Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2005 REPORT B52

MODELS AND METHODS FOR BAYESIAN OBJECT MATCHING

Toni Tamminen

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2005 REPORT B52

MODELS AND METHODS FOR BAYESIAN OBJECT MATCHING

Toni Tamminen

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering, Helsinki University of Technology, for public examination and debate in Auditorium S4 at Helsinki University of Technology (Espoo, Finland) on the 11th of November, 2005, at 12 noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Laboratory of Computational Engineering

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Laskennallisen tekniikan laboratorio

Distribution:
Helsinki University of Technology
Laboratory of Computational Engineering
P. O. Box 9203
FIN-02015 HUT
FINLAND
Tel. +358-9-451 4826
Fax. +358-9-451 4830
http://www.lce.hut.fi

Online in PDF format: http://lib.hut.fi/Diss/2005/isbn951227907x/

E-mail: Toni.Tamminen@tkk.fi

ISBN 951-22-7906-1 (printed)
ISBN 951-22-7907-X (PDF)
ISSN 1455-0474
Picaset Oy
Helsinki 2005

Abstract

This thesis is concerned with a central aspect of computer vision, the object matching problem. In object matching the aim is to detect and precisely localize instances of a known object class in a novel image. Factors complicating the problem include the internal variability of object classes and external factors such as rotation, occlusion, and scale changes. In this thesis, the problem is approached from the feature-based point of view, in which objects are considered to consist of certain pertinent features, which are then located in the perceived image.

The methodological framework applied in this thesis is probabilistic Bayesian inference. Bayesian inference is a branch of statistics which assigns a great role to the mathematical modeling of uncertainty. After describing the basics of Bayesian statistics the object matching problem problem is formulated as a Bayesian probability model and it is shown how certain necessary sampling algorithms can be applied to analyze the resulting probability distributions.

The Bayesian approach to the problem partitions it naturally into two submodels; a feature appearance model and an object shape model. In this thesis, feature appearance is modeled statistically via a type of bandpass filters known as Gabor filters, whereas two different shape models are presented: a simpler hierarchical model with uncorrelated feature location variations, and a full covariance model containing the interdependeces of the features. Furthermore, a novel model for the dynamics of object shape changes is introduced.

The most important contributions of this thesis are the proposed extensions to the basic matching model. It is demonstrated how it is very straightforward to adjust the Bayesian probability model when difficulties such as scale changes, occlusions and multiple object instances arise. The changes required to the sampling algorithms and their applicability to the changed conditions are also discussed.

The matching performance of the proposed system is tested with different datasets, and capabilities of the extended model in adverse conditions are demonstrated. The results indicate that the proposed model is a viable alternative to object matching, with performance equal or superior to existing approaches.

List of abbreviations

AAM	Active appearance model
ASM	Active shape model
BP	Belief propagation
EBGM	Elastic bunch graph matching
IMM-DTU	Informatics and Mathematical Modelling, Technical University of Denmark
JPDA	Joint probability data association
KLT	Kanade-Lucas-Tomasi
NBP	Nonparametric belief propagation
MCMC	Markov chain Monte Carlo
MAP	Maximum a posteriori
ML	Maximum likelihood
MC	Monte Carlo
P2P	Point-to-point
P2C	Point-to-curve
PCA	Principal component analysis
PPCA	Probabilistic principal component analysis
RGB	Red-green-blue
SIFT	Scale invariant feature transform
SIR	Sequential importance resampling
SMC	Sequential Monte Carlo

Preface

This thesis is the result of my research carried out at the Laboratory of Computational Engineering at Helsinki University of Techonology during the years 2001-2005, begun as a Master's Thesis worker and continued as a graduate student from 2002 onwards. At first, my work was funded by the TEKES USIX project, and afterwards by the ComMIT graduate school and the Academy of Finland Centre of Excellence in Computational Science and Engineering. In addition to these, my research has been supported financially by the Finnish Cultural Foundation, the Jenny and Antti Wihuri Foundation, and the Nokia Foundation. I am grateful to all of these parties who have made the completion of this thesis possible.

I wish to express my deepest gratitude to Prof. Jouko Lampinen for his excellent instruction and supervision of my thesis - whenever there was a problem, I could pop in his office and we would work it out. I also wish to thank Dr. Aki Vehtari for his help with all kinds of stuff ranging from posterior distributions to LaTeX tables. Besides relying on their expertise in scientific matters, it has been really fun to work and spend time with Jouko and Aki. Dr. Michael Frydrych, Ilkka Kalliomäki, Timo Kostiainen, Jari Kätsyri, Gabrielle Simonetti, and Simo Särkkä deserve thanks for research collaboration and their help with various parts of the thesis. I am also thankful to the other past and present members of the laboratory, especially Aatu Kaapro, Laura Kauhanen, Dr. Teemu Leppänen, Aapo Nummenmaa, Tommi Nykopp, Janne Ojanen, and Sebastian von Alfthan, for providing a nice and inspiring work atmosphere. For helping with the more concrete work facilities and practicalities, I wish to thank Prof. Kimmo Kaski and Eeva Lampinen.

Finally, I want to thank all my friends and family for helping me to maintain a life outside the laboratory, with special thanks to the founding members of the PKL Society: Aapo, Anne-Mari, Antti, Ilkka, Laura and Sebastian, as well as the auxiliaries Pasi and Tommi, for all the good times we've had.

Thank you, all!

Toni Tamminen

Contents

Chapter 1

Introduction

The human vision system is the most sophisticated object recognition system known. Humans have no difficulties in interpreting scenes containing tens or even hundreds of objects of different sizes, shapes and poses. One of the major aims in the field of computer vision is to produce an artificial vision system capable of replicating the recognition abilities of human vision. The subject is an extensively researched one, and promising results have been obtained in the last few decades. However, the goal of emulating the human vision system still seems rather distant, which is because of the difficulty of the problem; as illustrated in Figure 1.1, the computer representation of images is very different from the human visual experience.

This thesis concerns itself with one aspect of the object recognition problem, namely the detection and matching of objects, where matching means the precise location of an object and its interesting visual landmarks. Object matching is a natural precursor to actual object recognition - for example, in order to recognize a person from an image, the person-specific features which make the recognition possible must first be located.

The general object matching problem can be phrased as follows: Given a set of examples of an object class, learn the fiducial properties of the class and use them to locate instances of the object class in a novel image. The difficulty of the problem is directly related to the internal variability of the object class. Other factors complicating the situation are varying scale and orientation of the objects and the possibility of occlusion. A complete solution to the problem should also be able to cope with multiple object classes.

Prominent approaches to the object matching problem include simple methods such as template matching, edge and corner point matching, and contour matching, as well as more complex systems such as Elastic Bunch Graph Matching (EBGM) (Wiskott et al., 1997) and Active Shape and Appearance (AAM) models (Cootes et al., 1995) (Cootes et al., 2001). A common theme in most of these is

111	89	88	118	105	115	114
111	111	102	117	114	107	107
17	41	19	31	17	27	112
136	58	49	46	159	110	58
116	106	148	151	151	151	59
113	120	123	124	126	128	119
163	173	173	128	104	104	112
173	174	175	174	174	174	175

Figure 1.1: Human and machine perception of a scene. For a human, it is easy to almost instantly locate and recognize the objects contained in the image on the left. However, the computer representation of an image is simply a matrix of numbers - depicted on the right are the values of one color component in a part of the image on the left. These numbers must be interpreted via computer vision methods to obtain a meaningful description of the scene.

that objects are represented as a combination of visual features and a some kind of shape model which links the locations of the features together.

As opposed to traditional solutions, in this thesis the problem is approached from a probabilistic Bayesian point of view. The Bayesian framework provides a natural way to join the appearance and shape models; furthermore, the Bayesian approach makes it possible to address some of the challenges of object matching in a very straightforward fashion. Also, a prominent view in the field of vision science and neurobiology holds that the human visual system employs Bayesian-like inference in its visual processing as well (Kersten and Yuille, 2003) (Lee and Mumford, 2003).

The aims of this thesis are to present a solution to the object matching problem and some related subproblems based on the Bayesian formulation of the problem and to analyze the capabilities of such a solution. The proposed approach contains both novel models for the parts of the problem as well as methodological advances in applying the models.

The thesis is organized as follows. Chapter 2 contains an introduction to Bayesian probability theory and related concepts. Since the analysis of complex Bayesian models often requires numerical sampling methods such as Markov chain Monte Carlo and sequential Monte Carlo, these are be discussed as well.

The chapter concludes by formulating the object matching problem in Bayesian terms. The main contributions here are the analysis of the various sampling methods as well as to some extent the Bayesian formulation of the feature-based object matching problem and the very use of sequential Monte Carlo in a static problem.

Chapter 3 is dedicated to models of visual appearance. Prominent existing approaches to modeling the appearance of visual features are described, alongside the novel statistical models proposed in this work. Additionally, more holistic appearance models are discussed. The main contribution is the Gabor-filter based feature appearance model, especially its statistical nature.

Chapter 4 concentrates on models of object shape. The discussion proceeds from simpler ad hoc models to more rigorous statistical models. In the final part of the chapter, a novel dynamical model for facial expression modeling is described. Here the contributions are the hierachical graph prior and the Bayesian approach to the estimation of the full covariance prior alongside the facial expression model.

Chapter 5 extends the basic matching model. Traditional problems in object matching such as scale-invariance, occlusion, and multiresolution matching are addressed by modifying the matching model accordingly. The chapter clearly demonstrates the power of the Bayesian approach: to accommodate for changes in the matching environment, it is sufficient to adjust the probability model. The whole chapter consists of novel contributions and can be considered the most significant part of this thesis.

Chapter 6 applies the results of the previous chapters to actual matching problems and compares them to state-of-the-art published approaches. It is shown that the proposed system is able to match objects with equal or greater accuracy as previous approaches, also in very adverse matching conditions.

Finally, Chapter 7 reviews the proposed models and methods and concludes the work.

Chapter 2

The Bayesian approach to object matching

2.1 Introduction

Bayesian probability theory derives its name from Reverend Thomas Bayes, who is credited with discovering Bayes' theorem, the cornerstone of Bayesian statistical analysis. Bayesian theory is based on the notion of subjective probability, which means that each observer measures probability according to his or her prior beliefs and his or her observations of past events. This differs radically from the definition used in traditional or frequentist statistics, where probability is defined as the limit of favorable results in a random test when the number of tests approaches infinity. Another fundamental difference between the Bayesian and frequentist approaches is that in Bayesian analysis there are no fundamental differences between the parameters of a model and the observable quantities - all are considered random variables. This is one of the main advantages of the Bayesian point of view, as it makes it very straightforward to express uncertainty mathematically.

This chapter describes the principles of Bayesian statistical inference as well as the sampling methods required in the analysis of complicated Bayesian models, and ends by representing the object matching problem in Bayesian terms. For a more general treatment of Bayesian data analysis, the reader is referred to (Gelman et al., 2004) and (Bernardo and Smith, 1994).

2.2 Bayesian inference

2.2.1 Bayes' theorem

In Bayesian inference, all observable and unobservable quantities are considered random variables. If the observable quantities or the data are denoted with D, the unobservable quantities or the model parameters with θ, and the implicit model assumptions such as distribution choices with M, the most general representation of any statistical inference problem is the joint distribution

$$p(D, \theta, M). \tag{2.1}$$

In principle it is possible to compute the probability distributions of the model parameters for any model and any data from this distribution. Usually only one model is examined at a time, and hence the joint distribution

$$p(D, \theta|M) \tag{2.2}$$

is considered. Using the rule of conditional probabilities, this can be written as

$$p(D, \theta|M) = p(D|\theta, M)\, p(\theta|M), \tag{2.3}$$

and by conditioning the left side on the data, Bayes' theorem is obtained:

$$p(\theta|D, M) = \frac{p(D|\theta, M)\, p(\theta|M)}{p(D|M)}. \tag{2.4}$$

The left side of Eq. 2.4 is the posterior probability distribution of the model parameters given the data. On the right-hand side, $p(D|\theta, M)$ is the likelihood of the data, which measures the probability of observing the data given a set of parameter values. $p(\theta|M)$ is the prior distribution of the parameter values before the data is observed. $p(D|M) = \int p(D|\theta, M)\, p(\theta|M) d\theta$ is the marginal distribution of the data given the model, also called the evidence of the model. This term measures the probability of the data under model M over all possible parameter values. Since often the interest lies only on the parameters, this term can be left out to produce the unnormalized posterior distribution

$$p(\theta|D, M) \propto p(D|\theta, M)\, p(\theta|M). \tag{2.5}$$

2.2.2 Posterior analysis and prediction

The posterior distribution contains all information that can be inferred about the values of the parameters given the data and the observer's prior beliefs. If a point estimate is required, it can be computed from the posterior distribution. Tradi-

tional point estimates include the maximum of the distribution (Maximum A Posteriori, or the MAP estimate), the posterior mean and the posterior median. Also, credible intervals, the counterpart of Bayesian analysis to the traditional confidence intervals, can be computed and hypothesis testing performed by analyzing the posterior distribution.

Besides analyzing the observed data, statistical inference often tries to predict the future. In Bayesian analysis this is straightforward: if the data generating process $p(y|\theta, M)$ is known, the posterior predictive distribution of a new observation can be computed as

$$p(y|D, M) = \int p(y|\theta, M)\, p(\theta|D, M) d\theta. \tag{2.6}$$

Similarly, if there are no observations, the prior predictive distribution is obtained from

$$p(y|M) = \int p(y|\theta, M)\, p(\theta|M) d\theta. \tag{2.7}$$

2.2.3 Hierarchical models

In many kinds of problems it is useful to include some kind of parameter hierarchy in the model. For example, it might be known that some low level parameters θ come from a common superpopulation, or the uncertainty about the model parameters might be so great that not even the parameters of their prior distributions can be assigned fixed values. In this case the model includes parameters ξ whose values are not interesting as such. Instead of analyzing the full posterior distribution $p(\theta, \xi|D, M)$, the marginal posterior distribution is examined:

$$p(\theta|D, M) = \int p(\theta, \xi|D, M) d\xi. \tag{2.8}$$

The process of integrating over the uninteresting or nuisance parameters is called marginalization. In this fashion it is possible to construct hierarchical models, in which the parameters of the model are also assigned their own priors:

$$p(\theta|D, M) = \int p(\theta, \xi|D, M) d\xi \propto \int p(D|\theta, \xi, M)\, p(\theta|\xi, M)\, p(\xi|M) d\xi. \tag{2.9}$$

Because of the complexity of hierarchical models, it is often impossible to analyze them in closed form. Instead, sampling methods are required, which will be discussed in Section 2.3.

2.3 Sampling methods

Analysis of Bayesian statistical models often require the evaluation of multidimensional integrals. Examples are the normalization integral

$$p(D|M) = \int p(D|\theta, M)\, p(\theta|M) d\theta, \tag{2.10}$$

the posterior mean

$$\mathrm{E}[p(\theta|D, M)] = \int \theta p(\theta|D, M) d\theta, \tag{2.11}$$

or any of the marginalization integrals described in Section 2.2.3. Difficulties tend to arise if the model contains any other than the most basic probability distributions, as the posterior can not be analyzed in closed form. Although it is possible to approximate posterior distributions with Gaussians by computing the mode and the Hessian at the mode of the posterior distribution (Gelman et al., 2004), this approach fails if the posterior is multimodal. Another possibility is to use traditional numerical integration methods, but these become too heavy computationally in high-dimensional problems. The two main solutions to the problem are Variational Bayes methods (Ghahramani and Beal, 2001), which seek to approximate the posterior analytically with a family of simpler distributions, and algorithms that sample from the posterior numerically. In this thesis, the sampling-based approach is adopted.

In low-dimensional problems it is often possible to sample directly from the posterior via the inverse cdf method or its extension, the grid sampling method (Milton and Arnold, 1995). However, as the dimensionality of the problem grows beyond two, also these methods become unfeasible. For high-dimensional Bayesian integrals a popular solution is to use Monte Carlo sampling algorithms.

2.3.1 Monte Carlo sampling

The aim of Monte Carlo (MC) methods is the production of samples from a target distribution $\pi(.)$, which allows the estimation of the the expected value by using the sample mean, that is, $E[f(x)]$ is estimated by drawing samples $\{x_i,\ i = 1, ..., n\}$ from $\pi(.)$ and then approximating

$$E[f(x)] \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i). \tag{2.12}$$

The law of large numbers (Milton and Arnold, 1995) ensures that the right-hand side of Eq. 2.12 converges to the left-hand side with probability 1 provided that the samples x_i are independent.

The aim can also be to directly evaluate a definite integral

$$I = \int_a^b f(x)dx. \tag{2.13}$$

This is done by identifying a random variable Y with support on (a, b) and a function g so that the expected value of $g(Y)$ is I:

$$\begin{aligned} E[g(Y)] &= \int_a^b g(y)p(y)dy && (2.14)\\ &= \int_a^b f(y)dy && (2.15)\\ &= I, && (2.16) \end{aligned}$$

i.e., the problem is to define suitable $g(y)$ and $p(y)$ and apply Eq. 2.12.

Thus in both cases, the actual task is to draw samples independently from a (multidimensional) probability distribution - in fact, in many cases the samples themselves are the object of interest instead of the estimate of the expected value. What makes Monte Carlo methods enticing in Bayesian analysis is that in order to apply them, it is sufficient to know the target distribution $\pi(.)$ in an unnormalized form $\pi^*(.)$ so that $\pi(.) = \pi^*(.)/Z$, where Z is a normalization coefficient.

Traditional Monte Carlo methods include uniform sampling, rejection sampling and importance sampling (for more, see (Gentle, 1998)).

Uniform sampling

If the expected value is the only thing of interest, and the dimension of $\pi^*(.)$ is low, Eq. 2.12 can be evaluated using uniform sampling, in which

$$\hat{E}[f(x)] = \sum_i f(x_i)\frac{\pi^*(x_i)}{\sum_j \pi^*(x_j)}, \tag{2.17}$$

i.e., the samples are drawn from a uniform distribution, and the values of the function and the corresponding distribution are computed in the sample points. Again, if the dimension of the distribution is large, this becomes computationally impossible as the number of samples required to preserve the accuracy of the estimate grows exponentially with the dimension.

Rejection sampling

Uniform sampling can only be used to evaluate the expected value of a function, i.e., no explicit samples from the target distribution are obtained. Rejection sampling is an MC algorithm that actually produces the samples, with the expected

value computed using Eq. 2.12 if necessary. The algorithm generates the samples x_i with the help of a proposal distribution $Q(.) \geq \pi(.)$, from which samples can be drawn easily. $Q(.)$ can for example be a uniform or Gaussian distribution, multiplied with a constant to fulfill the condition. The samples are generated by drawing x_i from $Q(.)$ and accepting the sample with probability $P = \frac{\pi(x_i)}{Q(x_i)}$.

The choice of the proposal distribution is of critical importance. If it does not resemble the target distribution, a large multiplication constant is required to fulfill the inequality condition, and the acceptance probability is generally low. Thus the amount of proposed samples can become very large compared to the number of accepted samples, which is inefficient and time-consuming.

Importance sampling

Importance sampling is an improved version of uniform sampling. It is based on the assumption that the integral being evaluated is not nearly uniform, and thus the variance of the estimate can be reduced by concentrating the sampling effort on the regions where the target distribution has large values. This again requires a proposal distribution $Q^*(.)$, which is not required to be normalized. The samples are drawn from this distribution and a weight $w_i = \frac{\pi^*(x_i)}{Q^*(x_i)}$ is assigned to each sample. The estimate of the expected value can thus be calculated as

$$\hat{E}[f(x)] = \frac{\sum_i w_i f(x_i)}{\sum_i w_i}. \tag{2.18}$$

Importance sampling suffers from the fact that if the proposal distribution does not resemble the target distribution very closely, in multidimensional problems there most probably are areas such that $\pi^* >> Q^*$. Thus samples drawn from these regions have very large weights, causing the estimate to be dominated mainly by them, which in turn leads to large variance. However, with a good proposal distribution importance sampling works very well, and even with an inadequate proposal distribution it outperforms uniform sampling in almost all cases. Furthermore, by performing importance resampling, that is, by sampling from x_i according to the importance weights, it is possible to obtain actual samples from the target distribution. Importance resampling is a crucial part of most sequential Monte Carlo algorithms, described in Section 2.3.3.

2.3.2 Markov chain Monte Carlo

The Monte Carlo sampling algorithms presented in Section 2.3.1 generate, at least in theory, independent samples x_i. Especially with multidimensional distributions they can be quite inefficient. Fortunately, it turns out that the samples x_i need not necessarily be independent - they can be generated through any process which draws samples through the support of $\pi(.)$ in correct proportions. One way of

doing this is to draw the samples with the help of a Markov chain (Brzezniak and Zastawniak, 1999). This is the essence of Markov chain Monte Carlo (or MCMC) methods.

Subject to regularity conditions, a Markov chain will gradually "forget" its initial state x_0 and converge to a unique stationary distribution, denoted $\phi(.)$, that is independent of both t and x_0. As t increases, the samples from the chain $\{x_t\}$ will look increasingly like samples from $\phi(.)$. Thus the samples x_i from $\pi(.)$ can be generated by constructing a Markov chain with $\pi(.)$ as its stationary distribution and letting the chain run sufficiently long so that the convergence is sufficient. What exactly is meant by "sufficient" is the topic in the field called convergence diagnostics. See (Gilks et al., 1996) for information on this, as well as on other topics pertaining to MCMC.

The Metropolis and Metropolis-Hastings algorithms

The principal differences between the various MCMC algorithms result from how the state of the chain is updated. The first MCMC algorithm, the Metropolis algorithm (Metropolis et al., 1953), utilizes a proposal distribution $p(.|.)$ to sample a candidate x^* for the next state. In the traditional Metropolis algorithm this proposal distribution is required to be symmetric, i.e., $p(x^*|x_t) = p(x_t|x^*)$. The state transition $x_{t+1} = x^*$ is accepted with probability

$$p = min\left(1, \frac{\pi^*(x^*)}{\pi^*(x_t)}\right). \tag{2.19}$$

If the transition is not accepted, the state is unchanged, i.e., $x_{t+1} = x_t$. The transition kernel of the underlying Markov chain thus consists of the proposal distribution and the acceptance probability parts. From Eq. 2.19 it can be seen that if the probability of the new state is higher than that of the old state according to $\pi(.)$, the state is always switched, and if the new state is less probable, the state is switched sometimes anyway. Figure 2.1 illustrates Metropolis sampling.

The remarkable thing about the Metropolis algorithm is that regardless of the shape of the proposal distribution $p(.|.)$ the resulting stationary distribution of the chain will be $\pi(.)$ (Gilks et al., 1996). However, the relationship between $\pi(.)$ and $p(.|.)$ will crucially affect the convergence speed, as well as mixing, i.e., the inter-distribution movement once the chain has converged. For continuous variables a multivariate Gaussian distribution usually works quite well as a proposal distribution.

The Metropolis-Hastings algorithm is a generalization of the standard Metropolis algorithm (Hastings, 1970). The difference is that in Metropolis-Hastings the

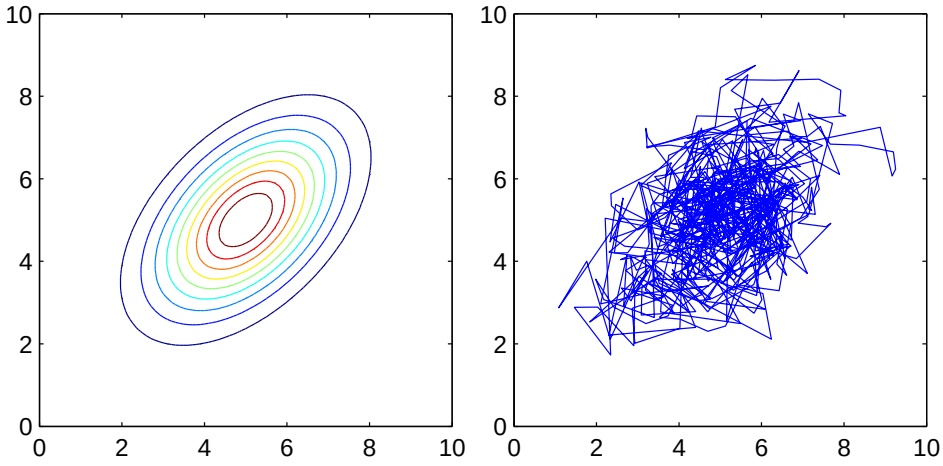

Figure 2.1: An example of Metropolis sampling. The image on the left shows the contours of a Gaussian density function and the image on the right shows the samples drawn by the Metropolis algorithm. The lines depict the moves of the algorithm; that is, the samples are located at the corner points of the curve.

proposal distribution may be asymmetric, leading to the acceptance probability

$$p = min\left(1, \frac{\pi^*(x^*)\,p(x_t|x^*)}{\pi^*(x_t)\,p(x^*|x_t)}\right). \tag{2.20}$$

Note that the basic Metropolis algorithm follows from this as the special case with the symmetric proposal distribution $p(x_t|x^*) = p(x^*|x_t)$. A special case is single-component Metropolis-Hastings, in which the state vector is divided into updating blocks $\{x_1, x_2, ..., x_h\}$, and the blocks are updated sequentially, but not necessarily in numerical order, using Metropolis-Hastings. The candidate state is generated from the proposal distribution $p_i(x_i^*|x_i, x_{\setminus i})$ where x_i is the state of the ith block at time step t and $x_{\setminus i}$ is the whole state vector with components 1, 2, ..., $i-1$ already updated. Each component thus has its own proposal distribution p_i which may depend on the current values of the components of x. The acceptance probability of the candidate is

$$p = \min\left(1, \frac{\pi^*(x_i^*|x_{\setminus i})\,p_i(x_i|x_i^*, x_{\setminus i})}{\pi^*(x_i|x_{\setminus i})\,p_i(x_i^*|x_i, x_{\setminus i})}\right). \tag{2.21}$$

The Gibbs sampling algorithm

Gibbs sampling is alongside Metropolis-Hastings the most widely used MCMC algorithm in statistical applications (Geman and Geman, 1984), (Gelfand and

Smith, 1990). Gibbs sampling is a special case of single-component Metropolis-Hastings, in which the proposal distribution for updating the i^{th} component of the state vector is

$$p_i(x_i^*|x_i, x_{\setminus i}) = \pi^*(x_i^*|x_{\setminus i}), \tag{2.22}$$

the full conditional distribution of the ith component given all the other components $x_{\setminus i}$. It can be seen by inserting this into Eq. 2.21 that this choice leads to the acceptance probability always being 1; that is, Gibbs sampler candidates are always accepted. Thus Gibbs sampling updates the components simply by sampling from their corresponding full conditional distributions, one at a time. Naturally, this requires that the conditional distributions can be formed, which can be considerably harder than simply defining the joint posterior distribution as in Metropolis, and can limit the applicability of the algorithm. Again it should be noted that the components are not required to be updated in numerical order. Moreover, even not all components need to be updated in each iteration. Gibbs sampling is illustrated in Figure 2.2.

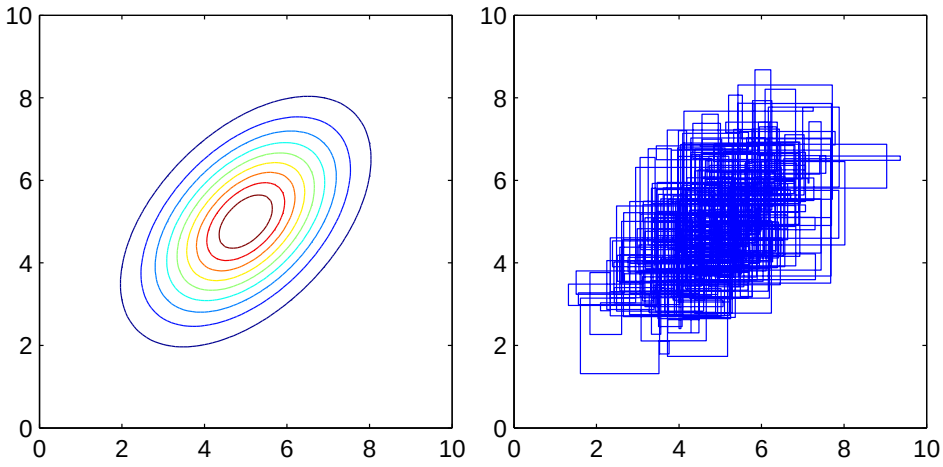

Figure 2.2: An example of Gibbs sampling. The image on the left shows the contours of a Gaussian density function and the image on the right shows the samples drawn by the Gibbs sampling algorithm. Again, the lines depict the moves and the samples are located at the corner points.

2.3.3 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods, or particle filters are a relatively novel tool in Bayesian inference (Gordon et al., 1993) (Doucet et al., 2001) that can be considered as the Bayesian extension of the widely-used Kalman filter (Welch

and Bishop, 1997). Sequential Monte Carlo is based on the notion of representing the probability mass of the posterior as a set of weighted samples, or particles, so that the samples are updated in a sequential fashion. A typical application is the tracking of a moving object, in which the samples represent the posterior of the current location of the object, and updating occurs as new measurements of the location of the object are obtained. SMC methods can also be used for evaluating static posteriors - Chopin (2002) does this to achieve time savings, Ridgeway and Madigan (2003) employ the method to make the analysis of massive datasets possible, and Pérez et al. (2001) use SMC for segmentation and road extraction in still images. In all of the above examples, the static problem is transformed into a type of tracking problem in which the data is processed sequentially. A different approach is proposed in this thesis: the amount of data does not change, and the parameters are updated sequentially. In tracking terms, this corresponds to the smoothing problem, in which all observations are available for the inference of the parameters. This kind of approach is very natural to the object matching problem, as will be described in Section 2.4. Previously, a similar SMC approach to sampling from static distributions in an object matching problem has been presented by Zhang et al. (2004).

