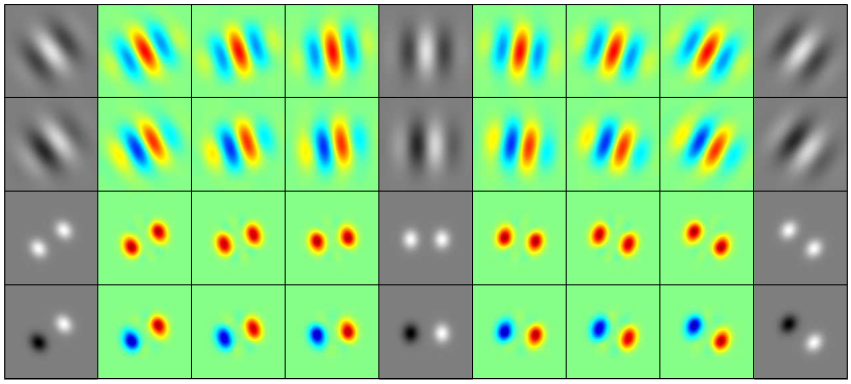
Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2007 REPORT B67

PROBABILISTIC METHODS FOR POSE-INVARIANT RECOGNITION IN COMPUTER VISION

Ilkka Kalliomäki



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 2007 REPORT B67

PROBABILISTIC METHODS FOR POSE-INVARIANT RECOGNITION IN COMPUTER VISION

Ilkka Kalliomäki

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 C at Helsinki University of Technology (Espoo, Finland) on the 2nd of November, 2007, 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.tkk.fi/Diss/2007/isbn9789512289967/

E-mail: ilkka.kalliomaki@tkk.fi

ISBN 978-951-22-8995-0 (printed)
ISBN 978-951-22-8996-7 (PDF)
ISSN 1455-0474
PicaSet Oy
Espoo 2007

Abstract

This thesis is concerned with two central themes in computer vision, the properties of oriented quadrature filters, and methods for implementing rotation invariance in an object matching and recognition system. Objects are modeled as combinations of local features, and human faces are used as the reference object class. The topics covered include optimal design of filter banks for feature detection and object recognition, modeling of pose effects in filter responses and the construction of probability-based pose-invariant object matching and recognition systems employing oriented filters.

Gabor filters have been derived as information-theoretically optimal bandpass filters, simultaneously maximizing the localization capability in space and spatial-frequency domains. Steerable oriented filters have been developed as a tool for reducing the amount of computation required in rotation invariant systems. In this work, the framework of steerable filters is applied to Gabor-type filters and novel analytical derivations for the required steering equations for them are presented. Gabor filters and some related filters are experimentally shown to be approximately steerable with low steering error, given suitable filter shape parameters. The effects of filter shape parameters in feature localization and object recognition are also studied using a complete feature matching system.

A novel approach for modeling the pose variation of features due to depth rotations is introduced. Instead of manifold learning methods, the use synthetic data makes it possible to apply simpler regression modeling methods. The use of synthetic data in learning the pose models for local features is a central contribution of the work.

The object matching methods considered in the work are based on probabilistic reasoning. The required object likelihood functions are constructed using feature similarity measures, and random sampling methods are applied for finding the modes of high probability in the likelihood probability distribution functions. The Population Monte Carlo algorithm is shown to solve successfully pose estimation problems in which simple Metropolis and Gibbs sampling methods give unsatisfactory performance.

Tiivistelmä

Tämä väitöskirja käsittelee kahta keskeistä tietokonenäön osa-aluetta, signaalin suunnalle herkkien kvadratuurisuodinten ominaisuuksia, ja näkymäsuunnasta riippumattomia menetelmiä kohteiden sovittamiseksi malliin ja tunnistamiseksi. Kohteet mallinnetaan paikallisten piirteiden yhdistelminä, ja esimerkkikohdeluokkana käytetään ihmiskasvoja. Työssä käsitellään suodinpankin optimaalista suunnittelua piirteiden havaitsemisen ja kohteen tunnistuksen kannalta, näkymäsuunnan piirteissä aiheuttamien ilmiöiden mallintamista sekä edellisen kaltaisia piirteitä käyttävän todennäköisyyspohjaisen, näkymäsuunnasta riippumattomaan havaitsemiseen kykenevän kohteidentunnistusjärjestelmän toteutusta.

Gabor-suotimet ovat informaatioteoreettisista lähtökohdista johdettuja, aika- ja taajuustason paikallistamiskyvyltään optimaalisia kaistanpäästösuotimia. Nk. ohjattavat (*steerable*) suuntaherkät suotimet on kehitetty vähentämään laskennan määrää tasorotaatioille invarianteissa järjestelmissä. Työssä laajennetaan ohjattavien suodinten teoriaa Gabor-suotimiin ja esitetään Gabor-suodinten ohjaukseen vaadittavien approksimointiyhtälöiden johtaminen analyyttisesti. Kokeellisesti näytetään, että Gabor-suotimet ja eräät niitä muistuttavat suotimet ovat sopivilla muotoparametrien arvoilla likimäärin ohjattavia. Lisäksi tutkitaan muotoparametrien vaikutusta piirteiden havaittavuuteen sekä kohteen tunnistamiseen kokonaista kohteidentunnistusjärjestelmää käyttäen.

Piirteiden näkymäsuunnasta johtuvaa vaihtelua mallinnetaan suoraviivaisesti regressiomenetelmillä. Näiden käyttäminen monisto-oppimismenetelmien (*manifold learning methods*) sijaan on mahdollista, koska malli muodostetaan synteettisen datan avulla. Työn keskeisiä kontribuutioita on synteettisen datan käyttäminen paikallisten piirteiden näkymämallien oppimisessa.

Työssä käsiteltävät mallinsovitusmenetelmät perustuvat todennäköisyyspohjaiseen päättelyyn. Tarvittavat kohteen uskottavuusfunktiot muodostetaan piirteiden samankaltaisuusmitoista, ja uskottavuusfunktion suuren todennäköisyysmassan keskittymät löydetään satunnaisotantamenetelmillä. Population Monte Carlo -algoritmin osoitetaan ratkaisevan onnistuneesti asennonestimointiongelmia, joissa Metropolis- ja Gibbs-otantamenetelmät antavat epätyydyttäviä tuloksia.

Preface

This thesis is the result of my research at the Laboratory of Computational Engineering at Helsinki University of Technology during the years 2002–2006. The work has been funded by the ComMIT graduate school and the Academy of Finland Centre of Excellence in Computational Science and Engineering. Additionally, the research has been financially supported by 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 most sincere respect and gratitude to Prof. Jouko Lampinen for his instruction and supervision, as well as his inexhaustible enthusiasm on highly diverse research topics, ranging from the composition of rock minerals to the inner workings of the human brain. I also wish to thank Dr. Toni Tamminen, Dr. Aki Vehtari, Dr. Timo Kostiainen, Miika Toivanen, Dr. Michael Frydrych and Dr. Simo Särkkä for research collaboration and comments on parts of the work, and Liisa Kuivasmäki for comments which improved the readability and linguistic quality of the work. I wish to thank Dr. Ville Kyrki and Prof. Olli Silvén for reviewing the thesis and for their comments and suggestions. I am grateful to Dr. Veit Schenk and Professor Sir Mike Brady for interesting and fruitful discussions during my two-month stay at University of Oxford. I wish to thank Prof. Kimmo Kaski for providing the concrete facilities for my research, and Eeva Lampinen, Aino Järvenpää and Kaija Virolainen for taking care of practicalities.

Thanks are due to all of the personnel in the laboratory, with several of whom I have had fun also outside office hours and even national borders. My friends on the other side of the laboratory walls deserve thanks for providing a balance with research in various activities. Mom and Dad, thank you for your continuous support.

Ilkka Kalliomäki

Contents

List of Symbols

a_i	Amplitude of a filter jet component
B	Filter radial spacing parameter
$D(\cdot, \cdot)$	Angular distance measure
e	Error
E_s	Steering error
f_c	Center frequency of a filter
$f()$	Generic function; oriented filter in spatial coordinates
$\mathcal{F}$	Fourier transform
$g()$	Oriented filter in spatial coordinates
$G()$	Oriented filter in spatial-frequency space
$\mathbf{G}$	Gram matrix of basis filters
$h(t)$	Impulse response (of a filter)
$\mathcal{H}$	One-dimensional Hilbert transform
I	Two-dimensional image
$\mathcal{J}$	Unnormalized filter jet
J	Normalized filter jet
$\mathbf{J}$	Collection of several normalized filter jets
$J(\cdot\|\cdot)$	Jumping distribution
$k(\theta)$	Steering function
$\mathbf{k}(\theta)$	Vector of steering functions
$\mathbf{M}$	Object geometry model
N	Number of basis filters
N_θ	Number of basis filter orientations
N_f	Number of basis filter scales
$N(\mu, S^{-1})$	Gaussian distribution with mean μ and covariance matrix S
$P(x)$	Polynomial function of x
$p(), \pi()$	Probability distribution
R_θ	2x2 rotation matrix
s	Global scale parameter
$s(t)$	Real-valued signal
S	2x2 Gabor filter shape parameter matrix
$S(\cdot, \cdot)$	Similarity function

$T(\theta)$	Generic transformation with parameter θ
$u(\theta)$	Inner product function of basis filters
$w(t)$	Analytic signal
$w(\theta)$	Steering weights
$\mathbf{x}$	Spatial feature locations
Z	Normalization factor of a distribution or a function
(x, y)	Two-dimensional Cartesian spatial coordinates
(r, θ)	Two-dimensional polar spatial coordinates
β	Likelihood steepness parameter
η	Hilbert transform convolution kernel
θ	Planar rotation angle
$\boldsymbol{\theta}$	Parameter vector
μ	Wave vector
ω	Frequency (one-dimensional frequency coordinate)
(ω_x, ω_y)	Two-dimensional Cartesian spatial-frequency coordinates
$(\omega_r, \omega_\theta)$	Two-dimensional polar spatial-frequency coordinates
ω_0	Center frequency of a filter
σ	Shape parameter of a Gabor filter, standard deviation of the Gaussian envelope function
σ_x, σ_y	Shape parameters of a non-spherical Gabor filter in Cartesian coordinates
ξ	Spatial coordinate vector
ϕ	Elevation rotation angle
ψ	Azimuth rotation angle

Chapter 1

Introduction

1.1 Background

Human beings have an innate ability to interpret visual scenes, locating objects in them, classifying them into different categories and recognizing familiar objects within the categories. The human brain is so efficient and seemingly effortless in its processing of visual information that it is perhaps surprising that human vision is actually an extremely complex phenomenon, and large parts of the brain are devoted for processing of visual information.

One of the aims of computer vision and image analysis is to emulate the visual capabilities of humans, given the assumption that vision is indeed a computational process, however a complex one, performed by neurons in the brain. Indeed, biological vision systems have been successfully used as an inspiration for artificial vision. In the past 25 years, two-dimensional oriented filters with spatially local receptive fields have proved to be highly useful in a wide variety of computational vision tasks, such as estimating the local orientation of a detected line or differentiating between textured regions of an image.

Psychophysical experiments suggest that the early stages of mammalian vision processes are based on similar orientation and frequency specific two-dimensional, approximately linear filters. Deeper structures of the visual cortex are less well known, and provide little information on how recognition of complete objects, for example, is performed in the brain. Despite the successes of computer vision, computers are sorely outperformed by humans in most vision-related tasks, and it is not likely that the situation would change in the near future. Indeed, vision has turned out to be a very difficult and computationally demanding problem.

1.2 Overview

In this work, local image features are described using responses of Gabor filters, which have been proposed as idealized mathematical models for oriented filter structures in the mammalian visual cortex. In computer vision, the responses of Gabor filters are commonly used as feature descriptors, due to their theoretically optimal feature detection properties and good practical recognition results. Because Gabor filters are orientation-sensitive, their responses change as the object rotates. Detection and recognition performance of a vision system based on oriented filters suffers if these effects are not taken into account. The framework of steerable filters (Knutsson et al., 1983),(Freeman and Adelson, 1991) provides the required rotation-invariant representation of oriented filter responses while preserving the orientation information about the gray-level structure of the feature.

Human face recognition is a widely researched problem, with many applications in access control and other security-related fields as well as for example automatic indexing of images. Prominent object matching systems applicable to human faces include Active Appearance Models (AAM) (Cootes et al., 2001) and Elastic Bunch Graph Matching (EBGM) (Wiskott et al., 1999). Both are based on representing the objects as a combination of a shape model and a feature texture model. AAM represents the whole object texture using a low-dimensional model of its main variations. In the EBGM model the object representation is based on a spatially sparse set of local features obtained from Gabor filter responses. Tamminen (2005) formulated the object matching problem in the Bayesian framework, using a local feature based object model similar to the EBGM model, but emplying random sampling from probability distributions derived from the similarity function of the oriented filter responses.

The visual tasks considered in this work include generic visual feature detection, human facial feature matching, face recognition and pose estimation. The approach of the work is based on parts-based object modeling, where objects are represented as constellations of local features. The feature descriptors can be for example local image patches, histograms or image derivatives. Object detection and recognition are sometimes considered two different subproblems, especially in the case of human faces, where face detection is typically the first step in automated face recognition. In the approach employed in this work, detection and recognition are combined in the same framework. The difference between the two is the complexity of the object models. Also pose estimation can be performed in the same framework. This approach can be considered to belong in the category of learning-based methods, in which the changes in local features due to pose changes are learned from a number of example poses.

1.3 Aims of the thesis and author's contributions

The primary aims of this thesis are to extend the probabilistic local feature based object matching model so that the effects of significant in-plane and depth rotations can be taken into account, and to develop the required methods for pose modeling of features and pose-invariant matching. The research problems and author's contributions are summarized briefly in the following.

Gabor-type filters are considered in the framework of steerable filters, and it is shown how Gabor filters can be used as approximately steerable filters. In-plane rotations of the EBGM object model can be then handled using the framework of steerable filters.

Rotation invariance of features has been typically achieved either by using features which are themselves rotation invariant, or by discrete approximations. Using steerability, it is shown how to construct a continuous rotation-invariant similarity measure for oriented filter responses. This formulation allows more accurate measurement of feature similarity compared to the discrete approximations.

In addition to rotation invariance of the feature representation, the design parameters of the filter bank affects the recognition performance of the object matching system. These effects are studied using two image databases, and good design parameters for the filter bank are systematically sought.

A major difficulty in constructing a pose invariant feature based recognition system is how to measure the similarity of features under out-of-plane rotations. A regression modeling approach for modeling the responses of oriented filters under depth rotations is presented. A novel contribution in the work is the use of synthetic data in learning the feature models.

The pose estimation problem is approached using random sampling methods. The focus of the work is in presenting the differences of the random sampling methods.

This thesis is organized as follows. Chapter 2 is introductory in nature and reviews quadrature based signal analysis and different quadrature filters. Various two-dimensional extensions of the Hilbert transform are discussed, and the most common families of quadrature filters in the literature are reviewed.

Chapter 3 concerns the steerability of quadrature filters. Traditionally Gabor filters have not been considered to be steerable, but with the parameterization presented here and using standard methods of analysis and linear algebra, it is shown that their approximate steering performance can be quite good. The main contribution of the chapter is the novel analytic derivation of steering functions for Gabor, DC free near-Gabor and angular Gaussian filters and the analysis of steering error with respect to filter parameters. The idea of exploring similarities between steerable and Gabor filters arose in discussions between the author and Veit Schenk.

Chapter 4 presents the probabilistic approach to image analysis applied in the work, starting from the similarity function between filter jets and ending in the joint probability model of a complete object. Additionally, the sampling algorithms applied in the thesis are briefly presented. The main novel contribution of Chapter 4 by the author is the application of steerability in the formulation of rotation invariant similarity functions. The probabilistic formulation of the similarity function leading to the object probability model is due to Prof. Jouko Lampinen.

Chapter 5 deals with the effect of the filter shape parameters to recognition performance in an object matching system. The chapter begins with examples showing the effects of steerable approximations and undersampled filter banks to the filter responses and similarity function values and then continues to find good parameters for the filter bank for simultaneous localization and recognition. The main contribution of Chapter 5 is the numerical analysis of filter shape parameters on recognition performance. The results presented in the chapter apply the object matching system developed by Prof. Jouko Lampinen, Toni Tamminen, Timo Kostiainen, and the author, with most of the program code written by Toni Tamminen. The experiments and their analysis have been performed by the author.

Chapter 6 presents a novel regression modeling approach for the pose variation in the filter responses. In the literature the problem has been typically addressed as a manifold learning problem, but by using synthetic data it is possible to apply simpler regression modeling methods. Two models, a piecewise linear and a mixture of Gaussian model are considered. The main contribution of Chapter 6 is the use of a synthetic model in generating a direct regression model of the features. The idea of using synthetic models and applying regression modeling was suggested by Professor Lampinen, while the implementation is the author's own.

Chapter 7 collects the presented methods into complete human face matching systems which are able to locate faces in all orientations, serving as a basis for person identification. Metropolis, Gibbs and Population Monte Carlo samplers are compared in a setting with one rotation angle, and the PMC sampler is extended to handle three rotation angles in addition to scale and displacement parameters. The main contribution of the chapter is the construction of the rotation invariant recognition system and the application of random sampling algorithms, especially the Population Monte Carlo algorithm, to the pose estimation problem.

The program codes for the face matching systems are derived from the work of Toni Tamminen, and the ideas for various samplers originate from discussions between Professor Lampinen, Aki Vehtari and the author.

Chapter 8 concludes the work.

Chapter 2

Signal analysis with quadrature filters

2.1 Introduction

We will begin building our object recognition system from the ground up and first consider the image processing operations which transform the input image into a representation which is easier to analyze. This can be considered a feature extraction stage. Instead of traditional and well-established optimized edge and corner feature detectors (Harris and Stephens, 1988) (Canny, 1986), responses of linear filter banks will be used as feature descriptors. This approach has the advantage that the features we can use are not limited to edges or corner points, but can be any local gray-level structures in the image.

In the first three sections we will review the mathematical background of quadrature filters which were proposed already by Granlund (1978) as the generic image processing operation for low-level vision tasks. We will relate the filter bank approach to the theory of wavelets in section 2.5, and conclude by presenting in section 2.6 some of the oriented quadrature filter families which will be employed later on in the work.

2.2 Magnitude, phase and the analytic signal

The Hilbert transform (Oppenheim et al., 1999) of a real-valued function $s(t)$ is an integral transform

$$\mathcal{H}[s(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau, \tag{2.1}$$

where the improper integral is considered as a Cauchy principal value, which is necessary due the singularity at $t = \tau$. The Hilbert transform is thus a convolution

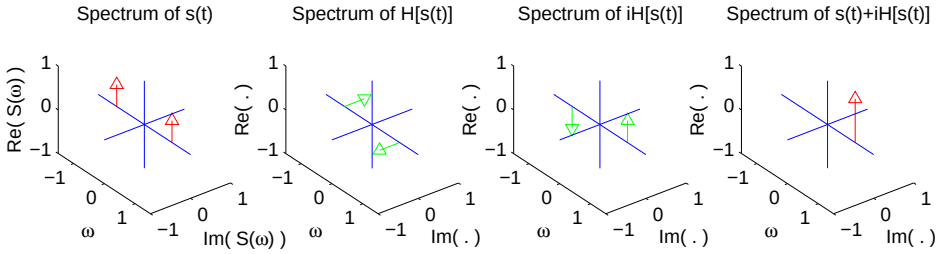


Figure 2.1: From left to right: The spectra of a cosine signal $s(t)$, its Hilbert transform, the Hilbert transform multiplied by the imaginary unit, and the analytic signal resulting from the sum of the first and third signals.

integral $\mathcal{H}[s(t)] = (\eta * s)(t)$ with the convolution kernel $\eta(t) = \frac{1}{\pi t}$. The Fourier transform of $\eta(t)$ is

$$H(\omega) = \mathcal{F}[\eta(t)](\omega) = -j \cdot \frac{\omega}{|\omega|} = -j \cdot \operatorname{sgn}(\omega). \tag{2.2}$$

We can interpret this to mean that in the frequency domain the Hilbert transform rotates the positive frequency components of a signal in the complex plane by $-\pi/2$ and negative frequency components by $\pi/2$.

The *analytic signal* (Smith, 2003) of a real-valued time-domain signal $s(t)$ is a complex-valued extension of the original signal defined by

$$w(t) = s(t) + i\mathcal{H}[s(t)]. \tag{2.3}$$

As a result, we obtain a signal $w(t)$ with the same unrotated positive frequency components as the original signal $s(t)$ (multiplied by two), and whose negative (mirror) frequencies have been eliminated completely. Figure 2.1 illustrates the generation of the analytic signal of a single cosine signal.

The name analytic signal stems from the fact that since its Fourier transform $\mathcal{F}[w(t)] = W(\omega)$ is one-sided, the corresponding Z-transform $W(z)$ does not have poles inside the unit circle and is thus analytic there, in the terminology of mathematical complex analysis.

In the Hilbert transform the real signal $s(t)$ is divided into two parts, instantaneous magnitude $|w(t)|$ and instantaneous phase $\arg(w(t))$. These signals are formally given by

$$|w(t)| = \sqrt{s(t)^2 + (\mathcal{H}[s(t)])^2} \tag{2.4}$$

and

$$\arg(w(t)) = \arctan\left(\mathcal{H}[s(t)]/s(t)\right). \tag{2.5}$$

Qualitatively speaking, in some loose sense the magnitude tells *where* something interesting is happening, and the phase describes *what* is happening there. From an engineering point of view, the usefulness of the Hilbert transform and the analytic signal lies in the fact that they can be used to compute useful estimates of the signal. The magnitude of the analytic signal in particular is a very good envelope estimator for narrow-band signals regardless of their center frequency.

Figure 2.2 shows a test signal which consists of a Gaussian wave packet with non-stationary frequency, a triangle wave and a single pulse. The magnitude of the analytic signal tracks the wave packet very well. The small ripple is in fact caused by the other signals, and becomes evident because the Hilbert transform is a global operation. The peaks of the magnitude locate the edges in the signal. At these points, the instantaneous phase tells the type of the edge. The peaks of the triangle wave have even symmetric phase (0 or $\pm\pi$), whereas the edges of the pulse have odd symmetric phase ($-\pi/2$ or $\pi/2$). The derivative of the instantaneous phase is related to the local frequency of the signal.

Oppenheim and Lim (1981) show that much of the perceptual information in a signal is carried in its phase. They also demonstrate how the amplitude can be estimated solely from the phase in global Fourier synthesis. The latter result is less surprising than it perhaps first seems, because in global Fourier analysis the basis functions (complex exponentials) are spatially unlocalized, and thus the magnitudes, which should contain information about where things are in the signal, are also unlocalized. Nevertheless, the examples show that phase information is both information-theoretically and perceptually very important.

2.3 Quadrature filters

The process of computing the analytic signal can be applied to any signals, and thus also to filters. The output of a filter pair with the impulse responses $h(t)$ and $\mathcal{H}\,[h(t)]$ is equivalent to filtering the complex-valued analytic signal with a single filter. We can identify outputs of the two filters with the real and imaginary parts of the analytic signal, and construct a complex-valued filter

$$h'(t) = h(t) + i\mathcal{H}\,[h(t)]\,. \tag{2.6}$$

Such a filter (or a pair of real-valued filters) is said to be *in quadrature* (Gabor, 1946). The underlying idea is to restrict analysis into some interesting parts of the original signal instead of computing the analytic signal which is necessarily a global process and includes all information present in the original signal.

Quadrature filters should not to be confused with the quadrature mirror filters (QMFs), which are real-valued filter pairs with a specific alias cancellation property so that the original signal can be reconstructed perfectly from the decimated and aliased subband signals (Fliege, 1993).

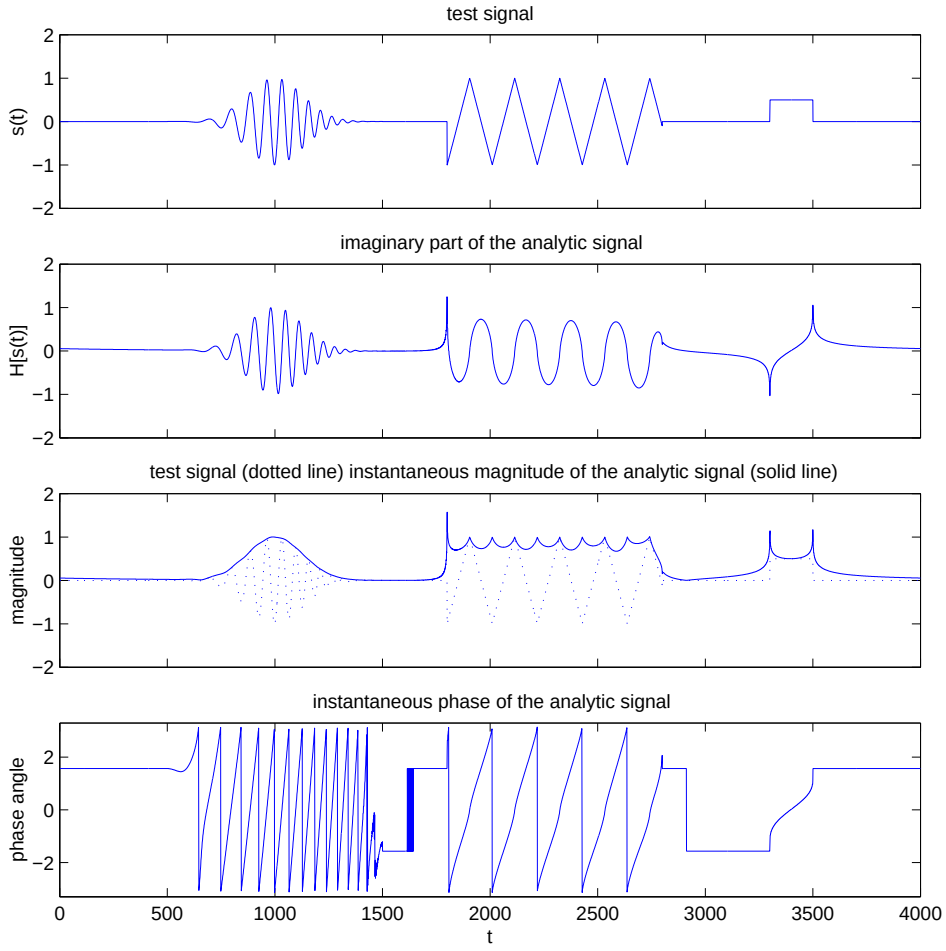


Figure 2.2: A test signal, its Hilbert transform, the instantaneous amplitude of the analytic signal and the instantaneous phase of the analytic signal. The amplitude of the analytic signal tracks the envelope of the original signal, plotted with dotted line.

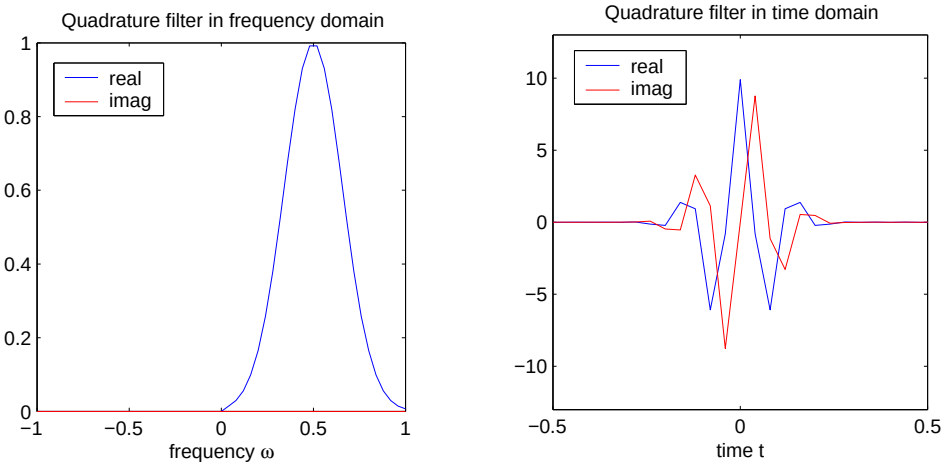


Figure 2.3: A simple quadrature filter in frequency and spatial domains. Note how the real and imaginary parts oscillate in the spatial domain with a phase difference of $\pi/2$.

A straightforward way to design a quadrature pair of filters is to choose a desired frequency response, ensure that it has no negative frequencies and compute its inverse Fourier transform. The resulting complex signal has the impulse responses of the filters $h(t)$ and $\mathcal{H}[h(t)]$ in its real and imaginary parts. Figure 2.3 shows an example bandpass design with a Gaussian frequency response. For reasons of convention and convenience, one can start with a purely real frequency response, which gives a time domain impulse response whose real and imaginary parts are even and odd symmetric, respectively, about the origin. The resulting filter is thus non-causal. For real-time systems this deficiency can be remedied by simply adding a suitable amount of delay. Because the forward and inverse Fourier transforms differ from each other by a single sign in the exponent, it follows that a causal filter has a frequency response in which the real and imaginary parts are also a Hilbert pair. A causal Hilbert pair of time domain filters is consequently a Hilbert pair also in the frequency domain.

2.4 Two-dimensional versions of the Hilbert transform

The Hilbert transform is defined only for one-dimensional signals. In order to construct two-dimensional quadrature filters, a 2D version of the Hilbert transform is needed. To accomplish this, we need to define an analogy for negative frequencies in two dimensions.

One possibility is to choose a preference direction $\vec{n}$ in the frequency domain $\vec{u} = (u_1, u_2)^T$, and deem the frequencies with $\langle \vec{u}, \vec{n} \rangle > 0$ positive, giving the

transfer function

$$H_P(\vec{u}) = j \cdot \mathrm{sgn}(\langle \vec{u}, \vec{n} \rangle). \tag{2.7}$$

This is called the *partial Hilbert transform* (Bulow, 1999). The main problem with this definition is that the choice of preference direction $\vec{n}$ is arbitrary and since the transform is not isotropic with respect to rotations, we get different results with different choices of $\vec{n}$. It is however useful for signals which vary only in a particular direction. Bulow (1999) also discusses the *total Hilbert transform* (which has the transfer function $\mathcal{H}_T = -j \cdot \mathrm{sgn}(u_1)\mathrm{sgn}(u_2)$) and combinations of partial and total transforms, but these cannot be considered valid generalizations of the one-dimensional transform, since they do not perform a phase shift of $\pi/2$ in any meaningful one-dimensional domain (Felsberg and Sommer, 2001).

Another approach is to consider the frequency domain in polar coordinates, since we would like the transform to be equivariant with respect to rotation. The two frequency coordinates are then the angular frequency $f_\phi \in [-\pi, \pi]$ which is cyclic and related to orientation, and the radial frequency $f_r \in [0, \infty]$ which is related to scale. The radial Hilbert transform (Davis et al., 2000) has the transfer function

$$H_r(r, \theta) = \exp(j\theta) \tag{2.8}$$

in polar coordinates (r, θ). It has the property that all lines passing through the origin are equivalents of one-dimensional Hilbert transforms in the sense that the two halves of the line on opposing sides of the origin have a phase difference of π. The problem with this approach is that each line uses a different transform, and they cannot be readily combined with the original signal in order to construct a two-dimensional analytic signal.

In computer vision literature the problem has been traditionally addressed in a manner which has common ground with both partial and radial Hilbert transforms. In the polar parameterization there are no negative radial frequencies by definition, so intuition suggests that the Hilbert transform must be done with respect to the angular frequency. Knutsson and Granlund (1983) already designed Hilbert pairs of bandpass filters using this approach. As long as the angular component of the bandpass filter is symmetric with respect to certain angular frequency f_θ and zero at $f_\theta + \pi/2$, a one-dimensional Hilbert transform in the angular direction is equivalent to a two-dimensional partial Hilbert transform with the preference direction f_θ, and the lines passing through the origin are also one-dimensional Hilbert transforms in the same sense as in the radial Hilbert transform.

The existence of the differently defined two-dimensional Hilbert transforms suggests that there is something unsatisfactory in all of the previous approaches. Indeed, while they are useful generalizations of the one-dimensional case for certain narrow-band signals, they cannot be used to compute a two-dimensional version of the analytic signal of arbitrary two-dimensional signals. A mathematically more elegant extension of the analytic signal into two dimensions has

been proposed by Felsberg and Sommer (2001), named the *monogenic signal*. It consists of a single magnitude and two phase components, one of which is related to local geometric (orientation) information and the other to local structural (phase) information. A side effect of this additional information is that the algebra of complex numbers is not sufficient to embed three quantities of information into a single point in the two-dimensional signal and we need a quaternionic signal. Let us denote the base elements of the quaternion with $\{1, i, j, k\}$. By embedding of the three-dimensional signal into the three first elements of the quaternionic algebra, the monogenic signal has the transfer function

$$H_M(\vec{u}) = \frac{|\vec{u}| + (1, k)\vec{u}}{|\vec{u}|}, \tag{2.9}$$

where $\vec{u} = (u_1, u_2)^T$ is a frequency vector with two components. This is in spatial domain equivalent to

$$f_M(\vec{x}) = f(\vec{x}) - (i, j)\, f_R(\vec{x}), \tag{2.10}$$

where $f_R(\vec{x})$ is the Riesz transform of $f(\vec{x})$, i.e. in frequency domain the two are related by $F_R(\vec{u}) = i\frac{\vec{u}}{|\vec{u}|}F(\vec{u})$. The monogenic signal shares many of the properties of the analytic signal, but it is not one-sided. Felsberg and Sommer (2001) note that this property is irrelevant for image recognition, because images are real-valued and their spectra are therefore symmetric.

