KTH Computer Science
and Communication

Probabilistic Tracking of Multiple Rodent Whiskers in Monocular Video Sequences <FIRST DRAFT>

A small step towards cheap, reliable, and non-intrusive automatic tracking of whiskers.

JIM HOLMSTRÖM
JIMHO@KTH.SE
EMIL LUNDBERG
EMLUN@KTH.SE

SA104X Examensarbete inom teknisk fysik, grundnivå
Bachelor's Thesis at CSC, KTH
Supervisor: Prof. Örjan Ekeberg
Examiner: ???

TRITA xxx yyyy-nn

Abstract

The interest in studying rodent whiskers has recently seen a significant increase, particularly in the field of neurophysiology. As a result, there is a need for automatic tracking of whisker movements. Currently available commercial solutions either are extremely expensive, restrict the experiment setup, or fail in the presence of *clutter* or occlusion.

This thesis proposes a proof-of-concept implementation of a probabilistic tracking system. This solution uses a technique known as the *Particle Filter* to propagate a whisker model between frames of high speed video. In each frame, the next state of the model is predicted by querying a pretrained database and filtering the results through the Particle Filter. The implementation is written in Python using NumPy and an SQLite3 database.

First, it successfully tracks multiple whiskers at once, even under clutter. Second, being a standalone program operating on pre-recorded video, it does not notably restrict the experiment.

Keywords: Tracking, Multiple, Whisker, Particle Filter, Transition Database, Model Evaluation, Proof-of-Concept

Referat

Statistisk Följning av Multipla Morrhår på Gnagare i Video.

Contents

1 Introduction **1**
1.1 Background . . . 1
1.2 Difficulties . . . 1
1.3 Contributions of this thesis . . . 2

2 Related Work **3**
2.1 Other whisker tracking systems . . . 3
2.1.1 Unsupervised.. . . . 3
2.1.2 MoCap . . . 3
2.2 Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences<ref> . . . 3

3 Definitions **5**
3.1 Images and videos . . . 5
3.1.1 Grayscale image . . . 5
3.1.2 Video . . . 5
3.2 Image Processing . . . 6
3.3 States, hypotheses and estimates . . . 6

4 Theory **7**
4.1 Introduction . . . 7
4.2 Markov processes . . . 7
4.3 The Hidden Markov Model . . . 7
4.4 The Curse of Dimensionality . . . 8
4.4.1 Overcoming the Curse . . . 9
4.5 The Particle Filter . . . 9
4.5.1 The Particle Filter algorithm . . . 10
4.6 Sensory Cues . . . 11
4.7 Model . . . 11
4.7.1 L_p . . . 12
4.7.2 Polynomial $a_3\omega^3 + a_2\omega^2 + a_1\omega$ . . . 12
4.7.3 Sinus serie$\sum a_n \sin(\frac{2\pi n}{L}\omega)$ . . . 13
4.7.4 Model Summary . . . 13

5 Algorithms and Implementations **15**
5.1 The particle filter 15
5.1.1 Initilization x_0 15
5.1.2 Goodness $\kappa p(x_t|I_t)$ 16
5.1.3 Sampling $p(x_t|x_{t-1})$ 16
5.2 The state transition database 16
5.3 Real data 16
5.3.1 Pre-processing 16

6 Results **17**

7 Analysis and Discussion **19**

Bibliography **21**

Chapter 1

Introduction

1.1 Background

With the ever increasing power and mobility of computers, computer vision has recently seen a large increase in interest. Encompassing problems such as classification, recognition, perception and tracking, application of the discipline could make many tasks easier and more efficient.

In particular, biology and neuroscience researchers are interested in tracking the movements and posture <ref:(2)> of animals or parts of animals. One such field aims to study the movements of rodent whiskers. However, most whisker tracking software available today suffers a few fundamental flaws. They are either so expensive that not even well funded laboratories feel they can afford them or have problems tracking multiple whiskers at once, often requiring removal of almost all whiskers. Some higher precision systems impose other restrictions on the experiment, such as restraining the animal or attaching motion capture markers to the whiskers [2]. Such restrictions may cause systematic errors and not give a representative view of how the animal naturally uses its whiskers.

