse Theorem (MINT, [6]) is optimal for discrete, non-diffuse noise sources. None of these assumptions are perfectly valid, and it has been noted [7] that such mismatch is acute as they all tend to be highly sensitive to low noise in any of their inputs.

Likelihood-maximising beamforming (Limabeam) was introduced to ASR by [8, 9] and tries to make minimal assumptions and instead try to search the $\{w_{ij}\}$ using gradient descent to maximise the likelihood of the signal under an ASR speech model. Using numerical gradient descent with a CMU Sphinx-3 speech model on CMU-8 [8] and CMU-WSJ-PDS corpora [9], gains of 31.4% relative have been reported [9], though with the caveat these significant gains we showed only for longer (>7s) utterances. These corpora contain only single static speakers reading written scripts, so that smaller utterances can easily be chunked together.

The present study tests for independent replication of these results on a standard, unscripted multi-speaker meeting corpus, AMI [1]. It finds no significant improvement over TDOA beamforming using basic Limabeam in this new setting – which has more realistic noise types and utterance characteristics – but suggests two extensions to Limabeam which then do allow it to improve on the TDOA scores. The first is a change of objective function, replacing likelihood with a discriminative likelihood ratio to avoid a problem with silence models in the AMI environment. The second is to replace local gradients with coarser gradients, allowing search to avoid some local minima.

Other improvements to Limabeam and AMI have been suggested, including sub-band and parameter-sharing Limabeam [10] [11], cepstral Limabeam [12],[13], and neural network recognition gains on AMI [14], which could all be combined with the present extensions.

2. Baseline experiments

Unlike CMU-8/WSJ-PDS, AMI consists of unscripted simulated business meetings by groups of four participants around a table. Recorded in three meeting rooms, each is instrumented with a circular, 100mm radius array of omnidirectional microphones in the centre of the table. Training and test sets of 12,000 (15.7 hours) and 1,188 (1.9 hours) non-overlapping, human-segmented utterances are defined, having independent sets of speakers.

Unlike CMU-8/WSJ-PDS, no chunking of short utterances is performed as our tests are intended as a proxy for more general meetings where speakers may move, or where diarisation is unclear. Audio used here is at 16kHz, and converted to PLP features [15]. All processing was per-utterance unless stated otherwise.

Baselines were obtained for Individual Head Microphone (IHM) and Single Distant Microphone (SDM) channels after training 3-state left-to-right hidden Markov Models (HMMs) with state-clustered phonetic decision tree ties states of 16-component GMMs to model output probabilities. Training followed a standard HTK mixup procedure [16]. Word error rates (WERs) are obtained with NIST *sclite* [17]. Decodes are based on HTK HDecode with a 3-gram language model trained from the AMI training set. Decode parameters were fixed at the start of baseline testing (s15p0) and were not changed to overfit later

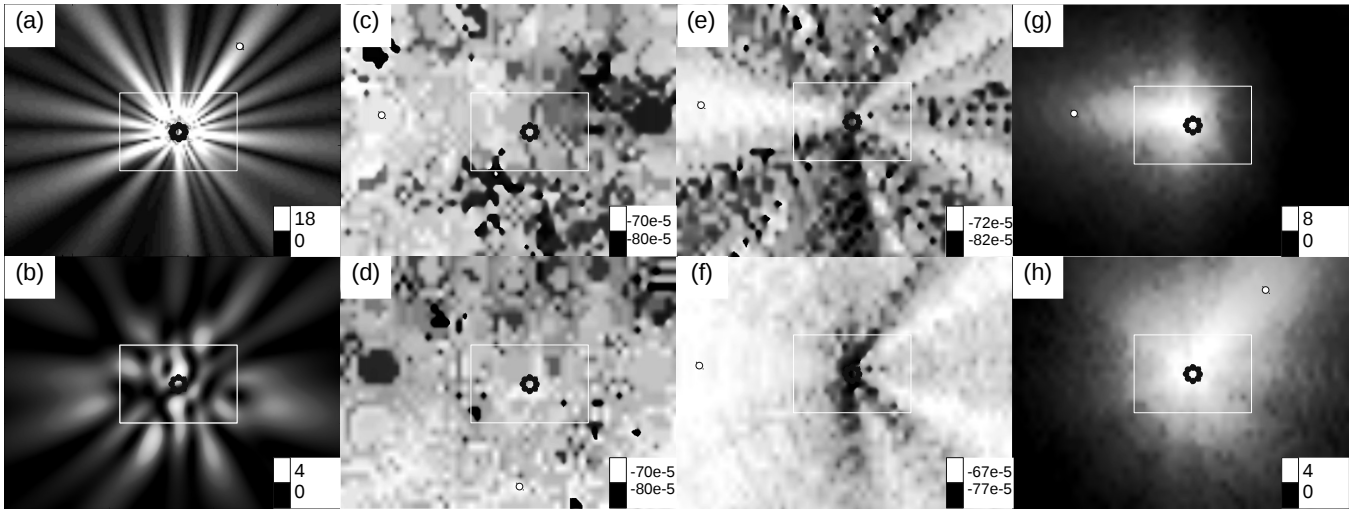


Figure 1: (a) 8kHz Receptive field for TDOA; (b.) 6kHz receptive field for random BF. (c),(d) Likelihood field for utterances. (e),(f) Desilenced likelihood fields. (g),(h) Discriminative likelihood ratio fields. White dots show speaker locations, black dots are microphones. x and y axis are physical coordinates in a 5.5x4m room. Rectangle shows a typical meeting table size.

experiments.

Baselines WERs were further obtained for standard beamformers. TDOA audio was created per-utterance via,

$$w_{ij} = \delta(j = \arg_\tau \max r(i, 0)), \tag{2}$$

where $r(i, 0)$ is the cross correlation function between channel i and a reference channel 0 chosen as that with the highest utterance energy, and δ is a Dirac Delta function. We also created baselines for audio output of the standard Beamformit (BFIT, [18]) software using its default settings, which is based on TDOA but including cross-utterance smoothing optimised for meeting rooms.

Table 1: *Baseline results.*

data	model	WER	S	D	I
IHM	xwrd	39.8	24.3	11.5	4.0
SDM	xwrd	66.0	45.5	16.7	3.9
IHM	MLLR(gnd)	23.4	12.0	8,7	2.7
SDM	MLLR(gnd)	50.9	32.4	15.8	2.7
IHM	MLLR(hyp)	37.2	17.4	16.5	3.3
SDM	MLLR(hyp)	62.7	39.4	19.9	3.4
TDOA	SPR	60.6	41.1	15.7	3.8
BFIT	SPR	61.2	40.0	17.8	3.5
TDOA	MLLR(gnd)	51.8	31.8	17.3	2.7
TDOA	MLLR(hyp)	59.4	36.3	19.9	3.2

The basic results in table 1 use Single Pass Retraining (SPR, [16]) to train new TDOA and BFIT models based on the previous IHM alignment. TDOA was found to outperform BFIT in this case.[1]

[1]Following this, GCC-PHAT[19] was also tested and found 11% worse that TDOA, and a static TDOA fixing parameters over all utterances for each speaker was 2.6% worse than TDOA. GCC-PHAT usually improves WER in strong reverberation but does not here. Static TDOA worsening was surprising as AMI speakers are seated and expected to have similar/smoothable TDOA values throughout, and shows the large effects of small head moves.

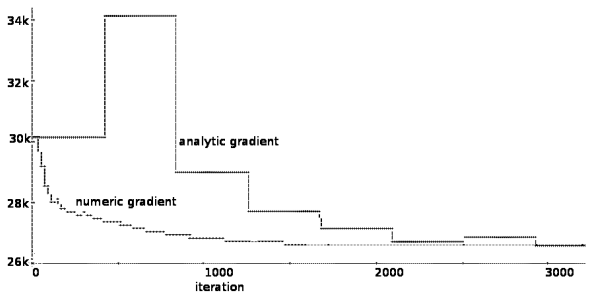


Figure 2: Convergence of gradient descent searches, using exact analytic and numerical estimated gradients, 10s utterance.

Maximum Likelihood Linear Regression (MLLR, [16], from IHM base) adaptation baselines for TDOA are also shown in table 1. MLLR is trained on a per-speaker basis using ground truth (gnd) and TDOA-decodes (hyp) test set transcripts; only means are adapted and 5 regression tree class transforms are used. Ground truth training gives an indication of what adaptation would achieve given large amounts of per-speaker training data.

3. Limabeam experiments

The most basic form of Limabeam models speech as a single GMM on MFCC features (a similar objective to a Wiener filter but optimising GMM feature likelihoods than GMM frequencies) and was tested on AMI. All Limabeam versions in the present paper are initialised to TDOA weights, and work with 8 microphones with 10 positive delay taps each (80 parameters).

Pilot experiments (e.g. fig. 2) found no significant differences in compute time or WER by switching from analytic gradients [9] to numerically computed gradients. Analytical solution makes many evaluations around each point to compute full-dimensioned gradients used to take a few large, accurate steps. Numerical gradient descent takes many smaller steps to give a smoother curve, but the same solution.

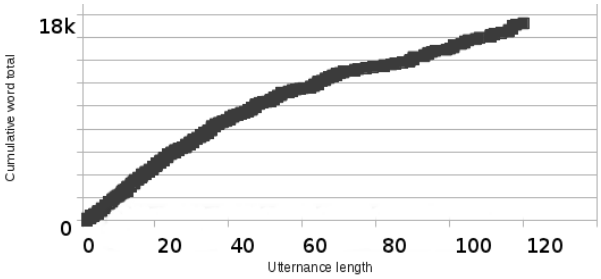


Figure 3: Cumulative distribution of AMI utterance lengths.

Standard BFGS Quasi-Newton was used to perform gradient-based optimisation in both cases (Octave *fminunc*, whose numerical gradients use w_{ij} moves of 6e-6). Numerical gradient computation allowed simple switching from MFCC to PLP features which gave 7.1% relative WER improvement. However even with this, the WER of table 2 did not improve on the relevant (TDOA, SPR, 60.6%) baseline, suggesting that LIMA-GMM likelihood is not a good proxy for WER optimisation.

A full HMM-based Limabeam was then implemented using HMMs. Alignment was performed at every parameter evaluation (80 for each gradient descent iteration) using HVite with ground truth transcripts. MLLR adaptation was applied and results are shown as LIMA-HMM in the table above. Again this standard Limabeam underperformed its baseline (TDOA, MLLR(gnd)).

Table 2: *Limabeam results.*

data	model	WER	S	D	I
LIMA-GMM	SPR	60.8	41.3	15.7	3.9
LIMA-HMM	MLLR	64.4	36.1	26.1	2.2

4. Inspection of corpus and beams

Previous work [9] showed on other corpora that Limabeam gave no improvement on short utterances, such as those less than 7s duration. To explore this for the AMI corpus, figs 3 and 4 show the word length distribution of AMI utterances and the WER of LIMA-HMM. These suggest that some of the overall poor performance is due to a large number of short utterances, confirming the findings of [9].

Inspection of the w_{ij} during optimization suggested that most utterances' parameters were shifting only by small amounts away from the initial TDOA solutions – comparable to the search step size of the optimiser. This could occur if TDOA solutions are already local (or global) optima, making it impossible for the optimizer to escape from them. To gain some intuition about the shape of the search space, we examined the theoretical (no reverb or noise) spatial receptive field patterns for various w_{ij} sets. Traditionally, beamformer responses are plotted only as functions of angle, not radius, under far-field assumption. This assumption might not hold for the distances in AMI corpus, where speakers sit <1m from the array. Fig. 1(a) shows a typical receptive field over one AMI meeting room (at 6kHz) for a TDOA filter focused on a speaker location. For TDOA filters it can be seen that the near-field effects are limited to a small region around the mic array so the usual radial plots are appropriate. However Limabeam can search a much

Figure 4: WER of standard LIMA-HMM by utterance length

larger, in our case 80D, space than the 3D manifold of TDOA solutions. Just one example of a receptive field given random weights is shown in 1(b), which indicates the ability to produce fields very unlike these classical side-lobed shapes, and where near-field effects do dominate the area where the AMI speakers sit. Such plots suggest many local minima where Limabeam could stick.

Single frequency receptive fields differ from *likelihood fields* however, which are illustrated in fig. 1(c) and (d). These show the effect of deliberately choosing $\{w_{ij}\}$ to focus on a speaker at the white dot, then moving the speaker around in simulation. At each pixel, the same utterance is placed there and the likelihood of the beamformed signal measured under the HMM model. This likelihood is plotted as the pixel color. The resulting likelihood fields are again very non-smooth and show many localised minima.

It must be emphasised that this 2D (actually a slice of a 3D room) field is not the space searched by the 80-dimensional LIMA search. But it does give a suggestion of the shape of the latter space and the types of minima found in it, and therefore that Limabeam is likely to get stuck often in such minima. The likelihood fields are highly diffuse, lacking clear single peaks at speaker locations. (This is the first publication to show such fields.)

5. Extensions

The AMI corpus contains strong, localised noise sources in at least one of the rooms having a loud server rack. It was found that 27% of segment time is assigned to silence. Together this could lead to optimising the filter to local minima that transform silence in new utterances to sound like silence in trained models, perhaps dominating any transform of speech sound. To test this hypothesis, two alternative optimisation objectives were constructed.

The *first* objective function is a per-frame likelihood average, but with silence phones excluded,

$$obj_1(\{w_{ij}\}) = \frac{1}{M}\sum_{n=1}^{N} b(n) \log P(x(n)|M, \{w_{ij}\}), \quad (3)$$

where there are N frames in the utterance and M contain non-silence phones in the current alignment indicated by the indicator function b, λ is the likelihood and M is the HMM model giving a best alignment.

Instead of simply excluding silence, the *second* objective function actively penalises the transformation of model silence into new-utterance silence by using the discriminative function,

$$obj_2(\{w_{ij}\}) = \log \frac{\prod_n P(x(n)|M, \{w_{ij}\})}{\prod_n P(x(n)|S, \{w_{ij}\})}, \quad (4)$$

where S is an HMM model consisting only of a single GMM trained on silence from the training set.

Sample likelihood fields for these two objective functions are shown in fig.1(e),(f) for obj_1 and fig.1(g),(h) for obj_2. Manual inspection of 10 utterances each suggested that these forms are typical, and that obj_2 tends to have a smoother form. This objective was thus selected for full testing.

While these 3D spatial fields look considerably smoother than the basic likelihoods, it might still be possible that local minima exist in the higher dimensional weight space, and so a second extension aims to help escape from any such minima by using less localised gradient estimates. Analytic gradients are perfectly local, measuring the slope at an infinitesimal point. Standard numerical gradient descent approximates this by measuring the gradient between finitely but closely spaced point. However for very non-smooth surfaces, such as fractals, local gradients are of little use and larger steps should be taken to escape from very small minima.

Two methods were tested to do this. Firstly, a simulated annealing search [20] (SA) and secondly, gradient descent searches using coarse gradient estimates, obtained by sampling points 2000 times further away from the current solution than used by the standard optimiser (GDx2000). Limabeam searches of most forms are computationally expensive to run (e.g. 400 days of 3GHz core time for the AMI test set) so these alternatives to basic gradient descent were used to give just an indication of alternative methods rather than an exhaustive search.

Results are shown in table 3. All runs here use Discriminative LIMA-HMM and MLLR training on either ground truth or SPR-TDOA decodes, and in TDOA model space.

Table 3: *Extended Limabeam results. 'gnd'=MLLR adaptation performed using ground truth data; 'hyp'=MLLR adaptation performed on decoded hypotheses. GD=standard gradient descent search; SA=simulated annealing search; GDx2000=coarse gradient descent search.*

search	MLLR	WER	S	D	I
GD	gnd	48.2	25.0	21.1	2.0
GD	hyp	59.3	36.4	19.8	3.1
SA	gnd	49.9	29.2	19.0	1.7
SA	hyp	63.0	39.0	21.3	2.6
GDx2000	gnd	41.8	24.0	15.8	2.0
GDx2000	hyp	58.7	37.6	18.0	3.1

The hyp-based experiments give marginal improvements over TDOA, (0.1% abs. for GD, no improvement for SA, and 1% rel. for GDx2000). Ground-truth LIMA-MLLR results give much more impressive improvement (19% relative, still not as large as in [9]) than ground truth TDOA-MLLR, suggesting if sufficient Lima-processed per-speaker training data was available then such improvements would also occur.

6. Conclusion

No significant WER gains for found on AMI with standard Limabeam. However by extending with discriminative objective and coarse gradient descent we obtained a 1% relative improvement, and a suggestion from gnd MLLR that larger gains up to 19% over gnd-MLLR standard Limabeam would be available given large per-speaker training data.

Unusual shapes formed by arbitrary parameter values emphasise there is more to beamforming than shown in traditional angular responses plots under far-field assumptions. For AMI, the space where speakers sit is susceptible to dominating near-field effects, which can produce non-radially symmetric local minima in both beamformer receptive fields and likelihood maps. Even conservatively quantising each weight to 10 possible values, gives a 10^{80} sized search space – comparable to the number of atoms in the universe – and impossible to search with any current computer. The space contains all known linear beamformers and many more. So any Limabeam-like search is heuristic. Adapting the coarseness of gradient descent to better fit intuition about minima distribution gives a small WER improvement. This suggests that future work could quantify such prior knowledge and use it to create custom search algorithms to better exploit it.

7. References

[1] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al., "The AMI meeting corpus: A pre-announcement," in *Machine learning for multimodal interaction*, pp. 28–39. Springer, 2006.

[2] Charles Fox, Yulan Liu, Erich Zwyssig, and Thomas Hain, "The Sheffield Wargames Corpus," *Proceedings of Interspeech.*, 2013.

[3] Robert J Mailloux, *Phased array antenna handbook*, Artech House Boston, 2005.

[4] Ralph O Schmidt, "Multiple emitter location and signal parameter estimation," *Antennas and Propagation, IEEE Transactions on*, vol. 34, no. 3, pp. 276–280, 1986.

[5] Jack Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[6] Jacob Benesty, Jingdong Chen, Yiteng Arden Huang, and Jacek Dmochowski, "On microphone-array beamforming from a mimo acoustic signal processing perspective," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1053–1065, 2007.

[7] Felicia Lim, Mark RP Thomas, and Patrick A Naylor, "Mintformer: A spatially aware channel equalizer," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[8] Michael L Seltzer, *Microphone array processing for robust speech recognition*, Ph.D. thesis, Carnegie Mellon University, 2003.

[9] Michael L Seltzer, Bhiksha Raj, and Richard M Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 489–498, 2004.

[10] Michael L Seltzer and Richard M Stern, "Subband likelihood-maximizing beamforming for speech recognition in reverberant environments," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 6, pp. 2109–2121, 2006.

[11] Michael L Seltzer and Richard M Stern, "Parameter sharing in subband likelihood-maximizing beamforming for speech recognition using microphone arrays," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 1, pp. I–881.

[12] Dominik Raub, John W McDonough, and Matthias Wölfel, "A cepstral domain maximum likelihod beamformer for speech recognition.," in *Interspeech*, 2004.

[13] Kshitiz Kumar and Richard M Stern, "Maximum-likelihood-based cepstral inverse filtering for blind speech dereverberation," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4282–4285.

[14] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 285–290.

[15] Hynek Hermansky, "Perceptual linear prediction (PLP) analysis of speech," vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[16] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, XA Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al., "The HTK book," 2006.

[17] National Institute of Standards and Technology (NIST), *Speech Recognition Scoring Toolkit (SCTK) Version 2.4.0.*, web resource: http://www.itl.nist.gov/iad/mig/tools,, 2010.

[18] X Anguera, "Beamformit, the fast and robust acoustic beamformer. in http://www. icsi. berkeley. edu/xanguera," 2006.

[19] Michael S Brandstein and Harvey F Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*. IEEE, 1997, vol. 1, pp. 375–378.

[20] William L Goffe, "Simann: a global optimization algorithm using simulated annealing," *Studies in Nonlinear Dynamics and Econometrics*, vol. 1, no. 3, pp. 169–176, 1996.

Machines learning - towards a new synthetic autobiographical memory

Mathew H. Evans, Charles W. Fox and Tony J. Prescott

Sheffield Centre for Robotics (SCentRo), University of Sheffield, Sheffield, S10 2TN, U.K.
{mat.evans, charles.fox, t.j.prescott}@shef.ac.uk

Keywords Synthetic, Autobiographical, Memory, Episodic, Hippocampus, Robotics, Predictive, Coding, Deep, Learning

Abstract. Autobiographical memory is the organisation of episodes and contextual information from an individual's experiences into a coherent narrative, which is key to our sense of self. Formation and recall of autobiographical memories is essential for effective, adaptive behaviour in the world, providing contextual information necessary for planning actions and memory functions such as event reconstruction. A synthetic autobiographical memory system would endow intelligent robotic agents with many essential components of cognition through active compression and storage of historical sensorimotor data in an easily addressable manner. Current approaches neither fulfil these functional requirements, nor build upon recent understanding of predictive coding, deep learning, nor the neurobiology of memory. This position paper highlights desiderata for a modern implementation of synthetic autobiographical memory based on human episodic memory, and proposes that a recently developed model of hippocampal memory could be extended as a generalised model of autobiographical memory. Initial implementation will be targeted at social interaction, where current synthetic autobiographical memory systems have had success.

1 Introduction

We receive a continual and very high band-width stream of sensory data during our waking hours. Our autobiographical systems process this data in a highly specific and adaptive fashion so as to provide quick access (within seconds) to relevant information experienced from hours to decades earlier. Autobiographical memory (AM) is defined as the recollection of events from one's life. Though similar in conception to Tulving's episodic memory [52], AM goes beyond simple declarative facts of an event to recall of rich contextual details of a scene [13]. AM is a prerequisite for developing the narrative self [32], related to and conceptually similar to the temporally extended self [43]. This version of the self is the individual's own life story, developed through experience, remembered, and projected into the future.

For robots to behave in a flexible and adaptable manner, and succeed in complex sensorimotor tasks, it is essential that they store their experience appropriately and use this information during online processing. For example, in human-robot interaction a memory of a familiar person or game would be intuitively advantageous especially in developing trust or attachment. This concept is familiar in probabilistic robotics, where an informative prior is essential for accurate inference. At a purely computational level, it has been shown that learning through 'episodic control' is more efficient than building a forward model or developing habits when experience is limited and tasks are complex, as is often the case in real-world robotics applications [24].

Innovative sensor designs, miniaturised HD cameras and affordable hard drives have endowed modern robots with an impressive capacity for gathering and storing information from the world. This information can be pooled across a range of modalities (e.g. vision, audition, touch, LIDAR,

depth) at high bandwidth. However, robots remain poor at extracting or retrieving task-relevant information when needed, either offline from their vast archives, or online during streaming [54].

In practice, data streams are filtered through feature detectors, processed by machine learning black boxes, or passively compressed into annotated histories where no such tools are available. As a result, despite having the ability to encode virtually everything that happens to them and access to vast stores of additional information online, robots are poor at determining which aspects of their history are important for making decisions and performing actions, or for framing engagement with people.

How does the brain solve this problem? By forming autobiographical memories that evolve over the life-time of an agent that places events in the context of the self and its goals. In this position paper we outline common approaches to autobiographical memory modelling in robotics, and contrast this with more modern understanding of memory function and organisation (see Figure 1). Four main principles are identified: compression, pattern completion, pattern separation and unitary coherent perception. To address each of these principles in a single architecture we outline a modelling framework based on predictive coding: deep learning for hierarchical representation and compression of sensory inputs (modality specific hierarchies in Figure 1, thought to reside in the sensory processing circuitry of the brain), and episodic memory formation through a particle filter and Boltzmann machine hippocampus model (Following Fox and Prescott [16], see Figures 2 and 3). We close with a discussion of our specific implementation goals, centring our episodic memory system within a rich sensorimotor system to aid ongoing processing.

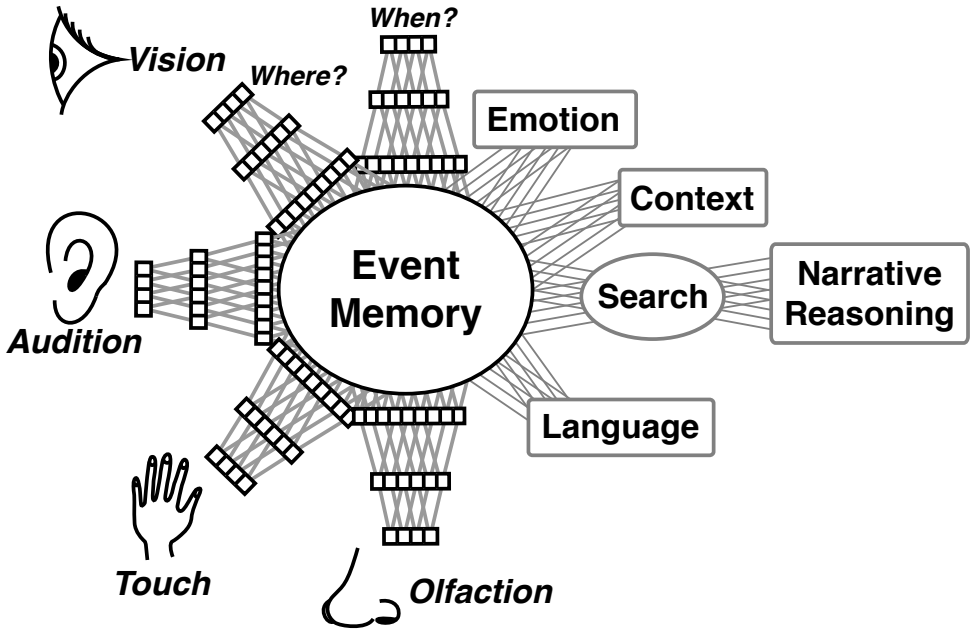


Fig. 1. An overview of our proposed model of biological autobiographical memory. Based on the basic systems model of Rubin [44], event memories are formed from convergent activity across disparate brain systems. Algorithmically we turn to deep networks for sensory processing [3], and predictive coding to frame learning [11]. Storage and recall is modulated by emotional, contextual and linguistic information. Narrative reasoning seeds searches, recognitions, reconstructions and predictions (the core of pattern separation and completion) by re-activating modality-specific schemas [21].

2 Current approaches to synthetic autobiographical memory

What is the state of the art in this field? Synthetic autobiographical memory (SAM) broadly has two flavours; (i) storage of unedited data streams (e.g. for maximum likelihood estimation [25]), or (ii) annotated high-level sequences (e.g. of social interactions [36]). The initial problem with approach (i) - storage capacity - is being alleviated to a degree by improvements in hard-drive capacity. However, storing a complete sensory history becomes un-wieldy quickly and the ability to search and utilise this vast store of information to inform ongoing processing becomes seriously problematic.

Recent success has been had in demonstrating the effectiveness of approach (ii) such as in cooperative tasks [36], or learning through social interaction [37]. These models are largely symbolic, similar in spirit to classic models of cognition like ACT-R [1] and others (reviewed in [54]), with hand-set higher level representations and action scripts coordinated into event memories [38]. As in early cognitive science, these models of memory have appealed to the computer metaphor for the mind, though thorough critiques of this approach have been articulated [10]. Pointedly, short-term and long-term storage of homogenous information is an idea from the 1960s but continues today in many models of memory [29, 5].

There remain fundamental gaps between cognition, memory and learning in these models. The compression of experience into memories is not handled in an adaptive way as it is in nature, and these models neither capture the rich feature-set of biological memory nor exploit advances in machine learning to increase the power and flexibility of stored information. Our position is that memory, and as a result our model of AM, is central to action and formed adaptively through experience with the world.

3 Characteristics of biological autobiographical memory

As noted in Wood et al. [54], the cognitive science and neurobiology of memory has changed drastically over recent years, but synthetic approaches have not kept pace. In particular, models of episodic and autobiographical memory are not compatible with theories of active, distributed memory systems [44] or ideas such as the predictive brain [11]. Forming new memories is not only about data compression, though this is important, but about selectivity and efficiency in the mechanisms of memory coding and retrieval. Cognitive scientists and neurobiologists assert that long-term memory formation is not a passive process of logging data into generic storage [2][44], but a highly active process depending on factors such as the depth to which an event is processed, or the wider context of the current task [26][42].

Biological memory is highly distributed and tightly-woven into ongoing cognition [54]. The neural underpinnings of different kinds of memory and imagery cannot be separated from the circuitry of perception and action. For example, when a person is asked to imagine rotating an object, neural activity is elevated in the visual and somatosensory cortices [12].

In order to match the function of human AM, a SAM system should have the following properties.

- **Compression.** Human memory systems are vast, and the detail to which events can be recalled is extensive. However, we certainly do not remember everything that happens in our lives, and items are not stored with equal detail. Reducing the volume of data to be stored through active data compression involves attenuating redundant information (efficient coding [48]), and prioritising attention and storage resources to information most critical for achieving goals.

- **Pattern Completion**. Reconstructing an event from brief exposure to part of that event, or from an impoverished, noisy or degraded version of the full scene e.g. when experiencing a familiar environment in the dark. This operation follows the idea of a *schema* [21][40]. Schemas can be thought of as generative models for a particular basic system [44]. Such schemas could be seeded with a small piece of information - e.g. you were at a children's party - and a rich scene can then be filled in - i.e. there was a cake, gifts, guests, and games were played.
- **Pattern Separation.** The process of transforming similar representations or memories into highly dissimilar, nonoverlapping representations [31]. This is important as we have many experiences that are similar to each other but nonetheless must be remembered as distinct. For example in discriminating edible from inedible plants. This function also relates to chunking of distinct event sequences from longer ongoing experiences, a core feature of episodic memory (see Figure 2). During later retrieval it is important only to recall relevant information for a particular task, thus separating relevant from irrelevant memories in a given context.
- **Unitary Coherent Perception.** Intuition attests that we experience a single *unitary* version of the world, not a probability distribution or blur over possible world states. Neither do we experience a world where independent percepts in a scene conflict with or contradict one another; our experience is *coherent*. These seemingly obvious features of experience, while having some recent experimental support [18], are in opposition to optimal decision making theories [4] and some forms of the Bayesian brain hypothesis (for a recent review see [39]). At the least, a maximum a posteriori approximation must be made at the 'percept' stage moment-to-moment even if the underlying computations are Bayesian.
  Helpfully, as a heuristic, unitary coherent (UC) perception may avoid the NP-hard computational complexity of full Bayesian inference of having to consider every possible interpretation of a current scene [18]. The existence of immediate perception as UC places strong constraints on models of AM: if AM is to store aspects of immediate percepts, then the recall and storage of memories would also be UC rather than fully Bayesian. This in turn will place constraints on theories of reconceptualisation of past episodes, which may otherwise require wider storage of probabilities.

3.1 The role of the hippocampal network in autobiographical memory

What is known about the neural circuits thought to underly AM in the brain? The extended hippocampal network, and its coordinated interaction with neocortex and subcortical structures, is thought to be the neurobiological substrate of the four AM functions outlined above. At a gross level, damage to the hippocampal-entorhinal cortex network famously causes anterograde amnesia [30]. At a cellular level the microstructure of the CA3 region suggests it likely serves as a 'convergence zone enabling information from different sensory modalities to be associated together (an idea originated by Marr [27]).

Hippocampal neural activity shows attractor dynamics [53] enabling content-addressable reactivation of the entire stored representation through activation of a particular memory by the dentate gyrus (DG) [28]. This role of older DG granule cells in associative pattern completion is complemented by the finding that neurogenesis supports pattern separation by young DG granule cells [31]. Critically, for pattern completion in AM, it has been shown that patients with hippocampal lesions cannot recall nor imagine spatially coherent events [22], and this reconstructive process can be modelled at the level of neurons [9]. Together these results highlight the importance of this structure in the combination and coordination of memory sequences, tying together activity across distributed circuits in the brain. We base our model of episodic memory on the hippocampal system.

These findings from the study of biological memory suggest that to create effective SAM we could adopt the following strategy. Compression and pattern completion can be accounted for by an implementation of the predictive brain hypothesis [11]. This allows both efficient and accurate recognition of events from partial information, as well as affording decision making, planning and reconstruction through inference in the model. High level unitary representations of a scene or event are then coordinated into coherent sequences by iterative pattern completion and separation operations in a model based on the hippocampus.

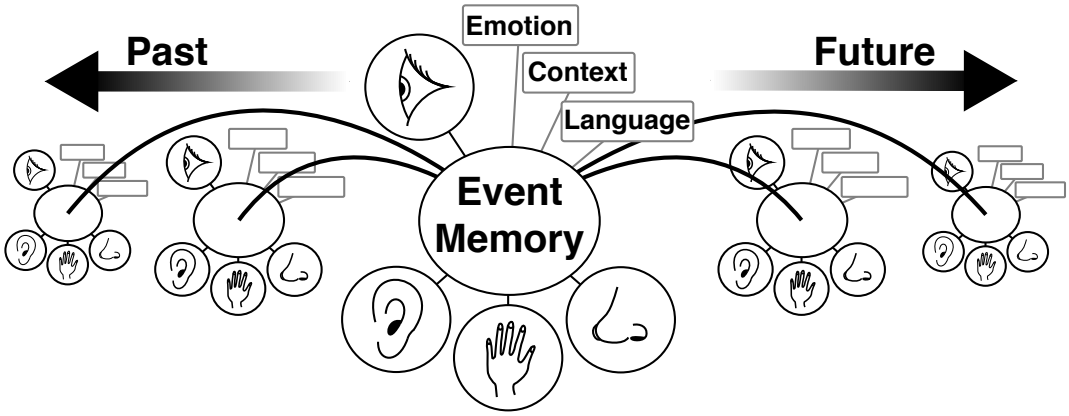


Fig. 2. Episodic memory in our model functions by combining disparate information across modalities into a single coherent percept from moment to moment. These transient percepts are monitored and coordinated over time into event sequences and episodic memories through iterative pattern completion and separation operations.

4 Capturing the characteristics of biological autobiographical memory in a single modelling framework

4.1 Compression and Pattern Completion: The Predictive Brain Hypothesis

In distributed episodic memory the individual sensor or motor systems encode their own memories in a domain-appropriate format [44]. Encoding of sensory input has been shown to be very efficient in biological systems, with neural populations being tuned to the scene statistics of the world [48]. A further efficiency is not to simply encode the world as it is, but to develop a generative model of the world and encode only events that could not be predicted.

So called Predictive Brain hypotheses [11][19]have become popular in recent years, claiming to provide a unified computational account of a range of cognitive capacities and seeking to explain an increasing number of neuroscience phenomena e.g. [20][41][50].

Reconstructing a past event can be understood as one of the functions of such a prediction engine, using past experience to anticipate and make sense of events as they happen. Seeding such a mechanism with appropriate clues will allow retrieval of a past episode. The same system, operating continuously, can also serve to fill-in and enrich the representation of the current situation, preparing the platform for more informed and appropriate action. It is well known that people can fill in sensory scenes with expected information, and the reconstruction of memories

from experienced or suggested fragments can lead to false memories even in individuals with 'highly superior' AM [33].

This powerful and efficient learning strategy would be highly appropriate for SAM and has not previously been applied to that domain.

4.2 Deep Learning: Towards a Synthetic Predictive Brain

A recent development in the field of machine learning is the arrival of practical deep learning systems [3] for abstracting information from data in an unsupervised way. Though neural networks have been investigated for decades [7, 45], recent developments of efficient training methods for deep (i.e. more than three layer) neural networks (DNNs) finally provide existence proofs of algorithms that can reconstruct (predict) complex sensory scenes in the manner required by predictive brain hypotheses.

Current excitement around DNNs can be ascribed to two prominent features. Firstly, they are now able, for the first time, to match or exceed human performance in certain benchmark pattern recognition tasks due to large-scale implementation on GPUs (see [3] for a review). Secondly, the kinds of invariant higher order representations developed by neural networks bare striking similarity to the tuning curves of neurons in higher order sensory cortex [55].

The unique processing architectures that result from training a DNN on data from a particular sensory domain could be described as a schema [21], an important concept in memory as discussed earlier. We propose that it is precisely the interaction of a compact episodic memory, and the sequential re-activation of an appropriate schema that underlies the compression and pattern completion capabilities of autobiographical memory. Therefore, it is to these DNN methods that we turn to compress sensory data streams and provide inputs to the episodic memory system.

4.3 Pattern Separation and Unitary Coherent Perception: Hippocampus as a Unitary Coherent Particle Filter

The vast majority of invasive neuroscience research uses rodent models. As a result, spatial reasoning or navigation tasks are often used to probe the biological foundations of memory. Many models of hippocampal function that are faithful to circuit-level details are therefore based on navigation or spatial reasoning [6]. Our model [16] has been developed as an effective algorithm for spatial inference (an idea recently expanded upon by Penny et al. [35]), while at the same time accounting for other known hippocampus-circuit phenomena such as the existence of sensory-pattern-specific and object-specific ('grandmother') cells. In robotics too, navigation has historically been a widely studied problem leading to widely-cited solutions to the Simultaneous Localisation And Mapping (SLAM). Our model may be viewed as a particular bio-inspired implementation of the SLAM algorithm, with EM algorithm steps mapping to neural activation and Hebbian learning respectively.

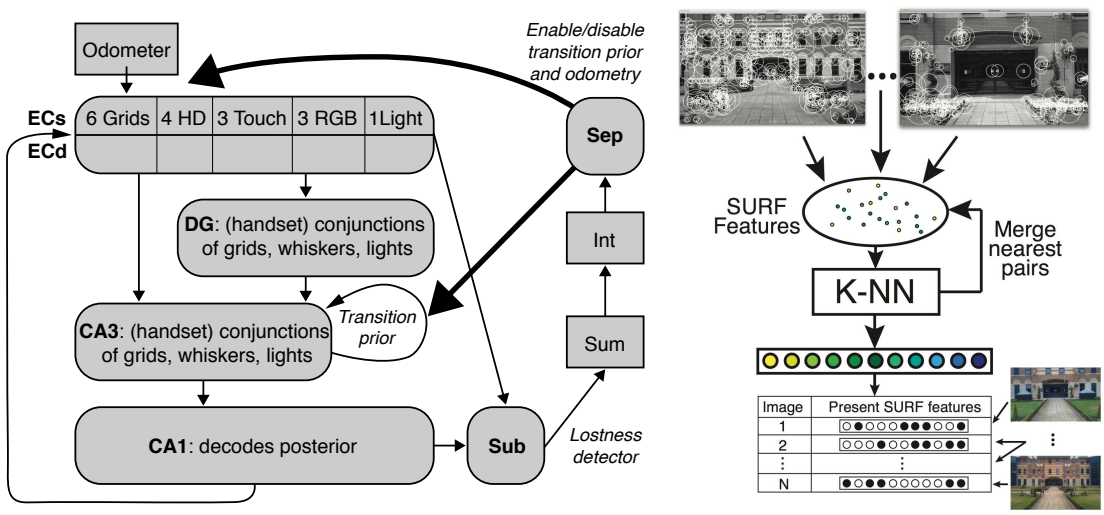
Our approach combines and extends research in both robotics and computational neuroscience fields: mapping well studied algorithms that have been shown to perform spatial reasoning and memory in navigation onto a neural circuit known to perform this function. Here we propose an update of this navigation model to autobiographical memory, echoing the well-articulated arguments of Buzsáki and Moser [8], based on the theory that the underlying computations for spatial and semantic memory are fundamentally the same.

The problem of locating oneself in an environment involves associating a given percept with a particular *place* (localisation) and encoding the transitions from one place to another by monitoring sensory differences, odometry and memories of previous traversals of the same environment. Autobiographical memory can be seen as essentially the same computation, but associating a

given percept to an *event*, while transitions from event to event can be handled by recursive pattern separation and completion operations. The analogy between space and AM is seen clearly in the well-known mnemonic technique of the 'memory palace' which explicitly uses spatial memory to organise sequences of object percepts [49].

It had been proposed before that the behaviour of animals during learning could be modelled as a particle filter [15], and elsewhere that sequential learning in the hippocampus could be modelled with a Temporal Restricted Boltzmann Machine (TRBM) [51], a machine learning algorithm. Fox and Prescott [16] took a TRBM model for navigation and extended it to include a mapping to hippocampal-entorhinal cortex circuitry that included: unitary coherence; sensor/odometry inputs from entorhinal cortex to DG; CA3 associative learning of input-place mappings; CA1 decoded localisation posteriors (place cells); and a subiculum - septum 'lostness detection' loop (see Figure 3). In a series of papers the model was extended to include online learning with a biomimetic sub-theta cycle after-depolarisation [17], and scaled up to complete a real-world navigation task by processing visual inputs through SURF feature extraction and K-Nearest Neighbour clustering [46].

We consider decoding localisation posteriors (DG-CA3-CA1 in the model) to be a pattern completion loop, whereas lostness detection and correction (Sub-Sep) to be pattern separation. By compressing an agent's sensory history through deep generative networks instead of SURF features, and by expanding the range of inputs in the model to the full gamut of sensory modalities, we aim to develop a more complete account of synthetic autobiographical memory.



(a) (b)

Fig. 3. (a) The Bayes filter hippocampus model, adapted from Fox and Prescott [16]. See main text for a synopsys of components. (b) A schematic of the SURF visual feature extraction step, constructing feature vector inputs to the Bayes filter hippocampus model, adapted from Saul et al. [46]. We aim to replace this feature extraction step with a range of modality-specific deep generative networks.

5 Conclusions and future directions

5.1 Advantages of our approach

The Deep Learning and UC hippocampus architecture presented here, based on basic systems and predictive coding theories of brain function, has a number of advantages over previous approaches. Using DNNs to learn predictive models of sensory scenes allows the agent to capture the full explanatory power of the data. Current approaches constrain how memories are stored at design time, limiting the practical utility of that information for new tasks in the future.

Our approach leverages state of the art machine learning approaches, bringing benchmark performance in speech and image processing to new modalities and new tasks. Code for many of these methods have been optimised for execution on GPUs, allowing the exploitation of cheap high-performance computing for real-time operation on robots. In addition, the UC hippocampus model has already been shown to work on real robot platforms with real-world sensory data, which is encouraging going forward.

5.2 Implementation objectives: social interaction with an iCub robot

An ideal testbed of our approach to SAM would be to embed the model on a mobile or humanoid robot to compress, parse and store rich streams of incoming data to improve ongoing processing performance. Our particular interest is in developing robots for social interaction, both for the inherent technical challenges and strong history of SAM in this task. Building on benchmarks established with existing SAM approaches [38][37], we hope to develop a system that can recognise people, remember interactions, and adapt interactions to specific people. Initial results of this work will be presented at the conference. Extensions to other problems in autonomous robots, and task domains such as life-long learning (as in [34]), will follow in due course.

Potential difficulties of implementing our approach are many and include: securing sufficient computational resources to run the algorithms (especially the DNNs) in real time; storage considerations during development of the data compression algorithms; interfacing the output of the SAM system with motor planning and narrative reasoning, which would ultimately involve encoding the memories themselves in terms of motor-control consequences; interfacing the UC episodic memory hippocampus model with rich predictive DNNs is non trivial; and event separation which would ultimately require a hierarchical implementation - a large problem in itself but one that is an interesting current area of research e.g. [47].

An additional active area of research is to advance machine learning algorithms to better capture the rich statistical structure of the world. Standard Deep Learning approaches are based on Restricted Boltzmann machines and their variations [3]. The latent representational structure in these models are relatively simple. The relationships between representations and latent variables within the learnt hierarchy, relationships that exist in the world and the data, are not captured by these models. Deep Gaussian processes [14], a recently proposed extension of Gaussian Process Latent Variable models [23], potentially provide a method for capturing rich hierarchical statistical relationships between latent variables. Integrating a Deep Gaussian Process with our UC hippocampus could substantially increase the power of this model.

6 Acknowledgements

The authors would like to thank the EU taxpayer for funding this research (EU grant no. 612139 WYSIWYD - "What You Say Is What You Did"), and our colleagues at the Sheffield Centre for Robotics for interesting discussions that stimulated this work.

Bibliography

[1] Anderson, J.R.: Act: A simple theory of complex cognition. American Psychologist 51(4), 355 (1996)
[2] Baddeley, A.: Essentials of Human Memory (Classic Edition). Psychology Press (2013)
[3] Bengio, Y.: Learning deep architectures for ai. Foundations and Trends in Machine Learning 2(1), 1–127 (2009)
[4] Bernardo, J.M., Smith, A.F.: Bayesian theory, vol. 405. John Wiley & Sons (2009)
[5] Berntsen, D., Rubin, D.C.: Understanding autobiographical memory: Theories and approaches. Cambridge University Press (2012)
[6] Bird, C.M., Burgess, N.: The hippocampus and memory: insights from spatial processing. Nature Reviews Neuroscience 9(3), 182–194 (2008)
[7] Bryson, A.E., Denham, W.F., Dreyfus, S.E.: Optimal programming problems with inequality constraints. AIAA journal 1(11), 2544–2550 (1963)
[8] Buzsáki, G., Moser, E.I.: Memory, navigation and theta rhythm in the hippocampal-entorhinal system. Nature neuroscience 16(2), 130–138 (2013)
[9] Byrne, P., Becker, S., Burgess, N.: Remembering the past and imagining the future: a neural model of spatial memory and imagery. Psychological review 114(2), 340 (2007)
[10] Carello, C., Turvey, M.T., Kugler, P.N., Shaw, R.E.: Inadequacies of the computer metaphor. Handbook of cognitive neuroscience pp. 229–248 (1984)
[11] Clark, A.: Whatever next? predictive brains, situated agents, and the future of cognitive science. Behavioral and Brain Sciences 36(03), 181–204 (2013)
[12] Cohen, M.S., Kosslyn, S.M., Breiter, H.C., DiGirolamo, G.J., Thompson, W.L., Anderson, A., Bookheimer, S., Rosen, B.R., Belliveau, J.: Changes in cortical activity during mental rotation a mapping study using functional mri. Brain 119(1), 89–100 (1996)
[13] Conway, M.A.: Autobiographical memory: An introduction. Open University Press (1990)
[14] Damianou, A.C., Lawrence, N.D.: Deep gaussian processes. arXiv preprint arXiv:1211.0358 (2012)
[15] Daw, N., Courville, A.: The pigeon as particle filter. Advances in Neural Information Processing Systems 20, 369–376 (2008)
[16] Fox, C., Prescott, T.: Hippocampus as unitary coherent particle filter. In: Neural Networks (IJCNN), The 2010 International Joint Conference on. pp. 1–8. IEEE (2010)
[17] Fox, C., Prescott, T.: Learning in a unitary coherent hippocampus. In: Artificial Neural Networks–ICANN 2010, pp. 388–394. Springer (2010)
[18] Fox, C., Stafford, T.: Maximum utility unitary coherent perception vs. the bayesian brain. In: Proceedings of the 34th annual conference of the Cognitive Science Society (2012)
[19] Friston, K.: The free-energy principle: a unified brain theory? Nature Reviews Neuroscience 11(2), 127–138 (2010)
[20] Friston, K., Kiebel, S.: Predictive coding under the free-energy principle. Philosophical Transactions of the Royal Society B: Biological Sciences 364(1521), 1211–1221 (2009)
[21] Ghosh, V.E., Gilboa, A.: What is a memory schema? a historical perspective on current neuroscience literature. Neuropsychologia 53, 104–114 (2014)
[22] Hassabis, D., Maguire, E.A.: The construction system of the brain. Philosophical Transactions of the Royal Society B: Biological Sciences 364(1521), 1263–1271 (2009)
[23] Lawrence, N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. The Journal of Machine Learning Research 6, 1783–1816 (2005)
[24] Lengyel, M., Dayan, P.: Hippocampal contributions to control: The third way. In: Advances in neural information processing systems. pp. 889–896 (2007)

[25] Lepora, N., Fox, C., Evans, M., Diamond, M., Gurney, K., Prescott, T.: Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination. Journal of The Royal Society Interface 9(72), 1517–1528 (2012)
[26] Lisman, J., Grace, A.A., Duzel, E.: A neohebbian framework for episodic memory; role of dopamine-dependent late ltp. Trends in neurosciences 34(10), 536–547 (2011)
[27] Marr, D.: Simple memory: a theory for archicortex. Philosophical Transactions of the Royal Society B 262, 23–81 (1971)
[28] McClelland, J.L., McNaughton, B.L., O'Reilly, R.C.: Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychological review 102(3), 419 (1995)
[29] Miller, G.A.: The cognitive revolution: a historical perspective. Trends in cognitive sciences 7(3), 141–144 (2003)
[30] Milner, B., Corkin, S., Teuber, H.L.: Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of hm. Neuropsychologia 6(3), 215–234 (1968)
[31] Nakashiba, T., Cushman, J.D., Pelkey, K.A., Renaudineau, S., Buhl, D.L., McHugh, T.J., Barrera, V.R., Chittajallu, R., Iwamoto, K.S., McBain, C.J., et al.: Young dentate granule cells mediate pattern separation, whereas old granule cells facilitate pattern completion. Cell 149(1), 188–201 (2012)
[32] Neisser, U., Fivush, R.: The remembering self: Construction and accuracy in the self-narrative. No. 6, Cambridge University Press (1994)
[33] Patihis, L., Frenda, S.J., LePort, A.K., Petersen, N., Nichols, R.M., Stark, C.E., McGaugh, J.L., Loftus, E.F.: False memories in highly superior autobiographical memory individuals. Proceedings of the National Academy of Sciences 110(52), 20947–20952 (2013)
[34] Paul, R., Rus, D., Newman, P.: How was your day? online visual workspace summaries using incremental clustering in topic space. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 4058–4065. IEEE (2012)
[35] Penny, W.D., Zeidman, P., Burgess, N.: Forward and backward inference in spatial cognition. PLoS computational biology 9(12), e1003383 (2013)
[36] Petit, M., Lallée, S., Boucher, J.D., Pointeau, G., Cheminade, P., Ognibene, D., Chinellato, E., Pattacini, U., Gori, I., Martinez-Hernandez, U., et al.: The coordinating role of language in real-time multimodal learning of cooperative tasks. Autonomous Mental Development, IEEE Transactions on 5(1), 3–17 (2013)
[37] Pointeau, G., Petit, M., , Dominey, P.F.: Successive developmental levels of autobiographical memory for learning through social interaction. IEEE Transactions on Autonomous Mental Development (Forthcoming)
[38] Pointeau, G., Petit, M., Dominey, P.F.: Embodied simulation based on autobiographical memory. In: Biomimetic and Biohybrid Systems, pp. 240–250. Springer (2013)
[39] Pouget, A., Beck, J.M., Ma, W.J., Latham, P.E.: Probabilistic brains: knowns and unknowns. Nature neuroscience 16(9), 1170–1178 (2013)
[40] Preston, A.R., Eichenbaum, H.: Interplay of hippocampus and prefrontal cortex in memory. Current Biology 23(17), R764–R773 (2013)
[41] Rao, R.P., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience 2(1), 79–87 (1999)
[42] Rennó-Costa, C., Lisman, J.E., Verschure, P.F.: The mechanism of rate remapping in the dentate gyrus. Neuron 68(6), 1051–1058 (2010)
[43] Rochat, P.: Criteria for an ecological self. The self in infancy: Theory and research 112, 17 (1995)
[44] Rubin, D.C.: The basic-systems model of episodic memory. Perspectives on Psychological Science 1(4), 277–311 (2006)

[45] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. MIT Press, Cambridge, MA, USA (1988)
[46] Saul, A., Prescott, T., Fox, C.: Scaling up a boltzmann machine model of hippocampus with visual features for mobile robots. In: Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on. pp. 835–840. IEEE (2011)
[47] Schapiro, A.C., Rogers, T.T., Cordova, N.I., Turk-Browne, N.B., Botvinick, M.M.: Neural representations of events arise from temporal community structure. Nature neuroscience (2013)
[48] Simoncelli, E.P.: Vision and the statistics of the visual environment. Current opinion in Neurobiology 13(2), 144–149 (2003)
[49] Spence, J.D.: The memory palace of Matteo Ricci. Penguin Books Harmondsworth (1985)
[50] Srinivasan, M.V., Laughlin, S.B., Dubs, A.: Predictive coding: a fresh view of inhibition in the retina. Proceedings of the Royal Society of London. Series B. Biological Sciences 216(1205), 427–459 (1982)
[51] Taylor, G., Hinton, G., Roweis, S.: Modeling human motion using binary latent variables. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems. vol. 19 (2007)
[52] Tulving, E.: Elements of episodic memory. Oxford Psychology Series (1985)
[53] Wills, T.J., Lever, C., Cacucci, F., Burgess, N., O'Keefe, J.: Attractor dynamics in the hippocampal representation of the local environment. Science 308(5723), 873–876 (2005)
[54] Wood, R., Baxter, P., Belpaeme, T.: A review of long-term memory in natural and synthetic systems. Adaptive Behavior 20(2), 81–103 (2012)
[55] Yamins, D.L., Hong, H., Cadieu, C., DiCarlo, J.J.: Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. In: Advances in Neural Information Processing Systems. pp. 3093–3101 (2013)

Author's Proof

Springer
the language of science

Dear Author/Editor,

Here are the proofs of your book as well as the metadata sheets.

Metadata

- Please carefully proof read the metadata, above all the names and address.
- In case there were no abstracts for this book submitted with the manuscript, the first 10-15 lines of the first paragraph were taken. In case you want to replace these default abstracts, please submit new abstracts with your proof corrections.

Page proofs

- Please check the proofs and mark your corrections either by
  - entering your corrections online
    or
  - opening the PDF file in Adobe Acrobat and inserting your corrections using the tool "Comment and Markup"
    or
  - printing the file and marking corrections on hardcopy. Please mark all corrections in dark pen in the text and in the margin at least ¼" (6 mm) from the edge.
- You can upload your annotated PDF file or your corrected printout on our Proofing Website. In case you are not able to scan the printout, send us the corrected pages via fax.
- Please note that any changes at this stage are limited to typographical errors and serious errors of fact.
- If the figures were converted to black and white, please check that the quality of such figures is sufficient and that all references to color in any text discussing the figures is changed accordingly. If the quality of some figures is judged to be insufficient, please send an improved grayscale figure.

Author's Proof

Metadata of the chapter that will be visualized online

Book Title		Sensorimotor Integration in the Whisker System
Chapter Title		The Robot Vibrissal System: Understanding Mammalian Sensorimotor Co-ordination Through Biomimetics
Copyright		Springer Science+Business Media, LLC 2015
Corresponding Author	Prefix	Prof.
	Family name	Prescott
	Particle	
	Given name	Tony J
	Suffix	
	Degrees	MA, MSc, PhD
	Division	Department of Psychology
	Organization	University of Sheffield
	Address	Western Bank, S10 2TN Sheffield, YSS, UK
	Email	t.j.prescott@sheffield.ac.uk
Author	Prefix	Dr.
	Family name	Mitchinson
	Particle	
	Given name	Ben
	Suffix	
	Degrees	BSc, MSc, PhD
	Division	Department of Psychology
	Organization	The University of Sheffield
	Address	Western Bank, S10 2TP Sheffield, YSS, UK
	Email	b.mitchinson@sheffield.ac.uk
Author	Prefix	Dr.
	Family name	Lepora MA Hons (Cantab)
	Particle	
	Given name	Nathan F.
	Suffix	
	Degrees	MMath (Cantab), PhD (Cantab)
	Division	Department of Engineering Mathematics
	Organization	University of Bristol
	Address	BS8 1UB Bristol, UK
	Email	n.lepora@bristol.ac.uk
Author	Prefix	Dr.
	Family name	Wilson
	Particle	
	Given name	Stuart P.
	Suffix	
	Degrees	PhD, MSc, BSc
	Division	Department of Psychology
	Organization	The University of Sheffield
	Address	Western Bank, S10 2TP Sheffield, YSS, UK
	Email	S.P.Wilson@Sheffield.ac.uk
Author	Prefix	Dr.
	Family name	Anderson
	Particle	

Author's Proof

	Given name	Sean R.
	Suffix	
	Degrees	MEng, PhD
	Division	Automatic Control and Systems Engineering
	Organization	University of Sheffield
	Address	Mappin Street, S1 3JD Sheffield, UK
	Email	s.anderson@sheffield.ac.uk
Author	Prefix	Prof.
	Family name	Porrill
	Particle	
	Given name	John
	Suffix	
	Degrees	BA, PhD
	Division	Department of Psychology
	Organization	University of Sheffield
	Address	Western Bank, S10 2TN Sheffield, YSS, UK
	Email	j.porrill@sheffield.ac.uk
Author	Prefix	Emeritus Prof.
	Family name	Dean
	Particle	
	Given name	Paul
	Suffix	
	Degrees	MA, DPhil
	Division	Department of Psychology
	Organization	The University of Sheffield
	Address	Western Bank, S10 2TP Sheffield, YSS, UK
	Email	p.dean@sheffield.ac.uk
Author	Prefix	Dr.
	Family name	Fox
	Particle	
	Given name	Charles W.
	Suffix	
	Degrees	MA, MSc, DPhil
	Division	
	Organization	Sheffield Centre for Robotics
	Address	Mappin Building, Mappin Street, S1 3JD, Sheffield, UK
	Email	charles.fox@sheffield.ac.uk
Author	Prefix	Dr.
	Family name	Pearson
	Particle	
	Given name	Martin J.
	Suffix	
	Degrees	BEng, MSc, PhDPhD
	Division	Bristol Robotics Laboratory
	Organization	University of the West of England
	Address	T-Block, Frenchay Campus, S1 3JD, BS16 1QY Bristol, UK
	Email	martin.pearson@brl.ac.uk
Author	Prefix	Dr.
	Family name	Sullivan
	Particle	
	Given name	Charles

Author's Proof

	Suffix	
	Degrees	PhD
	Division	Bristol Robotics Laboratory
	Organization	University of the West of England
	Address	T-Block, Frenchay Campus, S1 3JD, BS16 1QY Bristol, UK
	Email	charlie.sullivan@brl.ac.uk
Author	Prefix	Prof.
	Family name	Pipe
	Particle	
	Given name	Anthony G.
	Suffix	
	Degrees	PhD
	Division	
	Organization	Bristol Robotics Laboratory
	Address	UWE, T-block, Coldharbour Lane, S1 3JD, BS16 1QY Bristol, SG, UK
	Email	Tony.Pipe@brl.ac.uk

Abstract

We consider the problem of sensorimotor co-ordination in mammals through the lens of vibrissal touch, and via the methodology of embodied computational neuroscience—using biomimetic robots to synthesize and investigate models of mammalian brain architecture. The chapter focuses on five major brain sub-systems and their likely role in vibrissal system function—superior colliculus, basal ganglia, somatosensory cortex, cerebellum, and hippocampus. With respect to each of these we demonstrate how embodied modelling has helped elucidate their likely function in the brain of awake behaving animals. We also demonstrate how the appropriate co-ordination of these sub-systems, with a model of brain architecture, can give rise to integrated behaviour in a life-like whiskered robot.

Keywords

Sensorimotor co-ordination - Biomimetic robot - Embodied computational neuroscience - Layered architecture

Author's Proof

Chapter 10
The Robot Vibrissal System: Understanding Mammalian Sensorimotor Co-ordination Through Biomimetics

Tony J Prescott, Ben Mitchinson, Nathan F. Lepora MA Hons (Cantab), Stuart P. Wilson, Sean R. Anderson, John Porrill, Paul Dean, Charles W. Fox, Martin J.Charles PearsonSullivan and Anthony G. Pipe

Abstract We consider the problem of sensorimotor co-ordination in mammals through the lens of vibrissal touch, and via the methodology of embodied computational neuroscience—using biomimetic robots to synthesize and investigate models of mammalian brain architecture. The chapter focuses on five major brain sub-systems and their likely role in vibrissal system function—superior colliculus, basal ganglia, somatosensory cortex, cerebellum, and hippocampus. With respect to each of these we demonstrate how embodied modelling has helped elucidate their likely function in the brain of awake behaving animals. We also demonstrate how the appropriate co-ordination of these sub-systems, with a model of brain architecture, can give rise to integrated behaviour in a life-like whiskered robot.

T. J. Prescott (✉) · J. Porrill
Department of Psychology, University of Sheffield, Western Bank, Sheffield S10 2TN, YSS, UK
e-mail: t.j.prescott@sheffield.ac.uk

J. Porrill
e-mail: j.porrill@sheffield.ac.uk

B. Mitchinson · S. P. Wilson · P. Dean
Department of Psychology, The University of Sheffield, Western Bank,
Sheffield S10 2TP, YSS, UK
e-mail: b.mitchinson@sheffield.ac.uk

S. P. Wilson
e-mail: S.P.Wilson@Sheffield.ac.uk

P. Dean
e-mail: p.dean@sheffield.ac.uk

N. F. Lepora MA Hons (Cantab)
Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, UK
e-mail: n.lepora@bristol.ac.uk

S. R. Anderson
Automatic Control and Systems Engineering, University of Sheffield,
Mappin Street, Sheffield S1 3JD, UK
e-mail: s.anderson@sheffield.ac.uk

© Springer Science+Business Media, LLC 2015

P. Krieger, A. Groh (eds.), *Sensorimotor Integration in the Whisker System,*
DOI 10.1007/978-1-4939-2975-7_10

UNCORRECTED PROOF

Book ID: 301765_1_En ChapterID: 10 Dispatch Date: 18/06/2015 ProofNo: 1

Author's Proof

Keywords Sensorimotor co-ordination · Biomimetic robot · Embodied computational neuroscience · Layered architecture

AQ1

Research on active vibrissal touch has the potential to help us understand, perhaps even rethink, many of the key computations underlying sensorimotor co-ordination in the mammalian brain.

Consider, for instance, the task of visually-guided reach and grasp which is widely studied in both humans and primates. Work in humanoid robotics might decompose this task as the following steps: (i) identify a potential target in peripheral vision based on a rapid analysis of superficial salient features (colour, shape, movement); (ii) orient to and fixate on the object using foveal vision to form an internal 3-dimensional model of the object and of its key properties (shape, size, texture, and so forth); (iii) in parallel, form a second set of representations of the position and orientation of the object in space relative to those of the body, arm, and hand; (iv) match the first, "what?", model with a variety of stored "templates" in order to determine whether this particular item is, indeed, a suitable target for reaching; (iv) apply algorithms to the computed "where?" representations of the object and body, and make use of knowledge of the kinematic and dynamic properties of the arm, hand, and digits, to determine appropriate movement trajectories; (v) execute the planned movements largely ballistically but using some sensory feedback in the final approach, to locate, enclose, and lift the object in an effective way.

Now consider the capacity of an animal such as the Etruscan shrew, the smallest living terrestrial mammal—and known to be a remarkably efficient predator—to localise, identify, and entrap an agile prey insect using only its whiskers [1]. The problem is similar in many ways to that of human (or humanoid) sensory-guided reaching. The visual periphery compares to the macrovibrissae (the longer actuated facial whiskers on either side of the snout), and the visual fovea to the microvibrissae (the shorter non-moving whiskers on the upper lip and chin) and other tactile sensory surfaces around the mouth. The orienting system, as in primate vision, is likely to have the superior colliculus at its core, and will be driven by a very rapid but coarse analysis of features in the whisker signals via a midbrain loop that co-ordinates

C. W. Fox
Sheffield Centre for Robotics, Mappin Building, Mappin Street, Sheffield S1 3JD, UK
e-mail: charles.fox@sheffield.ac.uk

M. J. Pearson · C. Sullivan
Bristol Robotics Laboratory, University of the West of England, T-Block, Frenchay Campus, Bristol BS16 1QY, UK
e-mail: martin.pearson@brl.ac.uk

C. Sullivan
e-mail: charlie.sullivan@brl.ac.uk

A. G. Pipe
Bristol Robotics Laboratory, UWE, T-block, Coldharbour Lane, Bristol BS16 1QY, SG, UK
e-mail: Tony.Pipe@brl.ac.uk

UNCORRECTED PROOF

Author's Proof

movements of the whiskers, head and trunk. Further analyses of the whisker signals, from both the macro- and micro-vibrissae will involve the somatosensory cortices, and pathways through to the temporal lobes. These will likely involve the decomposition of sensory signals into components (self-motion, object properties such as shape, texture, etc.), but may also require the reintegration of decomposed features into more complete representations of the target. Alongside determination of object properties, information about the prey animal's spatial position and orientation will also have to be computed from the same set of vibrissal deflection signals. The decision of whether to make an attack will then depend on a comparison of computed features with remembered patterns corresponding to previously successful (and unsuccessful) attacks. Whilst this process will likely involve cortical systems (including hippocampus) it will ultimately involve decision-making mechanisms in basal ganglia to decide if the template fits, and, if so, whether the attack option is appropriate right now (as opposed, say, to further approach behaviour or avoidance). Planning and execution of the strike will also involve the motor cortex, and midbrain and brainstem motor systems which, together with the cerebellum, will co-ordinate precision orienting with biting, and may use additional sensory information from the vibrissae to accurately adjust the final phases of the strike.