In sequential Monte Carlo, the target posterior distribution is represented as a weighted set of n particles $(\mathbf{x}_t^i, w_t^i)$ where t is the time step, $\mathbf{x}_t^i$ are the particles containing the parameter values, and w_t^i the weights. As in any sampling-based approach, all posterior inference is carried out using the particles; for example, the posterior mean of the parameter of interest θ_t at time step t can be computed as

$$\hat{E}[\theta_t] = \frac{\sum_i w_t^i \mathbf{x}_t^i}{\sum_i w_t^i}. \tag{2.23}$$

At each time step, a previously unsampled component x_t^i of $\mathbf{x}_t^i$ is updated as in importance sampling (Section 2.3.1) by drawing it from a proposal distribution $x_t^i \sim q_t(x|.)$, and the other components of $\mathbf{x}_t^i$, i.e., $\mathbf{x}_{t-1}^i$ are assumed to stay unchanged. The weights are then updated by computing the ratio of the target posterior distribution and the joint proposal distribution, including $\mathbf{x}_{t-1}^i$, at the sampled point (Robert and Casella, 2004):

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{x}_t^i|D)}{p(\mathbf{x}_{t-1}^i|D)q_t(x_t^i|.)} \propto w_{t-1}^i \frac{p(D|x_t^i)p(x_t^i|\mathbf{x}_{t-1}^i)}{q_t(x_t^i|.)}, \tag{2.24}$$

that is, the posterior density of the previously unsampled component is divided by its proposal density. In applications such as tracking, where also new data y_t arrives at each time step, the term $p(D|x_t^i)$ becomes $p(y_t|x_t^i)$, as the data points are assumed independent given $\mathbf{x}_{t-1}$. When all particles have been updated, a resampling step according to the particle weights may be performed so that the

weights do not degenerate to a few non-zero ones (Robert and Casella, 2004). After resampling all particle weights are set to $1/n$. This is the basic form of the sequential importance resampling (SIR) algorithm, illustrated in Figure 2.3. Note that in SIR the proposal distribution can be of any form, as long as it is included correctly in the weight update equation.

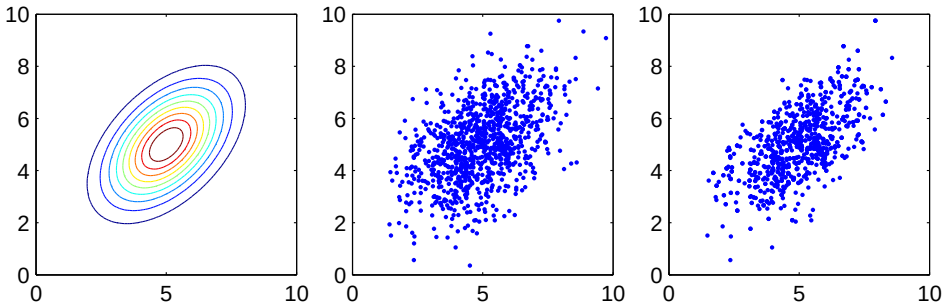

Figure 2.3: An example of sequential sampling. The image on the left shows the contours of a Gaussian density function, the center image shows the result of drawing the first parameter from its marginal distribution and the second from its conditional distribution, and the image on the right shows the particles after resampling. The target Gaussian distribution is assumed to be the product of identical prior and likelihood parts.

The first steps of an SMC simulation can be somewhat problematic. It is not possible to use Eq. 2.24 to compute the weights, as there are no previously drawn samples. One solution is to draw the first components z_1^i from a marginal distribution - either the marginal likelihood $p(D|z_1^i)$ or the marginal prior $p(x_1^i)$, depending on the problem at hand, and assume that the other term is constant, which causes all weights to be the same. In most cases, the marginal likelihood is probably more useful, as in many cases the marginal prior can be too vague to be of any practical benefit.

Sequential Monte Carlo has a certain undesirable characteristic: the degeneration of the distribution of previously sampled parameters (Robert and Casella, 2004). As the first parameters of the sequence are never re-simulated, due to repeated weighting and resampling the particle representation may contain just a few or even just one value for them. In this case the posterior uncertainty of these parameters is clearly underestimated. The problem can be averted by adding an MCMC step to the matching process so that the previously simulated parameters can be altered. However, with multidimensional posteriors this is very expensive computationally, and in practice using a large enough number of particles often sidesteps the problem.

As a side note, it can be mentioned that according to Lee and Mumford (2003), there is evidence that the human brain utilizises methods similar to particle fil-

tering in its visual processing, which is an additional incentive to use SMC in computer vision.

2.3.4 Nonparametric belief propagation

Nonparametric belief propagation (NBP)is a recently proposed algorithm for estimating the parameters of a graphical model (Sudderth et al., 2003). Another slight variation is the PAMPAS algorithm (Isard, 2003). As NBP has received much attention in the computer vision community since its introduction, the algorithm is outlined here briefly.

An undirected graph is defined as a set of nodes $\mathcal{V}$ and edges $\mathcal{E}$. The neighborhood of a node s is defined as $\Gamma(s) = \{v|(s, v) \in \mathcal{E}\}$ and each node is associated with a hidden variable x_s and an observation y_s. The joint probability density of the graph is

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{(s,v)\in\mathcal{E}} \psi_{s,v}(x_s, x_v) \prod_{s\in\mathcal{V}} \psi_s(x_s, y_s), \tag{2.25}$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the sets of all hidden variables and observations, Z is a normalization constant, $\psi_{s,v}(x_s, x_v)$ is the pairwise compatibility potential between nodes s and t, and $\psi_s(x_s, y_s)$ is the local observation potential for node s. These correspond to the prior and likelihood parts in the standard Bayesian formulation of an inference problem. The aim of the model is to estimate the local conditional marginal posteriors $p(x_s|\mathbf{y})$.

If the graph is acyclic or tree-structured, it is possible to compute the posteriors $p(x_s|\mathbf{y})$ directly with a local message-passing algorithm known as belief propagation (BP) (Yedidia et al., 2004). At iteration t of the BP algorithm each node v computes messages $m^t_{vs}(x_s)$ to be passed to its neighbors x_s :

$$m^t_{vs}(x_s) = \alpha \int \psi_{s,v}(x_s, x_v)\psi_v(x_v, y_v) \times \prod_{u\in\Gamma(v)\setminus s} m^{t-1}_{uv}(x_v)dx_v, \tag{2.26}$$

where α is a proportionality constant. That is, each node multiplies the messages passed to it, adjusts the product with the local observation, and passes the resulting message to its neighbors according to the pairwise potentials so that the uncertainty about the value of the variable x_v is integrated out. The marginal posteriors $p(x_s|\mathbf{y})$ can be estimated at any iteration by combining the incoming messages with the observation potential:

$$\hat{p}^t(x_s|\mathbf{y}) = \alpha\psi_x(x_s, y_s) \prod_{v\in\Gamma(s)} m^t_{vs}(x_s). \tag{2.27}$$

For an acyclic or tree-structured graph, these estimates will converge to the true posteriors once the messages from each node have propagated to every other node

of the graph.

If the graph contains cycles, the estimates $\hat{p}^t(x_s|\mathbf{y})$ will not generally converge to the correct posteriors, because the statistical dependencies of the messages are not properly taken into account. However, in many applications the loopy BP algorithm still achieves good empirical results (Murphy et al., 1999) (Frey et al., 2001). Another problem with standard belief propagation is that the integral of Eq. 2.26 becomes intractable if any of the messages or potentials are of non-Gaussian form. In this case the integral must be approximated.

The BP update equation (Eq. 2.26) can be decomposed into two stages. First, the message product $\prod_{u\in\Gamma(v)\setminus s} m_{uv}^{t-1}(x_v)$ is combined with the local observation potential $\psi_v(x_v, y_v)$ to produce the distribution of the variable x_v. Second, this distribution is combined with the compatibility potential $\psi_{s,v}(x_s, x_v)$ and integrated to produce the message sent to node x_s. The nonparametric belief propagation algorithm applies sampling to approximate theses two stages in order to produce consistent nonparametric representations of the messages $m_{vs}(x_s)$. In NBP, the messages are approximated as a mixture of M Gaussian kernels

$$m_{vs}(x_s) = \sum_{i=1}^{M} w_s^{(i)} \mathrm{N}(x_s|\mu_s^{(i)}, \Lambda_s), \tag{2.28}$$

where $\mathrm{N}(.\,|.)$ is the Gaussian density function, $w_s^{(i)}$ is the weight associated with the i^{th} kernel mean $\mu_s^{(i)}$, and Λ_s is the variance or the smoothing parameter of the kernel. If the observation potentials $\psi_v(x_v, y_v)$ are also represented with mixtures of Gaussians, the first stage of the BP update equation is in principle very straightforward, as the products of Gaussian mixtures are itself Gaussian mixtures. However, in practice approximations have to be made, as the number of mixture components grows exponentially in the number of input messages: for d input messages of M Gaussians each, the product contains M^d components. NBP approximates the product with M independent samples. To avoid computing each of the M^d component weights, the algorithm instead uses a Gibbs sampler on the kernel labels to select a single component from each of the messages, multiplies these, and draws a sample from the product Gaussian (for details, see (Sudderth et al., 2003)). If the observation potentials are not mixtures of Gaussians but instead an analytic function $f(x)$, the Gibbs sampler can still be utilized via importance sampling: the weights associated with the components are scaled with the observation potential at each kernel's center, after which an importance weight $f(\hat{x})/f(\bar{\mu})$, where $\bar{\mu}$ is the center of the product Gaussian, is assigned to the sampled particle $\hat{x}$.

In the second stage of the BP update equation, the result of the first stage is combined with the compatibility potential $\psi_{s,v}(x_s, x_v)$ and integrated to produce the outbound message. This requires that the marginal influence of the pair-

wise potential on x_v must be separated from the conditional relationships between x_v and x_s. The marginal influence function $\zeta(x_v)$ is obtained by integration:

$$\zeta(x_v) = \int \psi_{s,v}(x_s, x_v) dx_s. \tag{2.29}$$

In NBP, this function is incorporated into the Gibbs sampler so that the first stage product becomes $\zeta(x_v)\psi_v(x_v, y_v)\prod_u m_{uv}(x_v)$. If the potential is not a Gaussian mixture, importance sampling and symbolic or numeric integration may be required. To complete the integration, the samples $x_v^{(i)}$ obtained by the Gibbs sampler are propagated to node x_s by sampling

$$m_{vs}^{(i)}(x_s) \sim \psi_{s,v}(x_s, x_v^{(i)}), \tag{2.30}$$

which may require importance sampling or MCMC depending on the functional form of $\psi_{s,v}(x_s, x_v)$. Finally, to produce the nonparametric density estimate for the message, a Gaussian kernel with a suitable width is placed at each sample $m_{vs}^{(i)}$. The main difference between the PAMPAS algorithm (Isard, 2003) and NBP is that in PAMPAS, the final sampling and kernel placing step is omitted. Instead, the interaction potentials are assumed to be mixtures of just a few Gaussians and the message estimate is formed simply as

$$\hat{m}_{vs}(x_s) = \frac{1}{M}\sum_{i=1}^{M} \psi_{s,v}(x_s, x_v^{(i)}). \tag{2.31}$$

The additional kernel placement leads to variance estimates that are biased upwards of their true values; however, it allows more general compatibility potentials that are not necessarily mixtures of Gaussians.

2.3.5 Comparison of the sampling methods

The results of this section have previously been discussed in (Tamminen and Lampinen, 2003b) and (Tamminen and Lampinen, 2005). The presented main sampling methods (Metropolis, Gibbs, and SMC) differ mainly in three aspects: the ease of use, exploration of a single mode, and performance in multimodal situations. The Metropolis algorithm is clearly the most straightforward to employ: it only requires the joint posterior distribution of the parameters, whereas Gibbs sampling and sequential Monte Carlo need conditional distributions - Gibbs sampling requires the full conditional distributions, SMC the distributions given the previously sampled parameters, and additionally the marginal distributions for the first parameter.

When considering exploration of a single mode, Gibbs is mostly superior. The performance of Metropolis is very dependent on its proposal distribution - a

poor choice leads to poor results. Sequential Monte Carlo is also affected by the proposal distribution, and also suffers from the degeneration problem described in Section 2.3.3. Gibbs sampling explores modes efficiently, and requires no proposal distribution besides the conditional distributions. However, it has been shown that even if the distribution is unimodal, the convergence time of Gibbs sampling can increase exponentially with dimension - an example of this is the witch's hat distribution, which has a single sharp mode and a low uniform level elsewhere (Matthews, 1993). This is a consequence of the curse of dimensionality, as all parameters are required to have approximately correct values simultaneously for the sampler to find the mode. Sequential Monte Carlo outperforms MCMC algorithms in such situations because it progresses from low to high dimensional distributions instead of trying to draw all the parameters at once.

The largest difference between the algorithms is their behavior when the target distribution is multimodal. In this case, MCMC algorithms often fail to explore all modes, as they are very unlikely to change from one mode to another, distant mode. Sequential Monte Carlo does not suffer from multimodality as much due to its ability to maintain multiple parallel hypotheses about the parameter values. The inability of MCMC algorithms to change mode also causes them to be sensitive to initialization, which is clearly undesirable. In many cases, an enticing alternative to MCMC methods is slice sampling (Neal, 2003). The deeper analysis of the slice sampling algorithm and its application to the object matching problem have been left to a further study.

As to non-parametric belief propagation, it is not straightforward to compare it to the various sampling algorithms. NBP is a tool for a specific problem formulation - the graphical model - whereas MCMC and SMC methods are generic statistical methods that can be used to analyze almost any kind of model. It is important to note that NBP does not even asymptotically converge to the true posterior due to the statistical dependencies of the messages in a cyclic graph, whereas for the other algorithms asymptotical convergence can be shown. Furthermore, in NBP non-adjacent nodes affect each other only indirectly through other nodes, whereas in the model proposed in this thesis these kind of effects are direct, as will be seen in Section 2.4 and Chapter 4. Since the direct dependencies can often be estimated from the training data, it makes sense to use this information instead of discarding it as NBP does.

The following toy examples demonstrate the differences between the Metropolis algorithm, Gibbs sampling and sequential Monte Carlo in multimodal and witch's hat-like conditions. The target distribution of example 1 is illustrated in Figure 2.4 and the posterior samples obtained using the various algorithms in a single run are shown in Figure 2.5. The estimated and true masses of the posterior modes are shown in Table 2.1.

As demonstrated in Figure 2.5, in this kind of situation MCMC algorithms tend to get stuck in one part of the posterior distribution, while SMC is able to

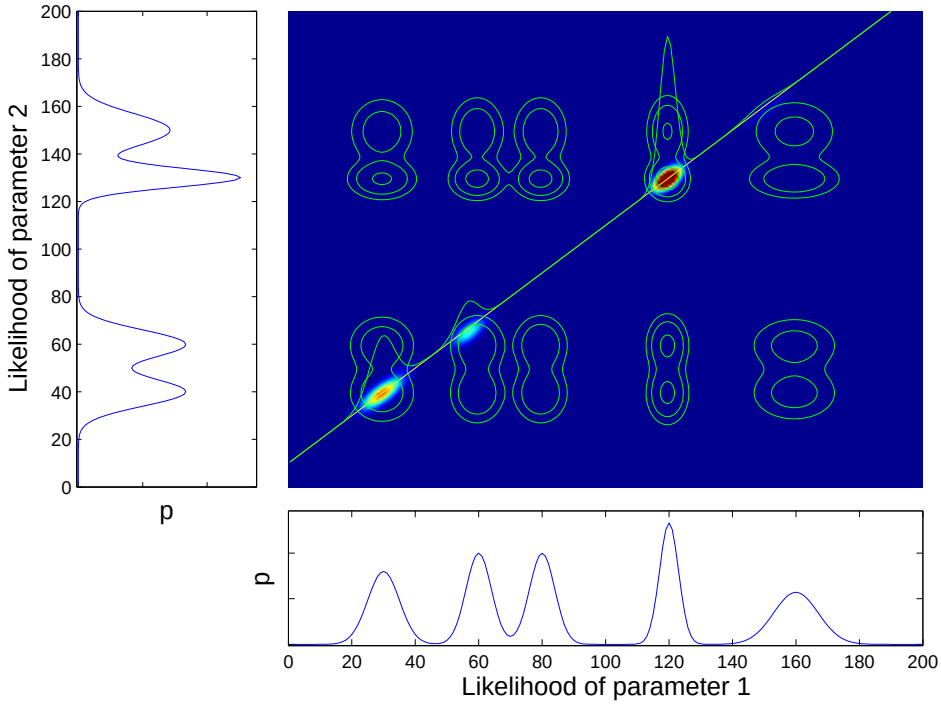

Figure 2.4: Toy example 1, target distribution. The contours show the joint likelihood as the product of the marginal likelihoods of the parameters while the tilted straight line is the prior mean. The peaks along the prior mean indicate the relative masses of the posterior modes.

Table 2.1: Toy example 1, posterior mode masses from a single run

	Peak 1	Peak 2	Peak 3	Peak 4
True mass	0.336	0.151	0.473	0.040
Metropolis	0.323	0.677	0.000	0.000
Gibbs	0.000	0.000	0.975	0.025
SMC	0.353	0.146	0.464	0.037

find all the modes of the distribution. Also, in SMC the masses of the modes are represented in correct proportions. It should be noted that both MCMC algorithms were initialized to the point $(100\ 100)^T$. Over repeated simulations, the Gibbs result stays the same, whereas Metropolis converges sometimes to peaks 1 and 2, sometimes to peak 3.

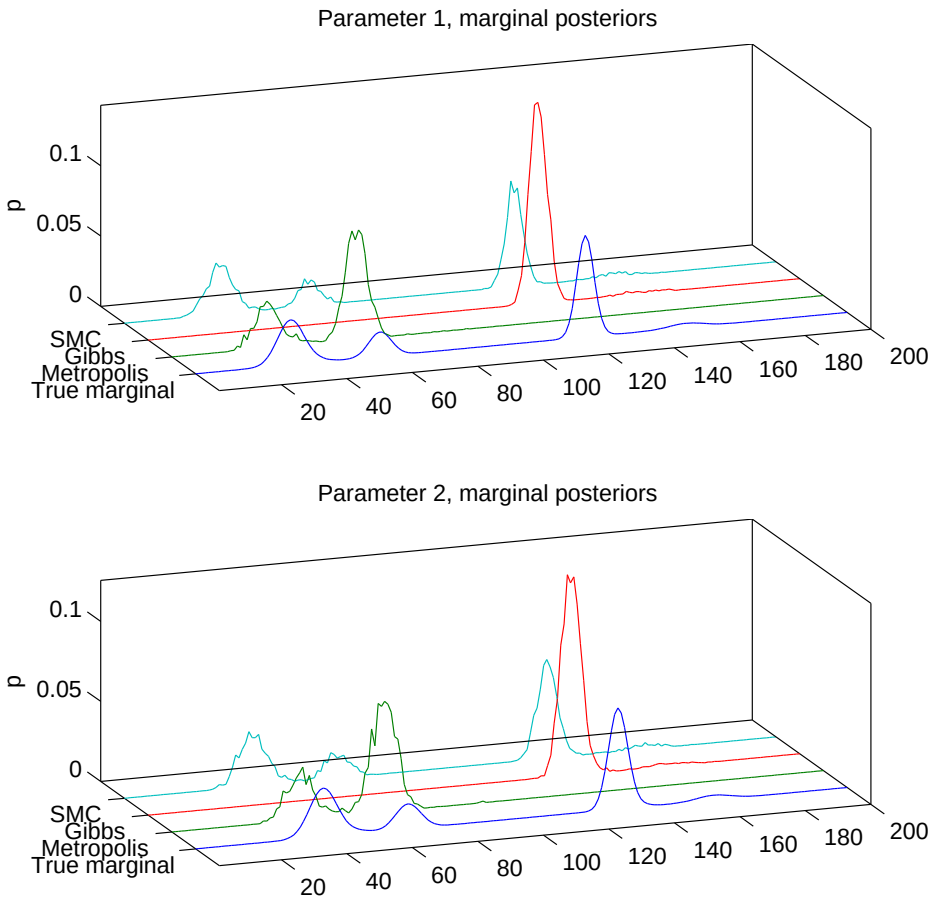

Figure 2.5: Toy example 1, sampling results from a single run. The plots show the posteriors estimated by the Metropolis algorithm, Gibbs sampling and Sequential Monte Carlo, as well as the true marginal posteriors. Both the Metropolis and Gibbs sampling algorithms have stayed in a certain part of the distribution, while the SMC particle representation contains all modes.

Toy example 2 illustrates the witch's hat distribution. Now the target distribution is a discrete lattice of size $256 \times 256 \times 256$, defined so that 95% of the probability mass is located in the area $[127, 129]^3$ and remainded is uniformly distributed over the rest of the lattice. Figure 2.6 illustrates the chains produced by a Gibbs and Metropolis sampler initialized in $(x_1\ x_2\ x_3)^T = (130\ 128\ 128)^T$, while Figure 2.7 shows the samples obtained with sequential Monte Carlo.

Even though two of the three parameters are initialized to the correct mode, Gibbs sampling requires close to 4000 iterations to converge, while Metropolis

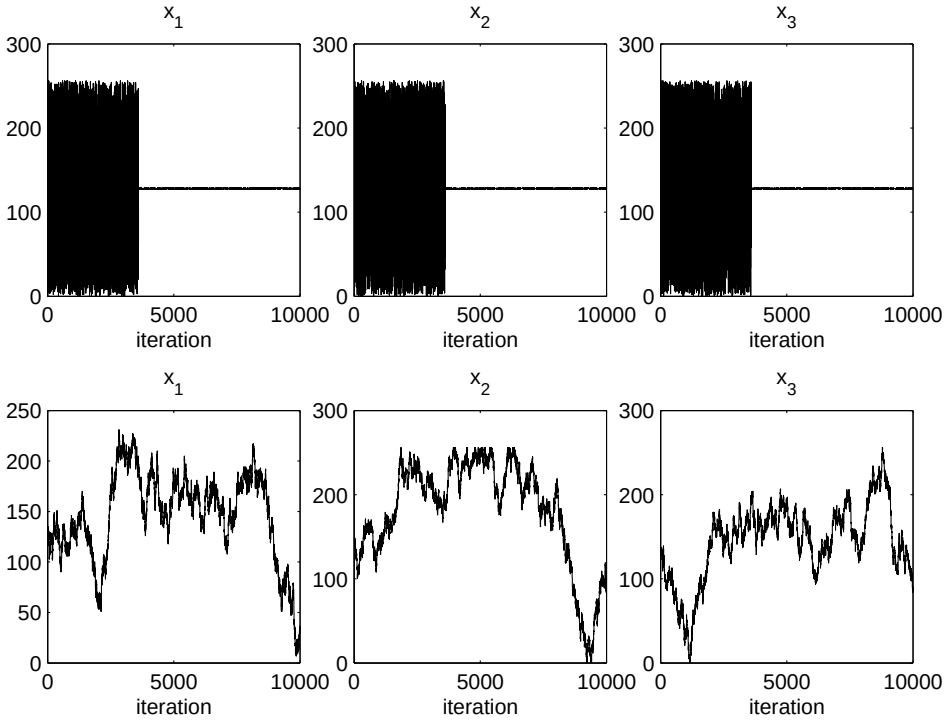

Figure 2.6: Toy example 2, sampling results from a single run, Gibbs and Metropols sampling. The images in the upper row show the samples of the three parameters produced by a Gibbs sampler, while the lower row shows the same for a Metropolis sampler. The Gibbs sampler requires almost 4000 iterations to converge to the main mode, while Metropolis never converges.

never converges. A SMC sampler run on the same distribution with 100 particles and no resampling results in 98 particles being located in the correct mode. Adding a resampling step eliminates the 2 spurious particles.

2.4 The object matching problem

In Chapter 1, the general object matching was phrased as "Given a set of examples of an object class, learn the fiducial properties of the class and use them to locate instances of the object class in a novel image", with a complete solution including the ability to handle multiple object classes. However, in this thesis the focus is on matching objects of a single class at a time and the multi-class problem

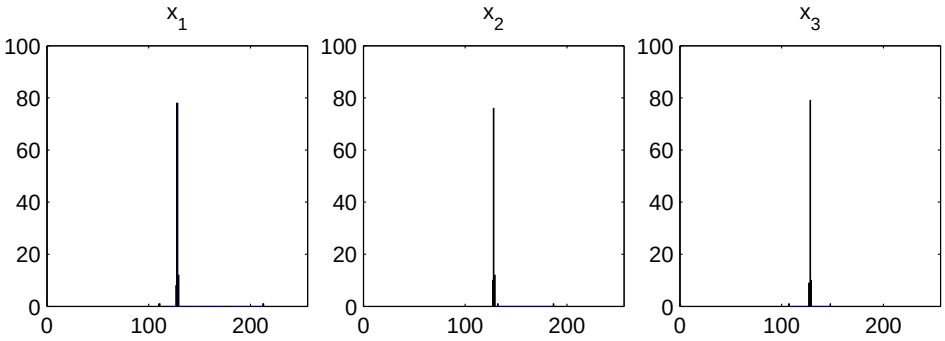

Figure 2.7: Toy example 2, sampling results from a single run, sequential sampler. The images depict the histograms of the particles produced by a simple SMC algorithm without resampling. Of the 100 SMC particles, 98 are located in the correct mode.

has been left to a further study. This section describes the basic object matching model in Bayesian terms. The first publication of the model is in (Tamminen and Lampinen, 2003a).

In this study, the feature-based approach is adopted: objects are represented as collections of certain fiducial features arranged in a planar graph. This is illustrated in Figure 2.8. The feature-based approach makes it natural to factorize the

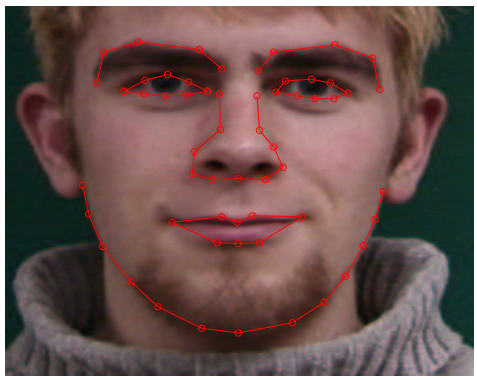

Figure 2.8: A sample feature grid. The target object class is the human face, and the target features are denoted by the circles.

problem into two separate parts: an appearance part, which models the visual appearance of the features, and the shape part, which models the relations between the feature locations, that is, the shape of the planar graph. As described in Chapter 1, this is very traditional in the object matching literature. Similar Bayesian

approaches to image analysis with distinct appearance and shape models have been proposed by Li et al. (2003) and Fergus et al. (2003), although their aim is classification of images according to the objects contained in them rather than the precise matching of the objects. Sullivan et al. (2001), Zhou et al. (2003) and Zhang et al. (2004) also use a similar factorization for non-feature based Bayesian object matching.

It should be noted that although Bayesian inference is a powerful tool for statistical learning, it is widely known that it is not all-powerful in the sense that only data that is consistent with the model can be handled. That is, if the prior probability for some hypothesis is set to zero, no observation can lead to the posterior probability of the hypothesis being non-zero. In the context of object matching this means that only the objects that are assumed to be in a scene are matched, and they are always matched. If the model assumes that there is a face in an image, and in the observed image there is no face but a chair, the face is matched somewhere nevertheless and the chair is ignored. There are methods for testing hypotheses such as Bayes factors (Bernardo and Smith, 1994), but these have their own limitations. Bayesian models are always a closed world; a complete Bayesian vision system would require a method for generating new hypotheses, which is beyond the scope of this study.

2.4.1 The basic matching model

As described in Section 2.2.1, according to the Bayesian point of view all observed and unobserved quantities are considered random variables following some distributions. With the appearance/shape factorization, it is now possible to define the variables of the model. As all natural images contain clutter and noise, to make the features of interest more distinctive in an image, the observed image $\mathcal{I}$ is transformed into a feature space $\mathbf{T}$, $\mathcal{I} \mapsto \mathbf{T}$, so that each image pixel $\mathcal{I}(u, v)$ has associated features $\mathbf{T}(u, v)$. Hence, the observed variables are the feature image $\mathbf{T}$, the training features $\mathbf{G}$, and the training shapes $\mathbf{Y}$. The unobserved variables are the locations of the N target features $\mathbf{x} = (x_1 \dots x_N)^T$ and any possible hyperparameters, denoted ξ. For notational simplicity, the model assumptions M have been left out of the following, but all distributions should be considered conditioned on them.

In Bayesian terms, the solution to the object matching problem is the posterior distribution of the feature locations given the image and the training data

$$p(\mathbf{x}|\mathbf{T}, \mathbf{G}, \mathbf{Y}) = \int p(\mathbf{x}, \xi|\mathbf{T}, \mathbf{G}, \mathbf{Y}) d\xi, \tag{2.32}$$

where the hyperparameters have been integrated out, as their actual values are not of interest.

Using Bayes' theorem and the rule of conditional probability, the joint poste-

rior distribution of the feature locations and the hyperparameters can be written as the product of the likelihood, prior and hyperprior parts:

$$p(\mathbf{x}|\mathbf{T}, \mathbf{G}, \mathbf{Y}) \propto \int p(\mathbf{T}|\mathbf{x}, \xi, \mathbf{G}, \mathbf{Y})\, p(\mathbf{x}|\xi, \mathbf{G}, \mathbf{Y})\, p(\xi|\mathbf{G}, \mathbf{Y}) d\xi. \qquad (2.33)$$

This can be simplified by some reasonable independence assumptions. Via the appearance/shape factorization, the training features $\mathbf{G}$ are assumed to affect only the likelihood and the training shapes $\mathbf{Y}$ only the prior. Furthermore, it is assumed that the possible hyperparameters only affect the prior, which leads to

$$p(\mathbf{x}|\mathbf{T}, \mathbf{G}, \mathbf{Y}) \propto p(\mathbf{T}|\mathbf{x}, \mathbf{G}) \int p(\mathbf{x}|\xi)\, p(\xi|\mathbf{Y}) d\xi = p(\mathbf{T}|\mathbf{x}, \mathbf{G})\, p(\mathbf{x}|\mathbf{Y}), \qquad (2.34)$$

where $p(\mathbf{T}|\mathbf{x}, \mathbf{G})$ is the image likelihood and $p(\mathbf{x}|\mathbf{Y}) = \int p(\mathbf{x}|\xi)\, p(\xi|\mathbf{Y}) d\xi$ the object shape prior, composed of the actual prior and hyperprior parts. The likelihood measures the probability of observing the feature image $\mathbf{T}$ given the feature locations $\mathbf{x}$ and the training features $\mathbf{G}$. The prior is actually the posterior predictive distribution of a new object shape $\mathbf{x}$ given the training shapes $\mathbf{Y}$.