1.2 Difficulties

In general, the main difficulty in tracking and localization is to separate the tracked object from clutter and occlusion. In the case of whisker tracking, the dominant problems are that whiskers often overlap and vary in size, and the relatively low spatial and temporal resolution in highspeed video<ref:(2)> results in subpixel whiskers and motion blur.

In 2001, Hedvig Sidenbladh investigated probabilistic methods for tracking three-dimensional human motion in monocular video [4]. Many of the problems inherent in computer vision were regarded, and ... «continue»

1.3 Contributions of this thesis

This thesis proposes a probabilistic tracking method using «continue»

Chapter 2

Related Work

2.1 Other whisker tracking systems

2.1.1 Unsupervised..

2.1.2 MoCap

2.2 Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences<ref>

Chapter 3

Definitions

3.1 Images and videos

3.1.1 Grayscale image

A grayscale *image* can be defined as a function

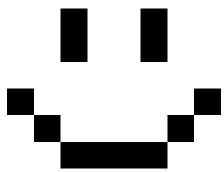
$$\begin{array}{rcl} I : \mathbb{N}^2 & \rightarrow & \mathbb{R}^+ \\ I : \text{position} & \rightarrow & \text{intensity}. \end{array} \tag{3.1}$$

It can also be identified with $\mathbb{N}^2 \times \mathbb{R}^+$ as the tuple $\langle \text{position}, \text{intensity} \rangle$. The *image space* is denoted as $\mathcal{I}$ in this thesis.

In a computer an image is represented as a integer matrix, often 8 bit integers.[1]

Example 1.

$$\begin{pmatrix} 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 0 & 255 & 255 & 0 & 255 & 255 & 255 \\ 255 & 255 & 255 & 0 & 255 & 255 & 0 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 0 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 255 \\ 255 & 255 & 0 & 255 & 255 & 255 & 255 & 0 & 255 & 255 \\ 255 & 255 & 255 & 0 & 0 & 0 & 0 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow$$



3.1.2 Video

A *video* is a function mapping an integer to an image:

$$\text{video} : \mathbb{N} \rightarrow \mathcal{I}. \tag{3.2}$$

[1]Integers in the range $[0, 255]$.

3.2 Image Processing

Definition 1. The rendering function R takes a hypothesis x and renders an image with a resemblance of how a real whisker would have looked like having the same underlying model and parameters as x.

$$\begin{aligned} R : \mathcal{X} &\to \mathcal{I} \\ x &\mapsto \text{Render}(x) \end{aligned} \tag{3.3}$$

Definition 2. The image transformation ϕ takes an image I and returns a transformed image of the same size as I.

$$\begin{aligned} \phi : \mathcal{I} &\to \mathcal{I} \\ I &\mapsto \text{Transform}(I) \end{aligned} \tag{3.4}$$

3.3 States, hypotheses and estimates

A system is said to have a *state*. The state is some quantity that defines the qualities of the system.

State The state of a system is denoted Z. When time is relevant, the state at time t is denoted Z_t.

State space The set $\mathcal{Z}$ of all possible states, $Z \in \mathcal{Z}$.

Hypothesis A guess x at the state Z of a system.

Hypothesis space The set $\mathcal{X}$ of all possible hypotheses, $x \in \mathcal{X}$. In general, $\mathcal{X} \neq \mathcal{Z}$ since most models are simplifications of the system.

Estimate The hypothesis x^* we believe approximates Z best.

Observation In general, it is not possible to directly record the state Z of a system.[2] We instead get an *observation* I of the state.

Degrees of Freedom The number of adjustable parameters in a model, often abbreviated DOF.

Note that all of the above depend on the model used.

[2]If it were, there would be no need for tracking.

Chapter 4

Theory

4.1 Introduction

The core of the tracking engine is a technique known as the *particle filter*. The particle filter is a kind of Bayesian filtering where one uses discrete hypotheses, also known as *particles*, to approximate continuous PDF [5]. It builds upon the theory of *Markov processes* and the *hidden Markov model*.

4.2 Markov processes

A Markov process is a special case of a stochastic process. For a Markov process, the next state depends only on the present state and not on past states. For this reason, a Markov process is often said to be "forgetful".