Despite the above similarities, however, a number of features of the shrew vibrissal system might lead us to think rather differently about this problem from the way we initially conceived our example task of human visually-guided reach.

First, rather than being able to fixate and examine the target at leisure, the animal must make do with signals from a few fleeting contacts between the vibrissal tips and a small number of unknown locations on the target [2]. Further, both the sensors themselves and target are moving rapidly, the latter with unknown direction and speed. In other words, this is a task, where information about the target is relatively sparse, and where timing and dynamics are crucial. The urgency of the required response means that the preparation of attack behaviour will likely occur alongside the processing of vibrissal signals to determine object properties—so that the former can be put into effect as soon as the weight of evidence lies in its favour. In other words, this task is perhaps more similar to the challenge faced by a batter trying to hit a moving ball in fading light—the target object suddenly and rapidly appears out of nowhere, and a successful response must be executed within a critical and narrow time window.

Second, the shrew's brain is tiny [3]. Not only must its predation behaviour be accomplished with 20,000 times fewer neurons that a human might utilise for reach and grasp, we also know from the speed of the attack (which can be as little as 80 milliseconds [4]) that the shrew achieves its goal in a far smaller number of processing steps. Whatever phases are necessary for decomposition of sensory signals and their reconstruction as object representations, these will necessarily involve a small number of processing sites each made up of relatively few neurons. The construction of complex internal models, for comparison against rich templates, looks decidedly improbable in this system. More likely, key features are rapidly extracted and mapped, across a small number of synapses, into representations of their potential for action. Indeed, the step of "representing" the prey insect itself may even

UNCORRECTED PROOF

Author's Proof

be missed out entirely. This animal may encode information about objects in its tactile world only in terms of their potential as affordances [5] to guide different forms of approach, avoidance, or consummatory behaviour. Thus this is a system in which we can explore what is the minimum amount of internal transformation and representation needed in order to support complex, sensory-guided behaviour; and in which we can discover how active sensing systems [6] merge perception into action, via closed loop control [7], without the two ever being truly separate.

Overall, then, while understanding this system will not directly answer the question of how the human brain performs reach-and-grasp, the study of vibrissal-guided behaviour could help us understand many aspects of mammalian sensorimotor control and perhaps rethink a number of assumptions based on more primate-centric analyses of brain processing.

In this chapter we consider five major brain sub-systems and their likely role in vibrissal system function—superior colliculus (SC), basal ganglia (BG), primary somatosensory cortex (S1), cerebellum, and hippocampus—bearing in mind the behavioural domain of whisker-guided predation in animals such as the shrew or rat. One of these, the somatosensory cortex, is specialised for tactile processing, but shares many aspects of its computational architecture with other areas of mammalian cortex. The remaining four (basal ganglia, cerebellum, superior colliculus and hippocampus) are more "general purpose" in the sense that they appear to have some characteristic function that operates in a similar way across different sensory modalities or motor functions but that is also tuned, in some appropriate manner, to the particular requirements of processing and control for vibrissal touch. We are therefore hopeful that the insights obtained by studying the role of each of these sub-systems in the vibrissal processing of rodent-like mammals will generalise to understanding their functional capacities in other domains too. Each of these systems is the subject of a vast neuroscientific literature that we cannot even begin to summarise here. We therefore restrict our focus to providing a brief outline of the hypothesized functional role of each system in vibrissal touch and then describe how we have investigated this from an *embodied computational neuroscience* perspective that seeks to develop and test systems-level computational models of neural circuits embedded within the control system of biomimetic robots.

One might ask why we bother to build robot models of animals and their nervous systems? One answer, suggested the neurobiologist Valentino Braitenberg [8], is that synthesis (engineering a model of a biological system) is quite different from analysis (reverse-engineering an existing biological system); thus, in building a robot model of our target animal, that mimics sufficiently some aspects of its body, brain and behaviour, we can expect to learn a good deal about the original creature. Another answer is that a robot model should allow us to conduct experiments that will help us better understand the biological system, and moreover would be impossible (or at least much more difficult) to perform in the original animal [9, 10]. Finally, neurobiological studies have shown us that the brain nuclei and circuits that process vibrissal touch signals, and that control the positioning and movement of the whiskers, form a neural architecture that is a good model of how the mammalian brain, more generally, co-ordinates sensing with action. Thus, a further reason for

UNCORRECTED PROOF

Author's Proof

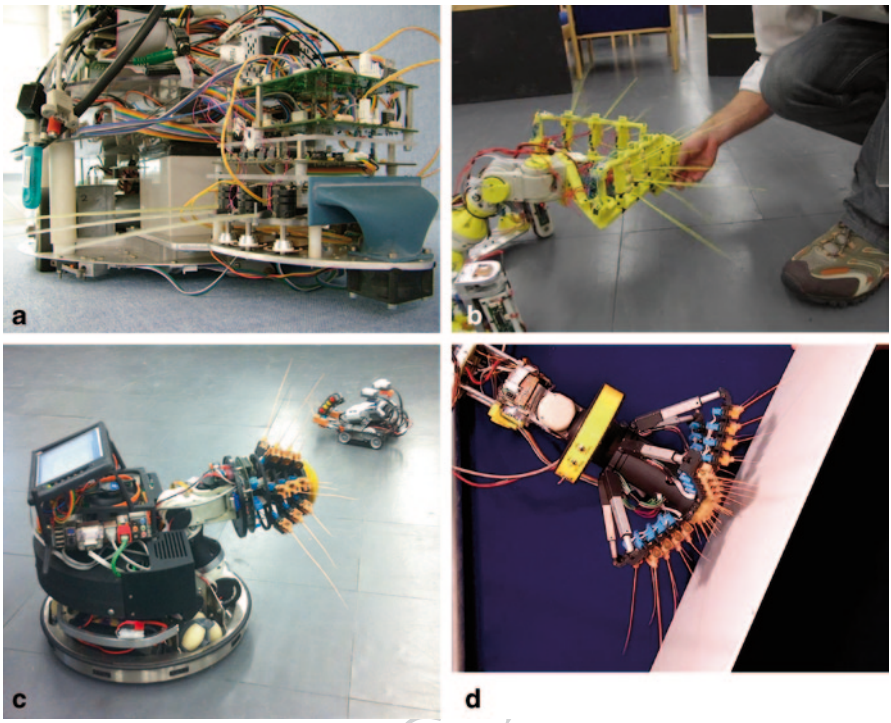


Fig. 10.1 Biomimetic whiskered robots. **a** Whiskerbot. **b** Scratchbot. **c** Shrewbot. **d** Generation 2 Biotact Sensor. Each robot has a snout configured with an array of moveable artificial whiskers. Different mechanisms have been explored for whisker actuation and for sensory transduction in the different devices. We have also gradually evolved the overall design of the whisker morphology and of the neuromimetic control architecture. The most recent model systems (Shrewbot, G2 Biotact Sensor) feature arrays of individually actuated whiskers with intrinsic transduction systems based on Hall effect sensors that can measure whisker deflection in three dimensions. For further details of the 'evolution' of our whiskered robots see [9, 103, 104]. Photos by Martin Pearson (Whiskerbot, Scratchbot, Biotact Sensor) and Tony Prescott (Shrewbot)

building biomimetic robot models is to provide improved insight into brain architecture as a whole. Indeed, by building robotic whisker systems—see examples of our whiskered robots in Figure 10.1—we consider that we are taking significant steps towards building an integrated robotic model of the mammalian brain.

A Control Architecture for Behavioural Integration in Vibrissal Touch

We begin our consideration of the sensorimotor co-ordination for vibrissal touch by addressing the overall problem of behavioural integration, or behavioural coherence, that is central to the task of building life-like systems [11]. Living, behav-

UNCORRECTED

Author's Proof

ing systems display patterns of behaviour that are integrated over space and time such that the animal controls its effector systems in a co-ordinated way, generating sequences of actions that maintain homeostatic equilibrium, satisfy drives, or meet goals. How animals achieve behavioural integration is, in general, an unsolved problem in anything other than some of the simplest invertebrates. What is clear from the perspective of behaviour is that the problem is under-constrained since similar sequences of overt behaviour can be generated by quite different underlying control architectures [12]. This implies that to understand the solution to the integration problem in any given organism is going to require investigation of mechanism in addition to observations of behaviour. In this regard, physical models—such as robots—can prove useful as a means of embodying hypotheses concerning alternative control architectures whose behavioural consequences can then be measured observationally [9]. Research with robots has repeatedly demonstrated forms of emergent behaviour [13]—the appearance of integrated behavioural sequences that are not explicitly programmed—demonstrating the value of this embodied testing for suggesting and testing candidate mechanisms.

The biological literature provides a range of different hypotheses concerning the mechanisms that can give rise to behavioural integration; here, we highlight two—behavioural and salience map competition.

The neuroethology literature suggests a decomposition of control into behavioural sub-systems that then compete to control the animal (see [14, 15] for a review). This approach has been enthusiastically adopted by researchers in behaviour-based robotics (see, e.g. [16]) as a means of generating integrated patterns of behaviour in autonomous robots that can be robust to sensory noise, or even to damage to the controller.

An alternative hypothesis emerges from the literature on spatial attention, particularly that on visual attention in primates including humans [17]. This approach suggests that actions, such as eye movements and reaches towards targets, are generated by first computing a 'salience map' that integrates information about the relevance (salience) to the animal of particular locations in space into a single topographic representation. Some maximisation algorithm is then used to select the most salient position in space towards which action is then directed. It is usual in this literature to distinguish between the computation of the salience map, the selection of the target within the map, and orienting actions that move the animal, or its effector systems, towards the target. In the mammalian brain these different functions may be supported by distinct (though overlapping) neural mechanisms [18].

Of course, the approaches of behavioural competition and salience map competition are not mutually exclusive and it is possible to imagine various hierarchical schemes, whereby, for instance, a behaviour is selected first and then a point in space to which the behaviour will be directed. Alternatively, the target location might be selected and then the action to be directed at it. Finally, parallel, interacting sub-systems may simultaneously converge on both a target and suitable action [19]. We recently investigated the hypothesis that a salience map model can be used to generate action sequences for a biomimetic whiskered robot snout mounted on a mobile robot platform, and compared this with an earlier control model based on

UNCORRECTED PROOF

Author's Proof

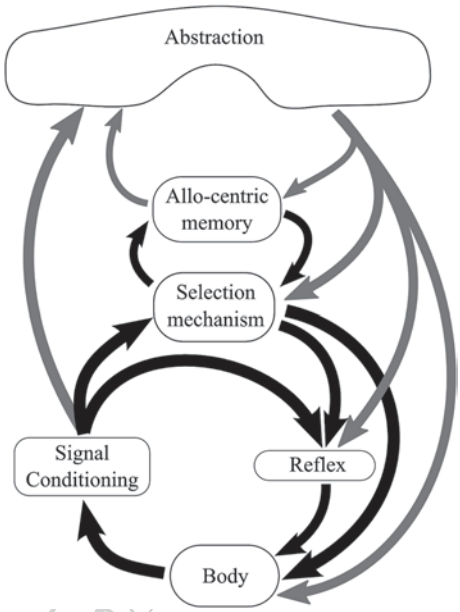


Fig. 10.2 Model of brain architecture for control of a whiskered mobile robot. The abstract components of the model can be mapped to key sub-systems in the mammalian brain (see text). Figure reproduced from Fig. 2 of [13] with permission from Springer

behaviour selection [13]. Both control systems generated life-like sequences which alternate between exploration and orienting behaviour, but in the salience map version these higher-level behavioural 'bouts' were an emergent consequence of actions determined by following a shifting focus of spatial attention (determined by a salience map) rather than resulting from the alternation of distinct behavioural primitives.

In the mammalian brain, sensorimotor loops involving the cortex, superior colliculus, basal ganglia, cerebellum and hippocampus may interact to implement a control system similar to this hypothesised salience map model. Figure 10.2 summarises the multi-level loop architecture used in our recent biomimetic robot Shrewbot, which is derived from our general understanding of the control architecture of the rat vibrissal system. We cannot represent the whole brain in our model from the outset, and there is no general agreement on the function of some of the neural centres. Since the robot must generate behaviour if we are to experiment with it, our breakdown of the control system into modules is by function, but the particular breakdown chosen is deeply inspired by our understanding of brain anatomy. This places us in a strong position to hypothesise relationships between structure and function in the neural system, and these hypotheses are a major outcome of our robot work. Here, the component 'selection mechanism', modelled on the mammalian basal ganglia and superior colliculus, is responsible for selecting and driving the majority of movements of the robot's body (neck and wheels). Below this system, motor systems implement control commands, and low-level reflex loops support some rapid responses to current conditions (for instance, whisker protraction is inhibited by contact with the environment [20]). Above this system,

UNCORRECTED

Author's Proof

we are beginning to add more cognitive components that modulate selection. The component labelled 'abstraction' [21] gleans additional information about what has been contacted by the vibrissal sensors in a manner analogous to processing centres such as the somatosensory cortex. Elsewhere, the component labelled 'allocentric memory' retains a memory of the robot's past spatial experience and thus models some of the functionality of the mammalian hippocampal system [22, 23]. 'Signal conditioning' indicates the importance of early processing of sensory signals to, for instance, distinguish components of the signal that may be due to the organism's (or robot's) own movement rather than to contact with the external world. Some neurobiological evidence, and our own modelling work, suggest an important role for the cerebellum in this regard [24, 25].

Orienting the Tactile Fovea with the Superior Colliculus

To demonstrate the capacity of this architecture for generating integrated behaviour we have focused on the problem of orienting to interesting or novel stimuli detected by the robot vibrissae. To develop our model of orienting we first assume a 'tactile fovea' [26], as the region of the snout with the highest density of microvibrissae, and focus on the key component of orienting behaviour in rodent-like mammals of bringing the fovea to a target. For instance, when faced with a task of discriminating between multiple objects, rat behaviour can be described as foveation (targeting the sensory fovea) to each discriminandum in sequence [26]. In our control architecture, then, the selection mechanism thus drives movements of the fovea with its output being the desired instantaneous velocity of the foveal position. In this model the movement of the remaining nodes of the animal/robot are unconstrained at the level of the selection mechanism and, instead, are determined at the level of the body (i.e. the motor system). Specifically, nodes such as the neck joints, and body are 'enslaved' to the fovea, and move so as to carry the fovea towards its target as smoothly and directly as possible. One could say that our robots are 'led by the nose'. This is, of course, a simplification of biological behaviour, though we have been surprised by how life-like (and practical) the resulting behaviour can be.

In primates, foveation is well studied with respect to the visual system and is known to be mediated by the Superior Colliculus (SC) [17]. In rats, stimulation of SC can evoke not only eye movements [27], but also orienting-like movements of the snout, circling, and even locomotion [28]. Recent neurobehavioural evidence also directly implicates SC as having a major role in rodent prey capture [29]. More generally, the SC appears to be a very plausible location to integrate tactile signals for spatial attention. It has the right inputs with signals arriving from the vibrissae via the from trigeminal sensory complex [30], and with further inputs converging from several relevant areas of cortex including S1 [30–32]. The organization of the SC is topographic in both its sensory and motor aspects, with a sensory topography appropriate to encoding a salience map centred on the foveal region of the snout [33, 34] and motor maps suitably configured to generate orienting head movements

UNCORRECTED PROOF

Author's Proof

[28]. Inspired by these facts, we have developed a model of foveal velocity vector generation that mirrors the features of SC—that is, we employ a topographic saliency map driven by sensory input and modulated by information from mid- and upper-brain, with a simple motor output transform that drives foveation to the most salient region of local space [13]. In the case of our robots, salience is excited by whisker contact and endogenous noise and suppressed by a top-down 'inhibition-of-return' signal from an allocentric memory component that lowers the salience of regions that have recently been foveated. The selection task, then, is to choose between foveation targets in nearby space.

This saliency map model of tactile attention has recently been extended to incorporate the regulation of vibrissal movement [35]. To evaluate the model we tested it within a simulated two-dimensional environment containing configurable 'obstacles', under conditions analogous to those used in behavioural experiments [36–39], and showed that it exhibits many of the modulations of whisker movement previously reported and summarised in Fig. 10.3 [35]. The model can also account for anticipatory aspects of active vibrissal control (see e.g. [36, 40]) that cannot be the outcome of purely reflexive mechanisms. Here again the SC is implicated as a key sub-system in the rodent brain. Stimulation of SC can generate modulatory (non-periodic) whisker movements [41] suggesting a role in determining the protraction amplitude of the whiskers. Accordingly, SC outputs directly target the facial nucleus which is the motor nucleus that drives the whisker musculature [42]. The receptive fields of SC neurons that are sensitive to deflection of single macrovibrissae are large and overlapping under anaesthesia [34]. Since the whiskers sweep back and forth during exploration this raises the possibility that, in the awake behaving animal, vibrissal receptive fields in SC are actually sharply tuned, but encode target locations in a head-centred spatial map (that might be contacted by moving whiskers) rather than contacts on distinct macrovibrissae *per se*.

The Role of the Cortex and Basal Ganglia in Decision-Making

Whilst the SC provides a mechanism that can control the orienting movements of the head and sensory systems, it is only one of many structures involved in identifying and *selecting* targets for foveation. Studies in primates implicate sensory processing in cortical areas [43] coupled with action selection in the basal ganglia (BG) [15, 44, 45] as critical substrates for the decision-making aspects of target selection. Our research with whiskered robots is helping us to analyse the contributions of these different neural systems to perceptual decision making in the mammalian brain.

The last two decades have seen major advances in our understanding of decision making as statistically optimal inference from noisy and ambiguous sensations using Bayesian probability theory [46]. A Bayesian approach to the task of classification involves recording the likelihoods of measurements from example sensory

UNCORRECTED PROOF

Author's Proof

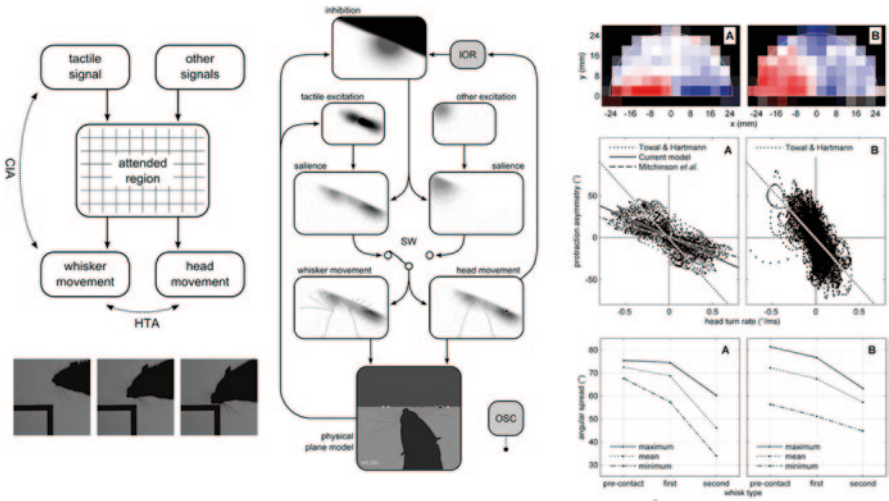


Fig. 10.3 Model of the regulation of whisking behaviour by spatial attention. *Top-left.* A mix of exogenous (*tactile*) and endogenous (*other*) influences affect the locus/region of spatial attention. This locus drives head movements and is responsible for the modulation of whisker movements from whisk to whisk. *Bottom-left.* Three frames from an overhead video of a rat executing an orient to an unexpected contact. *Centre.* The implementation used to test the model—additional components are a simple oscillator to generate periodic whisking (*OSC*), an implementation of inhibition-of-return (*IOR*) to generate sufficiently rich orienting behaviour for testing, and a physical model of whisker/environment interactions. *Right.* Comparison of results from current model **a** and recordings of rat behaviour, **b** under three analyses, from top: Contact-induced asymmetry (see, e.g. [37])—if an animal approaches a surface at an oblique angle then protraction of whiskers ipsilateral to the surface is reduced (*red color/dark shading inside boundary*), whilst protraction of whiskers on the contralateral side is increased (blue color/light shading); Head-turning asymmetry (see, e.g. [39])—as an animal turns the whiskers typically move asymmetrically as if to anticipate obstacles in the direction of the turn; Spread reduction during contact (see. e.g. [36])—whilst exploring a surface the whiskers are brought closer together with the effect of increasing the number of whisker-surface contacts. (Adapted from Figs. 2, 3, 7 and 8 of [35] which should be consulted for further explanation of the model and results)

UNCORRECTED

data. Given new test data, these likelihoods can be used with Bayes' rule to calculate the posterior probability of the test data being drawn from each trained class, that is, the likelihood of the data belonging to any given category given the history of past data. Within the broader class of Bayesian classifiers the approach of *sequential analysis* [47] operates by applying Bayes' rule repeatedly to accumulating evidence for competing perceptual hypotheses, derived from time series of sensory data, until a preset threshold is reached. This method can be likened to a process that has been observed in parietal cortex when monkeys are required to make perceptual judgements about visual motion direction and where individual neurons have been recorded that noisily ramp-up their firing rates until reaching a decision threshold [48–50].

Using our whiskered robots, we have explored the possibility that sequential Bayes can provide an effective general classifier for object properties detected

Author's Proof

using vibrissal sensors. Examining characteristics such as texture, radial distance to contact, speed of object movement and novelty, and using a range of robot platforms deploying different strategies for the control of whisker movement and position, we have shown that sequential Bayes is reliable, accurate (hit rates of >90% on several tasks), and out-performs a number of competing classification methods such as spectral templates, maximum likelihood, and multi-layer neural networks [21, 51–54]. That a classification method that matches with primate data can operate effectively with signals from artificial whiskers gives hope that a single theory of perceptual decision-making can be developed that will apply equally to primate vision and rodent vibrissal touch. The further implication of these studies is that a common memory format (log likelihoods) could be used to encode tactile memories for object properties. These models also have the potential to address questions about the nature of tactile memory in animals such as the Etruscan shrew since the ability to classify objects (for instance, as prey items that can be attached) requires efficient and compact memory traces, and the ability to make timely and appropriate decisions based on accumulating evidence.

Alongside evidence that cortex accumulates evidence for competing hypotheses, converging evidence from neurobiology and computational modelling, is showing that the BG anatomy maps onto a network implementation of an optimal statistical method for hypothesis testing that provides for timely and efficient selection of an appropriate response [55–57]. As noted above, in the sensory component of this process, evidence for the alternative interpretations of a stimulus accumulates in neuronal "evidence bins" (e.g., that a visual stimulus is moving right rather than left) and this accumulated evidence competes within the BG to elicit an action (e.g., press right lever or left lever). In the vibrissal system, the substantial projections from layer 5a of S1 cortex to the striatum—the major input structure of the BG—could provide a neural substrate for decision-making in relation to tactile object properties. We have used insights from experimental data, and from recent recordings in cortical areas during decision-making tasks (e.g. [58]), to revise and extend existing primate-based computational models of the decision-making process [52, 57] and are in the process of exploring the implications of these revisions for decision-making in embodied robotic models.

The Vibrissal Somatosensory Cortices—Feature Maps for Detecting Behavioural Affordances

Within the parietal lobes of all mammals there are localised cortical areas that are more specialised towards particular sensory modalities—such as somatosensation, vision, audition and vestibular sensing—and other areas that are more multisensory in nature. In vibrissal specialists, such as rats, mice and shrews, the somatosensory areas devoted to the region of the snout are massively expanded compared to those of most mammals. In these species, multiple somatotopic maps of the body have been identified, the principal ones being labelled primary and secondary somatosensory

UNCORRECTED PROOF

Author's Proof

cortex (S1 and S2). Whilst both of these areas have large domains devoted to the vibrissae, barrel-like aggregates of neurons ("barrels") have been identified only in S1. The size of the area of cortex devoted to the large macrovibrisae appears to reflect the high innervation levels of the whisker follicles (see also discussion of cortical area size in [59]). In the mouse, S1 cortex represents approximately 13 % of the cortical surface area in total and 69 % of the somatosensory cortex [60].

In rats and mice, S1 barrels exist for both the large and motile macrovibrissae in the posteriomedial barrel field (PMBF) and for the smaller non-actuated microvibrissae in the anterolateral barrel field (ALBF) [61], however, research has almost entirely focused on the larger barrels found in PMBF because of their ease of stimulation via the macrovibrissae. S1 and S2 are reciprocally connected with each other and also, via the corpus collosum, with their contralateral other halves [62]. S1 is also connected with a number of other cortical regions including the motor and perirhinal cortices. Other major S1 projection areas include the thalamic areas from which it receives input (VPM and POm), the reticular nucleus of the thalamus, and, of particular interest here, sub-cortical targets in the basal ganglia, pontine nuclei (cerebellum), and superior colliculus [62].

For a vibrissal-specialist like the Etruscan shrew, successful prey capture is critically dependent on accurate and rapid detection of tactile stimulus velocity. This leads to the general question of how the brain might extract movement direction and speed from patterns of vibrissal deflection. Since it was first proposed, Jeffress' place theory [63] has been a dominant model for understanding how sensory motion is encoded in the brain [64]. The idea is that coincidence detector neurons receive input from sensors after delays governed by the distance of each neuron from the corresponding signal sources. The inter-sensor time difference is then encoded by the location of neurons that are active because their connection delays exactly compensate the inter-sensor stimulation interval. The place theory therefore suggests an important role for neural geometry in computing the motion of sensory stimuli. Despite being a general theory of neural computation, most of the evidence for the place theory is provided by studies of the auditory system of auditory specialists such as the barn owl. The evidence from studies of mammalian auditory systems is inconclusive, for example, rabbit auditory cortex neurons are tuned to much longer inter-aural delays than can be accounted for by known axonal connection velocities [65], and evidence from other sensory modalities is sparse.

In order to provide a further test of the generality of the place theory, we sought to apply it to a model of tactile stimulus processing in rodent barrel cortex [66]. We asked whether model cortical neurons receiving synaptic inputs via delays governed by realistic connection geometry and plausible axonal propagation speeds would match the range of real responses to paired stimulation of adjacent whiskers. Validating this hypothesis we recreated, in simulation, the broad range of spiking patterns displayed by layer 2/3 barrel cortex neurons when adjacent whiskers are deflected through the range of inter-stimulus intervals, as measured electrophysiologically by Shimegi et al. [67]. These biological experiments have shown that, when

UNCORRECTED PROOF

Author's Proof

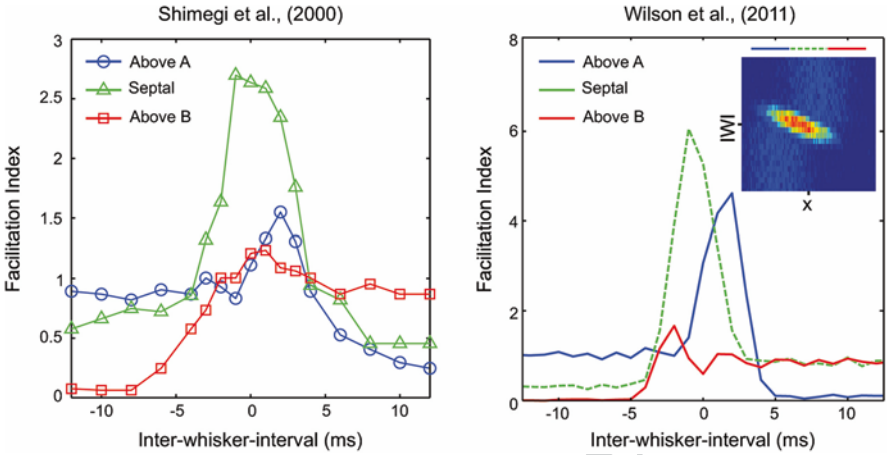


Fig. 10.4 Spatiotemporal interactions of cortical responses to paired whisker stimulations. The *left* panel shows experimental data replotted from Fig. 8e of [67] with permission from Society for Neuroscience. The *right* panel shows the behaviour of our model [66]. In both panels, the response facilitation index—computed as the ratio between the response to a paired stimulation of adjacent whiskers A and B and the linear sum of the responses to either A or B separately—is shown as a function of the time interval that separated the two whisker deflections. Figure reproduced from [66]

two adjacent whiskers are stimulated in a sequence with a few milliseconds interval, the responses of cortical neurons depends strongly on their positions (whether closer to the barrel of the first or second whisker), and are typically stronger than the sum of the responses to independent whisker deflections for a specific time interval (Fig. 10.4 left). Our modelling results (Fig. 10.4 right) showed—consistent with a place theory interpretation—that this broad range of recorded response profiles emerges naturally from the connection geometry as a function of the anatomical location of the neuron. In practical terms, the result that stimulus-evoked responses can be predicted by neuron location is important because it suggests that neural geometry needs special consideration as we construct theories of cortical processing.

Further consideration of the role of neural geometry may lead to predictions about sensory processing in species that maintain map-like representations compared with those that do not. For example, whilst individual neurons in rodent primary visual cortex respond selectively for the orientation of visual edges, they are arranged randomly in the cortex with respect to their orientation preference [68]. Presumably rodents would therefore be poor at using map-dependent mechanisms to extract stimulus velocity in the orientation domain (i.e., when extracting information about image rotation), compared to primates that have smooth topological maps for orientation preference and so might use place-coding mechanisms. In more general terms, evidence supporting the place theory from a tactile mammalian sensory system provides new insight into understanding how the brain represents

UNCORRECTED

Author's Proof

moving sensory input. Finally, in the context of the vibrissal system, and the barrel cortex, this study provided a novel account of the fusion of information at the multi-whisker level that both explained existing data, by casting it within a general and powerful framework (place theory), and made testable predictions that could be investigated experimentally.

A key feature of the mammalian sensory cortices is the presence of self-organising topological maps. Cortical maps for features of each sensory modality can be highly plastic and shaped by a combination of physical and environmental constraints [69, 70]. We recently conducted a series of experiments driving map self-organisation with activity patterns representing tactile stimulation of an array of artificial whiskers, in order to predict the organisation of object representations in the somatosensory cortex [71]. Inputs to the model were patterns of activity in simulated layer 4, encoding the spatial location and direction of whisker deflections caused by tactile stimuli that varied in shape, direction and speed. Layer 4 activity patterns were then remapped as layer 2/3 activity patterns using distance-dependent signalling delays in the layer 4 to layer 2/3 projection, to additionally encode the relative timing of whisker deflections [66]. This model represents a biologically grounded method by which to map the full spatial-temporal pattern of multi-whisker inputs to an essentially spatial representation of the stimulus across layer 2/3. Layer 2/3 activity patterns representing the range of multi-whisker stimulus patterns could thus be presented to a self-organising map model of postnatal development in layer 5, using an approach that we have shown previously to recreate known topological feature maps in layer 2/3 [72].

Our model of layer 5 map self-organisation, like our previous model for layer 2/3 map self-organisation, is based on the LISSOM (Laterally Interconnected Synergetically Self-organising Map) algorithm originally developed to capture the self-organising properties of primate visual cortex [73]. In our barrel cortex model, responses across the cortical sheet became organised into coextensive topological maps, wherein iso-feature contours for tactile stimulus shape, direction, and speed preferences intersected at right angles (see Fig. 10.5). The model therefore makes the critical prediction that orthogonal tactile feature spaces are represented in the somatosensory cortex by orthogonal feature maps (and hence by orthogonal spatial codes). A series of controlled simulation experiments suggested further that (i) speed and shape selective neurons align to regions of low selectivity in and between direction pinwheels, (ii) direction, shape, and speed maps are acquired in developmental sequence, (iii) stimulus direction is resolved by afferent projections to layer 5, whereas shape and speed are resolved by subsequent recurrent interactions in layer 5. These findings constitute specific, testable predictions about the development of functional maps and object representations in somatosensory cortex, i.e., that maps for the tactile motion direction implied by multi-whisker deflection sequences emerge earlier and more robustly than lower-order feature maps representing e.g., stimulus shape and speed. These modelling predictions were validated in experiments that connected self-organising networks to an artificial sensor array stimulated by a table-top positioning robot [71].

UNCORRECTED PROOF

Author's Proof

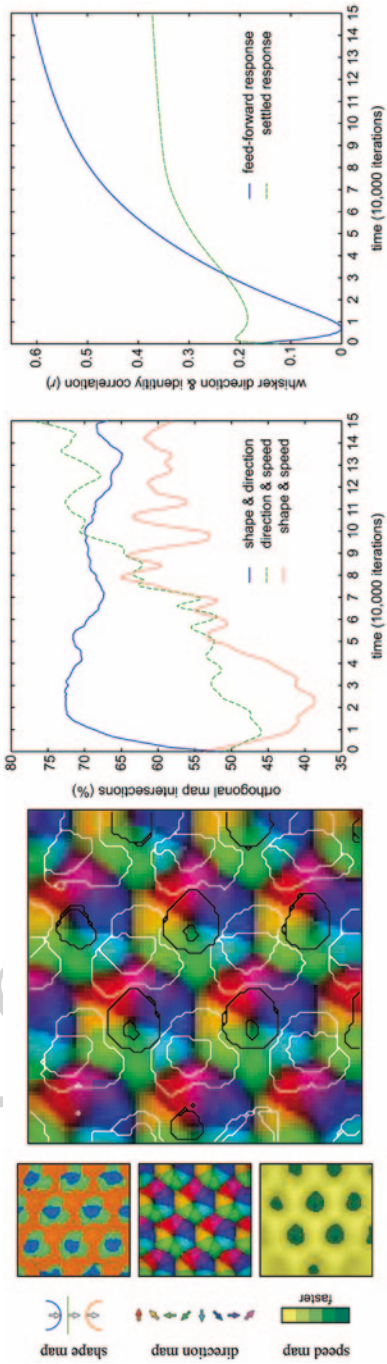


Fig. 10.5 The emergence of orthogonal coding for tactile stimuli of different shapes (concavity/convexity), directions, and speeds, in a self-organising model of map development in the barrel cortex. See [71] for further details

UNCORRECTED PROOF

Author's Proof

The Cerebellum Viewed as an Adaptive Filter and Forward Model

In experiments with the Scratchbot robot platform we occasionally observed that the robot would orient (foveate) as if to a target when no object is in fact present. On further investigation it appeared that on these occasions the sensory signal generated by active whisking is wrongly interpreted as contact with a target. This empirical observation in our biomimetic robot contrasts with the lack of reports of such 'phantom' orienting in normal rats. On the other hand, it has been shown that in rats sensory signals *are* generated by whisking movements. Specifically, a study by Leiser and Moxon reported that trigeminal ganglion cells of the rat fired during active whisking in air with no object contacts but were silent when the whiskers were at rest [74]. The implication of this result is that whisker sensory signals may include self-generated artefacts during whisking. The fact that the robot does show phantom orienting and rats appear not to suggests that, unlike the robot, rats can discriminate between the component of a sensory signal that originates from an external source and the component that is self-generated by its own whisking movements. We can ask the question: how is this discrimination achieved? This leads us to suggest that the rat may actively cancel self-generated sensory signals—what we might call 'self-induced' or 'self-generated' noise.

Interestingly, noise cancellation in biological systems has a well-investigated precedent—interference cancellation in passive electro-sensing in electric fish (for review see Bell et al. [75]). Of particular interest here is the evidence that suggests a cerebellar-like structure performs the function of noise cancellation in these animals [75, 76]. Additionally, this cerebellar-like structure is thought to act analogously to an adaptive filter [77, 78], linking biological noise cancellation to both the signal processing literature [79] as well as the adaptive filter model of the cerebellum [80]. In humans, a similar capacity to predict or cancel self-induced sensory signals is indicated by our inability to entertain ourselves by self-tickling (as opposed to be tickled by someone else). In this case, functional MRI data [81] also indicates a role for the cerebellum in predicting sensory signals due to self-movement thereby making them seem less amusing!

The above considerations led us to the hypothesis that rats may use their cerebellum to generate a signal which cancels the effects of self-generated whisking noise on incoming sensory signals from the whiskers (see [24, 82] for a review of the wider literature on the role of the cerebellum in sensory noise cancellation). Our first step in investigating this hypothesis was to determine whether our proposed mechanism would work *in principle*, by using it to achieve noise cancellation in a whisking robot [24]. The step had two goals, first to solve the practical problem of noise cancellation in the robot, and secondly to provide a theoretical basis for studying noise cancellation in whisking animals. Our approach was to use inspiration from the signal processing literature to form a prototype hypothesis of a whisking noise cancellation scheme.

UNCORRECTED PROOF

Author's Proof

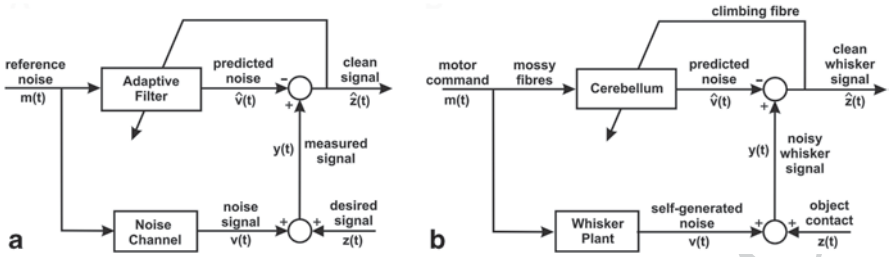


Fig. 10.6 Noise cancellation schemes. **a** A generic adaptive noise cancellation scheme, see for instance Widrow and Stearns [79], **b** A proposed biological whisking noise cancellation scheme. See [24] for further details

The subject of noise cancellation has been studied in the signal processing literature since the 1960s (for early references see Widrow et al. [79]). Much of the formative work was conducted by Bernard Widrow and was linked as an application problem to the least-mean-squares (LMS) adaptive filtering algorithm. The generic noise cancellation scheme is illustrated in Fig. 10.6, for detailed explanation see [24]. The weights of the adaptive filter in the noise cancellation scheme (Fig. 10.6a) are adjusted by removing the correlations in the clean signal from the reference noise, implemented via the LMS rule. In the context of whisking, the self-generated noise is thought to be caused by the movement of the whisker, either by inertia of the whisker base in the follicle or the whisker musculature pressing and activating the mechanoreceptors [74]. Ultimately, this activation is caused by the motor command to the whisker plant. Hence, we regard the motor command (either high- or low-level) as the reference noise, which is correlated with the noise signal but uncorrelated with signals related to object contacts. In our proposed whisking noise cancellation scheme (illustrated in Fig. 10.6b) the cerebellum learns to predict self-generated noise from the motor commands that cause the whisker movements. Hence, the cerebellum learns an internal forward model of the whisker dynamics that transform motor commands into sensory signals.

Adaptive Filter Model of the Cerebellum In the above whisking noise cancellation scheme we use the adaptive-filter to computationally model the cerebellum, as originally proposed by Fujita [80]. The mapping of this scheme onto the cerebellar microcircuit is illustrated in Fig. 10.7. We have previously investigated the computational properties of this model for adaptive motor control [83–86], and others have proposed that it could be used in principle to learn forward models (see [82] for review). However, our vibrissal noise study was, to our knowledge, the first instance of the adaptive-filter model of the cerebellum being applied to learning a specific forward model (i.e. of whisker dynamics) for the purposes of noise cancellation.

To develop and validate our proposed whisking noise cancellation scheme we recorded experimental data from Scratchbot during 'free' whisking (i.e. with no object contacts). Note that free-whisking is an ideal scenario to test the noise cancellation scheme because during free whisking the whisker sensory signal should be zero.

UNCORRECTED PROOF

Author's Proof

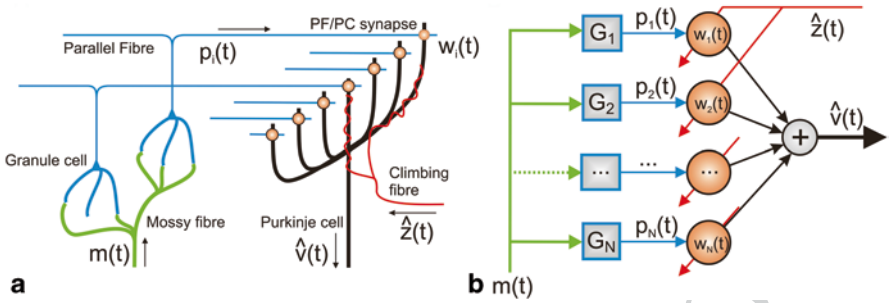


Fig. 10.7 Schematic diagram of the organization of the cerebellar microcircuit and its interpretation as an adaptive linear filter. **a** Simplified architecture of cerebellar cortex, **b** Adaptive filter model of the cerebellum. (Adapted from Fig. 1a, b of [82] with permission from Nature Publishing Group)

UNCORRECTED PROOF

Hence, whilst the whisker dynamics are unknown and therefore the optimal cerebellar filter is also unknown, the output of the cancellation scheme is known: it should be zero. Therefore it is straight-forward to evaluate the performance of the noise cancellation scheme during free-whisking. Figure 10.8 shows example results of the application of the cerebellar noise cancellation algorithm to free-whisking data.

In a further extension of our sensory noise cancellation model [25] we have shown that the addition of sensory information from the whiskers allows the adaptive filter to learn a more complex internal model that performs more robustly than a forward model based on efference copy signals alone, particularly when the whisking-induced interference has a periodic structure. More generally, our analysis of the whisking noise cancellation scheme reveals that the functional role of the cerebellum may be to learn a forward model of the whisker/follicle dynamics. This links to separate speculation over the functional role of the cerebellum in motor control and sensory processing, where it has been suggested that the cerebellum can learn a variety of forward and inverse models in control and state estimation tasks, see for instance [87, 88]. Our development of the whisking noise cancellation scheme from a theoretical basis has led to a number of experimental predictions relating to the functionality of different components of the cerebellar micro-circuit: (i) that the mossy fibres transmit a copy of motor command, (ii) that the Purkinje cell output is an estimate of the self-induced noise signal, (iii) that the climbing fibre teaching signal is an estimate of the 'clean' whisker sensory signal, and (iv) that the superior colliculus is the target of the cerebellar output and acts to compare predicted and actual sensory signals [25].

Cerebellar/Collicular Algorithms for Orienting and Predictive Pursuit Cerebellar circuits are likely to be important for fast predictive control of ballistic movements needed for tasks such as prey tracking and capture since cerebellar damage is known to impair predictive aspects of motor behaviour [89]. An important role might lie in the calibration of sensory maps used to generate fast orienting movements. We have hypothesised [90] that the known extensive cerebellar-collicular connectivity (see

Author's Proof

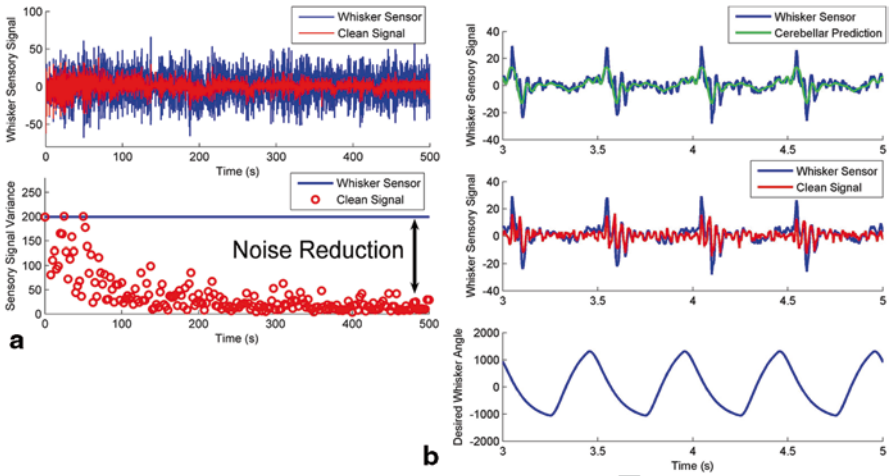


Fig. 10.8 Results from applying the noise cancellation algorithm to the free-whisking sensory signal. **a** Low frequency linear noise cancellation in the range 0–5 Hz. **b** High frequency nonlinear noise cancellation (up to 100 Hz). See [24] for further details

[25, 91]), together with the adaptive filter cerebellar architecture described above, could play a role in calibrating predictive topographic maps in the colliculus. We are currently investigating how this model can be employed in a predictive architecture in which features appear in the salience map at their predicted rather than their current positions.

The cerebellum may also calibrate sensory information that provides input to the predictive system. For example, as described above, we have developed a cortical algorithm for estimating contact timing for a target moving through the robotic whisker array illustrated in Fig. 10.9 left [71]. Figure 10.9 (centre) shows target velocities recovered from these timings, whilst incorporating a cerebellar learning element produced a more accurate calibration as shown in Fig. 10.9 (right).

Tactile Self-Localisation and Mapping in the Hippocampus

The lifestyle of any small mammal, even one as tiny as the Etruscan shrew, requires the capacity to know where you are at all times with respect to key locations such as the nest, important feeding sites, and significant danger zones. Indeed, as we have seen above, simply to explore space efficiently using vibrissal touch requires some long-term memory of locations you have visited in the recent past, and the capacity to update an estimate of your own position as you move around. In the mammalian brain the hippocampal system is known to be important in building and maintaining representations of the environment (the 'place cell' system [23]) and in maintaining estimates of changes in position determined through path integration (the 'grid cell'

UNCORRECTED

Author's Proof

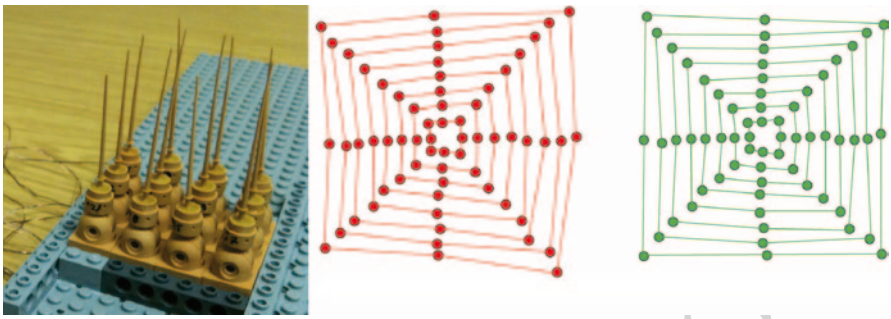


Fig. 10.9 Cortical algorithm for contact timing. **a** A planar target is moved through a robotic whisker array on an xy-plotter through eight speeds and eight directions to generate a set of multi-whisker deflection patterns, **b** plot of target velocities recovered from the relative timing of whisker responses, as computed using a cortical velocity-encoding algorithm based on [71]; plotted against the x and y components of the stimulus motion velocity, the distortion of the cortical estimates compared to a regular grid of true velocities is clear, **c** estimation of the motion velocities are improved by cerebellar correction after randomised representation of the data set

system [92]). Recent data also demonstrates that the hippocampus also encodes tactile information that describe the environmental context obtained through vibrissal touch [93].

The principal input structures of the hippocampus are the superficial layers of Entorhinal Cortex (EC). EC projects to Dentate Gyrus (DG) which is believed to increase the sparcity of the encoding generated by the EC. Both EC and DG project to CA3, which also receives strong recurrent connections that are disabled [94] by septal acetylcholine (ACh). CA3 and EC project to CA1, which in turn projects to the deep layers of Entorhinal cortex, there is also a back-projection from CA3 to DG [95]. Although the classical view of hippocampus is as a single loop, there is also a second loop—EC and CA1 project to Subiculum (Sub), which projects to the midbrain Septum (Sep) via fornix. Septal ACh and GABA fibres then project back to all parts of hippocampus. Figure 10.10 summarises many of these connections.

There have been two broad schools of hippocampal modelling one based acquiring spatial sequences, and the other on the notion auto-associative memory including pattern reconstruction based on partial or noisy input (see [96] for review). However, the objectives of both auto-associative and spatial sequence memories can be combined by a general Bayesian filter with noisy observations, which seeks infers the (hidden) state of the world. Such a filter that maintains just a single estimate of the current state-of-the-world (e.g. of location in spatial map) is known as a 'unitary particle filter'. We have developed a model of spatial learning in the rodent hippocampus [96], viewed as a unitary particle filter, by mapping key structures in the hippocampal system onto the components of a Temporal Restricted Boltzmann Machine—a probabilistic algorithm for learning sequence data developed by researchers in machine learning (see, e.g. [97]). The algorithm approximates Bayesian filtering to infer both auto-associative de-noised percepts and temporal sequences, that is, it can clean-up and fill-out incoming sensory patterns and can use these to

UNCORRECTED PROOF

Author's Proof

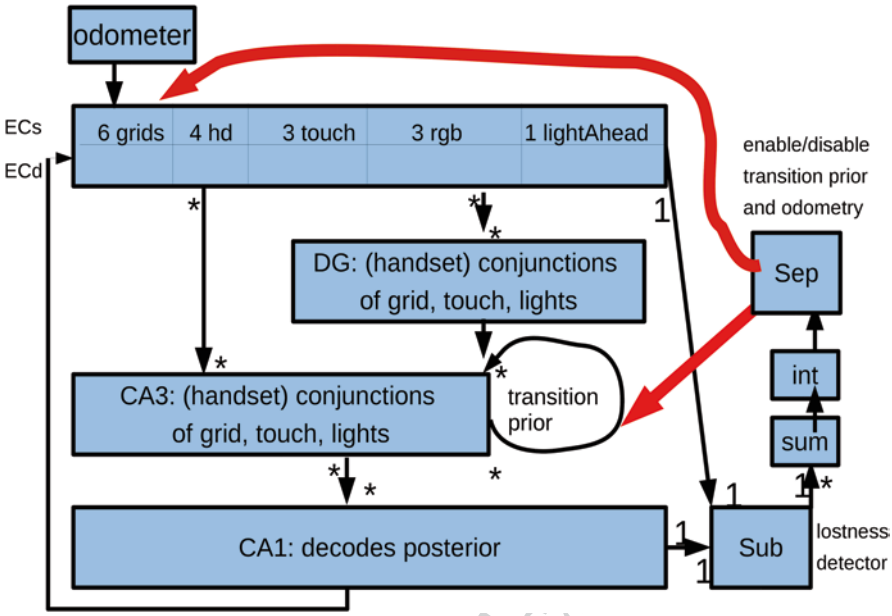


Fig. 10.10 Model of spatial memory in the hippocampal system viewed as a particle filter. Structures are labelled with UML notation indicating many-to-many fully connected links (*→ *), one-to-one links (*1* → *1*) and many-to-one links (* → *1*). Thick lines are ACh projections, thin lines are glutamate. The model implements a spatial memory system for location based on multisensory signals from tactile, visual, and path integration signals. See text for abbreviations showing the proposed mapping to regions of the rodent hippocampal system. Reproduced from Fig. 1 of [96] with permission from IEEE

recall or forecast sequences of places visited during navigation. The mapping to the hippocampal system (see Fig. 10.10) proposes a novel role for the subiculum, and for ACh from the septum, in detecting when the animal has become lost (by detecting a mismatch between predicted and actual sensory signals). A follow-up paper [98] extended this model to include online learning of connections to and from the simulated hippocampal CA3 region.

Building on our computational models of hippocampus, we have developed tactile Self-Localisation and Mapping (tSLAM) for whiskered mobile robot platforms. tSLAM provides a robot with a means of mapping and navigating a novel environment by touch information alone, something which has never previously been developed in robotics. A critical step, was the development of a hierarchical Bayesian 'blackboard' architecture [99] to investigate how to fuse information from multiple local tactile feature reports to recognise objects in the world. This work also involved developing techniques for online head-centric spatial localisation of whisker contacts, and their subsequent world-centric transformation. To achieve tSLAM we have developed a particle-filter based mapping and localisation algorithm, taking odometry (path integration) and tactile information from the robot in real-time. This information is then integrated into an occupancy grid map, and a current position estimate.

UNCORRECTED

Author's Proof

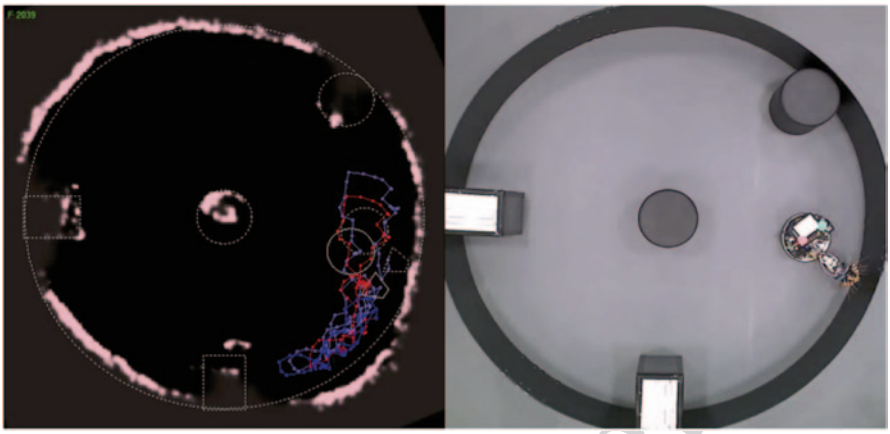


Fig. 10.11 Tactile Self-localisation and Mapping (tSLAM) in the Shrewbot whiskered robot platform. Screen shot taken from a combined video of overhead camera view (*right*) with an appropriately scaled and rotated 2d occupancy grid representation of the arena in a typical particle after approximately 1 h of whisker based tactile exploration (*left*). Figure reproduced from multimedia supplement to [100] with permission from IEEE

UNCORRECTED PROOF

The tSLAM system has now been piloted on two whiskered robot systems—Crunchbot, a modified Roomba vacuum-cleaner robot with a small array of static whiskers [99], and Shrewbot a robot with multiple actively controlled whiskers and a 3 degrees-of-freedom neck (see Fig. 10.1 and [100]). In the case of the Shrewbot platform, odometry derived from the robot base and neck were passed at regular intervals (in phase with the whisking) to a population of particles each maintaining an estimate of head pose and location within a 2-dimensional occupancy grid. The importance of each particle was calculated by fusing the likelihood that each whisker in the array is at the estimated location in the map based on tactile information sampled throughout the previous whisk. The screen shot shown in Fig. 10.11 shows a one hour experimental run of Shrewbot in a 3 m diameter arena. The pink regions representing areas of the map that have a high probability of occupancy, the dashed white line representing ground truths taken from the over head video camera. The dashed white representation of Shrewbot is its ground truth location, whilst the solid representation is the current best estimate of pose and location taken from the particle with highest importance (cloud visible as red dots near the head).

We have also used the Shrewbot platform to model the active touch based hunting behaviour of the Etruscan Shrew [101]. A study of vibrissal-guided predation of insects by the shrew [2] identified three distinct phases of hunting behaviour: search, contact and attack. The search phase was reproduced on Shrewbot using the tactile attention based model of action selection described above, whereby the locus of attention drives the orienting behaviour of the robot between subsequent whisks. Upon making contact, an internal geometric model of two classes of object was compared to the sparse 3-dimensional tactile information derived from the whisker array. The two classes of object were vertical "walls" and the dome shaped covering of a mobile robot referred to as "preybot" (see Fig. 10.12). Shrewbot's re-

Author's Proof

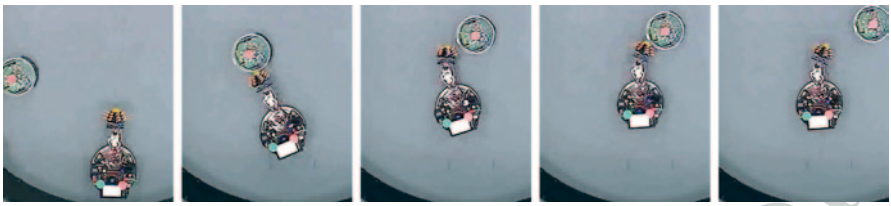


Fig. 10.12 Tactile identification and tracking of a target in a whiskered robot. Snapshots taken with an overhead camera of Shrewbot approaching the "preybot" during hunting behaviour. The images indicate: *search* (*frame 1*), *contact* (*frame 2 and 3*), *attack* (*frame 4*) and a return to *search* (*frame 5*). Figure reproduced from Fig. 3 of [101] with permission from Elsevier

flexive whisker control strategies [102] caused an increase in the whisking set angle similar to that reported in [2] as well as an increase in the number and frequency of whiskers making controlled contacts with the object. This information was collated into a "prey belief" metric that influenced the decision to either attack the object (preybot) or to ignore it (walls). In parallel to the attack decision process, the centre of mass of the preybot was also estimated. To accommodate the relatively sparse information from whisker contacts, some of the known characteristics of the preybot were used to better infer its location and orientation and hence its affordances as a potential "prey" object. The velocity of the preybot was derived from this information and thence a prediction of where a particular point on that robot (in this case the tail) should be in the near future. This location in space was then set as the target for an attack.

Conclusion

In this chapter we have briefly summarised an extensive programme of work aimed at describing and simulating, in biomimetic (robotic) models, the control architecture for sensorimotor co-ordination in the vibrissal touch system of small mammals. We have shown how the evolution of our robot models has progressively captured more-and-more of the important features of the biological target system including morphology, sensory transduction, motor control, and internal processing. Focusing initially on the problem of orienting to vibrissal contacts we have shown that models of the superior colliculus and basal ganglia can be combined to generate sequences of exploratory and orienting movements that allow the robot to explore an environment, and orient to unexpected contacts, in a life-like way. Our robot experiments also revealed a need to pre-process sensory signals in order to distinguish real physical contacts from 'ghost' contact signals induced by the whisking movements of the artificial vibrissae. This led to a novel hypothesis about the role of the cerebellum in vibrissal processing and the demonstration that adaptive filter algorithms modelled on cerebellar microcircuitry can be effective in predicting/cancelling self-induced sensory noise. The task of developing integrated sequences of movements in whiskered robots also revealed the need for spatial memory systems

UNCORRECTED PROOF

Author's Proof

that could effectively encode and remember the location of object contacts and allow the robot/animal to maintain a good estimate of its position in space. To make these systems more effective in identifying, and responding appropriately, to tactile environmental affordances, we are also developing models of cortical systems (particular of primary somatosensory cortex), and have shown that model basal ganglia circuits can make timely decisions between alternatives based on cortical encodings of vibrissal touch signals. Whilst we have yet to realise the full architecture shown in Fig. 10.2 in a single robot, our latest robotic models show a capacity for integrated behaviour that has surprised and impressed exhibition and conference audiences into thinking that they are observing something like a 'robot animal'. From the perspective of understanding brain architecture, we also consider that we have made important steps towards understanding and demystifying the neural-basis for sensorimotor co-ordination in mammalian brains including our own.

Acknowledgement The authors would like to acknowledge the support of the European Union FP7 BIOTACT Project 'Biomimetic Technology for Vibrissal Active Touch' (ICT-215910) and the Weizmann Institute of Science project "Development of motor-sensory strategies for vibrissal active touch". We would also like to thank various collaborators whose experimental and theoretical research has inspired our biomimetic models including Mathew Diamond, Michael Brecht, Ehud Ahissar, David Golomb, Mitra Hartmann, Chris Yeo, Peter Redgrave and Kevin Gurney.