2.4.2 The likelihood problem

It is relatively straightforward to produce the predictive distribution of the object shape given the training data (Chapter 4), but the likelihood is a different matter. In the strict statistical sense, measuring the likelihood of observing an image given the feature locations $\mathbf{x}$ would require a generative model for the image: as discussed by Sullivan et al. (2001), it is not enough to compute the image likelihood as a function of a hypothesis about the object location, since the likelihood should explain the appearance of the whole image, including the background in addition to the object $\mathbf{x}$. However, including a generative model for the background is not straighforward due to the unlimited variability of its appearance. Sullivan et al. (2001) use similar feature histogram models for the objects and the background, yielding directly comparable probabilities for assigning a part of the image to the object or to the background. This is applicable when the objects do not require complex structured models. With class-specific structured object models, such as the joint model of feature appearances and relative positions proposed in this thesis, it is practically impossible to have a generative model of the same detail level for the background. The decision whether to explain a part of an image with a (specific) object model or with a simpler and more generic background model is similar to model complexity selection, that is, the comparison between a complex model with small residuals and a simpler, less accurate model. For some object models the complexity can be controlled directly with suitable priors, leading to a MDL or MAP decision criterion (Mumford, 1996). For the object models ap-

plied in this work, it is not straighforward to compare the matching probabilities to ones from a generic background model. At this stage, therefore, the sub-optimal approach of ignoring the background has been chosen.

If the background is ignored, the likelihood can be approximated with any function S that measures the similarity between the observed image and the stored feature representation. It is possible to use either a joint representation for the features, such as a texture model (Section 3.4), or treat the features separately (Sections 3.2 and 3.3). The latter requires the simplifying assumption that the likelihoods of the transformed pixels T_i of the image are independent of each other and dependent only on the individual feature locations x_i and training features g_i pertaining to the ith feature, in which case the joint likelihood can be approximated with the product of the individual feature similarities:

$$p(\mathbf{T}|\mathbf{x}, \mathbf{G}) = \prod_i p(T_i|x_i, \mathbf{G}) = \prod_i p(T_i|x_i, g_i) \approx \prod_i S(\mathbf{T}(x_i), g_i), \qquad (2.35)$$

that is, the joint likelihood of observing an image given feature locations $\mathbf{x}$ is approximated by multiplying the similarities at the feature locations. If a joint representation is used, the likelihood $p(\mathbf{T}|\mathbf{x}, \mathbf{G})$ can be computed directly.

2.4.3 Implementing the model

Since a priori the target object and its features can be located anywhere in the perceived image, to obtain a full representation of the posterior the posterior density would have to be evaluated for all possible feature configurations, which is clearly impossible - for example, for 60 features, each with an u- and a v-coordinate, and 256×256 images there are $256^{2 \cdot 60}$ configurations. Instead, by using the sampling methods described in Section 2.3, it is possible to obtain samples from the posterior distribution and estimate the quanties of interest such as the mean from the samples.

In order to use the various sampling methods, the joint posterior distribution must be manipulated into suitable forms. The Metropolis algorithm can use the joint posterior (Eq. 2.34) directly, utilizing Eq. 2.35 to form the joint likelihood if necessary. Gibbs sampling requires the full conditional distributions of the single feature locations given the other features $\mathbf{x}_{\setminus i}$:

$$p(x_i|\mathbf{T}, \mathbf{x}_{\setminus i}, \mathbf{G}, \mathbf{Y}) \propto p(T_i|x_i, g_i)\, p(x_i|\mathbf{x}_{\setminus i}, \mathbf{Y}), \qquad (2.36)$$

whereas sequential Monte Carlo needs the conditional posteriors of singular feature locations given the previously matched features:

$$p(x_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}) \propto p(T_i|x_i, g_i)\, p(x_i|\mathbf{x_i}, \mathbf{Y}), \qquad (2.37)$$

where $\mathbf{x_i} = (x_1 \dots x_{i-1})^T$ denotes the features matched before the ith one. It should be noted that the requirement of conditional distributions for Gibbs sampling and SMC means, in practice, that the feature likelihoods must be treated separately and the object shape model must have a form from which it is possible to compute arbitrary conditional distributions.

As to the performance differences of the sampling algorithms described in Section 2.3.5, it can be said that the ability to perform in multimodal and witch's hat distribution-like conditions is the most critical one. Chapters 3 and 4 will demonstrate that due to clutter, occlusion and other factors, the target posterior distribution is in most cases extremely multimodal, with spurious peaks in multiple locations, and the occlusion model proposed in Chapter 5 results in the posteriors resembling a witch's hat distribution. The need for conditional and marginal distributions is hardly a problem due to the form of the appearance and shape models used in this study, and the ability to explore a single mode thoroughly, while important, is not very useful if the mode in question is not the one with the most probability mass. This subject will be discussed more extensively in Chapter 6.

Chapter 3

Models of visual appearance

3.1 Introduction

The Bayesian object matching model described in Chapter 2 requires a model for the visual appearance of the object that can be used to approximate the likelihood $p(\mathbf{T}|\mathbf{x}, \mathbf{G})$, which in turn requires the definition of an image-feature transform $\mathcal{I} \mapsto \mathbf{T}$. In principle the functional form of the model is irrelevant to the Bayesian formulation of the problem: any method that approximates the likelihood by measuring the similarity between the observed image and an object location hypothesis can be used. Of course, in practice there are large differences between methods with respect to accuracy, specificity and performance in clutter. This chapter describes several different models for the visual appearance of objects, ranging from simple template matching to texture approaches and sophisticated Gabor filter-based methods, including the model developed for this study.

3.2 Simple features

3.2.1 Template matching

Template matching is one of the oldest feature matching techniques (for a review, see Cox (1995)), with an even older history as matched spatial filtering in signal processing and optics (Casasent and Furman, 1977). Template matching methods use the actual training image data to represent the features of interest, that is, there is no explicit transformation $\mathcal{I} \mapsto \mathbf{T}$. In practice this means that for each feature, a template $\mathbf{q}$ is extracted from the training images via a rectangular or a Gaussian window and a function measuring the similarity between the template and locations in the target image is defined. Numerous similarity functions have

been proposed. The most common are the exponentiated squared error

$$e(\mathbf{q}, \mathcal{I}_A) = \exp(\sum_A w_i(\mathbf{q} - \mathcal{I}_A)^2), \tag{3.1}$$

where A is the area of the target image being compared to the template and w_i the window weights, and the normalized cross-correlation

$$r(\mathbf{q}, \mathcal{I}_A) = \frac{\sum_A w_i \mathbf{q} \cdot \mathcal{I}_A}{\sqrt{\sum \mathbf{q}^2 \sum_A \mathcal{I}_A^2}}, \tag{3.2}$$

where the vectors t and $\mathcal{I}_A$ are required to have zero mean. Template matching is illustrated in Figure 3.1.

The main disadvantage of template matching algorithms is their sensitivity to noise, especially non-white noise, and clutter, as well as intra-class variation associated with most types of features. In addition, as the templates are rather rigid, scale and orientation changes can also cause matching problems. However, the simplest form of template matching can be elaborated, which solves many issues - for example, Brunelli and Poggio (1995) build a "Synthetic Discriminant Function" from multiple templates, which results in greatly improved performance.

3.2.2 Corners and edges

A more robust method for extracting and matching features than using the image data directly is to associate the features with certain geometrical landmarks, such as edges and corners, and then find similar landmarks in the novel image. In terms of the object matching framework, the computation of the landmarks constitutes the transform $\mathcal{I} \mapsto \mathbf{T}$. The matching of corner points is especially popular in stereo vision research (Xu and Zhang, 1996). The standard procedure is to select points of maximum curvature as the features to be matched, and then find the correct match among all the corner points of the target image by utilizing simple or enhanced template matching and the epipolar constraint relating the two images (Brandt and Heikkonen, 2001). The corner points can be detected using a number of methods: either by utilizing specific corner location algorithms, such as the Harris corner detector (Harris and Stephens, 1988), or by analyzing the edge maps obtained using any standard edge detection algorithm, such as the standard simple line mask or derivative operators (Gonzalez and Woods, 1993) or the more sophisticated Canny edge detector (Canny, 1986). These are illustrated in Figure 3.2. Outside the stereo vision community corner point matching is not nearly as popular, as it can be applied only to features that are associated with a clear corner, which is relatively uncommon for many object classes.

Objects can also be matched by using edges or line segments as the pertinent

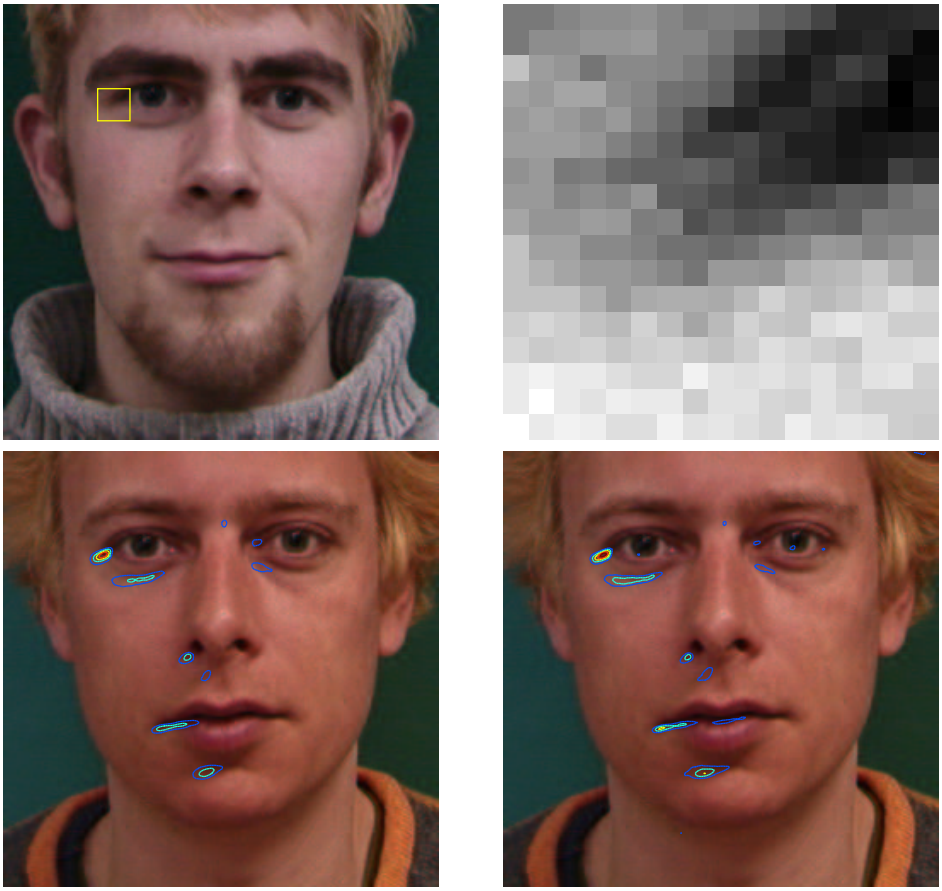

Figure 3.1: An example of template matching. The images in the upper row show the training image and the extracted 17×17 template (the yellow rectangle) while the images in the lower row show the similarity fields using the exponentiated squared error (left) and the normalized cross-correlation (right) similarity measures. The target feature has been localized well, although there are spurious matches. The difference between the similarity measures is very small.

features. The matching can be done individually or by using groups of lines or segments. Individual line segments are most often matched on their geometric attributes such as length and orientation (Zhang, 1994). The actual line features are again extracted with, e.g., the Canny edge detector (Figure 3.2). The matching of multiple line segments has the advantage of having more geometric information available at the cost of increased complexity. Different approaches here include graph-matching methods based on geometrical and topological relationships (Horaud and Skordas, 1989) as well as methods utilizing the intensity neighborhoods

of the lines, which can be considered a form of template matching (Schmid and Zisserman, 1997). Another type of edge features are the SIFT (Scale Invariant Feature Transform) features proposed by Lowe (1999). SIFT features, or keys, are located at the minima and maxima of a difference-of-Gaussian function applied in scale space, that is, over different resolutions. The keys are then characterized by the image gradient magnitude and orientation at the localized points. As will be seen in Section 3.3, this multiple-frequency processing is very similar to the Gabor filter approach employed in this work.

The main problem with edge and line features is their non-specificity: whereas it is seldom possible to associate a corner with a feature, it is nearly always possible to associate an edge, which leads to a large number of false matches. SIFT features are much more specific and hence more suitable for object matching, although the automatic selection of invariant points takes the choice of which features to match out of the hands of the user.

Figure 3.2: Corner and edge detection. In the image on the left, the crosses mark the corner points located by the Harris corner detector. The image on the right shows the edge map obtained with the Canny edge detector.

3.2.3 Using color

For some object classes, it is possible to use color to model the appearance of the object. The most prominent examples of this are the human face and other bare parts of the human body. Color research results indicate that the human skin forms a distinct cluster in the RGB color space that can be approximated with a Gaussian distribution (Yang et al., 1998) or a histogram model (Jones and Rehg, 1998), which makes it possible to match skin-colored features in images (Figure

3.3). Also, color can be useful in low-contrast images, such as discriminating the lips from the face (Blake and Isard, 1998). However, it is clear that color-based features are rather coarse, and more suitable for detection than precise matching. Color-based methods are often used to initalize a matching system, after which other kinds of features are employed to improve the matching and detection results (Hadid et al., 2002).

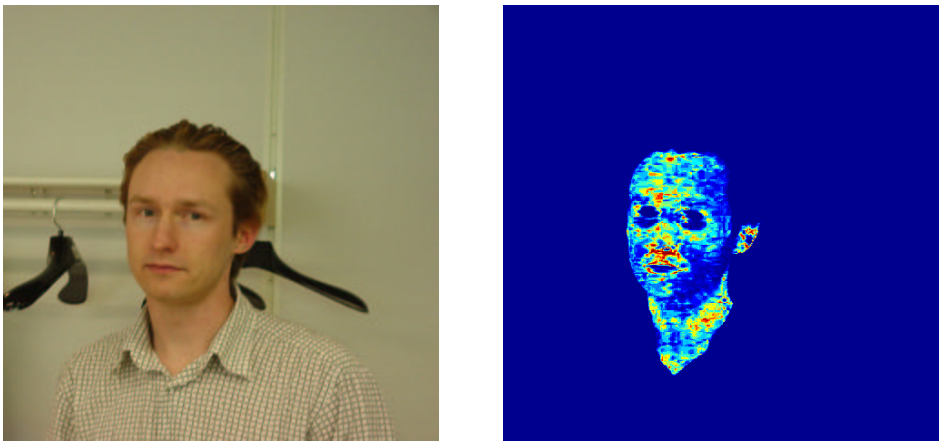

Figure 3.3: Skin color matching. The image on the right shows the result of building a Gaussian distribution in the color space from a set of skin colored patches and then computing skin color probabilities for the pixels of the image on the left.

3.3 Gabor features

The appearance models presented in Section 3.2 are too simple to deal with complex features found on objects such as human faces - a richer set of features is needed. A common solution is to use multiple digital filters, or a filter bank, to process the image and then combine the filter responses to produce the feature representations (Lades et al., 1993) (Wiskott et al., 1997). Sullivan et al. (2001) utilize downsampled convolution, i.e., the copies of the single filter do not overlap so that the filter responses are approximately independent, in accordance with their rigorous modeling of fore- and background likelihoods (see Section 2.4.2).

The basic filters of a bank are often edge detectors of some kind such as the derivative of Gaussian and Laplacian of Gaussian filters (Freeman and Adelson, 1991). In this study, the Gabor filter is used. Sections 3.3.1-3.3.4 describe the Gabor filter and its characteristics and present the feature model developed for this work.

3.3.1 The Gabor filter

Gabor filters are direction-sensitive edge detectors well-suited to feature matching tasks (Daugman, 1988) (Wiskott et al., 1997). The impulse response of the Gabor filter can be summarized as a complex sinuous waveform restricted by a Gaussian window, with the filter parameters controlling the width of the filter and the number of side lobes (see Eq. 3.3). The filters used in this study contain few side lobes, as an increasing number of side lobes narrows the passband of the filter, which in turn makes it necessary to have a large number of filters to cover the frequency space. Furthermore, side lobes tend to cause ripple in the filter response. The real and imaginary parts of the impulse response of an 1D Gabor filter are shown in Figure 3.4.

For image processing a two-dimensional filter is needed, which, correspondingly, is a complex plane wave restricted by a Gaussian envelope,

$$h(u,v) = \frac{f^2}{2\pi\sigma_u\sigma_v}\exp\left(i(f\cos(\theta)u + f\sin(\theta)v)\right)\exp\left(-\frac{f^2}{2}(\frac{u^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2})\right), \tag{3.3}$$

where f is the central frequency of the filter, θ the directional angle of the filter and σ_u^2 and σ_v^2 the spatial-domain variances of the filter. The coefficient $\frac{f^2}{2\pi\sigma_u\sigma_v}$ compensates for the frequency-related decrease of the power spectrum in natural images (Daugman, 1988). The Fourier transform or the transfer function of Eq. 3.3 is the Gaussian

$$H(\omega_u,\omega_v) = \exp\left(-\frac{1}{2f^2}(\sigma_u^2(\omega_u - f\cos(\theta))^2 + \sigma_v^2(\omega_v - f\sin(\theta))^2)\right), \tag{3.4}$$

from which the role of the central frequency can clearly be seen. The filter can also be interpreted as a two-dimensional wavelet with the u- and v-coordinates as the translation and $(\frac{\sigma}{f})^2$ as the wavelet scaling parameter. Figure 3.5 illustrates the 2D Gabor filter.

From Eq. 3.4 it can be seen that the basic Gabor filter has a nonzero mean or, equivalently, a nonzero DC component. In image analysis this is generally undesirable, as a nonzero mean causes the response of the filter to be dependent on the mean grayscale level of the processed image. The DC component can be removed by calculating it via Fourier transform and then subtracting a suitable Gaussian from the impulse response of the basic filter (the Gaussian should have the same shape as the original transfer function but with zero frequency and the DC component as the amplitude). This leads to the impulse response

$$h(u,v) = \frac{f^2}{2\pi\sigma_u\sigma_v}\exp\left(-f^2(\frac{u^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2})\right)\cdot$$
$$\left(\exp(i(f\cos(\theta)u + f\sin(\theta)v)) - \exp(-\tfrac{1}{2}(\sigma_u^2\cos^2(\theta)+\sigma_v^2\sin^2(\theta)))\right), \tag{3.5}$$

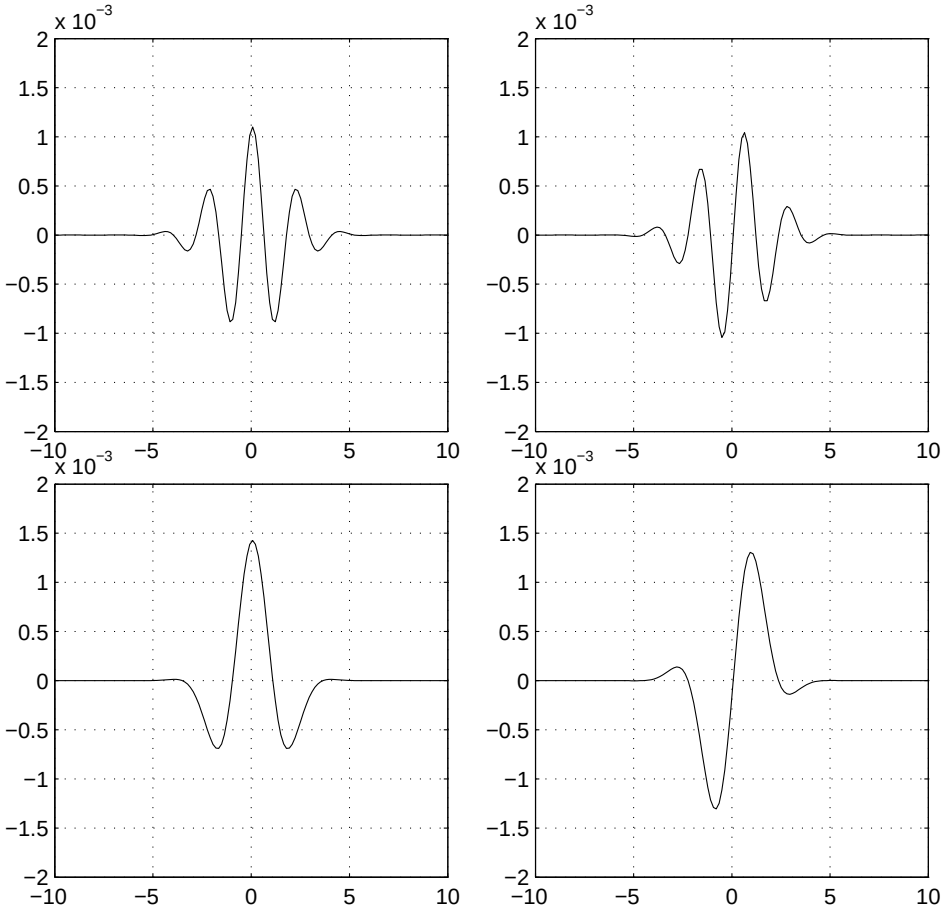

Figure 3.4: The real and imaginary parts of a one-dimensional Gabor filter. The upper images illustrate a typical Gabor function, with multiple side lobes, while the images below show the type of Gabor filter used in this study.

illustrated in Figure 3.6. The resulting filter is no longer a Gabor filter, but resembles it closely.

3.3.2 Properties of the Gabor filter

Gabor filters have certain properties that make them a particularly enticing choice as an image processing tool. From the purely mathematical point of view, the complex-valued family of 2D Gabor filters is conjointly optimal in both spatial and frequency resolutions (Daugman, 1988). That is, the passband of the filter is optimally localized in terms of the width of the passband and the extent of

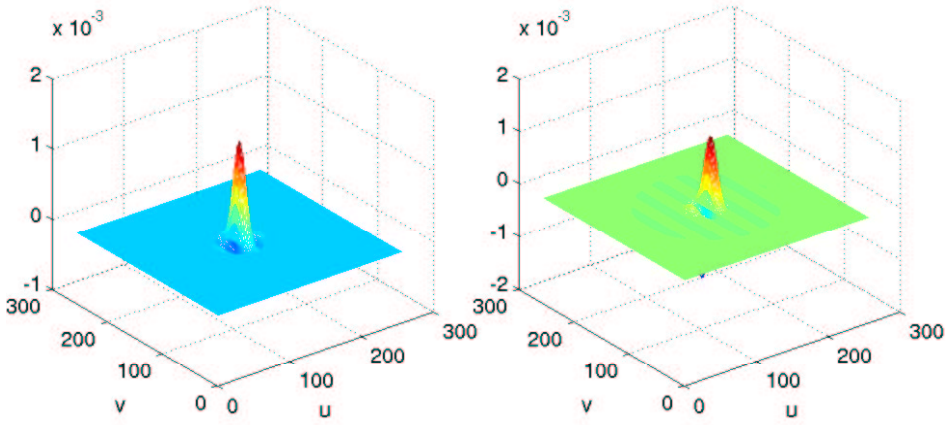

Figure 3.5: The real and imaginary parts of the impulse response of a two-dimensional Gabor filter.

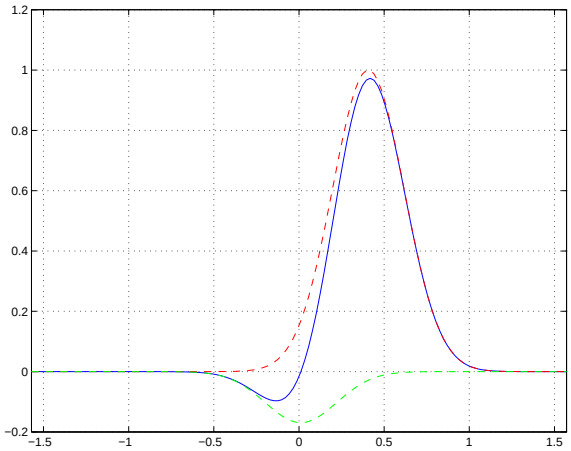

Figure 3.6: The Gabor filter transfer function with the DC component removed. The blue line represents the altered transfer function, while the red line is the original function and the green line the subtracted Gaussian.

the impulse response. This makes the filter suitable for extracting local spatial frequency contents from as small areas as possible. Furthermore, it can be shown (Mallat, 1998) that the Gabor filter uniquely achieves the theoretical lower bound of uncertainty over the four spatial and frequency variables, namely

$$\Delta u^2 \Delta v^2 \Delta \omega_u^2 \Delta \omega_v^2 = \frac{1}{16\pi^2}. \tag{3.6}$$

In pattern recognition and computer vision, Gabor filters are used as the feature spaces produced by them are distortion tolerant (Lampinen and Oja, 1995), making them suitable for feature matching. This is largely due to the limitations in the spectra of natural images as compared to noise (Daugman, 1988). It should be noted, though, that the family of Gabor filters is non-orthogonal. In general filter bank theory orthogonal filter banks are preferred, i.e., ones consisting of filters with zero inner products between them, as this ensures that any signal can be reconstructed perfectly and uniquely as the sum of the filter responses. However, in pattern recognition and feature extraction nonorthogonal filters are often used, since it would seem that the orthogonality is not required for good performance (Daugman, 1988).

As mentioned in Chapter 1, one of the major aims of computer vision is to emulate the human vision system. There is ample evidence that the mammal brain employs Gabor-like filtering as a part of its low-level image processing (Jones and Palmer, 1987), and hence it makes sense to explore the possibilities of similar structures in artificial solutions to the problem. The primary cortical area for vision, the striate cortex, is the single largest cortical area in the human brain, and it is where the first stage of human visual processing occurs. The mapping of the receptive fields of the striate cortex cells (Palmer, 1999) has shown that they contain multiple lobes of excitation and inhibition, i.e., the receptive fields look like Gabor functions. This is an indication that these cells may be performing a local spatial frequency analysis of incoming images. Furthermore, there is evidence that different cells are tuned differently to spatial frequency and orientation, forming a filter bank similar to the ones used in this study, although the spatial extent of these filters is smaller than of the ones used here.

3.3.3 Gabor filter banks

The Gabor filter impulse response (Eq. 3.3) actually defines a whole family of Gabor filters, each with its own center frequency f and directional angle θ. Each of these filters functions as an edge detector in the direction orthogonal to the angle parameter and on the frequency band specified by the center frequency. For the matching of complex features each of these is inadequate on its own (cf. Section 3.2.2), but by specifying a number of different center frequencies $f_k, k \in \{1..C_f\}$ and directional angles $\theta_l, l \in \{1..C_\theta\}$, a filter bank of several Gabor filters can be constructed. A sample Gabor filter bank is illustrated in Figure 3.7 in the spatial domain and in Figure 3.8 in the frequency domain. The filters overlap in the Fourier domain both in the frequency and orientation directions to produce somewhat correlated outputs.

The Gabor filter bank constitutes the transform $\mathcal{I} \mapsto \mathbf{T}$. When filtered with the bank, an image $\mathcal{I}$ of size $C \times C$ is transformed into an array of size $C \times C \times C_f \times C_\theta$,

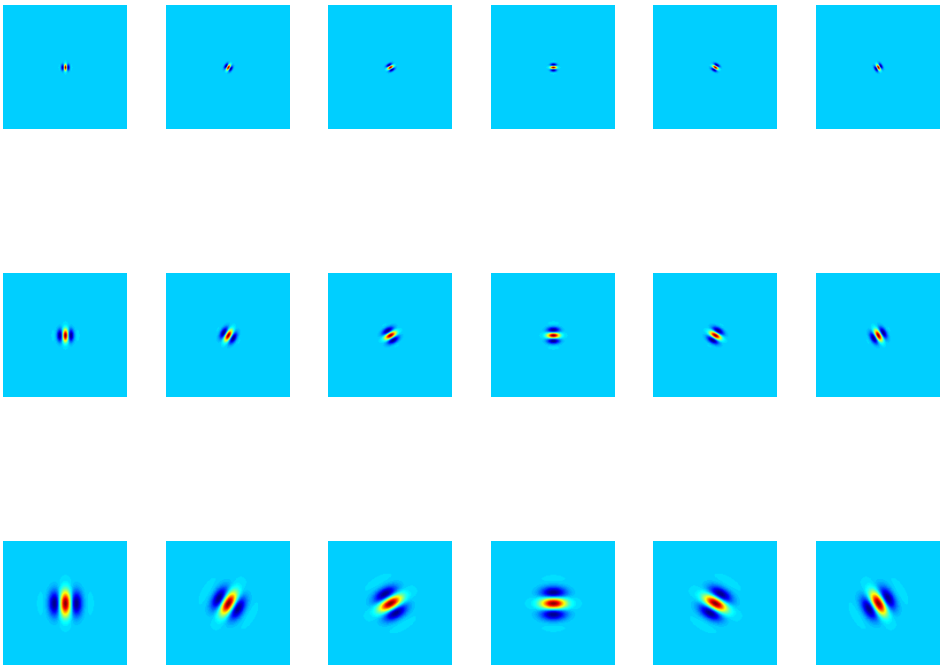

Figure 3.7: The real parts of the impulse responses of a Gabor filter bank with 3 frequencies and 6 orientations.

consisting of matrices

$$\mathbf{T}_{(k,l)} = \mathcal{F}^{-1}(\mathbf{H}_{(k,l)}\mathcal{F}(\mathcal{I})), \tag{3.7}$$

where $\mathbf{T}_{(k,l)}$ is the response of the filter with central frequency f_k and directional angle θ_l, $\mathbf{H}_{(k,l)}$ is the transfer function of the filter, and $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse. The transform can also be computed via convolution.