In mathematical terms, a Markov process satisfies the following:

$$p\left(Z_t|Z_{t-1} \wedge Z_{t-2} \wedge \cdots \wedge Z_0\right) = p\left(Z_t|Z_{t-1}\right), \tag{4.1}$$

$p\left(Z_t|Z_{t-1} \wedge Z_{t-2} \wedge \cdots \wedge Z_0\right)$ is the probability that the system will have state Z_t at time t, given that the previous states where $Z_{t-1}, Z_{t-2}, \ldots, Z_0$.

«figure of markov process only»

4.3 The Hidden Markov Model

The working principle of the particle filter is based on the *hidden Markov model* (HMM). A HMM describes a Markov process where we cannot measure the state directly - it is "hidden"[3]. Instead we obtain an *observation* I[1] of the state. This *perception* is generally non-deterministic, so we need to denote it as $p(I_t|Z_t)$ which is the probability that we will observe I_t if the state is Z_t.

[1] In this thesis, the observation is always a grayscale *image*, therefore the observation is denoted I.

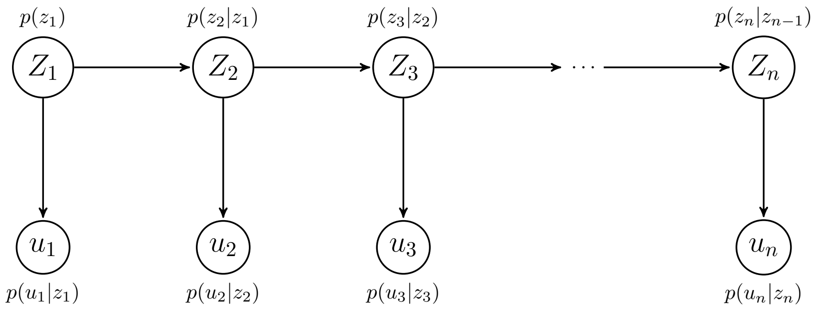


Figure 4.1. Schematic image of a hidden Markov model.

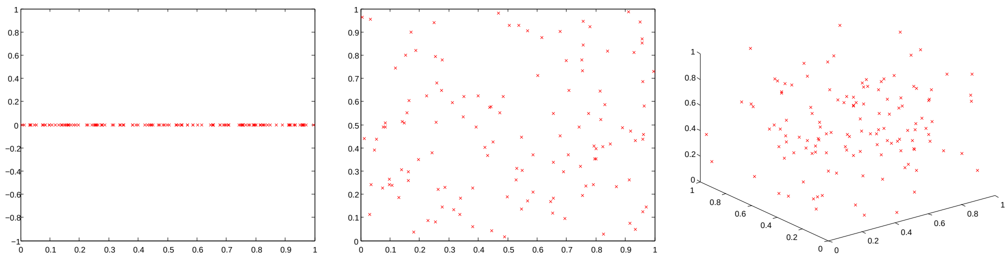


Figure 4.2. Plots of 128 scattered samples in 1, 2 and 3 dimensions, respectively.

4.4 The Curse of Dimensionality

A phenomenon that becomes apparent in high-dimensional spaces is the so-called "Curse of dimensionality" [3]. The problem is that the search volume grows exponentially with the number of dimensions. It originates from the fact that we need $\mathcal{O}(C^n)$ samples to obtain a sample density of C in a n-dimensional space.

The first consequence of this is that in order to approximate a high-dimensional function one needs orders of magnitude more samples.

The other drawback with high dimensional space is the large "borders" of the sample-set compared to lower dimensional space which results in orders of magnitude higher chance for an point one want to approximate to fall outside the sample-set and needs to be extrapolated instead of the better alternative of interpolation.

Example 2. Figure 4.2 shows 128 randomly scattered points in 1, 2 and 3 dimensions. Notice how the density decreases with increasing dimension.

Example 3. For a 16 DOF model one needs 10^{16} = 10 quadrillion datapoints to acquire a density of 10 samples per unit volume. Millions of gigabytes would be needed just to store the samples.

Example 4. In 2 dimensions it is sometimes feasible to use an exhaustive search. An example of this is the Hough transform [1], where the search is done through the $\rho\theta$ space of line responses on images.