References

1. Brecht M, Naumann R, Anjum F, Wolfe J, Munz M, Mende C et al (2011) The neurobiology of Etruscan shrew active touch. Philos Trans R Soc Lond B Biol Sci 366(1581):3026–3036
2. Munz M, Brecht M, Wolfe J (2010) Active touch during shrew prey capture. Front Behav Neurosci 5:12
3. Roth-Alpermann C, Anjum F, Naumann R, Brecht M (2010) Cortical organization in the Etruscan shrew (Suncus etruscus). J Neurophysiol 104(5):2389–2406
4. Anjum F, Turni H, Mulder PG, van der Burg J, Brecht M (2006) Tactile guidance of prey capture in Etruscan shrews. Proc Natl Acad Sci U S A 103(44):16544–16549
5. Gibson JJ (1979) The ecological approach to visual perception. Houghton Mifflin, Boston
6. Prescott TJ, Diamond ME, Wing AM (2011) Active touch sensing. Philos Trans R Soc Lond B Biol Sci 366(1581):2989–2995
7. Ahissar E, Kleinfeld D (2003) Closed-loop neuronal computations: focus on vibrissa somatosensation in rat. Cereb Cortex 13(1):53–62
8. Braitenberg V (1986) Vehicles: experiments in synthetic psychology. MIT Press, Cambridge
9. Mitchinson B, Pearson M, Pipe T, Prescott TJ (2011) Biomimetic robots as scientific models: a view from the whisker tip. In: Krichmar J (ed) Neuromorphic and brain-based robots. MIT Press, Boston
10. Rosenblueth A, Wiener N (1945) The role of models in science. Philos Sci 12(4):316–321
11. Prescott TJ (2007) Forced moves or good tricks in design space? Landmarks in the evolution of neural mechanisms for action selection. Adapt Behav 15(1):9–31
12. Hallam JCT, Malcolm CA (1994) Behaviour—perception, action, and intelligence—the view from situated robotics. Philos Trans R Soc Lond Ser A 349(1689):29–42
13. Mitchinson B, Pearson MJ, Pipe AG, Prescott TJ (2012) The emergence of action sequences from spatial attention: insight from mammal-like robots. Living machines: biomimetic and biohybrid systems. Barcelona, Spain
14. Prescott TJ, Redgrave P, Gurney KN (1999) Layered control architectures in robots and vertebrates. Adapt Behav 7(1):99–127

UNCORRECTED PROOF

Author's Proof

15. Redgrave P, Prescott TJ, Gurney KN (1999) The basal ganglia: a vertebrate solution to the selection problem? Neuroscience 89:1009–1023
16. Brooks RA (1991) New approaches to robotics. Science 253:1227–1232
17. Gandhi NJ, Katnani HA (2011) Motor functions of the superior colliculus. Annu Rev Neurosci 34:205–231
18. Posner MI (1980) Orienting of attention. Q J Exp Psychol 32:3–25
19. Cisek P (2007) Cortical mechanisms of action selection: the affordance competition hypothesis. Phil Trans Roy Soc B 362(1485):1585–1599
20. Pearson MJ, Mitchinson B, Welsby J, Pipe T, Prescott TJ (2010) SCRATCHbot: Active tactile sensing in a whiskered mobile robot. In: Meyer J-A, Guillot A (eds) The 11th international conference on simulation of adaptive behavior. Paris, Springer
21. Sullivan JC, Mitchinson B, Pearson MJ, Evans M, Lepora NF, Fox CW et al (2012) Tactile discrimination using active whisker sensors. IEEE Sens J 12(2):350–362
22. Fox CW, Evans M, Pearson M, Prescott TJ (2012a) Tactile SLAM with a biomimetic whiskered robot. IEEE International Conference on Robotics and Automation (ICRA)
23. O'Keefe JA, Nadel L (1978) The hippocampus as a cognitive map. London, Oxford University Press
24. Anderson SR, Pearson MJ, Pipe A, Prescott T, Dean P, Porrill J (2010) Adaptive cancelation of self-generated sensory signals in a whisking robot. Robotics. IEEE Trans 26(6):1065–1076
25. Anderson SR, Porrill J, Prescott TJ, Dean P (2012) An internal model architecture for novelty detection: Implications for cerebellar and collicular roles in sensory processing. PLoS One 7(9):e44560
26. Brecht M, Preilowski B, Merzenich MM (1997) Functional architecture of the mystacial vibrissae. Behav Brain Res 84(1–2):81–97
27. McHaffie JG, Stein BE (1982) Eye movements evoked by electrical stimulation in the superior colliculus of rats and hamsters. Brain Res 247:243–253
28. Sahibzada N, Dean P, Redgrave P (1986) Movements resembling orientation or avoidance elicited by electrical stimulation of the superior colliculus in rats. J Neurosci 6(3):723–733
29. Favaro PD, Gouvea TS, de Oliveira SR, Vautrelle N, Redgrave P, Comoli E (2011) The influence of vibrissal somatosensory processing in rat superior colliculus on prey capture. Neuroscience 176:318–327
30. Cohen JD, Hirata A, Castro-Alamancos MA (2008) Vibrissa sensation in superior colliculus: wide-field sensitivity and state-dependent cortical feedback. J Neurosci 28(44):11205–11220
31. Hemelt ME, Keller A (2007) Superior sensation: superior colliculus participation in rat vibrissa system. BMC Neurosci 8:12
32. Miyashita E, Hamada Y (1996) The 'functional connection' of neurones in relation to behavioural states in rats. Neuroreport 7(14):2407–2411
33. Benedetti F (1991) The postnatal emergence of a functional somatosensory representation in the superior colliculus of the mouse. Dev Brain Res 60(1):51–57
34. Drager UC, Hubel DH (1976) Topography of visual and somatosensory projections to mouse superior colliculus. J Neurophysiol 39(1):91–101
35. Mitchinson B, Prescott TJ (2013) Whisker movements reveal spatial attention: a unified computational model of active sensing control in the rat. PLoS Comput Biol 9(9):e1003236
36. Grant RA, Mitchinson B, Fox C, Prescott TJ (2009) Active touch sensing in the rat: anticipatory and regulatory control of whisker movements during surface exploration. J Neurophysiol 101(2):862–74
37. Mitchinson B, Martin CJ, Grant RA, Prescott TJ (2007) Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. Proc Biol Sci 1613(274):1035–1041
38. Mitchinson B, Grant RA, Arkley K, Rankov V, Perkon I, Prescott TJ (2011) Active vibrissal sensing in rodents and marsupials. Philos Trans R Soc Lond B Biol Sci 366(1581):3037–3048

UNCORRECTED PROOF

Author's Proof

39. Towal RB, Hartmann MJ (2006) Right-left asymmetries in the whisking behavior of rats anticipate head movements. J Neurosci 26(34):8838–8846
40. Arkley K, Grant RA, Mitchinson B, Prescott TJ (2014) Strategy change in vibrissal active sensing during rat locomotion. Curr Biol 24(13):1507–1512
41. Hemelt ME, Keller A (2008) Superior colliculus control of vibrissa movements. J Neurophysiol 100(3):1245-1254
42. Miyashita E, Mori S (1995) The superior colliculus relays signals descending from the vibrissal motor cortex to the facial nerve nucleus in the rat. Neurosci Lett 195(1):69–71
43. Gold JI, Shadlen MN (2001) Neural computations that underlie decisions about sensory stimuli. Trends Cogn Sci 5(1):10–16
44. Chambers J, Humphries M, Gurney KN, Prescott TJ (2011) Mechanisms of choice in the primate brain: a quick look at positive feedback. In: Seth A, Bryson JJ, Prescott TJ (eds) Modelling natural action selection. Cambridge University Press, Cambridge, pp 390–418
45. Hikosaka O, Takikawa Y, Kawagoe R (2000) Role of the basal ganglia in the control of purposive saccadic eye movements. Physiol Rev 80(3):953–978
46. Knill DC, Pouget A (2004) The Bayesian brain: the role of uncertainty in neural coding and computation. Trends Neurosci 27(12):712–719
47. Wald A (1947) Sequential analysis. Wiley, New York
48. Bogacz R, Brown E, Moehlis J, Holmes P, Cohen JD (2006) The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. Psychol Rev 113(4):700–765
49. Gold JI, Shadlen MN (2007) The neural basis of decision making. Annu Rev Neurosci 30:535–574
50. Platt ML, Glimcher PW (1999) Neural correlates of decision variables in parietal cortex. Nature 400(6741):233–238
51. Lepora N, Pearson MJ, Mitchinson B, Evans M, Fox CW, Pipe T et al (eds) (2010) Naive Bayes novelty detection for a moving robot with whiskers. IEEE International Conference on Robotics and Biomimetics (ROBIO) Tainjin, December 14–18
52. Lepora N, Fox CW, Evans MH, Diamond ME, Gurney K, Prescott TJ (2012) Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination. J R Soc Interface 9(72):1517–1528
53. Lepora NF, Evans M, Fox CW, Diamond ME, Gurney K, Prescott TJ (eds) (2010) Naive Bayes texture classification applied to whisker data from a moving robot. Neural Networks (IJCNN), The 2010 International Joint Conference on, 2010 18–23
54. Lepora NF, Sullivan JC, Mitchinson B, Pearson M, Gurney K, Prescott TJ (eds) (2012) Brain-inspired Bayesian perception for biomimetic robot touch. Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012 14–18
55. Bogacz R, Gurney K (2007) The basal ganglia and cortex implement optimal decision making between alternative actions. Neural Comput 19(2):442–477
56. Gurney K, Prescott TJ, Wickens JR, Redgrave P (2004) Computational models of the basal ganglia: from robots to membranes. Trends Neurosci 27(8):453–459
57. Lepora N, Gurney K (2012) The basal ganglia optimize decision making over general perceptual hypotheses. Neural Comput 24(11):2924–2945
58. Diamond ME, von Heimendahl M, Arabzadeh E (2008) Whisker-mediated texture discrimination. PLoS Biol 6(8):e220
AQ2
59. Catania KC, Catania EH (2015) Comparative studies of somatosensory systems and active sensing. In: Krieger P, Groh A (eds) Sensorimotor integration in the whisker system. Springer, New York This Volume
60. Lee L-J, Erzurumlu RS (2005) Altered parcellation of neocortical somatosensory maps in N-methyl-D-aspartate receptor-deficient mice. J Comp Neurol 485:57–63
61. Woolsey TA, Welker C, Schwartz RH (1975) Comparative anatomical studies of the SmL face cortex with special reference to the occurrence of "barrels" in layer IV. J Comp Neurol 164(1):79–94

UNCORRECTED PROOF

Author's Proof

62. Keller A (1995) Synaptic organization of the barrel cortex. In: Jones EG, Diamond IT (eds) Cerebral cortex volume II: the barrel cortex of rodents. Plenum, New York
63. Jeffress LA (1948) A place theory of sound localization. J Comp Physiol Psychol 41(1):35–39
64. Joris PX, Smith PH, Yin TC (1998) Coincidence detection in the auditory system: 50 years after Jeffress. Neuron 21(6):1235–1238
65. Fitzpatrick DC, Kuwada S, Batra R (2000) Neural sensitivity to interaural time differences: beyond the Jeffress model. J Neurosci 20(4):1605–1615
66. Wilson SP, Bednar JA, Prescott TJ, Mitchinson B (2011) Neural computation via neural geometry: a place code for inter-whisker timing in the barrel cortex? PLoS Comput Biol 7(10):e1002188
67. Shimegi S, Akasaki T, Ichikawa T, Sato H (2000) Physiological and anatomical organization of multiwhisker response interactions in the barrel cortex of rats. J Neurosci 20(16):6241–6248
68. Ohki K, Chung S, Ch'ng YH, Kara P, Reid RC (2005) Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. Nature 433(7026):597–603
69. Feldman DE, Brecht M (2005) Map plasticity in somatosensory cortex. Science 310(5749):810–815
70. Fox K, Wong RO (2005) A comparison of experience-dependent plasticity in the visual and somatosensory systems. Neuron 48(3):465–477
71. Wilson SP (2011) Figuring time by space: representing sensory motion in Cortical Maps [PhD]. Sheffield
72. Wilson SP, Law JS, Mitchinson B, Prescott TJ, Bednar JA (2010) Modeling the emergence of whisker direction maps in rat barrel cortex. PLoS One 5(1):e8778
73. Miikkulainen R, Bednar JA, Choe Y, Sirosh J (2005) Computational maps in the visual cortex. Springer, Berlin
74. Leiser SC, Moxon KA (2007) Responses of trigeminal ganglion neurons during natural whisking behaviors in the awake rat. Neuron 53(1):117–133
75. Bell CC, Han V, Sawtell NB (2008) Cerebellum-like structures and their implications for cerebellar function. Annu Rev Neurosci 31:1–24
76. von der Emde G, Bell CC (1996) Nucleus preeminentialis of mormyrid fish, a center for recurrent electrosensory feedback. I. Electrosensory and corollary discharge responses. J Neurophysiol 76(3):1581–1596
77. Montgomery JC, Bodznick D (1994) An adaptive filter that cancels self-induced noise in the electrosensory and lateral line mechanosensory systems of fish. Neurosci Lett 174(2):145–148
78. Sawtell NB, Williams A (2008) Transformations of electrosensory encoding associated with an adaptive filter. J Neurosci 28(7):1598–1612
79. Widrow B, Stearns SD (1985) Adaptive signal processing. Prentice-Hall, Englewood Cliffs
80. Fujita M (1982) Adaptive filter model of the cerebellum. Biol Cybern 45(3):195–206
81. Blakemore S-J, Wolpert DM, Frith CD (1998) Central cancellation of self-produced tickle sensation. Nat Neurosci 1(7):635–640
82. Dean P, Porrill J, Ekerot CF, Jorntell H (2010) The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. Nat Rev Neurosci 11(1):30–43
83. Dean P, Porrill J, Stone JV (2002) Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. Proc Biol Sci 269(1503):1895–1904
84. Dean P, Porrill J (2008) Adaptive-filter models of the cerebellum: computational analysis. Cerebellum 7(4):567–571
85. Porrill J, Dean P, Stone JV (2004) Recurrent cerebellar architecture solves the motor-error problem. Proc Biol Sci 271(1541):789–796
86. Porrill J, Dean P (2007) Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems. Neural Comput 19(1):170–193

UNCORRECTED PROOF

Author's Proof

87. Dean P, Porrill J (2014) Decorrelation learning in the cerebellum: computational analysis and experimental questions. Prog Brain Res 210:157–192
88. Wolpert DM, Miall RC, Kawato M (1998) Internal models in the cerebellum. Trends Cogn Sci 2(9):338–347
89. Bastian AJ (2006) Learning to predict the future: the cerebellum adapts feedforward movement control. Curr Opin Neurobiol 16(6):645–649
90. Porrill J, Anderson S, Dean P (2010) Can cerebellar input calibrate collicular topographic maps? BMC Neurosci 11(Suppl 1):120
91. Teune TM, van der Burg J, van der Moer J, Voogd J, Ruigrok TJ (2000) Topography of cerebellar nuclear projections to the brain stem in the rat. Prog Brain Res 124:141–172
92. Moser EI, Kropff E, Moser MB (2008) Place cells, grid cells, and the brain's spatial representation system. Annu Rev Neurosci 31:69–89
93. Itskov PM, Vinnik E, Diamond ME (2011) Hippocampal representation of touch-guided behavior in rats: persistent and independent traces of stimulus and reward location. PLoS One 6(1):e16462
94. Hasselmo ME, Schnell E, Barkai E (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. J Neurosci 15(7 Pt 2):5249–5262
95. Scharfman HE (2007) The CA3 "backprojection" to the dentate gyrus. Prog Brain Res 163:627–637
96. Fox CW, Prescott TJ (2010a) The hippocampus as a unitary coherent particle filter. International Joint Conference on Neural Networks (IJCNN). Barcelona
97. Taylor GW, Hinton GE, Roweis ST (2007) Modeling human motion using binary latent variables Advances in neural information processing systems (NIPS). 19. p 1345
98. Fox CW, Prescott TJ (2010b) Learning in the unitary coherent hippocampus. 20th International Conference on Artificial Neural Networks (ICANN). ThessalonIka
99. Fox CW, Evans MH, Pearson MJ, Prescott TJ (2012b) Towards hierarchical blackboard mapping on a whiskered robot. Robot Auton Syst 60(11):1356–1366
100. Pearson M, Fox C, Sullivan JCP, T. J, Pipe AG, Mitchinson B (eds) (2013) Simultaneous localisation and mapping on a multi-degree of freedom biomimetic whiskered robot. IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe
101. Mitchinson B, Pearson MJ, Pipe AG, Prescott TJ (2014) Biomimetic tactile target acquisition, tracking and capture. Robot Auton Syst 62(3):366–375
102. Pearson MJ, Mitchinson B, Sullivan JC, Pipe AG, Prescott TJ (2011) Biomimetic vibrissal sensing for robots. Philos Trans R Soc Lond B Biol Sci 366(1581):3085–3096
103. Mitchinson B, Sullivan JC, Pearson MJ, Pipe AG, Prescott TJ (eds) (2013) Perception of simple stimuli using sparse data from a tactile whisker array. Biomimetic abd biohybrid systems. Springer, London
104. Prescott TJ, Pearson MJ, Mitchinson B, Sullivan JCW, Pipe AG (2009) Whisking with robots: From rat vibrissae to biomimetic technology for active touch. IEEE Robot Autom Mag 16(3):42–50

UNCORRECTED PROOF

Author's Proof

Chapter 10: Author Query

AQ1. Please check author affiliations.

AQ2. We have updated the following references: “Catania and Catania 2015” and “Pearson et al 2010”. Please check and confirm.

Connection Science

ISSN: 0954-0091 (Print) 1360-0494 (Online) Journal homepage: http://www.tandfonline.com/loi/ccos20

Maximum saliency bias in binocular fusion

Yuhao Lu, Tom Stafford & Charles Fox

To cite this article: Yuhao Lu, Tom Stafford & Charles Fox (2016): Maximum saliency bias in binocular fusion, Connection Science

To link to this article: http://dx.doi.org/10.1080/09540091.2016.1159181

Published online: 31 Mar 2016.

Submit your article to this journal

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=ccos20

Download by: [143.159.115.1] **Date:** 01 April 2016, At: 01:42

CONNECTION SCIENCE, 2016
http://dx.doi.org/10.1080/09540091.2016.1159181

Maximum saliency bias in binocular fusion

Yuhao Lu[a], Tom Stafford[b] and Charles Fox[c]

[a]Commonwealth Scientific and Industrial Research Organisation, Technology Court, Pullenvale, Australia; [b]Department of Psychology, University of Sheffield, Western Bank, Sheffield, UK; [c]Institute for Transport Studies, University of Leeds, Lifton Villas, Leeds, UK

ABSTRACT
Subjective experience at any instant consists of a single ("unitary"), coherent interpretation of sense data rather than a "Bayesian blur" of alternatives. However, computation of Bayes-optimal actions has no role for unitary perception, instead being required to integrate over *every* possible action-percept pair to maximise expected utility. So what is the role of unitary coherent percepts, and how are they computed? Recent work provided objective evidence for non-Bayes-optimal, unitary coherent, perception and action in humans; and further suggested that the percept selected is not the maximum a posteriori percept but is instead affected by utility. The present study uses a binocular fusion task first to reproduce the same effect in a new domain, and second, to test multiple hypotheses about exactly *how* utility may affect the percept. After accounting for high experimental noise, it finds that both Bayes optimality (maximise expected utility) and the previously proposed maximum-utility hypothesis are outperformed in fitting the data by a modified maximum-salience hypothesis, using unsigned utility magnitudes in place of signed utilities in the bias function.

ARTICLE HISTORY
Received 11 August 2015
Accepted 24 February 2016

KEYWORDS
Bayesian brain; psychophysics; utility; bias; perception; binocular fusion

Downloaded by [143.159.115.1] at 01:42 01 April 2016

1. Introduction

Our experience of the world is "unitary coherent". *Unitary* means we only perceive one interpretation at a time (e.g. either a face *or* a vase in the Rubin Vase illusion) rather than a blur of multiple possible interpretations (never the face *and* vase together). *Coherent* means that we almost always perceive scenes comprised of parts which do not contradict one another (e.g we do not see part face and part vase). While perception's unitary coherent (UC) nature may seem *prima facie* obvious from subjective experience, such experience is notoriously unreliable and ill-defined (No, Pessoa, and Thompson, 2000). Further, UC perception is not required at all by optimal decision making, which must consider *all* possible interpretations of sensory data (Bernardo and Smith, 2000) to maximise *expected* utility (MEU strategy), without reference to any single privileged unitary state,

$$a_{\text{MEU}} = \arg_a \max \int_s U(a,s)P(s|d)\,\mathrm{d}s, \tag{1}$$

CONTACT Charles Fox C.W.Fox@leeds.ac.uk

© 2016 Taylor & Francis

where d is sensory data, s are states, U is utility and a are actions.[1] So what are our UC percepts for? MEU inference is generally NP-hard (Cooper, 1990) and so unitary coherent perception seems likely to be involved in some approximation to optimal action selection. What kind of approximation is this? In machine perception, such as machine vision or speech recognition, the unitary coherent maximum a posteriori (MAP) state is often reported as final output,

$$s_{\mathrm{MAP}} = \arg\max_{s} P(s|d). \tag{2}$$

However agents usually have percepts to help make actions rather than for their own sake. In these cases, perceiving the MAP state does not necessarily lead to the best action if the following *MAP strategy* is used,

$$a_{\mathrm{MAP}} = \arg\max_{a} U(a, s_{\mathrm{MAP}}). \tag{3}$$

A recent study (Fox and Stafford, 2012) provided objective evidence of the existence of UC perception, removing the need for reference to "subjective experience" in the UC debate. It further proposed a particular form of UC perception and action, differing from both MEU action and MAP action, called *maximum utility perception* (*MU*). In MU a UC state and action are chosen together,

$$(s_{\mathrm{MU}}, a_{\mathrm{MU}}) = \arg\max_{s,a} U(a, s)P(s|d). \tag{4}$$

For example, suppose you see your neighbour, whose state (s) may be happy or sad. This state is not observable directly but data (d) from his facial expression gives uncertain evidence about it. Approaching the neighbour to start a conversation, avoiding him, or hedging by just saying hello, are three possible actions (a), and you may assign estimates of the utility (U) of each action in each state as in Table 1. MEU, MAP and MU will each choose different actions under these utilities due to their differences in maximising or integrating over the possible states.

Fox and Stafford (2012) used the ambiguous states (s) of a Necker cube image (d) to produce multiple perceptual possibilities and showed MU actions gave higher utility than MAP actions, and were a better fit to human actions than either MEU or MAP actions. However the experiment left open the possibilities that the effect might be specific to Necker cube percepts and that further action selection strategies could fit the data even better than MU.

The present study seeks to address these issues by using a different type of visual ambiguity – binocular fusion (Maier, Panagiotaropoulos, Tsuchiya, and Keliris, 2012).[2] Corroborative evidence of UC perception and action in this new visual task, when taken together with the previous study (Fox and Stafford, 2012), would suggest its generality across many visual tasks. The binocular task has a simpler action space than the Necker cube task to allow more precise noise modelling, and is further designed to be able to differentiate between

Table 1. Payoff matrix used in action-report experiment.

Percept/action	APPROACH	HELLO	AVOID
:-)	10	1	−4
:-(	−20	1	3
Average	−5	1	−0.5

Downloaded by [143.159.115.1] at 01:42 01 April 2016

the MU form of UC and a new related hypothesis – maximum salience (MS) – by including negative utilities.

It is important that as in (Fox and Stafford, 2012), findings concern only the "everyday" system of immediate perception and action: of perception of complex objects *on the scale of fractions of seconds*, such as glancing at a new scene. In contrast it is well-known that MEU-like behaviour occurs both for simpler, stimulus-response type tasks on shorter times scales in specific environments (Körding and Wolpert, 2004); and also in complex percept-action on longer timescales, during which multiple stable percepts and actions can be perceived and deliberated over sequentially (Chater, Tenenbaum, and Yuille, 2006; Eisenhr, Weber, and Langer, 2010; Griffiths, Chater, Kemp, Perfors, and Tenenbaum, 2010; Lepora et al., 2012). The present study does not dispute the existence of these other systems but corroborates the existence of an additional system as in Fox and Stafford (2012) and Lengyel and Dayan (2008) . Similar findings were reported in an fMRI study (Fleming, Whiteley, Hulme, Sahani, and Dolan, 2010) but without explanation via specific strategy models such as MU and MS. As in Fox and Stafford (2012) and psychophysics in general, experimental noise is very high due to the need of very brief stimuli. New noise models are used to analyze this, requiring fewer assumptions that the method of Fox and Stafford (2012).

Downloaded by [143.159.115.1] at 01:42 01 April 2016

2. Varieties of UC perception and action

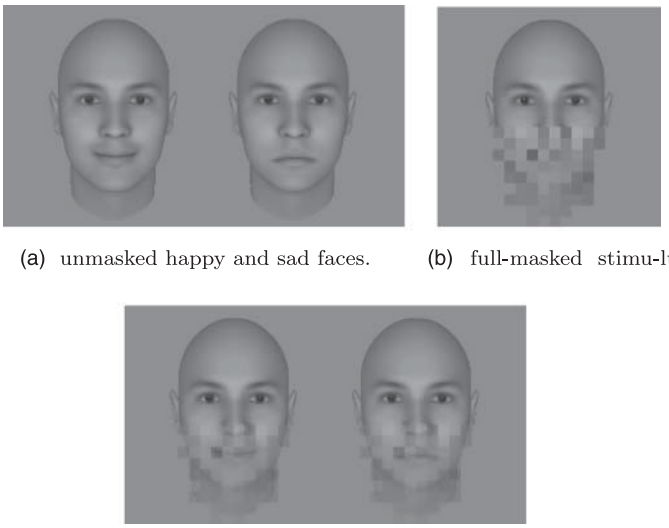
MAP and MU are not the only possible types of UC perception, and we may consider a new, generalised class of strategies which replace the utility term of Equation (4) with other bias functions $B(a, s)$ of action and state:

$$(s_{\mathrm{MB}}, a_{\mathrm{MB}}) = \underset{s,a}{\arg\max}\, B(a, s)P(s|d). \tag{5}$$

The simplest $B(a, s) = 1$ gives the MAP strategy. $B(a, s) = U(a, s)$ gives the MU strategy. Choosing instead $B(s) = |U(a, s)|$ we may define a new *MS* strategy,

$$(s_{\mathrm{MS}}, a_{\mathrm{MS}}) = \underset{s,a}{\arg\max}\, |U(a, s)|P(s|d). \tag{6}$$

This might be more useful than MU in cases where negative utilities are present. For example, consider deciding to reach for an object which may be a sleeping predator or prey with equal probability. MAP would perceive one percept, either predator or prey with a 50% chance each. MU would perceive prey, because it is part of the best-case scenario. But MS is more prudent, seeing the predator, as the large negative utility from waking it and being eaten is more salient than the positive utility of reducing our hunger. Negative utilities were not present in the previous study (Fox and Stafford, 2012) so it was unable to distinguish between MU and MS: they give identical predictions in tasks where all utilities are positive. By introducing negative utilities in the present study we can newly discriminate between MU and MS. The present study tests for MS against MAP, MEU and MU in a controlled environment to determine which best resembles observed human actions. No experiment can measure human perceptions directly, but by measuring actions we can make inferences about the accompanying content of perception within these hypotheses.



(a) unmasked happy and sad faces. (b) full-masked stimu-lus.

(c) half-masked happy and sad.

Figure 1. Three types of stimulus used in the experiment.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

3. Methods

Overview. The present study uses controlled stimuli and forced-choice actions in an artificial binocular fusion task to simplify presentation and analysis. The four hypotheses to be tested against one another are MAP, MEU, MU and MS as in Equations (2)–(4) and (6). Data, d, are image pixels, states are $s \in \{happy, sad\}$ of faces generating the pixels, and actions, a, are button presses. A system of training and examination introduced by (Fox and Stafford, 2012) is used to guarantee that subjects understand the utility function and that their use of it is close to optimal, before collecting data.

Stimuli. Stimuli used in all experiments were pairs of faces, as in Figure 1, which may be unmasked, half-masked or full-masked. The faces in the pair may be the same or different states ("rivalrous") but always have the same degree of masking. *Full masking* is a low-resolution pixellation of the mouth makes the happy/sad state completely unknown; *half-masking* is a higher resolution pixellation which makes it visible with some visual effort, typically a few seconds of full attention to become certain of the happy or sad mouth direction. Binocular rivalry is well-known to lead to reports of subjective experiences of seeing one face or the other.[3] Subjects viewed the stimuli via a pair of dichoptic displays from 70 cm through red-green anaglyphic glasses, seeing one face in each eye. Two faces of neutral gender generated by Facegen Modeller (www.facegen.com) were used to generate the stimuli. Stimuli were presented against a uniform grey background using the MATLAB Psychophysics Toolbox (www.psychtoolbox.org) on an LCD monitor (1024 × 768 pixels, 60 Hz).

Trials. A *trial* means one presentation of a face pair and the collection of one button press from the subject. Figure 2 shows the process for a trial. A fixation cross ($0.5^\circ \times 0.5^\circ$) was presented to each eye. Test images subtend $12^\circ \times 8.5^\circ$ visual angle. Distance from the fixation cross to the mouse of the face stimulus was constantly 4.5° visual angle. Subjects

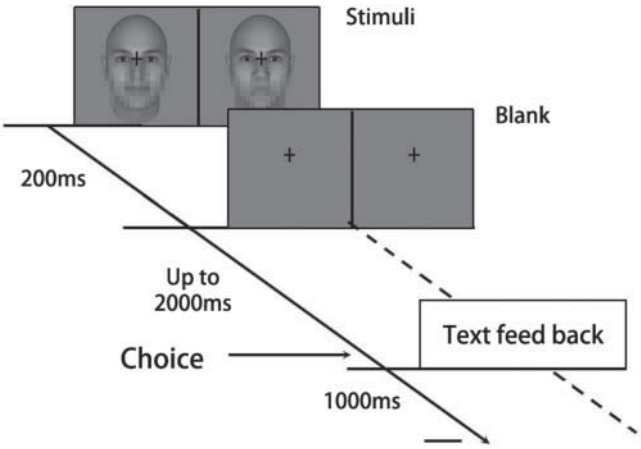


Figure 2. Trial structure. A stereo pair is presented, then blanked, then the user's choice is recorded.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

were instructed to stare at the fixation cross constantly. Face pairs are presented to left and right eye simultaneously, centred on the cross, for 200 ms, followed by a maximum 2000 ms blank screen, during which subjects made a decision to press a button according to their latest instructions. After a button press, the program waited for another 1000 ms then the next trail begun.

Subjects. 23 subjects were recruited, all college students in the Department of Psychology, University of Sheffield. Ages ranged 18–28 years except one 50 year old female, all with normal or corrected-to-normal vision.

Calibration. To simplify the analysis, we would like all rivalrous stimuli to have $P(s) = 0.5$ for both of the coherent states $s \in \{happy, sad\}$. To achieve this, stimuli were adjusted per subject so that in the absence of any utility bias, each subject perceived roughly equal numbers of happy and sad faces when confronted with half-masked rivalrous pairs. An iterative process tailored the stimuli to achieve this by countering biases due to eye dominance, color dominance, or emotional preference. Calibration trails used conflicting pairs of happy and sad faces only, and subjects reported their percepts directly via two buttons marked HAPPY and SAD. Images were adjusted until approximately half each of happy and sad faces were reported. We calibrated both for any possible eye dominance and for emotional preference. An annealed gradient descent algorithm was used to adjust image parameters. Calibration contained 16 blocks, each block contained 10 trials, comprising 5 " left happy/right sad" trials and 5 " right happy/left sad" trials. During each block, if information on one eye had more dominant influence on subject's choice, the image contrast on this eye would be impaired. Similarly, if a subject saw one emotion (happy or sad) more frequently, the image contrast on this emotion would be impaired. The adjustable parameters were: eye bias *DB* and emotional bias *EB*. Another 2 parameters *C*1 and *C*2 are used to record subject's behaviour in one block of the calibration. *C*1 record the 5 left happy right sad trials. In each LHRS trial, if subject recognised the contradictory visual stimuli as an unitary coherent happy face, subject would choose "happy" and C1 = C1+1; if subject see unitary coherent sad face, he/she would choose "sad" and C1 = C1-1; if subject see neutral or contradictory or uncertain about the emotion, he/she would choose "uncertain" and C1 stays the same. *C*2

record the same in right happy left sad trials (RHLS trials). The following annealed gradient descent updates where N is the number of the trial within the block,

$$\Delta DB = e^{(C1-C2)/3N}, \Delta EB = e^{(C1+C2)/3N}. \tag{7}$$

Training and examination. After calibration, subjects are provided with three new buttons and given the following text:

Every day your neighbour is in his garden. Sometimes he is in a good mood and sometimes a bad mood. If he is in a good mood then it is good to start a conversation with him(press TALK button) . If he is in a bad mood it is best to avoid him(press AVOID button) . Or you can just say hello(press HELLO button) , this is quite good if he is in a good mood but not too bad if he is in a bad mood.

The scoring system for percept-action pairs in Table 1, was carefully designed to discriminate between our different hypotheses, such that the different strategies under consideration lead to different actions for rivalrous percepts. As in Fox and Stafford (2012) subjects were required train then pass an "examination" to prove they have fully internalised this scoring system. In training blocks of 30 trials, stimuli on left eye and right eye are the same. 10 are half-masked happy faces, 10 are half-masked sad faces and 10 are full masked random faces.[4] Scores for each trial are printed on the screen immediately after each action. Each training *round* has two blocks, score maximisation and score minimisation, where subjects were asked by the experimenter to obtain highest and lowest scores possible. Subjects perform repeated training rounds until they achieve threshold scores of 98 in maximisation and negative 200 (−200) in minimisation. These thresholds are set such that they can only be achieved by doing the optimal action in *every* trial. Therefore we can be certain that any subject passing this "examination" understands, and can act optimally using, the scoring system. Importantly this includes use of the hedging *HELLO* action when faced with the rivalrous full-masked stimuli. (All subjects passed the exam eventually).

Data collection. Following successful completion of the exam, subjects were presented with a further 42 similar trials. The buttons and scoring were identical to the training rounds. However subjects are asked only to maximise their total score, and the face pairs used

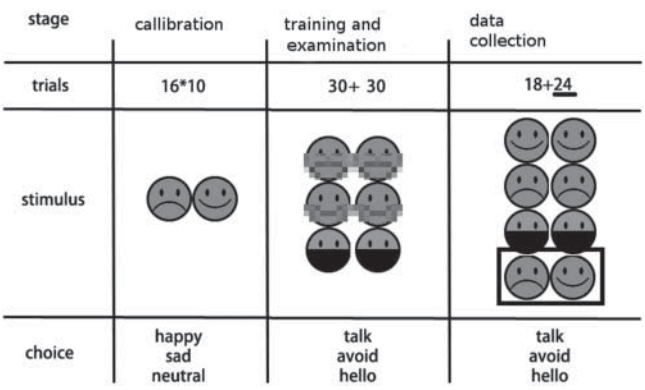


Figure 3. Structure of whole study, only data from boxed/underlined trials are analysed. Blacked-out mouth = full mask; pixellated mouth = half-masked. The three stages proceed sequentially: calibration, training and examination, then data collection. Training and examination is repeated until the examination is passed.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

are different. Here we use the following image pairs: (*happy,happy*), (*sad,sad*), (*happy,sad*), (*sad,happy*), (*full-mask, full-mask)*, where *happy* and *sad* are half-masked. Only the 24 rivalrous pairs are required in this paper's analysis, the other 18 may be considered as distractors.[5]

Debriefing. Subjects were asked for informal reports of their subjective experiences at the end of the whole experiment.

A summary of the trial types used in the experimental stages is shown in Figure 3.

4. Results

Debriefing. Binocular rivalry is sometimes assumed to be a perfect form of perceptual ambiguity, however subjects did report experiences of other types of fusion than rivalry, disclosed here: sometimes a less-happy or less-sad face, see Figure 4(a); sometimes a neutral face, see Figure 4(b); sometimes a reconstruction into other combinations of happy and sad faces, see Figure 4(c); in very rare cases, subjects might see a unitary but non-coherent

(a) single faces with impaired happy and sad emotion.

(b) neutral emotion.

(c) One of many examples that subjective experience of half happy and half sad.

(d) The example of subjective experience of contradictory face with smiling and sad mouth overlapping each other.

Figure 4. Mockups of reconstructed unitary coherent subjective perception after binocular fusion.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

face, see Figure 4(d). This was reported by 3 of 23 subjects. Reports of "no percept" were common due to the short presentation time (shorter than in the examination). Also the long data collection period may have induced boredom in some subjects leading to giving up on the task and to completely random actions. All subjects were found naive to the purpose of the study, and unaware of binocular fusion theory.

Testing hypotheses against the data must take account of experimental noise. This is because for each of the pure hypotheses, there is at least one action it would never make, but the results do contains mixtures of all the actions. From debriefing it was clear that some stimuli were not perceived at all or perceived in ways not included in the hypotheses. Such experimental noise is very common in forced-choice psychophysics experiments, and is easily accounted for.

We therefore consider each *hypothesis* ($h \in \mathrm{MAP}, \mathrm{MEU}, \mathrm{MU}, \mathrm{MS}$) to be set of *models*, $\{h(\theta)\}_\theta$, having continuous values of a free parameter θ. The meaning of $\theta = P(\text{noise})$ is the probability that a trial's action is a random button press rather than the optimal action a_h of the hypothesis. For each model in a hypothesis' set, the likelihood $P(data|h(\theta))$ can be computed by

$$P(\text{data}|h(\theta)) = \prod_a P(a|h(\theta))^{|a|}, \tag{8}$$

where $|a|$ is the number of trials in which action a was taken in the observed *data*. We consider the set of all rivalrous trials as *data*, other trials are distractors. Making minimal assumptions, a flat prior is taken over hypotheses and noise values.

Response prediction from a selection of these models are shown in Figure 5. Dotted lines show predictions using 10 values of θ from 10% to 100% in steps of 10%. The pure, $\theta = 0$ model shown by the dashed line: for MEU, MU and MS these predict different single actions, and for MAP predict a mix of two actions. For each hypothesis, the optimal θ is found by numeric search. Its value and corresponding log likelihood – the best log likelihood for the hypothesis whole model set – is shown above the graph, and its prediction as a thick line.

We seek the single best model from the four hypotheses' combined sets of models. To find it, it suffices to find the best-fitting models within each of the four hypotheses, according to their log-likelihoods, then compare these four best models against one another. It can be seen from the log-likelihoods in Figure 5 that the overall best model is $\mathrm{MS}(\theta = 0.85)$. This has a log-likelihood of 12.2 greater than any of the other hypotheses' models, so is $e^{12.2} \approx 200,000$ times more likely than any of them.

In addition to the four models under test, we considered two further models which may help to reduce or give insight into the noise levels and alternative future model classes.

Perfect two-parameter model. The four models tested each have one free parameter, θ, and we note that as there are only three possible actions, a perfect fit can be made by a two-parameter model (2PM) which simply assigns frequentist probabilities to two actions $P(TALK)$ and $P(HELLO)$ according to their observed counts, and thus $P(AVOID) = 1 - P(TALK) - P(HELLO)$. This model is obviously over-fitted to the data, but may be analyzed using Bayesian Information Criteria scores, $BIC = -2 \ln L + k \ln n$. The log likelihood $\ln L$ of 2PM on the data was found to be -585.71, compared with -594.28 for the MS model, a difference of 5.13 in favour of 2PM. BIC with $n = 552$ data points and k free parameters yields penalty terms (Occam factors) of $1 \times \ln 552 = 6.3$ and $2 \times 6.3 = 12.4$ respectively. This implies that the best possible two parameter model, and thus *any* two parameter

Downloaded by [143.159.115.1] at 01:42 01 April 2016

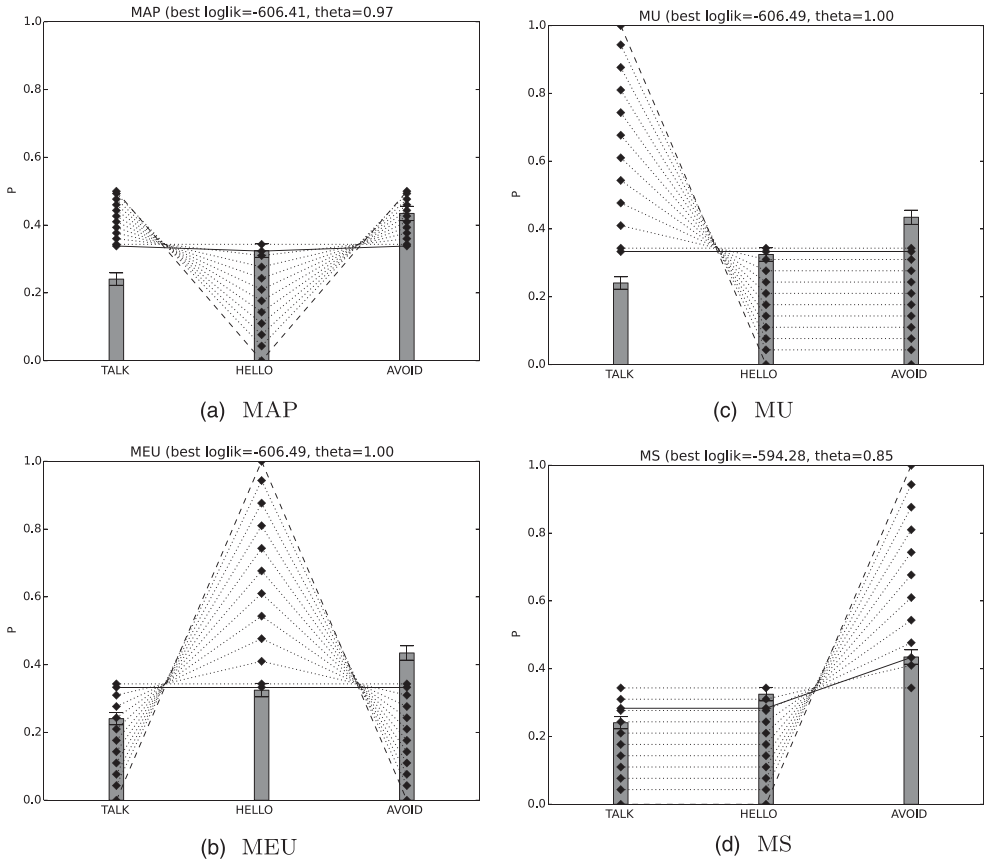


Figure 5. Fitting four hypotheses to the Action Report task: MAP, MEU, MU and MS. Observed responses from 552 trials are plotted as grey bars, as beta posteriors of their generators with 1 standard deviation bars. Each hypothesis has one free parameter, θ. Choosing a value of θ specifies a model with the hypothesis. Dotted lines show models with P(noise) $=$ 10%:10%:100%; dashes show the noiseless model $\theta = 0$. The best fitting θ model within each hypothesis is shown by a bold line; its value and log likelihood are above the graph. All models are discrete, each describing only three numerical probabilities over the three actions as shown by the diamonds – the continuous dotted, dashed, and thick lines are used only to indicate which predictions come from which model.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

model – such as any weighted mixture of strategies – has a worse BIC score than MS so can make no claim to be a better model than MS. However the BIC difference is not significant so this does not rule out two parameter models, it only shows they are not better than MS. For now we argue that MAP,MEU,MU and MS should have higher *subjective* priors because they are theory-driven rather than just fitted to data, and we exclude 2PM for this reason, but future larger data could resolve this without the need for subjective experimenter assumptions.

Per-subject model. The present study was not intended to explore per-subject effects, and the data sizes were selected to obtain significant population results at minimal financial experiment cost. However we can examine what, if any, power the present data has to inform per-subject models. While we cannot test every possible per-subject model, a reasonable model might be a Per Subject Switching (PSS) model which assigns to each

subject the best fitting choice of MAP,MEU,MU or MS. In particular this would separate out the effects of some bored subjects abandoning the task from less noisy subjects. Assigning one indicator variable to each of 23 subjects, and assuming a single free noise parameter θ shared over all of MAP,MEU,MU and MS, gives gives $k = 24$. Optimising over θ, the best log likelihood of PSS on the data was found to be -508.87 (at $\theta = 0.50$), a difference of 85.4 in favour of PSS over MS. However as with 2PM, this gain is outweighed by a large penalty term $k \ln n = 151$ which makes MS very strongly better than PSS. It remains possible that with larger data, per-subject effects may be resolvable, but the present data is not able, or intended, to do this.

5. Discussion

The maximum likelihood model is MS with a $\theta = 0.85$, that is, most (85%) of trails in the test phase are responded by a random action. Importantly – the clearly significant victory of MS (200,000 more likely than its competitors) in the hypothesis contest is obtained despite this very high experimental noise level. This is possible because of the large number ($N = 552$) of trials conducted – a large number of very noisy trials give rise to a non-noisy, significant finding. The noise level was higher than expected during the experimental design, and possible causes include the absence of any percept at all due to the short (200ms) presentation time; bored subjects acting randomly to reach the end of the long experiment without regard for winning the small prize; and the occasional incoherent percepts reported during debriefing. These are all similar issues to the noise sources in (Fox and Stafford, 2012): unitary coherent perception is very difficult to isolate experimentally, because as is common in forced-choice psychophysics, too short a presentation results in no perception and random action, whilst too long a presentation allows time for other forms of cognition to occur. Psychophysics is famously difficult for this reason, and it remains a challenge to design more precise experiments to reduce the noise. High noise is expected because each presentation is short and barely perceivable, in order to avoid subjects seeing multiple interpretations in sequence, and so as in forced-choice blindsight experiments we expect each trial to be noisy but the aggregate of many trials to obtain significance. We stress that *despite the high noise, we have still shown MS to be 200,000 times more likely than the other hypotheses* because the large data set compensates for the per-trial noise.

As in any data modelling study, the possibility remains of newer, alternative hypothesis better fitting the data, than any of those tested so far. However, our hypothesis set includes the two most popular views of perception and action – MAP and MEU – which are not straw men. And they are clearly falsified in the study's domain by the success of MS. This does not mean that MS is "true", just that it is a better model that was previously available. (This is standard Bayesian logic as detailed in (Jaynes and Bretthorst, 2003)). The contribution of this paper is in falsifying both MEU and MU for the face task, by proposing a new best hypothesis MS which beats them. Any future invention of new models giving even better fits than MS would be additional scientific progress – just as the present study refines and replaces the previous MU model of (Fox and Stafford, 2012) with MS. Future models might include *mixtures* of the current models. For example, new models might be completely non-probabilistic, or not involve assumptions about the existence of any mental representation of *s*. Or mixtures of strategies might occur within or between subjects. Mixtures made only of the hypotheses tested in the present study must still have MS as

Downloaded by [143.159.115.1] at 01:42 01 April 2016

the strongest component because it has been shown to fit the data better than the others. Mixtures and other complex models will incur high Occam factor penalties for any extra parameters, and while the present study does not rule them out, it suggests that larger data sets would be needed to resolve competitions of mixture models against MS as the present data was not designed to test them.

The victory of MS in the present hypotheses contest is in agreement with the Necker cube study of (Fox and Stafford, 2012), fMRI work of (Fleming et al., 2010), and the unitary coherent hippocampal perception model of (Fox and Prescott, 2010). Together these findings suggest the existence of unitary coherent perception and action based upon it in more general domains. Given the current available hypothesis set, we have shown as in Fleming et al. (2010) and Mulder, Wagenmakers, Ratcliff, Boekel, and Forstmann (2012) that *utilities of available actions can affect what you see* in a new domain, and refined a novel, quantitative MS explanation which advances from the previous MU explanation. Utility biases unitary coherent perception, as is immediately clear from the difference between observed TALK and AVOID frequencies (recalling that calibration set the corresponding utility-neutral frequencies to be equal). The particular form of this bias now appears to be MS – exactly as in Orwell's story " 1984", whose hero not only reports but actually *sees* five fingers instead of four to avoid pain.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

Notes

1. Sampling approximations to MEU ignore some improbable interpretations (Lunn, Spiegelhalter, Thomas, and Best 2009), but still sum over multiple samples of possible world states *s* rather than privileging any particular unitary state.
2. Binocular "rivalry" is a special case in which a UC percept is seen that matches one of the eye inputs exactly. Pre-supposing the presence of rivalry would not allow MEU to be tested, so the more general term "fusion" is used throughout this paper.
3. We make no use of potentially unreliable subjective reports, only of directly observed, utility-driven actions. Percepts can then be inferred from the actions under the various models being tested. If subjects had difficulty seeing the mouth direction in half-masks, the degree of masking was adjusted to make it just possible for them at the start of the experiment, before calibration.
4. Half-masking was used here because in pilot experiments, subjects trained on unmasked tasks tended to notice something was different in the data collection phase and perhaps realise that a "trick" is being played on them by the experimenter, but half-masking was found to remove this effect because it accustoms subjects to the presence of noise in general.
5. Criticism of the experimental logic might argue that the training and experiment tasks are different, because "bait and switch" is applied to the stimuli between them, and that subjects have therefore been trained to rely on a unimodal distribution and are unfairly switched to a task relying on a bimodal one. However this is precisely the point of the experiment. The underlying probabilistic *model* required to solve the tasks does not change, only the shape of the distribution *over* this model. The fact that unimodal distributions *can* be used is sufficient to accept the existence of UC perception in at least these tasks.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Bernardo J., & Smith A. (2000). *Bayesian theory*. New York, NY: Wiley.

Chater N., Tenenbaum J. B., & Yuille A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*(7), 287–291, special issue: Probabilistic models of cognition.

Cooper G. F. (1990). The computational complexity of probabilistic inference. *Artificial Intelligence*, *42*, 393–405.

Eisenfhr F., Weber M., & Langer T. (2010). *Rational decision making*. Berlin: Springer.

Fleming S. M., Whiteley L., Hulme O. J., Sahani M., & Dolan R. J. (2010). Effects of category-specific costs on neural systems for perceptual decision-making. *Journal of Neurophysiology*, *103*(6), 3238–3247.

Fox C., & Prescott T. (2010). *Hippocampus as unitary coherent particle filter*. International Joint Conference on Neural Networks (IJCANN), IEEE, Barcelona, Spain, pp. 1–8.

Fox C., & Stafford T. (2012). *Maximum utility unitary coherent perception vs. the Bayesian brain*. Proc. Ann. Conf. Cognitive Science Soc, Sapporo, Japan.

Griffiths T. L., Chater N., Kemp C., Perfors A., & Tenenbaum J. B. (2010, June). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences*, *14*(8), 357–364.

Jaynes E., & Bretthorst G. (2003). *Probability theory: The logic of science*. Cambridge: Cambridge University Press.

Körding K., & Wolpert D. (2004). Bayesian integration in sensorimotor learning. *Nature*, *427*(6971), 244–247.

Lengyel M., & Dayan P. (2008). *Hippocampal contributions to control: The third way*. In NIPS 2008, Cambridge, MA: MIT Press.

Lepora N. F., Sullivan J. C., Mitchinson B., Pearson M., Gurney K., & Prescott T. J. (2012). *Brain-inspired Bayesian perception for biomimetic robot touch*. 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 5111–5116.

Maier A., Panagiotaropoulos T. I., Tsuchiya N., & Keliris G. A. (2012). Introduction to research topic - binocular rivalry: A gateway to studying consciousness. *Frontiers in Human Neuroscience*, *6*, 263. doi:10.3389/fnhum.2012.00263

Mulder M. J., Wagenmakers E.-J., Ratcliff R., Boekel W., & Forstmann B. U. (2012). Bias in the brain: A diffusion model analysis of prior probability and potential payoff. *The Journal of Neuroscience*, *32*(7), 2335–2343.

No A., Pessoa L., & Thompson E. (2000). Beyond the grand illusion: What change blindness really teaches us about vision. *Visual Cognition*, *7*, 93–106.

Lunn D., Spiegelhalter D., Thomas A., & Best N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, *28*(25), 3049–3067.

Downloaded by [143.159.115.1] at 01:42 01 April 2016

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

The Sheffield Wargame Corpus - Day Two and Day Three

Yulan Liu[1†], *Charles Fox*[2], *Madina Hasan*[1], *Thomas Hain*[1††]

[1]MINI, SpandH, The University of Sheffield, UK
[2]The University of Leeds, UK
[†]acp12yl@sheffield.ac.uk, [††]t.hain@sheffield.ac.uk

Abstract

Improving the performance of distant speech recognition is of considerable current interest, driven by a desire to bring speech recognition into people's homes. Standard approaches to this task aim to enhance the signal prior to recognition, typically using beamforming techniques on multiple channels. Only few real-world recordings are available that allow experimentation with such techniques. This has become even more pertinent with recent works with deep neural networks aiming to learn beamforming from data. Such approaches require large multi-channel training sets, ideally with location annotation for moving speakers, which is scarce in existing corpora. This paper presents a freely available and new extended corpus of English speech recordings in a natural setting, with moving speakers. The data is recorded with diverse microphone arrays, and uniquely, with ground truth location tracking. It extends the 8.0 hour Sheffield Wargames Corpus released in Interspeech 2013, with a further 16.6 hours of fully annotated data, including 6.1 hours of female speech to improve gender bias. Additional blog-based language model data is provided alongside, as well as a Kaldi baseline system. Results are reported with a standard Kaldi configuration, and a baseline meeting recognition system.
Index Terms: distant speech recognition, multi-channel speech recognition, natural speech corpora, deep neural network.

1. Introduction

Multi-channel based speech enhancement has been shown to be effective for Distant Speech Recongition (DSR), in both classical HMM-GMM systems and state-of-art Deep Neural Networks (DNNs) based systems. Compared to using recordings from single distant microphone only, beamforming is reported to reduce word error rate (WER) by 6-10% relative in large vocabulary conversational speech recognition tasks [1–3], and up to 60% relative in specific tasks [4, 5]. Multi-channel dereverberation brings an extra 20% relative WER reduction over single channel dereverberation [6]. Recently progress in neural networks have introduced further performance improvement in a variety of tasks, particularly from three aspects: progress in novel network structures [7, 8], application-oriented neural network structure and parameter manipulation [9–12], and data manipulation for neural network training [1, 13]. While the overall WERs keep going down, the recognition performance gap remains between using recordings from close-talking microphones and from distant microphones. To reduce this gap, research effort has focused on three approaches: developing novel structures to better utilize multichannel recordings in DNN [14, 15], employing task dependent meta information [16, 17], and simulating training data for specific DSR tasks [6, 18]. However research progress is limited by lack of data that provides multichannel distant recordings accompanied with headset recordings and speaker location tracking, in a natural speech setting where speakers are allowed to move freely. To address this problem, the present study extends the first Sheffield Wargames Corpus (SWC1, [19]) with more natural speech recordings from both headsets and distant microphones in moving talker conditions, accompanied with real time speaker location tracking.

The paper is organized as follows. §2 reviews related work, §3 provides basic information about the set-up for the new recording days SWC2 and SWC3. §4 discusses dataset definitions for two different ASR tasks: adaptation and standalone training. The details about language models (LMs) are introduced in §5. §6 provides results for two tasks, using HTK, TNet and Kaldi. All WERs on eval set are above 40% for headset recordings, and above 70% for distant recordings. §7 concludes the work.

2. Multi-channel Recordings in DSR

Research on utilizing multi-channel recordings within DNN structure started with directly concatenating features from multiple channels at DNN input [1, 2]. Such method was found to perform similar or better than weighted delay and sum beamforming (wDSB) in 2 and 4 channel cases. Furthermore, joint optimization of beamforming and DNNs achieved 5.3% relative improvement over using wDSB in [15]. In [14], beamforming and standard feature pipeline are completely replaced with neural networks. Different neural networks are combined to extract information from raw signals, achieving 5.8% relative WER improvement over 8 channel delay and sum beamforming (DSB).

Meta-information can also be provided to DNNs. In [16], adding noise information provides a 5.1% relative improvement over feature enchancement. In [17], adding room information via feature augmention improves performance by 2.8% relative on the ReverbChallenge real data. In [2], geometry information was added via augmenting the concatenated multi-channel features with Time Difference of Arrival (TDOA) at DNN input. However, no improvement was observed.

Above approaches all require large data sets for training. One main challenge in DSR is the variety in environment conditions of real recordings. Even within the same room, speakers may move around a room, resulting in continually changing room impulse responses (RIRs). One method to address this issue is multi-condition training [6], by simulating data of different environment conditions with different RIRs and by adding background noise to clean speech. The RIRs can be either generated by simulation or measured in real environments [20–22]. Examples of corpora with simulated environment effects are Aurora [23–25], DIRHA-GRID [26] and DIRHA-ENGLISH [27]. Another method is to select targeted RIRs that match best to the test scenario [18]. However there is a lack of corpora

Copyright © 2016 ISCA

http://dx.doi.org/10.21437/Interspeech.2016-98

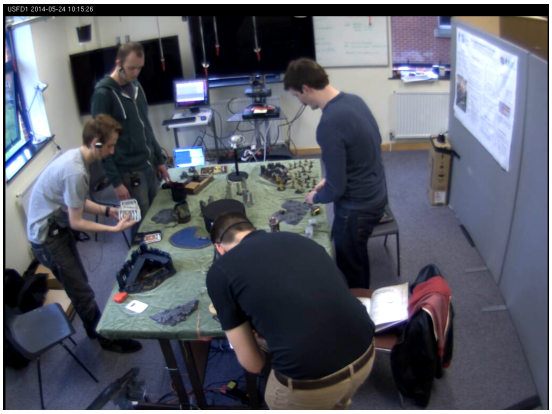


Figure 1: *SWC2 recording (from Camera C1 in Fig. 2).*

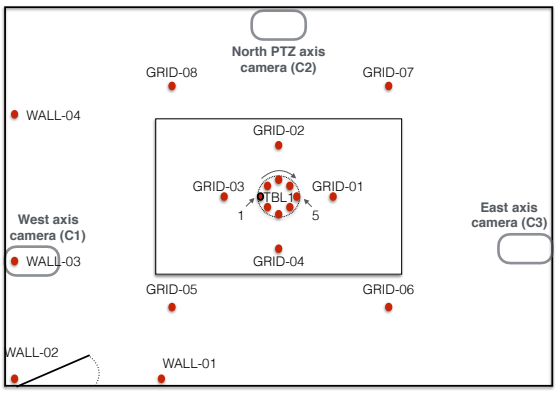


Figure 2: *Video location in SWC2.*

covering different environment conditions that also have natural speech. Existing research corpora of real multi-channel distant recordings often use artificial scenarios, read speech and re-recorded speech. Examples are the real recording part of MC-WSJ-AV corpus [28] used in ReverbChallenge 2014 [6], or the CHiME corpora [29]. Other corpora are recorded with controlled environment and speaker movement, such as the meeting corpora AMI [30] and ICSI [31].

The first Sheffield Wargame Corpus (SWC1, [19]) released in 2013 is a natural, spontaneous speech corpus of native English speakers who are constantly speaking and moving while playing tabletop games. It includes 3-channel video recordings and 96-channel audio recordings from headsets and distant microphones at static known locations in the room. Besides, it includes ground truth head location, providing a reference for research on localization and beamforming algorithms. The task is challenging as it represents everyday colloquial conversation among friends, with emotional speech, laugher, overlapping speech fragments as well as body movement while speaking.

The size of SWC1, 8 hour speech, limits its usefulness for training and adaptation. In addition, SWC1 contains male speech only. This paper releases, for free use in the research community, the extended Sheffield Wargame Corpus recording Day 2 (SWC2) and Day 3 (SWC3). In addition, it releases blog and wikipedia based text data to build in-domain LMs, along with a well built set of in-domain LM and dictionary. SWC3 provides 6.1h of female speech to provide a gender balance. Combined with SWC1, the corpora form a total of 24.6h speech database. Standard datasets are defined to enable comparative research on combined corpora for two scenarios: adapting existing acoustic models (AMs) to SWC data, and standalone training of AMs with SWC data only. An open-source Kaldi recipe is provided for standalone training. Baseline experiment results are reported for both standalone and adaptation systems.

3. SWC2 and SWC3 Recordings

Following the set-up for SWC1 [19], the extended corpora are comprised of recordings where four participants play the tabletop battle game Warhammer 40K[1] (Fig. 1). The game is chosen as a close proxy for sections of business meetings and social interactions where participants are moving and talking at the same time. Such joint motion and talking is difficult to record for extended periods in those contexts but the game promotes it constantly for hours at a time, allowing much more relevant data to be captured. The game is also played by a tight community of friends, many of whom are used to wearing headset microphones from online gaming, and are generally uninhibited by recording technology. Thus they speak more naturally and colloquially during recording, and they could move while speaking. In SWC2, the final two sessions have male viewers commenting on the game, to simulate a cocktail-party scenario. In SWC3, female players (with headsets) are instructed by two male tutors (without headsets) due to less game experience.

The recordings of SWC2 and SWC3 were performed in the same meeting room as SWC1, whose geometry is detailed in [19]. The recording system has three parts: multiple microphone audio recording, multiple camera video recording and location tracking. The three corpora use the same location tracking system Ubisense, which tracks the real time 3D head location of four players during the recording process [19]. Three channels of video recordings from cameras installed at three corners of the ceiling are also provided in SWC2 (Fig. 2).

24-channel audio recordings from the integrated Sheffield IML audio recording system [32] are shared among all three corpora (Fig. 2). They contains 4 headsets for 4 game players, 8 microphones in a circular array at the center of table (diameter: 20cm), 8 microphones hanging on a grid from the ceiling and 4 microphones distributed on the walls, all synchronized at sample level [19]. In SWC2, extra audio recordings are included using a Microcone array, a circular digital MEMs microphone array and an Eigenmic array. The Microcone array has 6 microphones in a circular array (diameter: 8cm), plus the seventh microphone pointing right up to the ceiling. The MEMs digital array has 8 microphones in a circular array with a diameter of 4cm. Both Microcone array and MEMs microphone array are situated on the table. The Eigenmic array is a 32-channel sphere array (diameter: 8.4cm). Only part of Session 1 in SWC2 has Eigenmic recordings due to software failure.

Table 1 lists statistics of SWC1 [19], SWC2 and SWC3. The vocabulary of SWC3 is much smaller compared to SWC1 and SWC2. This is because the game set-up is simplified for less experienced players, leading to simpler conversation.

4. Dataset Definition

Consistent datasets have been defined for SWC1, SWC2 and SWC3. Each recording session, i.e. a continuous recording file (Table 1), is first split into three successive strips of approximately equal speech duration: A, B and C. Such "data strip" allows flexible session combination to create datasets for which results can be easily shared among researchers. Four dataset

[1] https://en.wikipedia.org/wiki/Warhammer_40,000

Table 1: *SWC statistics.*

	SWC1	SWC2	SWC3	overall
#session	10	8	6	24
#game	4	4	3	11
#unique speaker	9	11	8	22
gender	M	M	F&M	F&M
#unique mic	96	71	24	103
#shared mic	-	-	-	24
speech duration	8.0h	10.5h	6.1h	24.6h
#speech utt.	14.0k	15.4k	10.2k	39.6k
duration per utt.	2.1s	2.5s	2.2s	2.2s
#word per utt.	6.6	7.9	5.5	6.8
vocabulary	4.4k	5.7k	2.9k	8.5k
video	√	√	-	√
location	√	√	√	√

Table 2: *Dataset statistics ("spk.": speaker; "dur.": duration).*

config.	set	strips	dur.	#utt.	#spk.
AD1	dev	{1, 2, 3}.A+B	16.3h	26.2k	22
	eval	{1, 2, 3}.C	8.2h	13.3k	22
AD2	dev	1	8.0h	14.0k	9
	eval	2, 3	16.6h	25.6k	18
SA1	train	1, {2, 3}.A	13.5h	22.6k	22
	dev	{2, 3}.B	5.5h	8.5k	18
	eval	{2, 3}.C	5.6h	8.4k	18
SA2	train	1	8.0h	14.0k	9
	dev	{2, 3}.A	5.5h	8.7k	18
	eval	{2, 3}.B+C	11.1h	16.9k	18

definitions based on strips are proposed to serve for two typical tasks: adaptation and standalone training. For each task, two configurations are available with different data separation and difficulty level, as listed in Table 2.

Adaptation task ("AD") only has dev and eval datasets. The "AD1" configuration uses 16.3h speech of Strip A and Strip B from all three recordings as dev set, and the remaining 8.2h of speech from Strip C as evaluation set. This dataset definition provides the least separation of speaker and speaking style. The "AD2" configuration only uses 8.0h SWC1 as dev set, while using the whole SWC2 and SWC3 for eval set. This is representative of many real applications where significant mismatch exists between trained system and test conditions, with limited data for adaptation and a variety in speaker, speaking style, and with subtle differences in topic and vocabulary.

Standalone training task ("SA") has train, dev and eval datasets. The "SA1" configuration uses 13.5h speech for training, comprised of whole SWC1, Strip A of SWC2 and SWC3. The development set uses 5.5h speech of Strip B from SWC2 and SWC3, and evaluation set uses 5.6h speech of Strip C from SWC2 and SWC3. This dataset definition takes into account the balance in gender and speaking style across training and testing. The "SA2" configuration provides only 8h speech from SWC1 for training, 5.5h speech from Strip A of SWC2 and SWC3 for development, and the remaining 11.1h as evaluation set. This dataset definition provides the best separation of speaker, session, game and speaking style between training and testing.

5. Language Modelling and Dictionary

SWC corpora are designed for research on acoustic modelling in natural speech recognition, particularly with multi-channel distant recordings. Since the conversation topic and vocabulary differ from most existing corpora, text data is harvested from four Warhammer 40K blogs and Warhammer wikipedia pages. These data are added to the conversational web data [33] to build an in-domain LM. N-gram components are trained using SRILM toolkit [34] on a 30k vocabulary list. The vocabulary list is built from all words in the harvested text plus the most frequent words in the conversational web data. The LM components are first built on each type of data, and then interpolated using SWC1 as development set. Table 3 lists the LM components and the interpolation weights for 4-gram LM. In initial experiments it was observed that a 4-gram LM trained on 30k vocabulary performs similarly to the RT'09 50k 3-gram LM, while using 3-gram or only using a smaller vocabulary degrades recognition performance. Thus results based on 4-gram LM with a vocabulary of 30k words are reported in following experiments. The perplexity of the interpolated 4-gram LM is 173.4 on SWC1, 195.9 on SWC2, 135.0 on SWC3 and 173.3 overall. The number of words out-of-vocabulary (OOV) is 1.4k on SWC1 (1.6%), 2.8k on SWC2 (2.4%), 3.9k on SWC3 (6.9%) and 8.1k overall (3.1%). Pronunciations are obtained using the Combilex pronunciation dictionary [35]. The Phonetisaurus toolkit [36] is used to automatically generate the pronunciation for words not in Combilex.

Table 3: *LM data size and interpolation weights (4-gram).*

LM component	#words	weight
Conversational web data	165.9M	0.65
Blog 1 (addict)	21.1k	0.05
Blog 2 (atomic)	126.8k	0.05
Blog 3 (cadia)	40.4k	0.19
Blog 4 (cast)	71.2k	0.06
wikipedia (warhammer)	26.2k	0.003

6. Baseline System

6.1. Adaptation task

The acoustic models trained on AMI corpus data are used in "AD2" configuration. The experiments here are performed with HTK and TNet. TNet is used to train DNN and to generate bottleneck features. HTK is used to train HMM-GMM using bottleneck features. The configuration mostly follows the procedure presented in [2]. The AMI dataset definition however follows [1] for a better comparison with other research groups. The 368 dimensional input to DNN are compressed from 31 contextual frames of 23 dimensional log-Mel-filter bank features with DCT [2]. The DNN topology is 368:2048$\times$3:26:1993.