Despite the theoretical superiority of the monogenic signal compared to two-dimensional extensions of the analytic signal, it has not yet been applied widely in computer vision applications. Due to this, only the analytic signal and quadrature filter bank based approach is considered in the rest of the work. While quadrature filters cannot be exactly isotropic, the error in amplitude and phase responses is often small enough to be negligible in practice, compared to other error sources.

2.5 Wavelets and filter banks

Wavelet analysis (Daubechies, 1990) is in some sense a generalization of Fourier analysis, and a formal refinement of short-time Fourier analysis, in which the aim is to describe simultaneously both time and frequency behavior of a signal. Mathematically, the continuous wavelet transform is a convolution integral between the signal $f(x)$ and the wavelet kernel $g_\omega(x)$,

$$\hat{f}(x, \omega) = f(x) * g_\omega(x) = \int_{-\infty}^{\infty} f(\xi) g_\omega(x - \xi) d\xi. \tag{2.11}$$

Since the wavelet transform of a one-dimensional signal is essentially a change from one-dimensional representation into two dimensions, there is redundancy in

the representation which can be reduced or eliminated completely by sampling (or subsampling) the continuous translation-scale space (x, ω), leading to the *discrete wavelet transform*, where we have only a discrete set of wavelets $g_{\omega^{(i)}}(x)$, and a discrete set of lattice points $x^{(i)}$.

In *dyadic sampling* the time-frequency space is covered so that each discrete wavelet has equal area and successive scales are related by a factor of two. This results in having a higher spatial sampling density at higher frequencies. The sampling is said to be *critical* if a minimum number of samples is used to represent the original data perfectly. Oversampling refers to the case when some redundancy is retained in the sampled representation, and undersampling to the case when the original signal is not represented completely by the samples. In the two-dimensional case we have even more freedom in the tessellation of the four-dimensional phase space, as the continuous wavelet transformation of a two-dimensional function $f(x, y)$ with the wavelet $g_{\omega_r,\omega_s}(x, y)$ is a convolution integral

$$\begin{aligned} \hat{f}_{\omega_r,\omega_s}(x, y) &= f(x, y) * g_{\omega_r,\omega_s}(x, y) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi, \upsilon) g_{\omega_r,\omega_s}(x - \xi, y - \upsilon) d\xi d\upsilon. \end{aligned} \quad (2.12)$$

Granlund (1978) was among the first to suggest that low-level computer vision should be based on a generic, parallel, and hierarchical convolution operator. Such banks of filters can be viewed as implementations of discrete wavelet transforms. In wavelet analysis the idea of a single universal operator is encapsulated in the concept of a *mother wavelet* which is translated and scaled in order to produce individual wavelets. In order to compute a wavelet transform, we choose some set of points $(x^{(i)}, y^{(i)}, \omega_r^{(i)}, \omega_s^{(i)})$ which gives us a finite number of wavelet kernels (or filters) whose responses are evaluated at a finite number of spatial locations.

For the spatial coordinates, a natural choice is to sample the spatial coordinates evenly in Cartesian coordinates. The sampling of spatial-frequency (or orientation and scale) coordinates is less self-evident. Regardless of the choice of the filters, a log-polar-type division of the frequency plane is a popular choice (Knutsson and Granlund, 1983), (Daugman, 1988), leading to a "daisy petal" arrangement of the filters (Bovik et al., 1990), where the filters in a single scale are rotated copies of each other, and successive scales are spaced logarithmically, each frequency scale possessing an equal number of filters. In other words, we use the dyadic sampling idea for the radial frequency coordinate, but uniform sampling for the orientation frequency coordinate. Another possibility would be to use Cartesian coordinates also for sampling the frequency plane. The log-polar sampling idea lends itself better to handling rotations, a phenomenon which does not exist in one dimension, since rotations correspond to single-parameter cyclic shifts in the log-polar coordinate system.

In image coding and compression applications the subbands are typically decimated because their representation is superfluous in the sense that the filter responses contain information only in a narrow part of the full bandwidth. The decimation causes aliasing, but suitably narrow bandpass filters will preserve information so that perfect reconstruction from the subsampled, aliased signals is possible under certain conditions. Without decimation, the division of the signal into subbands produces an expansion of data, a situation hardly beneficial in general for compression.

In analysis applications, apart from practical computational and memory storage requirements, there is no need to decimate the subband signals, because the filter response values need to be known at every pixel location. In fact decimation should be avoided when possible. Only ideal "brick-wall" filters eliminate aliasing altogether, and such filters necessarily produce prominent ringing (Gibbs phenomenon) in the spatial domain (Simoncelli and Adelson, 1990).

However, the dangers of aliasing are still present in the spatial domain even when we do not subsample the subbands. Care should be taken when designing discrete filter banks in order to make sure that the spatial extent of the filters is not too large. It is possible to compute bounds for the largest possible filter which can be contained in a given discrete lattice, but in practice spurious boundary effects become a problem much earlier. It is not possible to compute the filter responses correctly near the boundaries of the image simply because the spatial extent of the filters overlaps the image boundary and the values of the signal are not known outside the image boundary. Filters at low frequencies have the largest spatial extent, and thus their responses become unreliable even when the higher frequencies could be still computed accurately.

2.6 Oriented filter families

Let us review briefly some of the main types of oriented quadrature filters proposed in the literature. A good review of the properties of different one-dimensio-nal band-pass quadrature filters can be found in Boukerroui et al. (2004).

Strongly influenced by the mathematical formalism of quantum mechanics, Gabor (1946) derived the one-dimensional bandpass filter minimizing the joint uncertainty in time and frequency domains. As a measure of the uncertainty of a complex-valued function ψ, Gabor used the normalized root-mean-square bandwidth

$$\Delta\omega = \sqrt{\frac{\int_{-\infty}^{\infty}(\omega-\omega_0)^2\psi(\omega)\psi^*(\omega)d\omega}{\int_{-\infty}^{\infty}\psi(\omega)\psi*(\omega)d\omega}} \tag{2.13}$$

where

$$\omega_0 = \frac{\int_{-\infty}^{\infty} \omega \psi(\omega) \psi^*(\omega) d\omega}{\int_{-\infty}^{\infty} \psi(\omega) \psi * (\omega) d\omega} \tag{2.14}$$

is the center frequency of the function (the mean of the Gaussian distribution). Similarly one can define Δx as the effective width in the time domain. The Heisenberg uncertainty principle then states that since ω and x are conjugate variables, the product of the effective widths obeys the inequality

$$\Delta\omega \Delta x \geq \frac{1}{4\pi}. \tag{2.15}$$

The term "uncertainty" refers to the fact that in quantum mechanics, the product $\psi\psi^* = |\psi|^2$ is interpreted as the probability density of the quantity associated with the wave function ψ, and only the probabilities $|\psi|^2$ can be observed. In signal processing applications we deal directly with the complex-valued signals, and the uncertainty principle can be considered merely a mathematical property shared by the signal and its Fourier transform.

The function family which meets the lower bound of the uncertainty product is the complex exponential

$$g(x; \sigma, \omega_0) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp\left(i\omega_0 x\right) \tag{2.16}$$

where ω_0 and σ free parameters. Daugman (1985) generalized the argument into two dimensions and derived 2D Gabor filters which achieve the lower limit of joint uncertainty in spatial and frequency domains, given by

$$g(x, y; \sigma_x, \sigma_y, \omega_x, \omega_y) = \exp\left(-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)\right) \exp\left(i\left(\omega_x x + \omega_y y\right)\right) \tag{2.17}$$

in the spatial domain. These functions are equivalent to the canonical coherent states generated by the Weyl-Heisenberg group in quantum mechanics (Daubechies, 1990) (Lee, 1996). In the sense of the uncertainty principle the 2D Gabor filter then has some optimality properties for pattern recognition. There is also a strong body of psychophysical evidence supporting the hypothesis that mechanisms employing oriented linear filters are involved in mammalian vision, and they are well approximated with 2D Gabor filters (Daugman, 1988). While 2D Gabor filters are nonorthogonal, they can form a relatively good approximation of a tight wavelet frame and approximate reconstructions using direct summation as well as iterative methods are possible (Lee, 1996). Gabor filters have been used a wide variety in tasks requiring oriented filters. They have been especially popular in texture analysis and segmentation (e.g., (Dunn and Higgins, 1995),(S.E. Grigorescu and Kruizinga, 2002)) and face recognition

applications (e.g., (Wiskott et al., 1999), (Krüger, 2001). See Shen and Bai (2006) for a review). In practical applications, a modified form of the 2D Gabor filter (Ronse, 1993; Lades et al., 1993) is often used, with the transfer function

$$g(x, y; \sigma_x, \sigma_y, \omega_x, \omega_y) = \\ \exp\left(-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)\right)\left(\exp\left(i\left(\omega_x x + \omega_y y\right)\right) - \exp\left(-\frac{\sigma_x^2}{2}\right)\right). \quad (2.18)$$

The additional term subtracts the DC component in the real (symmetric) part of the filter. It should be noted that this modified filter does not strictly minimize the uncertainty product, and only approximates a 2D Gabor filter. 2D Gabor-type filters will be discussed more thoroughly in Section 3.4.1.

Nonetheless, 2D Gabor filters are only optimal in terms of uncertainty in the Cartesian coordinates. Polar coordinate representations may be considered to be perceptually more meaningful and have been proposed as more efficient in coding of natural images (Field, 1987). Defined in the frequency domain, polar Gabor filters (Haley and Manjunath, 1995), (Ro et al., 2001) with the transfer function

$$G(\omega_r, \omega_\theta; \omega_0, \sigma_r, \sigma_\theta) = \exp\left(-\frac{(\omega_r - \omega_0)^2}{2\sigma_r}\right)\exp\left(-\frac{\omega_\theta^2}{2\sigma_\theta}\right) \quad (2.19)$$

and log-Gabor filters (Field, 1987), (Kovesi, 1999), with the transfer function

$$G(\omega_r, \omega_\theta; \omega_0, \sigma_r, \sigma_\theta) = \exp\left(-\frac{\log_2(\omega/\omega_0)}{2\log_2(\sigma_r)}\right)\exp\left(-\frac{\omega_\theta^2}{2\sigma_\theta}\right) \quad (2.20)$$

arise as natural modifications of the Gabor filter for polar and log-polar frequency coordinates, respectively, but do not achieve minimum uncertainty in the spatial domain. A theoretical drawback of polar and log-Gabor filters is the absence of a closed-form expression for the filter in spatial domain. Also analytic derivation of minimum uncertainty filters for polar and log-polar frequency coordinate systems appears difficult.

There are also several filter types which have qualitatively similar shape as Gabor filters, although they have been derived from premises other than minimizing the uncertainty product. One such filter is the Gaussian derivative filter, which belongs to the larger group of *steerable filters*. Steerability, which will be discussed more thoroughly in Section 3, poses a constraint for the orientation bandwidth of the filter so that a rotated copy of the filter can be computed as a linear combination of the original filter bank (Freeman and Adelson, 1991). Steerable filters have found applications in many different tasks in computer vision, including adaptive filtering (Knutsson et al., 1983), (Freeman and Adelson, 1991), (Simoncelli et al., 1992), (Perona, 1995), (Simoncelli and

Farid, 1995), motion estimation (Fleet and Jepson, 1990), stereo vision (Fleet et al., 1991), shape from shading (Freeman and Adelson, 1991), texture analysis (Knutsson and Granlund, 1983), (Greenspan et al., 1994) and feature detection (Jacob and Unser, 2004), (Yokono and Poggio, 2004a).

Exactly steerable filters of the form $P(x)G(\sqrt{x^2+y^2})$, where $P(x)$ is a polynomial, such as the examples considered in (Freeman and Adelson, 1991) are quite inflexible, because orientation and radial frequency selectivities of such filters cannot be easily chosen independently. Polar-separable oriented filters were already proposed in (Knutsson and Granlund, 1983), who also used a special case of steerability for computing the principal local orientation. The feature localization performance, relating to the uncertainty principle, of exactly steerable filters is not ideal because steerability, like orthogonality, is a rather strict constraint on the filters. In practical applications, approximately steerable filters can often be used instead, if the required accuracy of orientation estimates is not very high, or the orientation estimation is only descriptive by nature. In addition, noise levels in natural images are often large enough to make the systematic steering approximation error relatively insignificant in comparison.

Steerable filters with wedge-shaped responses in the spatial domain have been developed for edge classification (Simoncelli and Farid, 1995), (Yu et al., 2001). Although influenced by the quadrature filter methodology, Yu et al. (2001) propose an approach which is a departure from quadrature based signal analysis, as it is based on the amplitude response and its derivative. The main appeal is that there is no forced symmetry in the responses of the filters and the amplitude response of the filter bank can be directly interpreted as an orientation signature. This is possible because the filters are tessellated around the origin both in spatial and frequency domains. However, this arrangement does not appear to lead to any obvious advantages in applications which use both amplitude and phase responses of the filters. The ambiguity caused by the symmetry of the amplitude response is resolved by the phase response in regular quadrature based analysis. Wedge filters cannot use phase information in the same sense as other oriented quadrature filters presented above, because the wedge filters at different orientations do not share the same spatial support, and there is thus no clear definition of "local phase".

Chapter 3

Steerability properties of Gabor-type filters

3.1 Introduction

In this chapter the steerability framework is extended to include Gabor filters, the related DC free Gabor filters and angular Gaussian filters. Novel analytical derivations of the required inner product functions are given for these three filter types.

Gabor filters have been considered by some to be "not steerable" (e.g. (Shi, 1999), (Greenberg et al., 2002)), but in this chapter it is shown that their steering error performance can be quite good with suitable filter shape parameters, and the error performance in the same order of magnitude with approximately steerable filters presented in the literature.

Section 3.2 reviews the theory of steerability. Section 3.3 presents the error metric which is used to evaluate steering performance. The required inner product functions for the different filter types are derived, and the steering performance is analyzed in Section 3.4. Section 3.5 discusses the accuracy of the approximations in computing the analytical inner product functions and the compares the steering properties of Gabor and angular Gaussian filters. Finally in Section 3.6 the presented direct steering method is compared to an alternative approach proposed by Teo and Hel-Or (1999), where Gabor filters are approximated by a set of exactly steerable basis functions.

Parts of the work in this chapter have been published in (Kalliomäki and Lampinen, 2005) and (Kalliomäki and Lampinen, 2007).

3.2 Steerability and shiftability

An oriented filter bank computes the response of the filters in some discrete orientations, and it would often be useful to be able to know what the response would be somewhere between the orientations. Subject to certain conditions, it is possible to adaptively "steer" the filters into arbitrary orientations by computing the linear sum

$$f^{\theta}(x) = \sum_{i=1}^{N} k_i(\theta)\, f^{\theta_i}(x), \tag{3.1}$$

where $f^{\theta_i}(x)$ are the original filters of the filter bank, also called *basis filters*, and $f^{\theta}(x)$ is the interpolated filter in a new orientation θ. The steering coefficients k_i depend only on θ and not x, thus also allowing computations performed with the linear filters to be interpolated using the same linear weights.

A simple example of a shiftable function is $\cos(\theta)$. It is exactly shiftable with two shifted copies of itself, namely

$$\cos(\theta - \hat{\theta}) = \cos(\hat{\theta})\cos(\theta) + \cos(\hat{\theta} - \pi/2)\cos(\theta - \pi/2), \tag{3.2}$$

when $\hat{\theta}$ is the amount of (phase) shift. The previous equation is equivalent to the well-known result that a cosine wave in arbitrary phase can be represented as a weighted sum of a cosine and a sine wave,

$$\begin{aligned} \cos(\theta - \hat{\theta}) &= k_1(\hat{\theta})\cos(\theta) + k_2(\hat{\theta})\cos(\theta - \pi/2) \\ &= k_1(\hat{\theta})\cos(\theta) + k_2(\hat{\theta})\sin(\theta), \end{aligned} \tag{3.3}$$

where the weights k_i again depend only on the amount of phase shift $\hat{\theta}$, not on the function parameter θ.

Steerability was proposed by Freeman and Adelson (1991) for the special case of rotation. Simoncelli et al. (1992) extended the same framework to include translation and scaling, and coined the term "shiftability". Perona (1995) proposed the term "deformable" to include interpolation capability of arbitrary transformations. Teo and Hel-Or (1999) use the term "shiftable" to include any Lie transformation groups. The function f is shiftable if any transformation $T(\theta)$ acting on f can be expressed as a linear combination of a fixed, finite set of basis functions f_i,

$$T(\theta)\, f(x) = \sum_{i=1}^{N} k_i(\theta)\, f^{i}(x). \tag{3.4}$$

In feature detection applications our main interest is in orientation steerability. The features one wishes to detect are typically lines, edges and junctions, which are locally almost independent of scale, that is, their orientation frequency response is very similar at all scales. This does not mean that the features are

intrinsically one-dimensional (simple lines or edges). For example the intersection of two lines has a two-dimensional grey-level structure, but its orientation response is highly similar at all scale levels. Thus orientation is usually more descriptive than scale, and the ability to interpolate orientation responses is more important than the ability to interpolate responses from one scale to another.

When designing the filter bank we must choose how many orientations we are going to have in the bank. Steerability approaches the same question from a different direction: how many basis filters do we need in general for the steering of a given angular component? Freeman and Adelson (1991) proved that the minimum number of shifted copies needed for fulfilling the steerability condition exactly is equal to the number of non-zero coefficients (positive and negative frequencies) in the Fourier expansion of the signal. Note that these do not have to be the M first coefficients of the Fourier expansion. Thus, for example, the cosine function requires two basis functions (a cosine and a sine at the same frequency as the original cosine function). The cosine function is however not very useful as an angular component of a steerable filter since it has a very wide orientation bandwidth. Filters with narrow orientation bandwidth are preferable in feature detection, since their feature representation capability is better, but they also need more basis filters in order to be steerable.

Perona (1995) proposed a Singular Value Decomposition based method for finding the optimal basis filters for a given transformation. Computing the SVD of a matrix of transformed versions of the filter, the optimal basis functions are the first N left singular vectors corresponding to the largest singular values. Alternatively, using the theory of Lie groups, a steerable basis can be found for arbitrary parameter groups by representing or approximating the filters in an equivariant function space (Michaelis and Sommer, 1995), (Hel-Or and Teo, 1998). For single-parameter 2D rotation expressed in polar coordinates (r, θ), this function space is $\{f(r)\exp(in\theta)\}, n \in Z$, i.e. complex harmonics together with an arbitrary (real-valued) radial component $f(r)$ (Teo, 1998). A viable approach for filter design is to start with an ideal filter prototype (for example, a Gabor filter) and approximate it in the appropriate equivariant function space (Teo and Hel-Or, 1999), which is guaranteed to be closed under the same transformational group.

In the one-dimensional case symmetry considerations can be used to justify the choice of basis functions which are evenly spaced shifted copies of a single function. In multidimensional parameter spaces the basis functions are not necessarily evenly spaced nor transformed copies of each other, and finding a parsimonious basis function set can be a demanding task. Teo and Hel-Or (1999) propose a method for finding basis function sets with optimal approximation properties for arbitrary multi-parameter transformations.

3.2.1 Least-squares steerability of filters

Freeman and Adelson (1991) originally derived the conditions for exact steerability by considering the Fourier series of the angular component of the filter in polar coordinates. An alternative approach is to use linear algebra to find the optimal linear steering functions for an arbitrary set of filters (Greenspan et al., 1994), which is briefly reviewed here.

Orientation steerability of a (real-valued) linear filter g means that arbitrary filter orientations can be computed (or at least approximated) by computing the sum of a set of basis filters $\mathbf{g} = \{g_{\theta_1}, g_{\theta_2}, \ldots, g_{\theta_N}\}$ weighted with steering coefficients $\mathbf{k} = \{k_1(\theta), k_2(\theta), \ldots, k_N(\theta)\}$,

$$g(\theta) \approx \sum_{j=1}^{N} k_j(\theta) g_{\theta_j} = \mathbf{k}^T \mathbf{g}. \tag{3.5}$$

In the case of complex-valued filters, we need separate real-valued steering coefficients $k_j(\theta)$ for the real and imaginary parts of the Gabor filter and assume that its basis filters g_{θ_j} share the same shape parameters S and frequency μ.

Let us define the inner product between normalized real-valued functions u and v as $\langle u, v \rangle = \int_{\omega \in R^2} u(\omega) v(\omega)\, d\omega$. The functions u and v are normalized without loss of generality so that $\langle u, u \rangle = \langle v, v \rangle = 1$. The optimal steering coefficients k can be solved analytically by minimizing the L2 norm of the error $e = g(\theta) - \mathbf{k}^T \mathbf{g}$,

$$\begin{aligned} \arg\min_{\mathbf{k}} ||e||^2 &= \arg\min_{\mathbf{k}} \left\langle g(\theta) - \mathbf{k}^T \mathbf{g},\, g(\theta) - \mathbf{k}^T \mathbf{g} \right\rangle \\ &= \arg\min_{\mathbf{k}} \langle g(\theta), g(\theta) \rangle - 2 \left\langle g(\theta), \mathbf{k}^T \mathbf{g} \right\rangle + \left\langle \mathbf{k}^T \mathbf{g}, \mathbf{k}^T \mathbf{g} \right\rangle \end{aligned} \tag{3.6}$$

The minimum of this expression is obtained by differentiating it with respect to $\mathbf{k}$ and setting the result to zero, leading to the matrix equation

$$\mathbf{G}\mathbf{k} = \gamma \tag{3.7}$$

where the matrix $\mathbf{G}$ and vector γ have the elements $\mathbf{G}_{i,j} = \left\langle g_{\theta_i}, g_{\theta_j} \right\rangle$ and $\gamma_i = \left\langle g(\theta), g_{\theta_i} \right\rangle$, respectively. In component form, Eq. (3.7) is written out as

$$\begin{bmatrix} \left\langle g(\theta), g_{\theta_1} \right\rangle \\ \left\langle g(\theta), g_{\theta_2} \right\rangle \\ \vdots \\ \left\langle g(\theta), g_{\theta_N} \right\rangle \end{bmatrix} = \begin{bmatrix} \left\langle g_{\theta_1}, g_{\theta_1} \right\rangle & \left\langle g_{\theta_1}, g_{\theta_2} \right\rangle & \cdots & \left\langle g_{\theta_1}, g_{\theta_N} \right\rangle \\ \left\langle g_{\theta_2}, g_{\theta_1} \right\rangle & \left\langle g_{\theta_2}, g_{\theta_2} \right\rangle & \cdots & \left\langle g_{\theta_2}, g_{\theta_N} \right\rangle \\ \vdots & & & \vdots \\ \left\langle g_{\theta_N}, g_{\theta_1} \right\rangle & \left\langle g_{\theta_N}, g_{\theta_2} \right\rangle & \cdots & \left\langle g_{\theta_N}, g_{\theta_N} \right\rangle \end{bmatrix} \mathbf{k}(\theta) \tag{3.8}$$

which holds for all θ and can be used to solve the optimal vector $\hat{\mathbf{k}}(\theta)$ via matrix

inversion,

$$\hat{\mathbf{k}}(\theta) = \begin{bmatrix} \langle g_{\theta_1}, g_{\theta_1} \rangle & \langle g_{\theta_1}, g_{\theta_2} \rangle & \dots & \langle g_{\theta_1}, g_{\theta_N} \rangle \\ \langle g_{\theta_2}, g_{\theta_1} \rangle & \langle g_{\theta_2}, g_{\theta_2} \rangle & \dots & \langle g_{\theta_2}, g_{\theta_N} \rangle \\ \vdots & & & \vdots \\ \langle g_{\theta_N}, g_{\theta_1} \rangle & \langle g_{\theta_N}, g_{\theta_2} \rangle & \dots & \langle g_{\theta_N}, g_{\theta_N} \rangle \end{bmatrix}^{-1} \begin{bmatrix} \langle g(\theta), g_{\theta_1} \rangle \\ \langle g(\theta), g_{\theta_2} \rangle \\ \vdots \\ \langle g(\theta), g_{\theta_N} \rangle \end{bmatrix}. \quad (3.9)$$

Unlike previous approaches ((Greenspan et al., 1994), (Sommer et al., 1998)), we will proceed by computing the inner products $u(\theta) = u(\alpha - \beta) = \langle g_\alpha, g_\beta \rangle$ analytically. The derived results are most similar to the ones given by Maurer and von der Malsburg (1995), who computed the inner product of two DC-free near-Gabor kernels with different wave vectors k and a common uniform shape parameter σ, but without considering steerability directly. The form of $u(\theta)$ depends on the type of the oriented filter family. We will derive results for Gabor, DC-free near-Gabor and angular Gaussian filters in Section 3.4.

3.3 Steering error

The steering property of Gabor-type filters is not exact, but only approximate. The error in the steering approximation depends heavily on the number of basis filters and shape parameters S. Let us define the measure for steering error by

$$E_s = \max_\theta \sqrt{\frac{\langle g(\theta) - \mathbf{k}(\theta)^T \mathbf{g}, g(\theta) - \mathbf{k}(\theta)^T \mathbf{g} \rangle}{\langle g(\theta), g(\theta) \rangle}}, \quad (3.10)$$

that is, the L2-norm distance of the maximum relative impulse response error. The same error measure was used in (Greenspan et al., 1994). In an evenly spaced filter bank the maximum error occurs always exactly between known filter orientations, that is, if filters are in orientations $\theta_i = \pi \frac{i}{N}, i \in \{0, 1, \dots, N-1\}$, maximum error is reached at $\theta = \frac{\pi}{2N}$. It is, then, straightforward to evaluate numerically the maximum steering error with different filter shape parameters S. Since we have separate steering functions $\mathbf{k}(\theta)$ for even and odd filters of the quadrature pair, we define the total steering error as the average of the even and odd filter errors,

$$E_s^{avg} = \frac{E_s^{even} + E_s^{odd}}{2}. \quad (3.11)$$

The level of acceptable approximation error depends on the application. For example, the quadrature pair formed by Gabor filters is not exact because of the infinite support of the Gaussian function. As a guideline, we might allow a roughly equal maximum error caused by the approximative steering. Also the noise level affects the choice of admissible error. In this context, the term 'noise'

means the error residual between the object model and the image, which is often significantly larger than the pixel noise of the image acquisition process.

It is possible to reduce the maximum steering error by having an offset of $\frac{\pi}{2N}$ in the orientations of even and odd filters or alternatively by having a different number of basis filters for even and odd filters (Schenk and Brady, 2003). The steering error then becomes more evenly distributed across the rotation angle and is nowhere zero. These improvements can be used without complications with the presented approach, but because of simplicity and clarity we will not consider them here.

3.4 Steering of Gabor-type filters

3.4.1 Parameterization of Gabor filters

The Gabor filter with a spherical Gaussian envelope function is described by

$$f(\xi;\ \mu, \sigma, \theta) = \frac{|\mu|^2}{\sigma^2} \exp\left(-\frac{|\mu|^2}{2\sigma^2}|\xi|^2\right) \exp\left(i\mu^T R_\theta \xi\right), \tag{3.12}$$

where $\xi = \begin{bmatrix} x & y \end{bmatrix}^T$ are the spatial coordinates, the wave vector $\mu = [\, f_c \;\; 0]^T$ determines the center frequency f_c of the filter and also acts as a scaling factor in this parameterization, σ controls the number and strength of spatial domain side lobes, and θ determines the orientation of the filter via the rotation matrix

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Following Daugman (1985), we will consider a more general form of Eq. (3.12), with different scaling constants σ_x and σ_y along the two axes in the spatial plane,

$$\begin{aligned} f(\xi;\ \mu, S, R_\theta) &= \frac{|\mu|^2}{\sqrt{\det(S)}} \exp\left(-\frac{|\mu|^2}{2}(R_\theta\xi)^T S^{-1} R_\theta \xi\right) \exp\left(i\mu^T R_\theta \xi\right) \\ &\propto N(0, |\mu|^2 R_\theta^T S R_\theta) \cdot \exp\left(i\mu^T R_\theta \xi\right). \end{aligned} \tag{3.13}$$

$N(0, |\mu|^2 R_\theta^T S R_\theta)$ denotes the Gaussian distribution with zero mean and covariance matrix

$$S = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{3.14}$$

which is rotated with the matrix R_θ and scaled by $|\mu|^2$. If $\sigma_x = \sigma_y$, Eq. (3.13) reduces to Eq. (3.12). Note that in this parameterization the resulting filters have a constant template shape determined by S, and the filter is rotated around

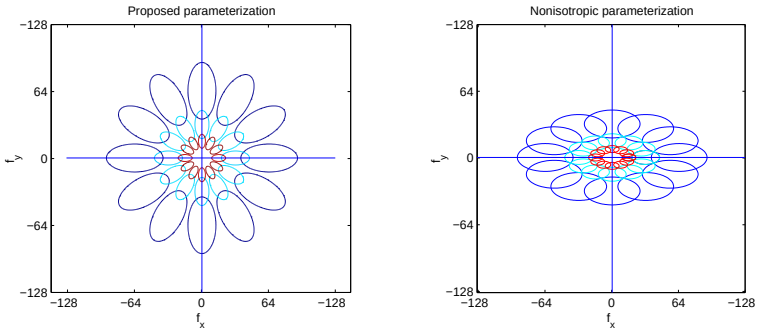


Figure 3.1: Left: Filter tessellation using the proposed parameterization. The filters retain their shape under rotation, and center frequency affects also the size of the filter envelope. Right: Alternative, non-isotropic parameterization. See text for explanation.

the origin by the parameter θ. Even more generic parameterizations which still achieve minimum uncertainty are possible (Daugman, 1985). For example the wave vector μ does not need to be aligned with axes of the Gaussian envelope. Also an arbitrary phase constant can be added so that the real and imaginary parts of the filter do not correspond to even and odd symmetric real-valued filters, but are weighted sums of both. We will follow Daugman (1985) and ignore these complications since our main interest is the effect of the shape parameters on the properties of the Gabor filter.

In order to make clear the properties of the proposed parameterization, Fig. 3.1 shows a tessellation of Gabor filters generated by directly changing the orientation parameter θ and center frequency f_c. It should be noted that the shape parameters σ_x and σ_y control the shape of the unrotated filter along the x- and y-coordinates. Since the shape of the filter remains constant under rotation in terms of angular and radial bandwidth, this means that σ_x and σ_y correspond to the spatial filter width in horizontal and vertical directions only in the unrotated orientation ($\theta = 0$). In general σ_x and σ_y are related to the radial and angular frequency bandwidths of the filter, respectively. Fig. 3.1 also shows an alternative parameterization, in which the filter shape is not preserved under rotation. This kind of filter tessellation is useful in situations where the horizontal and vertical coordinates themselves have different properties, for example if the sampling rate is not same in horizontal and vertical directions.

The Fourier transform of a zero-mean Gaussian function is also a Gaussian function, although no longer normalized, and modulation by complex plane wave corresponds to a shift from the origin in the Fourier plane by the amount described by $R_\theta^T \mu$. The rotation property of 2D Fourier transform states that rotations in the spatial plane correspond directly to rotations in the Fourier plane. As a result, the

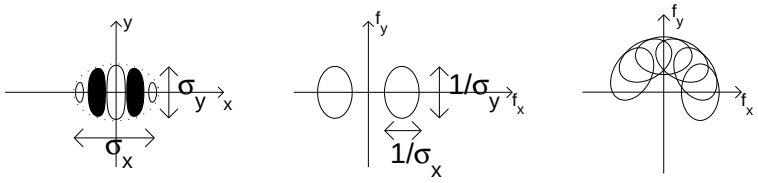


Figure 3.2: Left: Even Gabor filter in the spatial domain. Equal magnitude level of the complex filter is shown with dashed line. Middle: Even Gabor filter in the frequency domain. Right: Six complex Gabor filters in the frequency domain, with different preferred orientations θ. Only half of the frequency plane needs to be covered.

Fourier transform of the complex Gabor filter is a single Gaussian function

$$\mathcal{F}\{f\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta). \tag{3.15}$$

We denote the real-valued even and odd Gabor filters with $g = Re\{f\}$ and $h = Im\{f\}$, respectively, so that $f = g + ih$. Fourier transforms of real and imaginary parts of the complex filter are sums of two Gaussian functions,

$$\mathcal{F}\{g\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) \tag{3.16}$$

and

$$\mathcal{F}\{h\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) - N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta). \tag{3.17}$$

The uncertainty principle states that the product of the areas in spatial and frequency domains occupied by the filter is constant. This means that if the filter is made wider in one domain, it becomes narrower in the other. As an example, an even Gabor filter with shape parameters $\sigma_x = 3$ and $\sigma_y = 2$ is illustrated schematically in Fig. 3.2. The spatial width $\Delta x \propto \sigma_x$ and height $\Delta y \propto \sigma_y$ of the filter are conjugate variables with the spectral widths $\Delta f_x \propto 1/\sigma_x$ and $\Delta f_y \propto 1/\sigma_y$, so that their product is constant, conforming to the uncertainty principle. When we use the complex Gabor filter, which is a single Gaussian function in the frequency domain, only half of the frequency plane needs to be covered. This is due to the fact that the signals (images) we analyze are real-valued, and thus their Fourier spectra are symmetric.