4.4.1 Overcoming the Curse

One way to overcome the curse in the context of tracking is to perform a directed search. Let the search be in an n dimensional space with a grid of g grid lines in each direction.

1. Use the information about the most recent[2] location and assume that the tracked object cannot travel more than $R < g$ grid steps in one time step. This reduces the volume of the (discrete) search space from $\mathcal{O}(g^n)$ to $\mathcal{O}(R^n)$.

2. With prior knowledge of how the tracked objects move [3] we can direct our search to specific regions in the state space, depending on how probable it is for the tracked object to be located there. This reduces the size of the search space depending on how sure we are of the previous state.

4.5 The Particle Filter

One naïve way to compute the state Z_t would be to perform an exhaustive search in the state space, and select the state for which $p(\, I_t | Z_t \,)$ is maximised.<ref:COD>

The particle filter is a technique for reducing the search space<ref:COD>. It uses a finite set X_t of hypotheses to approximate the PDF $p(\, Z_t | Z_{t-1} \,)$ of a HMM. The hypotheses X_t are also refered to as *particles*, thereby the term "particle filter".

Figure 4.3 shows the principle of the particle filter working alongside a hidden Markov model. The following is the core function of the particle filter:

> *The particle filter attempts to approximate the PDF $p(\, Z_t | Z_{t-1} \,)$ as a set X_t of discrete hypotheses.*

More particles mean greater accuracy, since the PDF can then be approximated more closely. However, using many particles increses computational cost. Therefore the number of particles is an important trade-off. The particle filter employs a few tricks to *filter* the hypotheses, tending to keep probable ones and throwing improbable ones away, in order to intelligently reduce the number of particles needed for a good approximation. The filter works in four steps:

[2] In the Bayesian case, the most recent estimate

[3] Such as the state transition probabilites $p(\, Z_t | Z_{t-1} \,)$ in a HMM

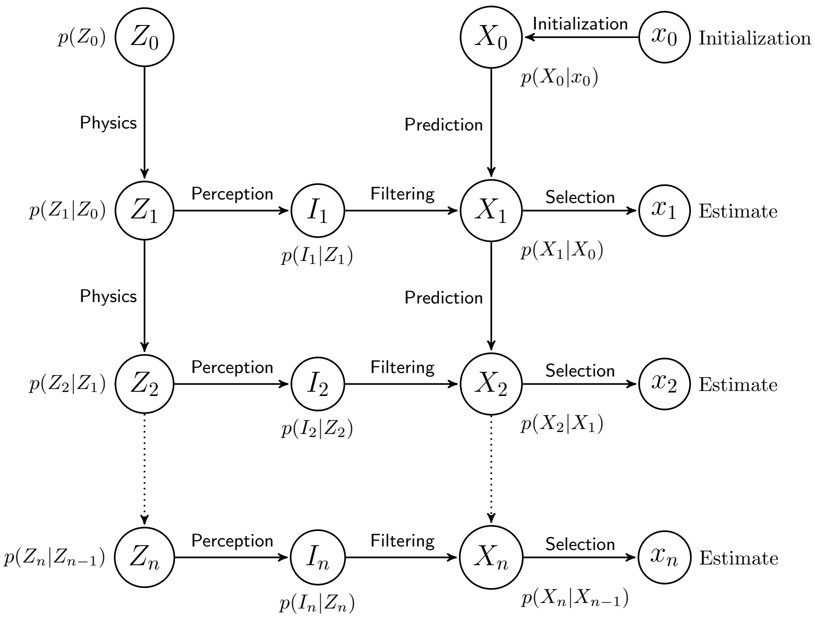


Figure 4.3. Schematic image of the particle filter alongside a HMM.

Prediction The hypotheses X_{t-1} are updated in the *prediction* step to an approximation $\bar{X}_t$ of $p(\,Z_t|Z_{t-1}\,)$. This is done by drawing new samples $p(\,x_t|x_{t-1}\,)$, for each x_{t-1} in X_{t-1}.

Perception By measuring the state of the system, we gain an *observation* $I_t \sim p(\,I_t|Z_t\,)$ of the state Z_t.