3.3.4 Interpretation and modeling of Gabor filter responses

Processing an image with a Gabor filter bank produces an array (Eq. 3.7) of complex filter responses. The amplitude and phase responses of the various filters are the corresponding complex magnitudes and arguments. An example of a Gabor filtered image is shown in Figure 3.9.

Gabor filters act as edge and feature detectors in their specific alignments, i.e., a vertical filter with a zero directional angle will produce high amplitude responses in the neighborhoods of the vertical edges in the image. The phase re-

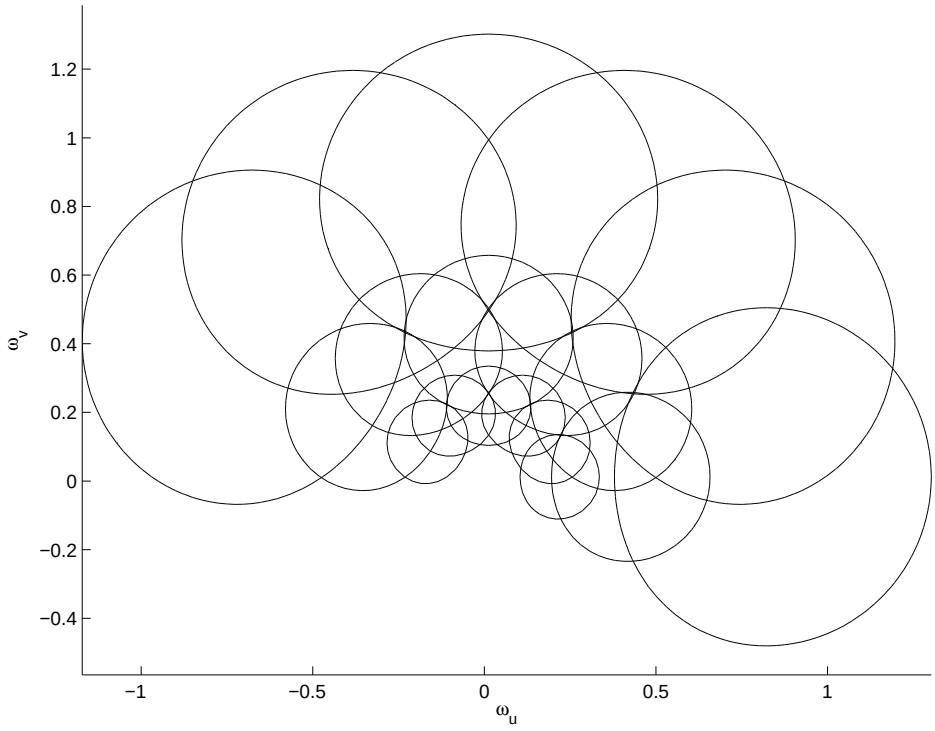

Figure 3.8: The filter bank of figure 3.7 in the frequency domain. Note how the contours of the filters are not exactly circular, as would be the case with the standard Gabor filters with the DC component included.

sponse further specifies the locations of the edges - as can be seen by computing the transforms, on any edge where the transition is from low grayscale values to high grayscale values, the phase response is $-\frac{\pi}{2}$, and $\frac{\pi}{2}$ on the edges with the opposite transition. Thus the primary areas of importance in the picture ("where something happens") can be located by inspecting the amplitude responses, while the phase response further specifies the locations of the details. This is illustrated in Figure 3.10. Besides acting as edge detectors, the amplitude and phase responses also clearly contain information about distance to the edge.

The responses of a Gabor filter bank can be combined by stacking them as jets (Koenderink and van Doorn, 1987) (Lades et al., 1993). The appearance of the features to be matched can then be represented with these jets, that is, feature appearance is specified by the distance of the feature to edges of various frequencies and orientations. The mathematical representation of a jet is a high-dimensional vector - for example, if three frequency resolutions and six orientations are used, the jets are 36-dimensional vectors, as each filter has both amplitude and phase re-

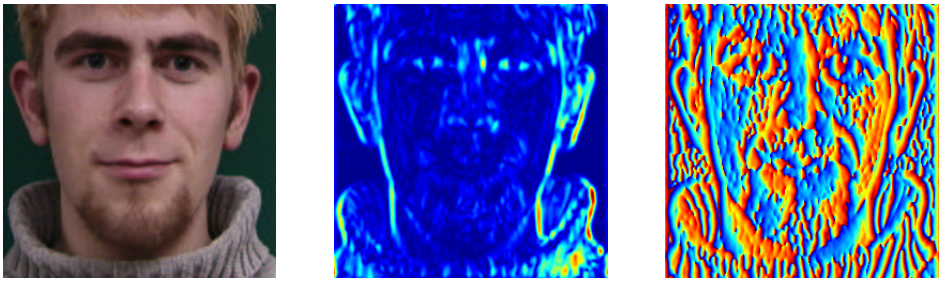

Figure 3.9: The amplitude (center image) and phase responses (right image) of an image filtered with a vertical ($\theta = 0$) Gabor filter. Bright red areas correspond to high amplitudes or phases close to π, while dark blue areas correspond to low amplitudes or phases close to $-\pi$.

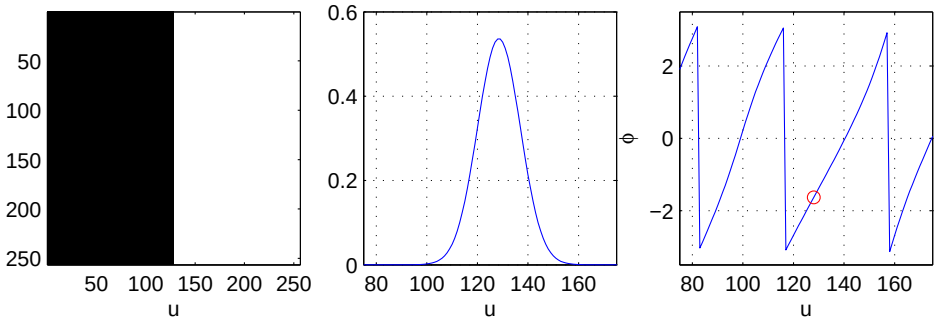

Figure 3.10: The Gabor filter applied to a binary test image. From left to right: the test image, the amplitude response and the phase response. In the phase response image, the circle marks the correct location of the edge - the phase at this point is $-\frac{\pi}{2}$.

sponses. An alternative representation is the simple Gabor feature space of Kyrki et al. (2004), in which the filter responses are arranged as feature matrices instead of jets. This has the advantage that rotations and scale changes correspond to matrix column and row shifts, which makes invariant matching more straightforward.

To find the locations of the features of an object in an image, the training jets **G** and the perceived jets **T** have to be compared, which requires that the appearance of the training features must be represented as jets, as well. A prominent method for jet comparison has been presented by Wiskott et al. (1997) as a part of the Elastic Bunch Graph Matching system. In their method, the similarity between

the complex Gabor jets g and g' is defined as

$$S(g, g') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \mathbf{dk}_j)}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}, \tag{3.8}$$

where a_j are the amplitudes and ϕ_j the phases of the complex Gabor jets, $\mathbf{k}$ is the spatial frequency or the characteristic wave vector of the corresponding Gabor filter and $\mathbf{d}$ is an estimated displacement vector, which is used to compensate for the rapid variance of the phase for small displacements in the image and can be estimated by maximizing the similarity in its Taylor expansion (Wiskott et al., 1997). The training jets $\mathbf{G}$ are all included in the appearance representation so that the perceived jets are compared against all the training jets and the best fitting training jets are selected and used for the final comparison. This increases the robustness of the model as the variability of the stored features increases, but the computational cost also rises. Furthermore, if there are errors in the annotation of the training data, the resulting outliers can hamper matching performance significantly.

As in this study the underlying formulation of the object matching problem is probabilistic, it is natural to model the jets statistically. Two statistical models have been presented in (Tamminen and Lampinen, 2003a) and (Tamminen and Lampinen, 2003b). In (Tamminen and Lampinen, 2003a), the amplitude jets were considered to form a linear subspace in the jet space and similarity between the amplitude component $\mathbf{A}$ of the perceived jet $\mathbf{T}(u, v)$ and the subspace defined by the projection matrix $\mathbf{P}_i$ pertaining to the ith feature was measured by computing the angle between the jet and the subspace:

$$S_{amp}(\mathbf{T}(u, v), \mathbf{P}_i) = \arccos\left(\frac{\mathbf{A}(u, v)^T \mathbf{P}_i \mathbf{P}_i^T \mathbf{A}(u, v)}{||\mathbf{A}(u, v)||||\mathbf{P}_i \mathbf{P}_i^T \mathbf{A}(u, v)||}\right), \tag{3.9}$$

The phase responses were handled separately with a square error model, and a term for Gabor jet energy (Eq. 3.11) was also included.

The subspace model of Eq. 3.9 only produces the similarity between the observed features and the training data. A true probabilistic appearance model should measure the likelihood of observing an image pixel given that the feature under scrutiny is located there. The feature appearance model used in this study, first presented in Tamminen and Lampinen (2003b), is such a model. In this probabilistic model the amplitude and phase jets, as well as the jet energy component are all assumed to follow Gaussian distributions. The multimodal approach of Elastic Bunch Graph Matching is probably a more realistic one; furthermore, the NBP approach of Sudderth et al. (2003) also shows non-Gaussian patterns in their chosen subspace features, which are principal component coefficients. Here, in order to have a true likelihood model for the jets, the Gaussian distribution was

chosen for simplicity.

For the N features to be matched, the distribution parameters $\mathbf{G} = (g_1 \ldots g_N)^T$ are determined by measuring the jets at the feature locations in a set of manually pre-annotated objects. However, it is not desirable to model the jets in the Gabor jet space as such. For example the amplitude response is highly sensitive to contrast, which makes the model highly sensitive to lighting conditions, if the modeling is done in unaltered amplitude space. This problem is addressed by transforming the amplitude and phase jets into hyperspherical angle coordinates:

$$h_i = \arctan \frac{c_{i+1}}{\sqrt{\sum_{j=1}^{i} c_j^2}}, \tag{3.10}$$

where h_i is the ith hyperspherical coordinate and c_i is the ith component of the jet vector. Note that this is just the generalization of 3-dimensional spherical coordinates into multiple dimensions (Hassani, 1998). On the other hand, neither total contrast-independence is desirable in an appearance model, as it causes sensitivity to faint patterns and noise in uniform areas. Hence, a Gaussian term measuring the energy of the Gabor jet is added to the model. This term is the sum of the squares of the jet lengths over a quadrangular area near the feature:

$$G_{\text{ene}}(u, v) = \sum_{i=u-m}^{u+m} \sum_{j=v-m}^{v+m} ||\mathbf{A}(i, j)||^2, \tag{3.11}$$

where $2m+1$ is the side of the chosen square (typical value is $m = 4$ for 256×256 images).

In practice the covariance matrices of the jet distributions can not be estimated directly due their high dimensionality. For a filter bank with 3 frequencies and 6 orientations, the distributions are 17-dimensional, as 1 dimension is eliminated in the transform to angle coordinates, and the covariance matrices have $17 \cdot 18/2 = 153$ free parameters. In most cases there is not enough training data to estimate so many parameters robustly, and hence the model has to be regularized so that it is possible to invert the covariance matrices. This is done in the usual manner by adding a constant ridge term ϵ_G to the diagonals of the estimated covariance matrices Σ^{est} so that the covariance matrices become (Bishop, 1995)

$$\Sigma = \Sigma^{est} + \epsilon_G \mathbf{I}. \tag{3.12}$$

This ridge parameter controls the steepness of the likelihood function.

By combining the amplitude, phase, and energy components, the likelihood

of observing a pixel represented with the transformation $T_i(u, v)$ is obtained:

$$p(T_i(u, v)|g_i) = \\ \mathrm{N}(G_{\mathrm{amp}}|\mu_{\mathrm{amp}}, \Sigma_{\mathrm{amp}}) \cdot \mathrm{N}(G_{\mathrm{pha}}|\mu_{\mathrm{pha}}, \Sigma_{\mathrm{pha}}) \cdot \mathrm{N}(G_{\mathrm{ene}}|\mu_{\mathrm{ene}}, \sigma^2_{\mathrm{ene}}), \quad (3.13)$$

where N is the Gaussian density function, G_{amp}, G_{pha} and G_{ene} are the amplitude, phase and energy Gabor jet properties corresponding to $\mathbf{T}(u, v)$, and μ_{amp}, μ_{pha}, μ_{ene}, Σ_{amp}, Σ_{pha} and σ_{ene} the means and regularized (co)variances of the distributions corresponding to the ith feature. The likelihood is illustrated in Figure 3.11.

Figure 3.11: Sample feature likelihood fields. The target features are the outer corner of the leftmost eye (left image), the point between the nostrils (center image) and the tip of the chin (right image). Note how the fields are clearly multimodal, with peaks in multiple distinct locations. Compare the left image with Figure 3.1, which depicts the template matching similarity for the same feature. Here there are much fewer false matches.

The presented pixel likelihood measure closely resembles the Wiskott et al. (1997) similarity measure (Eq. 3.8). Both of them are rather *ad hoc*, but this is the case for all similarity measures: as discussed in Section 2.4.2, for complex object models it is probably impossible to build a comprehensive generative image model, and hence one has to settle with engineering solutions such as these. All proposed measures of feature similarity lead to similarity fields with multiple modes (as seen in Figure 3.11) - if a measure is so specific that only the correct locations yield nonzero values of similarity, it will not be able to generalize to novel images. In most visual features there exist both considerable variations between examples of the same feature and similarities between examples of different features, and eliminating all false matches will lead to correct matches being eliminated as well. On a related note, neither Eq. 3.8 nor Eq. 3.13 takes into account the effect of background on the jets (cf. (Würtz, 1997)), which can cause matching errors in areas close to the edges of the target object, as the filter responses are affected by the arbitrary background. However, this problem is alleviated if the features are located directly on top of the edges, as then the background effects

are small in comparison to the large-magnitude response from the edge.

To assess its performance, the proposed likelihood (Eq. 3.13) was compared with Eq. 3.8 by using the IMM-DTU annotated face image database (Stegmann, 2002). Three distances from the selected feature locations were computed: distance from the local maximum of the similarity distribution closest to the correct location, distance from the maximum in a rectangular area around the correct location, and distance from the global maximum. The proposed model was formed in a leave-one-out cross-validation way (one image was left out and the model formed with the others), while Eq. 3.8 was evaluated with respect to one, three, and five randomly chosen images, as well as all 36 images. The largest similarity for each pixel was chosen, which is similar to the Elastic Bunch Graph Matching framework. With one reference image the computational complexity was approximately equivalent between the methods, while with more images the computational load with Eq. 3.8 was considerably heavier, as its computation time is linear in the number of training images. The results were averaged over features and images, and are shown in Table 3.1.

Table 3.1: Comparison of feature similarity measures. PM refers to Eq. 3.13, while Wx refers to Eq. 3.8 with x training images.

Error measure	PM	W1	W3	W5	W36
Local maximum	2.87	5.23	4.08	3.57	2.57
Maximum in local area	5.07	5.51	5.25	5.20	4.92
Global maximum	43.9	58.2	51.7	48.2	41.6

The accuracy of the methods is very similar. Eq. 3.8 outperforms the proposed method slightly when a large number of training images is used, at the cost of greater computational burden. With fewer training images, the proposed model is slighly more accurate than Eq. 3.8. The main advantage of the Bunch Graph approach over the proposed one is its capability to include greatly varying examples in the training data. As in the proposed model the feature distributions in the feature space are Gaussian, the training images must be at least somewhat similar to each other. For example, it is not possible to combine photorealistic training images with stylized drawings, as this would lead to the Gabor filter response distributions becoming too wide or even multimodal, making the Gaussian distribution inapplicable. On the other hand, as mentioned before, outliers in the training data affect Eq. 3.8 much more than the proposed likelihood. A statistical feature model which would be able to handle diverse images could be built by introducing similarity function parameters describing the image type (photo / drawing / comic etc.), lighting conditions and such. In practice this would probably lead into multiple similarity categories so that the type of the image would be

estimated first, after which the feature model corresponding to the estimated type would be applied. Another, a more straightforward possibility would be to use a mixture of Gaussians to model the filter responses, which would move the proposed system in the direction of Elastic Bunch Graph Matching. Such a Gaussian-mixture based feature appearance model has been proposed by Kämäräinen et al. (2005), with good results in face matching as a part of a recent face localization system (Hamouz et al., 2005).

3.4 Texture models

A common drawback with the numerous feature appearance models detailed in Sections 3.2 and 3.3.1 is that they only model the neighborhoods of the selected fiducial features and ignore the appearance of the object elsewhere. Although it would be possible to select each pixel of an object as a feature and model the appearance of each separately, the computational cost would quickly become prohibitive. Another proposal to joint modeling of the appearance of all the pixels of an object is to form a single parametrized model for object texture.

Texture modeling is a subject of great interest both in itself (Jain and Farrokhnia, 1990) (Zhu et al., 1997) as well as in combination with image segmentation (Tu and Zhu, 2002). In object recognition and matching, the best known texture-based methods are Eigenfaces (Pentland et al., 1994) and Active Appearance Models (Cootes et al., 2001). In the Eigenface framework, object appearance is modeled as a combination of a mean appearance vector and linear base vectors, learned via principal component analysis (PCA) (Chatfield and Collins, 1995):

$$\mathbf{V} = \mu_{text} + \mathbf{P}\beta_t, \tag{3.14}$$

where $\mathbf{V}$ is the texture vector (list of pixel values), μ_{text} is the mean texture, $\mathbf{P}$ is the set of base vectors, and β_t is the vector of object-specific coefficients. The training objects must be normalized so that background effects are not included in the representation. A set of Eigenfaces is illustrated in Figure 3.12.

An Eigenface model can be used for synthesizing and matching faces directly with an intensity difference error measure (cf. Eq. 3.1). The method has been developed further by separating intra- and extrapersonal variability and including a probabilistic similarity measure (Moghaddam et al., 1998). However, a major weakness of the method is its inseparable coupling of appearance and shape, which makes Eigenfaces rather rigid from the modeling point of view. Active Appearance Models by Cootes et al. (2001) rectify this problem by modeling the appearance and shape separately, as will be discussed in Chapter 4.

Figure 3.12: Examples of Eigenfaces. Training image shapes from the IMM-DTU database (Stegmann, 2002) were normalized and backgrounds eliminated. The first image on the left shows the mean texture and the following 5 columns depict the sum of the mean shape and the first five principal components, with positive coefficients in the top row and negative coefficients in the bottom row.

Chapter 4

Object shape models

4.1 Introduction

To form a representation of the whole object in feature based object matching, the locations of the features have to be linked together. This is done by including a model for the shape of the object in the matching framework. Again, as was the case with the models of feature appearance, any model which assigns probability densities for different feature configurations can be used, although some of the sampling methods described in Chapter 2 require that it is possible to compute the conditional distributions of the locations of single features given a set of other features. In this chapter the shape models used in this study are described and compared with the ones presented in literature.

4.2 Grid and graph models

The simplest possible object shape model is to consider the feature locations $\mathbf{x}$ a planar graph and define a reference graph corresponding to the expected shape of the object class under scrutiny and a cost function to measure deviation from this reference. Often, square error is used as the cost function and the deviations are assumed independent of each other. For example, Wiskott et al. (1997) use square error, although in EBGM all edges of the graph are considered instead of just the nodes. Minimizing the sum of squared errors is equivalent to assigning a Gaussian prior on the location of the features and maximizing the log-posterior. Hence, a sensible first prior is of the Gaussian form:

$$p(\mathbf{x}|\mu_{shape}, \sigma^2) = \mathrm{N}(\mathbf{x}|\mu_{shape}, \sigma^2\mathbf{I}), \tag{4.1}$$

where μ_{shape} is the reference graph and σ determines the extent of allowed deviations from the reference graph. The reference graph can be defined manually

or it can be learned from a set of training images (see Section 4.3). If the reference graph has a zero mean, also the current coordinates $\mathbf{x}$ must have their mean removed before the prior is evaluated.

For the Gibbs and sequential Monte Carlo sampling systems described in Chapter 2, conditional priors are needed. When the deviations are assumed to be independent of each other, the conditional prior on the location of feature i is simply

$$p(x_i|\mathbf{x}_{\setminus i}, \mu_{shape}, \sigma^2) = \mathrm{N}(x_i|\, f(\mu_{shape}, \mathbf{x}_{\setminus i}), \sigma^2\mathbf{I}), \tag{4.2}$$

where f is a function that computes the expected location of the ith feature by translating the graph according to features $\mathbf{x}_{\setminus i}$. That is, f computes the position and the scale of the graph given by $\mathbf{x}_{\setminus i}$, translates and scales the corresponding features of the reference graph to match these, and returns the reference position of feature x_i. This prior is illustrated in Figure 4.1.

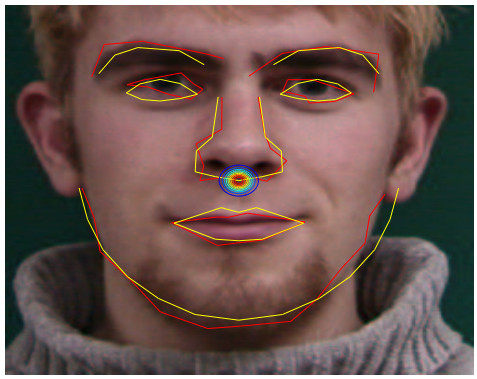

Figure 4.1: The simple grid prior. The target feature is the point between the nostrils. The red grid depicts the locations of the features $\mathbf{x}_{\setminus i}$, while the yellow grid marks the translated and scaled reference grid. The contours depict the prior.

The parameter σ defines the elasticity of the feature graph: small values force the grid to follow the reference grid closely, whereas with large values the prior becomes uninformative and the posterior distribution is dominated by the likelihood. Naturally, this makes the selection of σ very important. This can be done by guessing or, for example, with cross-validation (Hastie et al., 2001). The value of σ can also be determined automatically from the data via a hierarchical prior structure, which is discussed in Section 4.2.1.

As described in conjunction with non-parametric belief propagation, it is also possible to describe object shapes as graphical models so that the features are located at the nodes of the graph and the edges of the graph are defined by connecting the features in some fashion. In this case the shape is controlled by setting

priors on various parameters of the graph; for example, Sigal et al. (2003) model the human body with a set of parameters for cylinders representing various body parts and the angles between them. Zhang et al. (2004) utilize a similar model in feature based human body matching. In a sense, the simpler graph model is more general than graphical models because graphical models must be separately specified for all possible object classes, whereas for the simpler graph models it is enough to annotate the feature locations. Furthermore, the explicit interactions between non-adjacent nodes are often ignored in graphical models, which can be considered a waste of information. On the other hand, it is easier to add more nodes to a graphical model than to a planar graph; a graphical model requires only that the definition of the connection to the existing graph, whereas a feature graph model requires that the relations to all existing features are defined.

4.2.1 The hierarchical graph prior

The hierarchical prior used in this work differs from the simple graph prior of Section 4.2 in that the value of σ is not fixed, but instead a prior distribution is defined for it as well in line with accordance with Section 2.2.3. The other assumptions of Section 4.2 still stand, i.e., the deviations of the feature locations from their expected values are still assumed to be independent of each other with a common deviation parameter. As Eq. 4.2 is a Gaussian density function, it is natural to select a hyperprior distribution which is conjugate for the Gaussian distribution. This ensures that the resulting prior distribution is of the same functional form as the hyperprior, which makes computations easier. The inverse-gamma distribution is conjugate for the family of Gaussian distributions, and as such is a suitable hyperprior (Gelman et al., 2004). Its functional form is

$$\text{Inv-gamma}(\theta|\alpha, \nu) = \frac{\alpha^{\nu}}{\Gamma(\nu)}\theta^{-(\nu+1)}e^{-\alpha/\theta}, \tag{4.3}$$

where α is the scale parameter, ν determines the shape of the distribution, and $\Gamma(.)$ is the Gamma function. The prior thus becomes

$$\begin{aligned} p(x_i|\mathbf{x}_{\setminus i}, \mu_{shape}, \sigma^2) &= \text{N}(x_i|\, f(\mu_{shape}, \mathbf{x}_{\setminus i}), \sigma^2\mathbf{I}) && (4.4)\\ p(\sigma^2|\alpha, \nu) &= \text{Inv-gamma}(\sigma^2|\alpha, \nu), && (4.5) \end{aligned}$$

from which the conditional posterior distributions of the parameters can be derived (Neal, 1996):

$$p(x_i|\mathbf{x}_{\setminus i}, \mu_{shape}, \sigma^2) = \text{N}(x_i|\, f(\mu_{shape}, \mathbf{x}_{\setminus i}), \sigma^2\mathbf{I}) \tag{4.6}$$

$$p(\sigma^2|\mathbf{x}, \mu_{shape}, \alpha, \nu) =$$
$$\text{Inv-gamma}\left(\sigma^2|\frac{\alpha^2\nu + \sum_{i=1}^{N}(x_i - f(\mu_{shape}, \mathbf{x}_{\setminus i}))^2}{\nu + N}, \nu + N\right). \quad (4.7)$$

In $p(\sigma^2|\mathbf{x}, \mu_{shape}, \alpha, \nu)$ the term $\alpha^2\nu$ acts as the term coming from the prior on σ^2, while $\sum_{i=1}^{N}(x_i - f(\mu_{shape}, \mathbf{x}_{\setminus i}, \mu_{shape}))^2$ represents the likelihood term (N is the number of features). In this context α^2 is the prior mean of σ^2, and ν is the number of prior measurements.

It is also possible to have different σ's for the different features, in which case the conditional posteriors become

$$p(x_i|\mathbf{x}_{\setminus i}, \mu_{shape}, \sigma_i^2) = \text{N}(x_i|f(\mu_{shape}, \mathbf{x}_{\setminus i}), \sigma_i^2\mathbf{I}) \quad (4.8)$$
$$p(\sigma_i^2|\mathbf{x}, \mu_{shape}, \alpha, \nu) =$$
$$\text{Inv-gamma}\left(\sigma_i^2|\frac{\alpha^2\nu + (x_i - f(\mu_{shape}, \mathbf{x}_{\setminus i}))^2}{\nu + 1}, \nu + 1\right). \quad (4.9)$$

The manual selection of parameter values can be moved even further up the hierarchy, decreasing the significance of their fixing even further, by choosing a common prior for the parameter α of the inverse-gamma distribution. To maintain the conjugacy, the inverse-gamma distribution can again be selected for the task, leading to the prior model

$$p(x_i|\mathbf{x}_{\setminus i}, \mu_{shape}, \sigma_i^2) = N(x_i|f(\mu_{shape}, \mathbf{x}_{\setminus i}), \sigma_i^2) \quad (4.10)$$
$$p(\sigma_i^2|\alpha, \nu) = \text{Inv-gamma}(\sigma_i^2|\alpha, \nu) \quad (4.11)$$
$$p(\alpha^2|\alpha_0, \nu_o) = \text{Inv-gamma}(\alpha^2|\alpha_0, \nu_0). \quad (4.12)$$

The conditional posteriors for x_i and σ_i^2 are as before (except that in $p(\sigma_i^2|.)$ the value of α^2 changes), and the conditional posterior of α is (Neal, 1996):

$$p(\alpha^2|\sigma_{1\ldots N}, \nu, \alpha_0, \nu_0) = \text{Inv-gamma-inv-gamma}(\alpha^2|\alpha_0^2, \nu_0, \nu, \frac{1}{\sum_i \frac{1}{\sigma_i}}, N), \quad (4.13)$$

where α_0 and ν_0 are the parameters of the hyper-hyperprior. Inv-gamma-inv-gamma denotes the inverse-gamma-inverse-gamma distribution, which is the posterior distribution when both the prior and the likelihood are inverse-gamma distributions. The difficulty with having individual σ parameters for the features is that there is only one data point that can be used in inferring the parameter value and hence the prior on σ must be rather tight (large ν) or the posterior of σ will be too wide to be used in practice.

4.3 Learning the shape from data

The graph shape models presented in Section 4.2 are very general in the sense that when using them, the only requirement is to specify a reference graph for the object class to be matched. However, it is clear that it is possible to enhance matching performance for a single object class by including more information about the class in the model. This is usually done by modeling the variations of the feature locations besides the reference graph (Cootes et al., 1995) (Cootes et al., 2001) (Zhou et al., 2003). Due to the increased complexity of such models, more attention has to be paid to learning their parameters from the training data using statistical methods.

4.3.1 Principal components and Active Appearance Models

A seminal approach to shape modeling is the Active Shape Model by Cootes et al. (1995). In ASM, objects are modeled using the mean object shape and the weighted sum of k modes of variation from this shape:

$$\mathbf{x} = \mu_{shape} + \mathbf{S}\beta_s, \tag{4.14}$$

where $\mathbf{S}$ is a $2N \times k$ matrix containing the vectors depicting the variations, and β_s is a vector of coefficients. The variation modes are obtained with principal component analysis (Chatfield and Collins, 1995), that is, by computing the covariance matrix of the training data and finding the M leading eigenvectors of this matrix. Before computing the vectors, translation, scaling, and rotation effects are removed by applying the Procrustes method (Goodall, 1991) which iteratively minimizes a weighted sum of distances between the training examples so that the training examples correspond as closely as possible. Following Probabilistic PCA (PPCA) by Tipping and Bishop (1999), Zhou et al. (2003) have proposed a probabilistic extension to ASM, in which isotropic Gaussian noise is added to the coefficients. This makes it possible to use the PCA model as a shape prior in a Bayesian framework. A sample PCA shape base is shown in Figure 4.2. Note that in computing the illustrated base, the rotational transformations have not been removed from the training data, which is why there are components related to rotations present (see Section 4.3.2 for details).