3.4.2 Steering of Gabor filters

In the following we assume without loss of generality that the filters have unit center frequency $\mu = [1\ 0]^T$. The even and odd Gabor filter both need separate steering coefficients $\mathbf{k}$. We begin by computing the inner products in the elements of the matrix $\mathbf{G}$ in Eq. (3.7). The inner product integral $u(\theta)$ of two even Gabor

filters in the frequency space is

$$\begin{aligned}\langle g, g_\theta \rangle &= \langle \mathcal{F}\{g\}, \mathcal{F}\{g_\theta\} \rangle \\ &= \int \left(N(\mu, S^{-1}) + N(-\mu, S^{-1}) \right) \\ &\cdot \left(N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) \right) d\omega \end{aligned} \tag{3.18}$$

which, using a symmetry argument, is equivalent to

$$= 2 \int N(\mu, S^{-1}) N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(\mu, S^{-1}) N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) d\omega. \tag{3.19}$$

The inner product of two Gaussian functions (the normalization constant of a product of two Gaussian functions) is also Gaussian with respect to the parameters of the functions,

$$\langle N(a, A) \cdot N(b, B) \rangle \propto \sqrt{\frac{|C|}{|A||B|}} \exp\left(-\frac{1}{2} (a^T A^{-1} a + b^T B^{-1} b - c^T C^{-1} c) \right), \tag{3.20}$$

with $C = (A^{-1} + B^{-1})^{-1}$ and $c = CA^{-1}a + CB^{-1}b$.

We can now identify $a = \mu$, $A = S^{-1}$, $b = \pm R^T \mu$, $B = R^T S^{-1} R$, $C = (S + R^T S R)^{-1}$ and $c = (S + R^T S R)^{-1}(I \pm R^T) S\mu$ in order to compute the two integral terms. Applying the result gives after some manipulation

$$\begin{aligned}\langle g, g_\theta \rangle &= \frac{1}{Z_g} \sqrt{|U|} \exp\left(\frac{1}{2} \nu^T (U + R_\theta U R_\theta^T) \nu \right) \\ &\cdot \cosh\left(-\frac{1}{2} \nu^T (U R_\theta^T + R_\theta U) \nu \right) \end{aligned} \tag{3.21}$$

where $U = (S + R_\theta^T S R_\theta)^{-1}$, $\nu = S\mu = [\sigma_x^2 \; 0]^T$ and Z_g is a normalization factor. It is most conveniently computed by requiring that the inner product equals to one at $\theta = 0$, yielding the result

$$Z_g = \frac{1}{2} \sigma_x^{-1} \sigma_y^{-1} \exp\left(\frac{1}{2} \sigma_x^2 \right) \cosh\left(-\frac{1}{2} \sigma_x^2 \right). \tag{3.22}$$

Inner product function of two odd Gabor filters h and h_θ is obtained similarly,

$$\begin{aligned}\langle h, h_\theta \rangle &= \frac{1}{Z_h} \sqrt{|U|} \exp\left(\frac{1}{2} \nu^T (U + R_\theta U R_\theta^T) \nu \right) \\ &\cdot \sinh\left(-\frac{1}{2} \nu^T (U R_\theta^T + R_\theta U) \nu \right), \end{aligned} \tag{3.23}$$

with the normalization factor

$$Z_h = \frac{1}{2}\sigma_x^{-1}\sigma_y^{-1}\exp\left(\frac{1}{2}\sigma_x^2\right)\sinh\left(-\frac{1}{2}\sigma_x^2\right). \tag{3.24}$$

In the case of spherical Gabor filters ($\sigma_x = \sigma_y = \sigma$), the expressions of the inner products simplify significantly to

$$\langle g, g_\theta\rangle = \frac{\cosh(\frac{\sigma^2}{2}\cos\theta)}{\cosh(\frac{\sigma^2}{2})}. \tag{3.25}$$

and

$$\langle h, h_\theta\rangle = \frac{\sinh(\frac{\sigma^2}{2}\cos\theta)}{\sinh(\frac{\sigma^2}{2})}. \tag{3.26}$$

Having computed the inner products, we can now solve the optimal steering coefficients using Eq. (3.7). All of the elements in the matrix **G** are inner products between two rotated Gabor filters. Specifically, as the value of the inner product between two filters in orientations θ_1 and θ_2 depends only on the difference between orientations $\theta' = \theta_1 - \theta_2$, we can fix the coordinate system of the rotation and define

$$u(\theta') = \left\langle g_{\theta_1}, g_{\theta_2}\right\rangle = \langle g_0, g_{\theta'}\rangle . \tag{3.27}$$

Using only this single inner product function, for which it holds that $u(0) = 1$, Eq. (3.9) is expressed as

$$\hat{\mathbf{k}}(\theta) = \begin{bmatrix} 1 & u(\theta_1-\theta_2) & \dots & u(\theta_1-\theta_N) \\ u(\theta_2-\theta_1) & 1 & \dots & u(\theta_2-\theta_N) \\ u(\theta_3-\theta_1) & u(\theta_1-\theta_2) & \dots & u(\theta_3-\theta_N) \\ \vdots & & & \vdots \\ u(\theta_N-\theta_1) & u(\theta_{N-1}-\theta_2) & \dots & 1 \end{bmatrix}^{-1} \begin{bmatrix} u(\theta-\theta_1) \\ u(\theta-\theta_2) \\ u(\theta-\theta_3) \\ \vdots \\ u(\theta-\theta_N) \end{bmatrix}. \tag{3.28}$$

The matrix inverse is constant with respect to the steering angle θ, and it is conveniently solved numerically, N being small. The optimal steering functions are of the form $k_j(\theta) = \sum_i w_{ji}u(\theta-\theta_i)$, that is, sums of shifted versions of a single inner product function $u(\theta)$, the weights w_{ji} being the elements of the matrix $\mathbf{G}^{-1}$.

The matrix **G** is in principle not guaranteed to be invertible, and indeed with basis filters which are highly correlated, it can be numerically close to being singular. To overcome this difficulty, we can compute the Singular Value Decomposition of **G** and use it to calculate the SVD inverse (Greenspan et al., 1994).

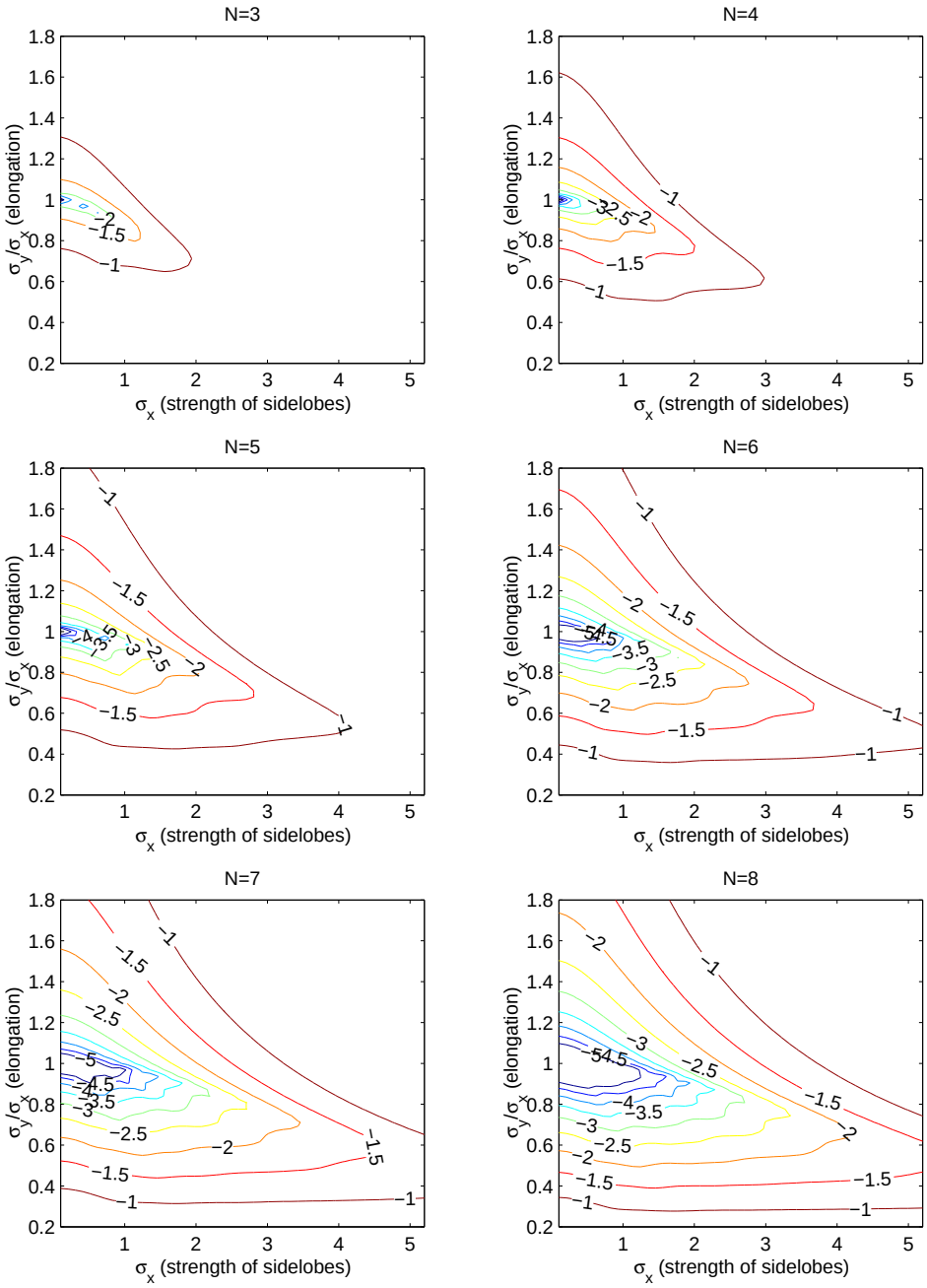


Figure 3.3: Base-10 logarithm of steering error in impulse responses of Gabor filters. In typical applications 1% error (the -2 contour) might be considered acceptable.

Once we have solved the steering coefficients, it is straightforward to compute the steered filter approximation as a linear sum of the original filters, using Eq. (3.5). In order to analyze the steering error performance of the filters, we evaluate the approximation error numerically using Eq. (3.10).

The error behavior of Gabor filters in banks of three to eight filters is shown in Fig. 3.3. The overall effects of the filter shape parameters S are similar with any number N of basis filters. Steering error becomes progressively lower as more basis filters are present, and while the spatial domain side lobes are not prominent. Most importantly, spherical Gabor filters are in general not optimal in terms of steering, and slightly flattened filters with $\sigma_y/\sigma_x < 1$ have considerably lower steering error. In other words, steerability is improved if the filters are less specific in the angular dimension than in the frequency dimension. This behavior is compatible with the properties of derivative of Gaussian filters, which are similarly flattened in the frequency space although their envelope function is a spherically symmetric Gaussian.

Fig. 3.4 shows the effect of flattening the filter in Gabor filters and third derivative of Gaussian filters. The latter are exactly steerable when the exponential term is a spherical Gaussian (that is, $\sigma_y/\sigma_x = 1$). However, exact steerability is a very brittle property of the filters, and the steerability of derivative of Gaussian filters quickly breaks down if the exponential term is not exactly spherical. Steerability has been proposed also to have biological relevance (Edelman, 1996), but for this reason, it is unlikely that exact steerability, instead of approximate steerability, could be relevant for biological vision, as the parameter values of oriented filters in biological systems have significant variation and probably cannot be specified very accurately. Non-spherical derivative of Gaussian filters and Gabor filters show more or less similar steering performance. It is also interesting to note that the optimum values for steering are slightly different for even and odd Gabor filters.

3.4.3 Steering of DC free near-Gabor filters

In many applications the DC component of the even Gabor filter is problematic, because it makes the filters sensitive to the absolute brightness of the image, and it is preferable to use filters with zero DC response. A simple way to remedy the deficiency is to subtract a second Gaussian term located at origin, forcing the DC response of the filter to zero (Ronse, 1993),(Lades et al., 1993). The resulting complex-valued filter

$$f = N(0,\, R_\theta^T S R_\theta) \cdot \left(\exp\left(i\mu^T R_\theta \xi\right) - \exp\left(-\frac{\sigma_x^2}{2}\right)\right) \qquad (3.29)$$

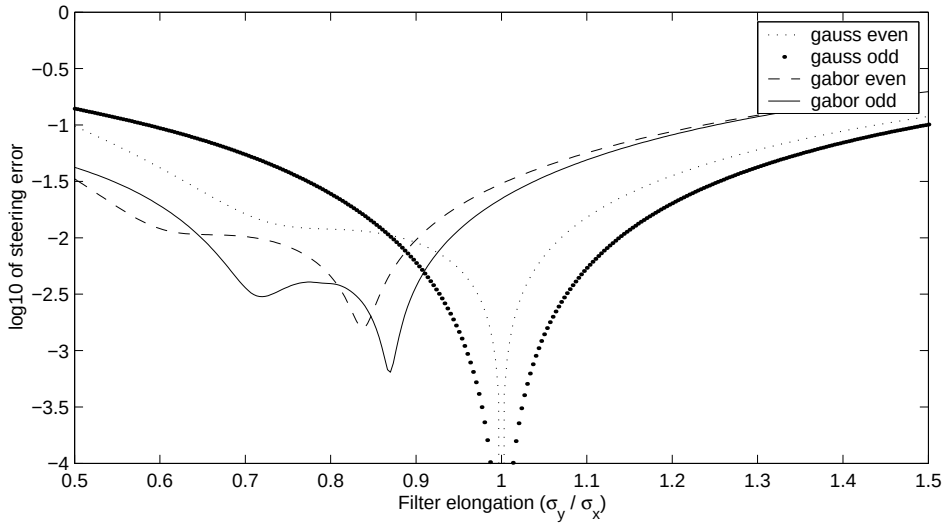


Figure 3.4: Maximum steering error of Gabor and third derivative of Gaussian filters with respect to elongation of the Gaussian envelope function. Both filter banks are evenly spaced in orientation and contain 6 filters, with $\sigma_x = 2$.

is no longer a Gabor filter, but approximates one quite well, especially with large σ_x when the DC component (and thus also the subtracted exponential term) is small.

Optimal steering functions $\mathbf{k}(\theta)$ can be derived for DC free near-Gabor filters using the presented approach. The odd real-valued filter remains unchanged. The inner product function of the even filter

$$\mathcal{F}\{g\} = N(\mu, S^{-1}) + N(-\mu, S^{-1}) - 2\exp(-\sigma_x^2/2)N(0, S^{-1}) \tag{3.30}$$

has now four terms,

$$\begin{aligned}\langle g, g_\theta \rangle = \int\ & N(\mu, S^{-1})N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) \\ & + N(\mu, S^{-1})N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) \\ & - 4\exp\left(-\sigma_x^2/2\right) N(\mu, S^{-1})N(0, R_\theta^T S^{-1} R_\theta) \\ & + 2\exp\left(-\sigma_x^2\right) N(0, S^{-1})N(0, R_\theta^T S^{-1} R_\theta) d\omega.\end{aligned} \tag{3.31}$$

Proceeding in the same manner as in the case of the Gabor filter, we obtain

$$\langle g, g_\theta \rangle = \frac{1}{Z_g}\sqrt{|U|}\left[\exp\left(\frac{1}{2}\nu^T(U + R_\theta U)(I + R_\theta^T)\nu\right) + \exp\left(\frac{1}{2}\nu^T(U - R_\theta U)(I - R_\theta^T)\nu\right) - 4\exp\left(\frac{1}{2}\nu^T U\nu\right) + 2\right], \tag{3.32}$$

with $U = (S + R_\theta^T S R_\theta)^{-1}$ and $\nu = S\mu$ as before, and the normalization factor

$$Z_g = \frac{1}{2}\sigma_x^{-1}\sigma_y^{-1}\left(\exp\left(\sigma_x^2\right) - 4\exp\left(\frac{1}{4}\sigma_x^2\right) + 3\right). \tag{3.33}$$

The effect of the two additional terms of the integrand is however quite small in practice unless σ_x is close to zero. The steering error, depicted in Fig. 3.5, is similar to that of Gabor filters, but with narrower region of good steerability.

3.4.4 Steering of angular Gaussian filters

Let us consider approximately steerable filters which are separable in polar frequency coordinates so that the filter can be expressed as a product of two univariate functions, $g(r, \theta) = p(r)q(\theta)$. We choose the angular component $q(\theta)$ to be the Gaussian function

$$N_q(\theta, \sigma_\theta) = \exp\left(-\frac{D(\theta, \theta')^2}{2\sigma_\theta^2}\right) \tag{3.34}$$

with the 2π-periodic distance measure (Yu et al., 2001)

$$D(\theta, \theta') = \min\left(|\theta - \theta'|, |\theta - \theta' - 2\pi|, |\theta - \theta' + 2\pi|\right). \tag{3.35}$$

Polar Gabor (Haley and Manjunath, 1995) and log-Gabor (log-Normal) filters (Field, 1987), (Kovesi, 1999) are both examples of this class of filters. Note that because of the periodicity of the distance measure D, the function N_q is only Gaussian with respect to the distance measure, but not with respect to the angle θ directly.

The inner product function of two angular Gaussians is

$$u(\theta) = \frac{1}{2}\int_{-\pi}^{\pi}\left(N_q(0, \sigma_\theta) \pm N(\pi, \sigma_\theta)\right)\left(N_q(\theta, \sigma_\theta) \pm N_q(\theta + \pi, \sigma_\theta)\right)d\theta', \tag{3.36}$$

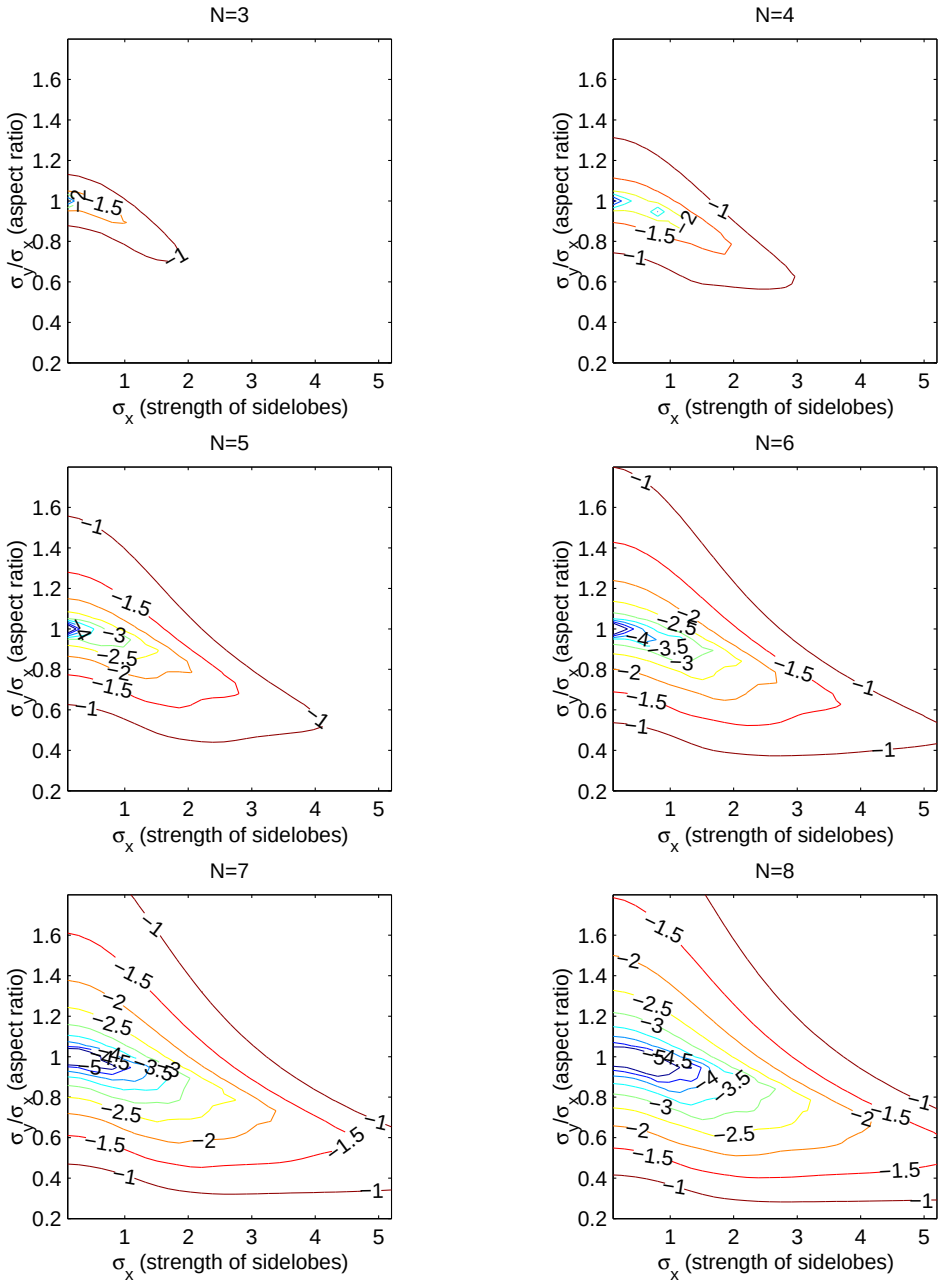


Figure 3.5: Base-10 logarithm of steering error in impulse responses of DC-free near-Gabor filters.

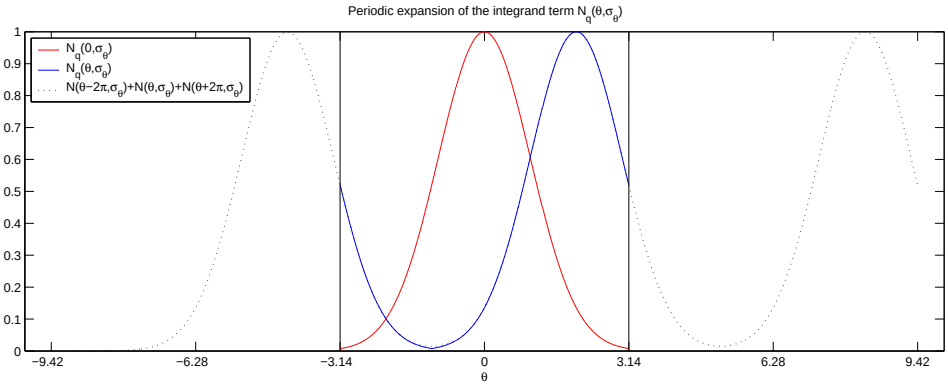


Figure 3.6: Periodic expansion of one integrand term $N_q(\theta, \sigma_\theta)$ over three period lengths.

where the choice of sign determines whether the filter is even or odd symmetric (real or imaginary component of the Hilbert pair). Again because of reasons of symmetry, it holds that

$$u(\theta) = \int_{-\pi}^{\pi} N_q(0, \sigma_\theta) \left(N_q(\theta, \sigma_\theta) \pm N_q(\theta + \pi, \sigma_\theta) \right) d\theta'. \tag{3.37}$$

The integral in Eq. (3.37) is difficult to evaluate because of the periodic distance measure $D(\theta, \theta')$, which causes the metric to wrap around from π to $-\pi$. In order to approximate the integral, we approximate the expression

$$\int_{-\pi}^{\pi} N_q(0, \sigma_\theta) N_q(\theta, \sigma_\theta) d\theta' \tag{3.38}$$

by eliminating the periodicity and expanding the latter of the two terms into

$$\int_{-\pi}^{\pi} N(0, \sigma_\theta) \left(N(\theta - 2\pi, \sigma_\theta) + N(\theta, \sigma_\theta) + N(\theta + 2\pi, \sigma_\theta) \right) d\theta', \tag{3.39}$$

where we have now converted the periodic Gaussian functions N_q into direct Gaussian functions of the angle parameter, $N(\theta, \sigma_\theta) \propto \exp(-\frac{(\theta-\theta')^2}{2\sigma_\theta^2})$. Figure 3.6 illustrates the idea of the approximation. The convolving integral term (denoted with blue line) is expanded into a sum of three shifted versions of the original function (denoted with dotted black line), taking into account the wrapping effect of the periodic distance measure.

If the integrands had support only in the interval $[-\pi, \pi]$, the approximation would be exact. A slight error is introduced with functions which have wider

support. The error due to summing of the shifted versions can be seen in the lowest point of the curve, where the approximation has slightly higher value than the original function. Practical oriented filters will have a narrow enough orientation bandwidth so that the approximation is valid. Further, the integration limits can be expanded to $[-\infty, \infty]$ under the same conditions.

The preceding approximation scheme leads to the approximation

$$u(\theta) \approx \int_{-\infty}^{\infty} \exp\left(-\frac{\theta'^2}{2\sigma_\theta^2}\right)\left(\sum_{n=-1}^{1} \exp\left(-\frac{(\theta-\theta'+2n\pi)^2}{2\sigma_\theta^2}\right) \pm \exp\left(-\frac{(\theta-\theta'+(2n-1)\pi)^2}{2\sigma_\theta^2}\right)\right)d\theta' \tag{3.40}$$

for the inner product function $u(\theta)$. Computing the integral, we obtain

$$u(\theta) = \frac{1}{Z}\sum_{n=-2}^{2} (\pm 1)^{n+1} \exp\left(-\frac{1}{4\sigma_\theta^2}(\theta - n\pi)^2\right), \tag{3.41}$$

which is a good approximation for the exact integral in the interval $\theta \in [-\pi, \pi]$ when σ_θ is not too large. The normalization term is now simply

$$Z = \sum_{n=-2}^{2} (\pm 1)^{n+1} \exp\left(-\frac{(n\pi)^2}{4\sigma_\theta^2}\right). \tag{3.42}$$

The steering error of angular Gaussian filters, depicted in Fig. 3.7, is easier to analyze because only the angular width parameter σ_θ affects the steering performance. Given the number of basis filters, an optimal filter width exists with respect to the steering error. The error starts to rise again with larger than optimal angular bandwidths. However, such filters are uninteresting for practical applications, because if a lower resolution is needed, the number of basis filters can be decreased instead. The useful filters lie in the region with filter bandwidths equal to or narrower than the optimal width of the given number of basis filters, and in this region the analytical approximation is nearly indistinguishable from a numerically computed optimal solution (not shown in the figure), confirming that the integral approximation is valid. It is possible to trade steering performance for bandwidth. The optimal width for a bank of eight filters is $\sigma_\theta = 37$ degrees, but if we allow a maximum error of 1% in the impulse response, the filter width can be reduced to 21 degrees, leading to a significantly improved angular resolution.

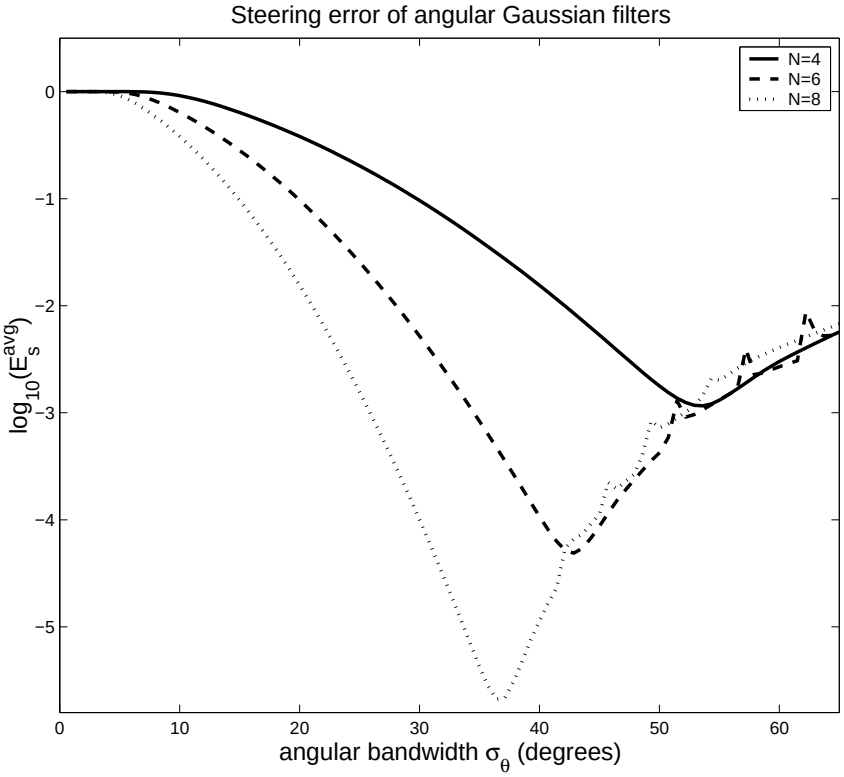


Figure 3.7: Base-10 logarithm of maximum relative steering error of filters with an angular Gaussian component as functions of their angular bandwidths in degrees.

3.5 Accuracy of analytical and numerical steering equations

Next, the accuracy of the inner product functions is discussed. Fig. 3.8 shows numerically computed inner product functions of Gabor, near-Gabor and angular Gaussian filters, compared to the evaluated analytical expressions in Eqs. 3.21, 3.32 and 3.41. The analytical and numerical inner products are nearly indistinguishable in practice, with a maximum error in the order of 10^{-12} for Gabor filters and 10^{-4} for angular Gaussian filters. While Eq. 3.41 is only an approximation of the exact inner product integral, the accuracy is good enough for all practical applications.

Fig. 3.9 shows the angular bandwidths of Gabor filters on top of the steering error in a bank of eight filters. A difficulty in regular Gabor filters is that the angular and radial frequency bandwidths depend nonlinearly on the shape parameters, and cannot be chosen independently if a particular angular or radial

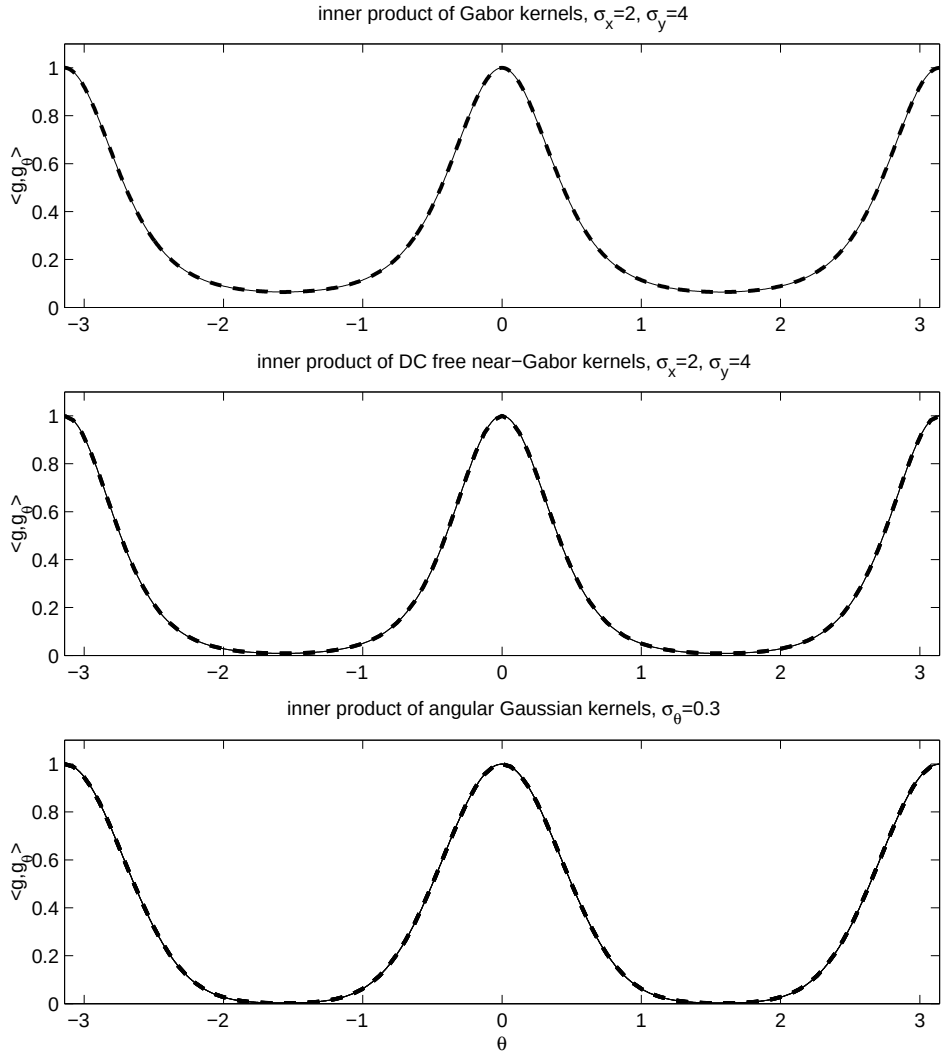


Figure 3.8: Inner product functions of Gabor and near-Gabor filters with shape parameters $\sigma_x = 2$, $\sigma_y = 4$, and inner product of angular Gaussian filters with $\sigma_\theta = 0.3$. Solid line denotes the numerical results and dashed line denotes the evaluated analytic results.