Filtering The observation I_t of the system is then used for filtering bad hypotheses out of $\bar{X}_t$. We draw samples X_t from $\bar{X}_t$ with probabilities given by $p(\,I_t\big|\bar{X}_t\,)$. The result will be a surjection, where X_t will be a subset of $\bar{X}_t$ where more probable hypotheses appear multiple times.<ref:problem without sigma> For this reason, this is also known as the *resampling* step. The set X_t is the *belief*, our approximation of $p(\,Z_t|Z_{t-1}\,)$.

Selection Finally, we produce a single hypothesis x_t from X_t as our *estimate* of the state Z_t. Assuming X_t is a good approximation of $p(\,Z_t|Z_{t-1}\,)$, and that $p(\,Z_t|Z_{t-1}\,)$ is unimodal, the means value of X_t is a good estimate since it approximates the expectation of $p(\,Z_t|Z_{t-1}\,)$. If $p(\,Z_t|Z_{t-1}\,)$ is multimodal, however, the mean could be a bad estimate since the expectation may be very improbable.

4.5.1 The Particle Filter algorithm

PARTICLE-FILTER(X_{t-1}, I_t)

$\bar{X}_t \leftarrow \emptyset$
for each $x_{t-1} \in X_{t-1}$
    do
        $x_t \leftarrow$ PREDICT(x_{t-1})
        $w \leftarrow$ IMPORTANCE(x_t, I_t)
        Append $\langle x_t, w \rangle$ to $\bar{X}_t$

$X_t \leftarrow \emptyset$
while $|X_t| < \left|\bar{X}_t\right|$
    do
        Take $\langle x_t, w \rangle$ from $\bar{X}_t$ with probability $\propto w$
        Append x_t to X_t
return X_t

Table 4.1. The particle filter algorithm.

Table 4.1 shows the particle filter algorithm. Note that the functions PREDICT and IMPORTANCE are unspecified - they are problem specific. They correspond to the PDF $p(x_t|x_{t-1})$ and $p(I_t|x_t)$, respectively.

In this thesis, the real x_{t-1} is not known. Rather x_{t-1} is estimated with a set X_{t-1} of N particles.

4.6 Sensory Cues

The biggest problem with computer vision is that computers do not have vision, only a data input device in the form of a camera.

!!Is a image transformation ϕ !!that renders(depicts?) one property of the !! image like intensity, edges or ridges.

<ref to machine vs biological comparison study>

<image showing the use of ϕ>

4.7 Model

In the following chapter we will formalize the whisker and discuss our model selection<ref:encyclo> process.

First of we formalize the whisker by the definition

Definition 3. Let

$$\begin{aligned} \text{whisker} : \mathbb{R}^+ &\to \mathbb{R}^2 \\ \omega &\mapsto \text{whisker}(\omega) \end{aligned} \tag{4.2}$$

The idea is to find a function whisker* that approximates the whisker. The following class of functions are available for consideration:

It must be able to sufficiently approximate the whisker

$$\forall \text{whisker} \exists \text{whisker}^* : \text{whisker} \approx \text{whisker}^* \tag{4.3}$$

that is, the model must be able to sufficiently fill the range of the whisker function.

Example 5. A straight line will not suffice since the whisker is generally curved and straight lines can not fill that partition of the space.

Additionally it must of course be C_1 and have finitely many parameters. We might also assume that a whisker has one fix point at the origin, which we choose as origo, the boundary condition is equivalent to

$$\text{whisker}^*(0) = \bar{0} \tag{4.4}$$

this is not the case in the real data, generally we have time dependency

$$\text{whisker}(0, t) \neq \bar{0} \tag{4.5}$$

«restate the following sentence» since rodents seems to have the ability to move around their whiskers root on the chins surface.

We are for all cases assuming that the whisker thickness is defined by

$$d(\omega) = \begin{cases} D, & \omega = 0 \\ D - \frac{D\omega}{L}, & \omega < L \\ 0, & \omega \geq L \end{cases} \tag{4.6}$$

where $D > 0$ is the thickness at the root and $L > 0$ is the total length of the whisker.