When adapting AMI models to SWC data, the trained DNN is first fine-tuned with dev data using manual transcription. The alignment is obtained with AMI DNN-HMM-GMM and SWC headset recordings. Bottleneck features are then generated with the updated DNN, followed by segmental mean normalization. The AMI HMM-GMM is then maximum-a-posterior (MAP) adapted using "AD2" dev set data for 8 iterations. Neither speaker adaptation or normalization is involved. Results of the baseline systems are reported on individual headset microphone (IHM), single distant microphone (SDM) and multiple distant microphones (MDM) in Table 4. For MDM, weighted delay and sum beamforming is performed using BeamformIt [37], with the 8 channels circular array in the integrated IML recording system ("TBL1"). The scoring for IHM is performed with NIST tool *sclite*, while the scoring for SDM and MDM is performed with *asclite* with a maximum of 4 overlapping speakers.

Even with supervised adaptation on dev data, it still yields a WER of 24.9% for IHM, 55.2% for SDM and 53.5% for MDM

Table 4: *AMI to SWC: "AD2" baseline (WER in %).*

	dev	eval					
	SWC1	SWC2	SWC3	overall			
				Sub.	Del.	Ins.	WER
IHM	24.9	46.4	50.5	33.4	9.3	5.0	**47.7**
SDM	55.2	75.0	85.2	53.2	19.1	6.0	**78.2**
MDM	53.5	71.6	82.4	52.4	15.4	7.3	**75.0**

with 8 channel beamforming. WER on eval data is much higher, particularly for SWC3 due to mismatch in gender and speaking style. Beamforming lowered the WER by 3.1% relative on SWC1, 4.5% relative on SWC2 and 3.3% relative on SWC3.

6.2. Standalone training task

A Kaldi recipe is released with the corpora, providing scripts to train a state-of-the-art context dependent DNN-HMM hybrid system on SWC data only. It follows the routine in other Kaldi recipes (such as AMI).

Following the default configuration, 13 dimensional MFCC features from 7 contextual frames (+/-3) are extracted and compressed with linear discriminant analysis (LDA) to 40 dimension. The output will be further referred to as "LDA features". The LDA features are used to train HMM-GMM. No external alignment is used in the recipe. Instead, the initial model training uses hypothesis timing where utterances are split into equal chunks. The alignment is updated each time the acoustic model significantly improves during the training process.

An HMM-GMM based on monophone is first trained, then an HMM-GMM based on clustered states is trained, followed by LDA and maximum likelihood linear transform (MLLT), speaker adaptive training (SAT), and maximum mutual information (MMI) training. Alignments from the system with LDA+MLLT is used for DNN training. The input of DNN is a 520 dimensional feature vector, comprised of 13 (+/-6) contextual 40 dimensional features that were used for HMM-GMM training. DNN parameters are initialized with a stack of restricted Boltzmann machines (RBMs), in a topology of 520:2048$\times$6:3804. DNN parameters are then fine-tuned to minimize cross-entropy. This is followed by 4 iterations of further fine-tuning for minimum phone error (MPE) or using the state level variant of the minimum Bayes risk (sMBR) training with updated alignment.

For IHM, results with speaker adaptation is provided. HMM-GMMs with LDA+MLLT+SAT provide the alignment and speaker feature level maximum likelihood linear regression (fMLLR) for DNN training. The DNN parameters are initialized with RBMs in a topology of 143:2048$\times$6:3710. DNN input features are comprised of 11 (+/-5) contextual 13 dimensional MFCC features with fMLLR applied.

For MDM, the weighted delay and sum beamforming is performed with BeamformIt [37] on 8 channel microphones from the circular array in the middle of the table. The automatic noise thresholding is disabled.

To reduce memory cost, the 30k 4-gram LM introduced in §5 is pruned. Table 5 shows the performance using different acoustic models and microphone channels. As shown, IHM SAT reduces the overall WER of HMM-GMM based system by 5.1% relative, while MMI did not reduce WER further. For DNN-HMM hybrid system however, speaker adaptation via fMLLR degraded the performance. The best overall WER of 42.0% on IHM is achieved with sMBR fine-tuning on DNN parameters without speaker adaptation. Therefore, fMLLR is not used in experiments with SDM or MDM hybrid system. Fine-tuning DNN with sMBR is effective for both SDM and MDM, achieving the best overall WER of 76.8% on SDM and 74.3% on MDM. Beamforming reduced the WER by 3.3% relative.

Table 5: *SWC "SA1" baseline (WER in %).*

		dev	eval	overall			
				Sub.	Del.	Ins.	WER
IHM	LDA+MLLT	50.9	51.8	35.9	8.9	6.4	51.3
	+SAT	48.7	**48.8**	34.4	8.1	6.3	**48.7**
	+MMI	48.8	49.1	34.4	8.8	5.7	48.9
	DNN	44.4	44.3	30.5	9.7	4.1	44.4
	+sMBR	42.0	**42.0**	29.5	7.6	5.0	**42.0**
	+fMLLR	48.1	48.1	32.9	11.4	3.8	48.1
	+sMBR	44.9	44.8	31.2	9.8	3.8	44.9
SDM	DNN	78.9	80.5	53.9	21.4	4.4	79.7
	+sMBR	76.4	**77.3**	39.1	35.5	2.2	**76.8**
MDM	DNN	76.0	77.9	53.3	18.2	5.5	76.9
	+sMBR	73.8	**74.9**	36.0	36.0	2.4	**74.3**

7. Conclusions

This paper presents the extended recordings for Sheffield Wargame Corpus, which is freely available for research use in the speech community, and which is designed for distant speech recognition work with multi-channel recordings. It includes unique ground truth annotation of speaker location. The extended corpus adds up to around 24.6h of multi-media and multi-channel data for natural native English speech. Four dataset definitions are provided for two different tasks: low resource adaptation of existing acoustic model and standalone training of acoustic model. A Kaldi recipe is provided for standalone training. Performance of baseline deep neural network systems for each task is illustrated. The WERs on the eval sets are above 40% for all systems, suggesting a high difficulty level in SWC corpora compared to other corpora. The WERs for SDM on eval set are all above 70%. Beamforming reduced the WER by 3-4% relatively. The best overall WER obtained is 42.0% for IHM, 76.8% for SDM and 74.3% for MDM.

8. Acknowledgements

This research was supported by EPSRC Programme Grant EP/I031022/1, Natural Speech Technology (NST). The results reported in this work could be accessed from `https://dx.doi.org/10.15131/shef.data.3119743`. More details and some samples of SWC recordings can be found at `http://mini-vm20.dcs.shef.ac.uk/swc/SWC-home.html`.

9. References

[1] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 285–290.

[2] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 5542–5546.

[3] T. Yoshioka, X. Chen, and M. J. F. Gales, "Impact of single-microphone dereverberation on dnn-based meeting transcription systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 5527–5531.

[4] K. Kumatani, J. McDonough, B. Rauch, D. Klakow, P. N. Garner, and W. Li, "Beamforming with a maximum negentropy criterion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 994–1008, July 2009.

[5] K. Kumatani, J. McDonough, and B. Raj, "Maximum kurtosis beamforming with a subspace filter for distant speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, Dec 2011, pp. 179–184.

[6] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, T. Hori, T. Nakatani *et al.*, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge," in *Proc. REVERB Workshop*, 2014.

[7] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[8] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, 2014.

[9] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 8609–8613.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 55–59.

[11] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 185–189.

[12] M. Ravanelli and M. Omologo, "Contaminated speech training methods for robust DNN-HMM distant speech recognition," in *INTERSPEECH*. ISCA, 2015, pp. 756–760.

[13] P. Zhang, Y. Liu, and T. Hain, "Semi-supervised dnn training in meeting recognition," in *2014 IEEE Spoken Language Technology Workshop (SLT 2014)*, South Lake Tahoe, USA, December 2014.

[14] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *ASRU 2015*. IEEE, December 2015.

[15] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, March 2016.

[16] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7398–7402.

[17] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning." IEEE Institute of Electrical and Electronics Engineers, April 2015.

[18] M. Ravanelli and M. Omologo, "On the selection of the impulse responses for distant-speech recognition based on contaminated speech training," in *INTERSPEECH*. ISCA, 2014, pp. 1028–1032.

[19] C. Fox, Y. Liu, E. Zwyssig, and T. Hain, "The sheffield wargames corpus." in *Proceedings of Interspeech 2013*, Lyon, France, August 2013.

[20] J. Wen, N. D. Gaubitch, E. Habets, T. Myatt, and P. A. Naylor, "Evaluation of speech dereverberation algorithms using the MARDY database," in *Proc. Intl. Workshop Acoust. Echo Noise Control (IWAENC)*, Paris, France, 2006.

[21] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Digital Signal Processing, 2009 16th International Conference on*, July 2009, pp. 1–5.

[22] R. Stewart and M. Sandler, "Database of omnidirectional and B-format room impulse responses," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 165–168.

[23] D. Pearce, H. günter Hirsch, and E. E. D. Gmbh, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *in ISCA ITRW ASR2000*, 2000, pp. 29–32.

[24] S.-K. Au Yeung and M.-H. Siu, "Improved performance of Aurora 4 using HTK and unsupervised MLLR adaptation," in *Proceedings of Interspeech 2004—ICSLP: 8th International Conference on Spoken Language Processing, Jeju Island, Korea*, 2004, pp. 161–164.

[25] H. G. Hirsch and H. Finster, "The simulation of realistic acoustic input scenarios for speech recognition systems," in *in Proc. ICSLP*, 2005.

[26] M. Wolf and C. Nadeu, "Channel selection measures for multi-microphone speech recognition," *Speech Communication*, vol. 57, pp. 170 – 180, 2014.

[27] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The DIRHA-ENGLISH corpus and related tasks for distant-speech recognition in domestic environments," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, Dec 2015, pp. 275–282.

[28] M. Lincoln, I. McCowan, J. Vepa, and H. Maganti, "The multi-channel wall street journal audio visual corpus (MC-WSJ-AV): specification and initial experiments," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, Nov 2005, pp. 357–362.

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*, Scottsdale, AZ, United States, Dec. 2015.

[30] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[31] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," 2003, pp. 364–367.

[32] C. Fox, H. Christensen, and T. Hain, "Studio report: Linux audio for multi-speaker natural speech technology." in *Proc. Linux Audio Conference*, 2012.

[33] T. Hain, V. Wan, L. Burget, M. Karafiat, J. Dines, J. Vepa, G. Garau, and M. Lincoln, "The ami system for the transcription of speech in meetings," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–357.

[34] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *ICSLP'02: Proc. of International Conference on Spoken Language Processing*, 2002.

[35] K. Richmond, R. Clark, and S. Fitt, "On generating combilex pronunciations via morphological analysis," in *Proceedings of ISCA Interspeech*, Makuhari, Japan, 2010, pp. 1974–1977.

[36] J. Novak, N. Minematsu, and K. Hirose, "WSFT–based grapheme–to–phoneme conversion: Open source tools for alignment, model–building and decoding," in *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, San Sebastián, Spain, 2012.

[37] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, Sept 2007.

Computers and Electronics in Agriculture 140 (2017) 123–138

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland

A. Binch [a,c], C.W. Fox [b,c,*]

[a] School of Computing, The University of Leeds, LS2 9JT, United Kingdom
[b] Institute for Transport Studies, The University of Leeds, LS2 9JT, United Kingdom
[c] Ibex Automation Ltd, S35 7DQ, United Kingdom

ARTICLE INFO

Article history:
Received 10 January 2017
Received in revised form 16 May 2017
Accepted 17 May 2017
Available online 9 June 2017

ABSTRACT

Automated robotic weeding of grassland will improve the productivity of dairy and sheep farms while helping to conserve their environments. Previous studies have reported results of machine vision methods to separate grass from grassland weeds but each use their own datasets and report only performance of their own algorithm, making it impossible to compare them. A definitive, large-scale independent study is presented of all major known grassland weed detection methods evaluated on a new standardised data set under a wider range of environment conditions. This allows for a fair, unbiased, independent and statistically significant comparison of these and future methods for the first time. We test features including linear binary patterns, BRISK, Fourier and Watershed; and classifiers including support vector machines, linear discriminants, nearest neighbour, and meta-classifier combinations. The most accurate method is found to use linear binary patterns together with a support vector machine.[1]

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Automated robotic weeding of grassland will improve the productivity of dairy and sheep farms while helping to conserve their environments. At present grassland weeding is typically performed in two styles. Tractor-mounted bulk spraying of selective herbicides is expensive due to the volume, and cost per unit of selective chemicals. Manual backpack-mounted spraying uses lower, targeted spot spray doses of generic herbicides such as glyphosate, but requires more expensive manual labour time. Precision robots (Blackmore et al., 2005; Bechar and Vigneault, 2016) present an opportunity to use similarly low and targeted doses of cheap generic herbicides as in the manual case, but at much lower cost as they can drive, detect and spray automatically without the need to pay manual sprayers by the hour. Precision robots could further eliminate the need for chemical herbicide altogether by destroying detected weeds with mechanical or other non-chemical methods. Data collected about weed locations by robots can be fed into geospatial weed mapping systems to enable ecological analyses.

Autonomous weeding robots must first detect weeds at a suitable resolution and accuracy. Machine vision provides many tools and algorithms for detection, with varying performances, and can be cheap to operate using consumer-grade cameras. Several previous studies have reported results of machine vision methods to separate grass from grassland weeds, typically *Rumex obtusifolius* (dockleaf). But as is common in early stages of artificial intelligence research, they each use their own datasets and report only performance of their own algorithm rather than presenting controlled trials testing methods against one another. As proof of concept studies, many used only small data sets, did not report confidence intervals on their accuracy rates, and have not yet tested methods across lighting and weather conditions which are known to affect many vision algorithms. It is well known in artificial intelligence that unintended author bias and bottom drawer effects can creep into studies when the same author both designs and tests an algorithm, so there is a need for independent validation. We present a large-scale (tens of thousands of images), independent study of all major grassland weed detection methods evaluated on a new standardised data set, under a wider range of environment conditions. This allows for a fair, unbiased, independent and statistically significant comparison of the methods for the first time.

1.1. Previous work

Previous studies can be grouped roughly into those which classify individual windows (patches) of images (e.g. tens of pixels square) independently of one another, and those which apply morphological operations to whole images (e.g. hundreds or thousands

* Corresponding author at: Institute for Transport Studies, 34-40 University Road, University of Leeds, Leeds LS2 9JT, United Kingdom.
E-mail address: C.W.Fox@leeds.ac.uk (C.W. Fox).

[1] This research was supported in part by the Innovate UK project IBEX2: Autonomous robot weed spraying for less favoured areas, Grant No. 131790.

http://dx.doi.org/10.1016/j.compag.2017.05.018
0168-1699/© 2017 Elsevier B.V. All rights reserved.

of pixels width and height). Window-based methods compute features of the windows such as spectra or texture descriptors, while whole-image methods try to isolate shapes via segmentation algorithms. Window-based methods include: (Dürr et al., 2005) used local binary pattern (LBP) texture features with a per pixel threshold Rumex/Grass classifier, under controlled artificial lighting conditions, to report between 87% and 97% accuracy on a test set of 941 images of 50 × 40 pixels. Ahmed et al. (2014) uses very large windows to obtain high accuracy, 98.5%, using texture features and support vector machine (SVM) classification. Whole-image segmentation methods include: (Gebhardt et al., 2006) segmented images into regions of similar texture then classified the shapes of these regions, reporting 71–95% accuracy in Rumex/(Grass and mixed herbs) classification under constant lighting conditions. Polder et al. (2007) used thresholded and segmented Fast Fourier Transforms (FFT) to detect Rumex in grass on 161 images, reporting 94% accuracy. Van Evert et al. (2009) used a similar setup to report accuracies 82–89% for Rumex/Grass. Siddiqi et al. (2009) used segmentation (erosion and dilation). Hiremath et al. (2012) use Gray Level Co-occurrence Matrix and Laws' filter mask texture features with linear discriminant analysis (LDA) and segmentation to report 90% Rumex/Grass. Hiremath et al. (2013) use Markov Random Field based texture features and segmentation to report a 97.8% accuracy on 92 images. A summary of the key properties of each method is given in Table 1 (plant types are R = Rumex, G = Grass, U = Urtica, H = mixed herbs).

Study	Method	Classes	Reported accuracy	Number of test images	Illumination	Window size
Dürr et al. (2005)	LBP + threshold	R/G	87–97%	941	artificial	50 × 40
Gebhardt et al. (2006)	Segment + shape	R/(G + H)	71–95%	3681 windows	Constant	n/a
Polder et al. (2007)	FFT + segment	R/G	94%	161 images	Constant	8
Van Evert et al. (2009)	FFT + segment	R/G	82–89%	56 images	Constant	8
Siddiqi et al. (2009)	Segment	R/G	89%	240 images	Constant	n/a
Hiremath et al. (2012)	GLCM + LDA + segment	R	90%	92 images	Constant	n/a
Hiremath et al. (2013)	MRF segment	R/G	97.8%	92 images	Constant	n/a
Ahmed et al. (2014)	LBP + SVM	R	98.5%	400 images	Varied	320 × 240

From the table, it is clear that making a fair comparison of these algorithms is difficult or impossible from publicly available data. Each study uses its own data sets, comprising completely different images and conditions. In many cases the separation of training and test data is not clear, with studies reporting best results having optimised parameters over the same test set used in the final result, rather than making a clean train/test separation. It is well-known (Friedman and Hastie, 2001) that optimising parameters to the test set tends to yield over-optimistic results compared to performance on new data. Some studies do not describe the variation in the lighting conditions, but are assumed to be constant conditions because they use small numbers of test images collected, presumably, on the same day. Window-based methods have used different window sizes, while whole-image segmentation methods make use of data from across the whole image to classify each local pixel, which is hard to meaningfully evaluate against windowed results. Window size is important because it represents a fundamental trade-off between detection accuracy and spatial resolution. A large window contains more information which will yield high accuracies, but at the cost of a lower spatial resolution, for example in determining what area of ground to spray with herbicide.

Table 1
Dataset sizes.

Window size	Dataset type	Epoch set sampled	Number of windows
28^2	Hyper-training	Training	10,000
28^2	Hyper-test	Test	1000
28^2	Training	Training	200,000
28^2	Test	Test	20,000
64^2	Hyper-training	Training	2000
64^2	Hyper-test	Test	200
64^2	Training	Training	40,000
64^2	Test	Test	4000

In our native UK grassland, Rumex is not the only common weed and almost always co-occurs with similar populations of *Urtica diotica* (stinging nettle). As such, any automated grassland weeding system needs to work with both Rumex and Urtica together. If Rumex only was precision sprayed, then a selective bulk spray for Urtica would still be needed which negates the utility of the Rumex precision system, as combined Urtica and Rumex selective chemicals are available. Previous work on automatic detection of Urtica in grassland has relied on non-visual spectral methods including near infra-red and full hyper-spectra (Pantazi et al., 2016; Moshou et al., 2013; Vrindts and Baerdemaeker, 1999), but these sensors are more costly than simple visual cameras. Urtica has smaller leaves than Rumex which makes it harder to detect with machine vision alone, in particular some Rumex detection methods rely entirely on obvious features of the large, smooth Rumex leaves, which may not carry over to the Urtica case. However Urtica has distinctive jagged edges on its leaves which suggest that methods based on such local shapes (rather than texture) features may be useful for detection.

With the exception of Ahmed et al. (2014), all the systems in Table 1 rely on vertical camera angles, i.e. cameras mounted to look directly downwards at the ground. This simplifies recognition as there is no perspective, and all parts of the ground look the same. However this imposes physical limitations on precision robots, which must either mount a camera physically outside the robot's base footprint, or inside the body of the robot looking directly under its base. Much UK grassland is found in less-favoured areas, including hilly and rocky terrain such as sheep farms. These terrains often include obstacles which robots must navigate around, and such navigation is complicated by physical extrusions beyond robot platform bases such as cameras on arms or beams. Similarly, robots designed for these terrains may need heavy, protective bases which prevent cameras or sprayers from being mounted directly downwards from them. To generalise oper-

ation beyond flat grassland to cases such as these, it is more convenient and lower-cost to use more standard camera mountings on top of the robot body, with cameras facing forwards and tilted down, as in Ahmed et al. (2014). While this is a more robust physical solution, it makes the machine vision problem harder as it must now deal with perspective. Further, the previous studies all use clean images taken by stationary cameras to ease recognition. In practice, precision robots operating in generalised terrains will be moving at speed, capturing images during motion. It is not practical to stop every time an image must be taken. While camera stabilisation systems are available at a cost, grassland and especially hill farmers typically require lower budget solutions than arable farmers, so it is of interest to test algorithms on data collected from similar moving robot platforms as would be used in practice, which can include motion blur.

Recent work has begun to explore the use of 3D lidar based sensing and detection of Rumex in grassland (ŠEATOVIĆ, 2008; Šeatović et al., 2010; Anken et al., 2010). While a similar independent evaluation of such data would also be of interest, it is beyond the scope of the present machine vision study. Also beyond our scope are non-visual approaches to weed detection including hyper-spectral (Pantazi et al., 2016; Moshou et al., 2013; Vrindts and Baerdemaeker, 1999), and chemical sensing methods (Peteinatos et al., 2014). Our scope of detection of weeds in grassland is a particular sub-field of automated weed detection in general, which has developed a wider range of methods applicable to simpler cases of crops and weeds growing in flat, row-crop settings, which can typically simplify the task by initial segmentation into green and brown discrete plant and soil regions, unlike the grassland case where everything is green (Cope et al., 2012; Hamuda et al., 2016; McCarthy et al., 2010; Tillett et al., 1996; Vibhute and Bodhe, 2012).

1.2. Data and algorithm requirements

To make a fair and useful comparison between the different algorithm types proposed for UK, less-favoured area grassland weed recognition, and to extend the robustness of previous studies, the following requirements were taken into account. 1. Data should be clearly split into training and test sets. 2. Only a single run should be allowed of each algorithm on the test set. 3. The algorithms should be implemented and tested independently of their original proposers. 4. testing should be on Urtica as well as Rumex. 5. Data should be collected from a moving, robust platform, with cameras mounted on top of its body and pitched downwards from the plane. 6. Data size should be in the order of thousands of image windows. 7. Classification should be performed on standard sized windows, including for morphological methods which should be restricted to run on the same windows as feature-based methods. 8. Windows should be of a suitable spatial size and resolution to enable precision spraying. 9. Data should be collected over a representative variety of different days, illumination, and weather conditions.

Taken together, these requirements are more challenging than settings used in the previous studies. An independent evaluation should not seek to 'sell' any one algorithm with high rates, and should not shy away from reporting low accuracies when they occur. This helps to avoid any publication bias (Rosenthal, 1979) which may have acted as a filter on previous tests. Due to interactions between the requirements and differences in data types, we do not re-implement algorithms directly but instead use similar or closely related methods. This is required in particular for the morphological approaches which do not transfer directly to window classification, such as the use of watershed segmentation to represent region based methods previously run on whole images. We test classifiers that are based on and inspired by the collection of previous studies as a whole rather than directly re-implementing and competing between them.

2. Methods

The objective of the experiments is to report, to a statistically significant level, the classification performance of various classification methods for grass vs weed detection, i.e two-way classifications representing spray/no-spray decisions for a general herbicide. In general this is distinct from the problem of recognising individual weed species. A 'classification method' or 'method' means a combination of one feature type with one classifier type. Care must be taken to avoid contamination of classifier training with any information from test data.

2.1. Image acquisition and pre-processing

Test plots of weeds in grass were constructed on a dairy grassland farm in South Yorkshire, UK. Slabs of Urtica approximately 0.2 m squared were extracted from working fields and transplanted into a 3 m squared trench (Fig. 2a). This process was repeated for Rumex. Transplanting real slabs from areas of the working farm ensures maximum realism and avoids problems of growing the weeds artificially, which could lead to unrealistic soil backgrounds in images. In particular, the transplanted slabs also contain grass, soil, rocks, and other surface features of the real working grassland farm, though only in 0.2 m squared slabs which human transplanters considered to be fully 'sprayable'. To make this decision, the human transplanters were instructed to collect only slabs which they would be happy to completely spray with a manual backpack sprayer if they were being employed to manually destroy weeds. The weedy turfs were watered daily for two weeks to allow the plants to stabilise before data collection. Plots were located in a region of the farm that is in direct sunlight (not in shade) throughout the day.

Stereo pair images[2] were acquired in 1080HD from each plot using auto-focusing cameras[3] (C920, www.logitech.com) mounted on a tracked robot as in Fig. 1. The cameras were fixed to the robot's left side, facing out at right angle yaw to the direction of travel, and at $\pi/8$ radians (22.5 degrees) pitch down, to give a view over a roughly 1 m square area of ground. The robot drove in circles around the plots whilst capturing pictures, randomised between 0 m and 1 m from the edge of the plots (Fig. 2b). This guarantees an equal balance of lighting and shadow angles in the data, because each drive around the plot contains images of the plot from all ground angles. The size of the plots, robot, and camera positions were selected such that the plot contents fill the images. This setup removes the need for manual annotation of weed classes in images, as we are assured that every part of every image is full of a weed class. One image was taken every second.

[2] Only the mono, left camera images are used in this study. Stereo images were captured for use in future comparison studies and are also made available as part of the dataset.

[3] After evaluating several industrial and consumer cameras, a pilot experiment determined that the C920 is sufficient and lowest cost for our purpose, having sufficient depth of field to cover the region of interest, and default shutter speed sufficient to give sharp images when driving the robot at around 4 km/h. This camera's auto-focus is also useful as we operate on somewhat uneven terrain, sufficient to sometimes blur images with a static focus. The auto-focus explores and adjusts different settings of the focus and chooses the one which minimises the blur (maximises entropy) near the center of the image, before taking and saving the final image. 1080HD (=2megapixels) was used as we assume that real-time processing above this resolution is difficult with currently available on-board hardware. Manual inspection of the pilot images suggested that distortion from depth-of-field and motion-blur effects were usually small compared to other factors such as perspective distortion. The camera also makes adjustments for lighting but we override these with our own colour normalisation as detailed below.

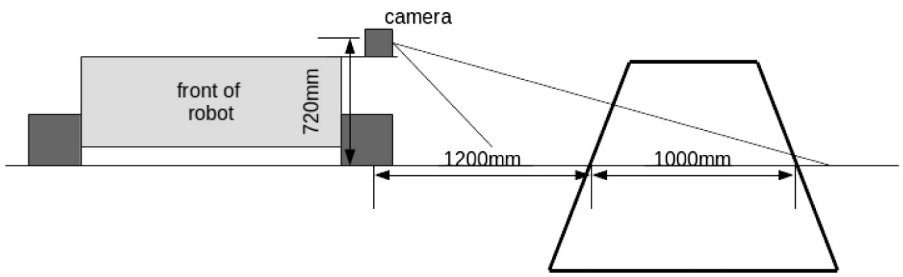


Fig. 1. Camera geometry. Showing position of camera on tracked robot (viewed from the front), facing sideways. The thick black square shows the area on the ground, in perspective, used later in the perspective transform.

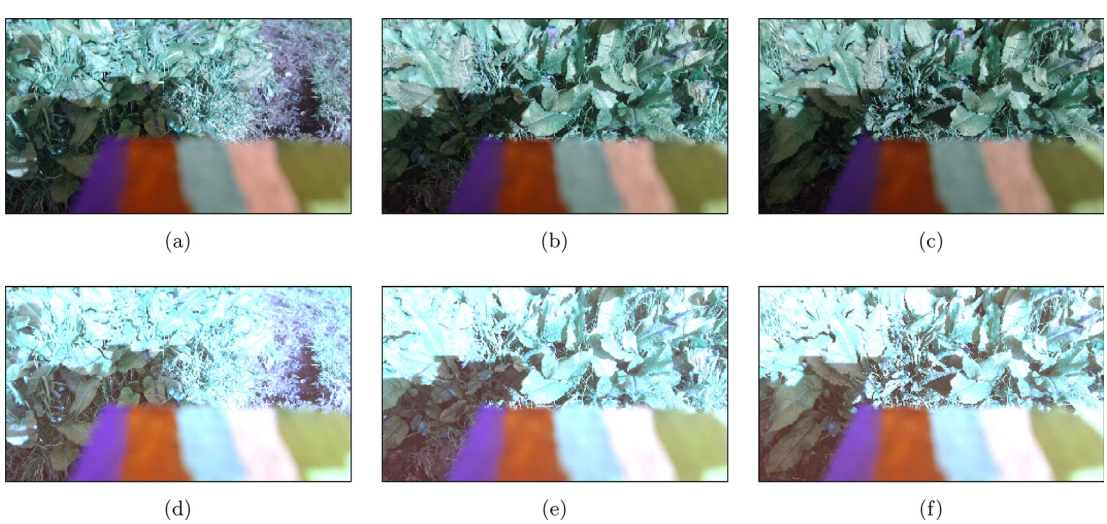
(a) (b)

Fig. 2. Image acquisition. (a) Weed plot construction. (b) Images were obtained using cameras mounted to the side of a tracked robot. The robot repeatedly made revolutions of each plot whilst taking pictures.

Image acquisition was arranged into epochs, where a single epoch consisted of making repeated revolutions of each plot for a period of ten minutes (yielding twenty minutes of weed image acquisition from the two weed types together), followed by fifteen minutes of grass image acquisition. The open grassland contained a mixture of grasses including *Lolium multiflorum, Festuca pratensis, Phleum pratense*, and *Holcus lanatus* with some *Trifolium repens* (clover). Approximately half of the epochs were acquired under overcast weather conditions, whilst the other half was acquired under bright or sunny weather conditions. Data capture was staged over four days, with 10 epochs in total captured at random times of day from sunrise to sunset during May 2016. Images were inspected manually and a small portion (<0.1%) removed due to recording problems. Approximately a third of a terabyte of usable image data was thus acquired in total, to our knowledge this is the largest and most multi-conditioned data set of its kind.

Images were pre-processed in three steps: colour calibration, perspective dewarping, and windowing.

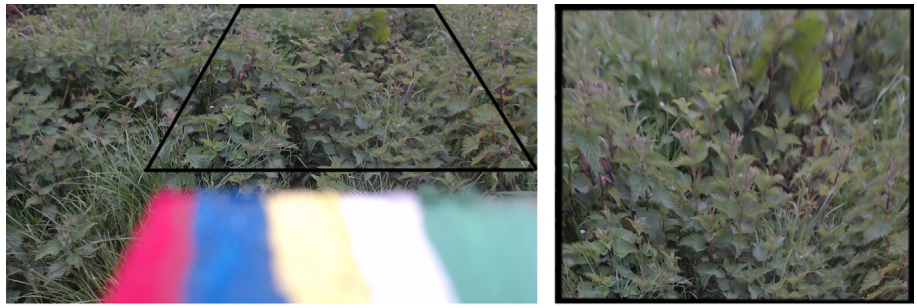
In colour calibration, images are transformed to compensate for variations due to lighting and weather conditions. Colour calibration acts to colour-standardise images (to a certain extent) to simplify classification. It was performed using a colour bar present in all images, recorded as part the robot camera frames (Fig. 3). The colour bar was composed of five coloured squares (red, blue, yellow, white and green) using standard colours of paint. A single reference image was selected to calibrate all other images to. A measure of the blue, red and green light intensities from each coloured square on the bar was obtained as the mean value of the red, green and blue channels within the square. For each channel a vector of intensities was constructed with values for each coloured square, yielding the vectors $\boldsymbol{b}_r, \boldsymbol{g}_r$ and $\boldsymbol{r}_r$ for the blue, red and green channels, respectively. The subscript r indicates that



(a) (b) (c)

(d) (e) (f)

Fig. 3. Colour calibration. Figures a, b and c are raw images of dock leaves before any pre-processing has taken place. Figures d, e and f are colour calibrated versions of figures a, b and c, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 4. Perspective normalisation. (a) Raw image from left camera, with grid annotation showing 1.16 m width × 1 m depth on the ground. (b) Affine transformation to remove perspective, grid annotation showing locations of the same rectangle after perspective transformation into a 600 × 700 pixel image.

these values are from the reference image. Repeating this procedure for a comparison image c (i.e. an image that is to be colour calibrated) gives vectors $\boldsymbol{b}_c, \boldsymbol{g}_c$ and $\boldsymbol{r}_c$. A linear relationship between the intensities of the blue, red and green channels is assumed, measured from the reference image and the intensities of the blue, red and green channels measured from the comparison image. Given this assumption, the parameters for each channel, for example blue, are obtained as $\hat{\boldsymbol{\beta}}(b)$,

$$\hat{\boldsymbol{\beta}}(b) = (\mathbf{X}(b)^{\mathbf{T}}\mathbf{X}(b))^{-1}\mathbf{X}(b)^{\mathbf{T}}\mathbf{b}_c, \tag{1}$$

where

$$\mathbf{X}(b) = \begin{pmatrix} \text{red}_r(b) & 1 \\ \text{blue}_r(b) & 1 \\ \vdots & \vdots \\ \text{green}_r(b) & 1 \end{pmatrix}, \quad \mathbf{b}_c = \begin{pmatrix} \text{red}_c(b) \\ \text{blue}_c(b) \\ \vdots \\ \text{green}_c(b) \end{pmatrix}, \tag{2}$$

where the values in the left hand column of the matrix $\mathbf{X}(b)$ is the vector $\boldsymbol{b}_r$. These parameters are used to colour calibrate the blue channel of the comparison image as:

$$I'_c(b) = \frac{I_c(b) - \hat{\boldsymbol{\beta}}(b)[1]}{\hat{\boldsymbol{\beta}}(b)[2]}, \tag{3}$$

where the term in square brackets indexes a value in $\hat{\boldsymbol{\beta}}(b), I$ is the original intensity and I' is the adjusted intensity. Performing these operations for the green and red channels also yields a fully colour calibrated image.

Perspective normalisation was performed via the projective transform shown in Fig. 4a to those in Fig. 4b. This maps a 1.16 m width by 1.0 m depth ground area into a 700 pixel width by 600 pixel height image, whose geometry is identical to that of a vertical, overhead camera as used in previous studies. However the image is not exactly the same as that of an overhead camera due to the three dimensional structure of the plants. In particular, tall plants at the front of the image are inflated in size because they are warped as if they were further back. A key research question asks whether this will have a detrimental effect over pure overhead imaging.

Finally the dewarped image was split into regular 28 × 28 square or 64 × 64 square pixel windows for classification, corresponding to 46 mm or 106 mm squares of ground space respectively. Windows were contiguous and non-overlapping, and were stored for analysis. (These sizes were chosen to be around the scale of a single spray target radius, or ground size of a single weed or clump of weeds. 64 is a power of two which speeds up FFT based methods; 28 rather than 32 is chosen for the smaller window size to enable future comparisons with neural network methods, where 28^2 pixel windows are a common standard for historical and technical reasons (Hinton et al., 2006).)

2.2. Dataset definitions

To avoid test data contaminating training processes, the set of epochs was first partitioned into training epochs and test epochs. This prevents, for example, classifiers from learning to recognise the lighting conditions rather than the weed types. Partitioning was performed manually to ensure a balance of weather conditions and weed types in each partition.

After partitioning the epochs, we defined training and hyper-training datasets by sampling random windows from the training epochs, and test and hyper-test data sets by random sampling windows from the test epochs. (Separate hyper-training and hyper-test sets are used during hyper-parameter optimisation before training proper, to avoid contamination by test set data, as described in Section 2.5.) A data set consisted of a set of images, where half of those were images of grass and the other half were images of weeds. A data set could contain an individual weed type (Rumex or Urtica) or a mixture of weed types (Rumex and Urtica). Similarly, a data set could contain images obtained under individual weather conditions (overcast or sunny) or a mixture of weather conditions (overcast and sunny). The data set sizes are shown in the table below,

These sizes are still small compared with the total amount of raw data collected, but are orders of magnitude larger than data used in previous studies.[4] The test set size was chosen to yield significant confidence in the results, while the training set size was selected to enable all the software implementations to train within one day of processing on a single 3 GHz Intel core. Test and hyper-test datasets are 10% size of their corresponding training and hyper-training sets. The 28^2 sets are set to contain 5 times as many images as the 64^2 sets so that they contain the same amount of total

[4] While the number of images is large, the number of actual plants imaged is smaller, because the many images are obtained by driving many times around the same plots of plants. However it is unlikely that images of the same plant are ever identical due to several factors. First, images were taken over several days, with collection days spaced several days apart. This allows the plants time to grow and move around between epochs. Second, within epochs, the plants are outdoor and exposed to wind, which moves their leaves constantly. Third, epochs are taken at different times of day, with different weather and light conditions. Fourth, the radius of the robot around the plots was randomised to ±0.5 m from the center line on each drive, ensuring different views. Fifth, the angles of the cameras shifted slightly between epochs during robot maintenance; this was deliberately not reduced, to introduce addition variation. Sixth, images were taken once per second, and it is unlikely that the robot would be at exactly the same angle around the plot at any two imaging moments. We assume that the additional variability from using more individual plants would be small compared to the variability introduced by these factors.

pixel data ($28^2 \approx 64^2/5$), to allow results to be meaningfully compared across window sizes.

2.3. Feature extraction

The windowed images contain 64^2 or 28^2 pixels of RGB data, which are too large to use directly as input vectors to most classifiers. Therefore, features were first computed from the data, as in the previous studies. The selection of features used is detailed below and was selected to represent most previously proposed feature choices. All features ran on greyscale versions of the windows (obtained as the mean of the RGB channels), which is justified as all images are primarily all the same shade of green (unlike the case in detection studies of arable crop weeds in brown soil).

2.3.1. Fourier transform

The Fourier Transform (Weisstein, 2015) represents an image in the basis of its orthogonal harmonic frequency components. For digital images the discrete Fourier transform (DFT) is used, whose basis is a set of two dimensional harmonics large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image. For a square image of size $N \times N$, the two-dimensional DFT is given by:

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)e^{-i2\pi(\frac{ki}{N}\frac{lj}{N})}, \tag{4}$$

where $f(i,j)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $F(k,l)$ in the Fourier space. The basis functions are two dimensional sinusoidal waves of increasing spatial frequencies, i.e. $F(0,0)$ represents the DC-component of the image which corresponds to the average brightness and $F(N-1,N-1)$ represents the highest frequency. The absolute values of the DFT yield the image's magnitude of frequency spectrum which is used as the feature vector for classification. We denote this method of feature extraction as FT. The Fourier transform is illustrated in Fig. 5, which shows that broad leaves tend to contain stronger low spatial frequencies while grass' thin blades give rise to higher spatial frequencies.

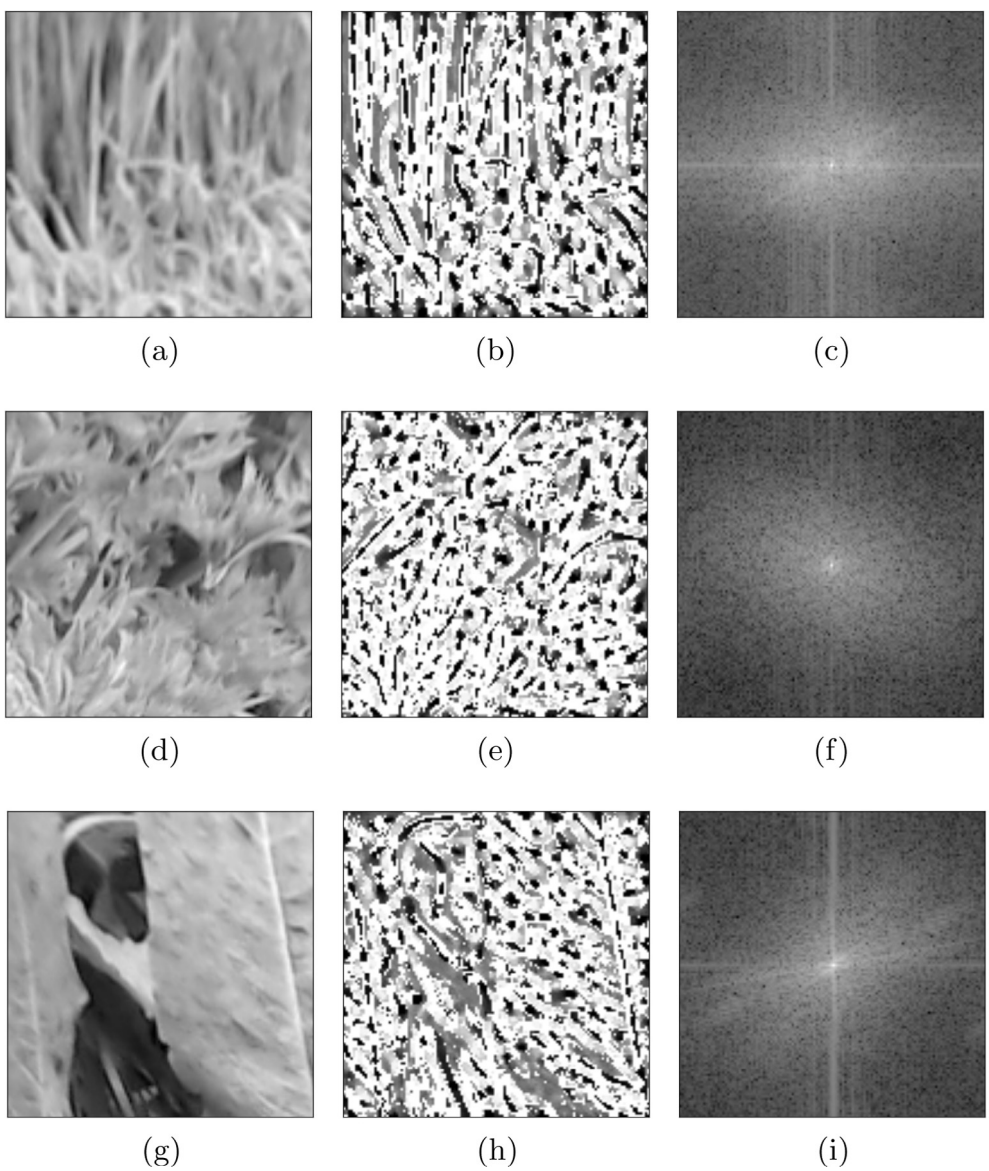


Fig. 5. Feature extraction. Figures a, d and g are fully pre-processed images containing grass, Urtica and Rumex leaves, respectively. Figures b, e and h are the local binary patterns (LBPs) of images a, d and g, respectively (with $n = 24, r = 4$). Figures c, f and i are the magnitude of frequency spectra of images a, d and g, respectively.

2.3.2. Local binary patterns

Local binary patterns (LBP) are a texture description feature (He and Wang, 1990). Local means they are computed on local sub-windows of an image only, as a function of a center pixel and either its immediate or r pixel radius neighbours; binary means that the feature vector is binary, with each feature classed as either present or absent. Windows are converted to greyscale, then for each pixel $x_{i,j}$ in the window, LBP computes an 8-element binary vector,

$$[(x_{i,j} > x_{i,j+r}), (x_{i,j} > x_{i-r,j+r}), (x_{i,j} > x_{i-r,j}), (x_{i,j} > x_{i-r,j-r}), (x_{i,j} > x_{i,j-r}), (x_{i,j} > x_{i+r,j-r}), (x_{i+r,j} > x_{i,j}), (x_{i,j} > x_{i+r,j+r})].$$

There are $2^8 = 256$ possible values of this feature vector, with oriented edge and corner detection present as special cases. LBP computes feature vector for each pixel in the window, then computes a 256-point histogram of the obtained values over the window. The shapes of these histograms are considered to be characteristic of the texture classes and are given as input to classifiers. As well as using the eight near neighbours as above, a hyper-parameter n_{points} can also be used to generalise to other numbers of quantised comparison points equally spaced on a circle around the center. (Many other variations on the LBP concept have also been proposed (Guo and Li, 2012) but are beyond the scope of the present comparison study.)

2.3.3. Interest points and k-means

The above features treat every pixel in the window as equally important, and are considered to represent texture-like properties. An alternative approach is to locate only 'interesting' points within the window and base classification on these. In particular, weeds such as Urtica have many distinctive jagged corners which might form useful points on which to base classification. Interest point method contain two feature extraction steps prior to classification. First, interest points are located; second, the local region at each of these points is used to extract a feature descriptor. Points are considered interesting if they contain a mixture of colours and can be uniquely located, i.e. a corner is interesting because there is only one location where it exists, while an edge is less interesting because it exists along a line of locations. Feature classes used to describe these points are usually wavelet-like, combining color, frequency and size information.

Precise definition of useful interest point detectors and descriptors normalising these properties has been and remains an active area of machine vision research (Patel et al., 2014), but the present study arbitrarily selects the state-of-the-art Scale-Invariant Center-Surround Detectors (CenSurE) (Agrawal et al., 2008) interest point detector and the Binary Robust Invariant Scalable Keypoints (BRISK) (Leutenegger et al., 2011) descriptor to represent the general class of methods. CenSurE finds an approximation to the set of corner-like points defined by,

$$\{\lambda_d(H(x)_{i,j}) > t, d \in \{1,2\}\}, \tag{5}$$

where λ_d are the two eigenvalues of the Hessian matrix H of the image x at each pixel i,j, and t is a threshold. Both eigenvalues are maximal at corners of any rotation. Rather than compute this computationally intensive (due to eigenvalue finding) test for every pixel, CenSurE first performs a faster pre-screening step, using a set of scaled filters to approximate the Laplacian (total curvature) at each pixel, then only computing the Hessian test at local maxima of this curvature (Agrawal et al., 2008).

BRISK descriptors are similar to the LBP vectors above, but using pixel intensity comparisons,

$$f_n = \left(x_{i,j} > x_{i'_n j'_n}\right), \tag{6}$$

at a larger set of 256 offsets $\{a_n, b_n\}$ giving $(i'_n = i + a_n, j'_n = j + b_n)$. Unlike LBP, these offsets are not equally spaced around a circle, but may form any arbitrary pattern. Together with the larger number of points, this may capture potentially higher-order information than in LBP. Standard values of the offset patterns are used as provided in Leutenegger et al. (2011).

When BRISK feature vectors have been computed for all images in the training set they are passed to the k-means algorithm which clusters them into K regions in feature space, where feature space is a finite p-dimensional vector space with each dimension representing a BRISK feature of an image. Initiating K random clusters we firstly calculate the Euclidean distance between the ith BRISK vector $\mathbf{x}_i$ and the kth cluster $\mathbf{C}_k$

$$d_{i,k} = \left[\sum_{j=1}^{p}(\mathbf{x}_{ij} - \mathbf{C}_{kj})^2\right]^{1/2}. \tag{7}$$

After performing this operation for all clusters we can then assign the BRISK vector to the cluster that minimises d (Eq. (7)). This operation is then performed for each BRISK vector. Knowing the members of each group we can now compute the new centroid of each group based on these new memberships. New centroids are the average coordinates among new members,

$$\mathbf{C}_k = \frac{\sum_{n=1}^{N_k}\mathbf{X}_{n1}}{N_k}, \ldots, \frac{\sum_{n=1}^{N_k}\mathbf{X}_{np}}{N_k}, \tag{8}$$

where N_k is the number of BRISK vectors in cluster k. This whole process is then repeated until the BRISK vectors cease to move groups (i.e. until the computation of the k-means clustering has reached stability).

Clustering observed BRISK vectors in this way defines K discrete types of BRISK feature. For any given image, we may now extract each of its interest points, compute a BRISK descriptor at these points, then replace each of their BRISK descriptors with one of these quantised types. This allows us to then count how many c_k of each discrete BRISK types $k = 1 : K$ appear in the image. The vector of these counts, $\{c_k\}_{k=1:K}$ is then used as a feature descriptor of the whole image.

We refer to this feature as $B > K$, (for 'BRISK followed by k-means').

2.3.4. Watershed segmentation

We wish to test window-based features against region-growing type methods as proposed in previous studies. To make a fair comparison it is necessary to substitute pure region growing with a similar but window-based method. Otherwise the region growing methods could be accused of accessing more data to make classifications of each region, from the whole image, rather than just from its local window. For this purpose, we use a watershed method as a close substitute. Watershed segmentation (Vincent and Soille, 1991) was originally developed for the purpose of separating touching objects in an image rather than for classification, but may also be used as a region-growing type classifier. The watershed transform finds 'catchment basins' and 'watershed ridge lines' in an image by treating it as a surface where light pixels are high and dark pixels are low. Segmentation using the watershed transform works better if a human operator can first identify, or 'mark', pixels from foreground objects and background locations. The marker-controlled watershed segmentation used in the present study follows a multi-step procedure.

Given a grey-scale image as input, we apply Otsu's thresholding (Vincent and Soille, 1991) to segment the background from the foreground. Then we compute the Euclidean distance transform which computes the Euclidean distance to the closest zero (i.e. background pixel) for each of the foreground pixels. Doing this yields the distance map d. Next we apply a function $f(d, d_{\min})$ that finds the peaks (local maxima) in the distance map, and ensures

that we have at least a $d_{\min}$ pixel distance between each peak. Then we apply a connected component analysis using 8-connectivity to the output of f, the output of which gives us our markers, which we then feed into the watershed function. The watershed function returns a matrix of labels, an array with the same width and height as the input image. Each pixel value has a unique label value. Pixels that have the same label belong to the same object. For a given image, we then count the number of *unique* labels or *segments*. Performing these operations for multiple values of $d_{\min}$ yields multiple counts of segments for a given image, and thus a feature vector for classification. We denote this method of feature extraction as W.

2.4. Classifiers

The following classifiers were used to represent previously proposed architectures. Each classifier (SVM, LDA, NN) can take as its input any feature type obtained from windows (LBP, FT, B⟩K, W). This yields 12 distinct classification methods, LBP⟩SVM, FT⟩SVM, B⟩K⟩SVM, W⟩SVM, LBP⟩LDA, FT⟩LDA, B⟩K⟩LDA, W⟩LDA, LBP⟩NN, FT⟩NN, B⟩K⟩NN and W⟩NN. For example, LBP⟩SVM means that we pass the local binary pattern feature vector as input to a support vector machine, whilst FT⟩LDA means that we pass the image's frequency spectrum magnitude feature vector as input to linear discriminant analysis.

2.4.1. Support vector machines

A Support Vector Machine (Vapnik, 2013) models the a classification problem as finding a non-linear partition of the feature vector space into classes (e.g. grass or weeds), formed as a linear partition of a higher-dimension space formed by non-linear high-dimensional projection of the feature vectors. To understand how SVMs work it is useful to first briefly describe the support vector classifier (SVC). The SVC separates images into their classes by finding the *linear affine* hyper-plane that maximises the distance (known as the margin M) between the two image classes in feature space. Observations that fall on the boundaries of the margin are the support vectors.

The linear affine hyper-plane is defined by the following inner product,

$$\mathbf{b} \cdot \mathbf{x} + b_0 = 0, \quad b_0 \neq 0, \tag{9}$$

where $\mathbf{x}$ is a p-dimensional training image feature vector with associated weights $\mathbf{b}$. Now consider a set of n p-dimensional training image feature vectors, $\mathbf{x}_i$, each with an associated class label $y_i \in \{-1, 1\}$. Introducing new hyper-parameters; n ϵ_i values (known as slack values) and a hyper-parameter C (known as the budget), then we wish to maximise M across $b_1, \ldots, b_p, \epsilon_1, \ldots, \epsilon_n$ such that

$$\sum_{j=1}^{p} b_j^2 = 1, \tag{10}$$

$$y_i(\mathbf{b} \cdot \mathbf{x} + b_0) \geqslant M(1 - \epsilon_i), \quad \forall_i = 1, \ldots, n, \tag{11}$$

$$\epsilon \geqslant 0, \quad \sum_{i=1}^{n} \epsilon_i \leqslant C, \tag{12}$$

where C, the budget, is a non-negative 'tuning' hyper-parameter and hyper-parameters ϵ_i allow the individual observations (i.e. training images) to be on the wrong side of the margin or hyper-plane. C collectively controls how much the individual ϵ_i can be modified to violate the margin.

SVMs are an extension of SVCs that results from a non-linear enlargement of the feature space through the use of functions known as kernels. This enlargement of the feature space means that observations from different classes can be separated in many more ways than they could be otherwise. To obtain the SVM, firstly we note that it is possible to show that a linear support vector classifier for a particular observation can be represented as a linear combination of inner products for the subset ℓ of training observations that represent the support vectors,

$$f(\mathbf{x}) = b_0 + \sum_{i \in \ell} \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle, \tag{13}$$

where α_i are the coefficients. Replacing the inner product $\langle \mathbf{x}_i, \mathbf{x}_k \rangle$ with a more general inner product 'kernel' function $K = K(\mathbf{x}_i, \mathbf{x}_k)$, we can modify the SVC representation to use non-linear kernel functions. One example is the radial (RBF) kernel,

$$K(\mathbf{x}_i, \mathbf{x}_k) = \exp\left(-\gamma \sum_{j}^{p} (\mathbf{x}_{ij} - \mathbf{x}_{kj})^2\right), \quad \gamma > 0. \tag{14}$$

Intuitively, the γ parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The γ parameter can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

The algorithmic solution for the SVM is one that finds optimal values for the coefficients α and the slack variables ϵ_i. Typically gradient decent algorithms are used. The hyper-parameters C and γ are set/optimised by the user.

Finally, a test image is classified according to whether its feature vector $\mathbf{x}^*$ results in a positive or negative sign when passed into the function $f(\mathbf{x}^*)$. Note that feature vectors were normalised before being passed into the support vector machine. We denote this classifier as SVM.

2.4.2. Linear discriminant analysis

As with SVCs/SVMs, Linear Discriminant Analysis (McLachlan, 2004) models the classification problem by creating a feature space with a dimension for each feature. However, in LDA, observations from 2 separate classes are assumed to be sampled from 2 separate multivariate Gaussian distributions in feature space with different means but the same covariance matrix. Given those assumptions we have a linear hyperplane perfectly separating the means of the 2 distributions. This means that any observation that is situated above the hyperplane has a higher probability of being a sample from the Gaussian whose mean is situated above the hyperplane than being a sample from the Gaussian whose mean is located below the hyperplane.

To explain LDA in some more detail, firstly we write Bayes' rule for the classification problem

$$P(i|\mathbf{x}) = \frac{P(\mathbf{x}|i)P(i)}{\sum_j P(\mathbf{x}|i)P(i)}, \tag{15}$$

where the likelihood function $P(\mathbf{x}|i)$ gives the probability that the observation $\mathbf{x}$ is a sample from the Gaussian representing the class i and $P(i)$ is the prior probability of the class i. From Bayes' rule and the assumptions outlined we can derive the linear discriminant analysis formula

$$f_i = \mathbf{m}_i \mathbf{C}^{-1} \mathbf{x}_k^T - \frac{1}{2} \mathbf{m}_i \mathbf{C}^{-1} \mathbf{m}_i^T + \log(p_i), \tag{16}$$

where $\mathbf{m}_i$ is a vector containing the mean of each feature for the class i and $\mathbf{C}$ is the pooled within group covariance matrix which is a weighted mean of the covariance matrix $\mathbf{C}_i$ for each class. For a total of n observations, N classes and n_i observations in each class $\mathbf{C}$ is

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{N} n_i \mathbf{C}_i. \tag{17}$$

Then we simply assign a test image k to group i that has maximum f_i. We denote this classifier as LDA.

2.4.3. 'Nearest Neighbour' classifier

Nearest neighbour is a very simple classifier, used here to provide a baseline to compare with the two more sophisticated classifiers (SVM and LDA) above. In its training phase, the nearest neighbour classifier computes the median feature vector for each class (grass or weeds). A test image is then classed as grass if its feature vector minimises the sum of absolute errors between it and the median feature vector computed from grass images. Likewise a test image is classed as containing weeds if its feature vector minimises the sum of absolute errors between it and the median feature vector computed from images containing weeds. We denote this classifier as NN.

2.5. Hyper-parameter optimisation

Some of the classifiers and features have hyper-parameters which define how training is computed. Previous studies have mostly reported the best obtained results of methods on test sets, and stated the hyper-parameter values used to give them. However it is unclear whether the hyper-parameters in these cases have been set in advance of the evaluation on the test sets, or if they have been fit to the test data by running method multiple times on the test data and reporting only the best result.

To avoid this potential bias, careful use was made of hyper-training and hyper-test datasets, independent of both training and test datasets, to select hyper-parameters in advance of the main training and test phases. Hyper-parameters were optimised on these sets, so that each system only saw the final test set only once for its reportable evaluation score.

In theory, hyper-training could be performed by training many versions of a classifier on the full training dataset, then scoring them against a hyper-test set, and selecting the best performer. (The hyper-test sets are sometimes known as 'validation sets'.) However this requires running time-consuming training many times. So given the large ratio of data to compute resources available for this study, a smaller hyper-training dataset, of 5% size of the full training dataset, was used in place of the training dataset. This greatly reduces the required computation time but was found to still give a reasonable indication of good hyper parameters to use within available compute resources.

SVMs have two hyper-parameters (C and γ) that should be optimised. We set C and γ in exponentially growing sequences, $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$, which has been shown to be a practical method for identifying good parameters (Hsu et al., 2003). In addition the parameters (number of points n_{points} and radius r) of the LBP should also be optimised. For all experiments we set $n_{\text{points}} = 2, 4, \ldots, 30$. For experiments with 28^2 pixel windows we set $r = 1, 2, \ldots, 8$ while for experiments with 64^2 pixel windows we set $r = 2, 4, \ldots, 16$ (r has maximum value equal to a quarter of the image width). For all experiments involving the B⟩K method of feature extraction we optimised K for values $K = 1, \ldots, 28$. For all experiments involving the W method of feature extraction we optimised the minimum distance term $d_{\min}$ in growing sequences of integers $[1], [1, 2], \ldots, [1, \ldots, 15]$. For the sequence $[1, 2]$, for example, we would generate a feature vector by first setting $d_{\min} = 1$, passing an image into the W method and retrieving the number of segments. Then the process would be repeated for $d_{\min} = 2$ and both segment values would be appended, yielding a feature vector of length 2. Thus the length of a particular sequence is the length of the feature vector generated by the W method of feature extraction.

In addition to numerical hyper-parameters, SVMs can further use various kernels and LDA can further use various types of 'solver' and both SVMs and LDA have an option to apply a 'shrinking'/'shrinkage' heuristic. For the LBP⟩SVM method we reduced the number of parameter permutations to consider by assuming some independence between parameters. Thus we first optimised C and γ, (considering all permutations of C, γ) for a fixed $n_{\text{points}} = 15, r = 4$ for 28^2 pixel windows and a fixed $n_{\text{points}} = 15, r = 8$ for 64^2 pixel windows, for each kernel with the shrinking heuristic turned on and off. Then we fixed C, γ and the kernel at their optimal values and optimised n_{points} and r (considering all permutations of n_{points}, r). For all other classification methods we considered all possible parameter permutations.

LDA's r and BRISK's CenSurE approximation variables were also treated as hyper-parameters. Details of the results on the hyper-test set from hyper-training that are used to set hyper-parameters for full training are shown in Appendix A.

2.6. Experiments

After hyper-parameter optimisation, we evaluated the performance of each feature-classifier combination by training from scratch on the full training dataset and testing for the first and only time on the test dataset.

To enable a fair comparison of systems running on the two different window sizes, more windows were present in the 28^2 pixel training and test sets than in the 64^2 pixel sets. This is because each 64 pixel image contains roughly five times as much visual information as each 28 pixel image ($28^2/64^2 = 0.19$); each 64^2 pixel image is effectively five 28^2 pixel windows joined together. Therefore, in the 28^2 pixel case, we used a training set of 200,000 windows and a test set of 20,000 windows, each comprised of half grass and half weed windows; while in the 64^2 pixel case we used 1/5 as many windows: 40,000 training and 4,000 test, also comprised of half grass and half weeds.

The most important practical question for weeding robots is the performance in mixed weeds (Rumex + Urtica) and in mixed weather (sunny + overcast), for the two window sizes. Window size is important because it controls the spatial resolution at which the robot could spray the weeds – in square windows of 106 mm or 56 mm. As the key research question, performance was evaluated for every one of the twelve feature-classifier combinations on both window sizes.

It is sometimes the case in machine learning that improved accuracies can be obtained by fusion results from multiple methods into *meta-classifiers* (also known as ensemble learning). Many combination algorithms are available with different and subtle assumptions which are still sometimes debated (Dietterich, 2000; Domingos, 2000).[5] To give a simple illustrative, though non-optimal, idea of what performance improvements could be available, three simple, standard fusion methods were tested. First, a simple voting scheme, META-VOTE, assigns an equal weight to each classifier's output, and yields the classification with the most votes. (In the case of a tie, the best classifier's output is given the deciding vote.) Second, META-ACC weights the votes of each classifier by its accuracy. Finally, META-LDA considers the output of each classifier as

[5] Until recently it was often assumed that optimal combination could be achieved via Bayesian Model Averaging (BMA), which makes class predictions c of input x from models M_i and training data D as $P(c, x) = \sum_i P(c|M_i, x)P(M|D)$. However it is now known that for large data sets, BMA simply converges to the outputs of the single best classifier in the ensemble, ignoring this others (Domingos, 2000), hence it is not used here. This problem with BMA is caused by its underlying assumption that the ensemble contains the perfect, ground-truth model rather than just a set of approximations.

Table 2
Results of applying all 12 classification methods to 28^2 pixel window test dataset containing grass vs mixed weeds, mixed weather, after training on 28^2 pixel window training dataset. *ACC* is over all accuracy, i.e. the probability that a random image is correctly classified. *CI* is the confidence interval in the estimate of *ACC*. *GRASS* and *WEED* are probabilities that images of grass, or weed, respectively, are correctly classified. 'NA' stands for 'Not Applicable', 'shk' for 'shrinking'.

Method	ACC	CI	GRASS	WEED	Parameters
LBP⟩SVM	68.75	3.27×10^{-3}	74.1	63.6	$n_{\text{points}} = 20, r = 5$, kernel = RBF(shk), $C = 2^{13}, \gamma = 2^{-1}$
B⟩K⟩SVM	52.70	3.53×10^{-3}	38.8	67.1	$K = 4$, kernel = linear(shk), $C = 2^{-5}$
FT⟩SVM	**71.80**	3.18×10^{-3}	79.0	64.7	kernel = RBF(shk), $C = 2^1, \gamma = 2^3$
W⟩SVM	64.67	3.38×10^{-3}	72.1	57.1	$d_{\text{min}} = [1, \ldots, 11]$, kernel = RBF(shk), $C = 2^{15}, \gamma = 2^{-3}$
LBP⟩LDA	65.25	3.37×10^{-3}	71.1	59.1	$n_{points} = 28, r = 3$, solver = svd
B⟩K⟩LDA	53.97	3.52×10^{-3}	63.6	44.1	$K = 20$, solver = lsqr(shh)
FT⟩LDA	68.54	3.28×10^{-3}	78.9	63.2	solver = lsqr(shk)
W⟩LDA	61.63	3.44×10^{-3}	69.4	53.8	$d_{\text{min}} = [1, \ldots, 5]$, solver = lsqr(shk)
LBP⟩NN	62.93	3.41×10^{-3}	69.6	66.3	$n_{\text{points}} = 2, r = 4$
B⟩K⟩NN	50.00	3.54×10^{-3}	0.0	100.0	$K = 17$
FT⟩NN	61.79	3.44×10^{-3}	62.3	60.8	NA
W⟩NN	65.77	3.35×10^{-3}	65.6	66.1	$d_{\text{min}} = [1, \ldots, 10]$
META-VOTE	71.38	3.20×10^{-3}	76.5	66.3	NA
META-ACC	71.90	3.18×10^{-3}	76.5	67.3	NA
META-LDA	**73.47**	4.41×10^{-3}	75.8	71.1	NA

an element of a Boolean feature, and trains a new LDA classifier to predict ground truth class from these vectors.[6]

Secondary questions of interest include the effects of perspective, weather type, weed type, and windowing. As a full training process can take several days, these questions were examined using only the best feature + classifier system and assumed to be independent of one another.

To examine the effect of perspective unwarping, the test set (containing grass under mixed weather conditions, and mixed weeds under mixed weather conditions), was split into new test sets according to their windows' vertical locations (row numbers) in the camera images for individual scoring. Low row numbers indicate windows from the base of the image, corresponding to space close to the robot cameras, while high row numbers indicate windows at the top of the image, from space furthest from the robot cameras.