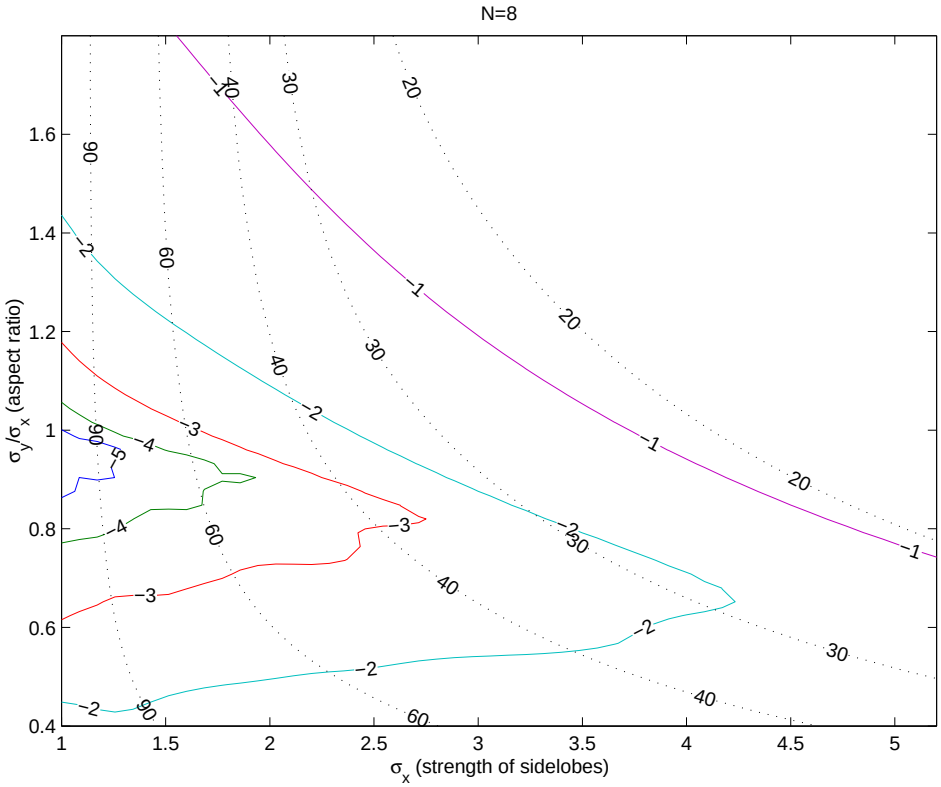


Figure 3.9: Angular bandwidth curves (in degrees) and steering error of eight Gabor filters. Angular bandwidth depends nonlinearly on the shape parameters σ_x and σ_y. Compare with Fig. 3.7.

frequency bandwidth is required. Supposing that we need, for example, an orientation bandwidth of $\sigma_\theta = 30$ degrees, it can be achieved with a Gabor filter with parameters $\sigma_x = 3.7$ and $\sigma_y/\sigma_x = 0.7$, with an error of approximately 1%. The radial frequency bandwidth is now fixed with these parameters, which cannot be significantly changed without increasing the error level. In contrast, an angular Gaussian filter reaches the angular bandwidth of $\sigma_\theta = 21$ degrees with the same error level, while still having complete freedom in choosing the radial frequency bandwidth. The price of this flexibility is the slightly increased joint uncertainty.

3.6 Exactly steerable filters and their Gabor approximations

In order to design exactly steerable filters which are close to Gabor filters in the sense of the L2-norm, one can start with a Gabor filter prototype and use Singular Value Decomposition to find the optimal basis giving exact steerability within an acceptable approximation error margin. We consider here only one-parameter rotation transformations. Following Perona (1995), we will compute numerically the exactly steerable basis functions with SVD. Cascade basis reduction (Teo and Hel-Or, 1999) can be used in order to reduce the dimensionality of the decomposition, when the number of transformation parameters makes the direct SVD computationally unfeasible.

In principle, there is no theoretical guarantee that the resulting filter will have the same properties as the filter it approximates. However, since the approximation error of the SVD method is constant with respect to orientation (Perona, 1995), that is, all orientations are approximated equally well by the basis functions in the sense of the L2-norm, the resulting filters will generally preserve their good localization properties. Additionally, zero DC is preserved since the complex harmonic basis functions $\exp(ik\theta)$ have zero DC themselves.

It is however reasonable to ask how much we gain by computing the SVD in the case of single parameter rotations. For example, given a DC free near-Gabor filter with shape parameters $\sigma_x = \sigma_y = 2$ and an approximation error of 1% (measured with Eq.3.10), seven basis functions are needed for the even filter and six for the odd filter, depicted in Fig. 3.10. From Fig. 3.5 it can be seen that the steering error when using seven Gabor filters has a similar amount of steering error (it is exactly 0.99%). Thus we have gained essentially nothing with the SVD computation.

Table 3.1 gives some additional examples of exactly steerable SVD approximations of Gabor filters. Again the 1% approximation error level was used. In general the SVD method appears to save one or two basis filters with larger shape parameter values. It can be concluded that the SVD approach becomes potentially useful in reducing the number of basis filters when the Gabor shape parameters are such that a large number of filters is required for steerability, but such high numbers of basis filters are not widely used in the literature.

If further analysis is performed in the Gabor filter space and not by directly using the SVD basis filter responses, we need to apply an additional linear transformation to the SVD basis filter responses in order to obtain the Gabor filter responses. Depending on the hardware architecture and software implementation, this additional computation can nullify the computation benefit achieved using the SVD approach, since the 2D FFT algorithm has computational complexity $O(N^2 \log N)$ while the linear transformation of a single filter response of an image

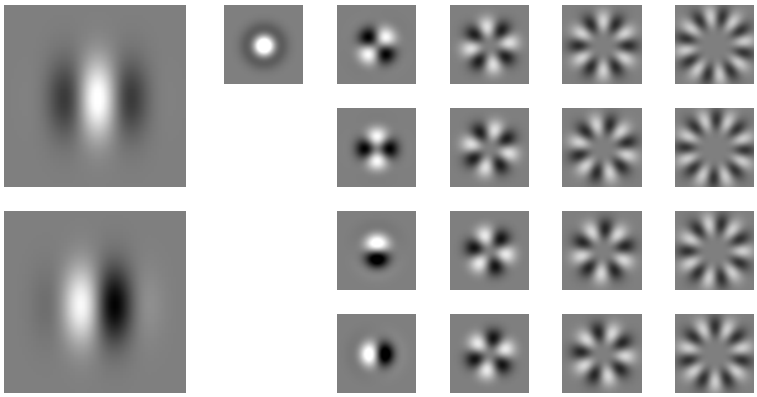


Figure 3.10: Even and odd DC free near-Gabor filters with shape parameters $\sigma_x = \sigma_y = 2$, and the corresponding basis filters (left singular vectors) computed with SVD. The basis filters consist of angular quadrature pairs of even and odd complex harmonics. The radial components of the basis filters are quite close to being Gaussian.

has complexity $O(N^2M)$, where N^2 is the number of pixels in the image and M is the number of SVD basis filters. In other words, it can be computationally less demanding to simply compute additional convolution results using the 2D FFT algorithm than to apply additional pixelwise processing to the filter reponses, if the image size is small. Using a straightforward, not particulary optimized Matlab implementation, it was found that for example in the case of $M = 10$ basis filters, the SVD approach combined with the linear transformation to Gabor space becomes computationally more efficient than performing one additional direct 2D FFT computation if $N > 150$. The exact difference in the amount of computation using the two methods is naturally both implementation and platform dependent, but it can be concluded that the SVD approach does not always give performance savings despite being able to use a slightly lower number of basis filters than the direct computation of Gabor responses, if the responses need to be transformed to the Gabor space for analysis. It should be noted here that the SVD methods presented in Teo and Hel-Or (1999) have been designed for multiparameter transformations where their benefits become apparent.

Conversely, one can consider approximating exactly steerable filters with Gabor filters. The primary reason for such an excercise is to compare the properties and practical performance of the filters. The latter will be considered in Section 5.5.

For example, the derivative of Gaussian filters are classical exactly steerable filters which are relatively close to Gabor filters. Directly minimizing the approximation error as defined by Eq. 3.10 one can find the closest Gabor equivalents to the derivative of Gaussian filters. In addition to shape parameters,

σ_x	σ_y	N_{even}^{svd}	N_{odd}^{svd}	E_{steer}^{Gabor}	$N_{<1\%}^{Gabor}$
2	1.6	4	5	0.85%	5
2	2	7	6	0.99%	7
4	3	8	8	1.86%	9
2	4	15	15	1.93%	17
3	6	19	20	1.72%	22

Table 3.1: Examples of near-Gabor filter banks with shape parameters σ_x and σ_y, and the number of required basis filters N_{even}^{svd},N_{odd}^{svd} for 1% approximation error with the SVD method, steering error E_{steer}^{Gabor} of a filter bank with the same number of Gabor filters, and number of Gabor filters $N_{<1\%}^{Gabor}$ required for 1% steering error.

target filter	σ_x	σ_y	E_{approx}^{Gabor}	E_{steer}^{Gabor}
2nd derivative of Gaussian	1.96	1.46	8.3%	3.0% (N=4)
3rd derivative of Gaussian	2.49	1.80	6.6%	2.1% (N=5)
4th derivative of Gaussian	2.92	2.06	6.1%	1.4% (N=6)
5th derivative of Gaussian	3.30	2.29	4.8%	0.9% (N=7)
⋮				
10th derivative of Gaussian	4.57	3.20	3.4%	0.1% (N=11)

Table 3.2: Shape parameters σ_x and σ_y for DC free near-Gabor approximations of derivative of Gaussian filters, the error performance E_{approx}^{Gabor} of their Gabor approximations, and the steering errors $E_{steer}Gabor$ of the approximations. The steering errors are given with the same number of basis filters (in parenthesis) with which the polynomial approximation of the DoG filter pair is exactly steerable.

the center frequency is also optimized, since it is slightly different due to the different shapes of the response envelopes.

Table 3.2 gives the approximation parameters and the related approximation and steering errors of DC free near-Gabor filters. As the order of the derivative increases, the derivative of Gaussian filters become more like Gabor filters, and the approximation error decreases. The aspect ratio of the approximating Gabor filter remains remarkably constant, the quotient σ_y/σ_x having the value of 0.70 with all derivative orders from the second to the tenth.

Yokono and Poggio (2004b) note that Gabor functions can be regarded as approximations of high order Gaussian derivatives. However, such an approximation requires a very particular choice of the Gabor shape parameters. Conversely, it can be said that derivative of Gaussian filters are increasingly Gabor-like as their order increases. Very high order derivatives such as the tenth

derivative have not been widely used in the literature, possibly because such high order image derivatives have no physical significance, and estimates of high order derivatives are informally known to be sensitive to noise in the one-dimensional case. Their close resemblance to certain Gabor filters however suggests that there is no inherent reason why the high order derivative of Gaussian filters would not be suitable for applications which use similarily shaped Gabor filters, such as feature detection and texture classification.

Greenspan et al. (1994) design a steerable filter bank with four orientations using the SVD approach and report a relative steering error of 0.5% with filters which are approximations of Gabor filters with shape parameter $\sigma = \pi/2 \approx 1.57$ using our parameterization. For comparison, directly steering the Gabor filters with the shape parameters $\sigma_x = \sigma_y = 1.57$ has the relative steering error of 8.8% and flattening the Gabor filter slightly using the shape parameters $\sigma = [1.57\ 1.29]$ has the relative steering error of 1.4%.

3.7 Discussion

As the steerability of a filter (or a function) is dependent only on the Fourier spectrum of the angular component of the filter, the derived results for Gabor filters are not entirely surprising, since as a consequence of the sampling theorem, *any* function is steerable given enough basis functions. However, the important empirical finding of this chapter is that Gabor-type filters can be approximately steerable (with a tolerable steering error for practical applications) using only a low number of basis functions. Excellent steering performance can be obtained using 6 or 8 basis filters, a number which is a typical design choice in texture analysis and feature detection applications.

The practical advantages of Gabor-type filters are mainly that they have a simple analytical form, which makes the filter bank design problem computationally straightforward, and that the filters can be tuned for different applications by adjusting the parameters. The steering error of Gabor filters depends dramatically on the shape parameters, and nonspherical Gabor filters with $\sigma_x > \sigma_y$ have significantly better error performance than the more commonly used spherical Gabor filters.

In filter bank design, steerability gives a guideline for determining an appropriate number of orientations and scales for given filter shape parameters in order to obtain (nearly) uniform coverage of the frequency space. Even in applications which do not use steerability directly, it is typical to aim for covering the orietation and scale space in some sense "uniformly", so that there are no holes in the coverage of the frequency space. The usefulness of steerability in filter bank design is that it gives a numerical value to the uniformity of the filter bank with respect to orientations.

Chapter 4

Probabilistic framework for inference of images

4.1 Introduction

In this chapter we will proceed to apply the oriented filter banks as generic local gray-level feature detectors and construct probability distributions which give information about the location of a feature in the image plane. The local features are then combined into probabilistic models of complete objects, which are matched into novel images using probabilistic methods. This combination of methodology forms the theoretical basis for the probabilistic local feature based object matching system considered in this work. The system is based on the framework presented in (Tamminen, 2005).

The object matching system can be divided into four parts or stages of processing which are independent to some degree:

- Feature representation
- Feature similarity model
- Object similarity model
- Matching method

Local image features are represented in the system by the responses of the quadrature filter banks, in particular, Gabor-type filters. Other possibilities for local descriptors include steerable filters (Yokono and Poggio, 2004b), local image patches (Weyrauch et al., 2004), (Rothganger et al., 2003) and histogram-type representations (Lowe, 2003), (Zhang et al., 2007). It is common to all these representations that individual feature models are not typically very strong in

their predictive capability, and object models are composed of a number of local features.

The SIFT features (Lowe, 2003) are especially noteworthy as the SIFT matching algorithm can be considered to be a state-of-the-art method in rigid object matching. The SIFT features consist of a histogram-type representation of local image gradients, and have been found to perform very well in redetection of a previously seen object under new image transformations. Serre et al. (2007) argues that the SIFT features are unlikely to perform well in generic object recognition tasks because of their high degree of invariance. However, it is possible to adapt the core SIFT algorithm to better suit recognition purposes (Bicego et al., 2006). Compared to the Gabor jet based feature representation, the main difference is that the SIFT features are intrinsically rotation, translation and scaling invariant.

Feature similarity defines a distance metric in the feature space, and its purpose is to assign a numerical value for the resemblance of two local gray-level image structures. The term similarity stems from the error minimization approach in (Lades et al., 1993). In a probabilistic framework, the analogue of feature or object similarity is probability, which follows naturally from the error function or distance metric.

Most object similarity models considered in this work are based only on the joint probability of individual features, which is appropriate because the object models have a relatively low number of parameters. An exception is Chapter 5, where the object model contains both feature and shape probability models in order to perform well in recognition tasks. This object model is directly based on (Tamminen, 2005).

The matching methods applied in this work are based on probabilistic inference using random sampling. Traditionally, computer vision applications have often used straightforward error minimization instead of probabilistic reasoning, but recently probabilistic approaches have gained popularity in computer vision applications, especially when handling multiple object classes. In (Fei-Fei et al., 2003), the PCA-based feature models are somewhat simpler than in our approach, but the probability model for objects is significantly more sophisticated. Instead of random sampling, variational methods are used for finding the posterior distribution of parameters of a multi-dimensional Gaussian mixture model. An analytic approximation for the posterior would be a viable alternative for random sampling also in our case. A generative, hierarchical object model based on SIFT features was presented in (Mikolajczyk et al., 2006). The tree structures for the object models resemble our local feature based object representation. However, since the aim is in multiple object class detection, the object models are not very detailed, and the main modeling effort is spent for building a generic common codebook for the parts (or features) of all object classes.

Figure 4.1 shows an overview of the levels of processing in our system.

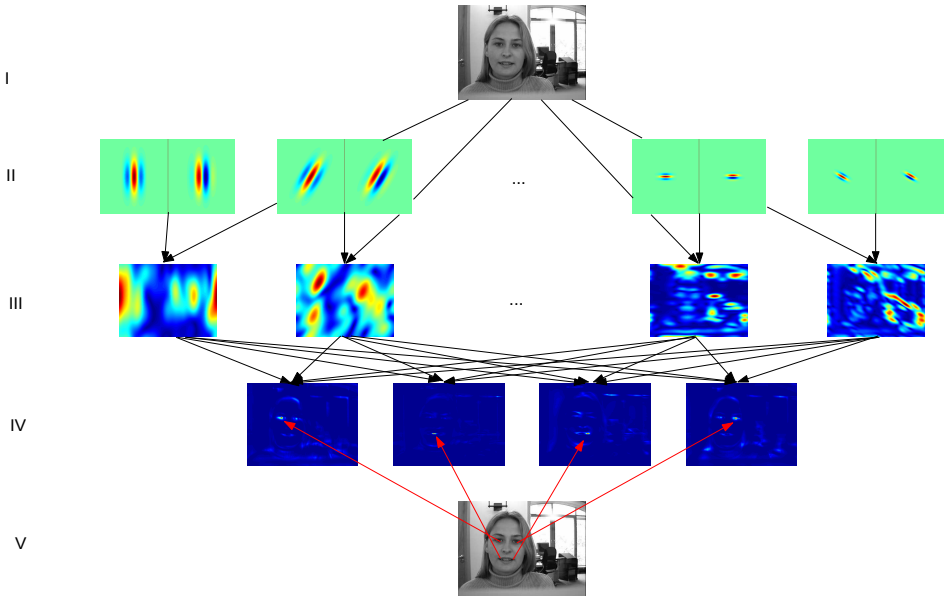


Figure 4.1: Overview of the object matching system considered in this work. From top to bottom: I Input image, II Multiscale, multiorientation filter bank, III Filter responses (magnitude shown), IV Feature likelihood distributions, V Local feature based object matching. Computation in stages III and IV is based on feedforward processing (denoted with black arrows), whereas the object matching in the final stage is based on random sampling of the feature location distributions (denoted with red arrows).

The image (I) is filtered first using a bank of multiscale, multiorientation filters (II). This can be considered a feature extraction stage. The responses of all filters (III) are combined and compared with feature prototypes. This comparison produces a number of probability distributions, each of which describes the presence of a single feature in a spatial location in the image (IV). Finally, the feature points of the object model are matched to image locations by randomly sampling the posterior distribution of feature location configurations (V). The system incorporates two different types of computation. The filter responses and the feature likelihood distributions are computed in a non-iterative, feedforward manner. In contrast, the random sampling of feature locations in the final stage proceeds in an iterative fashion. It is possible to implement rotation invariance in such a system on various levels of processing. The approach we will take in this work is to use feature detectors which are orientation-sensitive and preserve the orientation information for higher levels of processing, and handle the orientation parameter on levels IV and V.

Section 4.2 shows using examples how the responses filter banks describe the local structures in images. Section 4.3 reviews the similarity measures presented in the literature and relates the proposed similarity to them. Section 4.4 shows how to construct a likelihood probability distribution from the similarity measure. In Section 4.5 the ideas of the two previous chapters are combined with the concept of the similarity measure and present rotation-invariant versions of the similarity measures and the likelihood functions derived from them. Section 4.6 discusses restricted rotation invariance. Section 4.7 presents the object probability model considered in this work. The chapter concludes with a brief review of random sampling methods in Section 4.8. They are necessary in order to obtain samples from the object likelihood and posterior probability density functions. The random sampling methods are applied in Chapter 5 to analyze the effect of filter parameters on recognition, and in Chapter 7 to locate and recognize objects in rotated poses.

4.2 Quadrature filter banks and local features

Given a filter bank $\mathbf{f}$ containing complex-valued quadrature filters in different scales and orientations, the complex-valued filter responses $\hat{\mathbf{f}}_w$ are obtained by convolving the signal (or image) I with each filter,

$$\hat{\mathbf{f}}_w(x, y) = \mathbf{f} * I(x, y) = \{ f_1 * I,\ f_2 * I, \ldots,\ f_N * I \}. \tag{4.1}$$

We now interpret the response of a filter bank at a certain image plane location (x, y) as an abstract description of its local gray level structure. The filter bank responses describe the local gray-level structures in the image simultaneously in orientation, scale and location.

Figure 4.2 shows the amplitude and phase responses of oriented filters at two different frequencies (scales) and two different orientations (vertical and horizontal), using a human face image as a test image. The filters respond to structures which are different in both orientation and size. The vertical filter at frequency $\pi/4$ responds most to the edges of the head, while the horizontal filter at the same frequency gives largest response at the horizontal line of the mouth. The vertical filter at frequency $\pi/16$ gives a response maximum at the bridge of the nose, while the horizontal filter responds most to the dark and thick eyebrows. Phase responses of the filters vary at a rate determined largely by the center frequency, and in an orientation which corresponds to the orientation of the filter. In addition to the smooth near-linear variation, the phase responses also have bifurcation points with no well-defined phase especially at regions where the amplitude response is low.

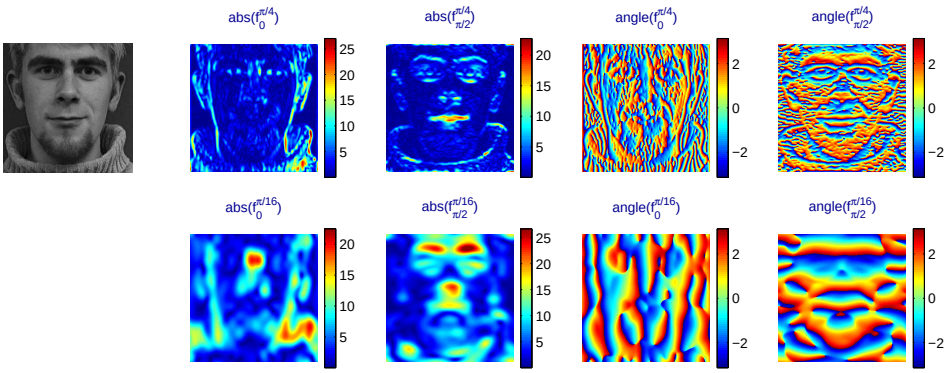


Figure 4.2: A test image and the amplitude and phase responses of filters at two different scales ($\pi/4$ and $\pi/16$) and two different orientations (0 and $\pi/2$).

Let us now define a quadrature filter bank response vector, a *filter jet* (Lades et al., 1993), by stacking the complex-valued filter responses $\mathcal{J}_j = (f_j * I)(x, y) \in \mathbf{C}$ at an image location (x, y) into a single long vector $\mathcal{J} \in \mathbf{C}^N$. N is the total number of different filters in the bank, for example if we use three different scales (center frequencies) and six different orientations, we have $N = 18$. An alternative bookkeeping scheme has been proposed in (Kyrki et al., 2004), where the filter responses are organized in a matrix instead of a vector. This has the advantage that cyclic shifts of the matrix correspond directly to appropriate changes in the filter parameters.

In order to make the filter responses at different locations in a single image and between two different images comparable with each other, it is a common procedure to normalize the filter responses by dividing the filter jet with its norm, $J = \mathcal{J}/||\mathcal{J}||$, so that the vector J has unit length. This normalization causes the complex-valued responses to have a maximum absolute value of 1.

Figure 4.3 shows the normalized filter responses of a synthetic image in a single frequency scale at a set of manually selected points, and illustrates how the filter responses of different gray-level structures occupy different regions in the four-dimensional unit ball of normalized filter responses.

The filter responses at all image locations have been plotted in black, and they fill the unit disk in the complex plane quite evenly, with slightly higher density both near the origin and close to the edge of the disk. The filter responses recorded at different image locations correspond to different regions inside the unit disk.

Denoted with green, the locations in the synthetic image with a horizontally oriented edge cause the horizontally oriented filter $f_{\pi/2}$ to respond with its antisymmetric imaginary part, giving a negative response due to the transition from dark to white. The response of the vertically oriented filter f_0 is almost zero. Both the vertical and horizontal filters respond with their imaginary antisymmetric

parts to the upper left hand corner of the light rectangle, denoted with the red cross.

While the filter responses at different features often coincide at a single orientation, they are well separated from each other in the filter response space if we consider the responses at all orientations simultaneously.

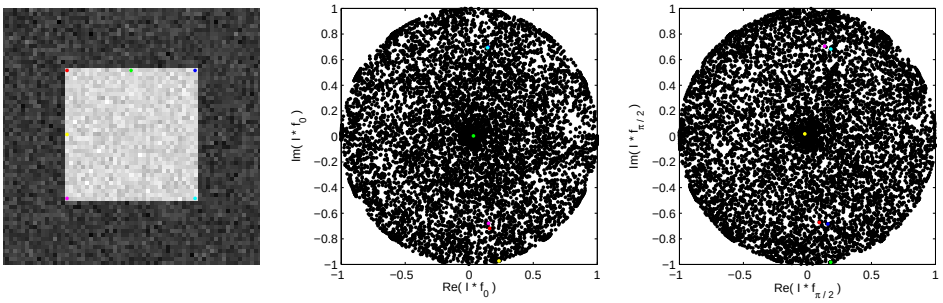


Figure 4.3: Distribution of normalized filter responses in a synthetic image. Responses corresponding to manually annotated locations are shown in color and responses from all image locations are shown in black. See text for discussion.

4.3 Similarity between filter bank responses

In order to compare, detect and classify local features we can define a *similarity function* which assigns a single numerical value for the discrepancy between two vectors of filter responses.

Lades et al. (1993) were the first to present a similarity measure between two filter bank responses, using the magnitude values of the complex-valued filters. Given two filter jets $\mathcal{J}$ and $\mathcal{J}'$ and denoting vectors which have the magnitude values as components with $\mathbf{a} = |\mathcal{J}|$ and $\mathbf{a}' = |\mathcal{J}'|$, the similarity between the vectors is defined as

$$S_a(\mathcal{J}, \mathcal{J}') = \frac{\sum_{j=1}^{N} a_j a'_j}{\sqrt{\sum_{j=1}^{N} a_j^2 \sum_{j=1}^{N} a_j'^2}} = \frac{\mathbf{a}^T \mathbf{a}'}{||\mathbf{a}||||\mathbf{a}'||}, \tag{4.2}$$

in other words, the inner product of vectors which have been normalized to unit length. The normalization of amplitudes achieves invariance to absolute brightness, and also makes the measure less sensitive to changes in contrast. However, the use of only absolute values of the filter responses ignores all phase information.

The use of only amplitude information causes the similarity measure to vary very smoothly, which is a good quality for optimization. However, the

ability to differentiate between features is compromised when ignoring the phase information. For example line and edge features which are in the same orientation differ only by their phase, and the similarity measure S_a cannot tell the difference between them. The measure is also invariant to the polarity of the image, in other words, an edge with a transition from dark to bright is equal to one with a transition from bright to dark.

A phase-sensitive similarity measure, using not only amplitude but also the phase responses of the filters, was presented in (Wiskott et al., 1999). The similarity measure is defined as

$$S_b(\mathcal{J}, \mathcal{J}') = \frac{\sum_{j=1}^{N} a_j a_j' \cos(\arg(\mathcal{J}_j) - \arg(\mathcal{J}_j') - \vec{d}^T \vec{k}_j)}{\sqrt{\sum_{j=1}^{N} a_j^2 \sum_{j=1}^{N} a_j'^2}}, \tag{4.3}$$

where $\vec{d}$ is the displacement vector and $\vec{k}$ is the wave vector of the filter. In addition to using the phase information, Wiskott et al also estimate the optimal displacement $\vec{d}$ which minimizes the phase difference between the jets.

The motivation for the optimization of the displacement vector $\vec{d}$ is that phase varies spatially very quickly especially in high frequencies. With zero displacement, similarity values are high only very near the maxima of the similarity field. Minimization of the phase difference between the jets widens the peaks in the similarity functions and thus broadens the basins of attraction near the correct optima in local optimization methods (Wiskott et al., 1999). Unfortunately it also causes the similarity fields to be much less smooth and contain discontinuities, which is problematic for most optimization methods.

4.4 Likelihood function

Next we wish to define a probability measure which tells how likely it is that the filter jet $\mathcal{J}$ represents the same feature as a reference jet $\mathcal{J}'$. In order to simplify the notation, define the two vectors $J = \mathcal{J}/||\mathcal{J}||$ and $J' = \mathcal{J}'/||\mathcal{J}'||$ normalized to unit length. For such vectors, we can interpret the square of the L2-norm distance, multiplied by minus one half, as a similarity measure, and it is equal to S_b up to an additive constant, since

$$\begin{aligned} -\frac{1}{2}||J - J'||^2 &= -\frac{1}{2}(J - J')^H (J - J') \\ &= -\frac{1}{2}\left(J^H J - 2Re\left\{J^H J'\right\} + J'^H J'\right) \\ &= -\left(1 - Re\left\{J^H J'\right\}\right) \\ &= \sum_j |J_j||J_j'| \cos\left(\arg(J_j) - \arg(J_j')\right) - 1. \end{aligned} \tag{4.4}$$

Now it holds that $-\frac{1}{2}||J - J'||^2 = S_b(J, J') - 1$.

Supposing that the difference between the two filter jets is approximately a Gaussian distribution with a diagonal covariance matrix $\beta^{-1}I$, the feature likelihood function has the form

$$\begin{aligned} p(J|J') &\propto e^{-\frac{\beta}{2}||J-J'||^2} = e^{-\beta\left(1-Re\left\{J^H J'\right\}\right)} \\ &\propto e^{\beta Re\left\{J^H J'\right\}}. \end{aligned} \tag{4.5}$$

The scalar parameter $\beta > 0$ affects the steepness of the likelihood function. The likelihood function is however strictly not a Gaussian function with respect to the unnormalized filter responses, since the value range of $Re\left\{J^H J'\right\}$ is limited to the interval $[-1, 1]$, which causes the tails of the distribution to be truncated. Our likelihood has only been given in an unnormalized form, that is, up to an unknown normalizing constant which makes it a proper probability distribution. The form of the likelihood function also assumes that the responses of the filters are independent, which is not true as they have been computed partially from the same image pixels. Williams (2005) shows how such measurements can be handled theoretically. In Section 5.2 we will consider the effects of the correlated measurements to the similarity values.

Figure 4.4 illustrates how the probability mass concentrates to the largest mode when the parameter β is increased and the distributions are normalized to unity mass. The three probability distributions have been derived from the same energy function (or similarity measure). With low values of β the distribution is almost uniform, but when β is large, most of the probability mass is contained in the largest mode of the distribution. On the other hand, with larger values of β, the different modes of the distribution become increasingly separated, with a large region of low probability between them.

Following the same idea, we can take *any* similarity function S which is reasonably close to being a L2-norm distance and compute the likelihood $\exp\left(\frac{\beta}{2}S\right)$, which will be close to being a Gaussian distribution.

It should be noted here that as we compute the similarity and likelihood functions, we implicitly assume that the elements of the feature vectors have more or less comparable statistics. Natural images have typically a decreasing energy distribution of approximately $1/f^2$ (Ruderman and Bialek, 1994),(Field, 1987). Due to this phenomenon, there is considerably less energy in the high frequencies. This causes filters which are small in the spatial domain and correspond to high frequencies to have smaller amplitudes in their response, and the assumption above does not hold.

The problem can be solved on the filter level by scaling the filter outputs with squares of their center frequencies so that filters with different frequencies have approximately equal variance (Lades et al., 1993). In our probabilistic framework this means that the similarity measure becomes a Mahalanobis distance $J^H C^{-1} J'$

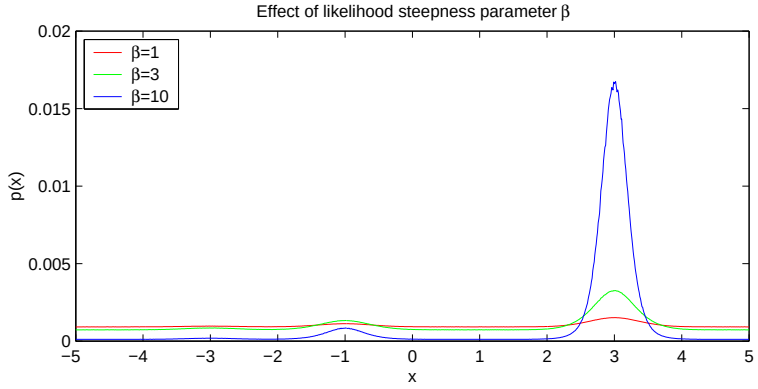


Figure 4.4: Three likelihood distributions generated from the same similarity function with different values of steepness parameter β.

with a diagonal covariance matrix C where the elements are defined by the center frequencies of the filters. Supposing that we have for example filters with two orientations and three scales $f_c = \left\{\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}\right\}$ in the filter bank, we would choose

$$C = \frac{4}{\pi}\text{diag}\left(\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{16}\right) = \text{diag}(1, 1, 1/2, 1/2, 1/4, 1/4). \quad (4.6)$$

The resulting likelihood function is

$$p(J\,|\,J') \propto e^{\beta Re\{J^H C^{-1} J'\}}. \quad (4.7)$$

In general the scaling constants in the diagonal of C^{-1} could be also learned from data, instead of setting them to be proportional to f^2.

As an example of the shape of the likelihood functions, Figure 4.5 shows three test images[1] and their likelihood fields with a filter jet which has been taken at a single location in the same images. Each of the bumps in the sewer grating in the leftmost image produces a clean maximum in the likelihood function, and there are no significant false maxima. Bumps which are in the different orientation than the reference feature are however not at all similar to the reference feature, as measured by our likelihood function.

In the centermost image, the test jet is taken inside one of the nuts, and all other nuts in the same orientation produce a maximum in the likelihood function. In addition, there are some spurious local maxima elsewhere in the image. In the rightmost image, the test jet is located at a petal of the flower. Only some of the other petals in the same orientation contain a likelihood maximum, and there are several false maxima in the background.

[1]The images appear courtesy of http://www.adigitaldreamer.com/

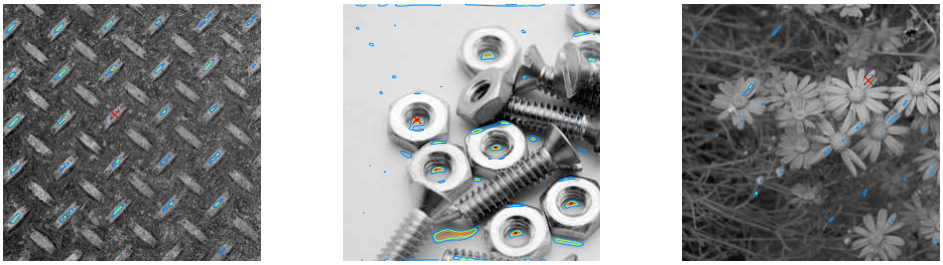


Figure 4.5: Likelihood functions of three natural images, with the reference jet taken from a single location at each image.

It can be concluded that while the single feature likelihood function is able to quantify the relationships of local gray-level structures, it is not by itself a sufficient solution for recognizing any but the simplest objects in a visual scene.