4.7.1 L_p

«»

4.7.2 Polynomial $a_3\omega^3 + a_2\omega^2 + a_1\omega$

Our first and simplest candidate is the third degree polynom $a_3\omega^3 + a_2\omega^2 + a_1\omega$.

Parametrically define as

$$(\omega, a_3\omega^3 + a_2\omega^2 + a_1\omega) \tag{4.7}$$

This chose can be justified by the theory of beams under small deformations in the area of strength of materials, after all a whisker is not to far away from a beamer.<ref: strength of materials> The two main assumption for this to hold is a linear elastic material and small deformations which ...

<example images compared with real whiskers>

«Lp analytical»

4.7.3 Sinus serie$\sum a_n \sin(\frac{2\pi n}{L}\omega)$

Another promising candidate is the sinus series. The Fourier transform becomes $\sum a_n \sin(\frac{2\pi n}{L}\omega)$ when considering (4.4). One can choose just a few n which can be viewed as a few different waves propagating trough the whisker propelled by the whisking.

«Lp analytical»

====================== The equation of elastic line ... [Grundläggande Hållfasthetslära - Hans Lundh p94 (7.6)]

The force that comes from the head moving on the base of the whisker is just sucked up by the Boundaryvalues and it will still be valid assumptions for the elasticline to hold.

Under just a few assumptions that the material is linear elastic and the deformations are small we have the ... ======================

4.7.4 Model Summary

It is hard to analytically justify the chose of whisker* since we do not have an analytic model for the whisker.

Theorem 1. Let f be a positive Riemann function with compact support, then

$$\underset{\bar{e}}{argmax}\left(\sum_{\Omega} f(\bar{x})f(\bar{x}-\bar{e})\right) = 0 \tag{4.8}$$

Proof. Firstly define the window function as

$$W_a^b(x) = \begin{cases} 0, & x < a \\ 1, & a =< x =< b \\ 0, & x > b \end{cases} \tag{4.9}$$

Multiplication

$$(W_a^b W_c^d)(x) = W_{\max(a,c)}^{\min(b,d)}(x) \tag{4.10}$$

Translation

$$W_a^b(x-e) = W_{a+e}^{b+e}(x) \tag{4.11}$$

Integration

$$\sum W_a^b(x)dx = \Theta(b-a) \tag{4.12}$$

$$\begin{gathered} \underset{e}{argmax}\left(\sum W_a^b(x)W_a^b(x-e)\right) = \\ \underset{e}{argmax}\left(\sum W_a^b(x)W_{a+e}^{b+e}(x)\right) = \\ \underset{e}{argmax}\left(\sum W_{\max(a,a+e)}^{\min(b,b+e)}(x)\right) = \\ \underset{e}{argmax}\left(\Theta(\min(b,b+e)-\max(a,a+e))\right) = \\ 0 \end{gathered} \tag{4.13}$$

This trivially holds with superposition of windows, since all windows will scale and translate the same way. With a finite support $e = 0$ is the only solution. Additionaly this also hold in higher finite dimensions since we can just repeat the process for one dimension at a time.

All riemann functions can be written as a superposition of windows like this

$$f(x) = \sum c_i W_{a_i}^{b_i}(x) \tag{4.14}$$

$\therefore$ Each riemann function f with finite support will have a $e = 0$ $\square$

$\square$

================================

Theoretical evalutaion (formal methods)

===============================

=============== MODELS ==================

One possible model is to borrow the model for beam under small deformations from the theory of strength of materials, after all the whisker is a beam but we dont have small deformations at all but we assume that the model will approximatly hold ony way.

Chapter 5

Algorithms and Implementations

The testing implementation was developed with high modularity in mind, since it is meant to be a proof-of-concept implementation and not a a production grade system. High modularity also makes development easier and the system more robust against changes, two very important qualities during this project.

The implementation consists of three main parts:

Particle Filter An implementation of the procedure in table 4.1.

Database A database with functions for extracting transition hypotheses. Provides the prediction PDF $p(x_t|x_{t-1})$ to the particle filter.

Tracker Manages the model and performs matching between hypotheses and images. Provides the filtering PDF $p(I_n|x_n)$ to the particle filter.