Active Appearance Models (Cootes et al., 2001) are an extension of Active Shape Models in which the appearance of the object is also modeled with a texture model via PCA (see Section 3.4), after which the shape and appearance models are combined by performing another principal component analysis on the coefficients of the lower level models. This results in a combined model, which represents the major modes of shape and appearance variation simultaneously. In mathematical terms, AAM concatenates the shape coefficients β_s and the texture coefficients β_t

Figure 4.2: Principal component shape model. In the upper row, the thick gray graphs show the mean shape and the thinner black graphs the leading eigenvectors added to the mean. In the other rows, the face on the left has been morphed according to the principal components, both in the positive (middle row) and negative (lower row) directions. Components 1 and 2 appear to be related to rotations, while components 3, 4, and 5 are shape-related.

into a single vector β:

$$\beta = \begin{pmatrix} \mathbf{W}_s \beta_s \\ \beta_t \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \mathbf{S}^T (\mathbf{x} - \mu_{shape}) \\ \mathbf{P}^T (\mathbf{T} - \mu_{text}) \end{pmatrix}, \tag{4.15}$$

where $\mathbf{W}_s$ is a diagonal matrix of weights for the shape parameters, which balances the difference in units between the shape and texture models. PCA is then performed on these vectors to produce the final AAM model

$$\beta = \mathbf{Q}\mathbf{c}, \tag{4.16}$$

where $\mathbf{Q}$ are the eigenvectors and $\mathbf{c}$ is a vector of appearance parameters. In the reverse direction, the shape and texture components can be expressed directly as functions of $\mathbf{c}$:

$$\mathbf{x} = \mu_{shape} + \mathbf{S}\mathbf{W}_s\mathbf{Q}_s\mathbf{c}, \mathbf{T} = \mu_{text} + \mathbf{P}\mathbf{Q}_t\mathbf{c}, \tag{4.17}$$

where $\mathbf{Q}_s$ and $\mathbf{Q}_t$ are the elements of $\mathbf{Q}$ corresponding to shape and texture:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_t \end{pmatrix}. \tag{4.18}$$

AAM is a very powerful appearance model with numerous applications in object matching and recognition, ranging from segmentation of 3-D medical images (Mitchell et al., 2002) to real-time face tracking (Ahlberg, 2002).

4.3.2 Full distribution modeling

Principal component analysis-based shape models have one notable weakness: they are parametrized in terms of the principal components, which means that it is not possible to compute the conditional or marginal distributions of individual features, and hence only the Metropolis-Hastings algorithm can be used to analyze the resulting posterior distributions. The aim of this section is to present a direct probabilistic model for the shape of the object from which these distributions can be computed. The model has been presented earlier in various forms in (Tamminen and Lampinen, 2003b), (Tamminen and Lampinen, 2004) and (Tamminen and Lampinen, 2005).

The object shape model is learned from a set of m training shapes $\mathbf{Y} = (\mathbf{Y}_1 \dots \mathbf{Y}_m)^T$, which in practice are the same as the annotations used in the learning of feature appearance. To eliminate pose effects and the random asymmetry of human faces, a mirrored replicate of each training shape is inserted as part of the training data set. As in Procrustes analysis, the shapes are first translated and scaled so that they are as alike as possible. However, rotational transformations are not removed (see Figure 4.2) - this eliminates the need for explicit rotation parameters for slight rotations from the matching model. In principle it should be possible to produce a rotation-invariant matching system by including the rotations in the shape model, as is done here, and utilize a mixture of Gaussians model for Gabor filter responses to account for the rotation-induced changes in feature appearance (see Section 3.3.4).

As in the principal component models, the shape of the object is considered to be the sum of a basic shape and variations from it. A convenient representation of this is a Gaussian distribution with the basic shape represented by the mean μ and the variations by the covariance Σ. Note that this corresponds to a principal component model with the eigenvectors of Σ as the principal components. Since all available information comes from the training shapes $\mathbf{Y}_i$ and no real prior information about the covariance of the feature locations can be realistically obtained, a vague conjugate inverse-Wishart prior is set on the covariance matrix (Gelman et al., 2004):

$$\Sigma \sim \text{Inv} - \text{Wishart}_{\nu_0}(\Lambda_0^{-1}),\ \mu|\Sigma \sim \text{N}(\bar{\mathbf{Y}},\ \Sigma/\kappa_0), \tag{4.19}$$

where ν_0 and Λ_0 describe the degrees of freedom and the scale matrix of the inverse-Wishart distribution, and κ_0 is the number of prior measurements. This kind of prior is equivalent to adding Gaussian noise to the locations of the in-

dividual features in the training shapes, similar to the PPCA approach of Zhou et al. (2003), and it makes the shape model more robust against random perturbations. Furthermore, if the amount of training data is small, the prior increases the rank of the covariance matrix, making it invertible, which is a requirement for all computations.

The posterior distribution of the distribution parameters is from the normal-inverse-Wishart family, parametrized as (Gelman et al., 2004)

$$p(\mu, \Sigma) \propto |\Sigma|^{-((\nu_m+d)/2+1)} \exp\left(-\frac{1}{2}\mathrm{tr}(\Lambda_m \Sigma^{-1}) - \frac{\kappa_m}{2}(\mu-\mu_m)^T \Sigma^{-1}(\mu-\mu_m)\right), \tag{4.20}$$

with parameters

$$\mu_m = \bar{\mathbf{Y}},\ \kappa_m = \kappa_0 + m,\ \nu_m = \nu_0 + m,\ \Lambda_m = \Lambda_0 + \sum_{i=1}^{m}(\mathbf{Y}_i - \bar{\mathbf{Y}})(\mathbf{Y}_i - \bar{\mathbf{Y}})^T, \tag{4.21}$$

with d indicating the dimension of the distribution ($d = 2N$, where N is the number of features).

When using this model, the interesting quantity is the predictive distribution of a new shape $\mathbf{x}$ given the training shapes, which can be computed with the result

$$p(\mathbf{x}|\mathbf{Y}) = \iint p(\mathbf{x}|\mu, \Sigma)\, p(\mu, \Sigma|\mathbf{Y}) d\mu d\Sigma = t_{\nu_m - d + 1}(\mu_m, \frac{\Lambda_m(\kappa_m+1)}{(\kappa_m(\nu_m - d + 1))}), \tag{4.22}$$

where t_α is the Student t-distribution with α degrees of freedom.

As described in Section 2.4.3, the Gibbs sampling and sequential Monte Carlo algorithms require the computation of conditional distributions of the object shape distribution. This in turn requires that the parameters of the prior distributions are set such that the covariance is reasonable (the denominator of the covariance becomes negative if $\nu_m < d - 1$) and that it is possible to approximate the t-distribution with a Gaussian one. Provided that this can be done, the conditional distributions can easily be computed from the Gaussian approximation, in which case the final model for the shape of the object to be matched is

$$p(\mathbf{x}|\mathbf{Y}) \approx \mathrm{N}(\mu_m, \frac{\Lambda_m(\kappa_m+1)}{(\kappa_m(\nu_m - d + 1))}). \tag{4.23}$$

4.4 Snakes and splines

Another alternative to modeling the shape of objects is to represent contours with parametrized curves. This approach is especially popular when the visual appearance of the object is modeled with edge features (Section 3.2.2). The curve $\mathbf{r}$ is

parametrized as $\mathbf{r}(s) = (u(s), v(s))$, where s is a parameter that increases as the curve is traversed, and $u(s)$ and $v(s)$ give the u- and v-coordinates of the curve as functions of s. These curves are called snakes (Kass et al., 1988). A snake aims to minimize an energy functional, which is composed of internal and external energy parts. The internal energy controls the shape of the snake by constraining the tension and rigidity of the curve, while the external energy guides the snake according to the perceived image. In Bayesian terms, these are equivalent to the prior and the likelihood.

Piecewise polynomials or splines are often chosen as the functional form of $u(s)$ and $v(s)$, in which case the curves are also called spline curves (Blake and Isard, 1998). A spline function of order d consists of concatenated polynomial segments or spans of order d, joined together at breakpoints or knots. Spline curves are attractive for contour modeling as they can represent complex shapes while retaining a low polynomial degree, which keeps computations simple and efficient.

A particularly convenient representation for the spline functions are so-called B-splines (Cipolla and Blake, 1990), in which the function is constructed as a weighted sum of N_B identical translated basis functions so that

$$u(s) = \mathbf{B}(s)^T \mathbf{w}^x, \tag{4.24}$$

where $\mathbf{B}(s)$ is a vector containing the basis functions and $\mathbf{w}^x$ is a vector of weights applied to the basis functions. A set of quadratic ($d = 3$) basis functions and the resulting spline function are illustrated in Figure 4.3. By regarding the weights as control points, the resulting spline curve $\mathbf{r}(s) = (u(s), v(s))$ is a smooth approximation of the polygon formed by the points. An example of this is shown in Figure 4.4.

Snakes and spline curves are very widely used in computer vision: the Kass et al. (1988) paper generates thousands of citations in research indices. They are computationally very light - for example, with B-splines, the transformation from control points to curve is a linear operation - but can nevertheless represent complex curves. Their main disadvantage is that they are explicitly contour models and as such can not be used to model object classes whose shape is not represented by the countour alone. Furthermore, as was the case with PCA models, the parametrization via control points instead of the curve itself - whose points correspond to the feature locations $\mathbf{x}$ of Chapter 2 - makes it difficult or impossible to compute the conditional distributions of the feature locations directly.

4.5 Dynamical modeling

In this chapter, numerous models for modeling the shape of objects in static scenes have been discussed. This section concerns modeling objects whose shape

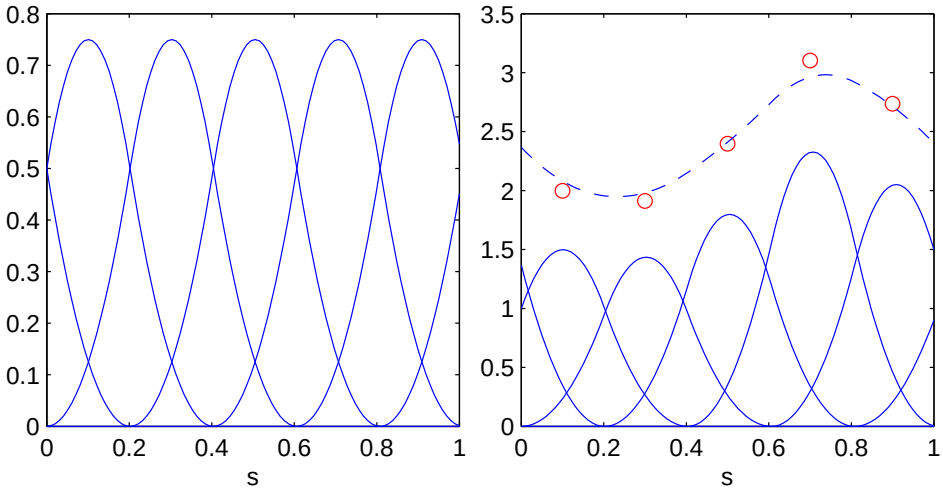

Figure 4.3: B-spline basis. The image on the left shows a set of periodic quadratic B-spline basis functions. The image on the right illustrates how the basis functions are multiplied by the coefficients (red circles) and combined to produce the spline function (dashed line).

changes over time. Such models are required for object classes such as human faces which exhibit specific modes of time variation, namely facial expressions in the case of human faces. The proposed model has been previously presented in (Tamminen et al., 2005).

4.5.1 Data acquisition and feature tracking

Including the time dimension in shape models increases the complexity of the models greatly, making it clear that the model has to be learned from experimental data. In this study, the objects under scrutiny were human faces. To obtain the training data, facial expressions were recorded from actors trained to express certain prototypical emotional facial expressions. The recordings included seven facial expressions related to basic emotions (Ekman, 1984) (anger, disgust, fear, happiness with the mouth open and closed, sadness, and surprise), two facial expressions related to blends of basic emotions (happiness + surprise and happiness + disgust) and one emotionally meaningless facial expression (the mouth opening). The recordings were made from 6 actor students from the Theatre Academy of Finland (3 men and 3 women, age range 23-32 years); hence, there were 60 video streams in total. The actors were asked both to express the given facial configuration exactly and to experience the required emotion. The actors practised the facial expressions individually for approximately 5-10 hours. One practise record-

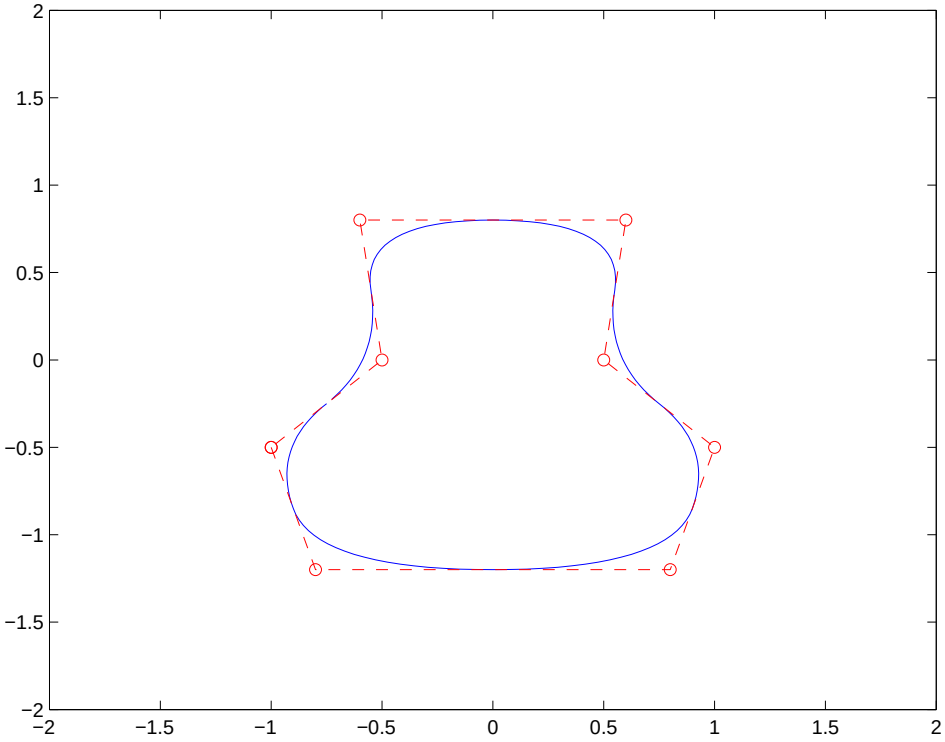

Figure 4.4: A 2-D spline contour. The red circles mark the control points and the dashed red line the control polygon. The blue curve is the corresponding spline curve. The curve follows the polygon closely.

ing was carried out with the possibility for feedback before the actual recording session. The recordings contained short (1-2s.) video sequences showing the change from neutral to the target state. Nine markers were placed on perceptually meaningful locations (Figure 4.5) to ease tracking of facial changes unrelated to clear facial features.

In visual feature tracking a set of features is annotated or matched on the first frame of a video stream and the objective is to track these features over the stream. The Kanade-Lucas-Tomasi (KLT) tracker and its derivatives are used widely in visual feature tracking (Lucas and Kanade, 1981) (Tomasi and Kanade, 1991) (Bourel et al., 2000). However, for this study it was decided to test the possibilities of an automated tracker based on Gabor filters as an extension of the static object matching system. A similar approach to facial feature tracking without the Bayesian context has previously been presented by McKenna et al. (1997).

As previously, the face is represented as a planar graph containing N nodes

(Fig. 4.5) with coordinates $\mathbf{X}^t = (x_1^t \dots x_N^t)^T$ at time step t. As in Section 3.3, each image in a video sequence is filtered with a Gabor filter bank and the responses are stacked as vectors to form the associated feature appearance vectors g_i^t.

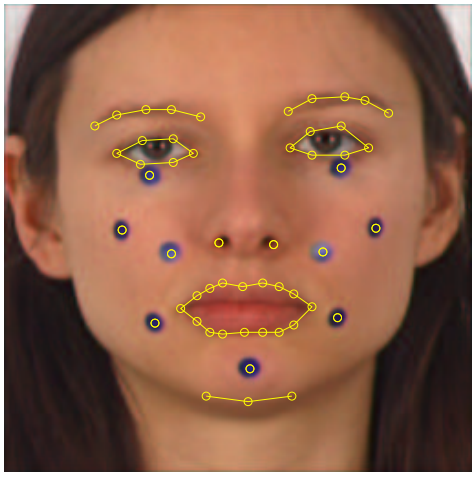

Figure 4.5: A sample feature graph, with the added dark markers showing. The yellow circles mark the tracked features.

The features are tracked by finding, at each time step, the maximum a posteriori estimate of the location of each feature around its previous location. That is, the posterior density of each feature is computed in some search area A_i given the transformed image, the corresponding feature vector g_i^t and the other feature locations $\mathbf{x}_{\setminus i}^t$, and maximized:

$$\max_{x_i^t \in A_i} p(x_i^t | \mathbf{T}^t, g_i^t, \mathbf{x}_{\setminus i}^t) \propto p(T_i^t | x_i^t, g_i^t)\, p(x_i^t | \mathbf{x}_{\setminus i}^t), \tag{4.25}$$

where Bayes's formula has been used to write the posterior probability as the product of the likelihood and prior parts. As in Chapter 2, the likelihood measures the probability of observing the image given a feature configuration, while the prior gives the distribution of the feature location given the locations of the other features. The likelihood is approximated with Eq. 3.8 due to its ability to compute the similarities relatively accurately given just one training image. The simple graph prior (Eq. 4.2) is used as the prior. σ^2 was set to such a value that the tracker would function well. With 256×256 images, $\sigma = 5$ was used.

As the video sequence progresses, both the features $\mathbf{g}$ and the mean shape $\mathbf{r}$ change. To adapt the tracker to this, at each time step the feature appearance vectors $\mathbf{g}$ and the mean graph shape $\mathbf{r}$ are changed according to the newly obtained

values:

$$\mathbf{g}^{t+1} = \alpha_g \mathbf{g}^t + (1-\alpha_g)\mathbf{g}^1 \quad (4.26)$$

$$\mathbf{r}^{t+1} = \alpha_r \mathbf{r}^t + (1-\alpha_r)\mathbf{r}^1, \quad (4.27)$$

where α_g and α_r are parameters controlling the extent of the adaptation. Using $\mathbf{g}^1$ and $\mathbf{r}^1$ as the baseline values reduces the probability of the tracker adapting to track a completely spurious feature, as the effect of the original Gabor jets and mean shape never disappears completely.

The initial feature locations $\mathbf{X}^1$ and Gabor jets g^1 are obtained by manually annotating the features on the first image of one video sequence and then using the image and the annotations as training data for matching the features in the first images of other sequences with the matching system presented in this thesis. The mean shape $\mathbf{r}^1$ is taken to be equal to $\mathbf{x}^1$.

The performance of the tracker was varying. In some streams it tracked the features perfectly, in some streams there were considerable errors. The tracking could be improved in numerous ways such as including a systematic model for the motion of the features or designing a more sophisticated adaptation scheme. However, since the main interest here lies on the dynamic shape model instead of the tracking, the improvements were left to a further study.

4.5.2 The expression model

In the model presented here, the aim is to find separate orthogonal bases for representing variations due to face shape and facial expression. A similar approach has been proposed in Abboud and Davoine (2004); however, they do their modeling within the AAM framework (Cootes et al., 2001) and model only the start- and endpoints of expressions, whereas here the interest lies in the the whole track of the fiducial feature points during an expression. Another, Eigenface-based non-dynamical model has been presented by Calder et al. (2001), whereas a dynamical model without the separation of shape and expression has been proposed by Faraway (2004).

To model the dynamics of the expressions, the time correlations of the feature point tracks are included into the expression model, that is, the expressions are described by vectors of length $2N \times n_t$, where n_t is the number of time steps. It is assumed that the tracks $\mathbf{X} = (\mathbf{X}^1, ..., \mathbf{X}^{t_f})^T$ can be represented as the sum of two linear models: a person-dependent shape model and a person-independent expression model so that

$$\mathbf{X} = \mathbf{1} \otimes (\mathbf{m} + \mathbf{S}\beta_{person}) + \mathbf{E}\beta_{expression} + \epsilon, \quad (4.28)$$

where $\mathbf{m}$ is the mean shape, $\mathbf{S}$ is the matrix of the base vectors of the shape space,

$\mathbf{E}$ is the matrix of the base vectors of the expression space, β_{person} is the person-dependent vector of coordinates in the shape space, $\beta_{expression}$ is the expression-dependent vector of coordinates in the expression space, $\mathbf{1}$ is a vector of ones, $\otimes$ is the Kronecker product, and ϵ is Gaussian noise. The Kronecker product expands the static shape model over time. At time step t the graph is

$$\mathbf{X}^t = \mathbf{m} + \mathbf{S}\beta_{person} + \mathbf{E}^t\beta_{expression} + \epsilon^t, \tag{4.29}$$

where $\mathbf{E}^t$ contains the elements of the expression base vectors that apply to time step t.

To estimate the base vectors of the shape and expression spaces, the shape and expression effects have to be separated. This is done in two phases:

1. Estimate the mean shape and the shape base vectors via PCA from the initial feature graphs $\mathbf{X}^1$. It is assumed that the video streams start from a neutral expression, that is, $\mathbf{E}^1 = 0$.

2. To remove the effect of the shape from subsequent images in the stream, subtract the projection of the initial graph onto the shape base $\mathbf{S}\mathbf{S}^T\mathbf{X}^1$ from the subsequent graphs. Then stack the graphs as vectors and perform PCA to obtain the expression base vectors.

In phase 2, PCA is performed on the correlation matrix of the vectors, that is, a "mean expression" is not subtracted from the graphs. Before estimating the model, the lengths of the tracks have to be normalized by selecting a common frame number (larger than the length of the longest video sequence) and interpolating the tracks as necessary so that their lengths match.

The model can also be described as the sum of two Gaussian distributions:

$$p(\mathbf{X}) = \mathbf{1} \otimes \mathrm{N}(\mathbf{m}, \Sigma_{shape}) + \mathrm{N}(0, \Sigma_{expression}), \tag{4.30}$$

where Σ_{shape} is the covariance matrix of the shape distribution and $\Sigma_{expression}$ the correlation matrix of the expression distribution (with $\mathbf{S}\mathbf{S}^T\mathbf{X}^1$ removed). The eigenvectors of these matrices are the base vectors mentioned above. The Gaussian representation makes it possible to use the model directly as a prior in object matching.

Chapter 5

Extensions to the matching model

5.1 Introduction

The basic matching model utilizing the feature appearance and object shape models of Chapters 3 and 4 is capable of matching objects provided that the observed objects resemble the training data quite closely. In practice the perceived objects can differ from the training examples greatly due to factors such as change of scale and occlusion. This chapter extends the Bayesian probability model so that the matching system is able to cope with more difficult matching situations.

5.2 Minor scale changes

As described in the Chapter 4, when learning the PCA shape base or the full shape distribution, the scaling factors of the training shapes are normalized to remove scale effects from the shape model. This means that the resulting shape model is scale-free, and the scale parameter of the observed object has to be estimated. If a Gaussian model is used for the object shape, the conditional distribution of the scaling factor s is

$$p(s|\mathbf{x}, \mu, \Sigma) \propto \frac{1}{|s^2\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu s)^T (s^2\Sigma)^{-1}(\mathbf{x} - \mu s)\right) \tag{5.1}$$

$$\propto \frac{1}{s^d} \exp\left(-\frac{1}{2s^2}(\mathbf{x} - \mu s)^T \Sigma^{-1}(\mathbf{x} - \mu s)\right), \tag{5.2}$$

where $\mathbf{x}$ is the vector of feature locations, μ is the mean shape, Σ is the shape covariance, and d is the dimension of the distribution. In sequential matching, the uncertainty about the scale parameter decreases rapidly as more features are

matched. This is illustrated in Figure 5.1.

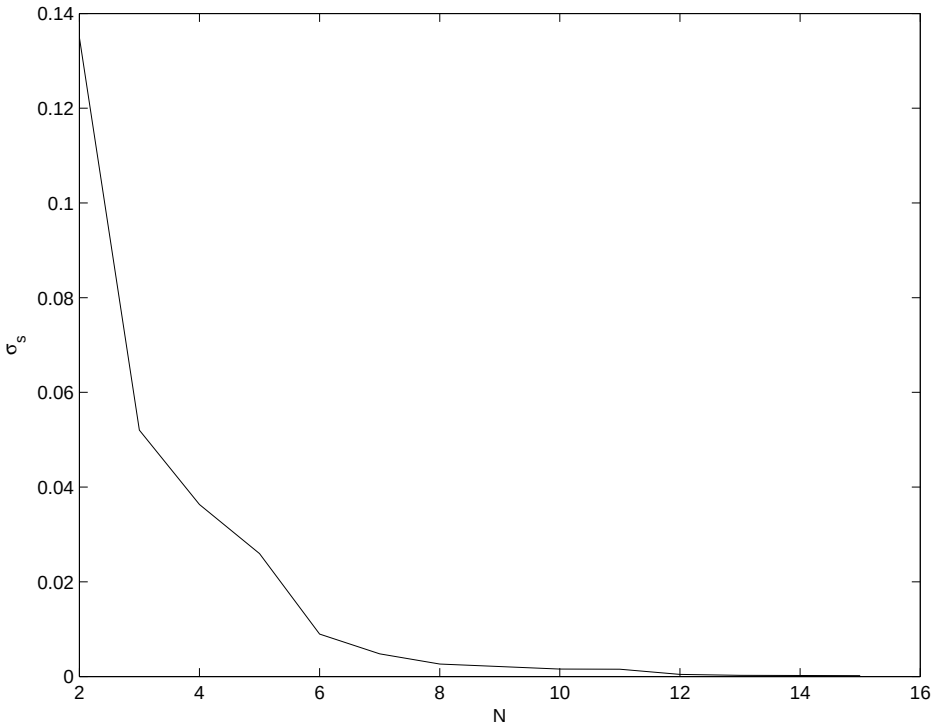

Figure 5.1: The scale parameter as a function of the number of matched features. The plot shows the standard deviation of of the distribution of the scaling factor as a function of the number of matched features from a single run of the system. The y-axis is set such that $s = 1$ corresponds to the mean size of the training shapes. Scale ambiguity is reduced rapidly as more features are matched.

In Gibbs sampling this is even quicker, as all features are included in Eq. 5.1 from the beginning. Hence, instead of using such a complicated distribution which would add yet another integration to the model, a simple scale estimator which is close to the maximum likelihood estimate of the distribution can be employed. Such a scale factor can be defined as

$$s = \sqrt{0.5\sigma_u^2 + 0.5\sigma_v^2}, \tag{5.3}$$

where σ_u and σ_v are the standard deviations of the feature location u- and v-coordinates. When more than 5 matched features are included in Eq. 5.1, the scale estimates obtained with this estimator are in practice equal to the maximum likelihood value of the true scale distribution.

5.3 Occlusion

Occlusion is one of the major problems in computer vision. In feature based object matching it is inconvenient since if the interesting features of an object are not detected, it is very hard to locate or recognize the object, and in holistic models occlusion causes the descriptor of the target object to become completely unlike the training data. For rigid object models, proposed approaches include employing edge detection and the Haussdorf distance (Rucklidge, 1997), directed edges and other similarity measures (Steger, 2002), and intensity-based matching in a Bayesian framework (Sullivan et al., 2001). These have obtained very good results, but for deformable models the task is more challenging: in rigid models the only problem is whether the feature is in the position where it should be, but in deformable models also the position is uncertain. Solutions have been proposed for dynamic tracking problems (Zhong et al., 2000) as well as static situations (Felzenszwalb, 2003), but also these approaches usually deal with occlusions with a "rigidness parameter" controlling the extent of allowed deformations instead of applying a formal occlusion model. Active Appearance Models of Cootes et al. (2001) (Section 4.3.1) have also been extended to handle occlusion: Edwards et al. (1999) disregard elements of the texture vector with unusually large pixel differences, whereas Gross et al. (2004) propose a robust fitting algorithm which considers occluded pixels outliers. Both these and the recent graphical models utilizing non-parametric belief propagation (Section 2.3.4) (Sudderth et al., 2003) (Sigal et al., 2003) (Coughlan and Shen, 2004), have obtained promising results in occluded conditions.

To make the proposed matching framework able to deal with ooclusion, in this section the basic matching model presented in Chapter 2 is extended to situations in which some of the features are occluded. The occlusion model has been discussed previously in (Tamminen and Lampinen, 2004) and (Tamminen and Lampinen, 2005).

To include the possibility of occlusion in the matching model, a vector of indicator variables γ is defined such that

$$\begin{aligned} \gamma_i &= 1, \text{ if the } i\text{th feature is detected} \\ \gamma_i &= 0, \text{ if the } i\text{th feature is not detected} \end{aligned}$$

Now the aim is to infer the marginal posterior distribution of location of the ith feature:

$$p(x_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i') = \int p(x_i, \gamma_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i')d\gamma_i, \tag{5.4}$$

where $\gamma_i' = \gamma_{1,\ldots,i-1}$ denotes whether the features matched before the ith one were detected. Since there are only two possible values for γ_i, the integral can be

written as the sum

$$p(x_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i') = p(x_i, V_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i') + p(x_i, \bar{V}_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i'), \tag{5.5}$$

where $\gamma_i = 1$ is denoted with V_i and $\gamma_i = 0$ with $\bar{V}_i$. Again, with Bayes's theorem, the posterior (Eq. 5.5) can be written as

$$\begin{aligned} &p(x_i|\mathbf{T}, \mathbf{x_i}, \mathbf{G}, \mathbf{Y}, \gamma_i') \propto \\ &[p(T_i|x_i, g_i, V_i)P(V_i|\gamma_i') + p(T_i|x_i, g_i, \bar{V}_i)P(\bar{V}_i|\gamma_i')]p(x_i|\mathbf{x_i}, \mathbf{Y}), \end{aligned} \tag{5.6}$$

where the object prior has been assumed independent of feature detection and the prior probabilities of detection and no-detection have been assumed to be dependent only on the detections of the previously matched features, whereas the likelihood has been assumed not to depend on the previous detections. Furthermore, the detection probabilities are assumed to be a priori independent of the previous detections, since no model for the interdependence of the occlusions, i.e., for the shape of the occluding object, is available. For known occlusion configurations such a model could be included. For example, if it was known that the object is occluded by a vertical half-plane so that features to the left of some x-coordinate are not detected, it would be possible to define

$$p(V_i|\gamma_i') = V_i^{\sum_i \gamma_i^{\text{left}}}(1 - V_i)^{\sum_i (1-\gamma_i^{\text{right}})}, \tag{5.7}$$

where γ_i^{left} denotes the detection statuses of the features left to the ith feature and γ_i^{right} the detection statuses of the features to the right of the ith feature. The probability $P(V_i|\gamma')$ could then be sampled from this distribution in Gibbs sampling-like fashion, or the maximum likelihood (ML) estimate or the mean could be used. In a general scene analysis setting, a spatial model for occlusions requires that all objects from an image are recognized, with the foreground objects providing occlusion models for the covered objects. In this study only a weaker independent occlusion model is considered.