4.5 Rotation invariant feature similarity

4.5.1 Motivation

One of the properties of the previously defined similarity measures and the likelihood functions derived from them is that in general only image features which are in the same orientation are considered similar. There are however situations in which it would be useful to have a similarity measure which is invariant to rotation, but would not have to use rotation-invariant features. Figure 4.6 shows face images and the corresponding similarity fields of a mouth corner feature. The similarity field in Fig. 4.6a) has a maximum at the approximately correct location, as well as a another local maximum at the left nostril. Fig. 4.6b) has been rotated twenty degrees, which causes the maximum at the mouth corner to diminish because the orientation of the feature is not correct. The other maximum at the nostril withstands rotation better, and becomes the strongest maximum in the image. However, rotation invariance of individual features can be useful even when the object as a whole is in the same orientation. In Fig. 4.6c) the corners of the mouth point downward due to individual variation, and the maximum of the similarity is in an incorrect position at the mouth line.

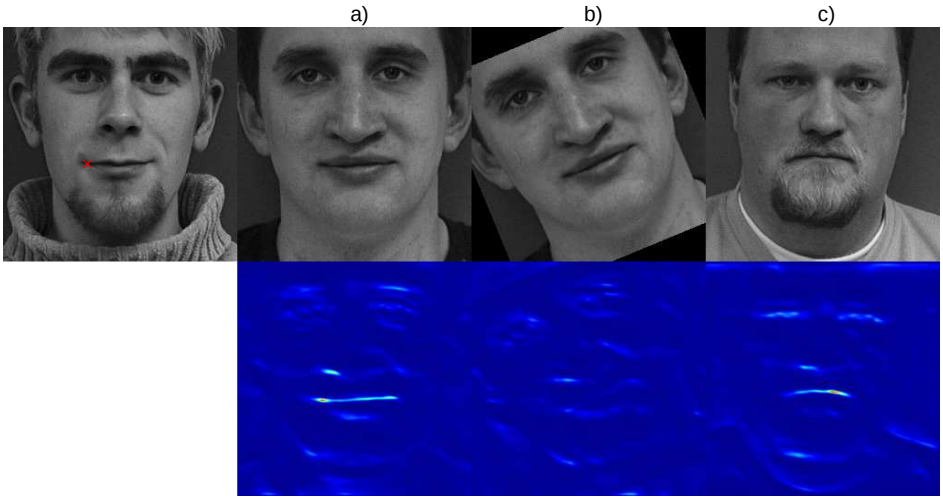


Figure 4.6: Face images and the similarity fields of a mouth corner feature. The reference jet has bee taken at the location denoted by the red cross in the leftmost image.

4.5.2 Rotation invariant similarity measures

Recall the phase sensitive similarity function (Wiskott et al., 1999)

$$S_1(\mathcal{J}^{(1)}, \mathcal{J}^{(2)}) = \frac{Re\left\{\mathcal{J}^{(1)H}\mathcal{J}^{(2)}\right\}}{||\mathcal{J}^{(1)}||||\mathcal{J}^{(2)}||}. \tag{4.8}$$

In order to extend the similarity measure to be rotation invariant, we can compute the inner product in all $2N$ relative discrete rotation angles of the filter jets and choose the largest (Ng et al., 2005), using

$$S_2 = \max_{i=-N,\ldots,N-1} S_1\left(\text{Shift}\left(\mathcal{J}^{(1)}, i\right), \mathcal{J}^{(2)}\right), \tag{4.9}$$

where $\text{Shift}(\mathcal{J}, i)$ means an operation where the components of $\mathcal{J}$ have been shifted i index locations, and complex conjugated when they wrap around to the beginning. In component form this is

$$S_2 = \frac{1}{||\mathcal{J}^{(1)}||||\mathcal{J}^{(2)}||}\left(\max_{i=-N,\ldots,N-1} \sum_{k=0}^{N-1} Re(j_{k-i}^{(1)} j_k^{(2)})\right). \tag{4.10}$$

Here a negative index filter $-i$ is interpreted as complex conjugate filter of the positive index $N - i$ (this is because the response of the Gabor filter has equal amplitude and opposite phase when a 180 degree rotation is performed) and

indices $N + i, i \geq 0$ wrap around to $-N + i$. Apart from the rather cumbersome indexing scheme, this is a straightforward, but computationally $2N$ times more demanding procedure compared to S_1.

Now, an extension of the previous discrete scheme to continuous case is proposed. Let us expand the discrete filter response $\mathcal{J}^{(1)}$ into a continuous one with rotation angle θ using the steering coefficients $\hat{k}_i(\theta)$, so that the jet $\mathcal{J}^{(1)}$ consists of filter responses $j_k^{(1)}(\theta) = \sum_i \hat{k}_i(\theta) j_{i+k}^{(1)}$. Now we can compute the similarity between $\mathcal{J}^{(1)}$ and $\mathcal{J}^{(2)}$ in any *continuous* relative orientation angle and choose the largest,

$$S_3 = \max_\theta S_1 \left(\text{Steer} \left(\mathcal{J}^{(1)}, \theta \right), \mathcal{J}^{(2)} \right), \tag{4.11}$$

where Steer $(\mathcal{J}, \theta)$ is the steering operation with a rotation angle θ, in component form

$$S_3 = \max_\theta \frac{1}{||J^{(1)}||||J^{(2)}||} \sum_k Re \left\{ \left(\sum_i \hat{k}_i(\theta) j_{i+k}^{(1)} \right) j_k^{(2)} \right\}. \tag{4.12}$$

The norm of the steered jet $J^{(1)}$ is preserved exactly only if the steering is exact. Approximate steering causes slight variation in the norm, but normalization may still be performed only after maximization to lessen the computational cost, if the maximum steering error is small. The computational cost of the steerable similarity measure is dependent on how the optimization of the relative rotation angle between the filter jets is performed. A simple exhaustive search in a dense grid increases the computational burden even more compared to the similarity function S_2.

Alternatively, the optimization of the steering angle can be performed only using local optimization. This is a similar idea to what Wiskott et al. (1999) used in estimating the displacement of filter jets: only now, we would try to estimate the relative rotation of the filter jets.

In Chapter 7, where the similarity functions introduced here will come into practical use, we will employ the similarity function S_1 and a variation of the similarity function S_3 without the maximization of orientation in this stage, described below in Section 4.7.2.

We illustrate the behavior of the different similarity functions with two example images. Fig. 4.7 shows the likelihood fields (derived from the similarity measures) of a corner feature (marked with a white cross) in a synthetic test image, computed using Eqs. (4.8), (4.10) and (4.12). A filter bank with four orientations is used, with shape parameters $\sigma = [2.5 \;\; 1.75]$.

The similarity function S_1 provides best localization of the correct feature, but withstands only small rotations. The similarity function S_2 has generally an unequal response with respect to rotation, and only orientations which are present in the filter bank provide the correct response. Orientations between those in the

bank have their similarity maxima split in two and shifted away from the correct feature location. The similarity function S_3 has equal response in all orientation angles. However, as a consequence of rotation invariance, feature specificity is lower than with the two other measures, and the localization capability is worse.

The normalization factor $\frac{1}{||J^{(1)}||||J^{(2)}||}$, which is present in all three similarity measures, provides contrast invariance and causes medium similarity values to be found also among background noise.

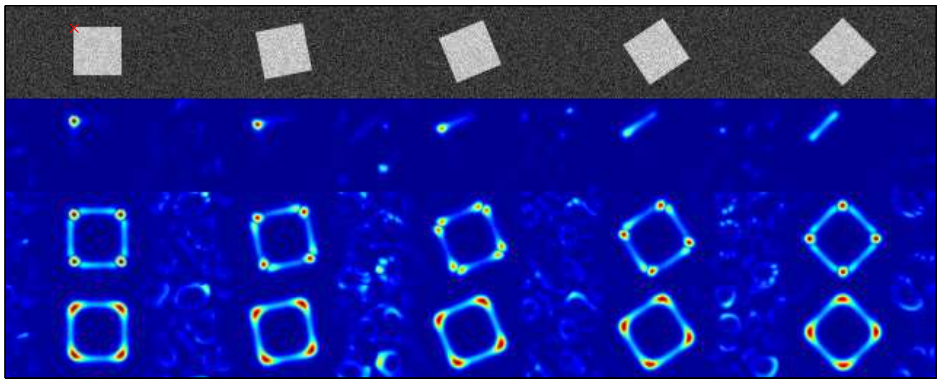


Figure 4.7: Behavior of three different similarity functions with a synthetic test image (top row), with the test feature marked with a red cross. Second row: Normalized inner product similarity S_1. Third row: Discrete angle rotation invariant similarity S_2. Bottom row: Continuous angle rotation invariant similarity S_3.

Fig. (4.8) shows the likelihood values evaluated at the correct feature location. It can be seen that while the likelihood peak fades away without any rotation invariance and drops quite low with discrete angle rotation invariant measure S_2, it is remarkably stable with the continuous angle rotation invariant measure S_3. The likelihood functions are not highly sensitive to the filter shape parameter values, and while the filters used in this example do not retain the shape of their impulse responses very well under steering (error is approximately 6%), there is hardly any noticeable variation due to rotation in the likelihood field of the function S_3, despite the steering error.

Fig. (4.9) shows the likelihood fields of S_1, S_2 and S_3 of real-world face images. Here, eight filters with shape parameters $\sigma = [2\ \ 4]$ were used. The reference Gabor jet feature was obtained from the mouth corner of the leftmost face, marked with a red circle. The five test images have been rotated 13 degrees, and the manually annotated feature locations are marked with circles in their similarity fields. Eight basis filters are useful in making the features specific so that the mouth corners are recognized well, but without rotation invariance even small rotations cause the similarity to drop drastically, and detection is not possible. Only image 2 has a good maximum at the correct location, and in images

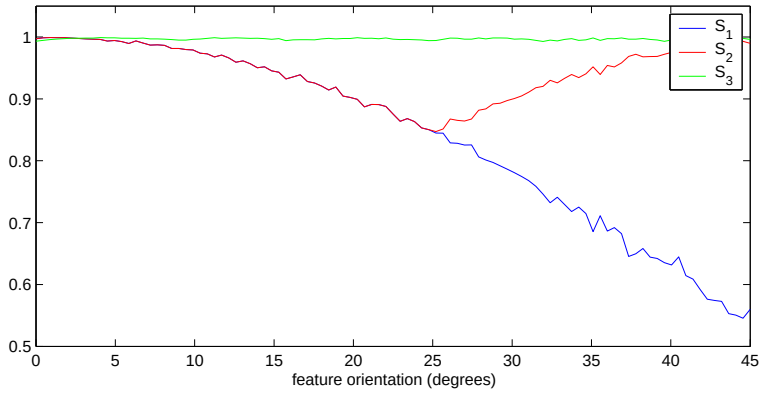


Figure 4.8: Likelihood values evaluated at the correct feature location in different feature orientations.

1, 4 and 5 the annotated locations have much lower probability than incorrect maxima elsewhere in the image. In contrast, similarity functions S_2 and S_3 provide very good feature localization, with a clear maximum near the mouth corner in all test images. In images 2 and 3 the maximum probability is not exactly at the annotated location, but this caused by the manual annotation being away from the mouth corner location in the gray-level image. The differences between the performance of S_2 and S_3 are masked by the large variation occurring in natural images. The similarity measures alone do not suffice in solving the face alignment problem, and the ambiguities caused by feature variability have to be resolved by including the information in the relative locations of the detected features.

We should note here that the rotation invariant feature similarity functions are not necessarily desirable for local feature based object modeling, which is the main topic of this work, as it may be better to handle the optimization of orientation in the object level. Instead, the similarity function S_3 may find use in other applications such as rotation invariant texture classification using Gabor-type filters, as steerability allows efficient optimization of orientation without recomputing the filter responses.

4.6 Orientation analysis with feature similarity

Full rotation invariance is not always a desirable property of a similarity measure. Its main drawback is that the detected features become unnecessarily generic, especially if it is known in advance that the features cannot appear in any possible orientations.

The probabilistic formulation of the feature likelihood function allows us to incorporate additional information into the feature matching scheme in a

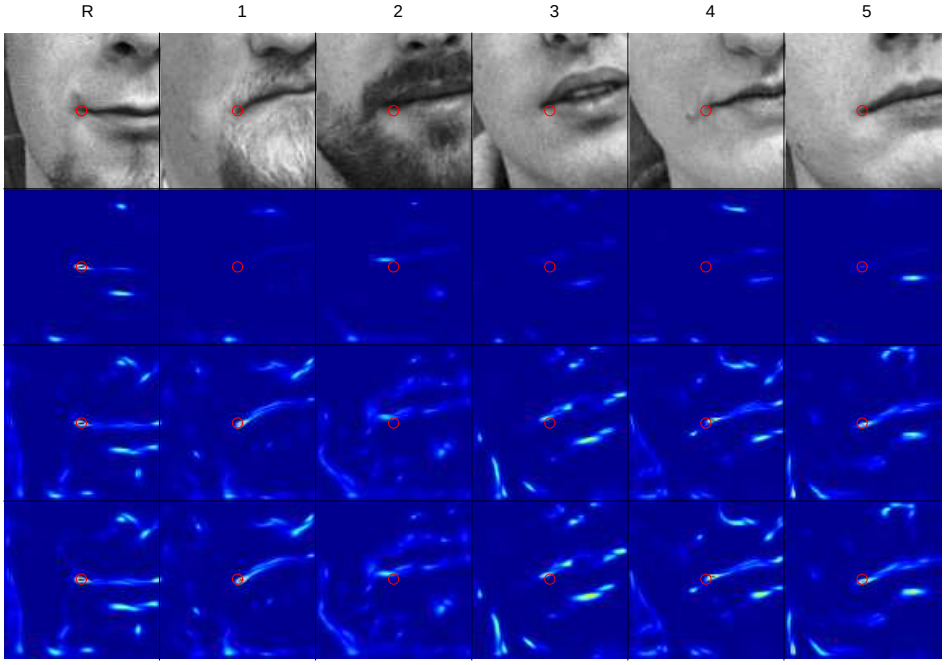


Figure 4.9: Similarity of a mouth corner feature, marked with a red circle, between the reference image R and five independent rotated test images (1-5). Rows from top to bottom: original images; Normalized inner product similarity S_1; Discrete angle rotation invariant similarity S_2; Continuous angle rotation invariant similarity S_3. Even a small plane rotation in the image necessitates rotation invariance because the filters are very orientation-specific.

theoretically consistent way. Instead of maximizing the similarity with respect to orientation, we can consider the likelihood function of a feature jet J, possibly in any orientation θ and a reference feature jet J',

$$p(J(\theta)|J',\theta) \propto \exp\left(S(J(\theta),J')\right) \tag{4.13}$$

and multiply it with a prior probability of the orientation angle $p(\theta)$, obtaining the joint posterior probability of the filter jet J and the orientation angle θ,

$$p(J(\theta),\theta|J') \propto p(J(\theta)|J',\theta)\,p(\theta). \tag{4.14}$$

The role of the prior is to make the likelihood less ambiguous by ruling out orientations which are not considered probable to begin with. While it may seem excessive to first compute the likelihood at all orientation angles and then ignore some of them in further analysis, this is the theoretically correct way to make inferences about the orientation angle in the Bayesian framework. Practical

engineering solutions may choose to skip a step in between and rule out some of the improbable orientations already when computing the likelihood because of performance reasons. The Bayesian framework is especially useful when all orientation angles can occur, but it is known in advance that some are more common than others.

4.7 From features to objects

Once we have defined how to measure the probability of two filter jets representing the same feature, we will proceed to construct a probability measure for objects which contain several feature points. In our framework objects are modeled as collections of filter bank responses at specific image locations, which are connected by a graph structure, similarly to the object model proposed in (Lades et al., 1993). This makes is possible to evaluate the probability of a complete object consisting of local features. Our probability model is mainly based on the likelihood model of local features.

4.7.1 Object likelihood models

Denote the locations of features in the image plane with $\mathbf{x}$. Assuming that the filter responses from jets at different locations are statistically independent, we can write the joint likelihood of all observed filter responses $\mathbf{J}$ at locations $\mathbf{x}$ as a product of the independent feature likelihoods,

$$\begin{aligned} p(\mathbf{J}|\mathbf{J}', \mathbf{x}) &= \prod_i p(J^{(i)}|J'^{(i)}, x_i) \\ &\propto \prod_i \exp\left(\beta S\left(J^{(i)}(x_i),\, J'^{(i)}\right)\right) \\ &= \exp\left(\beta \sum_i S\left(J^{(i)}(x_i),\, J'^{(i)}\right)\right). \end{aligned} \tag{4.15}$$

The likelihood of the object with a feature configuration $\mathbf{x}$ is thus simply the sum of the individual similarity scores, multiplied by the constant β and exponentiated.

The feature locations $\mathbf{x}$ can be themselves parameterized via an object geometry model M. The probabilistic inference is then performed on the parameters of the model M. The parameterization of M may contain for example pose parameters $\boldsymbol{\theta}$, so that the location of feature i in the image plane is given by $x_i = M_i(\boldsymbol{\theta})$. The resulting likelihood function is then given by

$$p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M) \propto \exp\left(\beta \sum_i S\left(J^{(i)}(M_i(\boldsymbol{\theta})),\, J'^{(i)}\right)\right). \tag{4.16}$$

In the case when steerability is taken into account, the model jet $J^{(i)}$ is dependent also directly on the pose $\boldsymbol{\theta}$, and the likelihood becomes

$$p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M) \propto \exp\left(\beta \sum_i S\left(J^{(i)}(M_i(\boldsymbol{\theta}), \boldsymbol{\theta}),\, J'^{(i)}\right)\right). \tag{4.17}$$

The likelihood functions are again only given in the unnormalized form. The scalar normalization constant which multiplies the unnormalized likelihood function and scales it so that it is a proper probability distribution with a total mass of unity, is very hard to compute, as it can be computed only by integrating over all possible configurations of all possible feature locations.

Sullivan et al. (2001) note that a single object model does not suffice as an adequate probabilistic account of all image data. In principle the background should also be modeled, so that we can compute the likelihood function of the whole image. Constructing a probabilistic model for generic backgrounds is however plausible only if the object models are also quite simple, such as the ones considered in (Sullivan et al., 2001). It would appear almost impossible to construct a probabilistic background model with the same level of complexity as the ones typically employed in human face recognition, for example. Like Tamminen (2005) notes, if the competing models do not describe the data with the same accuracy, we always have to make the decision between explaining the data with a complex, but accurate model or a simple, but inaccurate model, and it is not straightforward to compare the probabilities of the competing models, because the normalization factors of the likelihood functions are not known.

4.7.2 Posterior analysis

In a Bayesian framework, inference is performed on the posterior distribution of the random variables of interest. In an object recognition problem these variables can include for example object pose $\boldsymbol{\theta}$. Using the Bayes' theorem, the posterior distribution of the pose is formally obtained as

$$p(\boldsymbol{\theta}|\mathbf{J}, \mathbf{J}', M) = \frac{p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M)\, p(\boldsymbol{\theta}|\mathbf{J}', M)}{\int_{\boldsymbol{\theta}} p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M)\, p(\boldsymbol{\theta}|\mathbf{J}', M) d\boldsymbol{\theta}} \propto p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M)\, p(\boldsymbol{\theta}|\mathbf{J}', M) \tag{4.18}$$

The power of Bayesian analysis often stems from the fact that it is possible to write the posterior distribution in such a form that effective prior probability distributions can be constructed. In our case, this information would be included in the prior distribution $p(\boldsymbol{\theta}|\mathbf{J}', M)$. However, it is not obvious what kind on information should be built into the prior distribution, and for the purposes of Chapter 7, we will simply choose a flat prior for most elements of $\boldsymbol{\theta}$. Because of the flat priors, our apporach is not very different from Maximum likelihood

analysis, and the focus in Chapter 7 is in the complex shape of the likelihood distribution $p(\mathbf{J}|\mathbf{J}', \boldsymbol{\theta}, M)$ which largely determines the shape of the posterior distribution.

4.7.3 Practical implementation

Specifically, in Chapter 7 we consider two parameterized rigid object models M. In Section 7.2 which considers rotations in the image plane, the parameterization of M consists of planar rotation θ, global scale s and object center x_c, y_c in the image plane. In absence of strong prior models, we choose a flat prior for the term $p(\boldsymbol{\theta}|\mathbf{J}', M)$, and the posterior distribution is directly proportional to the likelihood function.

In Section 7.5 which considers rotations in depth, the parameterization of M consists of three rotation angles θ, ϕ and ψ as described in Section 6.3, global scale s and object center x_c, y_c in the image plane. The only informative prior probability term here is the choice $\phi \sim \cos(\phi)$ for the elevation angle ϕ. The justification for this choice is discussed in Section 6.3.

The rotation invariant feature similarity measures presented in Section 4.5 include optimization of the pose angle. This can be done already in the feature level for each feature separately, but since all features of a rigid object share the same orientation, it is typically more appropriate to estimate (or optimize) the pose of the object as a whole. Instead of maximizing the similarity of each feature in the measure S_3, the approach we will use in Section 7.4 is to define an orientation-dependent similarity function

$$S(\theta) = \frac{1}{||J^{(1)}||||J^{(2)}||} \sum_k Re\left\{ \left(\sum_i \hat{k}_i(\theta) j^{(1)}_{i+k} \right) j^{(2)}_k \right\}. \qquad (4.19)$$

and handle the optimization of planar rotation θ in the random sampling stage by finding the largest mode of the posterior distribution of θ among other model parameters.

4.8 Monte Carlo sampling algorithms

Regardless of their exact formulation, image likelihood functions are typically multimodal, having many maxima. Consequently, local optimization methods require good initialization heuristics in order to find the best maximum, and are otherwise prone to getting stuck in poor solutions, never finding the stronger maxima. In order to reliably find good solutions, global methods are required. In this work we will choose to employ random sampling methods in order to explore the likelihood and posterior probability distribution functions. Next we will briefly review some Monte Carlo sampling algorithms, which can be used to

obtain samples from the distributions even when the distributions are given only in unnormalized form.

The aim of a Monte Carlo sampling algorithm is to produce random samples which follow some given target distribution $p(x)$. From the samples x_i we can compute estimates such as the expected value $E[]$ of some quantity f by

$$E[f(x)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i). \tag{4.20}$$

The sampling methods we employ here require the target distribution to be known in the unnormalized form $p(x) = \frac{1}{Z} p^*(x)$, as the normalization constant Z cancels out in the computations. This fact makes the Monte Carlo methods very useful, as the computation of the normalization constant can be very hard or even impossible in practice because of the high dimensionality of the integrals.

4.8.1 Metropolis sampling

The classical Metropolis sampling algorithm, originally devised for problems in computational physics (Metropolis et al., 1953), uses an acceptance/rejection rule to converge to the specific target distribution (Gelman et al., 2003). Metropolis sampling is very straightforward to implement and requires only that the values of the function defining the target distribution can be computed.

First, the algorithm is initialized by sampling an initial state $\mathbf{x}^0$ from a *starting distribution* $p_0(\mathbf{x})$, for which $p_0(\mathbf{x}^0) > 0$.

In order to move from one state to another, we sample a candidate state $\mathbf{x}^*$ from a *jumping distribution* J, which can be chosen freely as long as it holds that $J_t(\mathbf{x}_a|\mathbf{x}_b) = J_t(\mathbf{x}_b|\mathbf{x}_a)$, that is, the probability to jump from state a to state b is the same as the probability to jump from state b to state a.

We then accept the candidate state $\mathbf{x}^*$ as the new state $\mathbf{x}^t$ with probability

$$p = \min\left(1, \frac{\pi(\mathbf{x}^*)}{\pi(\mathbf{x}^{t-1})}\right). \tag{4.21}$$

This rule means that we always move to the new state if it is more probable than the previous state. In addition we move to the new state also occasionally when it is less probable, with a probability given by the ratio of the probabilities of the two states. Otherwise we remain in the same state, and $\mathbf{x}^t = \mathbf{x}^{t-1}$.

Repeating the jumping procedure over and over again and updating the current state according to Eq. 4.21, we obtain the sequence $\mathbf{x}^1, \mathbf{x}^2, \ldots$, a random walk in the parameter space, which converges to the target distribution $p(\cdot)$. Typically some amount of samples from the beginning of the sequence are discarded in order to eliminate the bias due to the choice of the initial state $\mathbf{x}^0$.

Metropolis-Hastings algorithm is a generalization of the previous procedure that removes the requirement that the jumping distribution must be symmetric and modifies the acceptance probability rule accordingly to

$$p = \min\left(1, \frac{\pi(\mathbf{x}^*)/J_t(\mathbf{x}^*|\mathbf{x}^{t-1})}{\pi(\mathbf{x}^{t-1})/J_t(\mathbf{x}^{t-1}|\mathbf{x}^*)}\right) \tag{4.22}$$

in order to account for the asymmetry in the direction of the jumping probabilities.

The components of the parameter vector $\mathbf{x}$ can be even updated one by one, accepting the proposals according to

$$p = \min\left(1, \frac{\pi(\mathbf{x}_j^*|\mathbf{x}_{\setminus\{j\}}^{t-1})/J_t(\mathbf{x}_j^*|\mathbf{x}_j^{t-1}, \mathbf{x}_{\setminus\{j\}}^{t-1})}{\pi(\mathbf{x}_j^{t-1}|\mathbf{x}_{\setminus\{j\}}^{t-1})/J_t(\mathbf{x}_j^{t-1}|\mathbf{x}_j^*, \mathbf{x}_{\setminus\{j\}}^{t-1})}\right), \tag{4.23}$$

where $\setminus\{j\}$ denotes all components except j. This is the *single-component Metropolis-Hastings* algorithm, where each of the parameter components has its own jumping distribution, which can depend on the current values of all components.

4.8.2 Gibbs sampling

Gibbs sampling can be interpreted as a special case of single-component Metropolis-Hastings algorithm, where we choose the jumping distributions to be the full conditional distributions of the parameter components,

$$J_t(\mathbf{x}_j^*|\mathbf{x}_j^{t-1}, \mathbf{x}_{\setminus\{j\}}^{t-1}) \equiv \pi(\mathbf{x}_j^*|\mathbf{x}_{\setminus\{j\}}^{t-1}). \tag{4.24}$$

This results in the acceptance probability being always equal to 1 (Gelman et al., 2003). Whereas Metropolis and Metropolis-Hastings algorithms only require that we can compute the joint probability of any parameter values $\mathbf{x}$, in Gibbs sampling we need to be able to compute the full conditional distributions of the parameters, which can be significantly more difficult to obtain in analytical form. When this can be done, Gibbs sampling can be very efficient.

If the analytical expressions for the conditional distributions of parameters are impossible to obtain, one can resort to numerically evaluating the joint distribution $\pi(\mathbf{x})$ along the line $\mathbf{x}_j \in \mathrm{R}$ (or some smaller subset of possible values of $\mathbf{x}_j$), keeping the other parameters $\mathbf{x}_{\setminus\{j\}}$ fixed, and drawing a random sample from this empirical full conditional distribution using the inverse-CDF method (Gentle, 1998).

4.8.3 Population Monte Carlo sampling

Population-based Monte Carlo methods borrow ideas from several sources to produce a Monte Carlo simulation algorithm which is not concentrated on generating a single sequence of samples but rather samples the target distribution in a parallel manner (Cappe et al., 2004). In some respects the PMC algorithms can be thought to have common ground with genetic optimization algorithms (Liu, 2001), as they sequentially generate a new population of particles (or samples) based on the previous generation, and the fitness of the members of the population is measured individually by evaluating the target function. However, unlike genetic optimization algorithms, which concentrate on finding the maximum of the target function, Population Monte Carlo sampling produces actual samples from the whole of the target distribution. The PMC algorithm is given in pseudocode in the following.

Algorithm 1 PMC SAMPLER

Require: Density $\pi(\cdot)$ to be simulated

for $t = 1..T$ **do**
  for $i = 1..N$ **do**
    Select the generating distribution $q_{it}(\cdot)$
    Generate $\mathbf{x}_i^t \sim q_{it}(\mathbf{x})$
    Compute $\rho_i^t = \pi(\mathbf{x}_i^t)/q_{it}(\mathbf{x}_i^t)$
  end for
  Normalize the ρ_i^t's to sum up to 1
  Resample N times $\mathbf{x}_i^t$'s with replacement, using the weights ρ_i^t, to create the sample $\{\mathbf{x}_1^t \ldots \mathbf{x}_N^t\}$
end for

A remarkable property of the PMC scheme is that the generating distributions $q_{it}(\cdot)$ can be chosen freely for each particle i at each generation t. This makes it possible to use heuristics which guide the sampler toward the modes of the target distribution $\pi(\cdot)$ while the samples themselves are still guaranteed to follow the target distribution. These theoretical results are however true only for systems with an infinite number of particles. In practice, only up to a few thousand particles are used, and the samples can become biased.

Like Metropolis and Gibbs sampling, also PMC sampling requires initialization. Initial distributions for the parameters must be chosen, and the particles in the first iteration are generated from these. The initial distributions can be significantly wider than the generating distributions at subsequent iterations, so that the PMC sampler first spreads the particles everywhere in the parameter space and subsequent iterations will concentrate the samples to regions which have significant amounts of probability mass.

Chapter 5

Numerical experiments with oriented filters

5.1 Introduction

In order to lessen the computational cost and memory storage requirements one would like to find a generic set of filters and use it for all inference on images. The problem with this approach is that the desirable filter properties can be conflicting: good steerability provides rotation invariance, but poses a limitation for the angular bandwidth and thus for the feature representation capability of the filters. Feature detection and recognition may benefit from different properties of the data, and we will attempt to answer the question whether the same bank of filters can be good for both feature localization and recognition. Further, we wish to systematically find good design parameters for the Gabor-type filter bank employed in (Wiskott et al., 1999) and (Tamminen, 2005), who have used rather different filter bank designs in a highly similar object matching tasks.

In Section 5.2 it is first shown how the similarity values produced by the filter jets remain usable for recognition even when using shape parameters for which steering performance is relatively poor. The design parameters of a Gabor filter bank are experimentally evaluated in Section 5.3 using a full object matching system, in order to find the best parameters for object localization and recognition. In Section 5.4 the best parameters of a similar filter bank using angular Gaussian filters are sought, and their performance is compared to the filter bank with Gabor filters. Section 5.5 compares the recognition performance of different filter families and their Gabor-type approximations and Section 5.6 considers the effect of the complexity of the feature models. We will concentrate on systems using oriented filters. Comparisons of the localization performance of the object matching system with respect to other approaches presented in the literature can be found in (Tamminen, 2005), and extensive comparisons of recognition

performance between different face recognition approaches using Gabor-type filters can be found in (Shen and Bai, 2006).

The recognition method we use in the experiments is a simplified version of the probabilistic object matching system presented in (Tamminen, 2005). Namely, in our likelihood function the feature models of an individual are based on a single example image, whereas Tamminen (2005) uses distribution modeling and Wiskott et al. (1999) multiple feature prototypes learned from several example images. The simplified recognition method we employ here realistic, but not state-of-the-art, and instead of aiming to optimize the filter bank in terms of recognition performance in a practical setting, the goal of the recognition experiments is to bring out the differences in the feature representation capability of the filter banks. The quality of the pairwise feature similarity studied this way is highly relevant also for methods which use multiple feature prototypes and nearest neighbor classification, as the distance measure is based on pairwise comparison.

5.2 Filter jets as approximations of continuous responses

In this section, it is investigated how the theoretical results in Chapter 3 concerning the steerability of a filter bank relate to the filter responses from natural images in a practical setting, and how the steering approximation affects the similarity values in natural images.

Filter banks can be considered as a collection of discrete samples from a continuous filter function, if we consider the filter parameters to take continuous values. The responses of the continuous filter functions at a given location in the image plane (filter jets) are then also continuous-valued functions, with filter parameters as their variables. However, while one can compute closed-form expressions for the continuous-valued filter jets at image features such at lines and step edges, these are more useful in theoretical considerations. In practical image analysis in which the images themselves are typically represented only as samples, a sampled representation of the filter function is appropriate, although the sampling may in general lose some information.

Having resorted to sampled representations, one would wish that the samples will provide a good representation of the continuous filter jet. Exact steerability and shiftability guarantee that the full continuous filter function (and consequently also the linear filter responses) can be exactly reconstructed from the discrete samples. For parameters which have unbounded range, such as translation and scale, this can obviously be possible only in some finite interval. The orientation parameter is different in this sense because it is limited to a finite interval. Since orientation is inherently periodic, the exact reconstruction condition is in this case related to the classical Nyquist sampling theorem which states that in order to

reconstruct the original continuous signal from samples, the sampling frequency must be greater than twice the input signal bandwidth. This requirement proposes a strong constraint to the shape of the possible exactly steerable filters. On the other hand, filter jets with a finite amount of Gabor filters always violate the sampling theorem because the angular frequency of Gabor filters is not band-limited in principle.

Steerability allows us to compute a continuous approximation of the filter response given the discrete basis filter responses, using Eq. 3.5 and keeping in mind that the even and odd filters require separate steering functions. With complex Gabor filters g_{θ_j} centered on the feature in the image I, we approximate the continuous convolution result with

$$\begin{aligned} I * g(\theta) &\approx \sum_{j=1}^{N} k_j^{Re}(\theta)\left(I * Re\left\{g_{\theta_j}\right\}\right) + ik_j^{Im}(\theta)\left(I * Im\left\{g_{\theta_j}\right\}\right) \\ &= \sum_{j=1}^{N} k_j^{Re}(\theta) Re\left\{J_j\right\} + ik_j^{Im}(\theta) Im\left\{J_j\right\}. \end{aligned} \tag{5.1}$$

In other words, the filter jets are directly weighted with the steering functions of the basis filters.