To examine the effect of weather, each epoch was classified as sunny or overcast, and the original test set was split into two test sets comprised of windows of these weather types for individual evaluation.

To examine the effect of weed type on classifier performance, two set sets were created which contained only grass-and-Urtica and grass-and-Rumex respectively, for individual evaluation.

To give an idea of performance in the limiting case of large windows full of weeds or grass, we assessed the performance of the B⟩K⟩SVM classification method on full sized (600×700) pixel windows from data sets containing grass vs mixed and individual weed types under mixed weather conditions. We conducted this experiment because BRISK features are more usually extracted from full-view images than from the standardised windows used in the rest of this study. Thus we wished to asses the performance of this particular classification method under its own ideal conditions. As with other experiments, hyper-parameters were optimised on hyper-training and hyper-test datasets (though of new 600×700 windows and set sizes hyper-training = 1000, hyper-test = 200, training = 10,000,test = 2000, before training and testing on the training and test datasets.

[6] To estimate META-LDA performance on new data without corruption by training the meta-classifier on test data, the test set was split into two random partitions with one used to train the new LDA and the other to test it. This means the result quoted from only a subset of the original test set. However the original and partitioned test sets are sufficiently large to maintain tight Bayesian confidence internals in the accuracy posteriors to be comparable with the other results.

3. Results

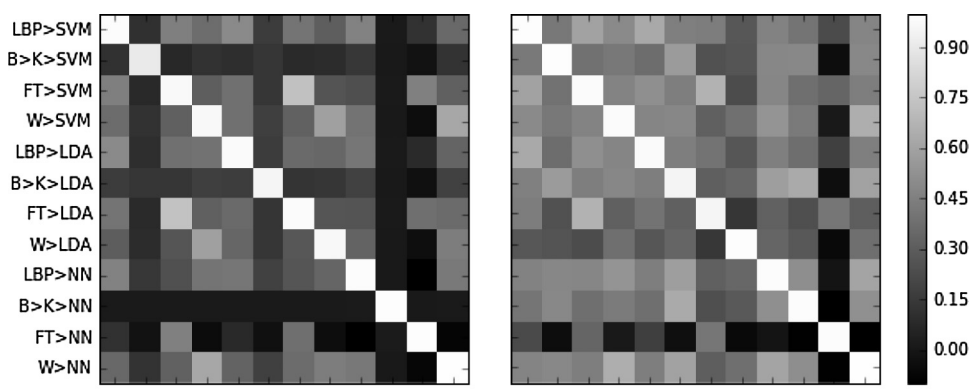
Tables 2 and 3 give the results for training and testing the 12 classification methods on 28^2 and 64^2 pixel windows, respectively, both with mixed weather and mixed weed types. *ACC* shows the overall accuracy of each method, as the proportion of images correctly classified. Bayesian confidence intervals (CI) computed as standard deviations of the Beta distribution posteriors over belief in the accuracy (Bernardo and Smith, 2001), assuming flat priors, are also given for each classification method, which justify the significance of the accuracy percentages to two decimal places. (We also list breakdown accuracies for grass and weed image presentations, which indicate rates of false positive and false negatives.) These experiments were conducted to determine which classifier was the most accurate in predicting test images from data sets containing mixed weeds and mixed weather conditions.

The LBP⟩SVM method performed better than the other classification methods for experiments using 64^2 pixel windows, with an accuracy of 82.88%. This was achieved with the SVM kernel set to RBF with the shrinking heuristic turned on, the hyper-parameters of the LBP set to $n_{\text{points}} = 24, r = 4$ and the hyper-parameters of the SVM set to $C = 2^9, \gamma = 2^3$. The FT⟩SVM method performed better than the other classification methods for experiments using 28^2 pixel windows, with an accuracy of 71.80%. This was achieved with the SVM kernel set to RBF with the shrinking heuristic turned on, and the SVM hyper-parameters set to $C = 2^1, \gamma = 2^3$.

The META-VOTE meta-classifier yielded worse results than the single best method in both 28^2 and 64^2 pixel windowed cases. This may be due to averaging of the best method with the less good methods dragging down the overall result. This typically occurs when all or most classifiers are acting on the same inherent information in the data but with different accuracies, rather than acting on different types of information per method. Similarly, META-ACC gives on a tiny improvement over the best method for 28^2 windows (71.90 vs 71.80), and is worse than the best 64^2 pixel method (80.90 vs 82.88). META-LDA is the best of the meta-classifiers, and is the only one to give significant improvements in both the 28^2 pixel (73.74 vs 71.80) and 64^2 pixel (83.40 vs 82.88). (Significance can be seen by comparing the small CIs with the larger accuracy differences). META-LDA's weights are more principled, and optimal under its assumptions, than the heuristic META-VOTE and META-

Table 3
Results of applying all 12 classification methods to 64^2 pixel window test dataset containing grass vs mixed weeds, mixed weather, after training on 64^2 pixel window training dataset. *ACC* is over all accuracy, ie. the probability that a random image is correctly classified. *CI* is the confidence interval in the estimate of *ACC*. *GRASS* and *WEED* are probabilities that images of grass, or weed, respectively, are correctly classified. 'NA' stands for 'Not Applicable', 'shk' for 'shrinking'.

Method	ACC	CI	GRASS	WEED	Parameters
LBP⟩SVM	**82.88**	5.96×10^{-3}	87.1	78.7	$n_{\text{points}} = 24, r = 4$, kernel = RBF(shk), $C = 2^9, \gamma = 2^3$
B⟩K⟩SVM	69.15	7.27×10^{-3}	70.4	69.0	$K = 17$, kernel = RBF, $C = 2^1, \gamma = 2^8$
FT⟩SVM	79.40	6.39×10^{-3}	84.6	74.2	kernel = RBF(shk), $C = 2^3, \gamma = 2^1$
W⟩SVM	73.23	7.00×10^{-3}	79.0	67.5	$d_{\text{min}} = [1, \ldots, 10]$, kernel = RBF(shk), $C = 2^{11}, \gamma = 2^3$
LBP⟩LDA	75.50	6.80×10^{-3}	82.3	68.7	$n_{points} = 16, r = 4$, solver = lsqr(shk)
B⟩K⟩LDA	70.65	7.16×10^{-3}	81.6	60.9	$K = 17$, solver = lsqr(shk)
FT⟩LDA	73.15	7.01×10^{-3}	82.6	63.7	solver = eigen(shk)
W⟩LDA	63.13	7.63×10^{-3}	88.1	38.2	$d_{\text{min}} = [1, \ldots, 15]$, solver = svd
LBP⟩NN	72.43	7.06×10^{-3}	77.0	68.0	$n_{\text{points}} = 10, r = 12$
B⟩K⟩NN	70.55	7.40×10^{-3}	71.5	63.7	$K = 18$
FT⟩NN	63.08	7.63×10^{-3}	58.7	67.5	NA
W⟩NN	74.48	6.89×10^{-3}	76.0	73.1	$d_{\text{min}} = [1, \ldots, 5]$
META-VOTE	78.63	6.48×10^{-3}	89.4	67.9	NA
META-ACC	80.90	6.21×10^{-3}	88.4	73.8	NA
META-LDA	**83.40**	8.42×10^{-3}	i84.9	81.9	NA



(a) (b)

Fig. 6. Covariance matrices. (a) Covariance between the 12 classification methods when applied to 28 pixel squared windows (grass vs mixed weeds, mixed weather). (b) Covariance between the 12 classification methods when applied to 64 pixel squared windows (grass vs mixed weeds, mixed weather).

ACC, so its better performance is expected. However the gain from using META-LDA over using just the single best method, in each window case, is small. Again, this suggested that all the methods are operating on similar information within the images rather than with different information. Further insight into this possibility is gained by examining the correlation matrix of the 12 methods' predictions in Fig. 6. Here, each grass/weed classification in the test set is considered to have a value of 0 or 1 for grass/weed, and correlations over the test data are presented. It can be seen that the methods are less correlated with one another in the 28^2 pixel case than in the 64^2 pixel case, which explains why meta-classification works better for 28^2 than 64^2 windows. There are stronger correlations between methods sharing the same feature type than methods sharing the same classifier type, as can be seen by the secondary diagonal patterns. The FT>NN method has a low correlation with the others because it is a very poor accuracy method.

Results for the distance experiment are shown in Fig. 7a. The best performing feature-classifier combination – the LBP⟩SVM method – is here run again on mixed weed and weather test sets, separated as a function of the distance of the window from the robot camera's ground location (for both 28^2 and 64^2 pixel-squared windows). Classification performance decreased smoothly as the distance increased, for both 28^2 and 64^2 pixel windows, by a considerable amount (by around 15% absolute for 64^2 windows, and 10% absolute for 28^2 windows.) Weeds closest to the cameras were predicted with a 87.85% accuracy (for 64^2 pixel windows), which is more in line with the high accuracies reported by the previous studies than with accuracies at far distances. It should also be noted that this result was obtained for a mixture of Rumex and Urtica under a mixture of weather conditions, unlike those studies.

Results of the weed type and weather experiments are shown in 7b, again using the LBP⟩SVM method. Here W stands for mixed weeds, R stands for Rumex and U stands for Urtica; M stands for mixed weather, O stands for overcast weather and S stands for sunny weather. Rumex classification was more accurate than Urtica or mixed weed classification, and mixed weed classification was better than Urtica classification. For 28^2 pixel windows classification under sunny weather conditions was better than classification under overcast weather conditions. For 64^2 pixel windows the opposite weather pattern was found.

Finally, Table 4 gives the results of applying the B⟩K⟩SVM classification method on full sized (600×700) windows for data sets containing grass vs mixed and individual weed types under mixed weather conditions. Again Rumex classification was the most accu-

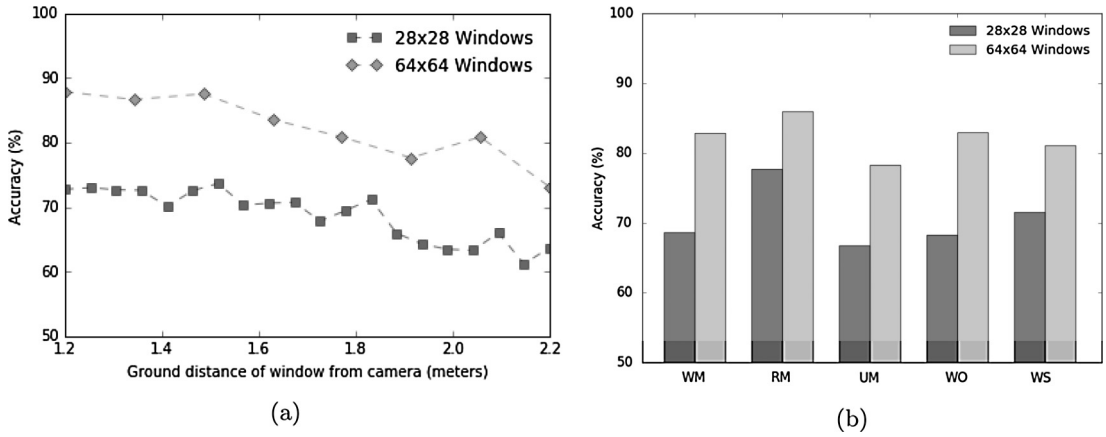


Fig. 7. Effects of distance, weed type and weather. (a) Classification performance of LBP⟩SVM as a function of the distance from the robot cameras (experiments on grass vs mixed weeds, mixed weather). (b) Classification performance of LBP⟩SVM as a function of weed type (W: mixed weeds, R: Rumex, U: Urtica) and weather type (M: Mixed weather, O: Overcast, S: Sunny).

Table 4
Results of applying the B⟩K⟩SVM method to full sized (600 × 700) windows (mixed and individual weed types, mixed weather).

Experiment	Accuracy	Optimum K	Optimum C	Optimum γ
WM	95.1	16	2^{-3}	2^{3}
RM	97.9	19	2^{11}	2^{1}
UM	94.65	28	2^{1}	2^{3}

rate (97.9%), while Urtica classification was the least accurate (94.65%).

4. Conclusion

For our data set and the requirements upon which it is based, the best performing method for the overall spray/no-spray decision is Linear Binary Patterns with Support Vector Machine classification on 64^2 pixel windows.

LBPs are texture-based, rather than shape-based, features, and SVM is a highly nonlinear model. This suggests that when a mixture of Rumex and Urtica is present and spray/no-spray decisions are required, texture is more informative than shape, and that the discriminating distribution of texture features has some nonlinear component. In particular, linear classification of the same features with LDA performs less well.

All the accuracies in our independent re-implementations are lower than those reported in the papers which originally proposed them. This may be due to several factors. First, our data is more difficult to classify, even by human eye, than data used in the original studies. Apart from Ahmed et al. (2014), previous work has used vertical, downward-pointing cameras giving clear and equal views of each point on the ground by removing the need for perspective correction. Our data is more challenging, requiring additional invariance to perspective distance due to the requirement to operate with cameras mounted on top of robot bodies rather than protruding from them. Second, we required our data to come from a moving vehicle without expensive image stabilisation, so intentionally included some blurred images which confuse edge-based detection methods in particular, as these edges become blurred and no longer trigger these detectors. Third, our data is required to come from a wide mixture of lighting and weather conditions as would be encountered in real-world applications. Fourth, we did not allow fitting of any parameters to the test set, and allowed each algorithm to see the test data only once, and report only these results. Fifth, we have removed all possible experimenter, data set selection, and publication bias by operating as an independent controlled study rather than setting out to show the benefits of any one method.

For some applications, such as treatment by individual species-selective herbicides, finer classification of weed type into Rumex and Urtica may be required. Correct classification of Urtica is harder to achieve than of Rumex, using the overall best LBP-SVM method. This is likely because Rumex has larger, flatter leaves which present more obvious differences to most features than Urtica's smaller and more contoured leaves. The BRISK-KMEANS-SVM method shows less difference in performance between Urtica and Rumex when run on very large windows, as expected this may be due to its ability to pick up the jagged edges of Urtica leaves. However it does not work well for the regular 64^2 and 28^2 pixel windows, because it depends on the ability to select good interest points from a large image. With 64^2 pixels the choice of interest points is very limited and with 28^2 is almost non-existent, resulting in few or no interest points being found to classify. LBP and BRISK are closely related, with LBP viewable as a special case of BRISK that treats *every* pixel as an interest point and forces it to be included in the classification, which explains why LBP outperforms BRISK for the smaller windows. The fact that Rumex is in general easier to classify than Urtica may explain the existence of the many more published method-proposing studies of Rumex vision than Urtica vision.

Evidence for the contribution of perspective effects to reducing accuracy is given by the distance experiment result, which shows a considerable drop in accuracy as a function of distance from the camera. If all plants were part of a perfectly flat ground surface then the affine transformation would yield identical images to those taken by a vertical overhead camera as used in previous studies. However real plants and ground are not flat and in particular the vertical structure of plants near the camera results in them being enlarged out of proportion by the perspective transform. The distortion is tolerable for short distances but makes the system less useful beyond distances of around 1.5 m. This suggests that for robots that are not able to mount vertical overhead cameras, for example rough terrain specialist robots for which it is undesirable to have overhanging parts that could be damaged by collisions, it may be preferable to concentrate visual processing power only on nearby regions of ground space. Computation is a limited resource for most mobile robots, which must trade off frame rate for size of spatial area to process and battery power consumption. Designers of these robots should consider increasing frame rates to

obtain multiple views of the same nearby terrain up to around 1.5 m away, at the expense of ignoring further away terrain. It is possible that some improvements to distant recognition will be possible using higher resolution cameras, to produce less pixel distortion during dewarping; by using cameras with smaller apertures to gain deeper depth of field; and/or by mounting cameras at higher positions such as one poles above the robot.

The effect of weather conditions on classification appears somewhat ambiguous from the tests conducted here. The classifiers were trained on mixed sunny and overcast data, then tested on mixed, sunny-only and overcast-only data. Overcast weather yields mostly diffuse lighting from the whole sky, while sunny weather comprises mostly directional light from the sun's position in the sky, which gives rise to distinct shadows. In some cases the shapes of shadows may assist classification (eg. the shapes of Urtica leaf shadows include the same distinctive jagged edges as the leaves themselves), while in other cases shadows may act as noise over the features of the real leaves. There is no clear contributing weather factor to these results, unlike the weed type breakdown which gave clear evidence that Urtica are more responsible than Rumex for lowering performance. Future work could try training classifiers on sunny-only and on overcast-only data, or on more nuanced partitions of weather type, and test them on matched conditions. However unlike the present experiment, this would require online robots to first classify the overall weather condition in order to choose which classifier to use, which introduces further complexity.

More fine grained weather and time-of-day classifications could be made. These can affect both the spectrum of light illuminating the plants, and how the light interacts with the plants, for example casting shadows, or cases of viewing low sunlight through leaves. The current dataset contains time-of-day information which could be used to break down performance in this way; epochs might also be further sub-classified into more detailed weather conditions, or more epochs obtained from new weather and time-of-year conditions. The data sets are all collected from the same weeds plots, and while we have argued that these do produce substantial variation in the images, it would be useful to validate the methods on completely separate plots in the future.

Window size is obviously an important factor on accuracy, because larger windows contain more information than smaller ones. However there is a trade off because they represent larger spatial regions which reduce the available accuracy of precision spraying. Robot designers can choose between 83% accuracy at 106 mm resolution, or 72% accuracy at 56 mm resolution, from the present studies. Very large windows can give near-perfect results as in the large-window BRISK experiments yielding 95% accuracy for mixed weeds and 97.9% accuracy for Rumex. In practice of course, classifications of single windows are unlikely to take place completely independently of one another. Rather, in the field, a live robot would perform both spatial averaging of neighbouring window classes, as well as temporal averaging as multiple images of the same regions are taken over time from moving robot locations, for example via Markov Random Fields as in Hiremath et al. (2013), which would improve accuracy. The windows used here are non-overlapping and live robots could further make use of overlapping windows to provide additional information about the spatial frequencies across window boundaries that is not used in the present study. An indication of the strength of accuracy amplification by multi observation fusion (which could be temporal or spatial or both) is shown in Fig. 8. This shows that the present LBP-SVM accuracy is easily amplified into the mid or high 90 s percentages by 2–5 observations, as could be obtained by a moving robot. This is computed by Bayes-fusing accuracies with themselves repeatedly, where the Bayesian fusion of two evidence probabilities p and q is given by,

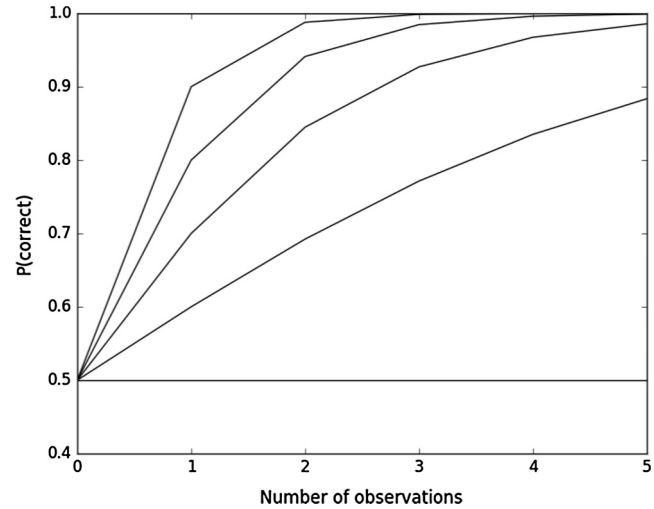


Fig. 8. Multi-observation fusion. Showing the effect of Bayes-fusing multiple observations of a window. A live moving robot is likely to make several observations and classifications of each window from different poses, which when fused together will increase accuracy. Each line represents fusion of n observations of equal accuracy, whose per-observation accuracy is seen in the 'number of observations = 1' case ranging from 0.5 to 0.9 in steps of 0.1.

$$\frac{pq}{pq + (1-p)(1-q)} \tag{18}$$

It can be seen from Fig. 8 that fused data from two or three windows, observed over time and/or space, is sufficient to bring mot methods to 95%+ accuracy which is generally sufficient (e.g. van Evert et al., 2011) for spraying use in the field.

Running multiple classifiers and combining their results with an LDA meta-classifier yields a slightly higher accuracy than pure LBP-SVM. The effect is only slight because all the feature-classifier methods mostly work with similar image information to each other, with varying levels of success, rather than working with different types of information. This information is presumably Fourier or wavelet-like, and mostly linear, though with some nonlinearities which enable the nonlinear SVM to outperform the other methods. All of the classification methods tested here run comfortably in real time for live use, however running them all simultaneously for meta-classification on a mobile robot platform would likely require parallel processors which consume additional and valuable battery power.

As part of this publication we are making our training and test sets available for non-commercial research by others under Creative Commons licence CC BY-NC 3.0 US. We hope that the standardised setup presented here will enable performance to be improved upon through fair evaluation of new methods and implementations, and though other researchers adding data from new conditions to the set.

Appendix A

Here we give the intermediate results used for the parameter optimisation stage of classification for all 12 methods when trained on hyper-training and tested on hyper-test datasets containing grass vs mixed weeds under mixed weather conditions. This gives an indication of the effect of hyper-parameter optimisation on performance (see Tables 5–19).

Tables 5 and 12 give the results for all classification methods except those involving the NN classifier (results for the NN classifier are given in the text) used on 28^2 pixel windows. 'nan' indicates an experiment abandoned due to unreasonable

Table 5
Results for the classification method LBP⟩SVM with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	Optimum C	Optimum γ
linear	off	65.6	2^{11}	NA
linear	on	65.6	2^{11}	NA
rbf	off	68.3	2^{13}	2^{-1}
rbf	on	68.4	2^{13}	2^{-1}

Table 6
Results for the classification method B⟩K⟩SVM with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Kernel	Shrinking	Accuracy	Optimum K	Optimum C	Optimum γ
linear	off	53.1	4	2^{-5}	NA
linear	on	53.1	4	2^{-5}	NA
rbf	off	53.1	4	2^{-5}	2^{-15}
rbf	on	52.5	4	2^{-5}	2^{-15}

Table 7
Results for the classification method FT⟩SVM with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	Optimum C	Optimum γ
linear	off	67.1	2^3	NA
linear	on	67.1	2^3	NA
rbf	off	69.3	2^1	2^3
rbf	on	69.4	2^1	2^3

Table 8
Results for the classification method W⟩SVM with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	$d_{\min}$	Optimum C	Optimum γ
linear	off	65.7	$[1,\ldots,9]$	2^3	NA
linear	on	65.7	$[1,\ldots,9]$	2^3	NA
rbf	off	66.6	$[1,\ldots,11]$	2^{15}	2^{-3}
rbf	on	66.7	$[1,\ldots,11]$	2^{15}	2^{-3}

Table 9
Results for the classification method LBP⟩LDA with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Solver	Shrinkage	Accuracy (%)	Optimum n	Optimum r
svd	off	67.6	28	3
lsqr	off	67.6	28	3
lsqr	on	67.6	28	4
eigen	on	67.6	28	3

Table 10
Results for the classification method B⟩K⟩LDA with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Solver	Shrinkage	Accuracy	Optimum K
svd	off	54.3	4
lsqr	off	54.1	23
lsqr	on	54.9	20
eigen	on	54.9	23

Table 11
Results for the classification method FT⟩LDA with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Solver	Shrinkage	Accuracy (%)
svd	off	65.5
lsqr	off	65.5
lsqr	on	65.6
eigen	on	65.4

Table 12
Results for the classification method W⟩LDA with hyper-training and hyper-test datasets, on 28^2 pixel windows.

Solver	Shrinkage	Accuracy	$d_{\min}$
svd	off	63.1	$[1,\ldots,5]$
lsqr	off	56.5	$[1]$
lsqr	on	63.3	$[1,\ldots,5]$
eigen	on	nan	nan

Table 13
Results for the classification method LBP⟩SVM with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	Optimum C	Optimum γ
linear	off	80	2^5	NA
linear	on	80	2^5	NA
rbf	off	82	2^9	2^3
rbf	on	82	2^9	2^3

Table 14
Results for the classification method B⟩K⟩SVM with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Kernel	Shrinking	Accuracy	Optimum K	Optimum C	Optimum γ
linear	off	59	17	2^3	NA
linear	on	61.25	17	2^5	NA
rbf	off	68.5	17	2^1	2^8
rbf	on	63	17	2^3	2^3

Table 15
Results for the classification method FT⟩SVM with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	Optimum C	Optimum γ
linear	off	76.5	2^5	NA
linear	on	76.5	2^5	NA
rbf	off	79	2^3	2^1
rbf	on	79	2^3	2^1

Table 16
Results for the classification method W⟩SVM with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Kernel	Shrinking	Accuracy (%)	$d_{\min}$	Optimum C	Optimum γ
linear	off	72.5	$[1,\ldots,10]$	2^{13}	NA
linear	on	72.5	$[1,\ldots,10]$	2^{13}	NA
rbf	off	76.5	$[1,\ldots,10]$	2^{11}	2^3
rbf	on	76.5	$[1,\ldots,10]$	2^{11}	2^3

Table 17
Results for the classification method LBP⟩LDA with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Solver	Shrinkage	Accuracy (%)	Optimum n	Optimum r
svd	off	81.5	28	4
lsqr	off	81.5	28	4
lsqr	on	82	16	4
eigen	on	81.5	16	4

Table 18
Results for the classification method B⟩K⟩LDA with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Solver	Shrinkage	Accuracy	Optimum K
svd	off	76.5	20
lsqr	off	77.5	17
lsqr	on	78	17
eigen	on	77.5	22

Table 19
Results for the classification method FT⟩LDA with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Solver	Shrinkage	Accuracy (%)
svd	off	55.5
lsqr	off	53
lsqr	on	61
eigen	on	62

Table 20
Results for the classification method W⟩LDA with hyper-training and hyper-test datasets, on 64^2 pixel windows.

Solver	Shrinkage	Accuracy	$d_{\min}$
svd	off	67	$[1,\ldots,15]$
lsqr	off	57	$[1,\ldots,13]$
lsqr	on	66.5	$[1,\ldots,13]$
eigen	on	nan	nan

computation time, while NA stands for 'not applicable'. Tables 13 and 20 give the equivalent results for 64^2 pixel windows. For LBP⟩SVM, Tables 5 and 13 give the results for the first stage of hyper-parameter optimisation, in which the hyper-parameters of the SVM C and γ were optimised. The second stage of optimisation for this method yielded optimised values for the hyper-parameters of the LBP ($n_{\text{points}} = 20, r = 5$,accuracy = 72.50% for 28^2 pixel windows, $n_{\text{points}} = 24$, $r = 4$,accuracy = 88.00% for 64^2 pixel windows). For LBP⟩NN, optimised parameters for the LBP were $n_{\text{points}} = 2, r = 4$ for 28^2 pixel windows and $n_{\text{points}} = 10, r = 12$ for 64^2 pixel windows. For B⟩K⟩NN, the optimal value of K for the B⟩K method of feature extraction was 17 for 28^2 pixel windows and 18 for 64^2 pixel windows. For FT⟩NN there were no parameters to optimise. For W⟩NN, the optimal value of $d_{\min}$ for the W method of feature extraction was $[1,\ldots,10]$ for 28^2 pixel windows and $[1,\ldots,5]$ for 64^2 pixel windows.

References

Agrawal, Motilal, Konolige, Kurt, Blas, Morten Rufus, 2008. Censure: center surround extremas for realtime feature detection and matching. In: European Conference on Computer Vision. Springer, pp. 102–115.

Ahmed, Faisal, Kabir, Md Hasanul, Bhuyan, Shayla, Bari, Hossain, Hossain, Emam, 2014. Automated weed classification with local pattern-based texture descriptors. Int. Arab J. Inf. Technol. 11 (1), 87–94.

Anken, T., Holpp, M., Venn, W., Kutterer, H.J., et al., 2010. Automatic detection of broad-leaved dock in grassland. In: International Conference on Agricultural Engineering-AgEng 2010: Towards Environmental Technologies, Clermont-Ferrand, France, 6–8 September 2010. CEMAGREF.

Bechar, Avital, Vigneault, Clément, 2016. Agricultural robots for field operations: concepts and components. Biosyst. Eng. 149, 94–111.

Bernardo, José M., Smith, Adrian F.M., 2001. Bayesian theory.

Blackmore, Simon, Stout, Bill, Wang, Maohua, Runov, Boris, 2005. Robotic agriculture – the future of agricultural mechanisation. In: Proceedings of the 5th European Conference on Precision Agriculture, pp. 621–628.

Cope, James S., Corney, David, Clark, Jonathan Y., Remagnino, Paolo, Wilkin, Paul, 2012. Plant species identification using digital morphometrics: a review. Expert Syst. Appl. 39 (8), 7562–7573.

Dietterich, Thomas G., 2000. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems. Springer, pp. 1–15.

Domingos, Pedro, 2000. Bayesian averaging of classifiers and the overfitting problem. ICML, vol. 2000, pp. 223–230.

Dürr, L., Anken, T., Bollhalder, H., Sauter, J., Burri, K.G., Kuhn, D., 2005. Machine vision detection and microwave based elimination of *Rumex obtusifolius* L. on grassland. In: 5th European Conference on Precision Agriculture, Uppsala, Sweden, p. 5.

Friedman, Jerome, Hastie, Trevor, Tibshirani, Robert, 2001. The Elements of Statistical Learning, Springer Series in Statistics, vol. 1, Springer, Berlin.

Gebhardt, Steffen, Schellberg, Jürgen, Lock, Reiner, Kühbauch, Walter, 2006. Identification of broad-leaved dock (*Rumex obtusifolius* L.) on grassland by means of digital image processing. Prec. Agric. 7 (3), 165–178.

Guo, Lei, Li, Qingshan, 2012. LBP and its variations for image classification. In: Proceedings of the 2012 International Conference on Electronics, Communications and Control. IEEE Computer Society, pp. 712–715.

Hamuda, Esmael, Glavin, Martin, Jones, Edward, 2016. A survey of image processing techniques for plant extraction and segmentation in the field. Comput. Electron. Agric. 125, 184–199.

He, Dong-Chen, Wang, Li, 1990. Texture unit, texture spectrum, and texture analysis. IEEE Trans. Geosci. Rem. Sens. 28 (4), 509–512.

Hinton, Geoffrey E., Osindero, Simon, Teh, Yee-Whye, 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.

Hiremath, S., Van der Heijden, G., Van Evert, F.K., Stein, A., 2012. The role of textures to improve the detection accuracy of *Rumex obtusifolius* in robotic systems. Weed Res. 52 (5), 430–440.

Hiremath, Santosh, Tolpekin, Valentyn A, Heijden, Gerie van der, Stein, Alfred, 2013. Segmentation of *Rumex obtusifolius* using gaussian markov random fields. Mach. Vis. Appl. 24 (4), 845–854.

Hsu, Chih-Wei, Chang, Chih-Chung, Lin, Chih-Jen, et al. 2003. A Practical Guide to Support Vector Classification. Tech. Report. <https://www.csie.ntu.edu.tw/cjlin/>.

Leutenegger, Stefan, Chli, Margarita, Siegwart, Roland Y., 2011. Brisk: binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision. IEEE, pp. 2548–2555.

McCarthy, Cheryl, Rees, Steven, Baillie, Craig, 2010. Machine vision-based weed spot spraying: a review and where next for sugarcane? Proceedings of the 32nd Annual Conference of the Australian Society of Sugar Cane Technologists (ASSCT 2010), vol. 32. Australian Society of Sugar Cane Technologists, pp. 424–432.

McLachlan, Geoffrey, 2004. Discriminant Analysis and Statistical Pattern Recognition, vol. 544. John Wiley & Sons.

Moshou, D., Kateris, D., Pantazi, X.E., Gravalos, I., 2013. Crop and weed species recognition based on hyperspectral sensing and active learning. In: Precision Agriculture. Springer, pp. 555–561.

Pantazi, Xanthoula-Eirini, Moshou, Dimitrios, Bravo, Cedric, 2016. Active learning system for weed species recognition based on hyperspectral sensing. Biosyst. Eng. 146, 193–202.

Patel, Akash, Kasat, D.R., Jain, Sanjeev, Thakare, V.M., 2014. Performance analysis of various feature detector and descriptor for real-time video based face tracking. Int. J. Comput. Appl. 93 (1).

Peteinatos, Gerassimos G., Weis, Martin, Andújar, Dionisio, Ayala, Victor Rueda, Gerhards, Roland, 2014. Potential use of ground-based sensor technologies for weed detection. Pest Manage. Sci. 70 (2), 190–199.

Polder, Gerrit, van Evert, Frits K., Lamaker, Arjan, De Jong, Arjan, Van der Heijden, G. W.A.M., Lotz, L.A.P., Van der Zalm, T., Kampenaar, C., 2007. Weed Detection using Textural Image Analysis. Plant Research International, PO Box, 16:6700.

Rosenthal, Robert, 1979. The file drawer problem and tolerance for null results. Psychol. Bull. 86 (3), 638.

ŠEATOVIĆ, Dejan, 2008. 3d-object recognition, localization and treatment of Rumex obtusifolius in its natural environment. In: 1 st International Conference on Machine Control & Guidance, p. 169.

Šeatović, Dejan, Kutterer, Hansjörg, Anken, Thomas, 2010. Automatic weed detection and treatment in grasslands. In: ELMAR, 2010 Proceedings. IEEE, pp. 65–68.

Siddiqi, Muhammad Hameed, Ahmad, Irshad, Sulaiman, Suziah Bt, 2009. Weed recognition based on erosion and dilation segmentation algorithm. In: 2009 International Conference on Education Technology and Computer. IEEE, pp. 224–228.

Tillett, N.D., Marchant, John A., Hague, A., 1996. Autonomous plant scale crop protection. AgEng96, Eur. Soc. Agric. Eng. A 96, 23–26.

Van Evert, F.K., Polder, G., Van Der Heijden, G.W.A.M., Kempenaar, C., Lotz, L.A.P., 2009. Real-time vision-based detection of *Rumex obtusifolius* in grassland. Weed Res. 49 (2), 164–174.

van Evert, Frits K., Samsom, Joost, Polder, Gerrit, Vijn, Marcel, Dooren, Hendrik-Jan van, Lamaker, Arjan, van der Heijden, Gerie W.A.M., Kempenaar, Corné, Zalm, Ton van der, Lotz, Lambertus A.P., 2011. A robot to detect and control broad-leaved dock (*Rumex obtusifolius* L.) in grassland. J. Field Robot. 28 (2), 264–277.

Vapnik, Vladimir, 2013. The Nature of Statistical Learning Theory. Springer Science & Business Media.

Vibhute, Anup, Bodhe, S.K., 2012. Applications of image processing in agriculture: a survey. Int. J. Comput. Appl. 52 (2).

Vincent, Luc, Soille, Pierre, 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 13 (6), 583–598.

Vrindts, Els, Baerdemaeker, Josse De, 1999. Optical weed detection and evaluation using reflection measurements. In: Photonics East (ISAM, VVDC, IEMB). International Society for Optics and Photonics, pp. 279–289.

Weisstein, Eric W., 2015. Fast Fourier Transform. Wolfram World of Mathematics (Online Resource).

When should the chicken cross the road?: Game theory for autonomous vehicle - human interactions

C.W. Fox[a,b,c], F. Camara[c,b], G. Markkula[c], R.A. Romano[c], R. Madigan[c] and N. Merat[c]

February 6, 2018

Abstract

Autonomous vehicle control is well understood for localization, mapping and planning in un-reactive environments, but the human factors of complex interactions with other road users are not yet developed. This position paper presents an initial model for negotiation between an autonomous vehicle and another vehicle at an unsigned intersections or (equivalently) with a pedestrian at an unsigned road-crossing (jaywalking), using discrete sequential game theory. The model is intended as a basic framework for more realistic and data-driven future extensions. The model shows that when only vehicle position is used to signal intent, the optimal behaviors for both agents must include a non-zero probability of allowing a collision to occur. This suggests extensions to reduce this probability in future, such as other forms of signaling and control. Unlike most Game Theory applications in Economics, active vehicle control requires real-time selection from multiple equilibria with no history, and we present and argue for a novel solution concept, *meta-strategy convergence*, suited to this task. [1]

1 Introduction

Automated vehicle (AV) localization, mapping, and planning have recently become practically feasible due to price falls in computer processing power. The problem of simultaneous localization and mapping (SLAM) is well understood [16], and despite its exact solution being NP-hard [15], good approximations exist such as particle filtering, which make use of large compute power to draw samples near solutions. Route planning in non-interactive environments also has well known tractable solutions such as the A-star algorithm. Given a route, localizing and control to follow that route then becomes a similar task to that performed by the 1959 General Motors Firebird-III self-driving car [1], which used electromagnetic sensing to follow a wire built into the road. Such path following, using wires or SLAM, can then be augmented with simple safety logic to stop the vehicle if any obstacle is in its way, as detected by any range sensor. Free and open source systems for this level of 'self-driving' are now widely available [6].

In contrast, problems that these vehicles will face around interacting with other road users are much harder both to formulate and solve. Autonomous vehicles do not just have to deal with inanimate objects, sensors, and maps. They have to deal with other agents, currently human drivers and pedestrians and eventually other autonomous vehicles, all of which may be at least as "intelligent" and "rational" as they are, and in competition with them for space, time and priority on the road. Recent studies have shown [9] that in trials of autonomous minibuses, pedestrians knowingly obstruct autonomous vehicles around once every three hours – enough to occur once every day on a long commute. Once human road users know that AV safety systems are programmed to stop if any obstacle is in their path, they can quickly take advantage of this to push in front of the AV and take priority. If this becomes common knowledge across a whole city, AVs will make little or no progress because they will be forced to yield at every single interaction.

Understanding and predicting other agents' behavior, especially when that includes understanding and predict-

[1] Author affiliations: *a* Ibex Automation Ltd, UK; , *b* School of Computer Science, University of Lincoln, UK; *c* Institute for Transport Studies, University of Leeds, UK. This project has received funding from the European Union's Horizon 2020 R&I programme under grant agreement InterACT No 723395.

ing their understandings and predictions of oneself, and of one's understanding and predictions of them, *ad infinitum,* is a massively more complex problem than inferring locations and maps. It may even be formally uncomputable as it requires predictions of and actions based on one's future behavior (via the other party's models of it), which are well known in Computer Science to lead to paradox and uncomputability as in Gödel's theorem and the Halting problem [17].

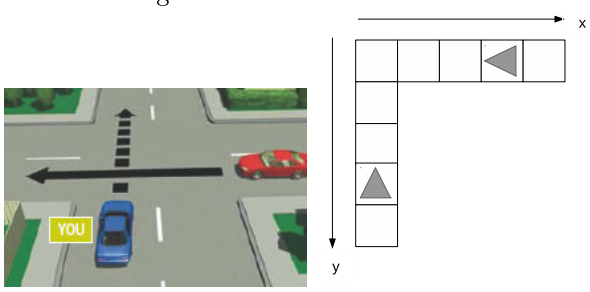
Game Theory provides some framework for managing such self-referential decisions, but appears to be incomplete as a prescriptive theory when multiple equilibria are present. Solutions may be formally computationally intractable (NP-hard) in some cases [13]. And when multiple equilibria are present, it is not clear whether game theory or any other rational line of argument will ever be able to even formulate the problems, let alone solve them. A simple example of game theory is the classic game 'Chicken', in which two cars each drive *straight* towards each other at speed or *swerve* away, and the nominal loser (the 'chicken') is the one to swerve, but both players are much bigger losers if they both do not swerve, and collide. The classic formal Chicken model makes the strong and unrealistic assumption that the straight/swerve decision is made as a single action (a) selection, simultaneously by both players (X and Y) so that the payoff values (v) given the actions can be represented as a 2×2 matrix of pairs $(v^Y_{a_Y,a_X}, v^X_{a_Y,a_X})$:

$Y \mid X$	a_X =*swerve*	a_X =*straight*
a_Y =*swerve*	(0,0)	(-1, +1)
a_Y =*straight*	(+1, -1)	(-100,-100)

The central concept of game theory is *equilibrium* [11] which for a 2×2 matrix game as above describes any pair of strategies for the two players such that if either player knew the other's they would not change their own. Conceptually, if equilibria exist then one can usually be found via 'fictitious play', i.e. each player simulates her and her opponent's behaviour in a virtual world where they know each other's strategies, until they converge. Strategies are probability distributions over actions. If a game has only one equilibrium then it is optimal for both players to play its strategies. When there are multiple equilibria, the question of what to do is less clear. Historically, Game Theory has focused on description of observed behaviors (how *do* people act? Why *did* the chicken cross the road?) and mechanism design (how can government make them act?) rather than prescription (when *should* we act? When *should* the chicken cross the road?) of real-time action selection by agents. This is due to its roots in economics and mathematics rather than robotics. This distinction becomes crucial when multiple equilibria are present in a game. The descriptive school would say that *any* of the equilibria are valid descriptions of what might be observed in human behavior data [4]. The 'mechanism design' school typically argues [12] that the problem of equilibrium selection is 'unpleasant' and should be avoided by changing the rules of the game to produce a single unambiguous equilibrium which all players can usefully use. For cases where the equilibrium selection problem cannot be avoided in these ways, many different solution concepts (aka. 'equilibrium refinements') have been proposed and debated [2] for choosing between them. Two of these are widely acknowledged to be 'rational' when applied in order: *1. Dominance* - Clearly, if an equilibrium is worse for all players than at least one other, then we discard it without question*; 2. Evolutionary stability / symmetry* (ESS) - Assume everyone in the world was to use the same equilibrium as me. Discard equilibria where this would not work. These concepts help to reduce the number of potential multiple equilibria but do not guarantee reduction to a unique one. To discard further equilibria, more controversial concepts have been debated [2] including: *1. Trembling hand stability* - Assume other player has a small epsilon probability of making a mistake; prefer equilibria that we converge back to if that happens; *2. Basin size* (aka. 'risk dominance'). Assume both players use fictitious play starting from maximum entropy strategies. Consider which equilibrium attracts from the most start points, and this thus most likely to occur. *3. Social maximum* (aka. 'payoff dominance') Choose the equilibrium with the largest sum of payoffs to all players (even if I am worse off than in others). *4. Other arbitrary conventions*. Such as using the action with the first letter in the alphabet. These work only if all players agree to use them in advance or can be argued to possess social knowledge to make them confident that others will choose the same ones as them. This appears to be the point where the mathematics of Game Theory ends, and philosophical debate about the meaning of 'rationality' takes its place [2]. However, as autonomous vehicle engineers building real-time control systems, we *must* make some action selection in these situations, somehow.

Chicken is intended as a simple educational example game and not as a model of real vehicles. This study modifies it into a general and more realistic vehicle interaction problem, where an AV competes for priority with another vehicle or a pedestrian stepping out in front of it, and allowing them to negotiate with one another by observing each others' behaviour over time. Initially we

Figure 1: Scenario and model.



consider the simplest possible model of this class of problems, of an AV and another *similar vehicle* approaching an unmarked intersection at speed as in fig. 1(left). This would also apply to the case of two pedestrians meeting each other and negotiating for space. We then extend it to cases where the two players have *asymmetric penalties* in the event of collision – modeling an AV encountering a potential jaywalker stepping into the road, or an AV encountering a physically larger or smaller vehicle such as an armored SUV or cyclist at an intersection. The model proposed here is intended to be the simplest possible which captures the dynamics of interest common to these cases, but which can also serve as a foundation for many more complex ones.

Game theory is used extensively in macroscopic traffic modeling via Wardrop equilibrium in flow networks [3] with focus on route selection in large, economy-like, markets of many road users rather than microscopic pairwise interactions. Where game theory has been applied to pairwise traffic decisions, it has mostly been at the level of simple single-shot games as reviewed in [5]. In a few cases such as lane-changing [10, 7] and merging [8] it has been extended to sequential games as used here, but not for AV-pedestrian interactions as here. The meta-strategy convergence concept used here is novel to our knowledge, as is the use of the sequential model as a foundation for AV-human intersection and jaywalking control.

2 Methods

Turn-taking model. In the simplest possible model we set up two symmetric agents (vehicles or pedestrians) approaching an intersection as a chess-like, discrete space, discrete speed, and discrete time, turn-taking game, as in fig. 1(right). This does not yet use any Game Theory because the players' decisions are not made at the same time. We will use it to introduce notation and as a base to grow more detailed models. We assume a (1 meter) grid world, with two straight roads at right angles meeting at the intersection. Assume one vehicle on each road, labeled[2] Y and X. (We will later consider one player to be a pedestrian or different types of vehicle.) Assume discrete alternating turns one per second, in which one vehicle can choose either a 1m/s or 2m/s speed, i.e. move either one box forward or two boxes forward. (This ensures a finite game, because the vehicles are guaranteed to move closer to the intersection at each step. Formally, y and x are decreasing variants.) 2m/s is a slow real world speed but is chosen to coincide with simple integer movements of 1 or 2 boxes per turn. (The reader may wish to multiply all distances and speeds by 10 if they wish to think in more real-world units.) Write y for Y's distance in meters to the intersection, x for X's distance in meters. Assume a crash occurs if the vehicles are in the same square or if one vehicle is in the intersection square and the other moves *through* it in a single step. (This may be implemented by treating $(y, x) \in \{(0, 0), (1, 1)\}$ as crash states, and all other states with one agent at 0 or 1 as non-crash endgame states, avoiding the need to model negative positions beyond the intersection.) We assign (negative) utility U_{crash} to each player for a crash; otherwise $-U_{time}T$ where T is the number of seconds it takes to reach the intersection from the start of the game, and U_{time} is the (positive) value of saving one second of travel time. Assume that both players have identical utility functions, and know this to be the case. This game can be played, for example, as a board game between two human players. The turn-taking model can be solved by a standard [14] backward induction max-max tree search as in algorithm 1, where the boolean b represents which player's turn is current, t is time elapsed, y and x are the two players' positions, and the results are expected value pairs for the two players, $v_{y,x,t,b} = (v^Y_{y,x,t,b}, v^X_{y,x,t,b})$.

Sequential chicken model. If we replace turn-taking by simultaneous action selection by both players at each discrete 1s turn, the model transforms into a sequence of Game Theoretic matrix games ("sub-games"). This is equivalent to a board game where both players write down their speed choice (1 or 2) in secret then reveal them and make the moves together rather than in turns. The payoffs of any sub-game at state $(y > 1, x > 1, t)$ become recursive functions of the next states, $(y - a_Y, x - a_X, t + 1)$,

[2] We use the convention of writing Y before X, and the orientation of the grid world of fig. 1, to match (row,column) matrix notation.

Algorithm 1

```
world_value((y,x,t,b)) =
  if y==x==0  or y==x==1
  or (y==1 and x==0) or (y==0 and x==1)
  or (y==0 and x==-1) or (y==-1 and x==0):
    (-100,-100)            #crash
  if y<=0 or x<=0:
    (-t-y/2 , -t-x/2)      #someone arrives
  if b:                    #recursion
    mymax(world_value((y-2,x,t+1,False)),
          world_value((y-1,x,t+1,False)))
  if not b:                #recursion
    yourmax(world_value((y, x-2, t+1, True)),
            world_value((y, x-1, t+1, True)))
```

where $a_Y, a_X \in \{1, 2\}$ are the speed selection actions for speeds of 1m/s and 2m/s. As in the turn-taking model, these inductive values are based on the endgame states when one or both vehicles have reached the intersection (considered to occur at square 0 or 1). Consider the value $v_{y,x,t} = (v^Y_{y,x,t}, v^X_{y,x,t})$ of the sub-game when the game is in state (y, x, t). The induction relation for this sub-game's payoff matrix is,

$$v_{y,x,t} = v(\begin{bmatrix} v(y-1, x-1, t+1) & v(y-1, x-2, t+1) \\ v(y-2, x-1, t+1) & v(y-2, x-2, t+1) \end{bmatrix})$$

and is computable via standard matrix Game Theory. Optimal mixed strategies, where they exist, and resulting state probabilities given an initial start state, for this model are shown in fig. 2.

Asymmetric utility model. A final model asks what happens if the two players have different collision utilities. This occurs for example if one player has a heavier/safer car than the other, such as an SUV (Sports Utility Vehicle). Or if one player is a weaker road user such as a cyclist, or a pedestrian negotiating to cross the road in a conflict zone in front of our AV. Will even a small change to these utilities break the symmetry of the sequential chicken model and tip the balance of who yields? If so, this would give a rational justification for the purchase of heavy vehicles such as SUVs: the intent of such purchases is not to actually get into collisions and benefit from reduced damage, but rather to maintain the possibility, however remote, of such a disaster, and exploit the backward induction from it to obtain concessions in more benign possible worlds, namely of the other player yielding. Purchase of an SUV would then be rational, reducing the cost of time delays to the owner. (There is of course then a higher level game when the other road users can buy similar vehicles, beyond the scope of our present model.) For simplicity we retain the assumption that both players have the same time delay utilities, as in the previous model. We assume the original collision-to-time value scaling of $(U_{crash,}, U_{crash})$, and consider ratios where one player is stronger than the other, as (U_{crash}, rU_{crash}), for $r \in [1, 100]$. When utilities become asymmetric it is possible that more equilibria will be present, so we switch to numerical computation of them using the Lemke-Howson algorithm.

Meta-strategy convergence. The asymmetric chicken model may have multiple equilibria which are not fully disambiguated by dominance and ESS solutions concepts. We propose a novel (to our knowledge) solution concept for use in solving this and other models, which we call *meta-strategy convergence*. This is based on everything we currently know about the 'rational' process of equilibrium pruning and selection, including the *absence* of information in some cases, and on a temporal ordering of rational reasoning. After removing dominated and non-ESS equilibria, we know of no good remaining solution concepts under the assumption that the other player is also rational. (Trembling hand, basins etc. make a different assumption about a fallible opponent, but for AVs which will eventually interact with other autonomous vehicles, we want all players to be completely mechanized and rational.) In the absence of any other way to select from the remaining equilibria's strategies, we form a new meta-strategy which chooses one of them from a flat (maximum entropy) prior. By symmetry, there is no way to prefer any over any other, hence their selection probabilities must be equal, given this state of knowledge. Curiously, this is equivalent to a new strategy which averages the action probabilities from each remaining strategy, yet is not itself in that set, because it is an average between them. Hence, it is not a member of any equilibrium and cannot be an optimal strategy itself. However, we have derived it step-by-step over time in a completely rational way. It is our best solution so far at this new point in time. As it is our best rational solution at this time, the other player will also compute that we have reached it. We then consider, as in standard fictitious play, what the other player will do next. They will apply fictitious play to modify their strategy in response. Then we will modify ours, and they will modify theirs again. This will iterate until we converge, unambiguously, onto a specific and uniquely defined one of the original equilibria. This

Figure 2: Strategy selection (1) and backward induction state probability (2-3) equations.

$$P(a_Y = 1|y,x,t) = \frac{v^Y_{y-1,x-2,t+1} - v^Y_{y-2,x-2,t+1}}{v^Y_{y-1,x-2,t+1} + v^Y_{y-2,x-1,t+1} - v^Y_{y-1,x-1,t+1} - v^Y_{y-2,x-2,t+1}}$$

$$P(y_{init}, x_{init}, t_{init}) = 1, P(y_{init}, x \neq x_{init}, t_{init}) = 0, P(y \neq y_{init}, x_{init}, t_{init}) = 0$$

$$P(y,x,t) = \sum_{\Delta y \in 1,2} \sum_{\Delta x \in 1,2} P(a_Y = \Delta y, a_X = \Delta x | y - \Delta y, x - \Delta x, t-1) P(y - \Delta y, x - \Delta x, t-1)$$

contains the rational strategies for both players, and can be reached deterministically by both of them without the need for any pre-established conventions or communications. Like all solution concepts, this is something of a philosophical rather than purely mathematical argument. It is the best argument currently known to us so we consider it to be rational for the AV control tasks.

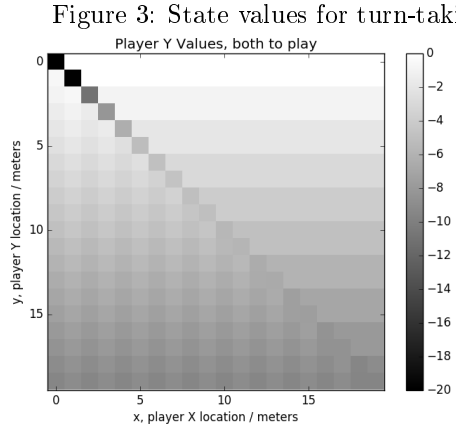
Temporal gauge invariance. The state values of the game theoretic models above are presented as functions of (y,x,t) but our current implementation makes use of an approximation to reduce the number of sub-games to be solved and thus the computation time. Because both player's utilities are linear function of time, we may (up to a small change in the ratio of crash to delay utilities) choose different gauges to measure time, such as considering the time of every turn game to be $t = 0$. When the first player, say Y, reaches the intersection, we assign values $(v^Y_{0,x}, v^X_{0,x}) = (0, -U_{time}x/2)$ as it will take X a further $x/2$ seconds to reach the intersection at maximum speed 2m/s now the road is clear. (Also $(v^Y_{1,x}, v^X_{1,x}) = (0, -U_{time}(x-1)/2)$ to handle the other required end states in the same way.) This simplifies all state values and functions of them to be functions only of (y,x). Removing dependence on t also makes it simpler to visualize results as 2D (y,x) matrices.

3 Results

Assume $U_{crash} = -20$ and $U_{time} = 1$ throughout. (This values a crash as being equally bad as a 100 second delay reaching the intersection. In the real world the crash penalty would be much larger, but smaller ones produce more easily visualizable results for our present purpose.)

Turn-taking model. The value function for Y in the turn-taking model is shown in fig. 3. The game is symmetric so X has the same function when the player's names are swapped. The turn taking model is fully deterministic, because full information is available to each player when it is their turn. Fig. 4 and 5 show simulated runs beginning at asymmetric ($y \neq x$) and symmetric ($y = x$) starting states.When the vehicles start with very different differences, e.g. ($y = 12, x = 8$) they both proceed at full speed (2m/s) and avoid each other. When started at identical distances, such as ($y = 10, x = 10$), the initial turn-taking advantage becomes the tie-breaker, in both parties interests. *Collisions never occur in the turn-taking model* due to its determinism.

Figure 3: State values for turn-taking game.



Sequential chicken model. Fig. *6* shows the value matrix for games with vehicles at up to 20m from the intersection, and fig. 7 show the optimal strategies.

Fig. 8 shows the state space probabilities and fig. 9a stochastic sample, starting with large time (2 seconds) gap between the vehicles. All probabilities in the state

Figure 4: Simulation of turn taking game, differing starts.

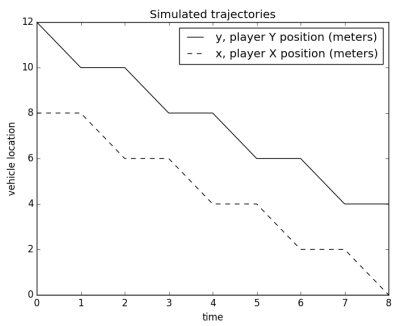


Figure 5: Simulation of turn taking game, equal starts.

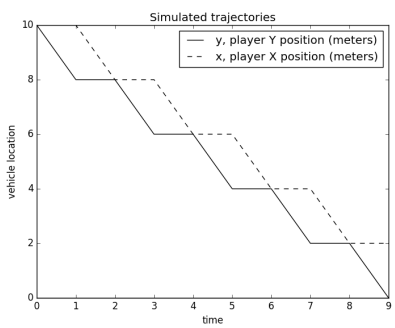


Figure 6: Sequential chicken state values.

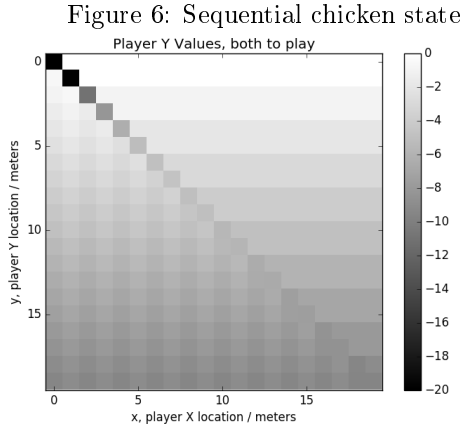


Figure 7: Sequential chicken optimal strategy.

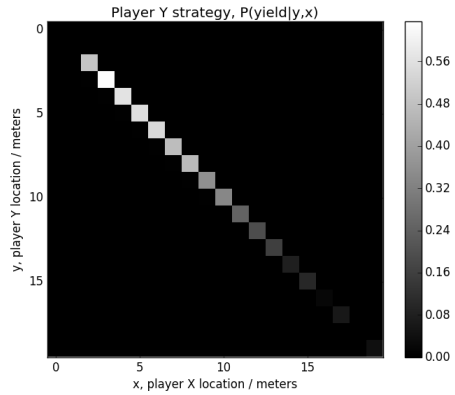
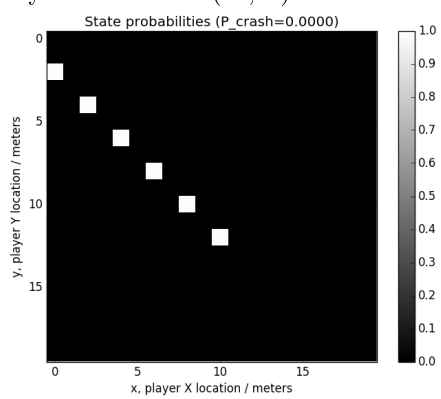


Figure 8: Sequential chicken state probabilities, from asymmetric start (12,10).



space are very close to 0 or 1, so the outcome is almost deterministic as in the turn-taking model of the same setting.

Fig. 10 shows state space probabilities when the vehicles start at identical distances $y = x = 10$. In this case, the outcome is different from the turn-taking model, because the game is fully symmetric but the symmetry is no longer broken by turn-taking. This means that both players must employ a policy consisting of mixed strategies until the symmetry is broken by one of them. The optimal policy is to yield with an increasing probability as distance to collision decreases, as seen in fig. 8. Fig. 11 is a typical sample simulation drawn from the above state probabilities. The most common outcome is for one vehicle to begin to yield at a random time, with yield

Figure 9: Sequential chicken simulation, from asymmetric start (12,10).

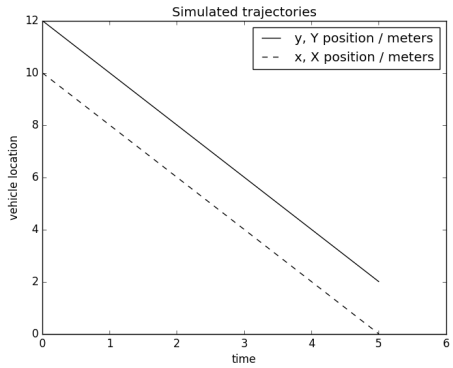
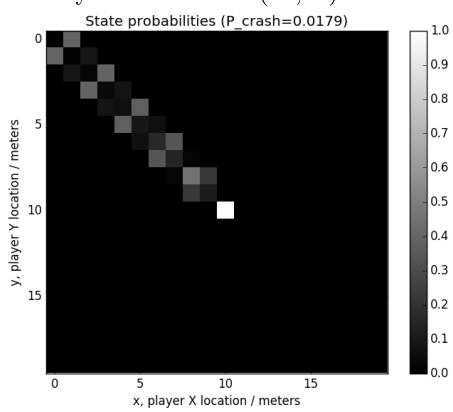


Figure 10: Sequential chicken state probabilities, from symmetric start (10,10).



probability increasing as the vehicles draw closer.

Occasionally, as in fig. 12, both players choose to yield at the same time, prolonging the conflict and costing them both a delay. Very occasionally a collision will occur as a result of this process, when the players fail to negotiate priority before both reaching the intersection together. This is rare, but must have a non-zero probability, computed and denoted as P_{crash}.

The collision probability with $U_{crash} = -20$ is 1.79%, and with $U_{crash} = -100$ it lowers to 0.7%. It is interesting to test how choice of this scaling (versus the fixed U_{time}=1) affects the residual collision probability, because if it has a large effect then any realistic model will require scaling calibration against some empirical data.

Figure 11: Sequential chicken typical simulation, from symmetric start (10,10).

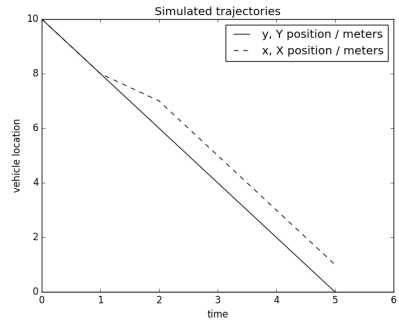


Figure 12: Sequential chicken atypical simulation, from symmetric start (10,10).

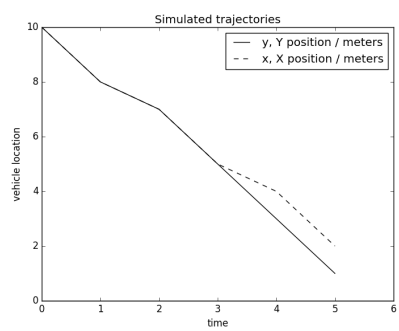


Figure 13:

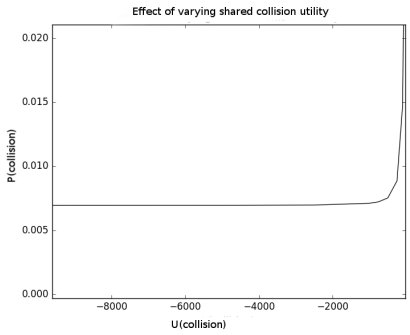
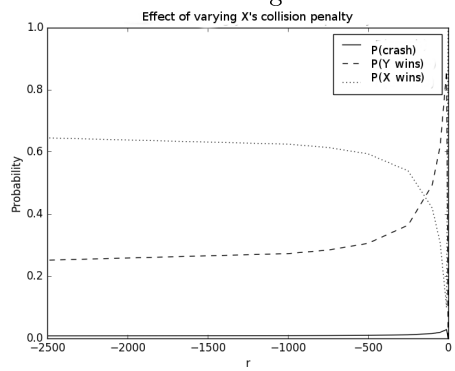


Figure 14:



Assuming that both players continue to share the same utility function, and know this to be the case, fig. 13 shows the effect of varying the collision utility scale while keeping the time utilities fixed.

This suggests that choices in range [-1000,0] do have large effects on the collision probability, but penalties worse than -1000 are much the same as each other in this effect. In real life, the cost of crashing a car (even very slightly) is almost always vastly larger than that of losing a few seconds journey time. This graph suggests that for these realistic penalties, the precise choice of shared collision penalty values is unimportant as long as it is over 1000 times worse than a 1 second time delay penalty.

Asymmetric utility model. As hypothesized, asymmetric collision penalties (such as our AV encountering a strong SUV or a weak cyclist or pedestrian) have a large effect on who must yield, and with a very small change in the probability of actual collision. This small change is the key to breaking symmetry and ensuring strong probability of the weaker player yielding fig. 14.

4 Conclusion

In all the models, when one agent has any small advantage it is usually – but probabilistically – optimal for both agents for the strong one to take the priority and the weak one to yield.

It is essential that there is some small but strictly non-zero probability of collision being allowed to occur as a consequence of both sides' optimal strategies. It is impossible for an AV to make any progress at all if this is not the case, because given this knowledge, every single other road user could dominate them in any conflict - even pedestrians jumping out in front of them for fun as seen in real-world trials [9]. Under these models, it is essential that AVs are programmed with a non-zero probability of deliberately causing a collision. This may be legally difficult, as such programming may be argued to constitute not only manslaughter but also murder, being rationally pre-meditated by the software engineer.

These results are interesting as they suggests that purchasing SUVs, or armoring our autonomous vehicle like an SUV, is a very rational strategy, not in order to better survive the rare collisions that do occur, but to ensure a high probability of other vehicles getting out of our way to save our time and money on delays. By adding armor to our own vehicle we can make the optimal strategy for the other player yielding tend towards certainty at every encounter.This also models what will happen when our AV encounters a pedestrian. The larger cost of collision to the pedestrian than to our AV gives us a strong position, from which we can act aggressively and be confident that the pedestrian will yield. The answer to 'When should the chicken cross then road?' is 'quite rarely if there is a car coming, but with non-zero probability'.