The difficult part of the model is $p(\mathbf{T}_i|x_i, g_i, \bar{V}_i)$, the likelihood when the feature is not detected. In theory this quantity should describe the distribution of the Gabor filter responses for all other image details besides the ones associated with the feature being matched. However, such a representative model is probably impossible to construct, and not even desirable: when a feature is occluded, it is preferable that the estimate of its location is not affected by what kind of filter responses are common in general. Hence, a flat likelihood is chosen - since the feature is not detected, no information about its location is obtained from the image. This flat likelihood is normalized so that it sums to unity over the image.

The a posteriori visibility status of a matched feature is straightforward to infer by computing the posterior probability of detection given the feature location

using the law of conditional probability and Bayes' rule:

$$P(V_i|\mathbf{T}_i, \mathbf{x}, g_i, \mathbf{Y}, \gamma_i') = \frac{p(x_i, V_i|\mathbf{T}_i, \mathbf{x_i}, g_i, \mathbf{Y}, \gamma_i')}{p(x_i, V_i|\mathbf{T}_i, \mathbf{x_i}, g_i, \mathbf{Y}, \gamma_i') + p(x_i, \bar{V}_i|\mathbf{T}_i, \mathbf{x_i}, g_i, \mathbf{Y}, \gamma_i')}$$
$$= \frac{p(\mathbf{T}_i|x_i, g_i, V_i)\, p(x_i|\mathbf{x_i}, \mathbf{Y})\, P(V_i|\gamma_i')/p(\mathbf{T}_i)}{p(\mathbf{T}_i|x_i, g_i, V_i)\, p(x_i|\mathbf{x_i}, \mathbf{Y})\, P(V_i|\gamma_i')/p(\mathbf{T}_i) + p(\mathbf{T}_i|x_i, g_i, \bar{V}_i)\, p(x_i|\mathbf{x_i}, \mathbf{Y})\, P(\bar{V}_i|\gamma_i')/p(\mathbf{T}_i)}$$
$$= \frac{p(\mathbf{T}_i|x_i, g_i, V_i)\, P(V_i|\gamma_i')}{p(\mathbf{T}_i|x_i, g_i, V_i)\, P(V_i|\gamma_i') + p(\mathbf{T}_i|x_i, g_i, \bar{V}_i)\, P(\bar{V}_i|\gamma_i')}, \tag{5.8}$$

which is simply the ratio of the likelihood when the feature is detected and the sum of the two likelihoods. The sum of the probabilities of detection of all matched features can be used as a measure for how many features have effectively been detected in the image. If the feature detections are assumed independent, the posterior probability of whether an object has been detected can be computed directly as

$$P_{det} = 1 - \prod_{i=1}^{N} 1 - P(V_i|\mathbf{T}_i, \mathbf{x}, g_i, \mathbf{Y}). \tag{5.9}$$

The implementation of the occlusion model is discussed in Section 5.6.

5.4 Multiresolution matching

The scale models discussed in Section 5.2 are able to cope with small scale changes in which the visual appearance of the features remains the same. However, if the size object to be matched is very different from the training objects, the system breaks down. This section presents an extended matching model which works on multiple levels of detail and is able to deal with large global scale changes.

5.4.1 The multiresolution object shape model

Real-world objects consist of multiple levels of detail - for example, in a human face the features can be grouped in levels such as "face", "eye", "corner of mouth" et cetera. If an object is sufficiently far away, only the coarsest features can be detected, and if it is close, not all of the coarser features might be observed. Matching objects of greatly varying sizes thus requires size-invariance both in the visual features as well as in the object models. Of the matching models presented in the literature with a large degree of scale-invariance, Miao et al. (1999) have proposed a multiscale template matching based system for face detection, Lowe (1999) uses the local scale-invariant features SIFT features described in Section 3.2.2, and Fergus et al. (2003) employ a Bayesian scale-invariant model for classifying images according to the objects contained in them. However, the proposed scale-invariant matching and detection systems usually represent objects at a single feature level,

which limits their applicability in the aforementioned cases when the target object is either very close or very far, although the unsupervised learning scheme of Fergus et al. makes an implicit hierarchical representation possible in some cases. Würtz (1997) presents a hierarchical matching system similar to the one described here. The main difference between the systems is that in (Würtz, 1997) the final result consists of matches only on the most fine detail level, whereas the proposed model matches the features of all levels so that the features of finer levels also affect the coarser features. Furthermore, in this study the dependencies of the features of various levels are modeled statistically, whereas in (Würtz, 1997) the constraints placed on the more detailed features are geometrical in nature.

In the model presented here, objects have three levels of detail, ranging from fine to crude. For example, for a human face the levels are: 1. the outlines of the eyes, the parts of the nose etc., 2. the eyes, nose, mouth etc., and 3. the face as a whole. It is assumed that these levels can be represented as a hierarchical structure where the finer details are grouped with respect to the corresponding coarser details - for example, the level 1 nose features are arranged around the level 2 nose feature. This is illustrated in Figure 5.2.

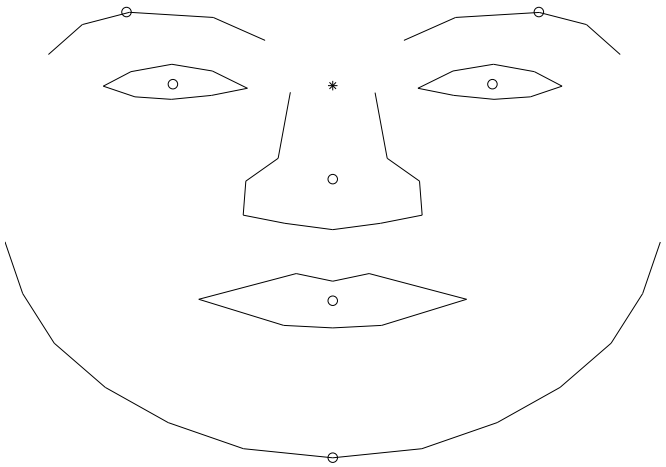

Figure 5.2: The multiresolution shape model. The nodes of the solid line mark the level 1 features, the circles the level 2 features, and the asterisk the level 3 feature. Note how some of the level 2 features are located in perceptually meaningful locations instead of the geometrical centre of the corresponding level 1 features.

The locations of level 1 features are denoted $x^i, i \in \{1..n_g\}$, where i is the group index and n_g the number of groups, the locations of level 2 features with $\mathbf{y}$ and the location of the level 3 feature with $\mathbf{z}$. As in Section 4.3.2, the distributions

of the feature locations at each detail level are assumed to be Gaussian:

$$p(\mathbf{x}|\mathbf{y}) = N(\mu_x + \mathbf{T}_x\mathbf{y}, \Sigma_x) \tag{5.10}$$

$$p(\mathbf{y}|\mathbf{z}) = N(\mu_y + \mathbf{T}_y\mathbf{z}, \Sigma_y) \tag{5.11}$$

$$p(\mathbf{z}) = N(\mu_z, \Sigma_z), \tag{5.12}$$

where μ_x, μ_y, μ_z, Σ_x, Σ_y, and Σ_z are the distribution means and covariances, and $\mathbf{T}_x$ and $\mathbf{T}_y$ are transformation matrices which match the dimensions of the different detail levels. The level 1 features are combined so that the vectors x^i and μ_x^i are stacked as the vectors $\mathbf{x}$ and μ_x and the covariances Σ_x^i are combined in the block diagonal matrix Σ_x.

The distribution parameters are learned as in Section 4.3.2 from a set of m training shapes $\mathbf{Y} = (\mathbf{Y}_1 \ldots \mathbf{Y}_m)^T$ so that each training shape contains all feature locations $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$. The distributions of all feature levels are learned separately with the same approximations as previously, with the result for a single feature level or group being

$$p(\mathbf{X}|\mathbf{Y}) \approx \mathrm{N}(\mu_m, \frac{\Lambda_m(\kappa_m + 1)}{(\kappa_m(\nu_m - d + 1))}) = N(\mu^*, \Sigma^*), \tag{5.13}$$

where $\mathbf{X}$ is a placeholder for any feature level or group, and the parameters μ_m, Λ_m, κ_m, ν_m, and d are as in Section 4.3.2. With the Gaussian approximation, it is possible to combine the individual distributions into a single joint Gaussian shape distribution with mean and covariance

$$\mu = \begin{bmatrix} \mu_x^* + \mathbf{T}_x\mu_y^* + \mathbf{T}_x\mathbf{T}_y\mu_z^* \\ \mu_y^* + \mathbf{T}_y\mu_z^* \\ \mu_z^* \end{bmatrix}, \tag{5.14}$$

$$\Sigma = \begin{bmatrix} \Sigma_x^* + \mathbf{T}_x\Sigma_y^*\mathbf{T}_x^T + \mathbf{T}_x\mathbf{T}_y\Sigma_z^*\mathbf{T}_y^T\mathbf{T}_x^T & \mathbf{T}_x\Sigma_y^* & \mathbf{T}_x\mathbf{T}_y\Sigma_z^* \\ (\mathbf{T}_x\Sigma_y^*)^T & \Sigma_y^* + \mathbf{T}_y\Sigma_z^*\mathbf{T}_y^T & \mathbf{T}_y\Sigma_z^* \\ (\mathbf{T}_x\mathbf{T}_y\Sigma_z^*)^T & (\mathbf{T}_y\Sigma_z^*)^T & \Sigma_z^* \end{bmatrix} \tag{5.15}$$

where μ_x^*,μ_y^*,μ_z^*,Σ_x^*,Σ_y^*, and Σ_z^* are as in Eq. 5.13. Note that this kind of object model is somewhat equivalent to the graphical model formulation of the non-parametric belief propagation based algorithms (Section 2.3.4): instead of modeling all dependencies of the variables, some of the correlations are intentionally left out. However, unlike NBP, the multiresolution model is valid in the sense that the dependencies that are included in the model are correct from the statistical point of view.

5.4.2 Multiresolution feature appearance

When Gabor filters are used to model feature appearance, it is rather straightforward to extend the appearance model to multiple resolutions by scaling the spatial frequency parameter accordingly, since the Gabor filter is self-similar under scaling (Daugman, 1988). Figure 5.3 illustrates this: when the object size is doubled, the Gabor filter responses remain approximately the same when the frequencies of the filters are halved. It is thus sufficient to learn the parameters of the feature appearance model using only one set of frequencies and then filter the observed image using multiple frequencies. Furthermore, as in the multiresolution object model there are features of markedly different scales, it makes sense to use different base frequencies for the individual feature levels so that the most detailed features are learned using large frequencies and the coarser features using smaller frequencies. However, there is an upper limit on how large frequecies can be used ($f \geq \pi$ is not feasible), which makes it impossible to detect the finest details when the perceived object is small enough.

When used as a whole, the multiresolution matching model causes little changes to the matching framework. The multiresolution shape model can be used directly in place of the simpler model, and the feature appearance model is basically the same - the only main difference is that the posterior sampling has to be performed multiple times so that the perceived image is filtered and the likelihoods computed with different base frequencies. Also, as the filter frequencies are changed, the starting value of the scaling factor naturally has to be changed accordingly as well. Finally, since in very close views of objects it is likely that some features are not included in the view, the occlusion model has to be used in conjunction with the multiresolution matching model. Using the occlusion model also makes it possible to assess whether there is an object of a particular size in the image by computing either how many features have effectively been detected or the detection probability via Eq. 5.9.

5.5 Multiple object instances

So far, the proposed matching framework has concentrated on finding a single instance of the target object class in the perceived image. It is also possible to extend the system to a multiple-object environment. In theory sequential Monte Carlo should be able to do this by itself, as it is capable of representing multimodal posteriors. However, in practice (see Section 6.7) only the mode with the largest posterior mass will survive to the end of the sampling due to the multiple resamplings, if the number of features is at least somewhat large.

There are numerous ways to augment the model so that it can handle multiple instances of an object class such as applying data association methods

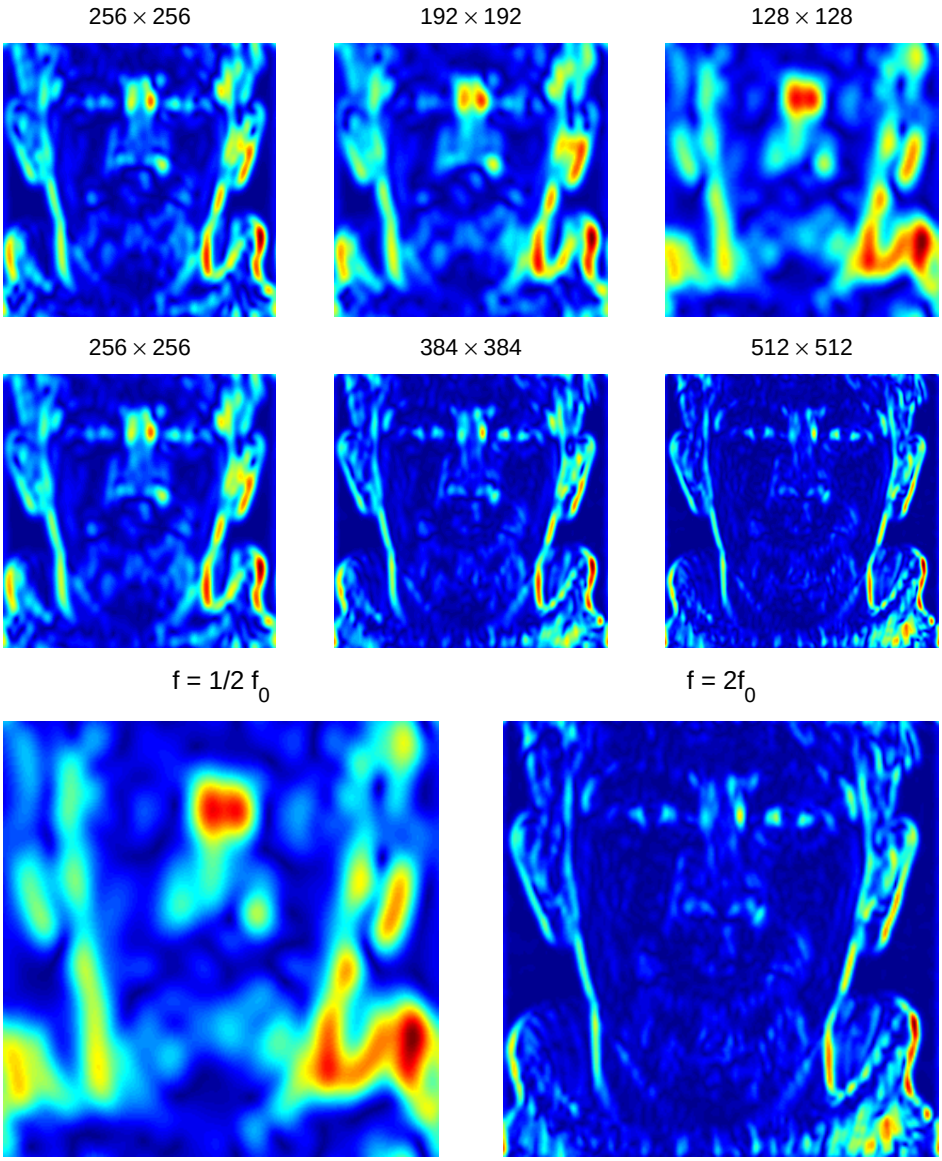

Figure 5.3: Gabor filter responses under scale changes. The images in the top two rows illustrate the amplitude response of a vertical Gabor filter when the size of the target object changes (the numbers above each image indicate the size of the filtered image). The bottom row shows the response of the 256×256 image when the filter spatial frequency is halved and doubled. It can be seen that if object scale is halved, the filter responses remain approximately the same if their frequencies are doubled. Objects of intermediate size can be matched using the frequencies of either scale.

from the field of multiple target tracking. In this study a straightforward method is presented: after one object has been matched, the corresponding area is marked occluded and the matching procedure is carried out again, until the sum of feature detection probabilities is deemed low enough. The multiple object case is modeled by introducing a new indicator variable δ for each pixel of the observed image. This variable is defined as

$$\begin{aligned}\delta(u,v) &= 1, \text{ if pixel } (u,v) \text{ belongs to the previously matched object(s)}\\ \delta(u,v) &= 0, \text{ if pixel } (u,v) \text{ does not belong to the previously matched object(s)}\end{aligned}$$

Now, as in the regular occlusion case, the marginal posterior of the feature locations is computed by integrating over this variable:

$$\begin{aligned}p(x_i|\mathbf{T},\mathbf{x_i},\mathbf{G},\mathbf{Y},\gamma_i') = \int p(x_i,\delta(u,v)|\mathbf{T},\mathbf{x_i},\mathbf{G},\mathbf{Y},\gamma_i')d\delta(u,v) = \\ p(x_i|d(u,v),\mathbf{T},\mathbf{x_i},\mathbf{G},\mathbf{Y},\gamma_i')P(d(u,v)) + \\ p(x_i|\bar{d}(u,v),\mathbf{T},\mathbf{x_i},\mathbf{G},\mathbf{Y},\gamma_i')P(\bar{d}(u,v)),\end{aligned} \tag{5.16}$$

where $d(u,v)$ indicates $\delta(u,v)=1$ and $\bar{d}(u,v)$ indicates $\delta(u,v)=0$. The pixels for which $\delta(u,v)=1$ are assumed to be occluded with probability 1, whereas the pixels for which $\delta(u,v)=0$ are handled as before. By applying Eq. 5.6 the model becomes

$$\begin{aligned}p(x_i|\mathbf{T},\mathbf{x_i},\mathbf{G},\mathbf{Y},\gamma_i') \propto \\ [(p(T_i|x_i,g_i,V_i)P(V_i|\gamma_i') + p(T_i|x_i,g_i,\bar{V}_i)P(\bar{V}_i|\gamma_i'))P(\bar{d}(u,v)) + \\ p(T_i|x_i,g_i,\bar{V}_i)P(\bar{V}_i|\gamma_i')P(d(u,v))]p(x_i|\mathbf{x_i},\mathbf{Y}).\end{aligned} \tag{5.17}$$

The probabilities $P(d(u,v))$ and $P(\bar{d}(u,v))$ are assigned manually according to some suitable criterion. For example, in the interior of the object that was matched first $P(d(u,v))$ can be set to unity, with a quick decay to zero around the edges.

5.6 Sampling considerations

The proposed extensions to the matching framework make the resulting posteriors rather more complicated than those of the basic model presented in Chapter 2. This necessitates extensions of the sampling algorithms, also. Especially the occlusion model makes the sampling more difficult due to the increased witch's hat nature of the resulting posteriors. In this section the capabilities of the various samplers are reanalyzed and the required extensions discussed.

5.6.1 MCMC algorithms

As mentioned in Section 2.4.3, the Metropolis-Hastings algorithm can be used to sample from any kind of posterior, no matter how it is parametrized. The main difficulty with using Metropolis-Hastings is constructing an efficient proposal distribution - for example, when the full covariance shape prior is utilized, using the standard proposal in which the parameters are uncorrelated will lead to a large sample rejection rate. Instead, it makes sense to use the prior, multiplied by a suitable small constant, such as 0.1, as the proposal so that the proposed graph shapes are sensible from the shape model's point of view. Another problem is the peaked nature of the likelihood, as it also can cause a large rejection rate if the proposed feature locations miss the peaks. This can be addressed by filtering the likelihood with a low-pass filter so that the peaks are flattened. Furthermore, Metropolis initialization is problematic, as the initial joint probability of the feature grid should be larger than zero. In practice this can be addressed by adding a small value to all the likelihood fields, which is equivalent to using the occlusion model.

In the Gibbs sampling algorithm, there are no tunable parameters or distributions - as long as the conditional distributions can be computed, the sampler is very straightforward to define. As to the general problems of Gibbs sampling, solutions to the mode-switching question (Section 2.3.5) include approaches such as reparametrization and adding artificial transition modes Gilks et al. (1996). Another proposal is the K-adventurers algorithm of Tu and Zhu (2002), which prunes the solutions produced by a reversible jump MCMC framework. The problem is that with very high-dimensional and correlated distributions such as the ones encountered in object matching, these solutions are often infeasible or computationally too demanding.

The difficulties arising from the curse of dimensionality (the witch's hat problem) have received much less attention in the literature. A sequential parallel tempering algorithm has been proposed which divides the high-dimensional problem into a sequence of subsystems which are then simulated using MCMC methods so that the information obtained from the low-dimensional systems is used to guide the simulation of the high-dimensional system Liang (2003). However, using Gibbs sampling in multiple nested problems is computationally very costly. As will be seen in Chapter 6, the curse of dimensionality is a major obstacle to analyzing the distributions produced by the proposed system with Gibbs sampling.

5.6.2 Sequential Monte Carlo

Similar to the Metropolis-Hastings algorithm, sequential Monte Carlo samplers require that a proposal distribution is specified. The target posterior distribution at each sampling step is given by Eq. 5.6 (inserting $P(\bar{V}_i) = 0$ yields the no-occlusion matching model). If a feature is visible in the image, the corresponding

likelihood is peaked, and a reasonable proposal is the likelihood itself. If a feature is not visible, the likelihood is flat, and it would be preferable to use the prior as the proposal. In SMC the proposal distribution may be chosen rather freely, as long as the particles are weighted correctly according to Eq. 2.24. One solution is to use a mixture proposal distribution composed of both the prior and likelihood terms:

$$\pi(\theta_{k+1}|y,\theta_k) = \phi p(y|\theta_{k+1}) + (1-\phi)\, p(\theta_{k+1}|\theta_k), \tag{5.18}$$

where ϕ is the mixing ratio between the two proposals (Fox et al., 2001). In this study, the mixing ratio is chosen adaptively by examining the ratio of the "detection" and "no-detection" likelihood masses (see Eq. 5.6) in some search area A around the prior mean:

$$\phi = 1 - \exp(1-R), \tag{5.19}$$

where

$$R = \frac{\sum_{T_i \in A} p(T_i|x_i, g_i, V_i) P(V_i)}{\sum_{T_i \in A} [p(T_i|x_i, g_i, V_i) P(V_i) + p(T_i|x_i, g_i, \bar{V}_i) P(\bar{V}_i)]}. \tag{5.20}$$

Selecting a "reasonable" search area, for example 3σ, also decreases the computational load greatly. In practice Eq. 5.19 causes the proposal to be equal to the likelihood when there is significant likelihood mass in the searching area and equal to the prior when the likelihood mass is low. This is illustrated in Figure 5.4. To summarize, the samples are drawn using Eq. 5.18 and weighted with

$$w_{k+1} = \frac{p(y|\theta_{k+1})\, p(\theta_{k+1}|\theta_k)}{\phi p(y|\theta_{k+1}) + (1-\phi)\, p(\theta_{k+1}|\theta_k)}. \tag{5.21}$$

In order to reduce the increase of variance of the weights induced by the resampling step, deterministic resampling was used as the resampling algorithm (Kitagawa, 1996), and to ensure that the variance of the weights remained at an acceptable level, the effective number of particles was monitored (Kong et al., 1994). When the occlusion model is used, it is not known which features are visible and which are not, and hence the matching order of the features is randomized over the particles. This is done by maintaining a feature matching order list $\mathbf{J}_t^j$ for each particle and randomly assigning each particle a feature from its list of unmatched features after the resampling step. As the occluded features have lower likelihood values and thus lower posterior probabilities than the visible ones, they receive lower weights, which causes them to have larger chance of getting eliminated during resampling. Hence, the visible features are (mostly) matched first. The first feature in the sequence is drawn directly from the corresponding likelihood field, implying a flat prior over the whole image. The sequential sampling process is summarized in the following:

Figure 5.4: The adaptive proposal distribution. In both images, two visible features have been matched: the corner of the eye and the corner of the mouth, marked with an 'o'. The red graphs show the mean of the conditional prior given the locations of the two features. The contours show the proposal distribution for the third feature to be matched: the visible point close to the tip of the ear in the image on the left and the occluded eye corner in the image on the right. The proposal for the visible feature is dominated by its very sharp likelihood, away from the prior, while the proposal for the occluded feature in practice equals its conditional prior.

1. **Initialization:** Assign first features to be matched, $\mathbf{J}_1^j = j \mod N$. For $j = 1$ to n, draw θ_1^j from the corresponding likelihood $p(T_i|x_i, g_i, V_i)$, $i = \mathbf{J}_1^j$, evaluated for all pixels of the image. Assign $\mathbf{J}_2^j$ randomly from $1...N$ so that $\mathbf{J}_2^j \neq \mathbf{J}_1^j$, and set $t = 2$.

2. **SMC sampling:** For $j = 1$ to n, draw θ_t^j according to Eq. (5.18) and compute the weight w_t^j according to Eq. (5.21) using $i = \mathbf{J}_t^j$ as the feature index.

3. **Resampling:** Normalize the weights w_t and resample n particles according to them. Set $w_t^j = 1/n$ and assign $\mathbf{J}_{t+1}^j$ randomly from from $1...N$ so that $\mathbf{J}_{t+1}^j \neq \mathbf{J}_{1...t}^j$. Set $t = t + 1$ and go to step 2.

It should be noted that when the particles have different feature matching sequences, the normalization of the weights is not strictly correct from the theoretical point of view. At $t = N$ the particles all represent the joint distribution of all features, but when $t < N$ the particles correspond to different random variables and there is no single distribution which would be represented by the particles and the weights. If a resampling step is not performed, this is not a problem, as

then the weights are not normalized until $t = N$. With resampling, the erroneus normalization causes an increase in the variance of the final estimates, since it can happen that sequences with small weights early in the sequence and large weights late in the sequence are eliminated in the early stages of the sampling. However, at $t = N$ the parameters in all particles have been drawn from the target distribution even if resampling is employed. A similar problem is encountered in target tracking when nearest neighbor-type methods are used for data association: the weights of particles conditioned on different measurements are not in principle normalizable. Rigorous data association methods such as the joint probability data association (JPDA) method have been presented in tracking literature, but these have their own problems such as the combinatorial explosion (Karlsson and Gustafsson, 2001).

Chapter 6

Evaluation of performance

6.1 Introduction

Chapters 2-5 have described the proposed matching model and its components: the Gabor filter-based feature appearance model, the object shape models (the hierarchical prior and the full covariance prior), and the sampling methods (Metropolis, Gibbs, and SMC sampling), as well as the various extensions to the basic matching model: the occlusion model, the multiresolution model, and the multiple object model. To form the complete object matching system, Eq. 3.13 is used to model the visual appearance of the features, and the samplers and priors are combined in the following ways:

- Sequential Monte Carlo (Section 2.3.3), full covariance prior (Section 4.3.2)
- Gibbs sampling (Section 2.3.2), common-variance hierachical prior (Section 4.2.1, Eqs. 4.4 - 4.7)
- Gibbs sampling, full covariance prior
- Metropolis algorithm (Section 2.3.2), full covariance prior

The occlusion model (Section 5.3), and hence the multiple object model (Section 5.5), can be used with any of the combinations, while the multiresolution matching model (Section 5.4) is only usable with sequential sampling. In principle it would be possible to use the hierarchical prior with SMC and Metropolis, as well, but the results are clearly inferior to Gibbs. This is because the benefit of the hierachical prior is the adaptation of the σ parameter according to Eq. 4.7, which is applicable only when using Gibbs sampling.

In this section, the matching results obtained by the various versions of the system are presented and compared with existing solutions to the object matching problem. In addition to this, some central practical implementation details are

discussed. The chapter concludes by analyzing the dynamical facial expression model proposed in Section 4.5.

6.2 Sensitivity to parameters

The proposed matching model contains a number of parameters whose values have to be assigned more or less arbitrarily according to the quality of the training images and other factors. In the Gabor filter-based feature appearance model the tunable parameters are the number of filters, their directions and frequencies, and the regularization term ϵ_G of the response distributions (Eq. 3.12). In the hierarchical graph prior there are no tuneable parameters; however, the full covariance model has the prior scale Λ_0 as well as the degrees of freedom ν_0 and prior measurements κ_0 as parameters (Eqs. 4.21 and 4.23). The occlusion model adds the prior detection probability $P(N_i|\gamma_i')$.

The proposed system is not very sensitive to the values of the parameters. The most important with regard to performance are the regularization term ϵ_G and the shape model prior parameters Λ_0, ν_0 and κ_0. ϵ_G determines the steepness of the likelihood; low values lead to very peaked likelihood fields, while higher values spread the likelihood mass more. If the training images are of poor quality, it is preferable to raise the value of this constant so that small peaks of the likelihood are included in the fields. For good-quality training images, suitable ϵ_G values are in the $0.01 \ldots 0.1$ range. Λ_0, ν_0, and κ_0 together with the training data control the width of the shape prior, that is, the extent of the allowed deviations from the mean. The values of these parameters are determined by how different the training data are expected to be from the observed novel shapes - in principle the prior should be as tight as possible to eliminate spurious matches due to clutter. With 58 features, suitable values for these parameters are in the $\Lambda_0 = 10 \ldots 100$, $\nu_0 \approx 100$ range, with κ_0 set to 1. The values of the other parameters are less significant; filter parameter values were assigned such that there were 3 frequencies (usually $\pi/4$, $\pi/8$, $\pi/16$, and globally scaled in multiresolution matching) and 6 orientations ($0, \pi/6$, $\pi/3$, $\pi/2$, $2\pi/3$, $5\pi/6$). $P(V_i|\gamma_i')$ was set to 0.5 when the occlusion model was used.