Fig. 5.1 shows the real and imaginary parts of the complex orientation responses at an eye corner feature with filters whose center frequency is $\pi/4$. At this scale, the eye corner is mostly a line feature, corresponding to a large negative response of the even filter (real part) at a diagonal orientation. The odd filter (imaginary part) responses are more varied for this feature, depending on the shape parameters, but all give some response to an approximately horizontal edge feature.

Because real and imaginary parts of the signal are modeled separately, the phase of the approximation is less accurate than its magnitude. However, inaccuracies in phase are large only when the magnitude of the signal is small. In general the approximations are perhaps even surprisingly good: even though a filter with the shape parameters $\sigma_x = 4, \sigma_y = 4$ is hardly steerable at all with four basis filters (maximum relative error in the impulse responses of the filters, measured with Eq. 3.10, is about 68%), the approximations are qualitatively quite similar. With eight basis filters and $\sigma_x = 4$, $\sigma_y = 4$, maximum relative error in the impulse response, computed with Eq. 3.10, is still 13%, but the filter amplitude response approximation is almost indistinguishable from the correct continuous response.

In general, Eq. 3.10 gives a worst-case estimate of the filter impulse response error, and the errors in the actual filter responses are much less severe. There appears to be no breakdown behavior in the filter response approximations, although the Gabor filters themselves lose their steerability rather quickly if

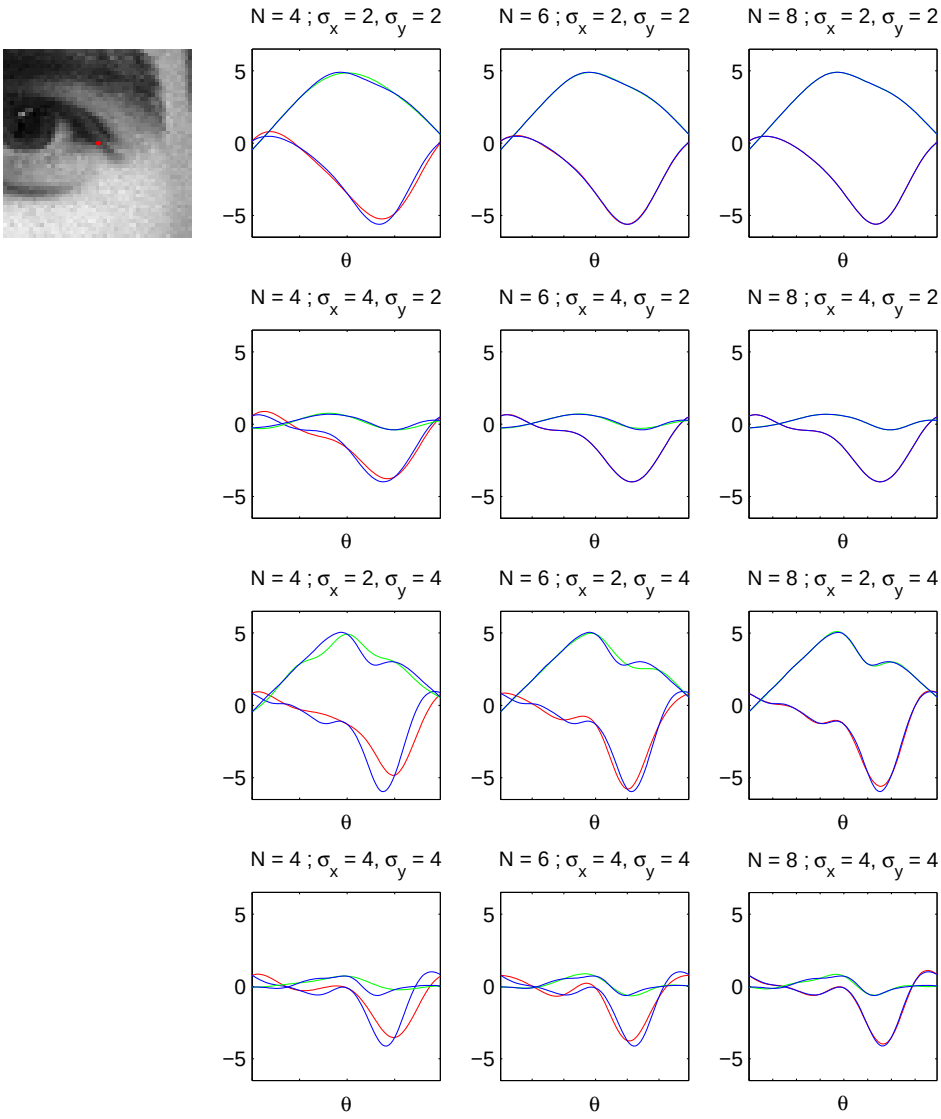


Figure 5.1: Steerable approximations of the real and imaginary parts (red and green line, respectively) of the continuous orientation response (blue line), with different number N of basis filters and different filter parameters. Ticks on the horizontal axis mark the angles of the basis filters, where the continuous response coincides with the approximation. The red dot in the eye image denotes the annotated feature location.

the shape parameters are not suitable. This phenomenon may be related to the frequency spectra of typical natural images, so that the convolved signals are in some sense smoother and easier to interpolate than the basis filters themselves. It should be noted that the continuous response cannot of course be exactly computed, and we have only approximated it here by using a large number ($N = 100$) of basis filters.

In the probabilistic feature detection framework outlined in Chapter 4 the filter responses are further processed by computing a similarity value between two filter bank responses. Ultimately, the factor that affects the performance of the complete object matching system is the quality of the similarity values, not the filter responses.

It turns out that even severely undersampled filter banks, which do not cover the whole frequency domain in orientations, give similarity values which are quite close to the similarity values of the densely oversampled filter bank (with $N = 100$ orientations).

Figure 5.2 shows correlation plots of the similarity values of annotated feature locations and random locations. The eye corner was again used as the test feature, and similarity values were computed in 37 test images. Four different filter shape parameter values, with 4, 6 and 8 basis filters were used. With low, spherical shape parameters ($\sigma_x = \sigma_y = 2$), even the bank with four basis filters gives very good approximations of similarity values of the continuous filter bank.

Increasing the radial shape parameter to $\sigma_x = 4$ or the angular shape parameter to $\sigma_y = 4$ results in additional variance between the medium similarity values of the two filter banks, but it is noteworthy that similarity values at the feature locations are less affected.

With larger shape parameters the similarity values of the four basis filter bank begin to deviate from those of the continuous bank. With shape parameters $\sigma_x = \sigma_y = 4$ there are quite large differences in general in the similarity values compared to the continuous filter bank. Four basis filters become now insufficient as the errors in the background similarity values are so large that they begin to overlap with the similarity values at annotated locations.

The increase in the background similarity values can be seen better with even larger shape parameters ($\sigma_x = \sigma_y = 8$). The similarity values at annotated locations still correlate quite well, but especially with four basis filters, some of the random background locations get incorrectly high similarity scores.

In general, larger values of the shape parameters result in compression of the similarity values toward zero. As the filters become spectrally very selective, it becomes rarer and rarer to find a feature location with a similar filter bank response. This compression effect occurs to some extent regardless of the number of filters in the filter bank.

The previous test measures feature representation capability only at a single orientation. To illustrate the effect of steering approximation in addition to the

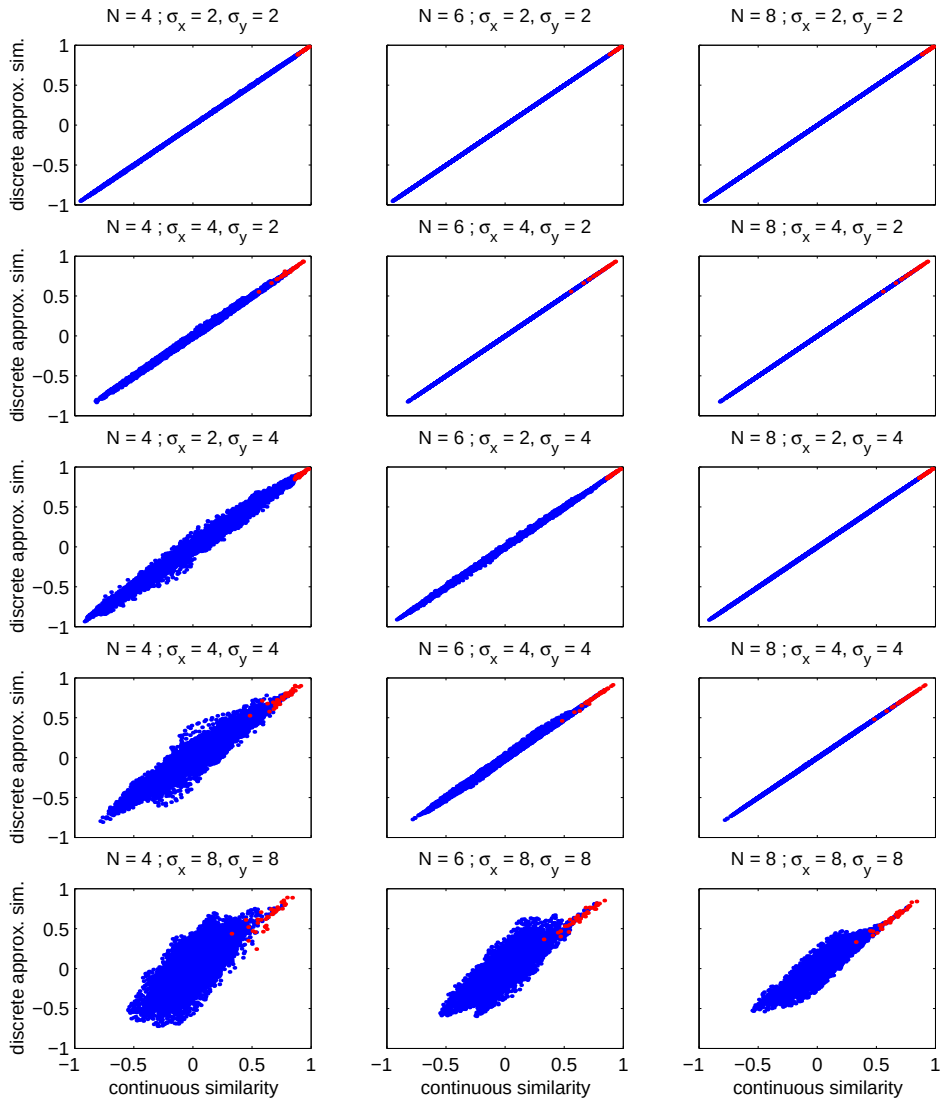


Figure 5.2: Correlation of continuous and discrete similarity values with different filter shape parameters. Red and blue dots correspond to similarity values at annotated feature locations and random locations anywhere in the image, respectively. See text for discussion.

effect of discrete orientations, we construct a filter bank which consists of filters which have been shaped as if they were steered to the intermediate angles, where the steering approximation has largest error. The filter orientations themselves are unchanged. This allows us to study the direct effect of the steering approximation to the similarity values. Figure 5.3 shows correlation plots of the continuous similarity values and the worst-case steered similarity values. Comparing to Figure 5.2, it can be observed that the discrepancy between the two similarity values is larger in general. However, it is noteworthy that even when steering error is fairly large (for example, 13% with the values $N = 8, \sigma_x = \sigma_y = 4$), the similarty values correlate very well.

Using the presented examples one may predict that in feature detection applications using the similarity function framework, gaps in the frequency plane coverage of the filter bank are not critical for detection performance: as we decrease the number of filters in the bank or increase the values of the filter shape parameters, the discrepancy between the similarity values of the continuous filter bank evaluated at the feature locations becomes evident later than what one would expect based on the errors in the steering performance or the function approximation using Eq. 5.1. Also, the steering error is qualitatively such that it appears to preserve the similarity values well at the feature locations, and larger discrepancies are mostly found in the random background locations. In practice, this means that in the presented feature detection framework we can use filter banks which have in principle quite poor steering performance, and still obtain good rotation invariance. However, it should be kept in mind that because the suitable filter parameters are dependent on the properties of the data, no universal conclusions can be made about the minimum number of filters in the bank.

5.3 Gabor parameters and recognition performance

In the previous section we saw that although steerability is a very fragile condition, approximations of continuous responses and especially the similarity values even without the steering correction can be good enough in practice with a wide range of filter parameters, even with highly undersampled filter banks which do not cover the whole frequency space evenly. In this section, the effect of filter shape parameters and the number of orientations and scales to the recognition performance of the system is studied. Moreno et al. (2005) considered the design of Gabor filter banks for local feature detection, but limited their analysis to spherical (isotropic) Gabor functions, with $\sigma_x = \sigma_y$.

We measure the recognition performance using different filter banks with two separate face image databases. Three approximately frontal images of 40 individuals were chosen from the ORL database. Five different images of 15 different persons were chosen from the image sequences of the BioID

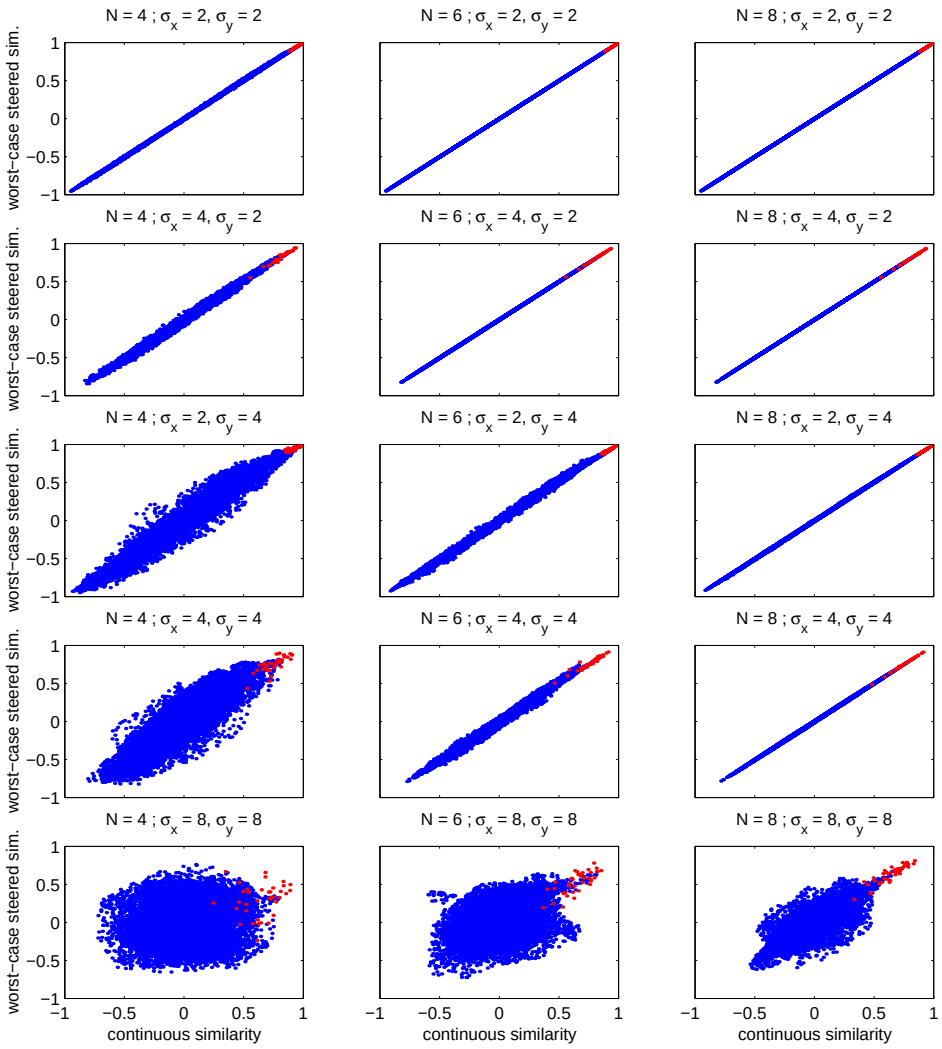


Figure 5.3: Correlation of continuous and worst-case steered approximations of discrete similarity values. Red and blue dots correspond to similarity values at annotated feature locations and random locations anywhere in the image, respectively. Compare with Fig. 5.2.

database. The first image of each identity was used as the reference image and the remaining images were classified. The images from the two databases are show in Appendix A. Since the probabilistic object matching process is computationally quite intensive, its use for the full BioID with 1511 images is unfortunately computationally prohibitive, and only the partial BioID database has be used throughout.

The annotated feature locations included for example eye and mouth corners, tip of the nose and points at the chin line. A total of eleven feature locations were annotated in the images from the ORL database. Twenty feature locations were used in the images from the BioID database. The resolution in the ORL database images is 112-by-92, with faces quite closely cropped in the images. The resolution in the BioID database images is 286-by-384 and there is usually a considerable amount of background visible around the facial region. For performance reasons the resolution of the BioID database was halved into 143-by-192 in the experiments, unless where noted.

The recognition method is similar to the one used in the Elastic bunch graph matching procedure (Wiskott et al., 1999). The total similarity S_G of the face graph is defined as the sum of the individual feature similarities between the reference and test images,

$$S_G(\mathbf{J}^{ref}, \mathbf{J}^{test}) = \sum_i S(\mathbf{J}_i^{ref}, \mathbf{J}_i^{test}). \tag{5.2}$$

This is equivalent to the assumption that the features are statistically independent, and the total probability of all features is the product $P = \prod_i p_i$ of the individual feature probabilities $p_i \propto \exp(\beta S_i)$. A person is recognized correctly if the test image has a higher graph similarity with the training image of the same individual than with any of the other training images. The ORL databases has three images of a single individual. The first one of these is used by the recognition model as the training image, and its similarity is compared to all images in two separate test sets, which consist of the second and third images of all individuals. The partial BioID database was similarly divided into a training set and four test sets, each containing all individuals' images once. The average recognition rate is simply the number of correct classifications divided by the total number of classification tests averaged over the test sets. Chance level of recognition is 1/40=0.025 for the ORL database and 1/15=0.067 for the partial BioID database.

Comparing to the Elastic Bunch Graph Matching method, we are using here only a single exemplar for each individual. Thus the recognition rate measures how well the filter responses describe identity in a pairwise comparison. Since the test images of each person are quite similar to the training image, with no large variation in pose, illumination or accessories, a pairwise comparison is appropriate and helps to bring out the differences in feature representation

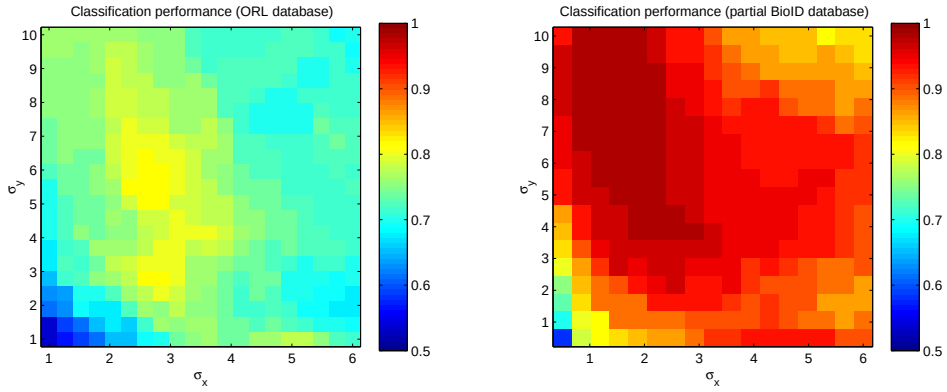


Figure 5.4: Classification results with the ORL and partial BioID databases using annotated feature locations.

capability of the filters.

5.3.1 Recognition performance with annotated locations

In order to separate localization and recognition effects, we first test recognition performance with different filter parameters using manually annotated feature locations. We begin by using DC free near-Gabor filter banks which have three scales in octave spacing, with the highest frequency at the Nyquist limit $f_c = \pi/2$. The tested parameter range $\sigma_x \in [1, 6]$, $\sigma_y \in [1, 10]$ covers all reasonable parameter choices, and at the largest parameter values the filters at the lowest frequency $f_c = \pi/8$ already produce prominent edge effects near the image boundaries.

Figure 5.4 shows the average recognition rate with ORL and partial BioID databases with varying filter shape parameters. The ORL database is more difficult of the two, containing larger number of individuals and also larger variation between the images. Apart from very small parameter values (both σ_x and σ_y less than 1.5), which lead to poor recognition performance with both test databases, the recognition rate is not highly dependent on the shape parameters. The ORL database gives best recognition scores with shape parameters in the region $\sigma_x \in [2, 3.5]$, $\sigma_y \in [2.5, 7]$, while the BioID database favors very narrow and elongated filters, with $\sigma_x \approx 1.5$ and $\sigma_y \in [4, 10]$.

5.3.2 Recognition with face matching

Next we test the effect of the filter shape parameters in a realistic face matching application. We use the hierarchical Bayesian feature matching system using Gibbs sampling presented in Tamminen (2005). The feature likelihood fields

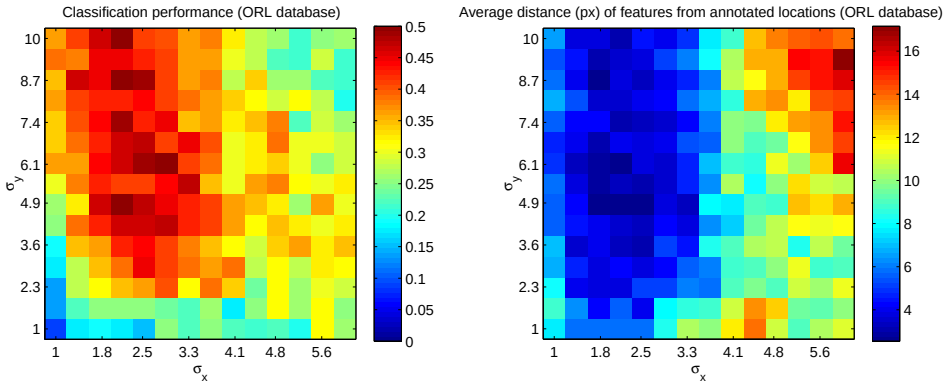


Figure 5.5: Classification results and average distance of detected features from their annotated locations using face matching.

are not rotation invariant and are computed with Eq. 4.5, using the likelihood steepness parameter $\beta = 10$. The Bayesian feature matching system employs a full covariance model for the feature locations in order to describe their interrelated correlations.

We begin by using a filter bank of three scales with center frequencies in octave spacing, $f_c = (\frac{\pi}{2}\ \frac{\pi}{4}\ \frac{\pi}{8})$, and six orientations at each scale. Figure 5.5 shows the recognition performance of the feature matching system with different filter shape parameters, and the average distance of the detected features from their annotated locations. Comparing the recognition results with the ones obtained with manually annotated locations, we can conclude that the two are qualitatively very similar. Recognition performance is in general slightly worse with face matching, reflecting the fact that the feature matching stage occasionally fails to find the correct feature locations. The average distance from the annotated features begins to grow when $\sigma_x > 4$, causing also the recognition performance to suffer. While very small filter parameter values ($\sigma_x = \sigma_y \approx 1$) cause the recognition performance to drop drastically, the average distance from annotated locations does not become excessively large. Such small filters still localize features quite well, but their responses are too generic to be efficient in recognition.

The choice of octave spacing in radial frequency bandwidth is not necessarily the best possible. Let us define radial bandwidth spacing B of the filter bank as a ratio of the filter center frequencies f_c and f_c' at successive scales with

$$B = \log_2 \frac{f_c}{f_c'}. \tag{5.3}$$

This means that $B = 1$ gives a filter bank with octave spacing, where the filter

scales double in size at every level. Figures 5.6 and 5.7 show the recognition results of ORL and partial BioID databases, respectively, with varying the value of B. It can be seen that good values for the filter shape parameters do not depend strongly on the value of B, but recognition performance does. Best classification results are obtained with $B = 1.6$. Note however that as we change the filter bandwidth spacing while keeping the highest frequency constant, these results are dependent on the choice of the highest frequency scale. Here we have used $f_c = \pi/2$ which is also the Nyquist limit.

Because the spatial extent of the filter jet grows larger when we add more scales to the jet, the number of scales N_f has a decisive effect in recognition performance if we keep the bandwidth spacing B constant. Figure 5.8 shows the recognition performance of the matching system using three, four and five levels of scale, with four sets of filter shape parameters. A higher number of scales favor filter banks with tighter spacing, and it is noteworthy that the best recognition performance is obtained at all number of scales when the largest filter is approximately ten times as large as the smallest filter. With very large filter spacings, the largest filters would become spatially wider than the image resolution. Each curve ends when the largest scale is approximately thirty times the size of the smallest scale. It can be seen that the recognition performance does not improve significantly if we increase the number of scales from three.

Figure 5.9 shows recognition performance with double resolution (286-by-384) and half resolution (72-by-96) images. The images at double resolution give nearly as good recognition scores as the regular ones, ane are slightly worse presumably only because the filters at the highest frequency $f_c = \pi/2$ are so small with respect to the features that they are irrelevant for recognition. In addition, the best bandwidth spacing is now larger than previously due to this effect. The half resolution images lose information and thus give worse recognition results. The octave spacing is now best with three scales, as larger values of B lead to the filters being spatially too large. We can however conclude from these results that the best bandwidth spacing is also dependent on the properties of the data being analyzed. Best performance is obtained when the spatial size of the filter matches that of the local features. Unfortunately it is often not clear what constitutes a "local" feature.

Because of the object matching algorithm is based on random sampling, there is some variation in the recognition performance in individual test runs. In order to reliably compare the recognition performance of the four sets of filter parameters in the previous experiment, we repeated the feature matching and recognition procedure with the partial BioID database 30 times, using three frequency scales and $B = 1.7$. The average recognition rates and their standard deviations are given in Table 5.1. The results certify that the best classification scores are obtained with the elongated filters.

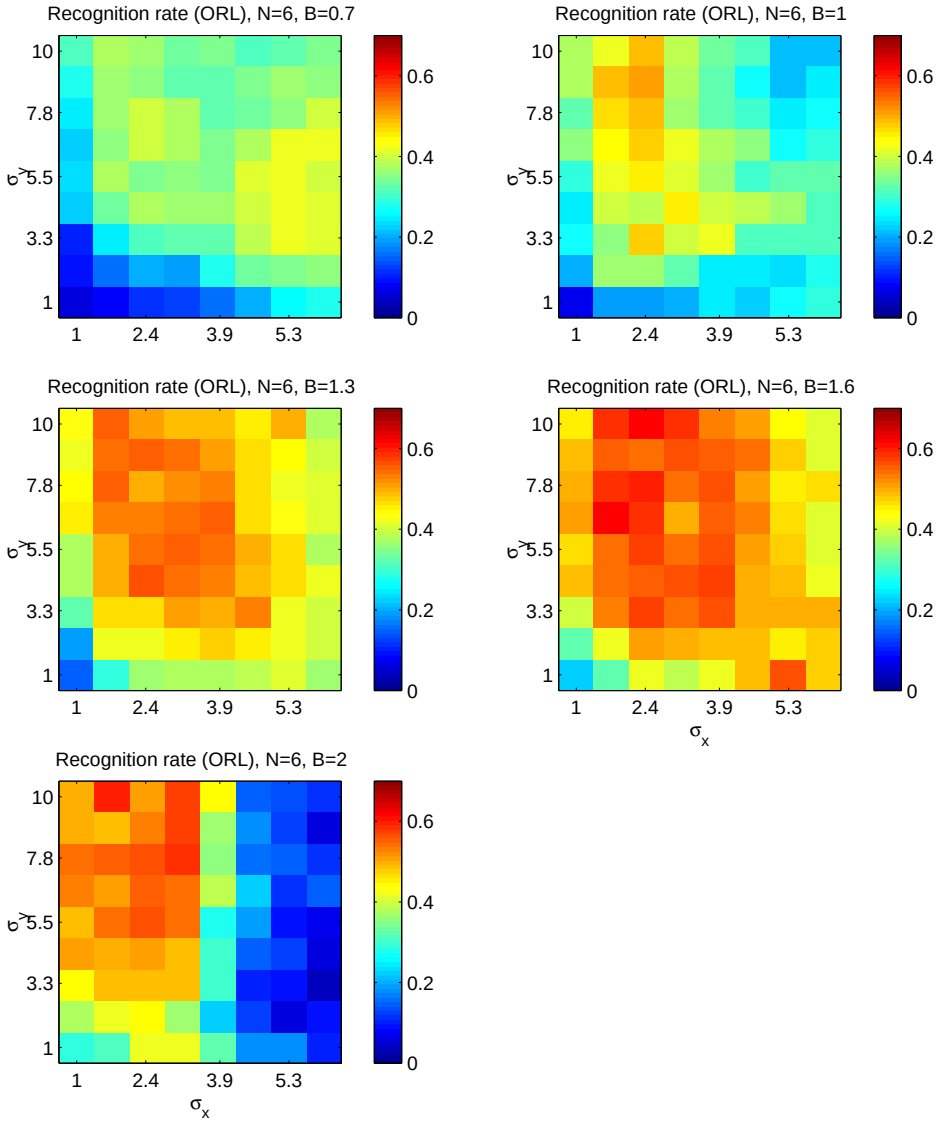


Figure 5.6: Classification results with the ORL database using face matching.

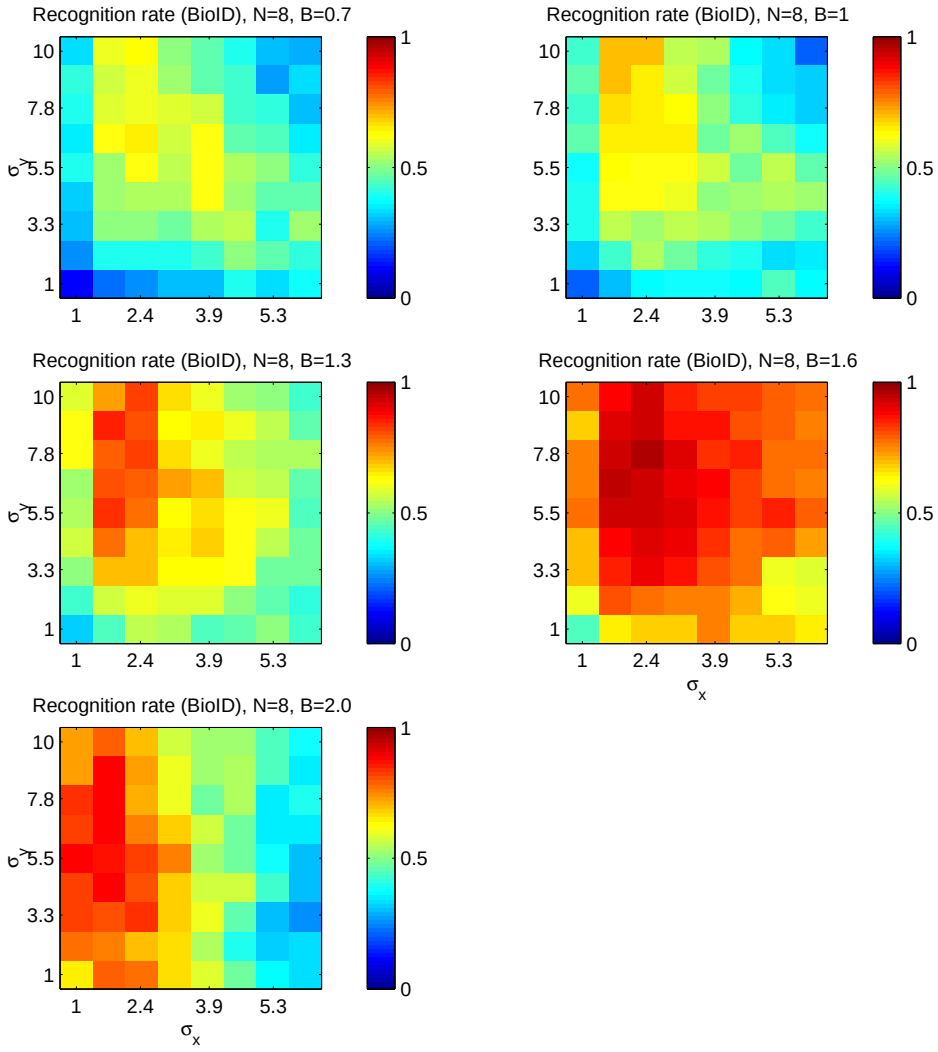


Figure 5.7: Classification results with the partial BioID database using face matching.

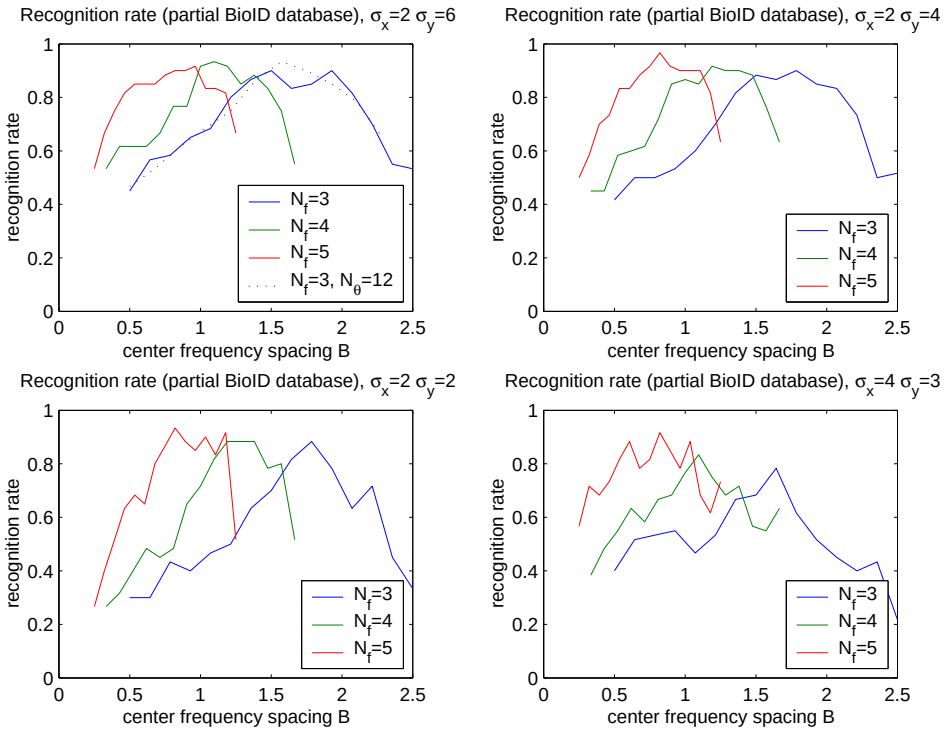


Figure 5.8: Classification results with varying bandwidth spacing and number of frequency scales in the partial BioID database using face matching. Six different orientations have been used, except where noted.