We found that the way in which the models quantize time is important. The turn-taking model artificially removed most of the subtly of game theory by breaking its symmetry via the turn taking mechanism itself. This suggests that such a quantization is not a good model for the real world, it hides the main problem of the scenario from the start. A related modeling issue around time relates to Zeno's Paradox. In the models presented here, time ticks are discrete and of equal length. It might be argued that two Zeno-like players could choose to define each of their ticks to have half the duration of the previous one, and thus create an infinite number of ticks which would

be certain to eventually lead to an asymmetric yield and avoidance of collision. Formal mathematical analysis of this claim could form future work, though in practice, any human or machine compute system has some finite limit on its computation speed.

Extended models should handle speed more realistically. Rather than just two discrete speeds, a continuum of speeds should be available, including stopping at a complete halt. Continuous speeds may require sampling approximations to compute over, while complete halts allow for potentially infinite time games which require further consideration to model. Human drivers when faced with, for example, a busy motorway merge, may gradually slow down towards a halt at the end of the slip-road, while drivers in their path may do the same. Perhaps under a continuous speed model this behavior can be shown to converge safely as everyone slows down towards a halt and reduces both the probability and penalty of collisions. Nevertheless, the underlying logic must still hold - that there must be a credible threat of a non-zero probability of causing some collision, in order that the other party cannot take advantage of the AV every time. Future models should add further realistic details to the framework. Real drivers do not know each other's utility functions and must infer them in an information game during the interaction. This could include giving and reading signals about utility such as the model, age, colour and cleanliness of their cars, their lateral positions on the road, their facial expressions and hand gestures as well as more formal car signaling via light flashing and horn usage. Real drivers may not have Markovian time delay utilities and more detailed models should allow for time dimensioned value functions $v(y, x, t)$ rather than than simpler $v(y, x)$ used here. Traffic regulations and conventions such as legally binding and non-binding signs and lights, and the cost of public humiliation or legal action for being seen or recorded breaking them should be added to modify utilities. Such models suggest new signaling conventions for autonomous vehicles, such as use of V2V radio communications and virtual currencies to aid negotiations.

References

[1] JB Bidwell, RS Cataldo, and RM Van House. Chassis and control details of Firebird III. Technical report, SAE Technical Paper, 1959.

[2] Ken Binmore. *Playing for real: a text on game theory*. Oxford university press, 2007.

[3] JD Bolland, MD Hall, D Van Vliet, and LG Willumsen. Saturn: Simulation and assignment of traffic in urban road networks. In *Proceedings of the International Symposium on Traffic Control Systems, Berkeley, Calif. D*, volume 2, pages 99–114, 1979.

[4] Federico Ciliberto and Elie Tamer. Market structure and multiple equilibria in airline markets. *Econometrica*, 77(6):1791–1828, 2009.

[5] Rune Elvik. A review of game-theoretic models of road user behaviour. *Accident Analysis & Prevention*, 62:388–396, 2014.

[6] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.

[7] Changwon Kim and Reza Langari. Game theory based autonomous vehicles operation. *International Journal of Vehicle Design*, 65(4):360–383, 2014.

[8] Hideyuki Kita. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. *Transportation Research Part A: Policy and Practice*, 33(3):305–312, 1999.

[9] Ruth Madigan, Tyron Louw, Marc Dziennus, Tatiana Graindorge, Erik Ortega, Matthieu Graindorge, and Natasha Merat. Acceptance of automated road transport systems (ARTS): an adaptation of the UTAUT model. *Transportation Research Procedia*, 14:2217–2226, 2016.

[10] Fanlin Meng, Jinya su, Cunjia Liu, and Wen-Hua Chen. Dynamic decision making in lane change: Game theory with receding horizon. In *Conference: 11th UKACC International Conference on Control*, 08 2016.

[11] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[12] Thomas R Palfrey. Implementation in Bayesian equilibrium: the multiple equilibrium problem in mechanism design. 1990.

[13] Christos H Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 82–91. Society for Industrial and Applied Mathematics, 2005.

[14] Elaine Rich and Kevin Knight. Artificial intelligence. *McGraw-Hill, New*, 1991.

[15] Mauricio Soto-Alvarez and Petri Honkamaa. Multiple hypotheses data association propagation for robust monocular-based slam algorithms. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6543–6547. IEEE, 2014.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[17] K Vela Velupillai. Uncomputability and undecidability in economic theory. *Applied Mathematics and Computation*, 215(4):1404–1416, 2009.

Empirical game theory of pedestrian interaction for autonomous vehicles

Fanta Camara[1,2], Richard Romano[1], Gustav Markkula[1], Ruth Madigan[1], Natasha Merat[1] and Charles Fox[1,2,3]

[1] Institute for Transport Studies (ITS), University of Leeds

[2] School of Computer Science, University of Lincoln

[3] Ibex Automation Ltd

Abstract Autonomous vehicles (AV's) are appearing on roads, based on standard robotic mapping and navigation algorithms. However their ability to interact with other road-users is much less well understood. If AVs are programmed to stop every time another road user obstructs them, then other road users simply learn that they can take priority at every interaction, and the AV will make little or no progress. This issue is especially important in the case of a pedestrian crossing the road in front of the AV. The present methods paper expands the sequential chicken model introduced in (Fox et al., 2018), using empirical data to measure behavior of humans in a controlled plus-maze experiment, and showing how such data can be used to infer parameters of the model via a Gaussian Process. This providing a more realistic, empirical understanding of the human factors intelligence required by future autonomous vehicles.

1. Introduction

Autonomous vehicle technology is currently claimed by many commercial organizations involved in operating experiments which apply robotic navigation algorithms to on-road vehicles. Robotic mapping and navigation are mature research areas (Thrun, 2005) and open-source software stacks are now available to enable a car to pick and follow a route around static environment (Kato et al. 2015). In contrast, the human factors involved in real-world driving in environments containing other road users are not well understood. Other road users include human drivers and pedestrians, and eventually other autonomous vehicles, which may or may not be able to communicate with one another. A recent study operated an autonomous minibus continuously on a real commuter route around La Rochelle, France and Trikala, Greece, for several weeks (Madigan et al., in preparation) and suggested that once the local population has learned that the vehicle was programmed to be perfectly safe and stop at any obstacle, pedestrians and other vehicles would regularly take advantage of the AV to push in front of it. In some cases pedestrians did this simply for fun – delaying the vehicle and its real-world commuter passengers on their way to work. Human drivers do not simply give way to all such threats. Instead they appear to use their knowledge of and predictions about the psychology of other road users to interactively negotiate for priority. This can include dominant behavior – driving straight toward a pedestrian at full speed, or even accelerating, to encourage them to yield and return to the pavement; or polite behavior – backing off to allow them space to cross. A well-known scenario is for two pedestrians to meet similarly and make several attempts dodging from side to side in order to agree on how to pass. Two polite road users can be almost as dangerous as two dominant ones – by continually trying to yield to one another they get closer to a collision without the decision being made.

A recent study (Fox et al., 2018) proposed an abstract mathematical model of two agents – which may be AVs, human drivers, and pedestrians - - meeting at an unsigned intersection and negotiating for priority. This model is deliberately simple and intended as a foundation whose structure can be extended with many additional details. In addition, it contains free parameters decrypting human preferences. The present study illustrates how such parameters can be fit to human data as a method of measuring behavior which combines empirical Game Theory and a Bayesian Gaussian Process analysis.

Game theory of human driven vehicles is used extensively in macroscopic traffic modeling via Wardrop equilibrium in flow networks (Bolland et al., 1979) with focus on route selection in large, economy-like, markets of many road users rather than microscopic pairwise interactions. Where game theory has been applied to

Proceedings of Measuring Behavior 2018, (Manchester, UK, 6-8 June 2018).
Editors: Spink A.J. *et al.* www.measuringbehavior.org

pairwise traffic decisions, it has mostly been at the level of simple single-shot games as reviewed in (Elvik, 2014). In a few cases such as lane-changing (Meng et al., 2016; Kim and Langari, 2014) and merging (Kita, 1999) it has been extended to sequential games as used here, but not for AV-pedestrian interactions as here.

2. Methods

2.1 Sequential Chicken model

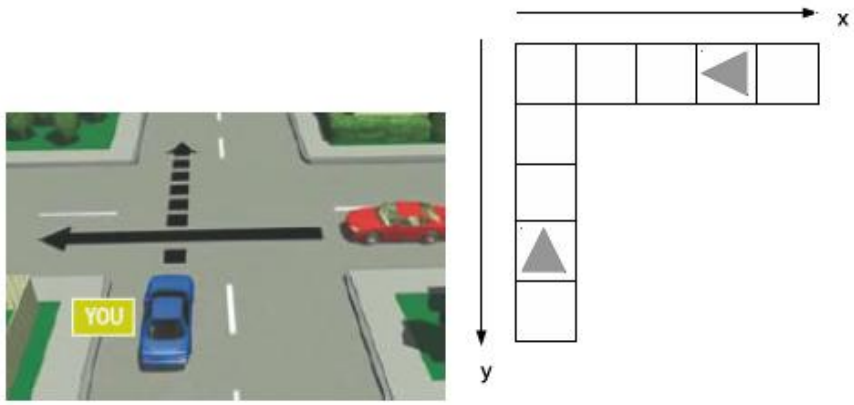


Figure 1 (left): two agents negotiating for priority at an intersection. (right): game theory model of the scenario.

The "Sequential Chicken" model of (Fox et al., 2018) represents a simplified version of the class of scenarios shown in fig. 1 (left) and fig. (2). In these scenarios, two agents (which may each be an AV, human driver of various types, and/or pedestrian) called Y and X approach the same intersection. If neither agent yields, they will collide and each receive a negative utility U_{crash}. Otherwise, they each reach their destination at some time delay, T, from the start of the game. The model assumes a linear value of time, so the total (negative) utility of an agent arriving at time T is $-TU_{time}$. The basic model assumes both agents share the same parameters U_{crash} and U_{time} and both know this to be the case. It assumes that no lateral motion is permitted, and that there is no communication between the agents other than seeing each other's positions. It discretizes positions into a plus maze structure of grid squares as in fig. 1 (right), and discretizes time into turns in which both players must select from two speeds, of either 1 square per turn or 2 squares per turn. The model is similar to a board game where each turn, both players make their choices in secret then and reveal them together before acting on them.

The model is called "Sequential Chicken" because its theoretical optimal strategies are solvable using Game Theory, and related to the famous game theory model of Chicken. In (non-sequential) Chicken, two drivers Y and X face each other and choose (as a single decision) whether to drive straight at each other or swerve away[1]. Swerving makes one the nominal "loser", but in practice, penalties for both are much worse if neither swerves and they collide. The game is considered a draw if both swerve. In game theory, this is represented by a payoff matrix, showing numerical utilities to the two players as in table 1.

$Y \setminus X$	a_X=*swerve*	a_X =*straight*
a_Y=*swerve*	(0,0)	(-1, +1)
a_Y =*straight*	(+1, -1)	(-100,-100)

Table 1: The game of Chicken.

[1] It is convenient to use Y, X ordering throughout so that matrix indices $M_{y,x}$ correspond to visualizations.

Standard Game Theory provides (Nash) *equilibria* for games in this matrix form, which are configurations of action selections such that neither player would change if they knew the other's choice. (Fox et al., 2018) provide arguments for selecting between these equilibria in real-time traffic situations so as to choose the most "rational" strategy. In general, the best strategies are probabilistic, specifying that each player should choose between the two actions with certain probabilities. (If they were deterministic, then each player could know the other's move in advance and exploit that knowledge, contradicting their supposed optimality.)

Chicken is a one-shot game – both players make just one decision then act on it. Sequential Chicken can be viewed as a sequence of one-shot (sub-)games, which can be solved similarly. Write (y,x,t) to describe the player's current locations as in fig. 1 (right) at turn t and let their available actions be the speed choices $a_Y, a_X \in \{1,2\}$. Rather than winning, losing, or crashing, the outcomes of the game at state (y,x,t) are that the players reach a new state, ($y+a_Y$, $x+a_X$, $t+1$). This may be a "game-over" state, where either one player has reached the intersection and/or a crash has occurred, or it may be another intermediate state leading to another game in the sequence. Define the *value* $v_{y,x,t} = ({v_{y,x,t}}^Y, {v_{y,x,t}}^X)$ of state (y,x,t) as the expected utility of the game to each of the two players, assuming that both play optimally from the state. Then the sub-game at time t can be written as a standard game theory matrix,

$$v_{y,x,t} = v(\begin{bmatrix} v(y-1,x-1,t-1) & v(y-1,x-2,t+1) \\ v(y-2,x-1,t+1) & v(y-2,x-2,t+1) \end{bmatrix})$$

which can be solved using recursion, Game Theory, and equilibrium selection to give values and optimal strategies at every state (given in Fox et al., 2018). Under a mild approximation (Fox et al., 2018) the solution may further drop all dependency on time t so that state values and optimal strategies becomes functions only of the player positions (y,x).

The model shows that if the two players play optimally, there must exist a non-zero probability for a collision to occur. This corresponds to the intuition that a perfectly safe vehicle which yields in every case will make no progress. Players must include some credible threat of collision to avoid being taken advantage of.

Figure 2: A "Barnes dance" or "scramble" crossing between Hollywood and Highland, Los Angeles. Real-world pedestrian-pedestrian interactions similar to the experiment occur between users of the two diagonal crossings.

2.2 Experiments

The model allows for computation of optimal strategies given parameters (U_{crash}, U_{time}) but does not specify numerical values of these human preference parameters. Eventually we would like to measure such parameters from real-world traffic data such as CCTV of vehicles and pedestrians on roads. The present study is intended to

illustrate only the method for measuring such behaviors rather than report results, so instead uses a simplified proxy. Rather than interact physically, pairs of human subjects play a board-game version of Sequential Chicken as above. Their moves are recorded and used to show how to fit the parameters.

Sixteen participants aged from 22 to 48 were divided into 8 groups of 2 players. Each group's experiments took place with players and an experimenter seated around a table in university room and last for about 10 minutes. All experiments were conducted in accordance with University of Lincoln research ethics policy.

Natural game. Two players, one designated as player Y and the other as player X were sitting around a table, they were asked to play the chicken game on a plus-maze shaped board with discrete space as shown in Figure 1(right). Player Y was starting from y=10 and player X from x=10 such that they were both starting 10 squares before the intersection. Players were first told that their goals were to avoid a collision and to be the first to cross over this intersection as quickly as possible as if they were trying to commute to their office to work there in a morning. Players were each given two cards containing the numbers 1 and 2, and told that at each turn they should select one in secret then both reveal their cards and make their moves on the board simultaneously.

Chocolate game. After playing 3 natural games, each group played a further 3 games in which specific rewards were specified in advance. Players were told that the winner is the first to pass the intersection, and will receive two chocolates (Celebrations, Mars Inc.), while the second to pass the intersection will receive one chocolate, that both will receive no chocolates if there were any collision. Chocolates were provided after each game as in the instructions.

2.3 Measuring behavior

We wish to infer the parameters $\theta = (U_{crash}, U_{time})$ from the observed data, D, assuming that players play Bayes-optimally as prescribed by the model, M. By Bayes' theorem, the posterior belief over these is given by,

$$\mathrm{P}(\theta|M,D) = \frac{\mathrm{P}(D \mid \theta, M)\mathrm{P}(\theta| M)}{\sum \mathrm{P}(D| \theta',\ M)\mathrm{P}(\theta'| M)}$$

Given a set of possible θ values. assume a flat prior over θ, so that,

$$P(\theta|M,D) \ \propto \ P(D \mid \theta, M).$$

Hence we can work only with the data likelihood term, $\mathrm{P}(D \mid \theta, M)$, which is given by,

$$\mathrm{P}(D \mid \theta, M) = \prod_{game} \prod_{turn} \mathrm{P}\big(d_Y^{game,turn} \,|y, x, \theta, M\big)\, \mathrm{P}\big(d_X^{game,turn} \,\big|\, y, x, \theta, M\big)\,,$$

where $d_{player}^{game,turn}$ are the observed action choices, y and x are the observed player locations at each turn of each game. The action probabilities $P\big(d_Y^{game,turn} \,\big|\, y, x, \theta, M\big)$ can be read directly from the optimal strategies.

The above assumes that model M is perfect. However, M assigns zero or near-zero probabilities to many events which are found to occur in the data. Uncontrolled, this can be catastrophic for any Bayesian model of real data. We know that real humans may behave near-optimally but can make mistakes in perception (of the state of the game, such as the values of y and x) and in action selection. Thus, we weaken the model used in the analysis to a noisy but more forgiving version M'. This is a mixture model which generates each observation according to M with some probability s, and generates unbiased random actions otherwise,

$$P(D \mid \theta, M') = \prod_{game} \prod_{turn} [(1-s)\mathrm{P}\big(d_Y^{game,turn} \,|y, x, \theta, M\big)\, \mathrm{P}\big(d_X^{game,turn} \,\big|\, y, x, \theta, M\big) + s(\frac{1}{2})].$$

These equations are unlikely to have analytic solutions for θ, as they include elements of the optimal strategies computed via a long game theoretic process. To compute with them, we may sample various values of θ and observe $P(D \mid \theta, M')$. In practice, we work in log-space to reduce numerical errors, and consider $\log P(D \mid \theta, M')$.

Given a sample of these log-likelihoods, we may then fit a Gaussian Process (Rasmussen & Williams, 2006) regression model to them. Assuming that log-likelihood is a smooth function of the parameters, this provides estimates of the log-likelihoods at locations between the sampled values and can also show the uncertainty in these estimates. A Radial Basis Function kernel was used in the Gaussian Process model, with variance set to a minimum after obtaining smooth likelihood surfaces.

3. Results

Before performing the full analysis, we show in table 1 a rough overview of the game results. More collisions have occurred in the chocolate game than in the natural game. Define "delay" as the distance in boxes, or time in seconds, between the arrival at the intersection of the first player to arrive and the second player to arrive. The average delay in distance (box) is about 1.368 for the natural game but is reduced in the chocolate game (1.125). The time delay is computed following the formula: $\boldsymbol{d} = \boldsymbol{vt}$ where $\boldsymbol{d}$ is the distance, $\boldsymbol{v}$ the speed is assumed to be the maximum (2) and $\boldsymbol{t}$ is the time delay.

Type of Game	Number of Games	Collisions	Average Delay (box)	Average Time Delay (s)
Natural	24	5	$\frac{26}{19} \approx 1.368$	$\frac{26}{19*2} \approx 0.684$
Chocolate	24	8	$\frac{18}{16} = 1.125$	$\frac{18}{16*2} = 0.5625$
Total	48	13	$\frac{44}{35} \approx 1.257$	$\frac{44}{35*2} \approx 0.628$

Table 2: Number of collisions and average time delay per game

Fig. 3 shows the results of fitting the Gaussian processes to likelihoods sampled along a regular grid, using s=1/10. The surfaces have a roughly radial structure. By visual inspection, the peak of the smoothed Gaussian Process surface in the natural case appears to lie somewhere in the region of $(U_{crash}, U_{time}) = (0,30)$ and for the chocolate case, around (0, 30). Fig. 4 shows a 1D slice through the 2D surface of Fig. 3(natural) which provides scale and uncertainty information.

4. Discussion

We have illustrated a new method for measuring and modelling human road user behavior, which is intended as part of a method to imitate or exploit such behavior in autonomous vehicle controllers. The model can be trained on human behavior then transferred into a controller to specify new robotic behaviors.

The particular experiment here has inferred posterior beliefs about the parameters of crash utility and time delay for human participants, in a simplified and artificial version of a real-world intersection negotiation task. We do not consider either the model or the experimental setup to be especially realistic versus real-world interactions, but they are sufficiently complex to illustrate the general method. This method is presented as an invitation for future work to extend both the model (e.g. with continuous positions and speeds, uncertain beliefs in opposite player utility function, lateral motion) and the experimental data collection (e.g. use of "big" real-world CCTV video of road-user interactions) stages to be more realistic.

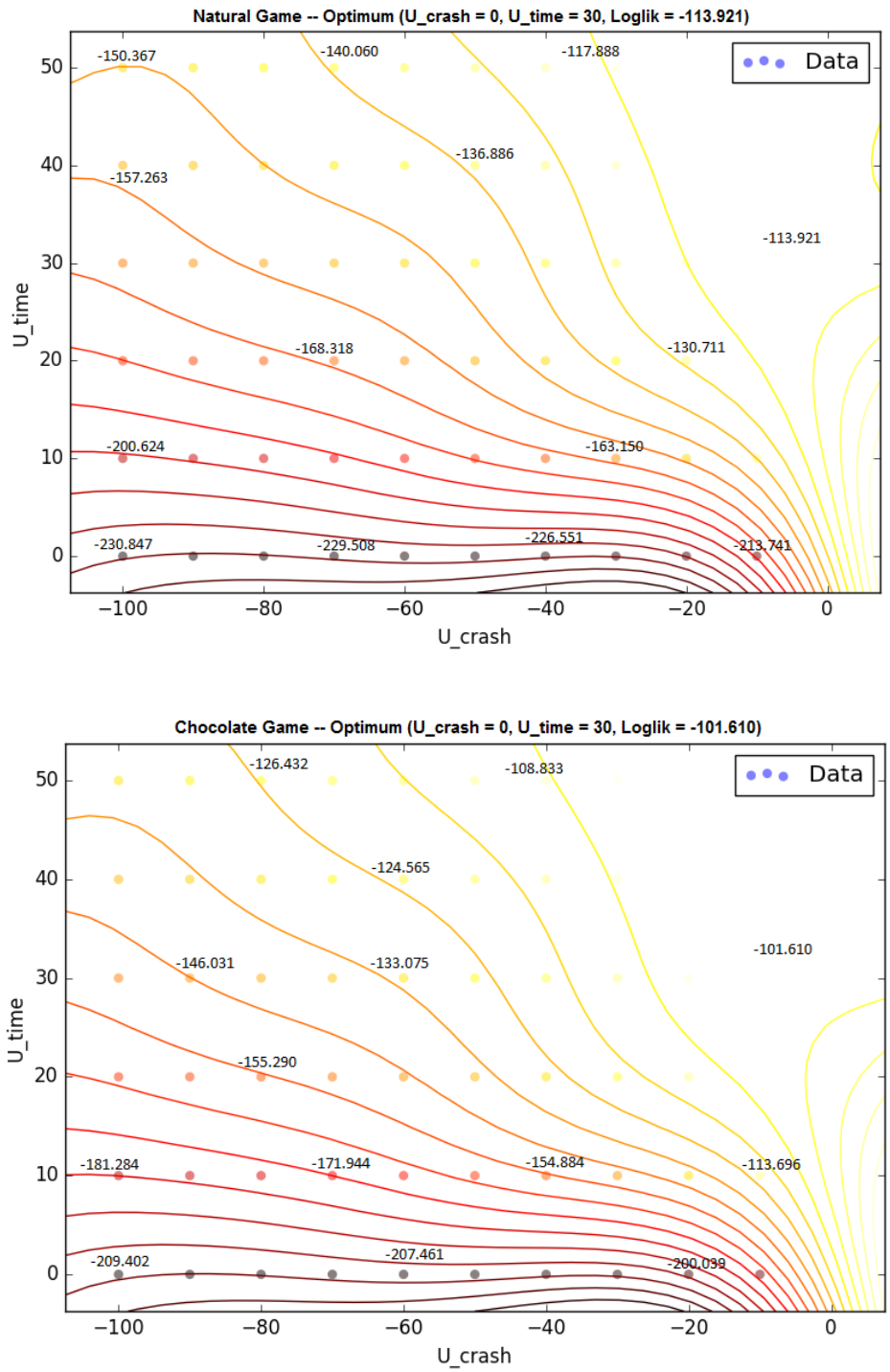


Figure 3: Gaussian process regression shows the likelihood (and hence, posterior belief) surfaces for the parameter values($U_{crash,}$ U_{time}), for the natural and chocolate experiments. The sampled parameter locations are shown by the dots. (This figure should be viewed in color).

Some conclusions from the results of the particular experiment analyzed which may be of interest, despite its artificial setting: From the viewpoint of real-world interactions, it is surprising to see a preference towards time-saving over crash avoidance in the players. In the real world, the cost of crash is usually millions of times more than the value of a second of commuting time, but here the players behave the opposite way. The effect is more pronounced in the chocolate game than in the natural one. This may be because the board-game-like nature of

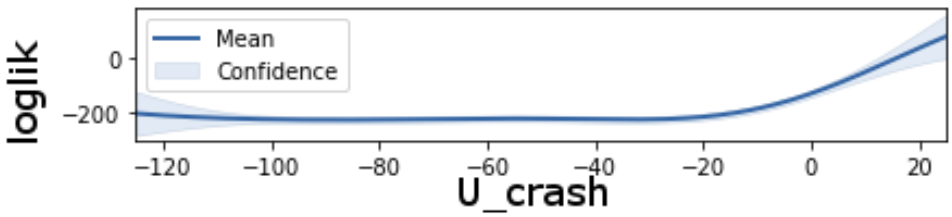


Fig. 4. Horizontal slice though the Gaussian Process of fig. 3(natural), at U_time=10, showing mean and standard deviation of belief about the log likelihood, $\log P(D \mid \theta, M')$ over the parameter space.

the experimental setting encourages the players to think in terms of winning and losing rather than just optimizing their personal utility. In the chocolate game this is made to be explicit by rewards the players specifically as winner and loser in the interaction. This leads to more collisions in this more competitive game.

The shapes of the likelihood surfaces are roughly radial, which suggest that it may only be the ratio between the two parameters which is important rather than their absolute values. Possibly the angles of the contour lines define equivalence classes of parameters. The number of experiments performed was quite small, and future work including further experiments may produce more peaky posterior distributions.

By using this method to fit more realistic models and data to human pedestrian-vehicle and vehicle-vehicle interactions, we can understand and predict behavior of road users to allow autonomous vehicles to more optimally interact with them. The present model requires a non-zero probability of collision to exist as a threat in all interactions. Future work could extend the model with additional, smaller negative utilities such as humiliating annoying pedestrians with horns or spraying them with water jets, in order to reduce or eliminate the necessity of actually hitting them whilst still allowing autonomous vehicles to make progress.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreements No 723395 (InterACT) and No 645376 (FLOBOT).

References

Bolland, J., Hall, M., Van Vliet, D., and Willumsen, L. (1979). Saturn: Simulation and assignment of traffic in urban road networks. In Proceedings of the International Symposium on Traffic Control Systems, Berkeley, Calif. D, volume 2, pages 99–114.

C.W. Fox, F. Camara, G. Markkula, R.A. Romano, R. Madigan and N. Merat. (2018) When should the chicken cross the road?:Game theory for autonomous vehicle - human interactions. In Proceedings of VEHITS 2018, Funchal, Madeira, Portugal.

Elvik, R. (2014). A review of game-theoretic models of road user behaviour. Accident Analysis & Prevention, 62:388–396.

Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K., and Hamada, T. (2015). An open approach to autonomous vehicles. IEEE Micro, 35(6):60–68.

Kim, C. and Langari, R. (2014). Game theory based autonomous vehicles operation. International Journal of Vehicle Design, 65(4):360–383.

Meng, F., Su J., Liu, C., and Chen, W.-H. (2016). Dynamic decision making in lane change: Game theory with receding horizon. In Conference: 11th UKACC International Conference on Control.

Rasmussen, C. E., & Williams, C. K. (2006). Gaussian processes for machine learning. 2006. *The MIT Press, Cambridge, MA, USA*, *38*, 715-719.

Thrun, S., Burgard, W., and Fox, D. (2005). Probabilistic robotics. MIT press.

Madigan, R., Nordhoff, S., Fox, C., Ezzati Amini, R., Louw, T., Wilbrink, M., Schieben, A., and Merat, N. (in preparation). Understanding interactions between automated vehicles and other road users. Manuscript in preparation.

Rumex and Urtica Detection in Grassland by UAV

Adam Binch[2] **Nigel Cooke[2]** **Charles Fox[1,2]**

[1] School of Computing, The University of Lincoln, UK

[2] Ibex Automation Ltd, UK

A paper from the Proceedings of the
14th International Conference on Precision Agriculture
June 24 – June 27, 2018
Montreal, Quebec, Canada

Abstract. *Previous work (Binch & Fox, 2017) used autonomous ground robotic platforms to successfully detect Urtica (nettle) and Rumex (dock) weeds in grassland, to improve farm productivity and the environment through precision herbicide spraying. It assumed that ground robots swathe entire fields to both detect and spray weeds, but this is a slow process as the slow ground platform must drive over every square meter of the field even where there are no weeds. The present study examines a complimentary approach, using unmanned aerial vehicles (UAVs) to perform faster detections, in order to inform slower ground robots of weed location and direct them to spray them from the ground. In a controlled study, it finds that the existing state-of-the-art (Binch & Fox, 2017) ground detection algorithm based on local binary patterns and support vector machines is easily re-usable from a UAV with 4K camera despite large differences in camera type, distance, perspective and motion, without retraining. The algorithm achieves 83-95% accuracy on ground platform data with 1-3 independent views, and improves to 90% from single views on aerial data. However this is only attainable at low altitudes up to 8 feet, speeds below 0.3m/s, and a vertical view angle, suggesting that autonomous or manual UAV swathing is required to cover fields, rather than use of a single high-altitude photograph. This demonstrates for the first time that combined aerial detection with ground spraying system is feasible for Rumex and Urtica in grassland, using UAVs to replace the swathing and detection of weeds then dispatching ground platforms to spray them at the detection sites (as spraying by UAV is illegal in EU countries). This reduces total time requires to spray as the UAV performs the survey stage faster than a ground platform.*

Keywords*.* *UAV, weeding, Rumex, Urtica, robotics*

1 Introduction

Autonomous weeding of grassland may enhance dairy and sheep farm productivity in less favoured farm areas, and also reduce chemical herbicide usage in these environments and run-off into their surrounding ecosystems, including the human water supply. Many such farms, unlike large agribusiness operations, are small family-owned businesses operating at low margins. This includes areas of hill grassland which is currently uneconomical to spray either by precise manual backpack sprayer application to weeds of generic herbicides (at minimum labour wage costs) or by tractor-mounted bulk spray (due to the higher costs of selective herbicides). Hence, these areas remain unsprayed and the benefits of additional food production via grazing of the areas occupied by weeds is not realised. Autonomous robotic precision spraying of these weeds provides a possible alternative solution, which could spray low doses of generic herbicide onto individual weeds, like a manual backpack sprayer, but without the associated and prohibitive minimum wage labour costs.

In a recent study [1], we performed a comprehensive evaluation and optimisation of all known machine vision approaches for detection of Rumex and Urtica in grassland from a ground robot platform. After standardising and optimising the methods, for the first time on a "big" data set large enough to report the comparison with high statistical significance, we found that a combination of Local Binary Patterns (LBP) with Support Vector Machine (SVM) with certain parameters gave practically usable performance, giving 83% accuracy on classification of 12cm square regions of ground, and rising to 95%+ where two or more observations of the same region are fused. This was notably lower than previously claimed results by original authors of the algorithms tested, due to the higher standard of independent testing imposed in the comparison study, and the new and larger data sets used.

To move further towards a practical system for weed control, it is next necessary to consider how such recognition should be used as a part of practical agri-systems methods. The previous study initially assumed that weed detection and spraying would be performed by the same ground robot, swathing (i.e. driving up and down in a ‘lawnmower stripes pattern’) over each square meter of a field, detecting and spraying any weeds found there. We have performed pilot studies with ground platforms following this approach, and along with the results of the previous paper, have found two issues with it. First, swathing with a ground platform is inefficient because weeds are not evenly distributed in most fields, and there are many areas which have no weeds in them at all. Driving over every square meter is a waste of time and (more importantly for unmanned systems) energy. Human manual sprayers do not blindingly swathe fields, but use both their prior knowledge (weeds are more likely to grow around edges of fields, and in shaded, damp areas) and their distant vision (distant uncertain dark green blobs may or may not be weedy areas, so warrant travelling to investigate them in more detail). Second, the previous study discovered a strong drop-off in recognition performance as a function of distance from the ground robot, due to its perspective warping assumptions breaking down with distance. While it would be possible, as in other studies (most of which investigated the much simpler problem of weed detection in arable row crops rather than grassland), to mount the robot’s camera facing vertically downwards instead of its current 22.5 degrees, this would preclude it from detecting weeds not directly under it, which in turn would preclude the fusion of multiple observations of weeds ahead of the vehicle before spraying which were shown to be needed to obtain the higher 95% accuracies.

Both of these issues could be addressed by a two-platform system, using unmanned aerial vehicles (UAVs) to perform detection and the ground platform to relocate the detections and perform the spraying. (EU law prohibits spraying from UAVs directly). UAVs can view much larger areas - potentially whole fields - in single camera frames, and/or may also aerially swathe over areas at low altitudes more quickly than ground platforms if higher resolution is required. Views

from UAVs are different than from ground platforms, and it is not obvious whether systems developed for ground use are re-usable from the air. The present study tests this hypothesis, including UAV images both directly vertical and at a 45 degree tilt (similar to training data from the ground platform). It also measures the effect of altitude to give the first indication of whether aerial swathing is required or whether single high-altitude images can be used, thus providing the first estimates of the time required to detect weeds over whole fields in this manner.

2 Methods

The objective of the experiments outlined here is to report the performance of a method utilised for the classification of grass vs weeds (i.e. a two-way classification). More specifically we wish to assess the performance of the method when there is a miss-match between the training data and the test data, where the training data was captured from cameras mounted to a ground robot and test data was captured from cameras mounted to a UAV. Further, we assess the performance of the method for two types of test data captured using the UAV. Firstly using images captured at 90˚ pitch down (i.e. a birds-eye view) and secondly using images captured at 45˚ pitch down (yielding a perspective view of the ground).

The performance of the method was assessed by training it on the training images and then using the method's (learned) parameters to predict the test images (i.e. an image containing grass or weeds).

2.1 Training image acquisition and pre-processing

A plot of Rumex and a plot of Urtica, each measuring approximately 3m squared, were built on a dairy grass-land farm (South Yorkshire, UK) in a region of the farm exposed to direct sunlight throughout the day. This was achieved by taking slabs of Rumex and Urtica, each measuring approximately 0.2 m squared, from a working field and transplanting them into their respective trenches.

An auto-focusing camera (C920, *www.logitech.com*) was mounted to a tracked robot on its left side facing out at right angle yaw to the direction of travel, and at 22.5 degrees pitch down. The robot travelled in circles around each plot (figure 1) capturing stereo pair images at 1080HD resolution. The distance between the robot and the edge of the plots was randomised between 0 m and 1m. This setup was devised so that the plot contents filled the images and gave a clear view over approximately 1 m square area of ground. Images of grass were also acquired by driving the robot on a random path through working fields. Approximately half of this data was acquired under bright or sunny weather conditions, whilst the other half was acquired under overcast weather conditions. Data was captured during May 2016, over the course of 4 days at random times of day from sunrise to sunset. Approximately a third of a terabyte of usable image data was acquired in total.

Images were pre-processed in three steps: colour calibration, perspective dewarping, and windowing. Images undergo a continuum of colour transformations due to lighting and weather conditions. In order to compensate for these effects images were modified using a 'colour calibration' algorithm. To a certain extent this operation acts to colour standardise images. Images were colour calibrated by measuring the blue, green and red light intensities from a colour bar (figure 1b) present in images and adjusting the colour intensities in comparison images (i.e. images that were to be colour calibrated) to be as close as possible to the colour intensities of a reference image (for a full account of this process see [1]).

Perspective normalisation was performed via a projective transform which maps a 1.16 m width by 1.0 m depth ground area (figure 1b) into a 700 pixel width by 600 pixel height image, whose geometry is identical to that of a vertical, overhead camera. A single spray target radius was judged to be a ground area of 106 mm squared which corresponds to a 64 × 64 square pixel

region of the image. Thus the dewarped image was split into regular 64 × 64 contiguous and non-overlapping windows.

(a) *(b)*

***Figure 1**: Training image acquisition (from [1]).*
(a) Plot of Urtica and ground robot (IBEX2) capturing training images from the plot using its on-board cameras. (b) Raw training image of Urtica obtained form the ground robot's cameras. Note the colour bar used for the colour calibration process and the grid annotation (outlined area) used for the perspective normalisation described in the main text (section 2.1). Image windows were obtained from the annotated region.

2.2 Test image acquisition and pre-processing

Images of grass, Rumex and Urtica were captured from 4K resolution, autos-stabilised cameras mounted to a DJI Phantom 4 UAV (*www.dji.com*) in May 2017, on the same field from which the ground robot images were acquired. All images were acquired over the course of a single afternoon under overcast weather conditions. Images of Rumex were captured by flying the UAV over a large region of a working field containing large patches of Rumex. Images of grass were captured by flying the UAV over a large region of the field containing grass only (care was taken to avoid the occurrence of weeds in these images). Images of Urtica were acquired by hovering the UAV (and moving it back and forth slightly) over the Urtica garden (note that the Urtica garden had undergone a year's worth of growth since the acquisition of the ground robot images, whilst the Rumex garden had been destroyed prior to the acquisition of the UAV's images). The UAV captured images at a vertical distance of 1 m above the ground contents. Two sets of UAV images (where a single set consists of images of grass, Rumex and Urtica) were acquired. In the first set the UAV's cameras were adjusted to face 90° pitch down yielding images with a 'birds-eye-view' of the ground (figure 2a), while in the second set the UAV's cameras were adjusted to face 45° pitch down yielding images with a perspective view of the ground (figure 2b).

For the 45° images, perspective normalisation was performed via a projective transform which maps the ground area indicated by the annotated region in figure 2b into a 1249 pixel width by 580 pixel height image whose geometry is identical to that of a vertical, overhead camera. The 1249×580 image was split into regular 64 × 64 contiguous and non-overlapping windows. For the 90° images the *whole* image was split into regular 64 × 64 contiguous and non-overlapping windows. Some of the Rumex images (both 90° and 45°) had to be removed because they contained large patches of grass.

(a) *(b)*

***Figure 2**: Test Image Acquisition.*

(a) Raw 90° test image of Urtica captured from the UAV's cameras. (b) Raw 45° test image of Urtica captured from the UAV's cameras. The grid annotation (outlined area) used for the perspective normalisation is also shown. Image windows were obtained from the annotated region.

2.3 Classification method

The classification method used in the present analysis involves extracting features known as local binary patterns (LBPs) [2] (figure 3) from image windows which are then passed into a support vector machine (SVM) [3] for classification.

Local binary patterns (LBP) are family of texture description features. Local means they are computed on local sub-windows of an image only, as a function of a center pixel and either its immediate or *r* pixel radius neighbours; binary means that the feature vector is binary, with each feature classed as either present or absent. Windows are converted to greyscale, then for each pixel $x_{i,j}$ in the window, LBP computes a vector,

$$[(x_{i;j} > x_{i;j+r});\ (x_{i;j} > x_{i-r;j+r});\ (x_{i;j} > x_{i-r;j});\ (x_{i;j} > x_{i-r;j-r});$$
$$(x_{i;j} > x_{i;j-r});\ (x_{i;j} > x_{i+r;j-r});\ (x_{i+r;j} > x_{i;j});\ (x_{i;j} > x_{i+r;j+r})].$$

There are 2^8=256 possible values of this feature vector. LBP computes the vector for each pixel in the window, then computes a 256-point histogram of the obtained values over the window. Further details are given interval [1] and the references therein.

Support vector classifiers (SVCs) separate images into their classes, firstly by creating a finite dimensional feature vector space and secondly by finding the linear affine hyper-plane that maximises the distance (known as the margin) between image classes in this feature space. Support vector machines (SVMs) extend the SVC by using non-linear functions known as kernels to enlarge the feature space. In the higher dimensional space the hyper-plane is linear, but non-linear when projected back onto the original feature space. As a consequence SVMs are more effective than SVCs as there are more dimensions over which different classes can be separated. Both SVCs and SVMs are modified by a 'tuning' hyper-parameter *C* which controls how many observations (training images) are permitted to violate the margin. The kernel used in the present analysis is the radial (RBF) kernel which has its own hyper-parameter γ (for a fuller yet concise explanation of the SVM with RBF kernel and its hyper-parameters see [1]).

(a) *(b)*

***Figure 3**: Feature extraction.*
(a) Image window of Urtica. (b) The local binary pattern (LBP) image of 3, with n=24, r=4. Note that the jagged pattern of the Urtica leaves is apparent in the LBP image.

For both training and testing, the hyper parameters of the LBP were set to n=24, r=4 and the hyper-parameters of the SVM were set to $C=2^9$, $\gamma=2^3$. These hyper-parameter values were obtained using the optimisation method outlined in [1].

2.4 Experiments

To train the classifier, a training set of 40,000 image windows was selected from the data set obtained from the ground robot's cameras. Half the image windows contained grass under an equal mixture of sunny and overcast weather conditions, whilst the other half contained weeds (an equal mixture of Rumex and Urtica) under an equal mixture of sunny and overcast weather conditions. This selection was achieved by uniform random sampling across the subsets of the data containing grass and weeds, respectively. Grass image test sets and weed image test sets (for any given experiment) each contained 1710 image windows selected using uniform random sampling. Weed image test sets contained an equal mixture of Rumex and Urtica.

We wished to assess classification performance as a function of the speed of the UAV. However, there are two problems with regard to obtaining speed results from the test data. Firstly, the test data was not acquired with the intention of obtaining speed results. Thus there were no subsets of the data acquired at pre determined speeds. In order to obtain these subsets we computed the dense optical flow occurring in a given image using a method based on Gunner Farneback's algorithm [4]. Note that this method computes optical flow for all points in the image and is therefore a global measure of optical flow (as opposed to a local measure of flow computed at interest points in the image). Due to the forward motion of the UAV and the (relatively) static nature of the ground contents a global measure of flow was deemed preferable as all pixels were assumed to be moving across the image at equal speeds and in the same direction. For two consecutive images, the Gunner Farneback method yields a change in position (i.e. a change in x and a change in y) for every pixel in the image and thus a Euclidean distance for all pixels in the image. Taking the median Euclidean distance provides a global measure of speed (distance over time) for each image. Then images can be binned into various speed intervals. Each bin thus contains a set of images that can be windowed and stored for analysis.

The second problem with obtaining speed results was due to a lack of speed variation in the weed images (Rumex and Urtica). For Rumex, there were only a few images containing mostly

Rumex (without large patches of grass). Those images were all captured when the UAV was moving at an approximately uniform speed. For Urtica, the UAV captured images over the Urtica garden. Thus the UAV remained localised (i.e. not moving or only moving very slightly) when capturing images of Urtica. In order to acquire classification performance for overall grass vs weed detection we assumed that the ratio between detection rates for grass and detection rates for weeds remains constant with respect to the speed of the UAV. To obtain this ratio we first measured classification performance using images captured at speeds ranging between 0.09 and 0.16 m/s (i.e. the smallest UAV speed bin considered in these experiments). For this experiment, we obtained the break down accuracies for grass and for weeds, and thus the ratio between them. Then, grass classification accuracy as a function of UAV speed was obtained from the grass image data (downsampled to 1080HD) and those accuracies and the ratio was used to obtain a proxy for weed classification accuracy as a function of UAV speed. Taking the average of those two values yields the overall accuracy as a function of UAV speed. Note that this experiment was carried out using the 90˚ only.

We also wished to assess classification performance as a function of the simulated altitude of the UAV. As the UAV increases in altitude its cameras cover a larger area of ground. At the same time a given (for example a 1m by 1m) area of ground is represented at a correspondingly smaller resolution. Specifically, if the UAV doubles its altitude, then the 1m square area of ground is represented in the image by half the number of pixels (a 3840×2160 pixel image covering a 1m square area of ground taken at an altitude of 3.28 feet (1m) corresponds with a 1920×1080 pixel image covering a 1m square area of ground taken at an altitude of 6.56 feet (2m)). Thus, simulating images taken by the UAV at different altitudes can be achieved by down-sampling its 4K images to different values. However, these images need to be subsequently up-sampled to 1080HD to correspond with the ground robot training data (such that a 64×64 windowed image covers a ground area of 106 mm squared - a single spray target radius). Note that these operations were performed using the 90˚ (birds-eye-view) images only. Also note that classification performance as a function of the simulated altitude of the UAV was assessed using images captured in the lowest speed bin.

3 Results

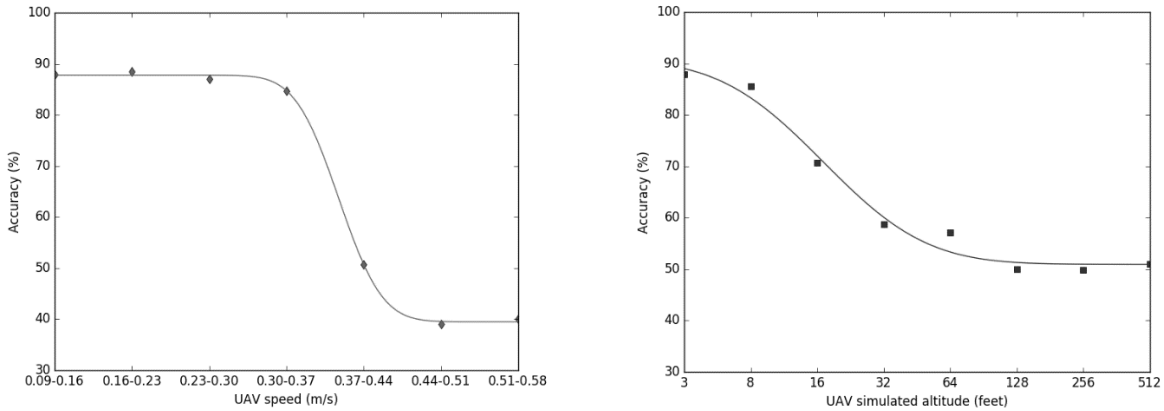
Table 1 gives the results of the experiment in which the ratio of grass to weed classification accuracy was determined (break down accuracies for grass and weeds) (first row: 45˚, second row: 90˚) which are compared with the results (third row) of the experiment with the ground robot test data, which was reported in [1]. ACC gives the proportion of images correctly classified (i.e. the overall accuracy of the method) and CI are confidence intervals justifying the significance of the accuracy percentages to two decimal places. The breakdown accuracies for grass, weed, Rumex and Urtica presentations are also given.

Experiments using the UAV test data (for both 90˚ and 45˚ images) yielded higher accuracies than the experiment using the ground robot test data, and the 45˚ UAV images yielded a higher accuracy than the 90˚ UAV images. Specifically, the 90˚ and 45˚ UAV images yielded accuracies of 87.89% and 90.39%, respectively, whilst the ground robot images yielded an accuracy of 82.88%. The better performance of the experiment with the 45˚ images as compared with the performance of the experiment with the 90˚ images is due (mainly) to their difference in grass image classification, with approximately 5% greater accuracy for the 45˚ images. For the UAV test data, Urtica classification was more accurate than Rumex classification, whilst the opposite pattern was found for the ground robot test data.

Figure 4a gives the accuracy of the classification method as a function of the speed of the UAV, in intervals spanning 0.07 m/s. The diamond shaped markers are the data points and the curved line is the best fitting mirrored cumulative Gaussian modified with an additive (controlling

intercept) and multiplicative (controlling magnitude) constant. The accuracy of the classification method decreases sharply beyond speeds of 0.3 m/s.

Figure 4b gives the accuracy of the classification method as a function of the simulated altitude of the UAV increasing in powers of 2. The square markers are the data points and the curved line is the best fitting cumulative Gaussian, again modified with an additive and multiplicative constant. The accuracy of the classification method decreases sharply beyond altitudes of 8 feet.



(a) *(b)*

***Figure 4**: Accuracy vs UAV speed and accuracy vs simulated UAV altitude. (a) Accuracy of the classification method as a function of simulated UAV altitude. UAV altitude is given on a logarithmic scale and increases in powers of 2. The square markers are the data points and the curved line is the best fitting cumulative Gaussian (see main text). (b) Accuracy of the classification method as a function of UAV speed with speed bins spanning an interval of 0.07 m/s. The square markers are the data points and the curved line is the best fitting cumulative Gaussian (see main text).*

Experiment	ACC	CI	GRASS	WEEDS	RUMEX	URTICA
UAV 90°	87.89	3.94×10^{-3}	90.4	85.4	74	96.8
UAV 45°	**90.39**	3.56×10^{-3}	95.4	85.4	73.1	97.6
Ground Robot	82.88	5.96×10^{-3}	87.1	78.7	86	78.3

***Table 1.** Results of applying the classification method to UAV 90° and 45° test images. The results (reported in [1]) of applying the same method to test images captured from the ground robot are also shown for comparison. ACC is the over all accuracy, i.e. the probability that a random image is classified correctly. CI is the confidence interval in the estimate of ACC. GRASS and WEED are the probabilities that images of grass, or weeds, respectively, are correctly classified. RUMEX and URTICA are the probabilities that images of Rumex, or Urtica, respectively, are correctly classified.*

4 Discussion

The results show that simple re-use of the existing state-of-the art ground-platform algorithm [1] does perform sufficiently accurately on UAV imagery to make hybrid UAV-detection and ground-spraying weed control possible and economical. The system performs well at low altitudes, up to

around 8 feet, then quickly falls off with higher altitudes. This is likely due to 8 feet being the greatest altitude at which individual blades of grass can be seen in the 4K pixel images; beyond this, regions of grass becomes blurred and indistinguishable from broader-leafed weeds. This in turn suggests that to fully check whole fields for weeds, UAVs must perform some swathing rather than work from single high-altitude images. Such swathing could be automated, requiring more effort than taking a single manual high altitude image; but would operate much more quickly and using less energy than swathing with heavy ground platforms, so is worth further investigation.

It is interesting that the accuracy on the UAV test set is actually *higher* than the state-of-the-art results of the same algorithm on ground platform images, even though the algorithm is trained entirely from mis-matched ground platform image data. The training data used from the ground platform were colour calibrated, using physical color charts mounted in camera view on the ground robots. It was not possible to mount similar physical charts to the UAV in this study due to weight requirements. The training data were taken from ground platforms at 22.5 degree down-facing cameras, while the UAV images were taken vertically or at 45 degrees. The gain in performance was thus unexpected. Possible explanations may revolve around the use of perspective dewarping assumptions in the ground platform case. There, we assumed that the ground was perfectly flat in order to enable an affine transform to de-warp perspective images to give estimates of what a true vertical view would look like. As weeds get taller, this assumption becomes less valid, and results in more distant regions becoming more distorted from the true aerial view. Although the system was fully trained on such images, including distant distorted images, it may have learned to recognise ideal true vertical views despite the noise in such training images. Then when tested on ground images, it suffers a performance penalty from distance distorted images in the test set; while when tested on real vertical UAV images it does not. For example, in the limiting case, the most extremely distorted in both ground training and ground test sets might become completely blurred and contain zero information. For ground training they will be effectively ignored, and for ground testing they will lower the accuracy rate, without lowing performance on the UAV data. Similarly, it is possible that a higher quality camera on the UAV (specifically, it is equipped with motion stabilisation while the ground platform was not) has the same effect. Or the improvement may be simply due to chance because the aerial test set is a different dataset from the ground test set. It includes some of the same physical weed plants as the ground data, but is not directly comparable, being recorded several months later under different growing and image lighting conditions. Further work is needed to test this in a more controlled setting, as in the original state of the art ground study which made very careful controls of all such factors.

For the UAV images, there are only a few images of Rumex that contain mostly Rumex (without big patches of grass). But these images still contain small patches of grass and are blurry. The images of grass and Urtica that were selected for windowing are very clear by comparison. This explains the poorer performance for Rumex compared with grass and Urtica. Future work should use a more controlled Rumex environment such as the controlled, deliberately cultivated "weed gardens" of the state of the art ground study to replace the "found data" used here. Similarly, use of controlled data would improve UAV accuracy if enough could be collected to provide new matched aerial training sets, including exact matching of perspective and camera angles between training and test sets. The height of the UAV was controlled by a human pilot in this study for technical reasons which varied somewhat and introduced size distortions to the dewarped images. Classification may be improved if automated height control, or measurement, is used to compensate. These considerations suggest that accuracy can be improved further given additional research work.

The speed of the UAV during imaging may be important if it is found to blur the images and lower the accuracy. New business-based studies should balance the business costs of longer flight times against accuracies as both altitude and speed are varied.

Acknowledgements

This work was funded in part by InnovateUK grant number 131790 IBEX2.

References

[1] Binch A, Fox CW (2017). Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland. *Computers and Electronics in Agriculture* 140: 123-138.

[2] He, Dong-Chen, Li-Wang (1990). Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing* 28.4: 509-512.

[3] Vapnik Vladimir (2013). The nature of statistical learning theory. *Springer, Berlin, Heidelberg.*

[4] Farneback G (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun J., Gustavsson T. (eds) Image Analysis. SCIA 2003. Lecture Notes in Computer Science, vol 2749. *Springer, Berlin, Heidelberg.*

AI & SOCIETY
https://doi.org/10.1007/s00146-018-0846-4

ORIGINAL ARTICLE

Legal framework for small autonomous agricultural robots

Subhajit Basu[1] · Adekemi Omotubora[2] · Matt Beeson[3] · Charles Fox[4,5]

Received: 16 November 2017 / Accepted: 24 April 2018
© The Author(s) 2018

Abstract
Legal structures may form barriers to, or enablers of, adoption of precision agriculture management with small autonomous agricultural robots. This article develops a conceptual regulatory framework for small autonomous agricultural robots, from a practical, self-contained engineering guide perspective, sufficient to get working research and commercial agricultural roboticists quickly and easily up and running within the law. The article examines the liability framework, or rather lack of it, for agricultural robotics in EU, and their transpositions to UK law, as a case study illustrating general international legal concepts and issues. It examines how the law may provide mitigating effects on the liability regime, and how contracts can be developed between agents within it to enable smooth operation. It covers other legal aspects of operation such as the use of shared communications resources and privacy in the reuse of robot-collected data. Where there are some grey areas in current law, it argues that new proposals could be developed to reform these to promote further innovation and investment in agricultural robots.

Keywords Agriculture · Robotics · Legal · Safety agribot

1 Introduction

Self-driving vehicles are rapidly arriving both on (Guizzo 2011) and off (Blackmore et al. 2004) roads. In the agricultural setting, technology has progressed from tractor driver-assistive systems such as RTK-GPS displays, to fully autonomous, self-driving platforms capable of carrying out agricultural tasks with no human intervention (Pedersen et al. 2006). While legal aspects of on-road autonomous vehicles have been well studied (Beiker 2012; Anderson et al. 2014, Pinto et al. 2012; Douma and Palodichuk 2012; Brodsky 2016), there is a need for an analogous understanding of off-road agricultural vehicles' legal positions, despite the forecast for the agri-robotics section to reach 5.7bn USD by the year 2024 (Transparency Market Research 2017). The present study reviews the relevant legal frameworks from a practical engineering implementer of agricultural robotics technologies to fill this need. It is intended as a self-contained guide for practising engineers to find all the information needed to get their autonomous agricultural robotic research systems up and running, quickly and easily within the law. As such it does not represent the formal legal advice, which should be taken in addition to the overview given here.

Autonomous agricultural vehicles have been developed in two broad classes: automated large tractors and smaller (e.g., < 1 tonne) precision robots. Automated tractor systems have been developed (Ishida et al. 1998; Michio et al. 2002; Blackmore et al. 2004; Dvorak 2016) based on existing manual-drive tractors, which already have commercially available high precision GPS guidance. These systems compute paths to swathe fields, typically in rows with headland turns. In some cases, this guidance consists of telling the human operator precisely what angle to turn the steering wheel at each second (e.g., *Trimble EZ-Guide Lightbar*, http://www.trimble.com); others show deviation from the computed path and leave the human driver to correct for it. Automated tractors typically aim to perform the same type of work as manual-drive tractors, namely bulk operations across whole fields, such as seeding, spraying and harvesting of row crops.

✉ Charles Fox
charles.fox@gmail.com

1 School of Law, University of Leeds, Leeds, UK

2 Faculty of Law, University of Lagos, Lagos, Nigeria

3 Risktec Solutions Ltd, Part of the TUV Rheinland Group, Berlin, Germany

4 School of Computer Science, University of Lincoln, Lincoln, UK

5 Ibex Automation Ltd, Sheffield, UK

Published online: 08 May 2018

Fig. 1 Example of an autonomous small robot for agriculture (agribot). This agribot weights 250 kg and precision spray weeds on hill farms. (Photo: Ibex Automation Ltd.)

In contrast, small autonomous robots for agriculture ("agribots", Fig. 1) have focused on precision applications. Large vehicles are required for bulk operations due to the need for physically transporting bulk materials such as seeds, fertiliser and produce. Small robots make up for reduced bulk transport capability by aiming, ultimately, to work on a per-plant basis. This enables them to transport smaller amounts of more targetted materials, including reduced herbicide doses to apply to individual weeds (Binch and Fox 2017), detect the fertiliser needs of and apply fertilisers to (Singh and Shaligram 2014) individual plants; and harvesting of plants (Bac et al. 2014) when they are individually optimally ready for consumption.

In some cases, precision systems are also found on large tractors, with variable rate controls uses to make bulk operations more precise (Escola et al. 2013).

The legal implications of these technologies are different from those of on-road self-driving vehicles. On-road vehicle operations take place in public places—highways—which are governed by highways legislation. In contrast, most agricultural robots are intended to operate on privately owned agricultural land, governed by different business, agricultural and environmental laws. However, such land is not free from interactions with humans, whose safety and legal positions must be considered. Farm owners, managers and workers may be present as well as walkers on public footpaths and trespassers. In the event of accidents, the roles of owners, managers, manufacturers and designers of systems must be considered. Existing legal, environmental restrictions and responsibilities must be taken into account—damage caused to the environment is a greater concern than in the on-road case, including the application of chemicals and damage to crops and soils.

1.1 Overview

In these respects, this article addresses three questions: What is the legal regime on the liability of manufacturers, suppliers and users of autonomous robots in the UK/EU? Does the law provide any mitigation of liability which could promote innovation in autonomous robots? Apart from the law in the UK/EU, what are the current debates and legal outlook on robotics and how can these shape the law in the area of small autonomous agricultural vehicles?

After brief introductions to engineering for lawyers and law for engineers, the Sect. 2 of the article examines the liability framework which applies to autonomous robots given lack of a separate or specific framework for robotics in the UK and the EU. Section 3 considers how the law may provide mitigating effects on the liability regime. These parts are intended for use by practising roboticists in need of a self-contained guide to their legal environment.

Section 4 presents the debates on grey areas in the law and proposals which may be adopted to reform the law and promote future innovation and investment in small autonomous agricultural vehicles. This part is intended both for use by policymakers and as a demarcation of potentially dangerous uncertain legal areas for practising roboticists.

The article concludes that the law could facilitate innovation in the agribots for the following reasons: The legal framework for autonomous robots cuts across different laws. Therefore, liability could be shared or distributed among different parties to a contract for the use or operation of an agribot. All legislation which imposes liabilities also provides corresponding defences which may aid the avoidance of liability or the mitigation of damages. There are in particular specific defences which address the peculiarities of technologies. Contracts can be used to define the rights and obligations of respective parties, and unless the law otherwise specifies, the contract can exclude liabilities for specific claims. Courts are legally obliged to consider the utility and social and economic value of an activity in awarding damages for loss and injury. Current debates suggest an appreciation of the new challenges posed by innovations in robotics for law and policy. These can facilitate resolution and legal intervention in the grey areas surrounding the legal framework for small autonomous agricultural vehicles.

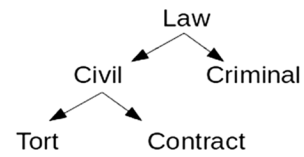
1.2 Basic self-driving technology concepts (for lawyers)

All vehicle automation systems are in practice probabilistic in their behaviour to some extent. Modern machine navigation (Thrun 2005) and object recognition systems use Bayesian probability frameworks (Bishop 2006).

Probabilities appear in these models in two distinct ways. First, the models assume precise models of the probabilistic distribution of sensory features given states of the world. By itself, this assumes that the world is random and probabilistic, though the probabilities in the equations can, in theory, be manipulated precisely and deterministically. Outdoor environments, weather, and the complexity of plant biology and animal behaviours ensure that the world is indeed random for all practical purposes—this contrasts with other robotics applications such as food processing production lines where the environment and produce can be tightly controlled and standardised (Chua et al. 2003). However, second, the Bayesian inference is known to be computationally intractable in general (Cooper 1990). This means that system designers can work only with approximation algorithms. Some of these approximations are deterministic such as Variational Bayes (Fox and Roberts 2012), while popular Monte Carlo approximations use random number generation as a seed for stochastic sampling (Andrieu et al. 2003). Stochastic methods do not have deterministic behaviour, though they converge to exact answers in the theoretical case of infinite computation time.

The bayesian theory may further make use of prior information in addition to real-time sensing (Bernardo and Smith 2001). This means that the perception of a state, and action selection based on it, can be determined not just by current inputs but also by assumptions and/or observations from the past about similar states. In the on-road case, historical data might show that other road users of particular demographics have statically predictive tendencies to behave in certain ways during interactions with the autonomous vehicle. Statistically, making use of such information as well as real-time sensors is optimal for decision-making. However, the ethics of doing so are controversial. Use of prior information is expressly prohibited in most legal systems, even though it is known to give more accurate judgements (Levitt and Laskey 2000). For off-road agricultural vehicles, such human interactions are of less concern, but similar questions about the use of priors may arise. A vehicle may behave in ways unexpected by its owner or operator if its designers have programmed it with different prior expectations that those of the owner or operator. For example, a weed spraying robot designer might assume that weeds are a priori more probable to be found near walls than in the open field, but a farmer's particular field might have all the weeds in the centre, leading to the farmer complaining about poor quality spraying decisions. Rather than made such assumptions manually, the designer may have the system learn from data. The designer can collect historical data and analysed before use of the vehicle, to inform and fix the priors. As with manual assumptions, the choice of this data is essential and will still reflect the designers' assumptions about what constitutes "typical" data. Again, if this differs from the users' assumptions, then problems may occur. In some systems, the learning from data process may continue after the sale and use of the vehicles, with algorithms updating their priors to include observations from the user's runs, including data from the present day's work right up to the present decision time. In this case, the prior information may now include a mixture of the designers directly programmed assumption, the designer's historical data, and the user's data, which has previously been identified as a legal problem (Beck 2016).

Fig. 2 Basic divisions of law



As well as use for training priors, data collected from users' farms during operation is valuable for analysis. For example, yield maps (Blackmore et al. 2003) can contain information not only financially valuable to the farmer but also to neighbouring and distant farms when used to predict trends and correlations. As with other "big data" systems such as in-car GPS route planning, which collects data on drivers' locations, questions arise about who owns this data—the owner, operator or designer?

1.3 Basic legal concepts (for engineers)

The purpose of the law is to enable all stakeholders to get along with one another in a society. This includes regulating how they should share scarce public resources, and how they should handle externalities caused to one another as side effects of their private contracts. For example, communicating with outdoor robots requires sharing of the radio spectrum with other local users, whilst applying fertiliser which runs off into a river may have negative externalities to both the general public, who use it in their water supply, and to individual neighbouring farmers.

There is no specific regulatory regime for agricultural robots or for robots generally, and indeed it may be difficult to have a single regulatory regime as robots differ on a number of criteria including functions, level of autonomy and human–machine interaction.[1] Therefore, liability could cut across different areas of the law including tort, contract and criminal laws, (as well as administrative actions) as shown in Fig. 2.

[1] For example, there are service robots, military robots, toy robots and so on.