Besides the parameters related to the actual matching model, the parameters of the sampling algorithms also affect the matching performance of the system. The effect of initialization of the samplers is discussed in Section 6.8. Another important parameter is the number of MCMC samples or SMC particles. In the case of MCMC, the numbers of samples were selected to be large enough to that the chain would seem to converge in half the total number of samples, after which the first half could be discarded as burn-in. To retain comparability, the number of SMC particles was chosen so that the computational cost of the algorithms was approximately the same. In practice the values were $n \approx 1000$ samples for Gibbs

sampling, $n \approx 5000$ samples for Metropolis, and $n \approx 1000$ particles for SMC.

6.3 Data sets and matching details

Three different data sets were used to evaluate the performance of the system: the IMM-DTU database (Stegmann, 2002), a database of automatically generated artificial head images generated by Marco Müller using the Facegen software (http://www.facegen.com/) and kindly provided by Rolf Würtz, and the BioID database (Jesorsky et al., 2001) annotated by David Cristinacce and Kola Babalola of the University of Manchester. In all databases, the target object class was the human face, which is a good test class due to the large variability in both feature appearance and object shape. The IMM-DTU database contains 37 images with 58 annotated feature locations, the Facegen database 1000 images with 52 feature locations. and the BioID database 1521 images with 20 feature locations. The datasets are illustrated in Figure 6.1. Additional pictures taken with a digital camera were also used in the testing.

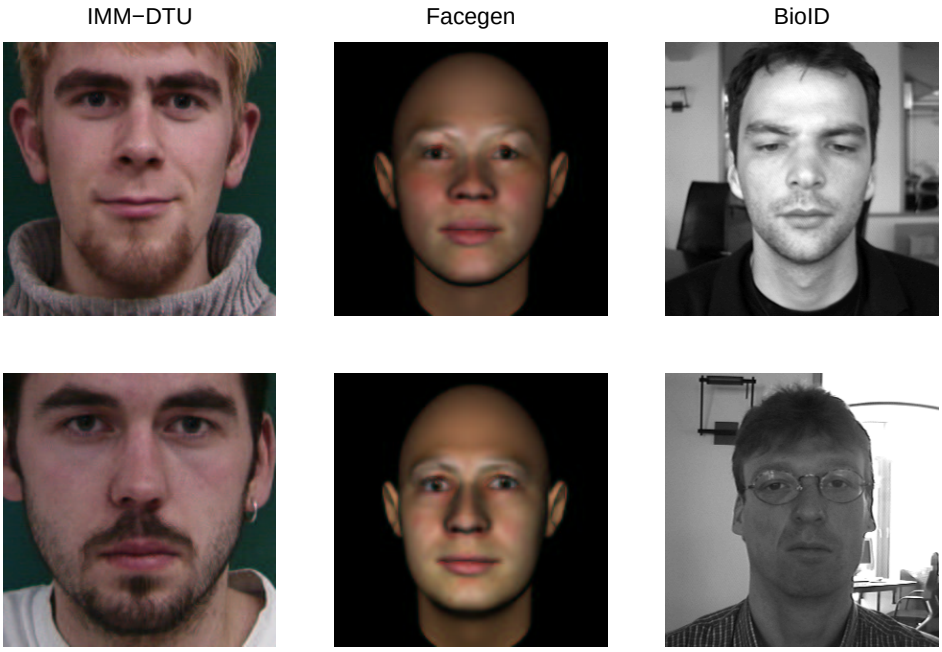

Figure 6.1: The various data sets used in the testing of the system. The images on the left are from the IMM-DTU database, the images in the center from the Facegen database, and the images on the right from the BioID database.

The matching process was different for the various data sets. For the Facegen

and BioID databases, 50 images were selected as training data and the rest were used for testing. This was not possible for the IMM-DTU database due to the low number of images, and hence the testing was carried out using leave-one-out cross-validation, so that each facial image was matched in turn by using the other 36 images as the training set. For the digital camera images, the whole IMM-DTU database was used as the training data.

The system was tested with unoccluded objects, occluded objects, objects with greatly varying scale, as well as with images containing multiple object instances. The matching error was assessed by estimating the posterior mean from the samples and computing the Euclidean point-to-point error (P2P) and the point-to-curve error (P2C) (measuring the Euclidean distance from the closest curve point) from the manual annotations.

6.4 Unoccluded objects

For unoccluded objects, all three databases were used. Gabor filters were used to model feature appearance, and object shape was modeled with the hierarchical graph prior with a common variance parameter and the full covariance prior. The resulting posteriors were analyzed by sampling with the Metropolis, Gibbs and sequential Monte Carlo algorithms as applicable. The performance of the system was compared with the AAM framework implementation of Stegmann (2002) for the IMM-DTU database and with the Elastic Bunch Graph Matching system (Wiskott et al., 1997) for all databases.

Both the case where all features were presumed to be detected ($P(V_i) = 1$) and the one where the possibility of occlusion was included ($P(V_i) = 0.5$) were considered, except for Metropolis, as described in Section 5.6.1. P2P and P2C error values and their standard deviations for the different databases are shown in Tables 6.1, 6.2, and 6.3, and some sample matches are shown in Figures 6.2, 6.3, and 6.4. In the tables, for many algorithms there are two entries: one for converged results and one for all images. In the entries for converged results, the matches which converged into an incorrect mode have been manually removed.

With the IMM-DTU database, Metropolis sampling and Elastic Bunch Graph Matching perform slightly worse than the other algorithms, whose results are more or less equivalent, except Gibbs sampling with the full covariance prior, which slightly outperforms the others. This is as expected - as there is only a single face in each image, the joint posterior should be unimodal and the Gibbs sampler utilizing the object shape model should obtain the best results due to its mode-exploring capabilities. Including the possibility of occlusion increases witch's hat nature of the distribution, which causes Gibbs sampling to fail sometimes.

The images of the Facegen database are rather easy to match: the features

Table 6.1: Matching results for different systems with unoccluded objects, IMM-DTU database. P2P is the point-to-point error and P2C is the point-to-curve error in pixels. $P(V_i)$ is the prior probability of occlusion.

Sampling algorithm and prior	P2P error	P2C error
SMC, full covariance prior		
$P(V_i) = 1$	5.45 ± 1.28	2.81 ± 0.82
$P(V_i) = 0.5$	5.52 ± 1.46	2.82 ± 0.99
Gibbs, hierarchical prior		
$P(V_i) = 1$	5.46 ± 1.55	2.69 ± 0.93
$P(V_i) = 0.5$	5.34 ± 1.40	2.78 ± 0.87
Gibbs, full covariance prior		
$P(V_i) = 1$	4.84 ± 1.35	2.57 ± 1.10
$P(V_i) = 0.5$, converged (n=31)	4.43 ± 0.84	2.39 ± 0.68
$P(V_i) = 0.5$, all images	13.7 ± 21.5	10.3 ± 18.6
Metropolis, full covariance prior		
$P(V_i) = 0.5$, converged (n=33)	6.24 ± 1.80	3.58 ± 1.37
$P(V_i) = 0.5$, all images	6.88 ± 2.57	4.11 ± 2.03
Matching system		
Elastic Bunch Graph Matching		
All images	6.16 ± 1.75	3.15 ± 1.05
AAM		
Grayscale	5.74 ± 1.18	3.04 ± 0.75
Color	5.54 ± 1.18	2.93 ± 0.81

are almost identical in the different images, and there is no background clutter. This is reflected in the matching errors, which are very low. Here the Gibbs sampler with the full covariance again slightly outperforms the other algorithms, with Metropolis obtaining the worst results.

The images of the BioID database are clearly the most difficult of the three due to variations in lighting, expression, and pose. As a result, the errors are rather high, and there are numerous occasions where the sampling algorithms have failed to converge. Sequential sampling is clearly the most robust algorithm with over 90% convergence rates with and without the occlusion model. The next best is Gibbs sampling with the hierarchical prior, followed by Elastic Bunch Graph

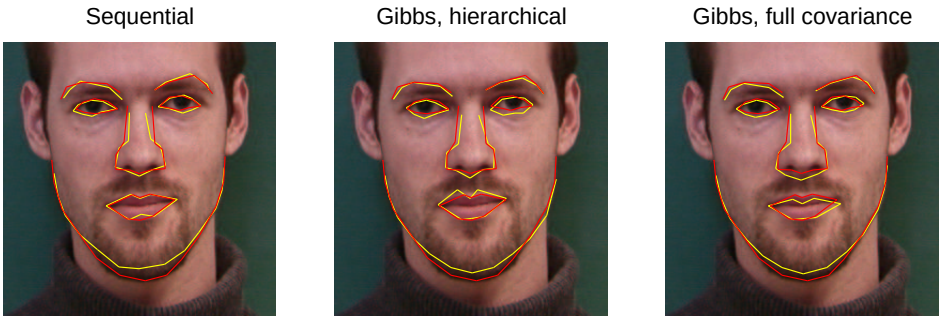

Figure 6.2: Sample matching results, IMM-DTU database. The image on the left has been matched with the sequential sampler, the center image with the Gibbs sampler and the hierarchical prior, and the image on the right with the Gibbs sampler and the full covariance prior. The yellow graphs show the sample means and the red graphs the manually annotated locations. Note how the hierarchical prior has been unable to capture the rotation of the face.

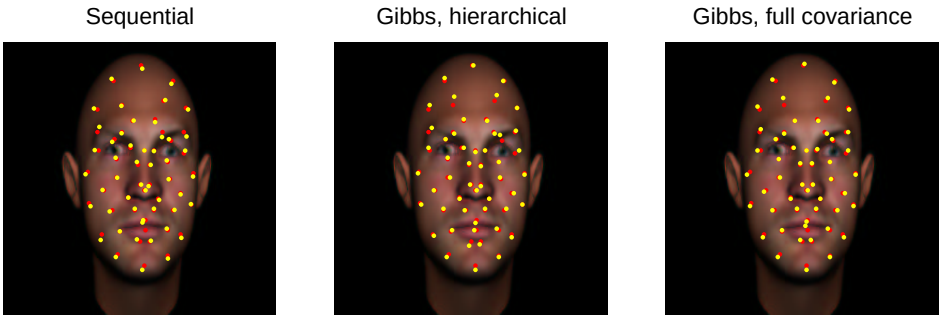

Figure 6.3: Sample matching results, Facegen database. The image on the left has been matched with the sequential sampler, the center image with the Gibbs sampler and the hierarchical prior, and the image on the right with the Gibbs sampler and the full covariance prior. The yellow dots mark the sample means, while the red dots are the annotated feature positions. There are very few differences in the matching results.

Matching and Metropolis. As can be expected for such a difficult dataset, Gibbs sampling with the full covariance prior obtains the worst results due to both multimodality and the witch's hat problem. An interesting detail is that for converged runs, EBGM obtains the most accurate matching results. This is probably due to the differences in the feature appearance model; EBGM can express much more variation in feature appearance than the Gaussian model utilized by the proposed matching system.

Table 6.2: Matching results for different systems with unoccluded objects, Facegen database. P2P is the point-to-point error in pixels.

Sampling algorithm and prior	P2P error
SMC, full covariance prior	
$P(V_i) = 1$	1.92 ± 0.41
$P(V_i) = 0.5$	1.87 ± 0.35
Gibbs, hierarchical prior	
$P(V_i) = 1$	1.78 ± 0.69
$P(V_i) = 0.5$	1.68 ± 0.45
Gibbs, full covariance prior	
$P(V_i) = 1$	1.47 ± 0.43
$P(V_i) = 0.5$	1.38 ± 0.30
Metropolis, full covariance prior	
$P(V_i) = 0.5$	2.27 ± 0.57
Elastic Bunch Graph Matching	
All images	1.73 ± 0.34

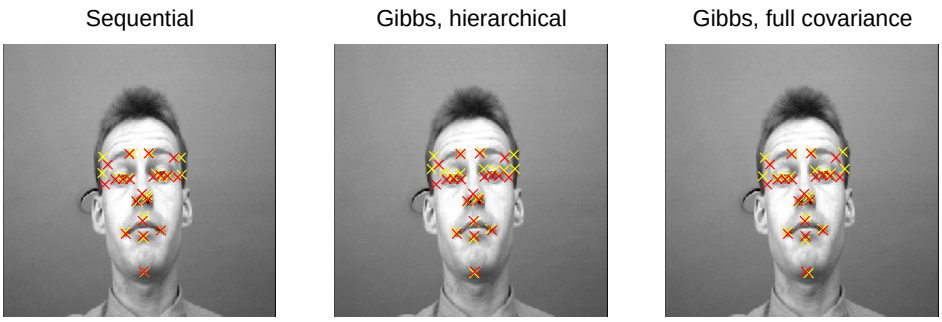

Figure 6.4: Sample matching results, BioID database. The image on the left has been matched with the sequential sampler, the center image with the Gibbs sampler and the hierarchical prior, and the image on the right with the Gibbs sampler and the full covariance prior. The yellow crosses mark the sample means, while the red crosses are the annotated feature positions. Despite scale, lighting and expression differences (compare with Figure 6.1), all samplers have converged to the correct mode.

Table 6.3: Matching results for different systems with unoccluded objects, BioID database. P2P is the point-to-point error in pixels. The total number of test images was $n = 1471$.

Sampling algorithm and prior	P2P error
SMC, full covariance prior	
$P(V_i) = 1$, converged (n=1394)	4.56 ± 1.53
$P(V_i) = 1$, all images	6.06 ± 10.5
$P(V_i) = 0.5$, converged (n=1362)	4.19 ± 1.52
$P(V_i) = 0.5$, all images	6.34 ± 10.8
Gibbs, hierarchical prior	
$P(V_i) = 1$, converged (n=1175)	4.10 ± 1.75
$P(V_i) = 1$, all images	9.35 ± 13.7
$P(V_i) = 0.5$, converged (n=992)	3.37 ± 1.48
$P(V_i) = 0.5$, all images	18.1 ± 25.3
Gibbs, full covariance prior	
$P(V_i) = 1$, converged (n=763)	4.73 ± 1.99
$P(V_i) = 1$, all images	18.2 ± 19.3
$P(V_i) = 0.5$, converged (n=651)	4.51 ± 2.02
$P(V_i) = 0.5$, all images	32.4 ± 28.6
Metropolis, full covariance prior	
$P(V_i) = 0.5$, converged (n=859)	5.16 ± 2.05
$P(V_i) = 0.5$, all images	15.2 ± 15.1
Elastic Bunch Graph Matching	
Converged (n=862)	3.02 ± 0.68
All images	43.5 ± 53.5

Note that for all datasets, including the occlusion model for all-visible objects does not increase the matching errors of the particle filter and the hierarchical prior Gibbs matching greatly. This is encouraging, as in practice it is not possible to assume that all features of an object are detected.

6.5 Occluded objects

The occlusion model was tested with both simulated and real occlusions. For the simulated occlusions, the IMM-DTU database was used due to the good quality of the images and the large number of annotations. The occlusions were generated for each image by replacing parts of the target image with another photograph. Sample results from matching the artificially occluded objects are shown in Figure 6.5. Quantitatively the system was tested by using the same occlusion - a half-occlusion, see the first column of Figure 6.5 - for all the images. The occlusions were assumed independent, so that the regular structure of the occlusion was not utilized in any way. In practice over half of the features could be considered occluded, as the Gabor filter responses of the features close to the occluded area are greatly affected by the edge of the occlusion. The effect of the occlusion to the matching error of the visible features, the occluded features, and the mean error of all features was measured. Furthermore, the weighted average of the feature errors with the detection probabilities as the weights was computed. For comparison, the Elastic Bunch Graph Matching system, which does not contain an occlusion model, was also used to match the images. The results are shown in Table 6.4.

In the images with artificial occlusions, Gibbs sampling with the full covariance prior fails completely - it converged to the correct mode only in 15 of the 37 images. The performance of Metropolis is approximately equal to Elastic Bunch Graph Matching, which does not use any kind of occlusion model. For the sequential system and the Gibbs sampler with the hierarchical prior, the increase in error of the visible features is very small. The matching error of the occluded features as well as the mean error of all the features are higher, as can be expected, since their locations are predicted using only the visible features. For this kind of simple artificial occlusions, the Gibbs sampler with the hierarchical prior slightly outperforms the sequential system. By analyzing the MCMC energies of the Gibbs samplers, it was concluded that the performance differences of Gibbs samplers with different priors were due to the witch's hat nature of the target posterior. With the full covariance prior, the sampler would at first converge towards the correct mode, but would then start to wander and fail to converge. The sampler utilizing the hierarchical prior avoids this as the prior width automatically decreases as the sampler converges, making this wandering off-phenomenon less likely.

The images with real occlusions were taken with a digital camera in uncontrolled office lighting conditions. All 37 IMM-DTU images were used as the training data, and some matching results are shown in Figure 6.6. Here only the sequential system was used, as both Gibbs samplers would get stuck in incorrect modes more often than half the time. This is due to the increased multimodality of the target posterior due to clutter.

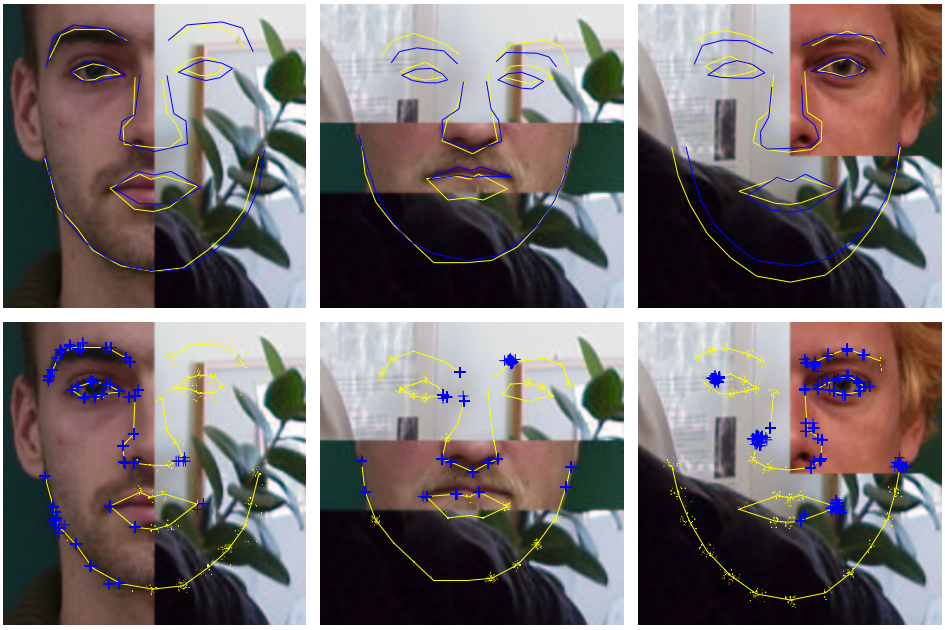

Figure 6.5: Matching results for images with simulated occlusions for the sequential system. In the top row, the light graphs are the sample means and the dark graphs the ground truth. In the bottom row, the light grids are again the sample means. The dark plus signs represent samples for which the probability of feature detection is above 0.5, and the light dots samples for which the probability is below 0.5. Note how even in occluded areas some features are considered detected, and in visible areas some features are considered occluded. Also, as no model for the geometrical properties of the occlusion is included, the system is able to match the objects with several different kinds of occlusions.

For the sequential sampler, the results are promising: in the IMM-DTU images (see Figures 3.11, 4.2 and 6.5) used as the training data the lighting conditions and image quality are very different from the test images, and still the system performs well in most cases. For these images no ground truth was available, which is why the results are demonstrated only visually.

6.6 Multiresolution matching

The multiresolution matching model was tested with the IMM-DTU database and images taken with a digital camera. 9 base frequencies with half an octave spacing were used. Only the sequential sampler was used, as Gibbs sampling and Metropolis algorithms with the full covariance prior did not work very well with the occlusion model even in the standard setting, and since for very large objects

Table 6.4: Matching results for objects with artificial occlusions, IMM-DTU database. P2P is the point-to-point error and P2C is the point-to-curve error in pixels. The weighted error denotes the weighted average of the individual feature errors with the feature detection probabilities as weights.

Sampling algorithm and prior	P2P error	P2C error
SMC, full covariance prior		
Visible features	5.85 ± 1.56	2.94 ± 0.96
Occluded features	8.26 ± 2.33	4.78 ± 1.66
All features	7.14 ± 1.72	3.92 ± 1.19
Weighted error	5.37 ± 1.03	2.66 ± 0.72
Gibbs, hierarchical prior		
Visible features	5.47 ± 1.63	2.62 ± 0.95
Occluded features	6.71 ± 1.98	4.28 ± 1.37
All features	6.14 ± 1.54	3.51 ± 1.04
Weighted error	4.36 ± 1.37	2.27 ± 0.81
Gibbs, full covariance prior		
Visible features	24.1 ± 25.8	20.9 ± 21.5
Occluded features	27.7 ± 23.8	19.1 ± 22.5
All features	26.0 ± 24.7	22.5 ± 20.8
Weighted error	11.8 ± 15.0	7.30 ± 10.8
Metropolis, full covariance prior		
Visible features	7.53 ± 3.82	3.66 ± 1.76
Occluded features	14.7 ± 6.23	8.41 ± 3.48
All features	11.4 ± 4.82	6.20 ± 2.55
Weighted error	6.82 ± 2.83	3.26 ± 1.12
Elastic Bunch Graph Matching		
Visible features	7.89 ± 2.60	3.88 ± 1.91
Occluded features	14.5 ± 2.58	10.0 ± 2.11
All features	11.4 ± 2.15	7.16 ± 1.58

some of the features can be located outside the image, the occlusion model has to be used. The Gibbs sampler with the hierarchical prior was not considered since the purpose behind the multiresolution matching scheme is the novel shape

Figure 6.6: Matching results for images with real occlusions. The graphs show the sample means. The dark portions of the graphs denote features with detection probabilities below 0.5, and the light portions features with detection probabilities above 0.5. The results are promising: most features are matched correctly, although there are errors both in the locations of the features as well as their detection statuses.

covariance structure. Figure 6.7 shows a set of matching results, so that in each image the result with the highest number of effectively detected features is shown. Table 6.5 presents the matching errors over all 37 images and the 6 different object sizes illustrated in Figure 6.7, and Table 6.6 shows the relative errors with object size adjusted to 100%.

Even though the model is in principle weaker than the standard full covariance model, as some of the dependencies between the feature locations are not modeled, the increase in matching error when matching the 100% size object is very small. As can be expected, the absolute error increases when the scale increases, and the relative error decreases - with smaller images, it becomes harder and sometimes impossible to detect some of the features and hence there is less information to guide the matching.

The system was tested also with real images, with the IMM-DTU database as the training data. Some results are shown in Figure 6.8 - again the different size object are robustly matched even though the training data is different from the test data.

Table 6.5: Multiresolution matching results for objects of different size, IMM-DTU database, absolute error. P2P is the point-to-point error and P2C is the point-to-curve error in pixels.

Object size with respect to training data	P2P error	P2C error
12%	1.26 ± 0.37	0.82 ± 0.30
20%	2.02 ± 1.24	1.31 ± 0.87
35%	2.48 ± 0.62	1.40 ± 0.51
59%	3.44 ± 0.69	1.81 ± 0.60
100%	5.86 ± 1.08	3.04 ± 0.80
219%, visible features	10.1 ± 3.42	6.15 ± 1.54
219%, all features	17.4 ± 6.55	11.2 ± 4.59

Table 6.6: Multiresolution matching results for objects of different size, IMM-DTU database, relative error. P2P is the point-to-point error and P2C is the point-to-curve error in pixels.

Object size with respect to training data	P2P error	P2C error
12%	10.8 ± 3.14	7.02 ± 2.58
20%	10.3 ± 6.33	6.69 ± 4.44
35%	7.05 ± 1.75	3.98 ± 1.45
59%	5.88 ± 1.18	3.08 ± 1.02
100%	5.86 ± 1.08	3.04 ± 0.80
219%, visible features	5.06 ± 1.56	2.81 ± 0.70
219%, all features	7.94 ± 2.99	5.12 ± 2.10

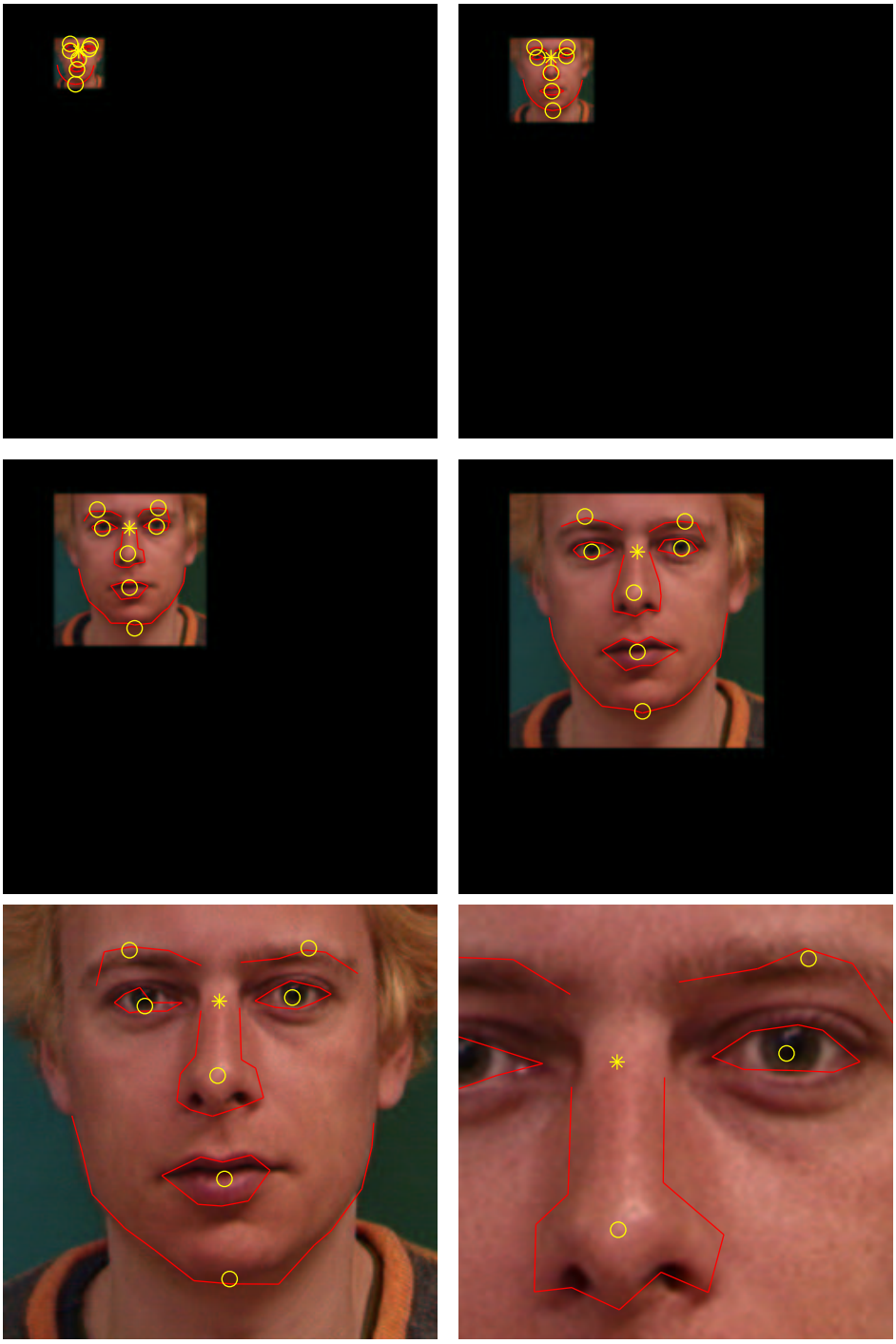

Figure 6.7: A set of matches with objects of different sizes. The graphs show the mean of the samples obtained using the base resolution with the largest number of effectively detected features.

Figure 6.8: A set of matched objects in real-world conditions. Again, the shown graphs are obtained using the base resolution with the largest number of effectively detected features.

6.7 Multiple object instances

As discussed in Chapter 5, the sequential sampler should in principle be able to match multiple instances of an object class simply by representing the whole multimodal posterior. To demonstrate this, the system was tested on a composite image with multiple faces. Figure 6.9 shows an intermediate result when three features (in random order) have been matched. The sequential scheme finds all the major modes of the posterior distribution - this would be simply impossible for MCMC-based sampling methods or approaches such as AAM, for which the changing of mode is very improbable. When all features are matched, also sequential matching will result in all particles being located in the mode of the posterior with the largest probability mass due to the mass differences being large in the joint posterior.

The model of Figure 6.9 is designed for the matching of a single object: the modes simply indicate where the one face could be. By utilizing the model presented in Section 5.5, all the objects can be matched without problem via repeated sampling, which is illustrated in Figures 6.10 and 6.11.

Figure 6.9: Matching with multiple object instances, no multiple-object model. The image is formed by combining 4 facial images into a single image, which leads to a joint posterior with four major modes. 3 features were matched, and all 4 modes are included in the particle representation. Note the difference in the number of particles from different modes. The light graphs represent the prior mean.

6.8 Initialization

It is important to note that the results of AAM and the MCMC sampling algorithms are sensitive to their initialization - poor initialization leads to poor results. For AAM, Stegmann (2002) used an initial displacement of 10% in x and y relative to the width and height of the object, which is a rather well-informed initial guess, although it should be noted that more robust initialization schemes for AAM have been proposed (Edwards et al., 1999). The initial displacement for the Gibbs and Metropolis samplers was about the same. To demonstrate the problems Gibbs sampling has with poor initalization, the Gibbs sampler with the full covariance prior was tested with a markedly larger initial displacement, with the result in Figure 6.12. With the usual initialization, the matching error for this image is about 4.5 pixels (P2P)/1.8 pixels (P2C). Here the errors are 10.4/5.2 pixels, for a clear reason: the chin features have stayed in their initial incorrect mode. The sequential algorithm and, to a lesser extent, Gibbs sampling with the hierarchical prior do not suffer from initialization problems - SMC because no initialization is needed as the first feature is simply drawn from its likelihood, the hierarchical prior because it automatically enlarges the prior width when the match is poor (that is, when the graph is very dissimilar from the prior). However, even with the

Figure 6.10: Matching with multiple object instances using the multiple object model. The matching has been performed 4 times, with the sample means depicted by the yellow, red, blue, and green graphs. The corresponding numbers of effectively detected features were approximately 55, 52, 50 and 49. After the fifth run, this value dropped to 10.

hierarchical prior, the starting point of the Gibbs algorithm in the parameter space must still be in the neighborhood of the correct mode.