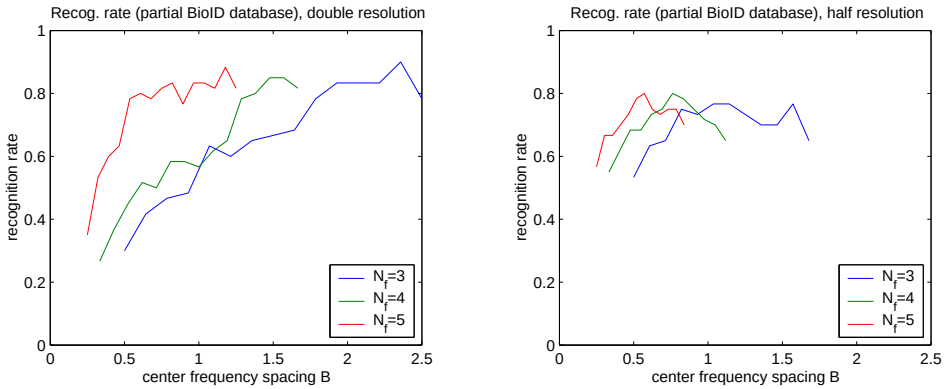


Figure 5.9: Classification results with double and half image resolution. Shape parameters were $\sigma_x = 2, \sigma_y = 6$.

σ_x	σ_y	$p_c \pm \sigma_c$
2	2	0.85 ± 0.03
2	4	0.90 ± 0.02
2	6	0.91 ± 0.02
4	3	0.72 ± 0.03

Table 5.1: Mean classification scores and their standard deviations in 30 repeated matching experiments, with four different sets of filter shape parameters. Bandwidth spacing was set at $B = 1.7$.

5.4 Recognition with angular Gaussian filters

The same approach for finding good filter shape parameters is next applied to an angular Gaussian type filter bank. Figure 5.10 shows the recognition performance using a filter bank of angular Gaussian filters. We used polar Gabor filters given by Eq. 2.19. Their recognition performance is highly dependent on the radial bandwidth, controlled by the parameter σ_r, while angular bandwidth, controlled by σ_θ, appears much less crucial. This is the same behavior that we saw with DC free near-Gabor filters, where σ_x, related mostly to the radial bandwidth, had a stronger effect to the performance than σ_y. We have used the bandwidth spacing $B = 1.5$. The optimal value of σ_r depends to some degree on the choice of the bandwidth spacing, and also on the properties of the data.

When the angular bandwidth is quite narrow, polar Gabor filters are quite close to Gabor filters. Indeed, the best recognition scores are not much worse than the best scores obtained with DC free near-Gabor filter banks. Comparing Figure 5.10 with Figure 3.7 one can see that while best recognition performance is obtained using filters with very narrow bandwidth of approximately 15 degrees, for best steerability the angular bandwidth should be increased to approximately 40 degrees, where recognition performance already begins to suffer.

5.5 Comparison between filter families

In this section, the recognition performance of the system is compared with the partial BioID database using the DC free near-Gabor, polar Gabor, raised cosine type filters (Knutsson et al., 1983) and polynomial (derivative of Gaussian) (Freeman and Adelson, 1991) filters in order to find out how much the choice of the filter family affects recognition performance. Additionally, the filter families and their Gabor-type approximations are compared in order to find out whether the near-optimality in the sense of the uncertainty principle, discussed in Section 2.6, manifests itself in practical recognition results. Six orientations and three

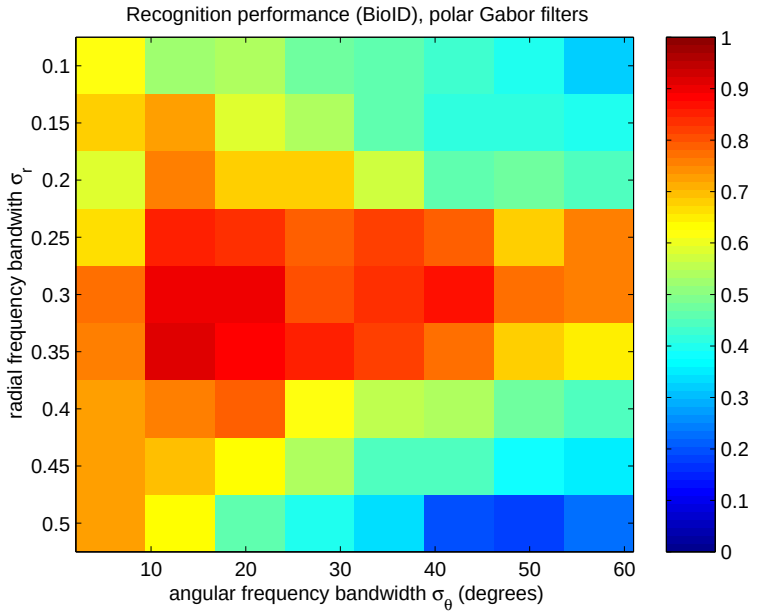


Figure 5.10: Classification results of face matching using polar Gabor filters. Six different orientations and three scales have been used, with bandwidth spacing $B = 1.5$.

scales were used. The response power of all filter types was equalized across scales using the $1/f^2$ rule.

Good filter parameters found in the previous experiments for Gabor and polar Gabor filters are used here. The parameters chosen for the polar Gabor filter bank were $\sigma_r = 0.3$ and $\sigma_\theta = 17°$, and for the DC free near-Gabor filter bank $\sigma_x = 2$ and $\sigma_y = 4.5$ were used. Bandwidth spacing was set to $B = 1.5$ in both filter banks.

The raised cosine filter bank with six orientations is designed following Knutsson et al. (1983) and use filters with a $\cos^4(\theta)$ shaped angular component. The angular bandwidth of the filters, defined as the standard deviation about the angular maximum response, is approximately $\sigma_\theta = 26°$, significantly wider than that of the best polar Gabor filter bank. The radial component suggested in Knutsson et al. (1983) is a log-Gaussian, and a rough optimization gives best results with the parameter $\rho = 0.3$. The log-Gaussian profile is disadvantaged by the fact that the tail of the radial bandwidth profile is quite heavy, which leads to some aliasing in our filter bank design in the highest frequency scale $f_c = \pi/4$.

The derivative of Gaussian filter is a popular choice in feature detection applications because of its simplicity and exact steerability (Boukerroui et al., 2004). The number of filters needed for exact steerability, as well as the angular selectivity of the filters, is determined by the order of the derivative. In order to

have good angular selectivity, we choose to use the fourth derivative of Gaussian filter, which has a fourth degree polynomial in spatial coordinates and is thus steerable with five basis filters (orientations). Its Hilbert pair approximation (from Freeman and Adelson (1991)) is a fifth degree polynomial, and requires six orientations. A disadvantage in using derivative of Gaussian filters is that we can only choose the derivative order, and the radial and angular selectivities cannot be chosen independently. Consequently, the only parameter we can adjust to suit the feature detection problem at hand is the filter center frequency spacing, for which the value of approximately $B = 1.4$ gives slightly better detection results than the spacing $B = 1.5$ which was used with other filter types.

In addition to the previously considered filter banks, the results of each filter type are contrasted with those of a DC free near-Gabor filter bank which has been fitted to have a impulse response with minimum approximation error as defined in Eq. 3.10. For the Polar Gabor and Derivative of Gaussian filters, this results in slightly flattened near-Gabor filters (with $\sigma = [3.5\ 3.3]$ and $\sigma = [2.9\ 2.1]$, respectively), whereas the raised cosine filters most resemble slightly elongated near-Gabor filters (with $\sigma = [2.0\ 2.3]$). The steering errors and approximation errors when applicable are given in Table 5.2.

The response profiles of different filters in the frequency domain are shown in Figure 5.11. Polar Gabor and Derivative of Gaussian filters have quite similar shape to a near-Gabor filter, whereas the raised cosine filter has a significantly non-symmetric response along the radial direction in linear coordinates due to the log-Gaussian radial part, and the long tail is not captured by the near-Gabor approximation with equal center frequency. Raised cosine and Derivative of Gaussian filters have nearly identical angular components, which are wider than those of Gabor and polar Gabor filters in order to facilitate exact steering.

The matching procedure was repeated nine times in order to reduce variance in the results due to the random sampling based matching process. Figure 5.12 shows the average Receiver Operating Characteristic (ROC) curves of the matching experiments. All filters have highly similar performance with very low false positive rates. At higher threshold levels where some false recognition events are tolerated, either DC free near-Gabor or polar Gabor have best performance, with raised cosine filters performing well at low false positive rates but falling behind at higher rates. Fourth derivative of Gaussian filters perform clearly worse, suggesting that they have problems in representing the image features compared to the other filters.

Table 5.2 summarizes the results and some properties of the filter banks in numerical form. The area under the ROC curve (AUC) is a compact measure of the recognition capability simultaneously at different detection threshold levels, and can reliably differentiate between "good" and "bad" models (although not between models which are "good" in different ways) (Marzban, 2004).

Using the AUC as a measure of fitness, the two Gabor-type filters, angular

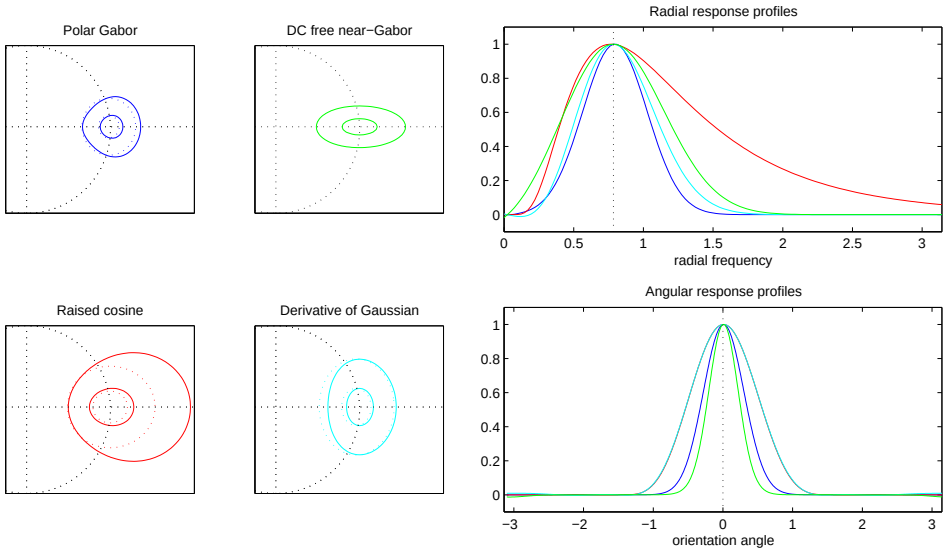


Figure 5.11: Left: Filter profiles in the frequency domain, with their Gabor approximations drawn in dashed line. Polar Gabor and derivative of Gaussian filters are quite well approximated with DC free near-Gabor filters, whereas raised cosine filters have significantly larger support than its Gabor approximation with equal center frequency. Right: Cross-sections of the filter responses through the response maximum in radial and angular directions.

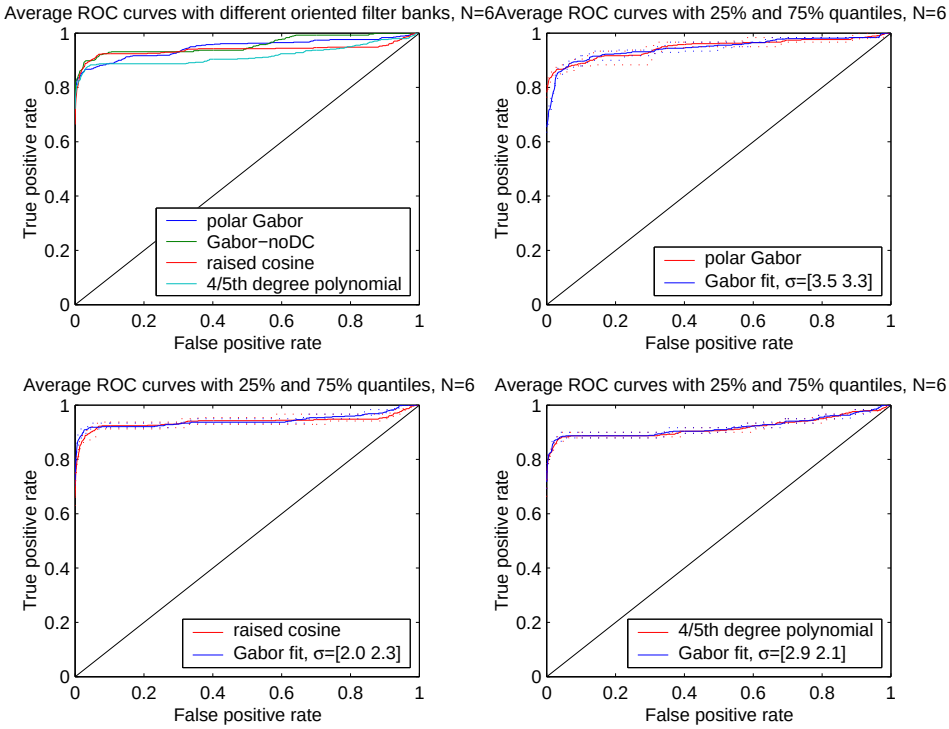


Figure 5.12: Upper left: Average Receiver Operating Characteristic (ROC) curves using polar Gabor, DC free near-Gabor, raised cosine and 4th derivative of Gaussian filters. Upper right: Comparison between the average ROC curves, with 25% and 75% quantiles, of angular Gaussian filters, and similarly shaped DC free near-Gabor filter approximations. Lower row: Average ROC curves and quantiles of raised cosine (left) and derivative of Gaussian filters (right), and their similarly shaped DC free near-Gabor filter approximations.

filter type	area under ROC curve	steering error (N=6)	Gabor fit error
angular Gaussian	0.947 ± 0.014	0.183	-
DC free Gabor fit, $\sigma = [3.5\ 3.3]$	0.942 ± 0.011	0.126	0.041
DC free Gabor, $\sigma = [2.0\ 4.5]$	0.956 ± 0.016	0.499	-
raised cosine	0.938 ± 0.017	$2 \cdot 10^{-5}$	-
DC free Gabor fit, $\sigma = [2.0\ 2.3]$	0.945 ± 0.008	0.067	0.292
4/5th order polynomial (4th DoG)	0.917 ± 0.012	$5 \cdot 10^{-15}$	-
DC free Gabor fit, $\sigma = [2.9\ 2.1]$	0.919 ± 0.010	0.014	0.061

Table 5.2: Summarized results of the matching experiments using different filter families, with the steering errors and Gabor approximation errors.

Gaussian and DC free near-Gabor, perform best although their optimal parameter choices are quite different. The sampling variation is here larger than the effect of the shape parameters. Raised cosine filters perform only slightly worse than the best Gabor-type filters and have the benefit of being very well steerable. Derivative of Gaussian filters are exactly steerable to the limit of numerical precision, but their recognition performance is lacking compared to all other filters in the test. The difference in recognition performance of the filters and their DC free Gabor approximations in terms of the AUC measure is smaller than the sampling variation in all three cases. It can be concluded that the overall filter envelope shape has a larger effect than the difference between the original filters and the approximations. A good property of Gabor-type filters for recognition is that their angular bandwidth is not tied to the number of orientations in the bank. Consequently it appears that at least part of the good performace of Gabor-type filters is due to the fact that the shape parameters can be more freely adjusted to suit the properties of the data, compared to exactly steerable filters.

5.6 Effect of the recognition method

The previous experiments have been conducted using models in which each feature is represented by a single prototype in the feature space. More robust recognition can be achieved by using several feature prototypes and defining the similarity function as the similarity value with the best matching prototype. This strategy allows the representation of more complicated shapes in the feature space compared to a single prototype models for filter responses. In order to extend the relevance of the previous results concerning the filter parameters, recognition results with single prototype and nearest neighbor features are compared in the following.

Figure 5.13 shows the average recognition rate with the full BioID database, consisting of a total of 1511 images of 25 individuals. One eighth of the image database is used as training images and recognition performance is tested with the remaining images in the database. Filter responses for both training and recognition are computed at the manually annotated locations. Eight orientations and three scales were used, with the bandwidth spacing $B = 1.6$. The full database can be used here as the automatic matching system is not used. These results are not directly comparable to the ones in Figure 5.4, where perfect classification was achieved using single models, but the results again show the preference for elongated filters. It is notable that the recognition results of the nearest neighbor model are almost uniformly good, and depend only slightly on the filter shape parameters. This is directly due to the fact that ambiguities in the single filter responses can be remedied by using more elaborate feature and object models in the classification stage. The filter responses appear to contain the relevant information for recognition almost independently of the filter shape parameters, but the features are clustered more tightly with respect to identity with elongated filters. Still, even the nearest neighbor model benefits from elongated filters, as the best recognition rates are achieved with $\sigma_x \approx 2$ and $\sigma_y > 4$. The results suggest that a single mean model for Gabor features gathered from manually annotated locations is not sufficient for reliable recognition of identity even when the images have a relatively consistent quality. We note that the results with 5 scales and $B = 0.7$ (the design choice employed e.g. in Wiskott et al. (1999)) are highly similar to the presented ones. In order to contrast the results above with those obtained with single models in Figure 5.4, we note that while the mean model performs worse than the nearest neighbor model, it is still much more effective than using a single example image from the training set. Choosing randomly a single example image of each identity from the same training set as above and using them to classify the test set achieves a maximum recognition performace of only 0.585, with filter parameters $\sigma_x = 3$ and $\sigma_y = 10$. This is significantly worse than the any of the test scores with the mean model. The difference in performance between the full and partial BioID databases is therefore due to the fact that the partial database is significantly easier to classify than the full database.

Figure 5.14 shows the recognition performance with the partial BioID database using feature locations found with probabilistic object matching. Since the nearest neighbour approach needs several prototypes for each feature, the recognition experiments have been performed in a leave-one-out fashion, building the feature models using four of the images of each individual, and testing the recognition capability with the fifth image. As a result, both feature models are able of achieve perfect recognition (all individuals were classified correctly using any four out of the five images as the training data) with some filter shape parameter combinations. More important than the absolute recognition rates, however, is the

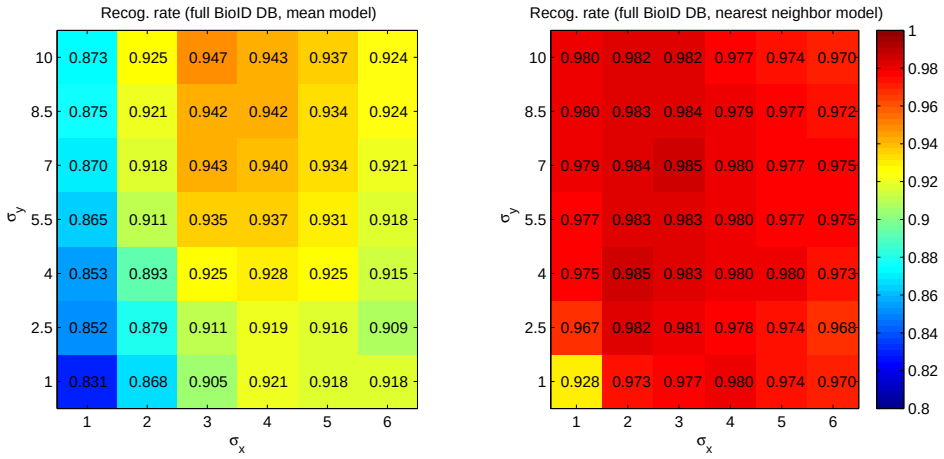


Figure 5.13: Recognition results with the full BioID database using mean and nearest neighbor models for the features, with eight orientations, bandwidth spacing $B = 1.6$ and 3 scales. Filter responses for recognition are computed at the manually annotated locations.

fact that difference in the performace of the two models with small σ_x vanishes, and the mean model is equally good in recognition even when $\sigma_x = 1$.

As the feature models in this experiment are based on automatically found locations, the features tend to be more tightly clustered in the feature space, because the feature models are learned using image locations which have been found using the same similarity measure which is applied in recognition. The difference is especially evident in high frequencies, where the phase component varies spatially very quickly, and manual annotations often have inconsistent phase, which makes their recognition performance poor although the feature points themselves are very close to locations which would cluster tightly with respect to identity. Two solutions to this problem of manual annotations are to either adjust the feature locations automatically or to discard the highest frequencies which are most sensitive to small displacements.

5.7 Discussion

From the previous experiments we can conclude that in order to achieve good recognition performance with a face recognition system based on Gabor-like oriented filters and numerical optimization of point feature locations, the DC free near-Gabor filter shape parameter σ_x should be quite small, so that the real and imaginary parts of the filters have only a few sidelobes in the spatial domain and do not cause false maxima to the similarity functions. On the other hand,

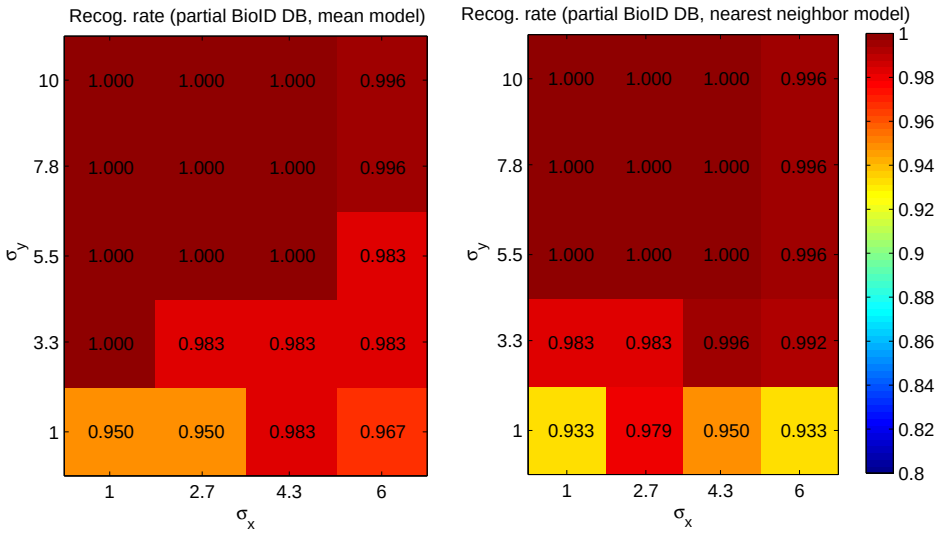


Figure 5.14: Recognition results with the partial BioID database using mean and nearest neighbor models for the features. Feature models for recognition are based on filter responses at locations which are found automatically using probalistic object matching.

the parameter σ_x should be large enough so that the DC correction does not worsen localization performance. The value of the parameter σ_y is less critical, but recognition performance is slightly improved with filters which are spatially elongated in the direction orthogonal to the wave vector (in other words, filters which have wider bandwidth in the angular than the radial frequency dimension), that is, $\sigma_y > \sigma_x$.

As the parameter region where Gabor and near-Gabor filters are best steerable lies where $\sigma_y \leq \sigma_x$, regardless of the number of basis filters, we face a dilemma in the filter bank design when attempting to use steerability as a guideline. Good recognition performance and good approximate steerability of Gabor-type filters, with a low number of basis filters, appear to be conflicting design goals. The same behavior was also seen with angular Gaussian filters. While DC free near-Gabor and polar Gabor filters which are approximately steerable can be quite efficient in recognition, best performance was obtained with filters which are narrower in angular bandwidth than what is required for good approximate steerability. Compared to exactly steerable filters, approximately steerable filters have the benefit that the number of orientations is not tied to the parameter controlling the radial bandwidth. This gives the filters more flexibility in adapting to the properties of the data, with the obvious cost that some steering error.

It is interesting to note that the data from physiological measurements of simple cells found in the mammalian visual cortex suggest filter banks with an

elongation ratio of $\sigma_y/\sigma_x = 2/1$ and bandwith spacing equivalent to $B = 1.5$ (Daugman, 1980),(Daugman, 1988). These parameter values are compatible with the findings of this chapter. However, drawing strong conclusions from such facts that may be only coincidental should be avoided. Our data represents only a small and very specialized feature detection task, whereas the mammalian visual system has to cope with a much wider variety of scenes. Also, although the EBGM model is biologically inspired, there is no direct biological evidence for the feature jets or the feature graph representation which our system uses.

Compared to other artificial engineered recognition systems, we first note that our findings regarding the shape of the filters are compatible with the features selected with both Gabor Wavelet Network and Adaboost algorithms (Shen and Bai, 2006), which both favor elongated filters with only a small number of sidelobes.

Table 5.3 gives some Gabor-type filter bank designs presented in the literature for feature detection and texture classification applications, and their shape parameter values converted to our parameterization.

Compared to the parameter choices in Wiskott et al. (1999), the presented results indicate that a significantly smaller value than $\sigma_x = 2\pi$ is beneficial for both localization and recognition. This may be in part due to the global optimization approach we use in solving the feature localization problem. If good initialization methods are available, local optimization are sufficient and the problem of false similarity maxima caused by the filter sidelobes are alleviated to some degree. Also the flexibility of the object shape model may affect the choice of optimal filter bank parameters. Nevertheless, as there appears to be no theoretical, biological or practical justification for the relatively widely used choice $\sigma = 2\pi$, care in the choice of the shape parameters is advised, as we have demonstrated that they can affect recognition performance. The bandwidth spacing $B = 0.7$ using five scales is in good agreement with our results, but it does not offer improvement in recognition results compared to using the bandwidth spacing $B = 1.6$ and three scales. Both banks have a span of slightly over three octaves in their center frequencies.

Compared to the parameter choices in (Tamminen, 2005), the presented results indicate that recognition will benefit from larger values of the shape parameters, especially σ_y. Also the bandwidth spacing should be increased, since only three scales are used. Tamminen (2005) considered only feature and object localization, and the filter bank parameters appear to be suboptimal for recognition, although the localization performance of the complete system is not hindered greatly because of the highly powerful matching method.

The filter bank design of Kruizinga and Petkov (1999) is most similar to what our results indicate, although their work concerns oriented texture classification applications. Typically, texture classification has preferred Gabor filters with relatively large σ_x and σ_y, such as in (Ro et al., 2001). However, in the work

of Kruizinga and Petkov (1999), Gabor filter outputs were used as inputs for nonlinear grating cell operators, which explains why a large σ_x is not preferred in the linear filtering stage. Wu et al. (2000) aim for rotation invariance without using steering (see also (Haley and Manjunath, 1995)), and consequently choose filters with small σ, giving a wide support in the spatial-frequency domain. The filter bank design in (Serre et al., 2007) is radically different from the others, with highly undersampled orientation dimension and highly oversampled scale dimension. It is clear that rotation invariance cannot be achieved in the filter level if the orientation dimension is significantly undersampled, and must be implemented higher up in the processing if needed.

	σ_x	σ_y	N_θ	B	N_f
Wu et al. (2000)	1.7	1.7	6	1	4
Kruizinga and Petkov (1999)	3.5	7.0	8	1	3
Ro et al. (2001)	4.4	5.6	6	1	5
Wiskott et al. (1999)	6.3	6.3	8	0.7	5
Tamminen (2005)	1.89	1.89	6	1	3
Serre et al. (2007)	5.0	16.6	4	0.1-0.4	16

Table 5.3: Parameter choices of Gabor-type filter banks found in the literature, converted to the parameterization in Eq. 3.13.

Chapter 6

Rotations in depth

6.1 Introduction

In this chapter, a regression-based approach for modeling out-of-plane or depth rotations of oriented filter based features is presented. The effects due to these rotations are significantly more varied than the plane rotations considered so far in the work, because the features change in a way which is dependent on the three-dimensional shape of the object.

The goal is to build a model for the change of appearance in local features in order to recognize the features and determine pose parameters in arbitrary pose. The proposed approach is best applicable to textured objects which have relatively smooth surfaces, so that the out-of-plane rotations cause relatively smooth variation in the filter responses. Typical object classes of this type include solid objects such as cans, boxes and human faces. We will continue to use human faces as the reference object class.

The use of synthetic data for learning pose-invariant object models has been proposed in (Vetter, 1996), who presented synthesis of novel views of human head models using morphable 3D models. This "interpretation through synthesis" approach has common ground with Active Appearance Models (Cootes et al., 2001). A component-based version for 3D face pose modeling using the same approach was presented in (Weyrauch et al., 2004), using automated generation of 3D face models from photographs. In this work, synthetic 3D face models are similarily generated from frontal photographs, and they are used in learning feature models for the spatially sparse local feature based object representation which is used in the Elastic Bunch Graph Matching model.

The appearance of features in the human face, such as eyes and mouth, vary characteristically depending on head pose. For example, the mouth appears mainly as a horizontal line in a directly frontal pose and neutral expression, but when the object is rotated, the main orientation of the local gray-level structure

changes, and so do also any features which are not rotation invariant, such as the responses of our oriented filters. Some of the variation is similar to what occurs in the case of plane rotations, but there are also new phenomena such as non-linear contractions and expansions of the gray level structure, and also self-occlusion.

Section 6.2 discusses object pose modeling in general and the regression approach employed in this work in particular. Section 6.3 introduces the parametrization of object pose and gives a justification for prior probability distributions for the pose angles in estimation problems. Section 6.4 presents two different regression models for the responses of oriented filters. The subsequent four sections consider human head models in particular. Section 6.5 presents the method for generating 3D head models from single photographs which has been used in this work. Section 6.6 discusses the recording of feature data and Section 6.7 considers self-occlusion of the features. The prediction performance of head feature models is evaluated in Section 6.8 using the feature similarity function.

The work presented in this chapter has been published in (Kalliomäki and Lampinen, 2003).

6.2 Subspace and regression modeling of object pose

Subspace methods such as PCA have been commonly applied to modeling identity variation of faces in known pose with good results (Pentland et al., 1994). As the subspace of identity variation is of unknown dimensionality and nontrivial to parameterize, the PCA approach is easily justifiable, especially if the values of the pose parameters are not known in the data. In contrast, the dimensionality of the pose subspace is known to be exactly three, since it is spanned by three rotations, and it can be fully parameterized by for example Euler angle or quaternion representations.

In pose modeling, one would expect worse performance from linear methods such as PCA, as the *pose manifold* (Gong et al., 1996) (the embedded space of all possible orientations of the face or its features, spanned by the rotation parameters) is low-dimensional but typically strongly nonlinear and its shape can be very complicated. Subspace methods can be applied also to modeling pose effects in features, although a single linear subspace is insufficient in explaining large pose variations. Combinations of several locally linear models have been proposed to overcome the limitations of linearity (Okada and von der Malsburg, 2001), (Tae-Kyun and Kittler, 2005).

Burns et al. (1993) proved that for point-set and line-segment features, there are no general-case view invariants, and proposed that view-varying features and probabilistic methods should be used for effective 3D object recognition. This is the approach we will take in this work, and model pose variation directly in the latent space generated by the pose parameters. This requires data with

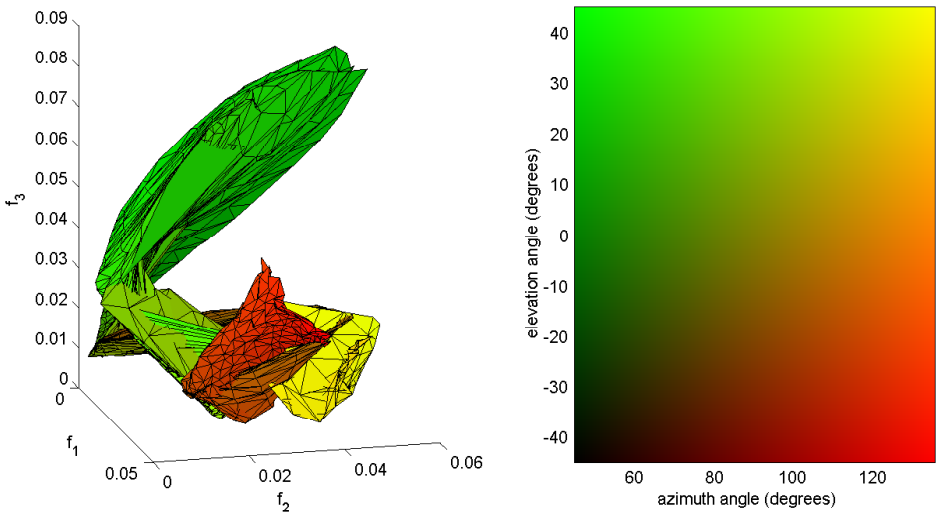


Figure 6.1: Two-dimensional pose manifold in a three-dimensional feature space. Absolute values of the responses of three oriented filters are used as features f_1, f_2 and f_3. Pose is parameterized with azimuth and elevation angles.

known values of the pose parameters in each training image. Collection and labeling of real-world training data would be a daunting task, and we will use synthetic, computer-generated models of the objects in model building stage. A disadvantage of using synthetic data is that the predictions of the model might not match real-world data. Also, direct regression modeling of the complicatedly shaped pose manifold requires a flexible nonlinear regression model, whose parameters may be difficult to learn from data.