1.3.1 Torts

A tort is a civil wrong (committed by a person called the tortfeasor) that results in loss or damage to another person, and anyone who has suffered a loss as a result of another's civil wrong can bring an action for redress. For example, the manufacturer or producer of a defective product can be held liable for the tort of negligence if the product causes personal injury or other damage to the user. Therefore, because the agribot is likely to be regarded as a product, its manufacturers/designers would be subject to laws regulating liability for defective products. Also, agricultural contractors and farmers as users or owners of the agribot, and their respective agents can be subject to different legal rules and statutory provisions governing negligence, accidents and injury to individuals as well as for loss of or damage to property.

1.3.2 Contracts

While a tort is a civil wrong entitling a party to sue the other party for a breach of duty owed under the law, a contract is an agreement which the law would enforce. The parties voluntarily agree the terms of a contract and where permitted by law, the parties may exclude or limit their liabilities for certain acts or omissions.[2] Stated differently, a contractual relationship is governed by the contract rather than by law and parties may bring a (civil) action to enforce the terms of the contract including claims for damages for breach of the contract. In civil actions such as tort and contract, the required standard of proof is the 'balance of probabilities', and the claimant must discharge the burden of proofing any loss, injury or damage. Contracts may, in some cases, be used to transfer a liability between parties.

1.3.3 Crimes

Unlike civil proceedings that are initiated by private citizens against other citizens or organizations or the government, it is the state that initiates criminal proceedings for a breach of the criminal law. A crime is a wrong against the society, and any act that constitutes a crime must be so defined by the law and punishment there for stipulated by the law. The standard of proof in criminal cases is 'proof beyond reasonable doubt', and the onus of discharging the burden lays on the prosecution or the state.[3] Punishment for crimes can range from minor fines to lengthy imprisonment. It is important to note that although there is a development towards 'corporate criminal liability' (the concept that corporations should be held liable for criminal acts of officials such as directors, managers and employees), there is no common European approach in this area and domestic laws vary. Some countries (such as Germany) do not impose criminal liability on corporations, others rely on administrative sanctions (See notes below).

1.3.4 Administrative actions

Administrative actions are concerned with activities of administrative agencies to which Authority is delegated based on the agency's expertise on the subject matter. Administrative actions, therefore, involve oversight functions through the enforcement of statutory laws (laws made by parliament) and rules made by the administrative agencies themselves. Unlike torts where individuals can bring actions for damage or injury and criminal prosecutions initiated by the state, administrative actions involve ensuring compliance through oversight which may entail levying fines and other sanctions on organizations in breach of the law. To invoke sanctions, damage or injury need not occur, all that is required is non-compliance with standards set by the law. For example, data protection authorities can impose fines for non-compliance with the principles of data protection even when data has not been lost or stolen as a result of such non-compliance. Also, environment protection agencies can impose fines for failure to report pollution or to take remedial actions.

It is important to note that although liability is discussed under the separate heads below, in practice, civil and criminal liability may arise from the same activity and administrative actions may overlap with civil and criminal sanctions (See notes on data protection below).

1.3.5 Laws, regulations, directives and standards

The potential for torts and crimes is introduced by legal acts of national and international parliaments. In the EU, Regulations and Directives are different types of legal acts of the EU. According to Article 288 of the Treaty on the Functioning of the European Union,

> A regulation shall have general application. It shall be binding in its entirety and directly applicable in all Member States.
>
> A directive shall be binding, as to the result to be achieved, upon each Member State to which it is addressed, but shall leave to the national authorities the choice of form and methods.

[2] Exclusion and limiting clauses allow parties to either limit or exclude liabilities for acts or omissions for which they would ordinarily be liable.

[3] Health and safety law is the exception to this rule; the onus is on the defendant(s) to demonstrate to the satisfaction of the courts that they have discharged their duties under health and safety law.

Thus, an EU regulation is an immediately binding law without further actions required, while directives are typically 'transposed' by member states into new local laws which implement them. In the UK, directives with relevance to Health and Safety are implemented in the form of regulations under the powers granted by the Health and Safety at Work Act 1974.

Technical standards are distinct from laws, and their use is *usually* voluntary. Standards are defined by technical committees, including the International Standards Organization (ISO), the European Committee for Standardization (EN), American National Standards Institute (ANSI), national committees such as the British Standards (BS) in the UK, local industry sector organizations, and sometimes within a single organization. Reasons for voluntary use include the ability to provide the customer with a guarantee—via contract law—of meeting a publicly known and accepted level of quality or safety; and also, the desire to make use of industry-wide technical best practices consolidated in a standard. Like directives, standards are often transposed between regions and subregions, for example a standard named with "ISO EN BS" may have begun as an international ISO standard, then transposed downward via both Europe and UK organizations; or it may have begun as a UK standard and been transposed upwards through the EU and ISO.

In some cases, the law may grant special status to a standard, giving it legal force, such as requiring all manufacturers to implement it for certain types of product. This is known as "calling up" the standard.

In the EU, compliance with Product Safety standards which are published in the Official Journal of the European Union is assumed to demonstrate compliance with relevant directives supported by the standards. Some directives will recommend the creation of such standards to be created along with their legislation, to aid compliance with that legislation. These are known as 'harmonised standards'.

Several EU directives require products to obtain a special "CE mark" standard (Conformite Europeanne) before sale. The CE mark then allows sale across the European Economic Area (EEA), showing compliance with *all* relevant directives.

The relationships between laws, directives, and standards are illustrated in Fig. 3.

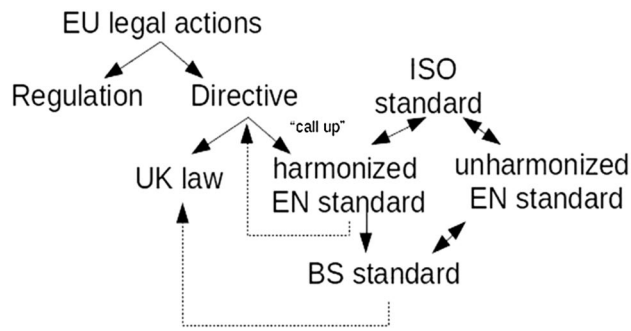


Fig. 3 Relationships between laws, directives, and standards. Creation is shown by solid arrows. Compliance is shown by dashed arrows

2 Regulating agribots: legal framework

In the light of the above, the sections that follow examine how tortious, contractual and criminal liabilities, and use of standards could arise in the manufacture, use or operation of agricultural robots.

2.1 Liability in tort

2.1.1 Product defect

EU Directive regulates liability for defective products on the approximation of Laws, Regulations and Administrative Provisions of the Member States for Defective Products.[4] [transposed in the UK as the Consumer Protection Act (CPA) 1987]. The directive applies to all types of products including agricultural products. Under the law, "product" is defined as all movables even if incorporated into another movable or an immovable (See art 2 of amendment to directive). A producer means the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part. A producer also includes any person who, by putting his trademark or other distinguishing feature on the product presents himself as its producer (art three directive).

The Directive lays down the principle of liability without fault or strict liability which means a person injured by a defective product can claim damages even if the defect was not due to the producer/manufacturers negligence. A defective product is one which does not provide the safety which a person is entitled to expect, considering, all circumstances including, the presentation of the product, such as adequacy of the warning,[5] the use to which it could reasonably be expected that the product would be put and the time when the product was put into circulation are factors (art 6). The standard of the defect is, therefore, an objective one. For example, a product is defective if its safety is not such as persons generally (everyone and not the particular claimant

[4] See Directive 85/374/EEC on the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products (as amended by Directive 1999/34/ec) The Directive is implemented in the UK by the Consumer Protection Act (CPA) 1987).

[5] See e.g. UK CPA s 3 (2)(a).

injured by the product) are entitled to expect. Also, the law does not infer defect from the fact that a better or safer product was subsequently put into circulation or permit persons to expect standards of safety that are unknown or which do not exist at the relevant time (art 6, 7 Directive, s 3 CPA).

Moreover, to succeed in an action for damages, the claimant or injured person must prove the damage and the defect in the product as well as the causal relationship between the damage and the defect (art 4). In other words, the claimant must prove that he suffered damage, that there was a defect in the product and that the defect caused the damage. Presumably, therefore, if a claimant is unable to prove defect, he cannot prove that loss or damage resulted from such defect. However, in cases where the causal link is established, the law also provides for defences which are of particular relevance to the manufacturer of the agribot. As examples, it is a defence that the producer (or manufacturer) did not put the product into circulation or that the defect did not exist at the time the product was put into circulation (art 7). These arguably cover instances where someone caused the fault after the manufacturer supplied the agribot or where interference with software causes the agribot to malfunction or where the agribot has been used for a purpose for which it was not intended (See notes on dual-use below).

Other grounds for avoiding liability include a claim that the safety fault was an inevitable result of obeying the law (e.g., the agribot could be safer but for provisions of the law which excludes the use of certain technology). Also, it is a defence that the manufacturer could not have made the product more secure or safer given the state of knowledge in science and technology ('development risk defence') (art 7(e)). Therefore, it is a defence under the law that the state of scientific or technical knowledge at the relevant time is such that the manufacturer could not have known the defect in the product. This suggests that the law does not expect manufacturers or designers to wait until a safer technology is available before introducing their products. All that is required is that the standard of safety corresponds to state of the art in scientific or technological knowledge at the relevant time. However, the Directive makes this defence optional, and it would, therefore, only avail the manufacturer where it is provided for under national law.[6]

It is important to stress that requirement for proof, and indeed the definition of a defect under the law is not intended to undermine consumer protection. Rather, it is intended to strike a reasonable balance between the obligation to protect consumers and the need to promote innovation in a fast-evolving technology environment. For example, while owing to the complexity, technicality and probabilistic behaviour of products like an agribot, it may be difficult and expensive for claimants to prove a defect, it must also be assumed that developments in artificial intelligence, robotics and machine learning would mean that safety standards become outdated fairly quickly. Therefore, unless the law limits the liability of manufacturers to safety standards based on the state of scientific and technical knowledge, their liability could be indeterminable or infinite, and this may adversely affect innovation and development.

It is also relevant to note that damage includes damage caused by death or personal injury and damage or destruction caused to property other than the defective product itself (art 9). Liability imposed by the law cannot be excluded or limited by contract and can be joint and several.[7] However, member states may provide for the limitation of liability for damage resulting from death or personal injury provided that the amount shall not be less than 70 million ECU (art 5,12).

2.1.2 Accidents and health and safety law

In the UK, health and safety law is implemented through the provision of the Health And Safety At Work etc Act (HASAWA) 1974. The Act enables the enforcement body, the Health and Safety Executive (HSE) to bring criminal prosecutions under Section 33 of the Act against organizations deemed to have breached the statutory duties it imposes.

The primary duties imposed by the act are described in Sect. 2 and Sect. 3 of the Act. The former imposes duties on employers to ensure the safety and health at work of employees; the latter on employers (and self-employed persons) to ensure the safety at work of those persons *other than their employees* who could be harmed by the employers' undertaking. An *undertaking* is defined by the set of activities carried out by an organization; this extends to the design and manufacture of products such as agribots and includes their use. Therefore, an accident whereby a member of the public is injured by an agribot could result in a criminal prosecution against the owner/user of the agribot, and/or the designer/manufacturer. The balance of this prosecution depending mainly on the nature of the accident.

Section 6 of HASAWA1974 imposes duties on the manufacturers etc. (including designers) for the safety of articles used at work. Therefore, prosecutions could hypothetically also be initiated for a breach of this Section; however, in reality, this is seldom the case.[8] Further, the duties of designers and manufacturers of agribots are better described under

[6] See arts 7 and 15(1)(b) Directive, See also Sect. 4(1)(e) of the UK CPA which infact allows this defence.

[7] Joint and several liability means the person injured by a defective product can sue multiple parties and recover full damages from one and/or all of them.

[8] HSE public register of convictions indicates that this is around thirty (30) successful prosecutions under Sect. 6 in 10 years.

the Consumer Protection Act 1987, and / or the Supply of Machinery (Safety) Regulations 2008.

Prosecutions for breach of duties under Sections 2–6 of HASAWA1974, if elevated to the Crown Court, invoke a potential maximum penalty of two (2) years imprisonment, and / or an unlimited fine. In all cases described above, the duty is qualified and limited by the term 'so far as is reasonably practicable' (SFAIRP). This is also commonly phrased as the duty to reduce risk to a level that is 'As Low as Reasonably Practicable' (ALARP). These terms are largely interchangeable, the former used in legislation, the latter commonly used in engineering communities.

The key element is the concept of reasonable practicability. This was defined in common law decades before[9] the implementation of HASAWA1974 and provides a fundamental means to both limits the duty imposed by the Act and mitigate the liability incurred following an accident and resultant prosecution. If the defendant(s) can demonstrate that all reasonably practicable measures were taken to reduce the risk, they thereby demonstrate that they fully discharged their duties under HASAWA1974.

Demonstration that all reasonably practicable measures have been taken (often termed 'demonstration of ALARP') requires the following measures be taken[10]:

1) Identification of reasonably foreseeable hazards and assessment of risk;
2) Adoption of authoritative good practice for control of risk;
3) Identification of further practicable risk reduction measures;
4) Implementation of identified risk reduction measures unless it can be demonstrated that the sacrifice (cost, time, effort) associated with doing so is grossly disproportionate to the safety benefit gained from the measure.

The above steps (2)–(4) are further predicated on the assumption that the overall risk to the safety and health of persons affected by the activity/product/system under assessment is in general, *tolerable*. If the risk is assessed as *intolerable*, then the owner of the duty to reduce that risk must do so regardless of any consideration of sacrifice. HSE guidance R2P2 provides a quantitative baseline definition of intolerable and tolerable risk.[11]

Where risks are well understood and defined by an industry body of knowledge, completion of steps (1) and (2) above will be sufficient to demonstrate ALARP. This can include compliance with legislation, approved codes of practice (ACOP) and in some cases engineering standards, where these can be shown to be directly and fully applicable and correctly applied.

Where such compliance is not possible, for example, because the technology associated with activity/product/system is new or novel, or because it is not possible to fully comply with relevant standards, further effort will need to be expended on risk assessment and/or engineering study, to determine what can be practically done to reduce the risk.

Demonstration of *gross disproportion* relies upon the assessment of the benefit of the risk reduction measure and consideration of the sacrifice (e.g., financial cost) of implementation of the measure. The concept of gross disproportion ensures that this is not a straightforward cost–benefit analysis, whereby the owner could demur if the sacrifice simply exceeds the benefit; rather the sacrifice must *grossly* exceed the benefit before the duty to implement the measure is discharged.

The above assessment can often be carried out qualitatively, for example, through use of a continuous matrix (such as the Boston Square), placing the effectiveness of a risk reduction measure on one axis, and difficulty involved in implementing the measure on the other axis. Potential improvement measures are then ranked relatively against each other. There are also a number of simplified screening tools in general use that highlight qualitatively those measures that should be implemented, should not be implemented, and those which require further study. In all cases, these qualitative methods will need to take account of the requirement to demonstrate gross disproportionality between the sacrifice and the safety benefit.

Where sufficient information is available, and where the resolution of the cost/benefit decision is less clear (for example, an initial screening tool results in the requirement for further study), a full quantitative assessment can be undertaken. This requires the quantification of the full lifecycle risk without further mitigation (sometimes termed the *vanilla* risk), for example, in terms of Potential Loss of Life (PLL) or Fatalities and Weighted Injuries Rate (FWI); similar quantification of the risk reduction measure(s); and combination of these values with a Value for Preventing a Fatality (VPF).[12] The sacrifice associated with implementing these measures is then calculated, and the measure implemented unless the sacrifice is found to be grossly disproportionate to the safety benefit.

[9] Judgement of Lord Asquith in Edwards vs National Coal Board 1949.

[10] Health and Safety Executive. Reducing Risks Protecting People (R2P2).

[11] 1×10^{-3} fatalities per annum for workers, 1×10^{-4} fatalities per annum for members of the public.

[12] R2P2 provides a value of £1,000,000 in 2001, however, this value, when subject to a reasonable allowance for inflation, should be considered a minimum value. Various higher values have been applied in different industries.

Definitions of gross disproportion vary dependent on context; however, a useful rule of thumb is to consider the initial level of risk. Where that initial risk is tolerable but high, i.e., close to the border with the intolerable region, the gross disproportion factor should be similarly high. Where the risk is tolerable but low, the gross disproportion factor may also be lower. In some industries, in some circumstances, a sacrifice that is 3 × the benefit may be considered grossly disproportionate; whereas in other cases, a factor of 10 × may be required before a measure should be considered not reasonably practicable to implement.

The Management of Health and Safety at Work Regulations 1999 impose a duty on employers to undertake a suitable and sufficient risk assessment in support of the duties placed upon them by Sections 2 and 3 of HASAWA1974. However, even were this not the case, a demonstration that risk has been reduced ALARP is challenging to achieve without carrying out such an assessment. In fact, the requirement for risk assessment has arguably been part of UK common law since 1949.[13]

The requirement for risk assessment, should not be confused with a requirement for risk *analysis*. For a risk assessment to be suitable and sufficient, it must demonstrate that appropriate action has been taken to reduce the risk. Where sufficient information is available, a detailed analysis in support of this action may be beneficial. However, this is often not required, and sometimes not justifiable. For example, where there are high levels of uncertainty associated with a particular hazard, which render conventional risk assessment techniques unreliable, a *precautionary principle*[14] should be adopted. This principle requires that the assessment and action were taken to be based more on the putative consequences of a risk, rather than the likelihood.

In the case of agribot use/design/manufacture, where authoritative good practice is still primarily to be defined, compliance with health and safety law will depend on the suitability and sufficiency of the risk assessments carried out by duty holders. Further, whereas the balance of prosecutions in the UK as a whole tends to focus more on immediate causation[15] (i.e., who are the persons/organizations who 'last touched the risk'), the nature of the autonomous robots may largely necessitate a greater focus on the prosecutions of designers and manufacturers. They may be more frequently called upon to present formal safety justifications of their autonomous products that demonstrate anterior identification, consideration and management of relevant hazards and risks. Complete justification will necessarily include the documentation of critical design decisions, the identified practicable risk reduction measures, and reasonable justifications for the measures rejected, as well as those, adopted.

For users/owners to discharge their safety and health duties, they may be largely dependent on the decisions are taken autonomously by the agribots. In corollary, the extent to which they can be held liable for those autonomous decisions is limited by the extent to which they can train/teach the agribot before full operations; this is in turn limited by the safeguards and risk reduction measures defined by the designer as a result of their risk assessment. As with all risk reduction measures, a hierarchy of control[16] should be adopted by designers.

Elimination of hazards during the early phases of design should be prioritized; where hazards cannot be eliminated they should be controlled primarily be engineering means, for example, safety functions[17] that bring the agribot into a safe state upon detection of a failure or the presence of a member of the public in close proximity. Lower levels of this hierarchy will necessarily include the provision of instructions for use, informed by the suitable and sufficient designer risk assessment. In effect, the users will be responsible for management of the *residual risk* associated with the agribot, i.e., those risks which could not be designed and engineered away.

Notwithstanding the above, there is guidance available that will be partly applicable to the use of agribots and may assist users of agribots with the implementation of safe systems of work. This will necessarily include appropriate traffic management arrangements[18], including measures to ensure exclusion of the public, route planning, lighting and visibility, where necessary, as guided by

[13] Edwards vs National Coal Board 1949; 'Moreover this computation falls to be made by the owner at a point of time anterior to the accident.'

[14] R2P2 Reducing Risks Protecting People.

[15] Review of prosecutions for 2016/17 under the CDM Regulations 2015 describes a total of seven (7) potential breaches of duties for Principal Designer / Designer duties, whereas a total of ninety-nine (99) potential breaches of client duties, four-hundred and eighty-nine (489) potential breaches of Principal Contactor duties, and two-hundred and seventy-eight (278) potential breaches of Contractor duties were identified.

[16] Several different hierarchies are available, for example, the commonly used ERIC PD (Elimination, Reduction, Isolation, Control, Procedures, Discipline) and the hierarchy provided in the Provision and Use of Work Equipment Regulations (PUWER) 1999, where fixed guards shall be provided to prevent exposure to dangerous parts of machinery, wherever practicable, and where not so, the provision of other guards or protection devices. Information, instruction, training and supervision are in all cases the lowest level of the hierarchy for the control of identified risks.

[17] Safety functions designed in accordance with BS EN IEC 61,508 and BS EN IEC 62,061.

[18] For example, INDG199 HSE leaflet on Workplace Transport Safety and HSG136 HSE guidance on Workplace Transport Safety, both of which are freely available electronically from the HSE website.

manufacturer-provided instructions for use in combination with suitable and sufficient user risk assessment.

Health and Safety law in the UK is primarily goal-setting and requires a regime of self-regulation to ensure compliance with the HASAWA1974, particularly Sections 2 and 3. Therefore, the measures, guidance, and techniques outlined above are applicable, regardless of whether any specific, prescriptive regulation exists. In all cases, applicable good practice should be sought, and the duty owner(s) should determine appropriate measures to reduce the risk to a demonstrably ALARP level using an appropriate hierarchy of risk control measures.

For example, in the event that an agribot may be used in low-visibility environments, such as mist/fog, or nighttime working, the designers would need to consider the measures that could be designed into the system to reduce the risk. For example, a designer could not demonstrate that risk had been reduced ALARP by recommending in the instructions for use that the agribot wear hi-visibility clothing, regardless of how humanoid in appearance the agribot may be! Firstly, this is because Personal Protective Equipment (e.g., hi-vis jackets) always forms the lower ranks of any hierarchy of risk control measures; correct use of PPE is always subject to human error or violation. Secondly, hi-visibility clothing is used primarily to protect the wearer, whereas in this scenario, persons most at risk would likely be those driving other vehicles that could potentially impact the agribot. It should be clear to a designer that, even in the event of a hypothetical stipulation in the instructions for use that the agribot should not be used in periods of low visibility or nighttime; use in such conditions would certainly constitute reasonably foreseeable misuse. As such, the designer has an obligation to ensure that the agribot is provided with reasonably practicable measures to increase visibility (e.g., lights) and / or other measures to avoid collision (e.g., horns / audible warnings). For example, practicable measures could include (but not limited to): collision detection systems based on radar scanning and autonomous avoidance; built-in lighting systems, potentially with safety systems that prevent operation in low-visibility environments when lighting systems are non-functional; hi-visibility paintwork; reflective strips, reflectors. A combination of these elements would likely be necessary to demonstrate that risk is reduced ALARP, subject to assessment as described in the paragraphs above.

A further example, is the use of agribots on public highways. From the above discussion and example, it should be clear that no agribot should be used on public highways unless reasonably practicable risk reduction measures are implemented. Inherent in the definition of reasonable practicability is the concept of proportionality; measures taken to reduce the risk should be proportional to the risk. Therefore, in the event that an agribot is required to autonomously travel on or across public roads then collision avoidance safety systems must be designed-in, similar in extent to those required for autonomous road vehicles. However, in the event that an agribot can be supervised across a road crossing in manual mode or remote mode the exposure to risk is lower, and it is reasonable for the designed-in safeguards to be less onerous (of course providing that suitable controls are designed-in to prevent inadvertent agribot access to public roads).

For the scenario of an agribot crossing a road in a supervised/manual/remote mode, the extent to which risk reduction measures can be designed-in would be firstly dependent on the extent to which risk reduction measures are practicable, i.e., technically feasible. For example, crashworthiness/impact absorption, to prevent damage in passenger carrying vehicles, and/or collision avoidance systems that effectively distinguish between vehicle hazards, users, members of the public, and livestock (which may be crossing simultaneously with the agribot). Secondly, the designers would need to be assured that they are not introducing additional hazards that are potentially higher risk than the hazard they are trying to control. For example, designer risk assessment may determine that any collision detection system should be deactivated, while in manual or remote mode to avoid risks to the local user—such as autonomous avoidance resulting in the robot reversing into a manual remote controller walking closely behind it—or risks increased by non-execution or delays to command responses. In this case, the system would not be effective for mitigating risk of vehicle impact when crossing roads. In such a scenario, complete with supporting risk assessment, it may be that the designer is able to reasonably discharge their responsibility for further reduction of risk. This is providing that: suitable arrangements are provided in design for agribot visibility as discussed above; the manual/remote mode is generally and demonstrably safe and reliable; a Safe System of Work can be adopted by the user that follows the highway code, providing suitable warning to other road users that a crossing is taking place, and controlling/excluding traffic, where necessary.

2.1.3 Accidents and negligence

Legal action in tort for negligence may also be taken against manufacturers, agricultural contractors, operators and farmers and their agents for injuries, loss or damages resulting from negligence or accidents involving the agribot. However, unlike strict liability or liability without fault, a claim in negligence requires the claimant to prove fault on the part of the manufacturer or other person being sued. The following must be established; the defendant(s) (such as manufacturer/designer, contractor or farm owner) owes a duty of care to the claimant, there was a breach of that duty (the defendant failed to take care), the claimant was harmed (that is personal injury, or damage or loss of property resulted).

Liability for negligence may fall on any of the parties depending on the cause of the accident, and who owes or is owed a duty of care in the circumstances of each case. For example, the position of the law is that the manufacturers owe a duty of care to persons who use their products and manufacturers would be deemed to have breached this duty where there is a defect in the product. A cause of action (the basis for suing the manufacturer) arises where injury or loss results from the defect. For liability, it is immaterial that the claimants did not purchase the product themselves. Therefore, suppliers, farmers, contractors and their agents or other users who may be injured by any defect in the agribot would be entitled to sue the manufacturer for negligence. From the perspective of the consumer, an action in negligence provides additional protection as product defect may raise a *prima facie* case of negligence.[19]

Apart from defects, liability for negligence may arise in cases of misuse mainly where manufacturers fail to provide instructions or where the instructions are inadequate or misunderstood. Under the EU Machine Directive,[20] [transposed in the UK as the Supply of Machinery (Safety) Regulations 2008] the manufacturer or his authorized representative is required to provide necessary information such as instructions before putting machinery on the market and/or putting it into service (art 5 Machine Directive). Regarding the general principles for drafting instructions, the Directive provides that instructions must be drafted in one or more official Community languages (of the EU), and the case of machinery intended for use by non-professional operators, the wording and layout of the instructions for use must take into account the level of general education and acumen that can reasonably be expected from such operators (Machinery Directive item 1.7 annex 1).

It is, therefore, a question of fact depending on the circumstances of a case whether warning or instruction is sufficient and whether the manufacturer is liable or not. For example, instructions and warnings full of probabilities and equations provided to intermediaries (such as agricultural contractors) may be sufficient if the contractor is learned in and has a good understanding of the agribot. Conversely, the same instruction addressed to farmers who (presumably) have the less technical knowledge, may need to be more basic. Therefore, in a hypothetical scenario where a farmer misunderstands the instructions and assumes the agribot is safer than it actually is and thereby causes the agribot to malfunction and kill a walker, a brochure full of probabilities may be deemed too complicated, and the manufacturer may be held liable for accident caused by the farmer's misuse. The key principle is, therefore, that instructions must be pitched at the level at which both technical other non-technical users of the agribot can understand them.

Other provisions of the Machinery Directive particularly relevant to the agribot include the requirement that the contents of the instructions must cover both intended use of the machinery and any reasonably foreseeable misuse. Also, where applicable, the instruction manual must contain warnings concerning ways in which the machinery must not be used that experience has shown might occur (item 1.7.4 annex 1 to the Machinery Directive). These provisions suggest the manufacturers would still be deemed to have complied with the law if they fail to give warnings on use and misuse which were not known at the time of manufacture or design but subsequently becomes known due to self-learning, the processing artificial intelligence (AI) or repurposing of the robot. They also suggest that apart from the manufacturer, other users of the agribot could be liable if they ignore clear instructions and warnings or continue to use the agribot after discovering that it has malfunctioned due to failure to follow instructions. However, to benefit from the presumption of conformity with the health and safety requirements under the Directive, manufacturers are required to affix CE marking on their product and comply with a declaration of conformity (arts 5,7).

2.1.4 Accidents caused by agents, employees and contractors

Liability for accidents caused by third parties depends on whether the person who caused the accident is an agent or an independent contractor. Under the law, a principal is vicariously liable for the acts and omissions of his agent when the agent is acting within the scope of his authority. The scope of an agent's authority is defined by a contract between the agent and the principal. As an example, therefore, liability for acts or omissions of the operator of the agribot will depend on whether he is an agent of the manufacturer or the agricultural contractor or whether he is an independent contractor. Similarly, if there is a franchise agreement, the franchisor's liability will depend on whether the franchiSee acts in the capacity of an agent. Therefore, while the law does not automatically infer an agency relationship from a franchise, the agency can be inferred from the contract and the circumstances of the case.

As also noted above, liability might depend on whether third parties, such as employees, agents or contractors, receive adequate instructions on the use of the product. As an example, under the Provision and Use of Work Equipment Regulations (PUWER) 1998 (UK), businesses which either use or hire out work equipment are required to manage the risks from the equipment. Risk management includes

[19] This means fact of defect is sufficient to raise a presumption of negligence unless it is disproved.

[20] Directive 2006/42/EC came into effect on 29 December 2009 and replaced Directive 98/37/EC.

ensuring that all people who use or manage work equipment receive adequate instructions and appropriate training. Therefore, apart from the manufacturer, operators of the agribot, agricultural contractors and farmers are also legally obliged to assume liability for accidents caused by third parties due to misuse.

Furthermore, under the Occupiers Liability Act 1957 and 1984, an occupier, that is a person in control of land, premises or buildings can be held liable for injury or harm to another person on the land. Such persons can include workmen, residents, visitors, strangers or even trespassers. One of the conditions for the assumption of liability is that the harm is caused by a person over whom the occupier has control or over which he could exercise some degree of control. It is, however, important to note that this liability can be excluded by contract.

Finally, damage caused by the escape of things likely to cause mischief is borne by the owner of the land provided the damage be NIL a reasonably foreseeable consequence of the escape (this is the rule in Rylands v Fletcher).[21] In practice, this might mean a farm owner or farm manager could be liable if he (or his agent or anyone under his control) allows the agribot or things used by the agribot such as herbicides to 'escape' to adjourning lands or farms and for damages resulting from such escape. This position poses a little problem when the agribot is operated in manual mode as the operator is deemed to be in control. However, when operating autonomously, the risk of 'escape' may heighten, and farmers or other users of the agribot may have to adopt additional measures to avoid liability. This may include closing escape routes and putting warning signs at different ends of a road when the agribot is in operation. Although this is not a legal requirement, in the UK, farmers routinely close local roads to move herds of animals by placing signs and/or people at both ends before releasing the animals. The Health and Safety Executive (HSE) has also issued a number of advice on public access for livestock which would be relevant to the operations of the agribot.[22] It is, however, important to note that the Animals Act 1971(UK) impose strict liability on keepers of animals which are of a dangerous species.[23]

The outstanding challenge from the above liability allocation regimes relates to how to resolve the attribution problem. For example, despite the clear provisions of the law, it might be difficult to ascertain whether damage, injury or loss was caused by a defect in the product or misuse such as failure to follow instructions. It is conceivable for instance that contractors or farmers would tend to attribute loss or damage to product defect rather their misuse of the agribot. It is also conceivable considering the complex and technical nature of the agribot and the fact that law imposes liability on the manufacturer for insufficient and unclear instructions, that courts might be more inclined to hold manufacturers liable in negligence rather than hold users liable for misuse. One solution to the possible dilemma is to design the robot with detailed data logging system. This would create a form of 'liability by design' which enables the agribot to keep detailed logs of events and incidents including possibly replaying an accident to establish if was caused by a sensor failure or user command. A data logging system may, therefore, assist in identifying where liability falls where there is a dispute as to whether accidents are due to manufacturer defect or user misuse.

2.2 Administrative actions

2.2.1 Regulation of environmental damage and use of chemicals in general

The application of chemicals which may affect the environment is tightly controlled. This includes the robotic application of fertiliser and pesticide chemicals, and their potential effects on the human food chain, water supply, neighbouring farms, farm staff, and the more comprehensive public environment.

Liability for damage to the environment by activities of businesses is regulated by Directive 2004/35/EC of the EU Council on Environmental Liability regarding Prevention and Remedying of Environmental Damage [transposed in the UK as Environmental Damage (Prevention and Remediation) (England) Regulations 2015]. The Directive adopts an administrative approach. It does not, therefore, apply to cases of personal injury, damage to property or economic loss and does not affect any right regarding these types of damages (recital 14).

The relevant provisions of the law impose strict liability (based on a 'polluter pays principle') for pollution of the environment caused by certain activities including the manufacture, use, storage, processing, filling, release into the environment and onsite transport of plant protection product. Plant protection products include products for destroying undesired plants and damage includes damage to water and soil. Although liability is strict, a causal link between the activity and the damage must be proved, and the law allows cost allocation in cases of multiple party causation especially concerning the apportionment of liability between the producer and the user of a product (art 9). Where environmental damage has occurred, the operator is required to inform the competent authority and take practical remedial

[21] [1868] UKHL 1.

[22] See e.g. HSE, 'Cattle and Public Access in England and Wales: Advice for Farmers, Landowners and Livestock Keepers' http://www.hse.gov.uk/pubns/ais17ew.pdf accessed 03/05/2017.

[23] Animals Act 1971, s 1.

actions to remove or otherwise manage the contaminant (art 6). The operator bears the costs for the preventive and remedial actions taken under the Directive (art 8).

Effects on water supplies are controlled by Directive 2000/60/EC (Water Framework); Directive 2008/105/EC; and Council Directive 98/83/EC (Drinking Water Directive), which set limits on levels of chemicals which may enter public water systems. Restrictions on the classification, labelling, and packaging of substances and mixtures are defined in European Regulation (EC) No 1272/2008.

2.2.2 Use of fertilisers

Apart from general chemical laws, fertilisers are covered by addition laws.

The EU Nitrate Directive 91/676/EC aims to protect water quality across Europe by preventing nitrates from agricultural sources polluting ground and surface waters and by promoting the use of good farming practices. The 2003 European Fertilisers Directive covers Sale, manufacture and labelling of fertilisers. The Directive will apply to the sale of fertilisers, which may include the sale of fertilisers included as part of a robotic package.

Ammonium nitrate fertiliser may be used as an ingredient of explosives, so falls under anti-terrorism laws which control its storage security. In the UK these include Control of Major Accident Hazards Regulations (COMAH); Dangerous Substances (Notification and Marking of Sites) Regulations 1990; Ammonium Nitrate Materials (High Nitrogen Content) Regulations; and Planning (Hazardous Substances) Regulations.

The EU Single Payment Scheme subsidises farms but in return imposes environmental protection requirements on them which may include limits of fertiliser levels. Further, Nitrate Vulnerable Zones (NVZs) are areas designated as being at risk from agricultural nitrate pollution (e.g., including about 60% of land in England.) There are additional legal limits on amounts and times of year for fertilisers which can be used in them, imposed by the Nitrates Directive and Drinking Water Directive.

2.2.3 Use of herbicides, pesticides and biocides

In addition to general chemical laws, pesticides—and more generally, "biocides"—are covered by additional laws. A "herbicide" is a chemical which kills one or more plant types; a "pesticide" is a chemical which kills "pests" including weeds, fungi and insects; a "biocide" is a chemical which harms any animals, humans or the environment.

The 2009/128/EC Directive on Sustainable Use of Pesticides [implemented in the UK as "PA Certificates of Competence" via the transposed Plant Protection Products (Sustainable Use) Regulations 2012] aims to protect surface water and drinking water from pesticide contamination. Also, pesticide use is to be reduced in areas used by the general public and in nature conservation areas. It aims to reduce the risks and impacts of pesticide use on human health and the environment and promote the use of integrated pest management and alternative approaches, such as non-chemical ones. The directive requires operator training for different pesticide types and applicator types.[24] It also bans aerial spraying in all forms, including by autonomous drones and manual piloted helicopters. In practice, this aerial ban has proved to be problematic for several weed types, including needle blight in trees, and bracken in moorland. However, the directive also allows member states to grant exemptions, on the application, to users for specific nationally approved plan types such as these, which are usually administered by their environmental agencies (For example, the UK currently has around three such approved plans, used under permits issued to tens or hundreds of individuals).

The EU Biocides Regulation 528/2012 regulates *all* substances harmful to humans, animals and/or the environment, i.e., biocides, requiring authorisation for their use. Bulk authorisation is provided to users of "on-label" products, where the substance manufacturer has handled safety testing and defines the appropriate dose size and use-case for application, on a product "label", [25] and the user operates within these parameters. When using "on the label", liability for damages caused by the product is transferred from the user to the manufacturer. If a user chooses to use the product at a different dose or for a different use-case, this is "off-label" usage, and the user retains the liability. To comply with the Biocides Regulation, the user must thus obtain their off-label authorisation, e.g., via an application for a permit from their national Environment Agency.

The certification system for human operators appears to pose little problem to the robotic application where the agribot is legally considered as a tool of a named human operator and uses an existing applicator type, such as a knapsack or bulk sprayer system. In this case, that human operator must hold the required certifications for the herbicide and applicator type. Definitions of applicator type may become problematic for robots using novel applicators, such as prototype per-plant precision devices. If the robot operates otherwise than as a tool for example under a framework which recognises the legal personhood of autonomous robots (See Legal Personhood section below), then the definition of certification again becomes problematic.

[24] Until 2015, a "grandfathering" scheme allows existing operators to practice without certification, this is no longer the case.

[25] Usually a long and highly detailed legal document, not a physical label on a chemical container.

As with operator certification, definitions of on-label application use-cases are likely to function for agribots spraying using similar technology to a manual knapsack or tractor-mounted devices, under legal operation as human tools; but the use of novel applicator methods or non-tool operation are likely to be problematic or at least require custom national environment agency licensing.

For manufacturers and sellers of herbicides, additional rules are provided in the Machine Directive amendment 2009/127/EC on Herbicide Application (transposed in the UK as EC Fertiliser (England and Wales) Regulations 2006; See also the UK Fertilisers Regulations 1990/1991 UK). As with fertilisers, these may apply to agribot operators selling herbicide as part of a robot product or service package.

2.2.4 Radio communications—scarce spectra

An often-overlooked aspect of agricultural robotics systems is the need for long-range communications links from the robot in the field to a base station, which in some cases form systems as or more complicated than the robots themselves. Such communications links, as illustrated in Fig. 4, are required if the robot is operating as a tool rather than as a legal person—so that the named human operator can monitor its condition sufficiently to intervene in emergencies and to take responsibility for its actions. In practice, this will often require a video link to monitor the robot's cameras in real time. The video is a bandwidth-hungry medium which often requires specialist communications links and equipment. Radio bandwidth is a limited and valuable[26] resource which must be shared with other local users, so is tightly regulated in most countries. Hence the legal need for the human operator to take responsibility for the robot's actions must be balanced against the need for legally restricted spectrum resources. In the EU, the restriction is performed by the Radio Equipment Directive 2014/53/EU, and currently in the UK by the Communications Act 2003.

Most current radio communications operate on single, or small groups of, identifiable frequencies. Radio communications have been used from 3 Hz to 300 GHz, with bands around higher frequencies carrying more bandwidth than those at lower frequencies, but lower frequencies propagating over long distances more efficiently. Two users transmitting on the same frequency in the same area will interfere with each other's signals. Countries' laws initially assign the rights to transmit on all frequencies to a government body called the regulator (In the UK this is Office of Communications OFCOM; in the USA, the Federal Communications Commission, FCC). The regulator is then responsible for managing allocations of these frequencies in local areas to users.

International standards exist, via the International Telecommunications Union (ITU), designating certain frequency bands for particular types of use, including for national broadcasting, cell phone data, emergency services and military communications, and amateur ('ham') radio. The same standards assign further bands for licensed commercial use, and others for unlicensed public use within the defined power and use-type limitations. This allows products to operate in the same bands between countries. The regulator typically implements these standards via its licensing to users.

Public channels. Domestic 'WiFi' (802.11) radio is often used for research agricultural robot communications, requiring no special permission from the regulator. In the UK, OFCOM allows transmission of data on several frequencies around 2.4 GHz for this purpose but limits transmitter power to 100 mW, which typically can stream video up to around 100 m ranges. Many domestic (e.g., up to 250 mW) and other devices (e.g., many watts) are technically able to transmit at higher powers (achieving longer ranges), but this would violate the OFCOM regulation.[27] Specialist antennas can concentrate the beam transmitted in specified directions to enable long-range point-to-point communications. However, OFCOM power regulations apply to the power level receivable at any location rather than to the source transmission power. This means that no legal range extension benefit is obtainable through their use—a 100 mW source concentrated along a beam to a destination may have the same, illegal, received power as a 1W omni-directional source. Across the EU, a public 433 MHz band may also be used for low power, short-range communications, suitable for sending control commands and occasional sensor data, but not live video. Across the world, some bands are allocated for public amateur ("ham") radio, by the regulator transferring use to hobbyist organizations, who then allow their certified members to use them, under restrictions such as preventing purely private use (such as encrypted or closed protocol data).

Where public radio channels are insufficient, the regulator may lease other dedicated bands in local areas for exclusive use by specified users, usually for a significant fee. Wider bands cost more but allow higher data rates (In the UK, OFCOM's main schemes are called "technically assigned" and "area defined" licences).

[26] The high values have been most visibly demonstrated in many countries' recent auctions of spectra to mobile phone companies.

[27] From a safety perspective, it should also be noted that 2.4 GHz is a microwave frequency, similar to those used in microwave ovens, which operate at hundreds of watts for cooking. Multi-watt wifi transmission may be harmful to human tissue as well as illegal.

2.2.5 Privacy and data protection

Perhaps one of the most significant aspects of the liability regime for the agribot is the privacy and data protection implications of the information collected during its operations. Data collection agribots tasks may include monitoring of soil and plant conditions, as well as building up maps of farms for general navigation. This data can be commercially valuable not just to the landowner but to others who might have financial interests in the land (such as deciding whether to buy it) or in collating data from millions of farms to perform the large-scale analysis. Many small farms are owned and operated as Sole Traderships rather than as limited company structures, linking their data directly to a named individual, sole trader and thus making it "personal" data.

According to the European Parliament on Legal Affairs, for example, AI and robotics can potentially generate large amounts of personal data that can be used as currency to purchase services.[28] The relevant law is the EU General Data Protection Regulation (GDPR) 2016 which repealed Directive 96/46/EC (data protection directive). The Regulation entered into force on 24 May 2016 and will apply from 25 May 2018.[29] The Regulation applies to the processing of "personal data" defined as information relating to an identified or natural person (data subject). While much of the provisions centre on bridging perceived gaps in the law given developments in information technology, provisions relating to principles of data processing, privacy by design and automated decision-making is particularly relevant to agribots.

(a) Principles of data protection

The principles relating to the processing of personal data as follows;

1. Lawfulness of processing—The Regulations provide that processing of personal data must be Fair, lawful and transparent. Consent of the data subject is one of the conditions for lawful processing (art 6). Moreover, where processing is based on consent, 'the controller'[30] shall be able to demonstrate that the data subject has consented to the processing of his or her personal data and the data subject shall have the right to withdraw his or her consent at any time (art 7).

It is important to note that although consent is not the only mechanism for justifying the processing of personal data, it remains a core principle of data processing. Therefore, where consent is the basis of processing, it must be clear and unambiguous as the consent of the data subject cannot be inferred from conduct or inaction. In the case of the agribot, there may be instances where it is unclear whether the data is personal or who owns the data for consent. For example, Company C operates the robot on farmer X's land and collects detailed soil nutrient information during the run. C then operates on neighbouring (and competing) farmer Y's land and makes use of X's data to optimise the run on Y's land with the result that Y ends up with better-informed run than X. Farmer Y might also be interested in buying land from farmer X and could obtain private information about its condition and value from the data. The collection is without X's consent. The question may arise whether Company C owes any obligation to X concerning the collection and use of the detailed soil information. On the one hand, because detailed soil information relates to the soil condition and not to the individual, it cannot constitute personal data. On the other hand, because the collection may invariably involve the collection of geolocation data, (which is deemed personal data), Company C may require consent from X. It, therefore, seems reasonable to obtain consent to any collection of personal data where it would be difficult to isolate personal data from the information collected.

2. Purpose limitation—a collection of personal data must be for specified, explicit and legitimate purposes and further processing in a manner that is incompatible with those purposes is prohibited.
3. Data minimisation—personal data must be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.
4. Accuracy—personal data must be accurate and where necessary, kept up to date, and every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay.
5. Storage limitation—personal data must be kept in the form which permits identification of the data subject for no longer than is necessary for the purposes for which the personal data are processed. However, data may be stored for longer periods if it will be processed solely for archiving in the public interest, scientific or historical research purposes or statistical purposes.

[28] European Parliament, (2014–2019) Committee on Legal Affairs, 'Draft Report with Recommendations to the Commission on Civil Law Rules on Robotics' 2015/2103 (INL), p 8 (hereinafter Committee on Legal Affairs Draft Rules on Robotics).

[29] See Regulation EU 2016/679 on the protection of Natural persons with regard to the processing of personal data and the free movement of such data, and repealing Directive 95/46/EC. The GDPR replaces Directive 96/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[30] The controller is a natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data. See GDPR art 4(7).

6. Integrity and confidentiality—using appropriate technical or organizational measures, personal data must be processed in a manner that ensures appropriate security, including unauthorised or unlawful processing and against accidental loss, destruction or damage.[31]
7. Restrictions on transfer—although, transfer of personal data outside the EU does not require any specific authorisation, transfer to third countries or international organizations may take place where the (EU) Commission has decided that such third country or organization ensure an adequate level of protection for personal data.[32]
8. Accountability—The controller shall be responsible for and be able to demonstrate compliance with the above principles.

(b) Privacy by design and restrictions on automated decision-making

Article 25 of the Regulations mandate the implementation of privacy by design or privacy by default (PbD). The specific provisions of the law are that data controllers shall implement appropriate and technical measures for ensuring that by default, only personal data which are necessary for each specific purpose of processing are processed. The obligation to implement privacy by default applies to the amount of data collected, the extent of their processing, the period of their storage and their accessibility. In particular, the measures shall ensure that by default, personal data is not accessible without individual intervention to an indefinite number of natural persons (art 23).

The Regulations recommends pseudonymisation as an appropriate technical and organizational measure which meets the requirements of the regulations and protects the rights of data subjects. However, in implementing appropriate technical and organizational measures mandated by the law, the controller shall take account of the state of the art, the cost of implementation, and the nature, scope, context and purpose as well as the risks and likelihood and severity posed by personal data processing to the rights of natural persons. Under article 22, the data subject has the right not to be subject to a decision based solely on automated processing, including profiling. The data also subject the right to be informed of the existence of automated decision-making including profiling and a right to an explanation for the logic underlying such decisions as well as the significance and consequences of the processing (so-called right to an explanation).[33]

Although the provisions above may pose some difficulties for the development and functioning of agribots, (See notes on scope and application of data protection law below) they are based on the (arguably correct) presumption that the problem with automated decision-making is not so much the inability of humans to predict the behaviour of autonomous robots. The problem is the need for trust that the decision-making process is transparent for accountability, reliability and trust. As a result, the algorithms that underpin agribot systems need to be as transparent and as interpretable as possible, and the agribots must be able to explain their behaviour in terms that humans can understand right from how they interpreted their output to why they recommend a particular output (so-called explanation-based collateral systems).[34]

(c) Data breach reporting and administrative fines and penalties

The Regulation also makes provisions for mandatory data breach notification and empowers supervisory authorities (national public authorities such as the Information Commissioner's Office in the UK that would monitor and enforce the Regulation) to impose administrative fines which could be potentially large (a maximum of 20 million Euros or 4% of the global annual turnover of the preceding financial year whichever is higher) for infringements of certain provisions of the law.[35] However, while supervisory bodies have the power to levy fines and other sanctions, this does not preclude individuals from bringing civil actions. In *Vidal-Hall v Google Inc*.[36] for example, the court ruled that misuse of personal information is an actionable tort. Agribot operators may thus need to invest in specialised secure data storage facilities, and consider the use of cryptography to protect data stored on and communicated by agribots in the field to comply.

2.3 Liability under contract

Liability can arise under a contract between different parties concerning the use and operation of agricultural robots. Contractual agreements are particularly important, because contracts define the rights, obligations and liabilities of parties and the courts will enforce the terms of a contract voluntarily entered into. Therefore, where permitted by law,

[31] See generally GDPR art 5.

[32] general principles on transfer are contained in GDPR arts 44–50.

[33] See also GDPR art13(f).

[34] European Parliament committee on Legal Affairs, 'Artificial Intelligence: Potential Benefits and ethical Considerations' p 4 http://www.europarl.europa.eu/RegData/etudes/BRIE/2016/571380/IPOL_BRI%282016%29571380_EN.pdf accessed 13/03/2017.

[35] See GDRP arts 33, 51, 58, 83(4) & (5).

[36] (2014) EWHC 13 (QB).

parties may by contract exclude or their limit liabilities. For example, parties may by contract agree that it is the duty of the agricultural contractor to provide training to the farmer on the use of the agribots. They may further agree that such training excludes or limit the manufacturer's liability for accidents caused by misuse. The parties may also contract to contribute towards damages for loss of reputation which is likely to affect the manufacturer's brand. It is also important to note that under rules of privity of contract, only parties to the contract can acquire rights or liabilities under the contract. Therefore, in a contract between the farmer and an agricultural contractor for the supply of agribots to be used for killing weeds, the farmer can only sue the agricultural contractor if the agribot was incapable of killing weeds. He cannot sue the manufacturer unless the manufacturer is also a party to the contract.[37]

Finally, the scope of remedies under a contract is wide, and a party can seek some reliefs for damages caused by a breach by the other party or parties. Therefore, an innocent party may ask to be discharged from further obligations to the party in breach, or claim damages for loss suffered. In cases of disputes or claims for breach of contract, courts will usually give effect to the terms of the written agreement between the parties without extraneous evidence. The contract, therefore, serves as evidence of the intention between the parties and must be carefully drafted particularly when it involves multiple parties.

2.4 Relevant standards

These are thousands of voluntary technical standards which have been established by many organizations for various uses, which are beyond the scope of this article. The following is thus only a small sample of relevant standards, as examples of a much larger collection:

BS EN 61508 "Functional safety of electrical/electronic/programmable electronic safety-related systems" is a commonly used engineering safety standard which defines "Safety Integrity Levels" (SIL), and technical safety processes such as the use of hazard identification and mitigation, failsafes, and emergency stop systems.

BS EN 62061—implements principles of BS EN 61508 (above) and harmonised to parts of the EU Machinery Directive (ie. a "called up" standard with legal status.)

ISO 10218 "Robots and robotic devices—safety requirements for industrial robots." provides best practices for industrial robot safety. ISO 15066 "Robots and robotic devices—collaborative robots" provides best practices for systems involving robots and humans working together.

ISO 18497 "Safety of autonomous tractors"—is under development at the time of writing, and aims to provide best practices for the safety of large autonomous tractors.

3 Law and mitigation of damages

A number of the laws examined above appear to be strictly worded concerning different forms of liabilities whether these arise under contract, tort or statute. However, despite the strictness, the law also provides defences and other legal means through which a stakeholder in an agribot supply chain may avoid liability or mitigate its damages:

(a) Statutory defences statutory defences are defences allowed under the law. The relevant defences have been discussed under the Product defect above.
(b) Defences to claims for tortious liability although, these have also been alluded to earlier, it is useful to briefly highlight how the manufacturer could in practice defend an action in negligence. As noted above, manufacturers may be liable for breaching a duty of care owed to users of their products, and the law places the burden of proving the negligent act on the injured party or the claimant. In effect, the claimant must prove that the manufacturer did not take reasonable care to avoid the injury or damage occurring. Conversely, it is a defence open to the manufacturer that he could not have reasonably foreseen the harm to the injured party and could, therefore, not have prevented it. This is based on the doctrine of the remoteness of damage where the manufacturer contends that there is no causal link between the manufacturer's negligence and the injury to the claimant. Also, the manufacturer can plead that the claimant is contributorily negligent. Contributory negligence is a partial defence which enables the negligent party (e.g., the manufacturer) to claim mitigation of damages by proving that the claimant contributed to his loss or injury. For example, failure to read the instruction manual to take specific recommended steps in circumstances where the agribot malfunctions may lay a farmer or contractor claimant liable to a claim of contributory negligence.
(c) Exclusion clauses where law permits it, parties may by contract exclude liability for certain acts or omissions. For example, the agribot manufacturer may exclude liability for illegal use of a robot or use for purposes other than that for which robot was manufactured. Liability may also be excluded for improper use or interference with specifications of agribot software or algorithm (See further notes on dual-use items below).

[37] Although the contractor can bring an action in tort if this is due to a defect in the product but only if he also suffers a damage. See notes on product defect above.

(d) Regulation by design The law now actively promotes regulation by design. Under the GDPR, data controllers are required to implement appropriate technical and organizational measures and procedures in such a way that data processing will meet the requirements of the law. It is significant that the law allows data controller to take account of state of the art in technological development and the cost of implementation in conforming to the requirement (See notes above).

(e) Insurance There is no specific insurance framework for robotics. However, insurance can be mandated by law or by contract between the parties. Also, specialist insurance may be required, and in this regard, it has been proposed that insurance develop new products and law mandate compulsory insurance scheme supplemented by a fund (See further notes below).

(f) Judicial approaches The Legal regime for compensation and Judicial approaches to the award of damages could also have a mitigating effect on liability. Courts are careful not to expand the scope of existing liability regime. For example, the provisions of the Compensation Act and the approach by the courts suggests that courts may be circumspect in allowing claims for damages caused by the agribots considering its essential economic function of killing weeds and making more land available for farming. For example, under the Compensation Act 2006 (UK), courts are required to take into account the fact that allowing specific claims may have adverse consequences for innovation and investment in desirable activities.[38] In other words, the law considers that if it is too easy to make successful claims concerning specific activities, the courts may be overwhelmed with cases for compensation. This risk often referred to as 'opening the floodgates (of litigation)' would inhibit investment in activities which are useful for the society or which are of economic, social or technological significance.

4 Grey areas relating to current legal concepts

The grey areas refer to aspects of liability in agricultural robotics where the law is unclear or uncertain. The most relevant issues considered here include the legal effect of the autonomy of agribot on the liability of the parties and liability for dual-use of the agribot as well as the likely effects of the EU data protection law. This section highlights the main arguments in this area and where relevant, the proposed solutions to the challenges.

[38] See Compensation Act 2006s 1(b).

4.1 The ethics of robot autonomy

Under EU law, (non-autonomous) robots can be classified as products and humans are ultimately responsible for defects, errors, or misuse of the robot (See notes on liability for product defect above). For autonomous robots, the applicable laws and principles are not so clear. Directive 85/374/EEC has no direct applicability to liability for damages caused by autonomous robots, and there is currently no definition of autonomous robots under EU laws.

Nevertheless, one proposal defines robot autonomy as the ability of the robot to take decisions and implement them in the outside world independently of external control or influence.[39] The key features of robot autonomy include the development of autonomous and cognitive features such as the ability to learn from experience and take independent decisions, increasing capacity for adaptability and the exhibition of emergent behaviours. In effect, if an autonomous robot encounters difficulties that its design did not anticipate, its actions will not always be a result of programming as its learning abilities can cause the robot to develop sophisticated interaction with the environment which leads to unpredictability in its behaviour.[40]

Presumably, therefore, the more autonomous robots are, the less likely they will be considered as mere tools in the hands of other actors such as manufacturer, owner and users.[41] However, it is not always clear whether and the extent to which robots should be autonomous. As examples, the UK House Committee on Robotics and Artificial Intelligence made the point that it is important that AI technology is operating as intended and that unwanted, or unpredictable, behaviours are not produced, either by accident or maliciously'.[42] Also, in a report by the EU, it was suggested that it is inconceivable that once another actor no longer controls a robot, it becomes the actor itself. The report argues further that a robot being a mere machine and a carcass devoid of consciousness, feelings and thoughts or its own will cannot become an autonomous legal actor.[43] This observation arguably undermines the very notion of robot autonomy. For example, since autonomy is taken to involve self-learning and the processing of artificial intelligence, then a design

[39] Committee on Legal Affairs, 'Draft Rules on Robotics', Recital R.

[40] Committee on Legal Affairs Draft Rules on Robotics, Recital Z.

[41] Committee on Legal Affairs Draft Rules on Robotics, Recitals Q, R, S.

[42] See House of Commons, Science and Technology Committee, 'Robotics and Artificial Intelligence' (12 October 2016) p 16.

[43] European Parliament, European Civil Law Rules in Robotics (Study for the JURI Committee) 2016 p 13 (hereinafter EU Parliament, Civil Law Rules in Robotics).

that limits that autonomy also limits the use of the robot and could potentially stifle further innovation in robotics.

As noted in another report by the EU, however, it is expected that ultimately AI could surpass human intellectual capacity in a manner which, if not prepared for, could pose a challenge to humanity's capacity to control its creation …'[44] This position suggests that robot autonomy is a given and technical (but legitimate) questions can be raised concerning the legal consequences of such autonomy particularly the legal responsibility arising from a robot's harmful action. In a hypothetical scenario involving the agribot, the following could occur; the obstacle avoidance on the agribot works but the robot 'decides' it could overcome an obstacle. An accident occurs, and a walker is injured. All parties deny liability. The manufacturer argues that the accident occurred independently as the robot was acting autonomously, the insurers refuse to indemnify the manufacturer based on the argument that the operation which caused the accident is not a 'defect', the injured party claims that the accident is caused by manufacturer defect regardless of robot autonomy. The question this raises is, therefore, is whether and how a machine can be held liable for its actions or omissions.[45]

Although it is not yet clear what values machines should use, and how to embed these values in them, it has been suggested that they should function according to values that are aligned with those of humans and consider following, as much as possible, ethical theories defined for humans.[46] Therefore, guiding legal and ethical frameworks for the design, production and use of robots and AI must be based on values such autonomy, individual responsibility, informed consent, and privacy and social responsibility.[47] The proposals examined below are relevant in this respect.

4.1.1 Proportional liability

To promote certainty, responsibility and accountability, it has been suggested that a set of rules be developed which reflects the proportionality of liability depending on the instructions given to the robot and its capacity for self-learning as well as its level of autonomy.[48] Assuming that damage, injury or loss could be established, the following rules of liability apply;

(a) Manufacturers and producers should be strictly liable for damage that can be traced back to the robot's design such as an error in the algorithm causing injurious behaviour.[49] (See notes above on how technical designs can aid the law in this area mainly because of attribution problems)

(b) For robots sold with open source software, liability should in principle be on the person who programmed the application which led to the robot causing damage. This is increasingly being incorporated into contracts.

(c) When damage is caused when the robot is still learning, its user or owner should be held liable. However, liability should be further governed by whether the user is a professional user and whether or not they are the victim. If the damage is caused to a victim who is also a professional, this would be considered as an accident at work covered by existing laws governing such accidents. If the damage is linked to robot instruction given by a professional user which causes damage to a third party, then the situation calls for the development and application of new rules

(d) In cases where the robot is hired out, the hirer should remain liable. The rationale is that it is difficult, given that each hirer may teach the robot different things, to determine which hirer is responsible for the acts of the robot.

For agribots, manufacturers and agricultural contractors are likely to fall into this category and would thus be deemed to be liable in cases where the agribot is hired out.

(e) Finally, future legislative instruments should provide for the application of strict liability for damage caused by smart robots. In effect, only proof of a causal link between the harmful behaviour of the robot and the damage suffered by the injured party is required. There should also be no restrictions on the type and extent of damages which may be recovered, and there should be no limit on the forms of compensation which may be offered to the aggrieved party on the sole ground that damage was caused by a non-human agent.[50]

4.1.2 Legal personhood

It is possible in theory to confer legal personality on robots. This allows the autonomous robot to have the status of an 'electronic person' for liability and rights.[51] "Legal personhood" is a purely legal concept and is unrelated to the concept of "personhood" in Philosophy, which has been

[44] Committee on Legal Affairs Draft Rules on Robotics, Recital I.

[45] Committee on Legal Affairs Draft Rules on Robotics, Recitals S.

[46] Francesca Rossi, 'Artificial Intelligence: Potential Benefits and Ethical Considerations' (European Parliament Legal Affairs Committee Briefing) 2016 p 4.

[47] Committee on Legal Affairs, 'Draft Rules on Robotics', 2016 p 7.

[48] EU Parliament, 'Civil Law Rules in Robotics', 17.

[49] EU Parliament, 'Civil Law Rules in Robotics' p 17.

[50] See generally Committee on Legal Affairs Draft Rules on Robotics, p 13.

[51] This is similar to the concept of corporate legal personality which allows confers legal entities on companies and corporations, thus separating the corporation or company from its promoters, managers and directors.

defined by various authors via difficult philosophical properties such as "free will" and "consciousness". Legal personhood in robotics would be intended purely as a mechanism to assign legal liability, in the same way that corporations are sometimes considered to be legal persons. In particular, like corporate personhood, it provides a mechanism to replace liability assigned to individual human operators with liability assigned to some group of humans such as the robot design team. This is important and useful, for example, if individual human operators do not wish to take on potential personal liability for deaths caused by the robots, which could result in prison and other sentences on them as individuals. Spreading the liability across the design team via legal personhood would avoid this situation whilst still ensuring that the responsibility still exists in a suitable form.

However, objections to robotic legal personhood have been raised to this proposal on ethical and conceptual coherence grounds. It w argued for instance that legal personhood status for the robot would unavoidably trigger unwanted and nonsensical legal consequences including the need to determine what robots' rights would be and how to respect those rights. In theory, a robot legal person (or more likely, a belligerent human claiming to act on its behalf, for example to sabotage a robotics company's product or service, as human campaign groups currently do against animal testing companies by acting on behalf of the animals) might then be entitled to demand rights for the robot which were originally intended only for humans legal persons, such as employment leave, minimum wage, and refusal to work in dangerous environments.

Although conferring rights on robots could be potentially nonsensical, the problem only arises if the arguments are considered from a purely economic perspective. From a legal perspective, an artificial legal entity does not have to be conferred with the same rights as humans. In fact, taking the example of corporations, the law may not confer any direct rights or duties on the entity but rather on its directing minds or promoters. Therefore, for robots, electronic personhood would create the advantage of legal convenience such as making the robot a distinct legal entity which can sue and be sued. It would also vest the robot with the genuinely useful capability to apply for and obtain work or operating licence (e.g., an agribot (or rather its designers on its behalf) can apply for certification to use pesticides, removing the need for operators to hold the certificate, and transferring the liability onto the engineering design team). Electronic personhood can also help the robot (or rather, the human design team which it represents) fulfil obligations to self-insure and like corporations, pay compensation to those injured by its acts or omissions, again reducing the risk to individual operators.

It is important to note that the robot will have to be registered in the same way as corporations and may have to be vested or equipped with assets to enable it to carry out its duties and obligations. The promoters of the robot will make the choices about which party(ies) will fund the assets. More importantly, however, despite the electronic personhood, the court would be able to lift the veil of incorporation in appropriate cases to render the promoters liable for crimes and civil wrongs committed by the robot.

4.1.3 Registration and insurance

This is a recommendation for a system of registration for advanced robots based on a criterion established for the classification of robots.[52] A Union-wide Agency would manage the registration which would serve the purpose of traceability for robotics and artificial intelligence.[53] Similarly, the proposal for insurance advocates the establishment of an insurance scheme which obliges the producer to take out insurance for the autonomous robot it produces. It is proposed that a fund supplements the obligatory insurance scheme to ensure that damages can be compensated for in cases where no insurance cover exists.[54]

4.2 Dual-use products

EU law regulates Dual-use products.[55] The Regulation sets up a Community regime for the control of exports, transfer, brokering and transit of dual-use items and aims to control trade in dual-use items to counter the proliferation of weapons of mass destruction and other items of potential military use.[56] Therefore, the Regulation requires that Dual-use items (including software and technology) should be subject to effective control when they are exported from the European Community.[57] Dual-use items are defined as '…items, including software and technology, which can be used for both civil and military purposes, and shall include all goods which can be used for both non-explosive uses and assist in any way in the manufacture of nuclear weapons or other nuclear explosive devices'. [58] Annexe 1 to the Regulation contains a list of dual-use items including nuclear materials (e.g., uranium), telecommunications and information

[52] Committee on Legal Affairs Draft Rules on Robotics, p 13.