6.9 Computational requirements

At the present stage, the proposed system is very demanding computationally, as are all sampling-based schemes. Due to Gabor filtering, the computation of the feature appearance model and the likelihood fields takes about 2 minutes on a regular Pentium IV PC, and with an unoptimized MATLAB implementation, the sequential matching of a single image takes about 3 minutes. The multiresolution and multiple object models are computationally even more demanding, as they require that the entire matching process is performed multiple times.

In the case of sequential Monte Carlo, the computational cost could be lowered by parallelization, as the particles are independent except during resampling. As its execution time is linear in the number of particles, parallelization of the sequential algorithm would reduce the computational load dramatically. The Gibbs and Metropolis algorithms are not parallelizable, since in accordance with their Markov chain nature they use the whole sample from the previous time step to produce the new sample.

Figure 6.11: Matching with multiple object instances using the multiple object model, real image. The yellow graph represents the sample mean of the first run of the sampler, and the red graph the sample mean of the second run of the sampler with the area around the first mean marked occluded. The number of effectively detected features was approximately 46 after both runs. After a third run this value dropped to 11.

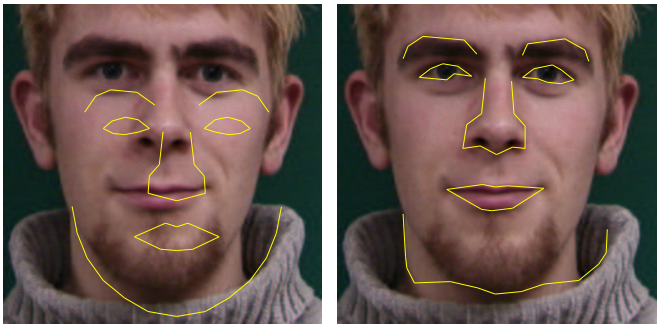

Figure 6.12: Gibbs sampling matching failure with poor initialization. The image on the left shows the initial position of the object graph, while the image on the right shows the matching result. The features of the chin have clearly stuck in the wrong mode of the posterior.

6.10 Analysis of the facial expression model

To analyze and assess the capabilities of the facial expression model presented in Section 4.5, a set of reconstruction-related tests was performed. The shape and expression bases were computed using the measured tracking results and the principal components inspected visually. The first two expression principal components are illustrated in Figure 6.13. The measured tracks were then projected onto the obtained bases and the coordinates analyzed to see whether the assumed separability of person-dependent shape and person-independent expression held. Some projection coordinate plots are shown in Figures 6.14 and 6.15. It would seem that the separability assumption holds: the shape space coordinates remain in most cases approximately equal for the same person, while the expression space coordinates are similar for the same expression.

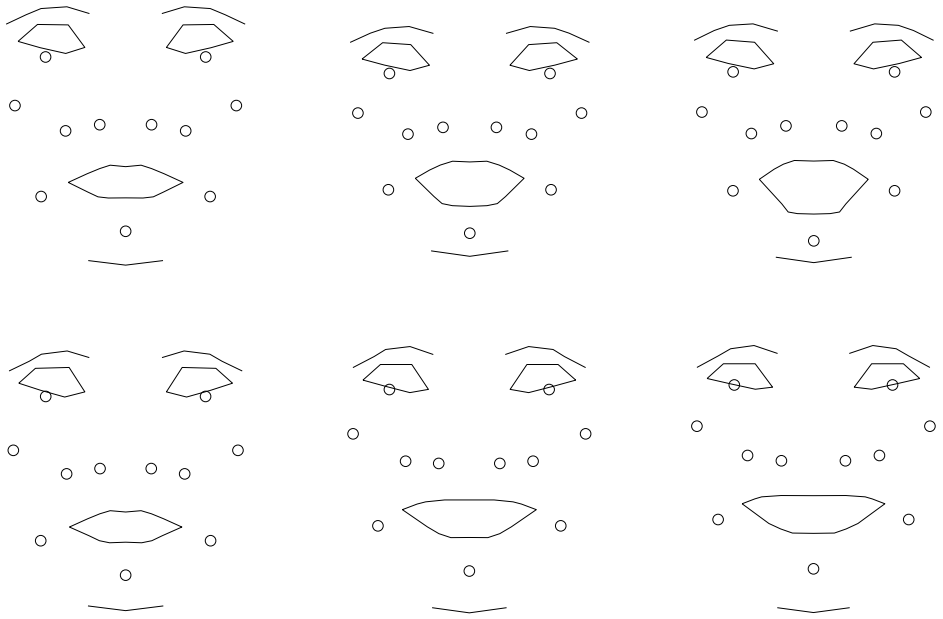

Figure 6.13: The first two expression principal components. The components are shown at time steps $t = 1$, $t = 1/2t_f$ and $t = t_f$. The first component (row 1) is mainly related to opening of the mouth, while the second component (row 2) seems to be a smile.

The actual reconstruction was done by projecting the measured tracks into the shape and expression spaces and then back to the original track space to obtain the reconstructed tracks $\mathbf{X}^*$,

$$\mathbf{X}^* = \mathbf{1} \otimes (\mathbf{m} + \mathbf{S}\mathbf{S}^T\mathbf{X}^1) + \mathbf{E}\mathbf{E}^T\mathbf{X}. \tag{6.1}$$

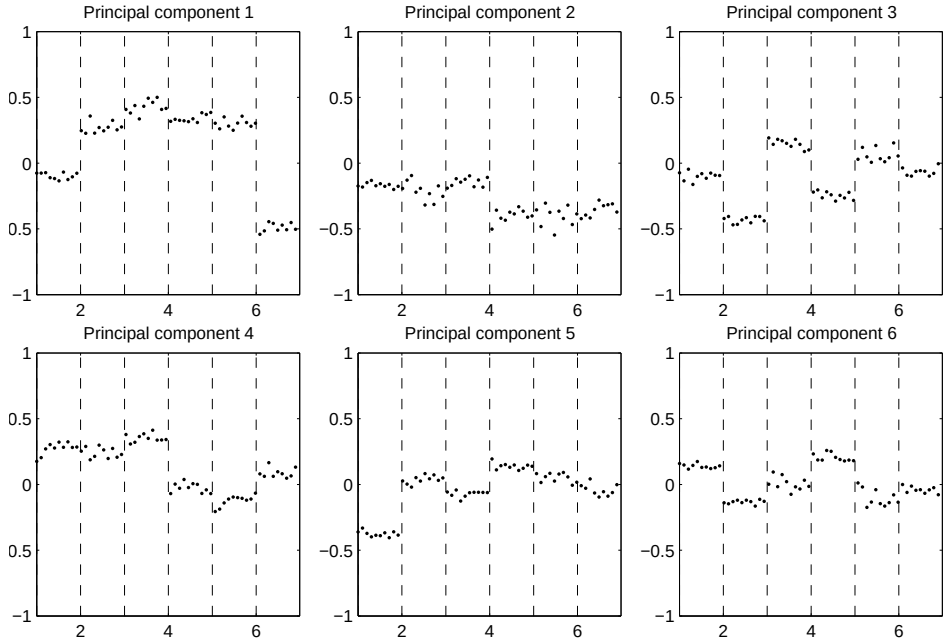

Figure 6.14: First six shape space coordinates for the 60 initial graphs $\mathbf{X}^1$. The x-axis is the person index from 1 to 6. Each image corresponds to a single principal component with 10 coordinate instances for each person. The dashed lines indicate change of person. In most cases, the persons are clearly distinct from one another, and the coordinates are similar for the same person.

15 principal components were used for the shape space and 6 for the expression space. In both cases this amounted to ca. 99% of the total variance. The original and reconstructed tracks were compared both visually and numerically. Two sample reconstructions are shown in Figure 6.16, and Table 6.7 contains mean reconstruction errors per unit of scale (see Section 5.2) (for the unscaled size 256×256 training data the scale was around 50). In Figure 6.16 the textures were obtained by morphing the time-corresponding original frame according to the reconstructed graph with the feature-based image morphing algorithm of Beier and Neely (1992).

The reconstruction results are rather promising: visually, the reconstructed expressions are easily recognizable and contain little distortion, and the numerical errors are low - for the original data, the mean error is below 2 pixels for most cases.

The model has several practical applications. In its probabilistic form (Eq. 4.30) the model can be used directly as a prior in expression-dependent object

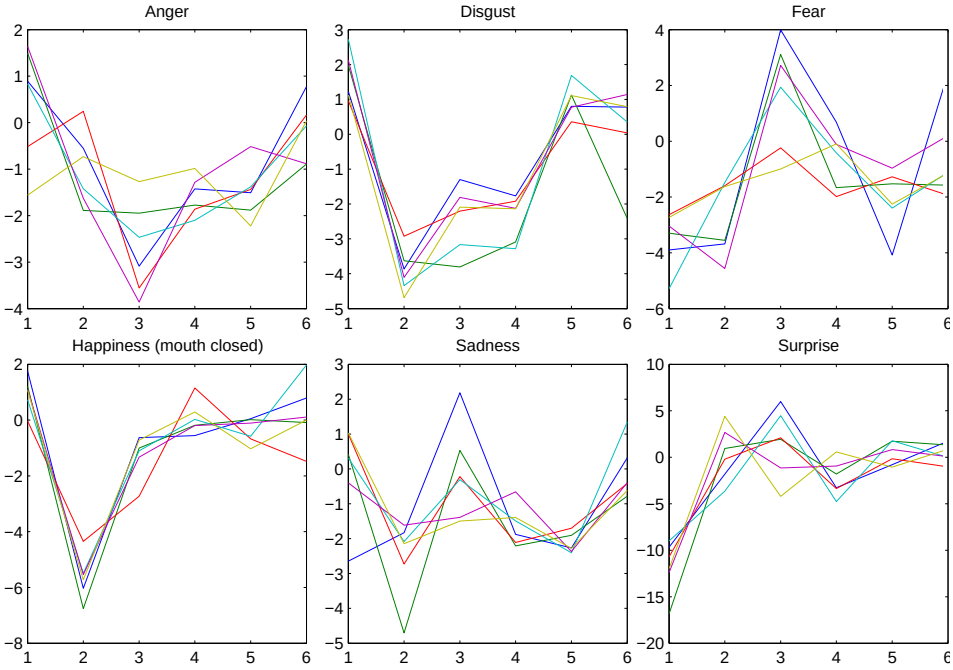

Figure 6.15: First six expression space coordinates for the six basic expressions. The x-axis is principal component index. Each line corresponds to a single expression instance. The expressions are similar to each other across persons, although there are differences, too. For example, the coordinates for the expressions of happiness show more similarity than the expressions of fear. The similar situation is encountered in everyday life - expressions of happiness are much more alike than expressions of fear.

matching. Furthermore, the obtained expressions could be implemented on a Talking Head model (Frydrych et al., 2003). The proposed model includes the dynamics of the expressions, and hence should be an improvement over the previously used expression model. Another interesting research topic is to compare the obtained expression principal components (Fig. 6.13) and FACS action units (Ekman et al., 1978) to see whether there is any systematic correspondence.

Table 6.7: Mean reconstruction error per unit of scale. t denotes the time step, and t_f marks the last image of a stream.

Expression	$t = 1$	$t = 1/2t_f$	$t = t_f$	$t = \{1...t_f\}$
Anger	0.0070	0.0267	0.0353	0.0214
Disgust	0.0071	0.0225	0.0296	0.0198
Fear	0.0082	0.0274	0.0353	0.0221
Happiness (mouth open)	0.0069	0.0250	0.0336	0.0208
Happiness (mouth closed)	0.0061	0.0246	0.0356	0.0212
Sadness	0.0073	0.0240	0.0311	0.0206
Surprise	0.0071	0.0265	0.0322	0.0229
Happiness + surprise	0.0072	0.0337	0.0411	0.0251
Happiness + disgust	0.0078	0.0282	0.0385	0.0246
Mouth opening	0.0063	0.0221	0.0258	0.0174
All expressions	0.0071	0.0261	0.0338	0.0216

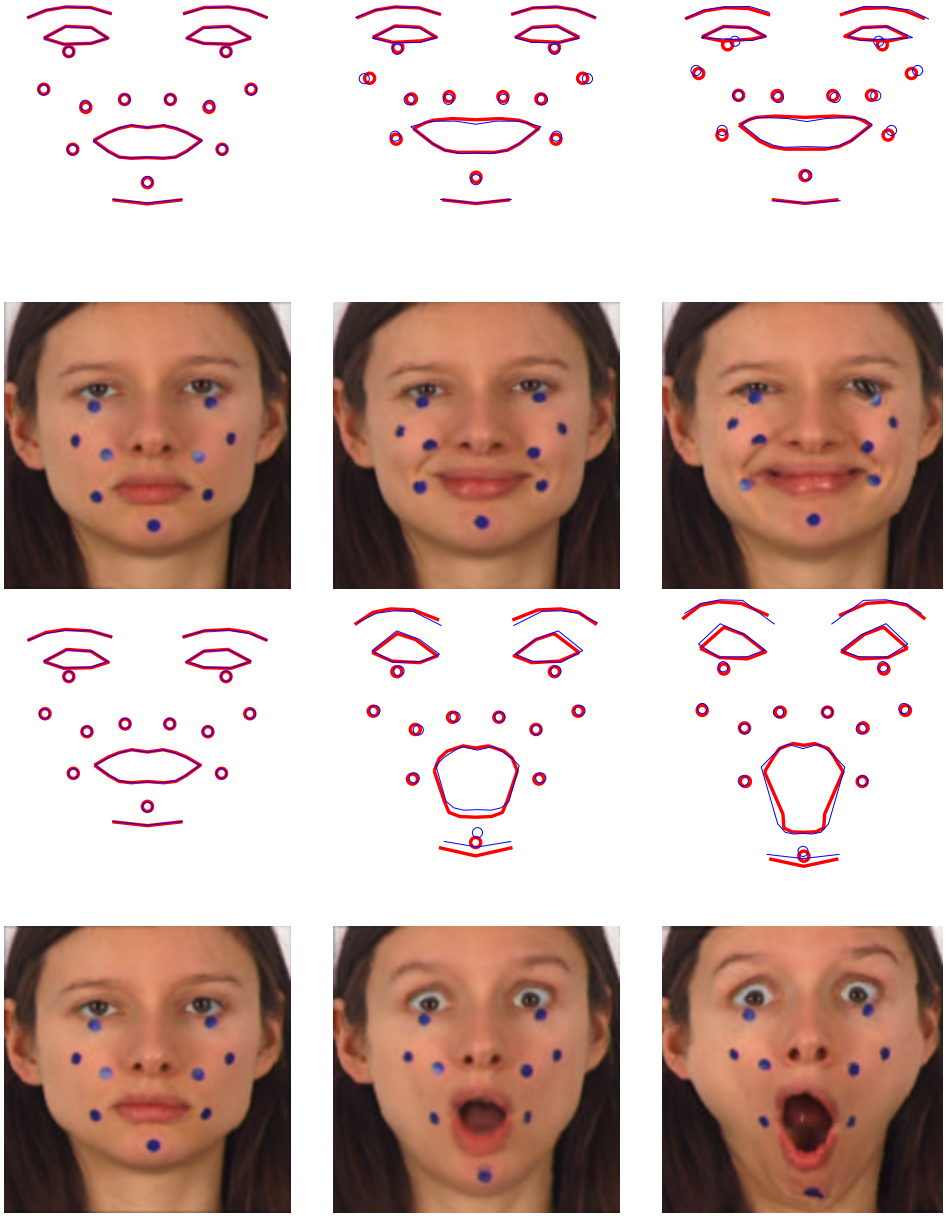

Figure 6.16: Reconstruction results for the "happiness (mouth closed)" (upper two rows) and "surprise" (lower two rows) expressions. The depicted time steps are $t = 1$, $t = 1/2t_f$ and $t = t_f$. The thinner graphs show the original data and the thicker graphs the reconstructed expressions, while the images show the results of morphing the video frame corresponding to the time step according to the reconstructed graph. The expressions are clearly recognizable, and there are few distortions.

Chapter 7

Conclusion

This thesis has presented a Bayesian probabilistic framework for the visual matching of objects and their pertinent features. Novel models and methods have been introduced for various parts of the matching problem: feature appearance, object shape, and analysis of the resulting posterior distributions. The statistical model for Gabor filter responses is robust against changes in feature appearance and allows the direct computation of pixel likelihoods instead of relying on an intermediate ad hoc transformation from feature similarity to likelihood. The various object shape priors are used to facilitate the matching in different situations: the hierarchical prior can be used even when there is very little training data available, as the deviations from the mean are assumed independent and their extent is estimated from the observed image, while the full covariance prior is suitable for object classes with specific structural variations and of which there are many examples available. The sampling algorithms, as applied to the object matching problem, also serve different purposes. Sequential Monte Carlo outperforms the other methods when the posterior distribution is multimodal or otherwise difficult, Gibbs sampling is superb at exploring single modes of the posterior, whereas Metropolis-Hastings can be used no matter how the posterior is parametrized.

In addition to the basic model and the contributions to its various parts, the thesis has introduced extensions to the matching model to deal with situations such as occlusion, large global scale changes, and multiple object instances. All of these are handled by altering the probability model to accommodate for the changed matching environment instead of utilizing ad hoc, situation-dependent approaches. This is a key advantage of the Bayesian approach to object matching: the main pieces of the puzzle and their analysis are always the same, and if something changes, it is enough to deal with the change in the corresponding module, not rewrite the whole algorithm. Another advantage of the Bayesian point of view is its ability to express uncertainty and lack of knowledge mathematically. Good examples of this are the hierarchical prior structure and the proposed occlu-

sion model, in which the Bayesian approach makes it straightforward to perform matching even though some aspects of the situation are unknown.

The two most prominent published approaches to object matching are Active Appearance Models (Cootes et al., 2001) and Elastic Bunch Graph Matching (Wiskott et al., 1997). As discussed in Chapter 6, in the tests undertaken for this thesis the results of the three systems were generally equivalent. There are certain aspects in which AAM and EBGM are superior to the proposed system: handling large rotations and/or view angle changes of the target objects, and computational requirements. With its multimodal approach to feature similarity computation, EBGM has no problems with any kind of rotations or pose changes, as long as similar non-frontal images have been included in the training data. Also AAM can handle these very well, with the same reservation. The computational requirements of AAM and EBGM are clearly lower than those of the proposed system, which is a natural consequence of the optimization vs. sampling dichotomy. On the other hand, as mentioned above, it is more straightforward to extend the proposed matching model to otherwise nonstandard matching situations, such as occlusions or large scale changes. AAM and EBGM usually require some kind of heuristic or an external model to deal with such conditions.

In summary, the aims of this thesis - to present a Bayesian probabilistic solution to the object matching problem and to analyze the capabilities of such a solution - have been met. While there clearly are some unsolved issues within the proposed system, the Bayesian approach to object matching seems a viable one and merits further research.

References

Abboud, B. and Davoine, F. (2004). Appearance factorization for facial expression analysis. In *Proceedings of the British Machine Vision Conference*.

Ahlberg, J. (2002). An active model for facial feature tracking. *EURASIP Journal on Applied Signal Processing*, 6:566–571.

Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. *Computer Graphics*, 26(2):35–42.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Blake, A. and Isard, M. (1998). *Active Contours*. Springer.

Bourel, F., Chibelushi, C. C., and Low, A. A. (2000). Robust facial feature tracking. In *Proceedings of the British Machine Vision Conference*.

Brandt, S. and Heikkonen, J. (2001). Multi-resolution matching of uncalibrated images utilizing epipolar geometry and its uncertainty. In *Proceedings of the IEEE International Conference on Image Processing*.

Brunelli, R. and Poggio, T. (1995). Template matching: Matched spatial filters and beyond. Technical Report AIM-1549, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Brzezniak, Z. and Zastawniak, T. (1999). *Basic Stochastic Processes*. Springer Undergraduate Mathematics Series. Springer.

Calder, A. J., Burton, A. M., Miller, P., Young, A. W., and Akamatsu, S. (2001). A principal component analysis of facial expressions. *Vision Research*, 41(9):1179–1208.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Casasent, D. and Furman, A. (1977). Optimization of parameters in matched spatial filter synthesis. *Applied Optics*, 16(6):1662–1669.

Chatfield, C. and Collins, A. J. (1995). *Introduction to Multivariate Analysis*. Texts in Statistical Science. Chapman & Hall.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Cipolla, R. and Blake, A. (1990). The dynamic analysis of apparent contours. In *Proceedings of the International Conference on Computer Vision*.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Coughlan, J. and Shen, H. (2004). Shape matching with belief propagation: Using dynamic quantization to accommodata occlusion and clutter. In *Proceedings of Workshop on Generative Model-Based Vision*.

Cox, G. S. (1995). Template matching and measures of match in image processing. Technical report, Department of Computer Science, University of Cape Town.

Daugman, J. G. (1988). Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

Edwards, G. J., Cootes, T. F., and Taylor, C. J. (1999). Advances in active appearance models. In *Proceedings of the International Conference on Computer Vision*.

Ekman, P. (1984). Expression and the nature of emotion. In Scherer, K. and Ekman, P., editors, *Approaches to Emotion*. Lawrence Erlbaum.

Ekman, P., Friesen, W., and Hager, J. (1978). *Facial Action Coding System*. Consulting Psychologists Press.

Faraway, J. J. (2004). Modeling continuous shape change for facial animation. *Statistics and Computing*, 14(4):357–363.

Felzenszwalb, P. F. (2003). Representation and detection of deformable shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Fox, D., Thrun, S., Burgard, W., and Dellaert, F. (2001). Particle filters for mobile robot localization. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*. Springer.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906.

Frey, B. J., Koetter, R., and Petrovic, N. (2001). Very loopy belief propagation for unwrapping phase images. In *Proceedings of the Neural Information Processing Systems Conference*.

Frydrych, M., Kätsyri, J., Dobsik, M., and Sams, M. (2003). Toolkit for animation of Finnish talking head. In *Proceedings of the International Conference on Auditory-Visual Speech Processing*.

Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. R. (2004). *Bayesian Data Analysis*. Chapman & Hall, second edition.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Gentle, J. E. (1998). *Random Number Generation and Monte Carlo Methods*. Springer Series in Statistics and Computing. Springer.

Ghahramani, Z. and Beal, M. J. (2001). Graphical models and variational methods: Theory and practice. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods*. The MIT Press.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Gonzalez, R. C. and Woods, R. E. (1993). *Digital Image Processing*. Addison-Wesley.

Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society, Series B*, 53(2):285–339.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F*, 140(2):107–113.

Gross, R., Matthews, I., and Baker, S. (2004). Constructing and fitting active appearance models with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Hadid, A., Pietikäinen, M., and Martinkauppi, B. (2002). Color-based face detection using skin locus model and hierarchical filtering. In *Proceedings of the International Conference on Pattern Recognition*.

Hamouz, M., Kittler, J., Kämäräinen, J.-K., Paalanen, P., Kälviäinen, H., and Matas, J. (2005). Feature-based affine-invariant localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1490–1495.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*.

Hassani, S. (1998). *Mathematical Physics. A Modern Introduction to its Foundations.* Springer.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Horaud, R. and Skordas, T. (1989). Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 11(11):1168–1180.

Isard, M. (2003). PAMPAS: Real-valued graphical models for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jain, A. K. and Farrokhnia, F. (1990). Unsupervised texture segmentation using Gabor filters. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*.

Jesorsky, O., Kirchberg, K. J., and Frischholz, R. W. (2001). Robust face detection using the Haussdorf distance. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258.

Jones, M. J. and Rehg, J. M. (1998). Statistical color models with application to skin detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Karlsson, R. and Gustafsson, F. (2001). Monte Carlo data association for multiple target tracking. In *Proceedings of the IEE International Seminar Target Tracking: Algorithms and Applications*.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kersten, D. and Yuille, A. (2003). Bayesian models of object perception. *Current Opinion in Neurobiology*, 13:150–158.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational & Graphical Statistics*, 5(1):1–25.

Kämäräinen, J.-K., Ilonen, J., Paalanen, P., Hamouz, M., Kälviäinen, H., and Kittler, J. (2005). Object evidence extraction using simple Gabor features and statistical ranking. In *Proceedings of the Scandinavian Conference on Image Analysis*.

Koenderink, J. J. and van Doorn, A. J. (1987). Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Kyrki, V., Kämäräinen, J.-K., and Kälviäinen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters*, 25(3):311–318.

Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–310.

Lampinen, J. and Oja, E. (1995). Distortion tolerant pattern recognition based on self-organizing feature extraction. *IEEE Transactions on Neural Networks*, 6(3):539–547.

Lee, T. S. and Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America A*, 20(7):1434–1448.

Li, F.-F., Fergus, R., and Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the International Conference on Computer Vision*.

Liang, F. (2003). Use of sequential structure in simulation from high-dimensional systems. *Physical Review E*, 67(056101).

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the Image Understanding Workshop*.

Mallat, S. (1998). *A Wavelet Tour of Signal Processing*. Academic Press.

Matthews, P. (1993). A slowly mixing Markov chain with implications for Gibbs sampling. *Statistics & Probability Letters*, 17:231–236.

McKenna, S. J., Gong, S., Würtz, R. P., Tanner, J., and Banin, D. (1997). Tracking facial feature points with Gabor wavelets and shape models. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*.

Metropolis, N., Rosenbluth, A., Rosenbluth, R., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Miao, J., Yin, B., Wang, K., Shen, L., and Chen, X. (1999). A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template. *Pattern Recognition*, 32(7):1237–1248.

Milton, J. S. and Arnold, J. C. (1995). *Introduction to Probability and Statistics*. McGraw-Hill.

Mitchell, S. C., Bosch, J. G., Lelieveldt, B. P. F., van der Geest, R. J., Reiber, J. H. C., and Sonka, M. (2002). 3-D active appearance models: segmentation of cardiac MR and ultrasound images. *IEEE Transactions on Medical Imaging*, 21(9):1167–1178.

Moghaddam, B., Wahid, W., and Pentland, A. (1998). Beyond eigenfaces: Probabilistic matching for face recognition. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*.

Mumford, D. (1996). Pattern theory: a unifying perspective. In Knill, D. C. and Richards, W., editors, *Perception as Bayesian Inference*. Cambridge University Press.

Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, 31(3):705–767.

Palmer, S. E. (1999). *Vision Science - Photons to Phenomenology*. The MIT Press.

Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Pérez, P., Blake, A., and Gangnet, M. (2001). Jetstream: Probabilistic contour extraction with particles. In *Proceedings of the IEEE International Conference on Computer Vision*.

Ridgeway, G. and Madigan, D. (2003). A sequential Monte Carlo method for Bayesian analysis of massive datasets. *Data Mining and Knowledge Discovery*, 7(3):301–319.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, second edition.

Rucklidge, W. J. (1997). Efficiently locating objects using the Haussdorf distance. *International Journal of Computer Vision*, 24(3):251–270.

Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sigal, L., Isard, M., Sigelman, B. H., and Black, M. J. (2003). Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Proceedings of the Neural Information Processing Systems Conference*.

Steger, C. (2002). Occlusion, clutter, and illumination invariant object recognition. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIV, part 3A.

Stegmann, M. B. (2002). Analysis and segmentation of face images using point annotations and linear subspace techniques. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark.

Sudderth, E. B., Ihler, A. T., Freeman, W. T., and Willsky, A. S. (2003). Nonparametric belief propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sullivan, J., Blake, A., Isard, M., and MacCormick, J. (2001). Bayesian object localisation in images. *International Journal of Computer Vision*, 44(2):111–135.

Tamminen, T., Kätsyri, J., Frydrych, M., and Lampinen, J. (2005). Joint modeling of facial expression and shape from video. In *Proceedings of the Scandinavian Conference on Image Analysis*.

Tamminen, T. and Lampinen, J. (2003a). Bayesian object matching with hierarchical priors and Markov chain Monte Carlo. In Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M., editors, *Bayesian Statistics 7*. Oxford University Press.

Tamminen, T. and Lampinen, J. (2003b). Learning an object model for feature matching in clutter. In *Proceedings of the Scandinavian Conference on Image Analysis*.

Tamminen, T. and Lampinen, J. (2004). A Bayesian occlusion model for sequential object matching. In *Proceedings of the British Machine Vision Conference*.

Tamminen, T. and Lampinen, J. (2005). Sequential Monte Carlo for Bayesian matching of objects with occlusions. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622.

Tomasi, C. and Kanade, T. (1991). Detection and tracking of feature points. Technical Report CMU-CS-91-132, Carnegie Mellon University.

Tu, Z. and Zhu, S.-C. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673.

Welch, G. and Bishop, G. (1997). An introduction to the Kalman filter. Technical Report TR-95-041, Department of Computer Science, University of North Carolina at Chapel Hill.

Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.

Würtz, R. P. (1997). Object recognition under translations, deformations, and changes in background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):769–775.

Xu, G. and Zhang, Z. (1996). *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, volume 6 of *Computational Imaging and Vision*. Kluwer Academic Publishers.

Yang, J., Lu, W., and Waibel, A. (1998). Skin-color modeling and adaptation. In *Proceedings of the Asian Conference on Computer Vision*.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2004). Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2004-040, Mitsubishi Electric Research Laboratories.

Zhang, J., Collins, R., and Liu, Y. (2004). Representation and matching of articulated shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, Z. (1994). Token tracking in a cluttered scene. Technical Report RR-2072, Institut National de Recherce en Informatique et en Automatique.

Zhong, Y., Jain, A. K., and Dubuisson-Jolly, M.-P. (2000). Object tracking using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):544–549.

Zhou, Y., Gu, L., and Zhang, H.-J. (2003). Bayesian tangent shape model: estimating shape and pose parameters via Bayesian inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhu, S.-C., Wu, Y. N., and Mumford, D. (1997). Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(9):1627–1660.

ISBN 951-22-7906-1 (printed)
ISBN 951-22-7907-X (PDF)
ISSN 1455-0474
Picaset Oy, Helsinki 2005