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

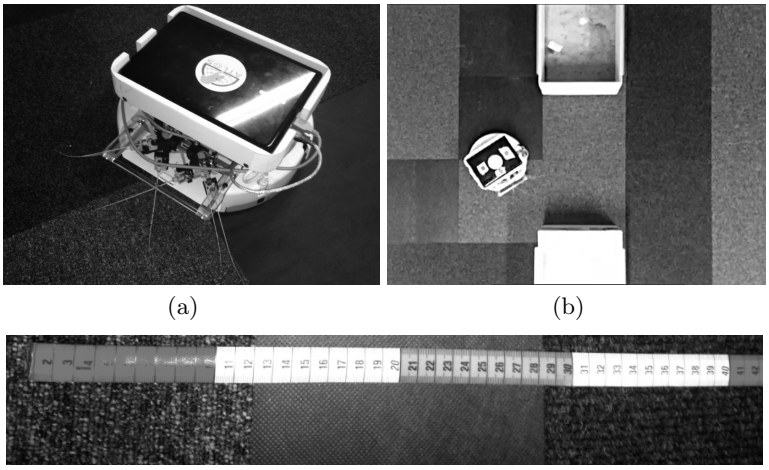
Abstract. CrunchBot is a robot platform for developing models of tactile perception and navigation. We present the architecture of CrunchBot, and show why tactile navigation is difficult. We give novel real-time performance results from components of a tactile navigation system and a description of how 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.

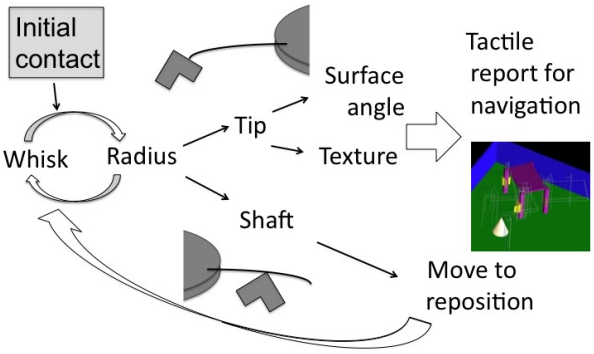


Fig. 2. Framework overview for whisker-based perception and navigation

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.)

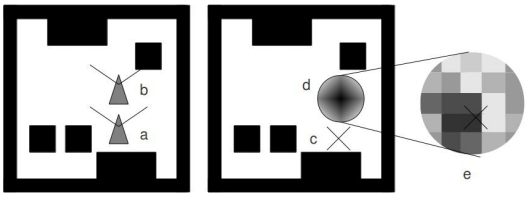


Fig. 3. Foveal grid based navigation.

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

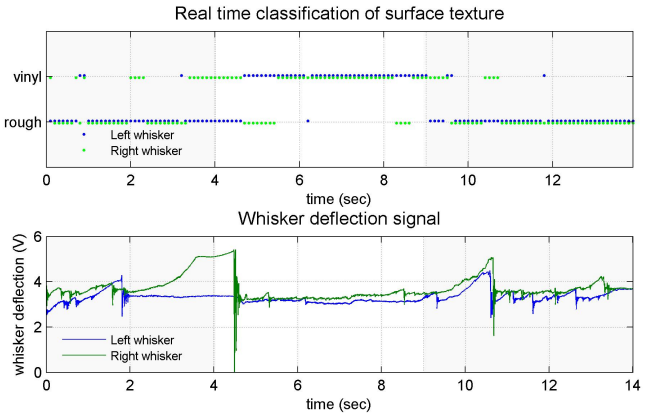


Fig. 4. Real-time classification of surface texture.
The top panel shows the classification results for the left (blue) and right (green) whiskers traversing a course over rough carpet (shaded regions), then vinyl (unshaded region) and back to rough carpet. The bottom panel shows the deflections associated 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.

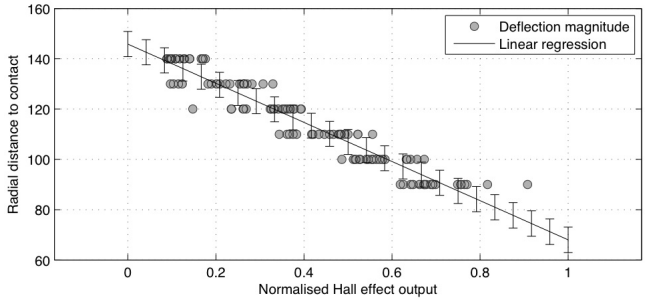


Fig. 5. Peak deflection magnitude for contacts along the shaft of the whisker. Standard error for the regression is 4.98mm

	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].

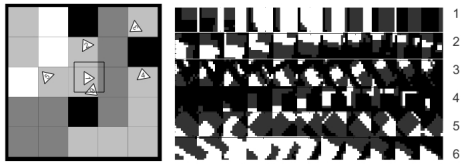
Although introduced purely for computational reasons, both the use of likelihood hash table and the mode-centred Gaussian posterior approximations are similar to, and loosely inspired by, biological models of hippocampal navigation [6]. However use of a single mode-centered Gaussian has lead to problems 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).

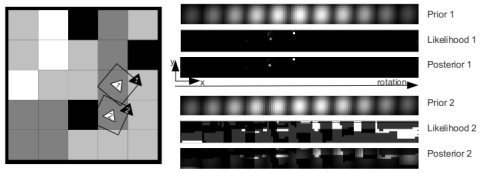
References

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.

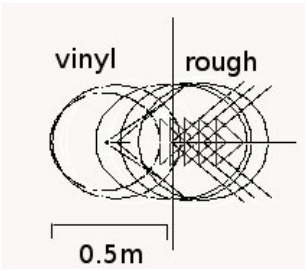
7. C. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. Submitted to TAROS 2011.
8. C. W. Fox, M. Evans, M. J. Pearson, and T. J. Prescott. Towards temporal inference for shape recognition from whiskers. In *Proc TAROS*, pages 226 – 233, 2008.
9. V. Gopal and M. 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.