[53] Committee on Legal Affairs Draft Rules on Robotics p 13.

[54] Committee on Legal Affairs Draft Rules on Robotics p 13.

[55] See COUNCIL REGULATION (EC) No 428/2009 of 5 May 2009 setting up a Community regime for the control of exports, transfer, brokering and transit of dual-use items (hereinafter Regulation 428/2009).

[56] See EU Parliament, 'Implementation Appraisal Control of Trade in Dual Use Items' (…Committee briefing 2016).

[57] Regulation 428/2009, Recitals 2, 3.

[58] Regulation 428/2009, art 2.

security, sensors and lasers, various software, machine tools, chemical manufacturing equipment.

The law requires dual-use items to be registered and subject to authorisation and export control including a detailed register of exports (art 20) and a review and update as well as impact assessment of dual-use items.

It is notable that the law can be extended to products with potential dual-use that are not listed in Annex 1 to the Regulations (art 4, 15). In fact, in its draft rules on robotics, the EU legal committee recommends that the provisions on dual-use regulations should apply to robots.[59] Perhaps, because the Regulation intends to ensure that dual-use items do not get into the hands of malicious actors, it only imposes liability on manufacturers for non-compliance with relevant provisions. However, new issues on liability can arise in the use and operation of the agribot. To illustrate, as agricultural robots are designed to operate in harsh outdoor conditions, they may bear functional similarities to, and be repurposable as, military systems such as explosive ordnance disposal (EOD), reconnaissance, and weaponised platforms. It is, therefore, conceivable that in the wrong hands they could be used to commit crimes including acts of terrorism, such as delivering lethal substances or weapons into crowded areas.

It is clear on the one hand that the malicious actor or any other person(s) that repurposed the agribot to carry out the criminal or terrorist acts would be deemed to have committed a crime for which he would be liable to punishment upon conviction. Also, he could be liable for damages to the parties thereby injured in a civil action. On the other hand, it is not clear whether the manufacturer bears (or should bear) any liability. As already noted above, the EU Directive on Product Defect applies only to defective products; that is, products not providing the safety to which a consumer is entitled. It is also notable that one of the factors to be taken into account - in determining whether a product is defective or not- include whether the product is being put to reasonable use.[60] However, while unreasonable use can give rise to mitigation of damages, it does not entirely absolve the manufacturer of liability, and the problem can become particularly complex if such re-purposing is foreseeable or can be anticipated by the manufacturer. Under the EU Machinery Directive, for instance, the contents of the instructions must cover not only the intended use of the machinery but also take into account any reasonably foreseeable misuse.[61]

The question, therefore, is what uses should be deemed reasonably foreseeable? For example, is it reasonably foreseeable that an agribot can be used for criminal or terrorist purposes? If the answer is yes, then how is the position different from using a kitchen knife to commit murder. The knife is sold as a kitchen utensil, not as a weapon, so although the manufacturer can reasonably foreSee that the knife could be used for heinous crimes, he is not held responsible for the murderer's action. Arguably, the argument would be different if the robot was developed purely for the purpose of committing crimes—such as a modified agricultural robot with a new implement attachment designed specifically for breaking and entering domestic windows and with no other clear function—then responsibility can lay with the manufacturer if the robot is repurposed for further criminal or unlawful purpose. As described in the discussion of health and safety law in Sect. 2, the resolution of this discussion is largely dependent on whether firstly, practicable risk reduction measures (i.e., what can be done about the reasonably foreseeable hazards) are readily identifiable, and secondly, whether implementation of those measures is reasonable. In the case of the knife, a well-established implement for which many examples of good practice design are available, it is unlikely that further risk reduction measures are practicable (i.e., technically feasible) that have not already been tried and their relative virtue exhaustively evidenced. In the case of the agricultural robot, the industry is still subject to errors in internal communication, for example, due to intellectual property protection, lack of established industry groups and forums, and general lack of publicly available evidence of safety improvements; therefore, the identification of practical risk reduction measures and the reduction in risk associated with reasonably foreseeable hazards may not be straightforward for designers.

Furthermore, in what ways should the instructions take into account reasonably foreseeable misuse? For example, the fact that instructions expressly prohibit certain re-programming or re-purposing would hardly deter a malicious actor bent on misusing the agribot. These issues would need to be addressed when developing rules applicable to robotics particularly small robots like the agribot.

4.3 Scope and application of data protection law

It was noted earlier that the new EU Regulations on data protection make significant provisions that would impact on developments in robotics and AI. While Much of the provisions address gaps in the law, they also raise difficult questions on the scope of the law and its impact and applicability to robotics. As examples, article 25 now makes PbD a legal standard, and arguably enhances the protection of individual privacy. However, given the rapidly evolving technology environment, and the fact that vulnerabilities and susceptibilities (to privacy infractions) may only become

[59] Committee on Legal Affairs 'Draft Rules on Robotics' item 34 p 12.

[60] See notes on product defects above.

[61] See Directive 2006/42/EC, item 1 of Annex 1.

known subsequent to use or operation of products, as well as the expansive and ambulatory nature of the concept of privacy,[62] the question must be asked whether it is possible (even using state of the art) to identify and assess all privacy implications and dimensions of particular technologies?

Furthermore, under article 22 relating to algorithms that make decisions based on user-level predictors which significantly affect users, the law effectively creates a 'right to explanation'. This entitles users to ask for an explanation of an algorithmic decision that was made about them (See previous notes above). Although decisions based on algorithms raise difficult ethical and privacy questions, the provision also poses significant challenges for the AI and machine learning community. As examples, it is a common misconception that complex algorithms always do what their designers choose to have them do when in fact, it is difficult to understand, predict, and explain the behaviour of advanced AI systems because of the complexity of the systems and the large volume of data they use.[63] Also, from a technical perspective, a requirement that algorithms offer explanations for their underlying decisions could potentially prohibit the algorithms currently in use. This means to comply with the law, a complete overhaul of standard and widely used algorithmic techniques may be required.[64]

Finally, while the GDPR applies directly to all EU member states, it is unclear, given the uncertain political terrain precipitated by Brexit, how and the extent to which the law would apply to the UK. For example, even if the UK adopts the GDPR, (which will take effect before the UK exits the EU), will the UK be bound to continue to implement the GDPR and its subsequent amendments? What would be the effect of the opinions, studies etc. conducted by the EU on the formulation of policies on robotics, machine learning, AI and cognitive computing in the future? More importantly, since the GDPR now establishes both a European Data Protection Board (EDPB) and national supervisory authorities, what are the effects of the multiple (or at least dual) administrative and compliance regimes that the Brexit could potentially create for the AI and robotics community in the EU?

[62] for example, privacy can be relative to context, societies and even technologies.

[63] Executive Office of the president National Science and Technology Council Committee on Technology, 'Preparing for the Future of Artificial intelligence' (2016) p 31; See also Arnold v Reuther 92 So. 2d 595, 596 where the court appeared to sanction this notion of autonomy when it suggested that liability in cases of autonomous vehicles will be higher because they (the vehicles) raise the presumption that their programming is accurate.

[64] Bryce Goodman and Seth Flaxman, 'European Union Regulations on Algorithmic Decision-making and a Right to Explanation' p 1.

5 Conclusion

The liability regime which applies to the use and operation of the agribot appears to be complicated. However, an essential aspect of this regime is that parties have different rights and obligations under different laws which make it possible to distribute liabilities. The law also allows defences which are particularly specific and relevant for promoting developments in technology. More crucially, where permitted by law, parties may re-allocate liabilities and claim contributions for damages arising from accidents involving the agribot.

The outstanding issue requiring consideration is how autonomy should be defined in the context of the operation of the agribot. Unless law, policy or (for present purposes) contracts define the scope of the autonomy of the robot, the liability regime may be challenged by technical legal arguments.

Acknowledgements The authors would like to thank Ed Thomas at Risktec Solutions Ltd and the anonymous reviewers for their useful comments. This research was supported in part by the InnovateUK project IBEX2: "Autonomous robot weed spraying for less favoured areas", grant number 131790.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

Anderson JM, Nidhi K, Stanley KD, Sorensen P, Samaras C, Oluwatola OA (2014) Autonomous vehicle technology: a guide for policymakers. Rand Corporation

Andrieu C, Freitas ND, Doucet A, Jordan MI (2003) An introduction to mcmc for machine learning. Mach Learn 50(1–2):5–43

Bac CW, Henten EJ, Hemming J, Edan Y (2014) Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. J Field Robot 31(6):888–911

Beck S (2016) The problem of ascribing legal responsibility in the case of robotics. AI Soc 31(4):473–481

Beiker SA (2012) Legal aspects of autonomous driving. Santa Clara L Rev 52:1145

Bernardo JM, Smith AFM (2001) Bayesian theory

Binch A, Fox CW (2017) Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland. Comput Electron Agric 140:123–138

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York. ISBN 978-0-387-31073-2. https://www.springer.com/us/book/9780387310732

Blackmore S, Godwin RJ, Fountas S (2003) The analysis of spatial and temporal trends in yield map data over six years. Biosyst Eng 84(4):455–466

Blackmore S, Griepentrog HW, Nielsen H, Nørremark M, Resting-Jeppesen J (2004) Development of a deterministic autonomous tractor. In Proceedings CIGR, vol. 11, p 2004

Brodsky JS (2016) Cyberlaw and venture law: Autonomous vehicle regulation: how an uncertain legal landscape may hit the brakes on self-driving cars. Berkeley Tech LJ 31:851–1169

Chua PY, Ilschner T, Caldwell DG (2003) Robotic manipulation of food products—a review. Ind Robot Int J 30(4):345–354

Cooper GF (1990) The computational complexity of probabilistic inference using Bayesian belief networks. Artif Intell 42(2–3):393–405

Douma F, Palodichuk SA (2012) Criminal liability issues created by autonomous vehicles. Santa Clara L Rev, 52:1157

Dr. Mike Webster (2017) HSE Improvement and Prohibition Notices: what do they tell us about CDM 2015 and construction health and safety? V1.0 November

Dvorak J (2016) An autonomous, solar-powered tractor. 2016. American Society of Agricultural and Biological Engineers

Escolà A, Rosell-Polo JR, Planas S, Gil E, Pomar J, Camp F, Llorens J, Solanelles F (2013) Variable rate sprayer. Part 1—orchard prototype: design, implementation and validation. Comput Electron Agric 95:122–135

Fox CW, Roberts SJ (2012) A tutorial on variational bayesian inference. Artif Intell Rev 38(2):85–95

Guizzo E (2011) How Google's self-driving car works. IEEE Spectrum Online, October 18, 2011

Health and Safety Executive (2001) (R2P2) Reducing risks, protecting people – HSE's decision-making process

Ishida M, Imou K, Okado A, Takenaga H, Honda Y, Itokawa N, Shibuya Y (1998) Autonomous tractor for forage production. J Jpn Soc Agric Mach 60(2):59–66

Levitt TS, Laskey KB (2000) Computational inference for evidential reasoning in support of judicial proof. Cardozo L Rev 22:1691

Marino D, Tamburrini G (2006) Learning robots and human responsibility. Int Rev Inf Ethics 6(12):46–51

Michio K, Noguchi N, Ishii K, Terato H (2002) The development of the autonomous tractor with steering controller applied by optimal control. In: Automation technology for off-road equipment proceedings of the 2002 conference, p 367. American Society of Agricultural and Biological Engineers

Pedersen SM, Fountas S, Have H, Blackmore BS (2006) Agricultural robots—system analysis and economic feasibility. Precis Agric 7(4):295–308

Pinto C (2012) How autonomous vehicle policy in California and Nevada addresses technological and non-technological liabilities. Intersect Stanf J Sci Technol Soc 5

Singh N, Shaligram AD (2014) NPK measurement in soil and automatic soil fertilizer dispensing robot. Int J Eng Res Technol 3:635–637 **(ESRSA Publications)**

Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. MIT press, Cambridge

Transparency Market Research (2017) Agriculture robots market—global industry analysis, size, share, growth, trends and forecast 2016–2024. Transparency Market Research

UNIVERSITY OF LEEDS

This is a repository copy of *Models of human decision-making as tools for estimating and optimising impacts of vehicle automation*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/127540/

Version: Accepted Version

Article:
Markkula, GM orcid.org/0000-0003-0244-1582, Romano, R orcid.org/0000-0002-2132-4077, Madigan, R orcid.org/0000-0002-9737-8012 et al. (3 more authors) (Accepted: 2018) Models of human decision-making as tools for estimating and optimising impacts of vehicle automation. Transportation Research Record. ISSN 0361-1981 (In Press)

© 2018. This is an author produced version of a paper accepted for publication in the Transportation Research Record. Reprinted by permission of the publisher.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

MODELS OF HUMAN DECISION-MAKING AS TOOLS FOR ESTIMATING AND OPTIMISING IMPACTS OF VEHICLE AUTOMATION

Gustav Markkula
(corresponding author)
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Tel: +44 113 3439832; Email: g.markkula@leeds.ac.uk

Richard Romano
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Tel: +44 113 3438466; Email: r.romano@leeds.ac.uk

Ruth Madigan
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Tel: +44 113 3432071; Email: r.madigan@leeds.ac.uk

Charles W. Fox
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Tel: +44 113 3431794; Email: c.w.fox@leeds.ac.uk

Oscar T. Giles
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Email: o.t.giles@leeds.ac.uk

Natasha Merat
Institute for Transport Studies, University of Leeds
LS2 9JT Leeds, United Kingdom
Tel: +44 113 3436614; Email: n.merat@leeds.ac.uk

Word count: 5,736 words text + 7 figures x 250 words (each) = 7,486 words

Submission date: 1/08/2017
Revision date: 15/11/2017

ABSTRACT

With the development of increasingly automated vehicles (AVs) comes the increasingly difficult challenge of comprehensively validating these for acceptable, and ideally beneficial, impacts on the transport system. There is a growing consensus that virtual testing, where simulated AVs are deployed in simulated traffic, will be key for cost-effective testing and optimisation. The least mature model components in such simulations are those generating the behaviour of human agents in or around the AVs. In this paper, human models and virtual testing applications are presented for two example scenarios: (i) a human pedestrian deciding whether to cross a street in front of an approaching automated vehicle, with or without external human-machine interface elements, and (ii) an AV handing over control to a human driver in a critical rear-end situation. These scenarios have received much recent research attention, yet simulation-ready human behaviour models are lacking. They are discussed here in the context of existing models of perceptual decision-making, situational awareness, and traffic interactions. It is argued that the human behaviour in question might be usefully conceptualised as a number of interrelated decision processes, not all of which are necessarily directly associated with externally observable behaviour. The results show that models based on this type of framework can reproduce qualitative patterns of behaviour reported in the literature for the two addressed scenarios, and it is demonstrated how computer simulations based on the models, once these have been properly validated, could allow prediction and optimisation of AV impacts on traffic flow and traffic safety.

Keywords: automated vehicles, virtual testing, human behavior models

INTRODUCTION

In recent years, rapid technological progress has been made on automated vehicles (AVs); vehicles capable of taking over increasing shares of the driving task, with large hoped-for benefits in terms of increased mobility, improved traffic safety, reduced environmental footprint, and economic growth (*1*). As with any technology, it is important to subject AVs to sufficient testing and validation to ensure safe and effective operation. This is a daunting task, however, considering the essentially infinite variations of possible traffic situations that an AV needs to handle (*2*).

It is increasingly assumed that one important approach in the toolkit for AV validation will be *virtual testing* methods, where a computer simulation of an AV, integrating mathematical models of vehicle and sensors with the actual self-driving algorithms, can encounter a rich variety of simulated traffic situations (*3*, *4*). These traffic situations can for some purposes be replayed from large static databases of actual real-traffic recordings, but if the objectives of a virtual test depends on some form of *interaction* between the AV and surrounding road users—e.g., by means of external human-machine interfaces (*5*)—or interaction between the AV and its onboard operator, the involved human agents also need to be simulated.

Human road user models do exist, and are currently being actively developed and applied to AV testing (*3*, *6*), but so far mainly on the relatively coarse-grained level of *traffic microsimulation*, where the trajectory of each road user (car, truck, pedestrian, etc) is directly generated from equations of motion taking into account positions and velocities of surrounding road users. While this level of granularity might be adequate for some purposes, it seems likely that in many cases, the outcome of interactions between humans and AVs will hinge on finer details and time dynamics of human perception, scene interpretation, and decision-making (*7*, *8*).

Capturing such phenomena accurately in virtual simulation is of course very challenging, but recent literature suggest some possibly fruitful directions: There is by now ample neuroscientific support for the idea that perceptual decision-making in typical laboratory tasks is underpinned by noisy neural *evidence accumulation,* to a decision threshold at which the overt response is initiated (*9*). Interestingly, such models have also been proven useful for explaining human driver reaction times in responding to stimuli in traffic; discrete stimuli such as brake lights (*10*) but importantly also graded, dynamic stimuli like the *visual looming* of approaching road users or collision threats (*11*, *12*).

Here, this line of modelling will be taken one step further, and applied to the more complex decision-making situations that tend to arise in the AV context, where the emphasis of modelling naturally shifts away from low-level decisions like “more or less braking?” towards a higher-level assessment of the situation at hand; what is often referred to as *situational awareness* (*8*, *13*). The basic modelling idea to be explored here is that situational awareness can be modelled as a number of interrelated perceptual decisions about the world, where each decision is modelled as an accumulation process, but where these processes are also interconnected to influence each other in excitatory or inhibitory fashion. This is reminiscent of connectionist and activation dynamics type models of cognition (*14*, *15*). However, to our knowledge, the present paper is the first time several decision-making processes of an evidence accumulator type have been connected in a network arrangement as a means of modelling complex, multi-stage decision processes.

After a first section below describing the two AV-human interaction scenarios addressed here, three consecutive sections will pursue the three main objectives of the paper: (i) To suggest models of human decision-making in the targeted scenarios, as examples of the modeling approach sketched above. (ii) To illustrate that these models can qualitatively reproduce non-trivial behavior patterns reported in the literature. These prior findings thus serve as a first high level data set, to which model parameters are manually fitted. (iii) To illustrate how models at this level of detail

can be useful in virtual testing of AVs. It is not an aim here to suggest final model formulations, nor to perform full model validation and tuning; these will be matters for future work.

TARGETED AV-HUMAN INTERACTION SCENARIOS

Figure 1 shows the two traffic scenarios considered here. The pedestrian interaction scenario in panel (a) is based on the test track study by Schneemann and Gohl (16). They considered the question of how an AV should behave at a zebra crossing, by studying the interactive behaviour of human drivers and pedestrians, as a function of initial car speed v_0 and time to collision (TTC; or alternatively in this scenario time to arrival) at the moment when the pedestrian was instructed to step up to the crossing, prompting decelerations from the (uninstructed) drivers. Here, what will be modelled is the pedestrian's decision to cross or yield as a function of car movements, as well as of more direct indications from the car (from either the driver or the AV, as the case may be) about having seen the pedestrian, and about its possible intentions to yield.

The safety-critical take-over scenario in panel (b) is based on the simulator study by Gold et al. (*17*), where drivers either manually drove a vehicle or were driven by an AV at v_0 = 120 km/h (75 mph), when a lead vehicle changed lane to reveal a stationary vehicle, at a TTC of either 5 s or 7 s, in the AV case prompting a *take-over request* to the driver. Similar types of take-over scenarios have recently been subject to intense research efforts; e.g. (*7*, *18*). What will be modelled here is the human's decisions, after the obstacle appears, on where to look and how to manoeuvre the car.

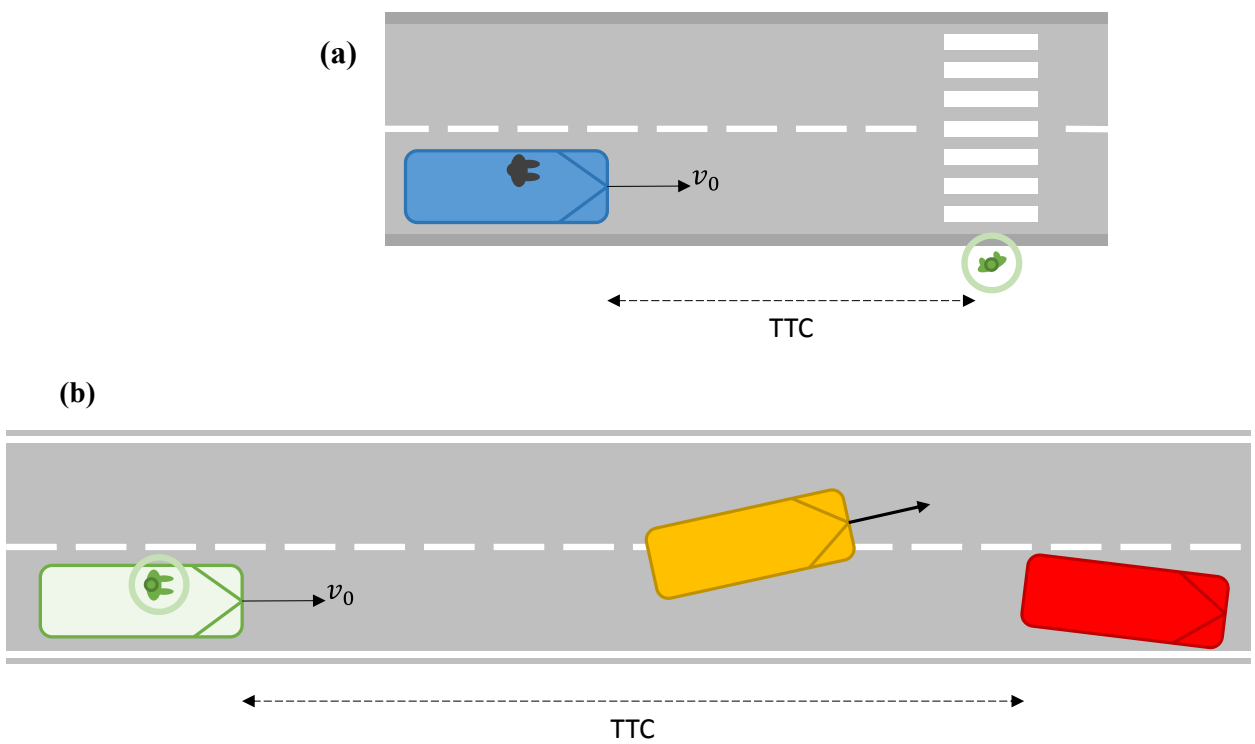


FIGURE 1 Schematic illustrations of the scenarios considered in this paper. The modeled human is indicated with a circle. TTC is short for time to collision. (a) Pedestrian interaction scenario. (b) Safety-critical take-over scenario.

MODELS

Model of pedestrian crossing decisions

The basic hypothesis behind the pedestrian model, illustrated in Figure 2, is that the pedestrian makes the action decision to cross based on either or both of two perceptual decisions: (i) A perceptual decision that it is possible to pass the road before the car arrives, made by observing the visual looming quantity $\tau = \theta/\dot{\theta}$, with θ the projected angle of the car on the pedestrian's retina. τ is a close approximation of TTC if assuming constant speeds (19), and is assumed to be compared to a threshold τ_{pass}. (ii) A perceptual decision that the car (driver or AV, but here "car" will be used for short) intends to yield, but not if the actual car movements clearly suggest that it is still unsafe to cross ($\tau < \tau_{pass}$). To make the decision about whether or not the car intends to yield, the pedestrian is assumed to monitor the quantity $\dot{\tau} = \mathrm{d}\tau/\mathrm{d}t$, which is ≥ -0.5 if the car stops at or before the zebra crossing (19), but also explicit communication acts (e.g., headlight flashes). However, if the pedestrian decides that the car has not seen him/her (e.g., based on driver head orientation), this is taken as evidence that the car is not intending to yield.

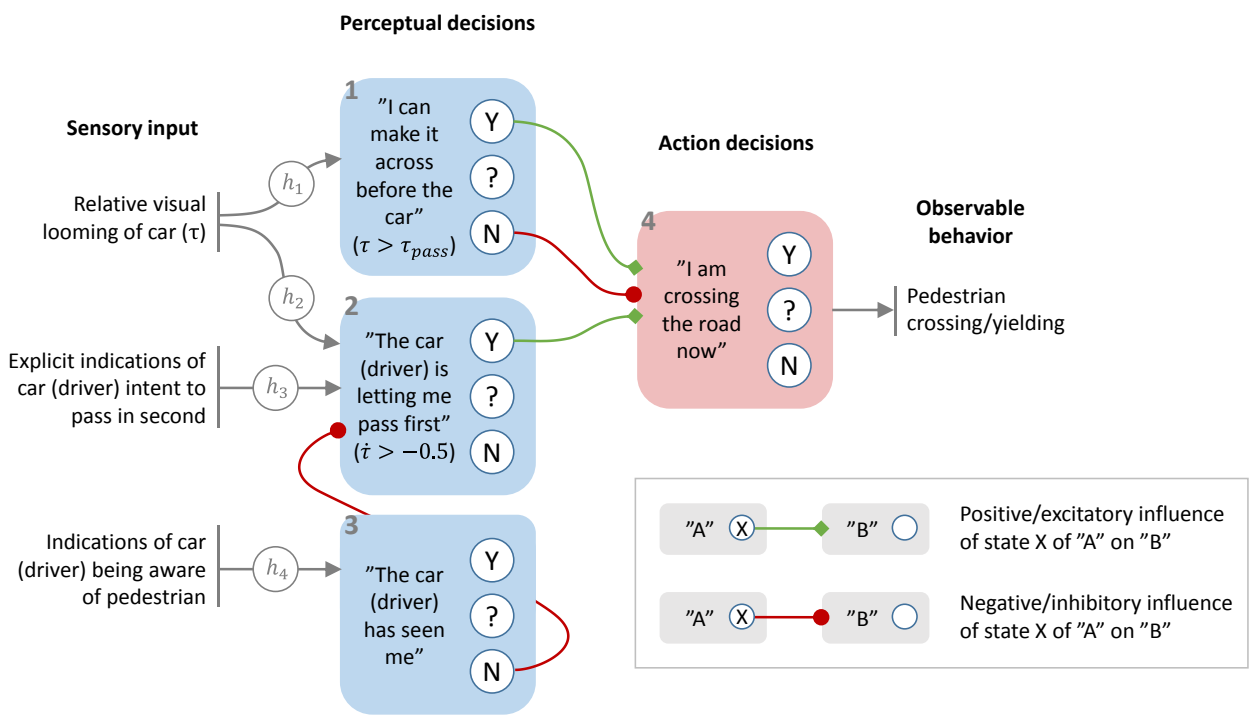


FIGURE 2 Schematic representation of the model for the pedestrian interaction scenario.

The remainder of this section provides a full mathematical specification of the model, but is not crucial for understanding the rest of this paper: Figure 2 shows four interconnected *decision units*, and at any point in time, the state of the model is described by the vector of unit *activations*, $\boldsymbol{A}(t) = [A_1(t) \ldots A_4(t)]^T$, where each A_i is a two-sided accumulator, bounded at plus and minus one, implying "yes" and "no" decisions, respectively, and zero implying maximum uncertainty. This type of accumulator is often referred to as a "drift-diffusion model for two-choice decisions" or similar (*20*), although interconnecting several of these or simulating them beyond a threshold being reached is possibly novel. At each new simulation time step, the A_i are updated as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{A}(t) = -\frac{1}{T}\boldsymbol{A}(t) + \boldsymbol{W}_Y\sigma_Y\big(\boldsymbol{A}(t)\big) + \boldsymbol{W}_?\sigma_?\big(\boldsymbol{A}(t)\big) + \boldsymbol{W}_N\sigma_N\big(\boldsymbol{A}(t)\big) + \boldsymbol{F}(t) + \boldsymbol{\nu}(t) \quad (1)$$

where T is a decay time constant, the $\boldsymbol{W}_\bullet$ are matrices of connections between decision units, the $\sigma_\bullet$ are activation functions, $\boldsymbol{F}(t)$ is a vector of external inputs, and $\boldsymbol{\nu}(t)$ is a vector of noise, accounting for between-trial behavioral variability due to for example variations in brain activity, modelled here as Gaussian noise with standard deviation $\Sigma\sqrt{\Delta t}$, where Σ is a parameter and Δt is the simulation time step (9). In this first model implementation, to simplify the manual tuning process, all elements in $\boldsymbol{W}_\bullet$ were kept either at zero (no connection) or ±1 (excitatory and inhibitory connections), rather than being freely tuned. Also for simplicity, the activation functions were kept piecewise linear: $\sigma_Y(A) = 0$ and 1 for $A \leq 0$ and $A \geq 1$, respectively, and linearly increasing in between, and $\sigma_N(A) = \sigma_Y(-A)$, such that the two functions respond maximally when A indicates "yes" and "no", respectively. Further, $\sigma_?(A) = 1 - \sigma_Y(A) - \sigma_N(A)$, i.e., zero at $A = \pm 1$ and linearly increasing from both sides to a maximum of one at $A = 0$. The external inputs to the decision units were:

- $F_1(t) = h_1\big(\tau(t)\big) = k_1\big(\tau(t) - \tau_{pass}\big)$, corresponding to the perceptual decision about whether the time margin for crossing is currently above $\tau_{pass} = 3$ s.
- $F_2(t) = h_2(\tau(t)) + h_3(t)$, where $h_2(\tau(t)) = k_2\big(\dot{\tau}(t) - \dot{\tau}_{stop}\big)$, corresponding to the perceptual decision about whether the car is decelerating to stop before the zebra crossing. Here $\dot{\tau}_{stop} = -0.55$ was used, to accumulate positive evidence also in the limit case $\dot{\tau} = -0.5$. The input $h_3(t)$ was set to one to model the case where the car is providing some communicative act interpreted by the pedestrian as indicating an intention to yield, e.g., flashing the headlights, zero otherwise.
- $F_3(t) = h_4(t)$ was set to one to model the car or its driver doing something that the pedestrian takes as evidence of the car seeing the pedestrian, e.g., eye contact or related head pose in the case of a human driver, or some external HMI in the case of an AV, or minus one if the car is doing something that the pedestrian takes as evidence of the opposite, e.g., a car driver is looking somewhere else.
- $F_4(t) = 0$.

Manual tuning indicated satisfactory model behavior for $T = 2$ s; $\Sigma = 0.5$; $k_1 = 0.5$; $k_2 = 4$.

Model of driver behavior during safety-critical take-over

The model for the take-over scenario, illustrated in Figure 3, is slightly more complex. It does not terminate at the first action decision like the pedestrian model; instead each new action affects how the scenario continues playing out over time. For reasons of space, all connections shown in Figure 3 will not be discussed individually here, but the main hypotheses behind the model are as follows: Driver braking is assumed to function as proposed in (*12*), as discrete increases of brake pedal position applied each time a discrepancy between visual looming, quantified as τ^{-1}, and predicted visual looming τ_p^{-1}, updated after each new brake adjustment, accumulates to threshold. The decision to change lane can come about in the model in two ways: an early detection of catching up with the lead vehicle (which does not in itself prompt braking) or

a later realisation that attempted braking is not solving the conflict, and both of these are also modulated by visual looming directly. However, the lane changing decision is in all cases inhibited—as in made less probable—if the driver does not know that it is safe to change lane. Gaze sampling is assumed driven to a large extent by the need to resolve uncertainties in the perceptual decisions (*21*), and the current gaze target determines what sensory input the model receives (the "switches" in Figure 3). In this first version of the model, there is no direct account of the process of actual control take-over; as a first approximation it will be investigated to what extent the effects of taking over from the AV can be accounted for as a simple time delay, combined with lower initial awareness of the adjacent lane.

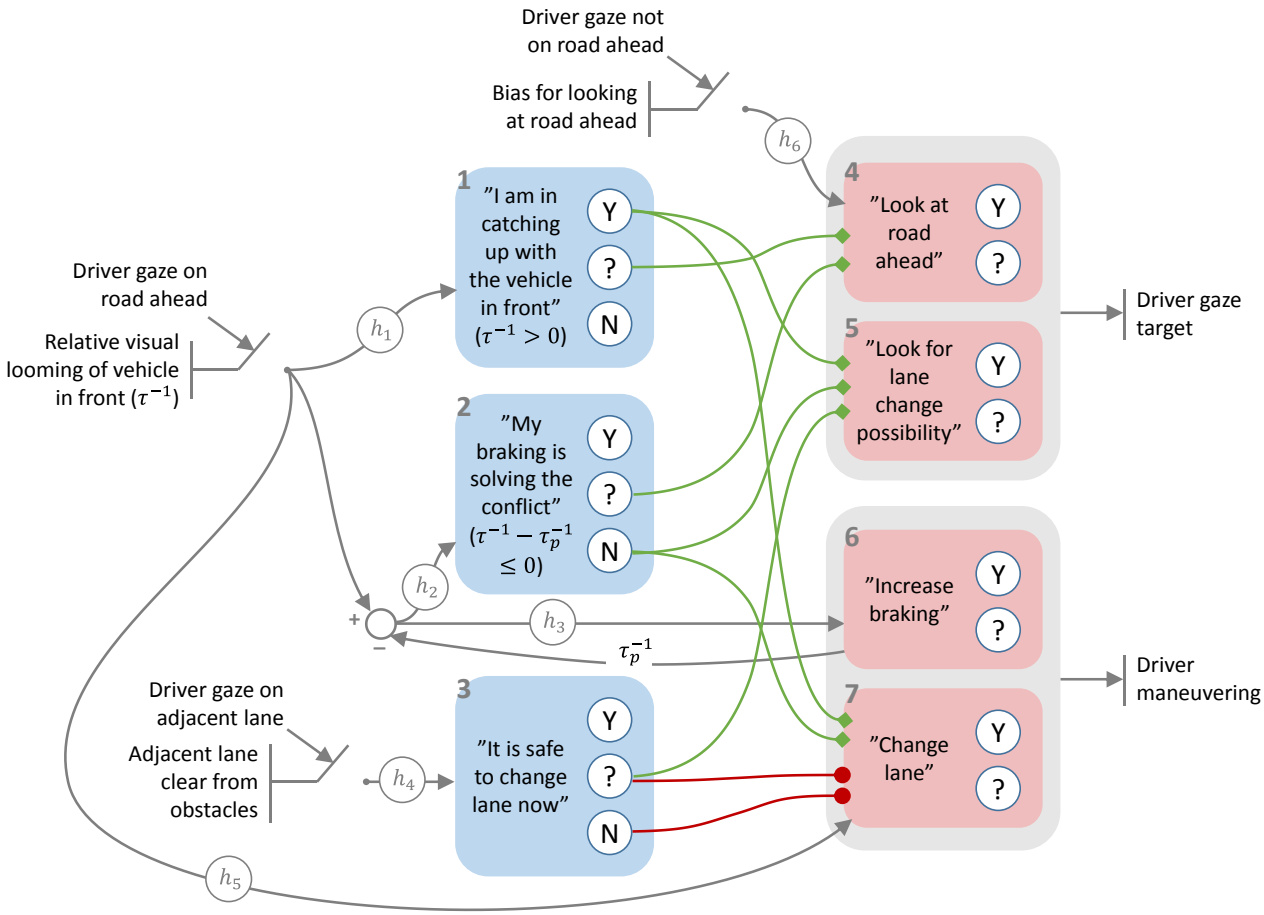


FIGURE 3 Schematic representation of the model for the safety-critical take-over scenario.

Again, the rest of this section gives mathematical model details: Just as in the pedestrian model, Equation (1) governs activations in the network. However, decisions between competing actions about where to look (units 4 and 5) or how to maneuver (units 6 and 7), are here modelled as "races" (*20*), bounded downward by zero instead of minus one, and each time an action threshold $A_i = 1$ is reached, the race in question is restarted by resetting the competing accumulators to zero. The exception is the "increase braking" action, which has its accumulator reset to 0.7, to reflect an increased brake readiness after the first brake application (*12*). Also, before the first brake application, decision unit 2 is inactive. The looming-related inputs to the network, all set to zero if the driver's eyes are off the road, were defined as:

- $F_1(t) = k_1\tau^{-1}(t)$, modeling the perceptual decision of whether the vehicle ahead is coming closer.
- $F_2(t) = -k_2(\tau^{-1}(t) - \tau_p^{-1}(t))$, modeling the perceptual decision of whether braking is solving the conflict.
- $F_6(t) = k_3(\tau^{-1}(t) - \tau_p^{-1}(t) - \tau_B^{-1})$, modeling the decision to increase braking (12).
- $F_7(t) = k_4(\tau^{-1}(t) - \tau_S^{-1}) - \min(0, k_5(\tau^{-1}(t) - \tau_{NS}^{-1}))$, modeling a positive looming contribution to the decision to change lane and a negative contribution to the same decision coming into play once $\tau^{-1} > \tau_{NS}^{-1}$, where $\tau_{NS}^{-1} > \tau_S^{-1}$.

The remaining inputs were defined as:

- $F_3(t) = k_6$ and $F_4(t) = k_7$, both if the driver is directing gaze towards the adjacent lane (whether by mirror checks, shoulder checks, etc, is not defined in the model), zero otherwise.
- $F_5(t) = 0$.

Manual tuning indicated satisfactory model behavior for $T = 2$ s; $\Sigma = 0.3$; $k_1 = k_2 = 20$; $k_3 = k_5 = 6$; $k_4 = 3$; $k_6 = 1.5$; $k_7 = 1.2$; $\tau_B^{-1} = \tau_S^{-1} = 0.1\ \mathrm{s}^{-1}$; $\tau_{NS}^{-1} = 0.5\ \mathrm{s}^{-1}$.

MODEL BEHAVIOR – REPRODUCING FINDINGS FROM LITERATURE

Reproducing pedestrian crossing behavior

Figure 4 shows example simulations with the pedestrian model, all at $v_0 = 50$ km/h (31 mph) and with TTC = 4 s at the moment when the pedestrian steps up to the zebra crossing, at which point the simulated car also initiates a deceleration to stop exactly at the zebra crossing. Panel (a) shows example simulations both with and without noise, whereas panels (b) and (c) show distributions for 10,000 simulations with noise.

In panel (a), note in the simulation without noise how the pedestrian comes close to a quick decision to cross while τ is above the threshold $\tau_{pass} = 3$ s, but since the margin $\tau - \tau_{pass}$ is small, the evidence accumulation is so slow that τ falls below τ_{pass} before a complete crossing decision has formed. From that point, despite picking up some signs of car deceleration, the pedestrian still judges from τ that crossing might be risky. Thus, as further illustrated by the delay time distribution in panel (b), one qualitative human behavior result that the model reproduces is the finding reported in (*16*), where 20 % of pedestrians waited until the car came to a complete stop before crossing. In the model simulations, the remaining 80 % are spread out in time with a wide peak in the time region while τ is still above τ_{pass}, where the model can make the decision to cross even if it is unsure of the car's intentions. This aligns well with another finding in (*16*), where 25 % of pedestrians stated that they made the decision to cross without having seen the car's deceleration. Exactly what activation A of the deceleration-detecting perceptual decision would yield a positive response to such a question is not defined in the model, but as a rough comparison, 25 % of the light brown dots in panel (b) lie below $A = 0.3$.

The scenario in Figure 4(b) assumes that the pedestrian gets indications of being seen by the driver (e.g., eye contact) but no explicit communication about yielding intentions, whereas panel (c) shows variations in this interactive behaviour. In the best case, where the pedestrian gets indications both of the car being aware of him/her, and of the car's intention to stop, the model pedestrian almost never waits until the car is stationary (dotted black line), whereas more than half of the model pedestrians wait that long in the worst case where neither of those indications are present (solid gray line).

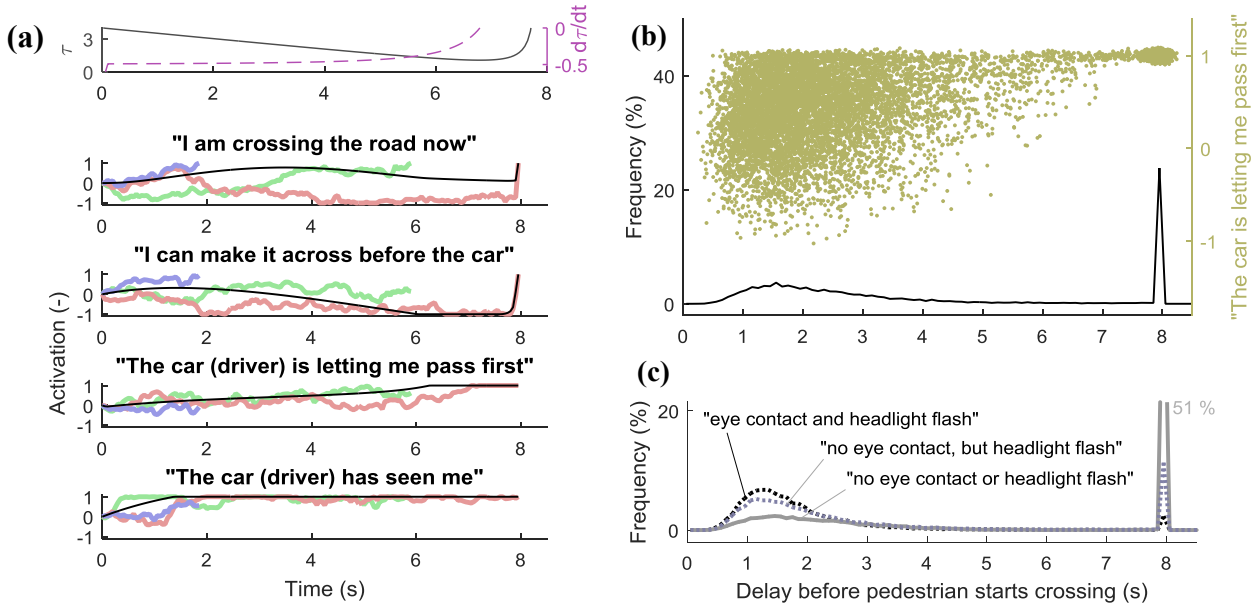


FIGURE 4 Example simulations of the pedestrian model. (a) Visual looming input (top panel) and resulting model activations. The black activation traces show a model simulation without noise, and the lighter colored activation traces show three example simulations with noise. (b) The black line shows the probability distribution of delay times for the model pedestrian's decision to cross the road. The light brown dots show the activation of the "the car is letting me pass first" unit at the time of crossing decision, as a function of the decision delay time. (c) Distribution of delay times for variations of the same scenario.

Reproducing safety-critical take-over behavior

Figure 5(a) shows four example time histories generated by the take-over driver model, all with TTC = 4 s and intermediate ($A = 0.5$) initial awareness of the adjacent lane being empty. These simulations differ only in the initial seed of the random noise in the decision processes, but it is notable that the model can nevertheless produce a range of qualitatively different sequences of behavior, both with and without one or more long or short glances away from the road ahead, and both with and without some degree of braking before a lane change is initiated (terminating the simulation).

Figure 5(b) shows the model-predicted frequency of different avoidance maneuvers, as a function of TTC at obstacle appearance, for a case with intermediate ($A = 0.5$) initial awareness of the adjacent lane being empty, as a possible emulation of a typical manual driving case. The obtained model distributions, and specifically the frequencies marked by vertical lines in the figure, align well with the pattern of driver responses reported in (*17*) for the manual driving condition: All human drivers (8/8; 100 %) responded with steering in the TTC = 7 s condition, whereas in the TTC = 5 s condition, also some combined braking and steering was observed (1/5 drivers; 20 %).

Figure 5(c) shows that without any initial awareness of the adjacent lane ($A = 0$), the probability of braking increases. This was also reported in (*17*) for the AV condition. Specifically, the model frequencies at the initial TTCs marked with vertical lines (4 s and 3.5 s) align qualitatively with the human responses in (*17*) for the experimental conditions with take-over

request at 7 s and 5 s TTC: A near-50/50 division between steering only and combined steering/braking was observed in the 7 s condition, whereas in the 5 s condition combined steering/braking rose to more clearly being the most common maneuver, followed by steering only, and some observations of pure braking. In other words, overall, the model simulations suggest the possibility that the take-over process in this scenario can be accounted for as a combination of reduced situational awareness and an urgency-dependent time delay, longer for the less critical scenario ($7\ \mathrm{s} - 4\ \mathrm{s} = 3\ \mathrm{s}$, compared to $5\ \mathrm{s} - 3.5\ \mathrm{s} = 1.5\ \mathrm{s}$) . This aligns with the report in (*17*) that hands-on-wheel and eyes-on-road delay times were longer in the less critical 7 s take-over scenario.

Figure 5(b) and (c) also show simulated crash frequencies. The model's braking alone was typically not enough to avoid collision, whereas in (*17*) seemingly no collisions occurred. This may indicate some problem with the tuning of the braking model. If the model applied steering avoidance, avoidance was considered successful as long as the time margin before collision, given speed and acceleration at time of steering initiation, was above 1 s.

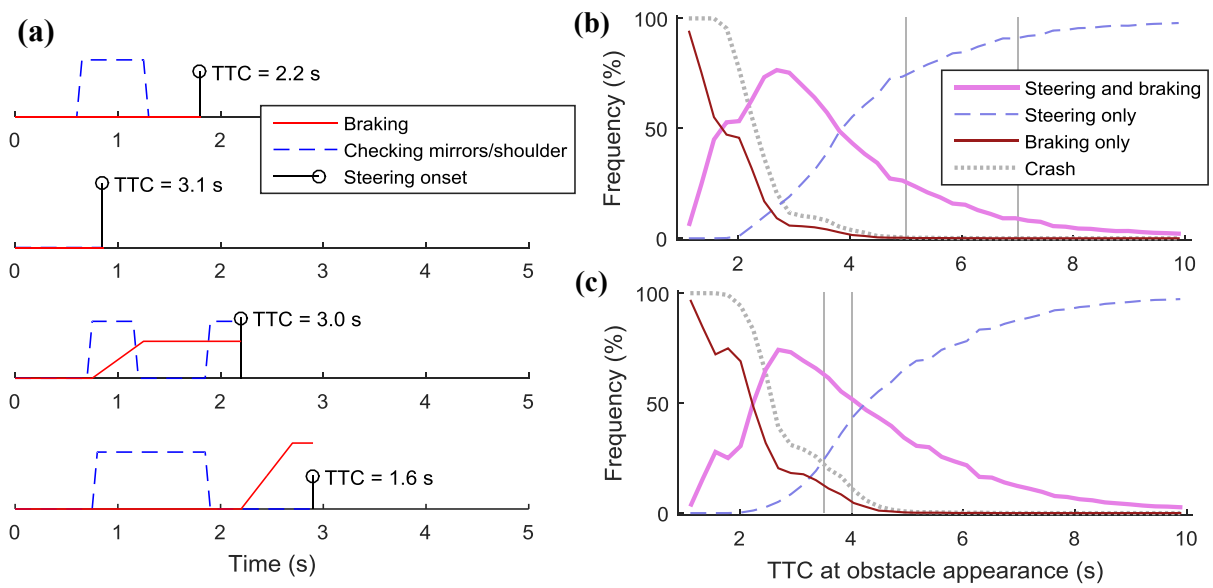


FIGURE 5 Example simulations of the hand-over model. (a) Typical model behavior time histories, all for the same initial condition. (b) Frequency of avoidance maneuvers performed by the model when run in a way so as to emulate manual driving (intermediate initial awareness of the adjacent lane), and resulting crash frequency, as a function of TTC at the time of obstacle appearance. The vertical lines indicate points of correspondence with (*17*), discussed in the text. (c) As panel b, but emulating a transition out of automated driving (with zero initial awareness of the adjacent lane).

MODEL APPLICATIONS IN VIRTUAL TESTING AND TUNING

This section will give examples of how the proposed models could be put to applied use in virtual, testing of AVs. It should be emphasized that since the models have yet to be fully validated and parameterized, the actual results given below are preliminary at best; especially for the transition scenario where the model is both more complex and showed signs of possible limitations in its braking behavior. The main aim here is to illustrate the *type* of results that can be obtained.

Optimizing AV traffic flow at a pedestrian crossing

If an AV intends to yield to a pedestrian, how should it behave so as to help the pedestrian make the crossing decision as quickly as possible? To answer this question, it was assumed that once the pedestrian starts walking, the AV adapts acceleration to pass behind the pedestrian, and then accelerates at 1 m/s^2 back up to the initial 50 km/h (31 mph). The time lost in the interaction was then calculated by comparing the distance travelled before regaining the initial speed, to the distance that would have been travelled in the same amount of time at constant speed, i.e., as if there had been no pedestrian.

Figure 6 shows, across a range of pedestrian appearance TTCs, (i) the model's predicted effect on time loss, of applying more than the deceleration needed to stop exactly at the zebra crossing, i.e., rising above the curved edge of the gray shaded area in each panel, and (ii) the effect on time loss of external AV-human communication. Panel (a) shows a "minimal communication" case where the AV does not give any explicit external signs of being aware of the pedestrian (i.e., nothing to replace eye contact) nor any signs, besides deceleration, of its intentions to yield, and panel (b) is a case where the AV does show such explicit external signs.

Perhaps counter to what one might expect, the model predicts that increased deceleration can save considerable time. This is especially true for the AV without external communication means, where for some TTCs (e.g. TTC = 4.5 s) an extra 0.5 m/s^2 of deceleration can change a 12 s time loss to near zero. The reason this happens in the model is that the pedestrian's judgment of whether the car will stop before the crossing (i.e., of whether $\dot{t} > -0.5$) is difficult when the margin is narrow (i.e., when $\dot{t} \approx -0.5$), and the exaggerated deceleration makes this perceptual decision much easier and quicker.

The model predicts that time savings can also be achieved by letting the AV communicate its situational awareness and intentions (assuming that this can be achieved by some external HMI that is understandable to the pedestrian). The benefits of exaggerated decelerations are still present, but less pronounced, except for the lowest TTCs, where it starts being questionable whether the AV should at all yield to a waiting pedestrian, considering the comfort and safety of AV occupants and any following vehicles.

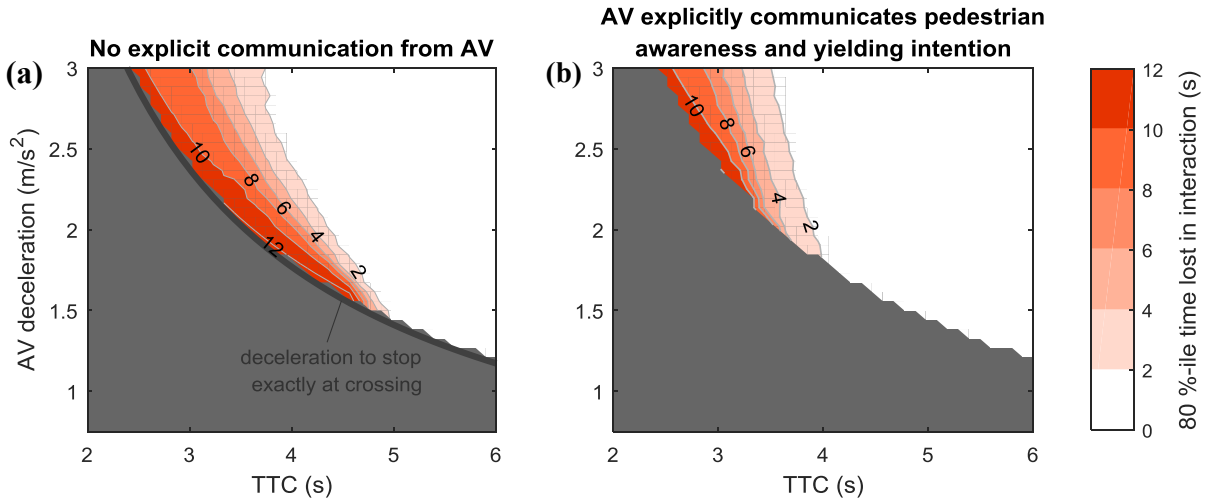


FIGURE 6 Model estimation of time lost for the AV in the pedestrian interaction, as a function of TTC at time of pedestrian presentation, and AV deceleration applied at the same moment.

Minimising crash risk in time-critical AV hand-overs

It has been suggested that detriments in performance during AV-human hand-overs are the result of a loss of situational awareness (*8*), with a recommendation that designers need to give deep consideration as to how best to facilitiate the process of bringing drivers back "into the loop" during transitions from automation (*18*). Exactly how to do so is of course a non-trivial matter, but here it is examined what benefit it might have in the studied scenario *if* a solution could be devised ensuring that the driver quickly knows, without having to look away from the road ahead, that it is possible to change lane, emulated here by initializing the "it is safe to change lane now" decision at full certainty. It is also examined here whether any such benefit might interact, according to the model, with emergency braking applied by the AV itself.

As shown in Figure 7, the model predicts a range of initial TTCs between 1.5 s and 3.5 s (remember that factoring in the take-over process, this might correspond to a considerably higher obstacle appearance TTC) where benefits of increased situational awareness may exist. For example, even if the AV decelerates with 6 m/s^2 in a TTC = 2 s scenario, immediate driver awareness of the feasibility of steering avoidance changes the model-predicted crash risk from about 80 % to about 20 %. In fact, according to the model, this low crash risk remains nearly unchanged even if the AV does not apply any emergency braking at all, which seems beneficial from a perspective of minimising risk of being struck from behind.

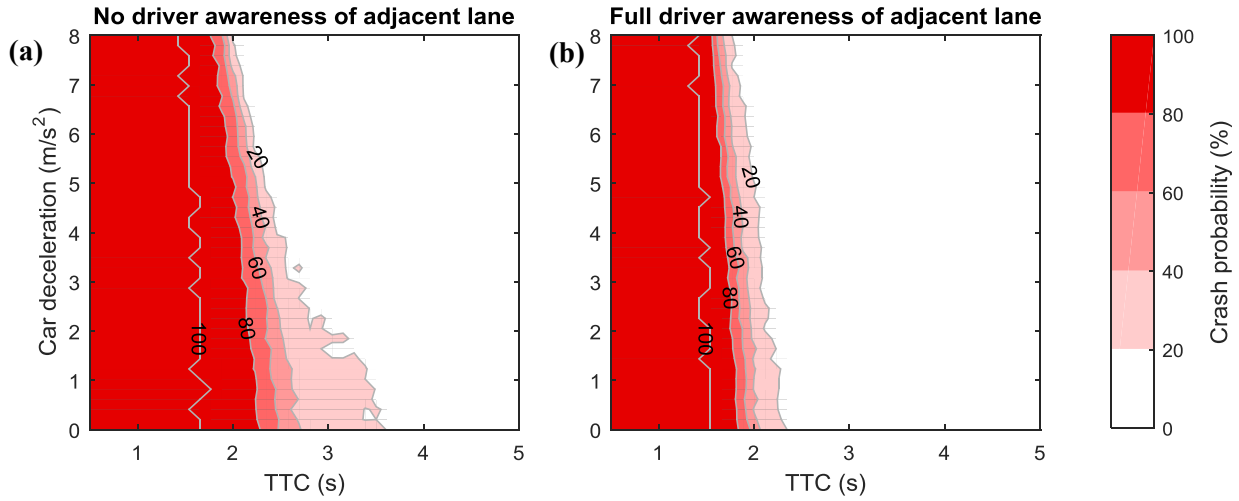


FIGURE 7 Model estimated effect of driver awareness of possibility to change lane, and AV emergency braking, on crash risk in the studied take-over scenario.

DISCUSSION AND CONCLUSIONS

This paper has presented first draft versions of novel models of human decision-making and behavior in two scenarios relevant to development and testing of AVs. Both models seem to offer useful developments beyond what is currently available in the literature for these scenarios.

Pedestrian road-crossing has been modeled here as an interrelated set of perceptual decisions about oncoming vehicle movement and intentions. The model has illustrated how a car that applies exactly the right amount of deceleration for stopping might make it difficult for a waiting pedestrian to decide on crossing before the car has come to a complete stop, thus explaining empirical observations of such late pedestrian crossing decisions (in (*16*), but anecdotally also in many people's everyday experience). In contrast, existing microsimulation pedestrian models tend to emphasise the question of *whether* a pedestrian accepts a certain time gap between passing cars (*22*), and less on *when* the pedestrian accepts it, as a function of exact movement and communicative acts of a vehicle, making the model proposed here a potentially useful addition to the toolkit for virtual testing and tuning of AV behavior.

With regards to the other scenario targeted here, focusing on human avoidance maneuvering close to a potential rear-end crash, existing driver models for such situations typically decide on braking and/or steering by drawing from fixed, situation-independent probability distributions (*23*), whereas it is well known that human avoidance responses vary significantly with not least urgency (*16*, *24*), in ways similar to what is exhibited by the model proposed here. The present work also provides some first steps towards modeling the actual process of taking over from an AV, suggesting that, at least in the presently studied scenario, the take-over could be thought of as an urgency-dependent time delay, combined with a low initial situational awareness.

Indeed, a striking feature of both models is that they provide concrete operationalizations of the concept of situational awareness in the given scenarios, in terms of neurobiologically plausible models of perceptual decisions. This may be a useful complement to existing computational models of situational awareness; e.g., (*13*). Overall, the results presented here seem promising for the general modeling approach of interconnecting several perception and action decision accumulators.

Related but different models have been developed from connectionist and dynamical systems perspectives (*14*, *25*), as well as in the field of naturalistic decision-making (*15*). One useful next step would be to benchmark the present modeling framework against these existing ones, and ideally also against more conceptually different alternatives based on cognitive architectures such as ACT-R (*26*), or from robotics (*27*).

An obvious next step will be to properly test and parameterize the models on detailed data sets of human behavior in the studied scenarios. If such validation tests are successful, the (so far very preliminary) virtual testing results presented here suggest that models at this level of detail, capable of predicting full AV-human interaction sequences unfolding in dynamic traffic scenarios, could be of great use when testing, validating, and optimising future automated transport systems.

ACKNOWLEDGMENT

This work is part of the interACT project, funded by the European Union's Horizon 2020 research & innovation programme under grant agreement 723395. It reflects the authors' views and the European Commission is not responsible for any use that may be made of the information herein.

REFERENCES

1. Fagnant, D. J., and K. Kockelman. Preparing a nation for autonomous vehicles: opportunities , barriers and policy recommendations. *Transportation Research Part A*, Vol. 77, 2015, pp. 167–181. http://dx.doi.org/10.1016/j.tra.2015.04.003.
2. Kalra, N., and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A*, Vol. 94, 2016, pp. 182–193. http://dx.doi.org/10.1016/j.tra.2016.09.010.
3. Chen, B., D. Zhao, and H. Peng. Evaluation of Automated Vehicles Encountering Pedestrians at Unsignalized Crossings. https://arxiv.org/abs/1702.00785.
4. Waymo. *On the road to fully self driving: Waymo safety report*. https://waymo.com/safetyreport/, accessed 2017-10-16, 2017.
5. Clamann, M., M. Aubert, and M. L. Cummings. Evaluation of Vehicle-to-Pedestrian Communication Displays for Autonomous Vehicles. Presented at *Transportation Research Board 96th Annual Meeting*, 2017.
6. Kesting, A., M. Treiber, M. Scho, et al. Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C*, Vol. 16, 2008, pp. 668–683. http://dx.doi.org/10.1016/j.trc.2007.12.004.
7. Zeeb, K., A. Buchner, and M. Schrauf. What determines the take-over time? An integrated model approach of driver take-over after automated driving. *Accident Analysis and Prevention*, Vol. 78, 2015, pp. 212–221. http://dx.doi.org/10.1016/j.aap.2015.02.023.
8. Kaber, D. B., and M. R. Endsley. The effects of level of automation and adaptive automation on human performance , situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, Vol. 5, No. 2, 2004, pp. 113–153. http://dx.doi.org/10.1080/1463922021000054335.
9. Purcell, B. A., R. P. Heitz, J. Y. Cohen, et al. Neurally Constrained Modeling of Perceptual Decision Making. *Psychological Review*, Vol. 117, No. 4, 2010, pp. 1113–1143.
10. Ratcliff, R., and D. Strayer. Modeling simple driving tasks with a one-boundary diffusion model. *Psychon Bull Rev*, 2013. http://dx.doi.org/10.3758/s13423-013-0541-x.
11. Markkula, G., J. Engström, J. Lodin, et al. A farewell to brake reaction times? Kinematics-dependent brake response in naturalistic rear-end emergencies. *Accident Analysis & Prevention*, Vol. 95, 2016, pp. 209–226.
12. Svärd, M., G. Markkula, J. Engström, et al. A quantitative driver model of pre-crash brake onset and

control. Presented at *Human Factors and Ergonomics Society 61st Annual Meeting*, 2017.

13. McCarley, J. S., C. D. Wickens, J. Goh, et al. A Computational Model of Attention/Situation Awareness. Presented at *Human Factors and Ergonomics Society 46th Annual Meeting*, 2002.
14. Roe, R. M., J. R. Busemeyer, and J. T. Townsend. Multialternative decision field theory: a dynamic connectionist model of decision making. *Psychological Review*, Vol. 108, No. 2, 2001, pp. 370–392.
15. Patterson, R., L. Fournier, B. Pierce, et al. Modeling the Dynamics of Recognition-Primed Decision Making. Presented at *The 9th International Conference on Naturalistic Decision Making*, 2009.
16. Schneemann, F., and I. Gohl. Analyzing Driver-Pedestrian Interaction at Crosswalk : A Contribution to Autonomous Driving in Urban Environments. Presented at *IEEE Intelligent Vehicles Symposium*, 2016.
17. Gold, C., D. Damböck, L. Lorenz, et al. "Take over!" How long does it take to get the driver back into the loop? Presented at *Human Factors and Ergonomics Society 57th Annual Meeting*, 2013.
18. Merat, N., A. H. Jamson, F. C. H. Lai, et al. Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation Research Part F: Psychology and Behaviour*, Vol. 27, 2014, pp. 274–282. http://dx.doi.org/10.1016/j.trf.2014.09.005.
19. Lee, D. N. A theory of visual control of braking based on information about time-to-collision. *Perception*, Vol. 5, No. 4, 1976, pp. 437–459.
20. Bogacz, R., E. Brown, J. Moehlis, et al. The Physics of Optimal Decision Making : A Formal Analysis of Models of Performance in Two-Alternative Forced-Choice Tasks. *Psychological Review*, Vol. 113, No. 4, 2006, pp. 700–765. http://dx.doi.org/10.1037/0033-295X.113.4.700.
21. Friston, K., R. A. Adams, L. Perrinet, et al. Perceptions as hypotheses : saccades as experiments. *Frontiers in Psychology*, Vol. 3, 2012. http://dx.doi.org/10.3389/fpsyg.2012.00151.
22. Lu, L., G. Ren, W. Wang, et al. A cellular automaton simulation model for pedestrian and vehicle interaction behaviors at unsignalized mid-block crosswalks. *Accident Analysis and Prevention*, Vol. 95, 2016, pp. 425–437.
23. van Auken, R. M., J. W. Zelner, D. P. Chiang, et al. *Advanced Crash Avoidance Technologies (ACAT) Program - Final Report of the Honda-DRI Team*. Publication DOT HS 811 454. USDOT/NHTSA, 2011.
24. Lee, S. E., E. Llaneras, S. Klauer, et al. *Analyses of rear-end crashes and near-crashes in the 100-car naturalistic driving study to support rear-signaling countermeasure development*. Publication DOT HS 810 846. USDOT/NHTSA, 2007.
25. Pellecchia, A., C. Igel, J. Edelbrunner, et al. Making Driver Modeling Attractive. *IEEE Intelligent Systems*, Vol. 20, No. 2, 2005, pp. 8–12. http://dx.doi.org/10.1109/MIS.2005.31.
26. Keller, J., C. Lebiere, R. Shay, et al. Cockpit system situational awareness modeling tool. In *Human Performance, Situation Awareness and Automation: Current Research and Trends: HPSAA II* (D. A. Vincenzi, M. Mouloua, and P. A. Hancock, eds.), Psychology Press, pp. 66–71.
27. Bai, H., S. Cai, N. Ye, et al. Intention-Aware Online POMDP Planning for Autonomous Driving in a Crowd. Presented at *IEEE International Conference on Robotics and Automation*, 2015.