5.1 The particle filter

A plug-n-play algorithm which only lacks the problem-domain<ref:encyc> specific parts:x_0, $p(x_t|x_{t-1})$ and $p(I_t|x_t)$

5.1.1 Initilization x_0

The particle filter only does tracking thus we need to have an initial guess for $t = 0$ which in our case will be hand labeled foreach whisker. ======

Since the algorithm only does tracking, we also need to initialize the algorithm with a start guess x_0. Using this we take a number of samples $X_0 \sim \mathcal{N}(x_0, \Sigma)$ and let the set X_0 be an approximation of $p(Z_0)$. However, the problem of automatic initialization is a difficult one[4], and is not covered in this thesis. In the testing implementation, knowledge of the start state Z_0 is assumed, meaning $x_0 := Z_0$ and $\Sigma := 0$. =====

5.1.2 Goodness $\kappa p(x_t|I_t)$

It also needs an direction to know how good the hypothesis x_t matches the given image I_t.

TODO <ref:theorem>

5.1.3 Sampling $p(x_t|x_{t-1})$

...

5.2 The state transition database

5.3 Real data

5.3.1 Pre-processing

We want to find the function $p(I = \text{whisker}) \in \mathcal{I}$

>We want to have a measure on how probable it is for a pixel to hosehould a whisker

Chapter 6

Results

<choose the word:(performance vs fitness)>

The bake-off <ref> setup for the experimental parameter evalution<ref>:

Evaluated parameters:

p The norm (where)

a Penelty for the norm

σ Resample blur (?)

ϕ Transformation

N Number of transitions

n Particle count

will be tested on a small set of benchmarks consisting of 4 disperse(d?)/scatter generated whisker videos.

The evaluation tensor:

$$\Phi_{p,a,N,\ldots}(benchmark) \tag{6.1}$$

TODO (check the comments here)

The measures that will be used as fitness of the parameters:

$\int ||\epsilon(t)||_{L^p} dt$ integrating over time the the difference with the ground truth (do this for L1:10 and see if it correlates with the p choosed (to see how much it deviates from the ground truth)

$\int \sum \phi R(x_t) \phi I_t dt$ integrating over time the response for the choosed hypothesis (to see how the different image transformation affects the results, that is if it only follows what it thinks is best (phi))

Subjective 4 image samples

And all this are done for all 4 benchmark videos.

"There are many metrics by which a model may be assessed." - Encyclopedia

The fitness test was runned on different machines but this wont effect the result since we initially wont consider the running time.

The runtime for the algorithm is handled separately on one machine setup <...>

Tillvägagångsätt:

1. Since we have prior knowledge about the effect off varying the parameters n,N they will firstly be set to a sufficently large value. 2. A partion of the test-matrix will then be evaluated by ... parameter group

Chapter 7

Analysis and Discussion

One must be carefull doing this type of experimental analysis on generate whiskers and then applying the conclusions on real whiskers, but we could still use the best parameters as a starting point for further studies on real whiskers.

" However, much machine learning research includes experimental studies in which algo- rithms are compared using a set of data sets with little or no consideration given to what class of applications those data sets might represent. It is dangerous to draw general conclusions about relative performance on any application from relative performance on this sample of some unknown class of applications. Such experi- mental evaluation has become known disparagingly as a bake-off . " - Encyclopedia (>algorithm evaluation)

"A learning algorithm must interpolate appropriate predictions for regions of the instance space that are not included in the training data." - Encyclopedia (>model evaluation)

Can a particlefilter overfit?(should this perhaps be in theory?)

Bibliography

[1] Rafael C. Gonzlez and Richard E. Woods. *Digital Image Processing, Third Edition.* Pearson Education, 2008.

[2] Snigdha Roy, Jerí L. Bryant, Ying Cao, and Detlef H. Heck. High-precision, three-dimensional tracking of mouse whisker movements with optical motion capture technology. *Frontiers in Behavioral Neuroscience*, 5(00027), 2011.

[3] Claude Sammut and Geoffrey Webb. *Encyclopedia of Machine Learning.* Springer, 2010 (First edition).

[4] Hedvig Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences.* PhD thesis, Kungliga Tekniska Hogskolan, 2001.

[5] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* The MIT Press, 2006.