High-dimensional spaces, such as our typical feature spaces, are difficult to illustrate effectively. As an example, consider the two-dimensional pose manifold spanned by the azimuth and elevation rotation angles in a three-dimensional feature space, portrayed in Fig. 6.1. Azimuth and elevation angles were perturbed at most 45 degrees from the directly frontal view. The value of the azimuth angle parameter is illustrated with color. It can be seen in Fig. 6.1 that the changes in the features are quite smooth. The resulting manifold is self-intersecting and has a difficult twisted shape, qualitatively quite similar to the ones observed in (Gong et al., 1996), who used global PCA features.

6.3 Parameterization of rotations

In the regression model we consider only 2D rotations and parameterize them with azimuth and elevation angles (ψ, ϕ), which act as latent variables, and the

response of each filter in an oriented filter bank is modeled by a function $f_k(\psi, \phi)$. The third rotation, parameterized as the rotation about the view axis, is easier to model. If we use a rotationally symmetric filter bank, in which the filters in a single scale are rotated versions of each other, rotating the filters is equivalent to rotating the image about the view axis, and we can use the results of Chapter 3 to model the changes in filter responses due to in-plane rotations.

The nonlinear rotation parameters cause some complications that need to be addressed. Namely, the regression model should consider the points of the view sphere equally important in principle. In other words, equal modeling effort should be given to all surface elements

$$dA = \cos(\phi)\, d\phi d\psi \tag{6.1}$$

of the view sphere. Uniform sampling of ψ and ϕ does not fulfill this requirement, since at high elevation angles rotation about the azimuth angle degenerates into in-plane rotation, and many of the samples describe then the same point in the view sphere. Speaking in probabilistic terms, in order to sample the surface elements dA uniformly, we must assign a $\cos(\phi)$ prior on the elevation angle parameter ϕ. We can design a deterministic sampling scheme of the view sphere easily by setting the number of sample points at the equator N_0 and sampling each latitude with $N_0 \cdot \cos(\phi)$ sample points. Fig. 6.2 shows how the sample points are distributed in the rectangular (ψ, ϕ)-coordinates.

6.4 Modeling oriented filter responses

We continue using Gabor filters as the recognition features, with a filter bank of three scales and six orientations. The modeling methodology proposed here can be used with any spatial oriented filters, such as steerable filters (Simoncelli and Freeman, 1995) or derivative of Gaussian filters.

Having obtained the filter response data, we need to model it as functions $f_j(\psi, \phi)$, where ψ and ϕ are the azimuth and elevation angles of the pose and i refers to filter index. This is a typical regression problem. Fig. 6.3 illustrates the modeling setup. In Fig. 6.3 *a*) a single feature, the left corner of the left eye, is tracked. The responses of a single oriented filter, responding to horizontal lines, are recorded in the sampling points of the view sphere.

Fig. 6.3 *b*) shows the measured amplitude responses of the filter tracking the corner of the eye. Large amplitude responses are obtained when the filter correlates strongly with the image. This includes a large area in the left half-plane. In the right half-plane, that is, with positive azimuth angles, the amplitude responses are low because the feature does not contain strong horizontal structures in those poses. Fig. 6.3 *c*) shows the measured phase responses of the same filter.

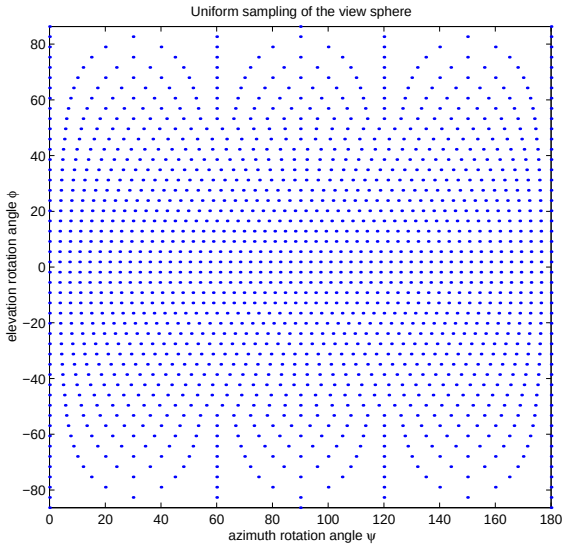


Figure 6.2: Distribution of uniformly sampled points in the view sphere. The sample points are more sparse near polar regions, corresponding to the fact that at high elevation angles, all azimuth angles represent nearly the same view of the object.

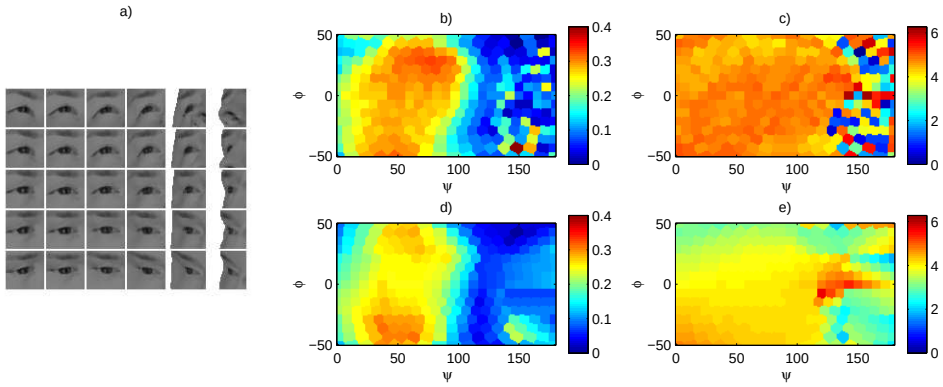


Figure 6.3: a) Samples from the pose space of the feature (center of the left eye). b) Measured amplitude response of the filter. c) Measured phase response of the filter. d) Modeled mean amplitude. e) Modeled mean phase. See text for explanation.

6.4.1 Piecewise linear model for filter responses

A flexible model is needed to capture the highly nonlinear effects in the amplitude and phase regression functions. In (Kalliomäki and Lampinen, 2003), we originally proposed a piecewise linear model for the filter responses, covering a smaller rectangle $\psi \in [-50°, 50°]$, $\phi \in [-30°, 30°]$ of the whole view

space. However, the same approach can be applied to the larger pose space $\psi \in [-90°, 90°]$, $\phi \in [-50°, 50°]$ considered here.

The locations of the piece boundaries are not optimized, but fixed, in order to simplify the modeling process. The complex response of an oriented filter is modeled as a product of the amplitude component $A_k(x; a)$ and the phase component $\phi_k(x; b)$, with

$$j_k(x; a, b) = A_k(x; a)\phi_k(x; b) = a^T x\, e^{ib^T x}, \tag{6.2}$$

where $x = [\psi\ \phi\ 1]^T$ is the pose angle vector and a are the linear model parameters for amplitude and b for phase, respectively, and k is the filter index. The prediction of the whole filter bank response is obtained by stacking the models $j_k = a_k^T x\, e^{ib_k^T x}$ into a vector $J = [j_1, \ldots, j_n]^T$.

Amplitude responses in the measurements are typically quite smooth and the modeling process is straightforward. The model is fitted simply by computing the pseudoinverse solution which minimizes the square error between the model predictions and data. Fig. 6.3 *d*) shows the predictions of the piecewise linear model for the amplitude. Phase information is most important when the amplitude is large. Because of this, we design the linear models for phase using the weighted least squares method, with the amplitudes acting as weights. Phase responses are more complicated to model since they contain discontinuous bifurcations where the phase jumps quickly from one arbitrary value to another. Because the phase values are 2π-periodic, we can remedy some of these jumps by changing the phase values which are lower than some threshold value to their 2π complements before computing the linear model, and choose the threshold value which produces the best predicting linear model. Fig. 6.3 *e*) shows the predictions of the piecewise linear model for mean phase.

The piecewise linear model is easily invertible, and it is possible to quickly compute the pose parameters that are most similar to a given filter jet. However, the linearity of the model inside each piece unfortunately also means that the optimal solutions will very often lie on the piece boundaries. This undermines the applicability of the approach, as the piece boundaries are not supposed to have a special status compared to other poses.

6.4.2 Mixture of Gaussians model for filter responses

Next we will consider an alternative, nonlinear regression model for the filter responses. As the magnitude of the complex Gabor filter has a Gaussian shape, the normalized filter jet amplitude data consists of quite smooth effects especially in the region of the pose space where the feature is visible and thus better predictable. The piecewise amplitude model is not able to describe the smooth variations very well, and also the number of parameters in the model grows large because a large

number of pieces is needed in order to model the nonlinear effects of the data.

As an improved model for the amplitude responses of the filters we consider a mixture of Gaussians model, in which the amplitude predictions model are given by

$$A_k(x;\mu,S) = \sum_{i=1}^{N} w_{ik} \exp\left(-\frac{1}{2}(x-\mu_i)^T S_i^{-1}(x-\mu_i)\right), \tag{6.3}$$

where $x = [\psi\ \phi]^T$ is the view vector, and μ_i are the centers and S_i the covariance matrices of the regression kernels. The centers and covariance matrices of the Gaussian kernels are optimized, but in order to reduce the number of free parameters in the model, all features use same centers and covariance matrices. The weights w_{ik} are the only parameters in the model which are feature-specific. The resulting predicitions are typically such that only a couple of the weights are active for each feature.

Phase responses of the filters are typically quite smooth as long as the feature describes the same gray-level structure in the image. A choice which suits the properties of the data quite well is to define the fixed piece boundaries using the centers of the Gaussian kernels such that the phase value is predicted by the linear model associated to the nearest center. The centers then define a Voronoi tessellation of the pose space, with a separate linear model in each Voronoi cell.

6.5 Synthetic head models

As an example object class, we will considering human faces. In order to measure the filter responses we generate synthetic head models[1]. The shape of the reference head model is deformed to match the feature locations in a frontal photograph, and texture mapped. Using the IMM-DTU database images (Stegmann, 2002), we construct 37 different 3D training head models.

We use the annotated feature locations in the images to guide the shape deformation process. The feature locations are connected by a graph structure consisting of quadrangles. The three-dimensional models are generated by deforming the reference 3D model in such a way that the feature locations in the image plane match the annotated locations.

We compute the piecewise linear mapping A_q from undeformed to deformed space for each quadrangle q of the feature grid using the Moore-Penrose pseudoinverse,

$$A_q = R^T P (P^T P)^{-1}, \tag{6.4}$$

where P contains the screen coordinates of the corners of the reference quadrangle as row vectors and R contains the screen coordinates of the target feature

[1] Shape model courtesy of University of Washington

locations. The depth coordinate remains unchanged in the deformation process, and thus each piece in the piecewise linear model A_q has six free parameters. As a quadrangle in two dimensions has eight free parameters, the linear system is over-constrained, which increases the stability of the solution while introducing only minor errors in the feature locations of the deformed shape. The mapping A_q is applied to all object vertices which are inside the quadrangle q.

After the shape deformation, optimal light direction parameters (ψ_l, ϕ_l) and ambient and diffuse reflection material parameters (A, D) are sought by minimizing the L_1 norm

$$E(\psi_l, \phi_l, A, D) = \sum_{(x,y)} |I(x, y) - f(x, y; \psi_l, \phi_l, A, D)|, \tag{6.5}$$

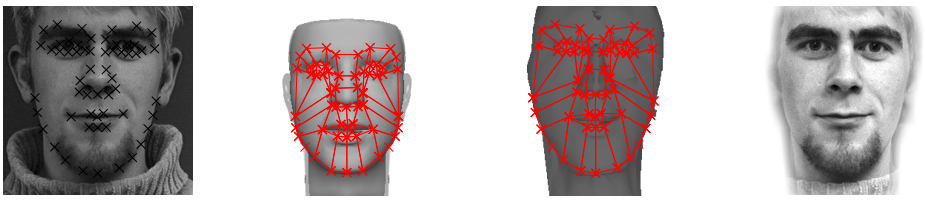
between the gray-scale model $f()$ and the perceived image I. We found that the L_1 norm yields visually more pleasing results compared to the L_2 norm. However, the estimated lighting parameters lead typically to shaded images which are too dark, because the shading model interprets some parts of the image, such as eyebrows, as shading phenomena.

Finally, the shape is textured with projective texture mapping. We use a multiplicative texturing model, where the final pixel color is the product of the texture value and the gray scale value after lighting computation. The required texture T is then easily computed by dividing pixelwise the perceived image I with the gray-scale image of the rendered model $f()$,

$$T(x, y) = I(x, y)/f(x, y). \tag{6.6}$$

Our deformable face model is slightly too flexible, allowing some physically impossible deformations in the head shapes, which consequently lead to unlikely texture estimates. The texturing however makes the visual quality of the model quite good under non-extreme rotations and lighting. Figure 6.4 shows the shape deformation process which generates the three-dimensional training head models.

Three additional reconstruction examples are shown in Figure 6.5. Some artifacts from texturing become visible as dark stripes, where the frontal texture estimate is based on the value of only few pixels. Generally the visual quality of our 3D head models is more or less equal to other image-based modeling approaches (e.g. (Zhang, 2001), (Liu, 2003)). Modeling based on data obtained with 3D scanners achieves better visual quality (Blanz and Vetter, 1999), but the data acquisition process is rather elaborate.



a) b) c) d)

Figure 6.4: Head shape reconstruction using point correspondences. *a)* Test image with annotated feature locations. *b)* Reference shape *c)* Deformed shape. *d)* Corresponding texture map.

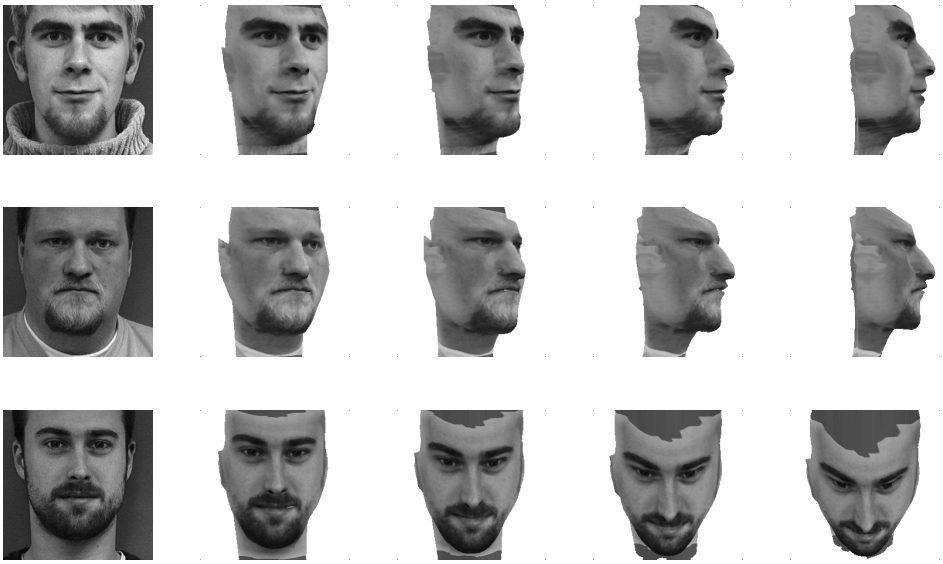


Figure 6.5: Three test images and the rotated head models

6.6 Recording feature data

We track feature locations in the synthetic face models for varying azimuth and elevation angles, and store the responses of filters. The filters remain centered to the feature locations as the head rotates. We have used 45 feature locations in the inner face and 13 locations in the jaw line in our experiments. A total of 34 head models were used in collecting the filter response data, and the remaining three head models were used for validating the performance of the model.

We cover a part of the two-dimensional pose space with a rectangle $\psi \in [0°,\ 180°]$, $\phi \in [-50°,\ 50°]$, with the point $(\psi, \phi) = (90°, 0°)$ corresponding to

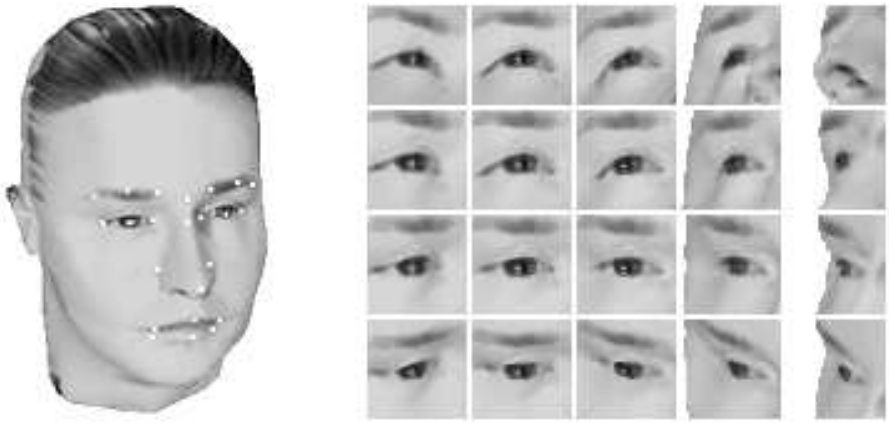


Figure 6.6: Left: Synthetic head model. Right: Tracking the center of the left eye across the pose space. The principal orientation of edges in the eye images changes considerably due to pose. In the rightmost column the eye becomes partially occluded by the nose and the main feature here is radically different from the others, namely the vertical edge of the head.

a frontal view. The rotation angles are sampled according to Section 6.3 so that we get approximately uniform coverage of the view sphere, with a total number of 321 different views of each head. Fig. 6.6 shows a rendering of the head model with white markers added to feature locations for visualization, and zoomed left eye in several orientations.

Alternatively, it would be possible to take a large number of photographs from a real head instead of using a synthetic head. However, there are many advantages of using synthetic data to build the filter response model. In practice the most important of them is that measuring the filter responses from synthetic data takes far less time. Instead of taking hundreds of photographs in varied poses, the head is rendered using efficient 3D graphics hardware, and the filter responses are computed from the rendered image. Furthermore, head pose and lighting conditions can be accurately controlled. Reliable control over pose angles is quite difficult to achieve in real-world photography. The tracking of feature locations is also easy and precise using a synthetic model. With real-world image data one must either label the feature locations manually or track them automatically.

Compared to real-world data, the main disadvantage is that the visual quality of the synthetic model is lower. The model has been built using a single frontal photograph, and its features become somewhat unrealistic, especially in highly rotated poses. Also, the used Phong lighting model gives rather unnatural results for human skin, and lacks cast shadows due to self-occlusions.

6.7 Self-occlusion of features

As the head model is rotated, some of the feature points in a particular view become occluded and are not directly detectable. It is then a reasonable question to ask whether we should spend modeling effort in the view space outside the occlusion boundary of a feature. The answer appears to be twofold. On one hand, some of the features have strong effects near or at the occlusion boundary, and the predictive power of the model can be quite good even to some extent beyond the occlusion boundary. On the other hand, typically the feature responses are quite smooth while the feature is visible, and become much less stable when the feature is under self-occlusion, because the feature location no longer corresponds to a stable gray-level structure of the image. The feature data is recorded at the image plane location where the feature would be if it were visible, and if the feature is under self-occlusion, the image plane location of the feature does not any more correspond to a single vertex on the 3D surface of the object.

We can determine the visibility of each feature by texture mapping the head model with a color-coded texture map, in which each pixel of the map has an unique RGB combination. Determination of visibility is efficiently computed by the Z buffer of the OpenGL renderer, and the test for visibility is easy. We need only test if the pixel color at the location of the feature is the same as the texture map color of the corresponding model vertex. Graphs of the face object with visible features are shown in Fig. 6.7.

As the features have been annotated at frontal pose, the feature locations are best representative of the object in near-frontal poses. Highly rotated near-profile views would benefit from additional feature points at the profile edge, but as our feature locations correspond to three-dimensional model vertices, the exact location of the profile edge is difficult to represent with them, and such vertex locations would be in most poses either away from the profile line or already occluded.

6.8 Model evaluation

Finally, we wish to evaluate the prediction performance of the regression model. As an example, Figure (6.8) shows the predictions of a complete filter jet, tracking the left eye corner, with the mixture of Gaussians model. The predictions are very good inside the visible region. The phase data varies quickly outside the visible region, where the filter responses do not correspond to any stable gray-level structure, and are difficult to predict.

In order to confirm that the model performs adequately, we compute the mean feature similarity between filter jets J_{test} from the three synthetic test head models

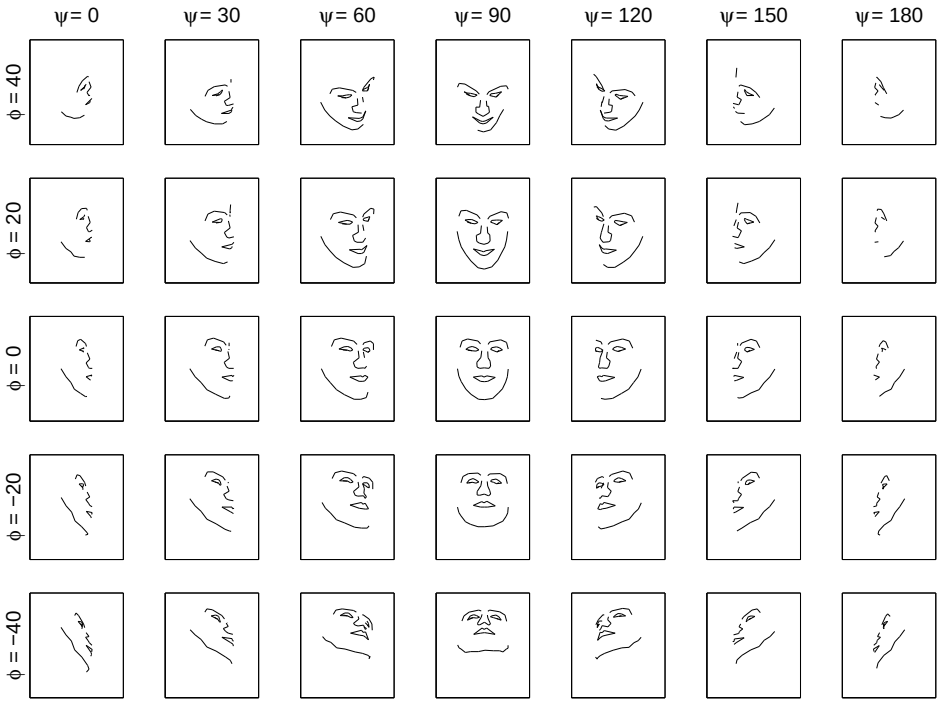


Figure 6.7: Graphs depicting the visible features of the average head model in varying poses.

and the regression model predictions J_{reg},

$$\frac{1}{N}\sum_{k=1}^{N}\frac{1}{|V_k|}\sum_{(\psi,\phi)\in V_k} S\left(J^k_{reg}(\psi,\phi),\, J^k_{test}(\psi,\phi)\right), \tag{6.7}$$

where V_k denotes the region in the pose space where the feature k is visible. This measure, which is evaluated at the known feature locations, is approximately 0.79 for the mixture of Gaussians model and 0.77 for the piecewise linear model. All of the ten best features have an average similarity of over 0.9 everywhere in the visible region of the pose space. In comparison, the mean feature similarity with a constant feature model, taken at a directly frontal pose, is only approximately 0.24, and the mean feature similarity of ten best features is 0.38. From this we can conclude that the regression models are consistent with the synthetic data and the predictions of the models are reasonably good and improve significantly the performance compared to a single frontal model.

Outside the visible region the features behave differently depending on mainly how far the feature location travels away from the occluding boundary. Feature

points at eyes, eyebrows and nostrils remain very predictable, with average similarity of 0.88 with the mixture of Gaussians model, while feature points at the jaw, mouth and nose lines have average similarities around 0.5 when occluded. It should be noted however that the similarity scores themselves do not tell much about the detectability of a feature. When occluded, feature points are often located in smooth regions in the cheeks with no edges or textures which, while very similar to each other, are not very specific.

The usefulness of the feature model will come under a true test in the next chapter, in which we will use it as a reference model in an object matching problem involving depth rotations.

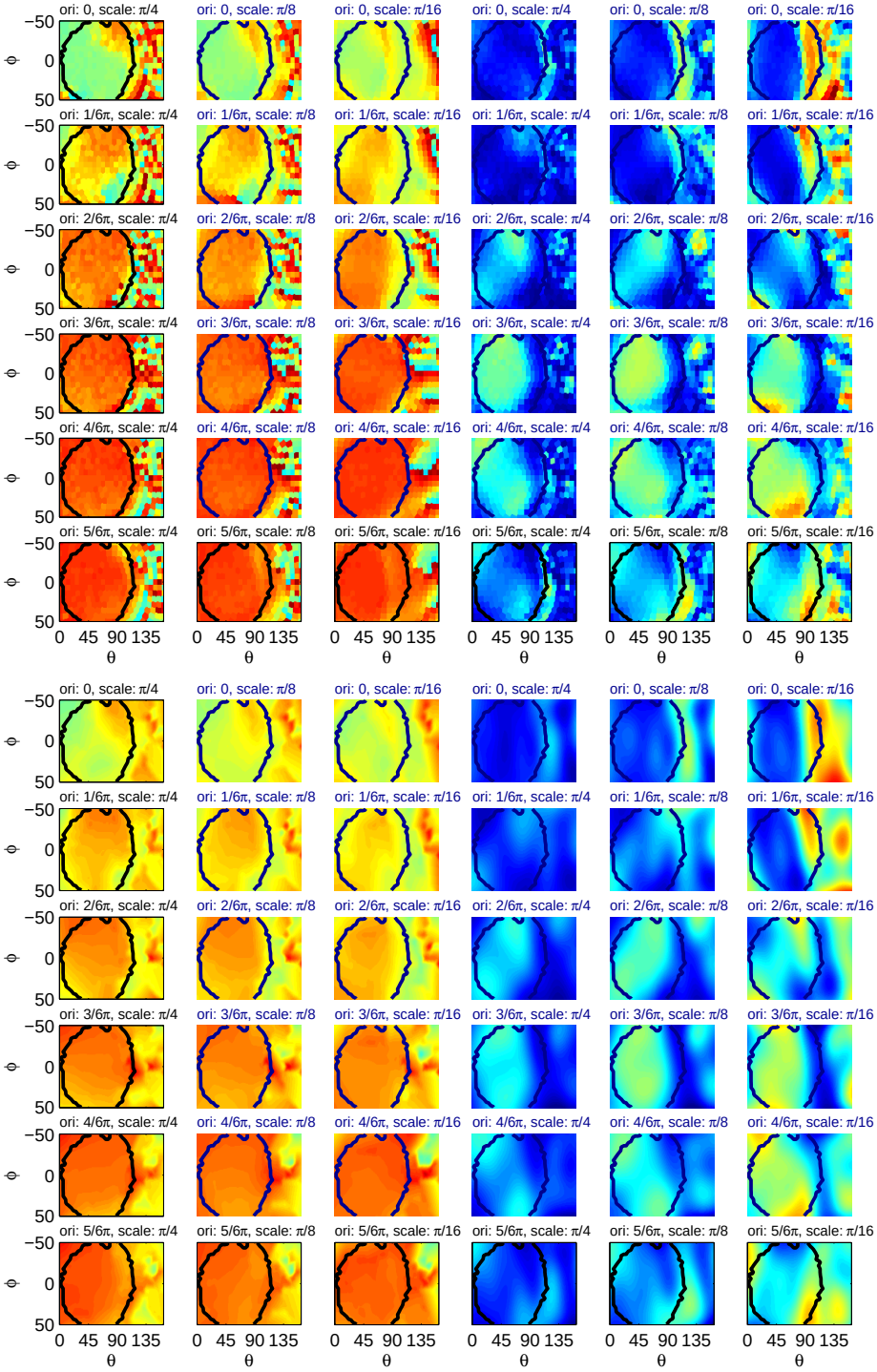


Figure 6.8: Blocks clockwise from upper left hand corner: Phase data, amplitude data, predictions of the amplitude model and predictions of the phase model. The average visibility region of the feature is denoted by the black line. Note how the phase data is quite smooth inside the visibility region, and spurious outside it.

Chapter 7

Pose estimation with random sampling

7.1 Introduction

In this chapter we consider the use of random sampling methods in object pose estimation problems. Section 7.2 discusses the shape of the likelihood functions in pose estimation and advocates the use of random sampling methods. Section 7.3 compares three different types of sampling algorithms in a problem in-plane rotations. Section 7.4 discusses the effect of steering correction of filter responses in the same in-plane rotation estimation problem. Finally, Section 7.5 considers the problem of estimating the model pose in the case of depth rotations.

The random sampling methods applied in this chapter include Metropolis, Gibbs and Population Monte Carlo (PMC) sampling algorithms introduced in Section 4.8. Population Monte Carlo methods are related to Sequential Monte Carlo (SMC) methods which have been applied in computer vision in tracking problems, such as in (Isard and Blake, 1998). Sequential Monte Carlo methods are applicable in dynamic state estimation problems, but in this chapter we consider static pose estimation problems and argue that Population Monte Carlo methods are applicable to them with better peformance than classic Metropolis and Gibbs sampling algorithms. Sequential matching of object features using SMC sampling has been proposed in (Tamminen and Lampinen, 2006), where it was found to perform well especially in handling feature occlusions.

Conceptually, our approach to the pose estimation problem is similar to (Lowe, 1989), where parameterized 3D object models are fitted into images, as both need relatively detailed, structured 3D models of the objects, and the pose estimate is based on the result of model fitting. Classical geometric methods such as (Haralick et al., 1989) and (Faugeras and Hebert, 1986) establish point correspodences between the model and the image, and solve for the pose

parameters directly or iteratively. Image-based object modeling methods such as (Lepetit et al., 2004) have been succesfully applied to the pose estimation problem. These are most effective if the object has a nearly planar shape in three dimensions.

The approach we take in this work in the 3D case is to learn the feature variation due to object pose from synthetic training data. Component-based face recognition using synthetic training data has been proposed by Huang et al. (2003), who use local gray level values directly as features and recognition is performed with a combination of Support Vector Machine classifiers. In this work, we use significantly more sophisticated feature models.

7.2 Object matching with in-plane rotations

In-plane rotations are difficult in local feature based recognition because the rotation parameter is global and affects the relative locations of all features simultaneously. In this section we aim to show that the likelihood distribution is typically multimodal with respect to the rotation and object location parameters, and local optimization methods do not solve the pose estimation problem reliably. Consequently, random sampling methods are applicable for finding the largest mode of the probability distribution, corresponding to the globally optimal pose parameters. Global optimization methods such as simulated annealing and genetic algorithms could be alternatively used while remaining in the error-minimization framework. The probabilistic approach we take in this work allows the use of a number of powerful random sampling algorithms which have been devised for statistical inference problems.

We will consider a rotation-invariant object matching system which can recognize objects undergoing in-plane rotations and changes in scale. Our shape model is very simple in order to highlight the differences of the sampling methods, and has only four parameters for the locations of the features: orientation angle θ, global scale s, and the center x_c, y_c of the feature configuration, as described in Section 4.7.1. These compose the parameter vector $\boldsymbol{\theta} = [\theta\ s\ x_c\ y_c]$. The locations of individual features are given by an overall object shape model M, which is simply the mean of the training data. In other words, the object shape model is rigid, and individual locations of features are not optimized. This is appropriate because the aim is in pose estimation, not person identification. Feature models are also computed as mean jets at the annotated locations of the training images. We have again used the DTU face database with 37 high resolution images in which 58 features are annotated in each, performing the tests with leave-one-out cross-validation where all other images except the one to be tested are used in training the model (computing the mean shape and mean filter responses). The rotated images are generated synthetically by simply rotating the original image

into new orientations.

Since displacement, scale and rotation parameters all affect directly the locations of the individual features, the parameterization makes things quite difficult although the number of parameters is as low as four. Typically the shape of the target probability distribution (i.e. the likelihood or posterior probability distribution of model parameters) is such that it has a very sharp peak at the parameter combination where all features are well matched. Additionally there are several weaker local maxima where only some of the features are matched, for example when the left eye features are matched at the right eye. The relative height of the peaks is also affected by the parameter β in the likelihood function (Eq. 4.5). High values of β cause the largest maxima to contain most of the probability mass, but also constraint the mass into a smaller region in the parameter space and thus make it harder to find. Outside the immediate vicinity of probability maxima, the probability distribution is often quite flat, especially with respect to the displacement parameters, and if the current estimates of orientation and scale parameters are not close to the true values, a wide variety of displacement parameters appear almost equally probable, giving little information about their true values.

Figure 7.1 illustrates the difficult shape of the pose probability distribution with respect to the displacement parameters at seven different pose angles. In order to simplify matters, the global scale of the object is assumed to be known. Starting with the pose $\theta = 20^\circ$, the highest probability peak occurs when the the left eye has been approximately matched. Let us suppose that we would use a local optimization scheme and adjust the orientation parameter slightly, simultaneously with the displacement parameters. The likelihood function grows larger when we decrease the orientation to $\theta = 17^\circ$ and yet more with $\theta = 13^\circ$. However, at this point the optimization path reaches a dead end. The left eye of the model has been matched approximately correctly, while the right eye of the model is on top of the eyebrow of the image. We have found a local maximum in the likelihood function: decreasing the orientation parameter any further lowers the likelihood peak.

At $\theta = 13^\circ$, correctly matching either the left or the right eye results in roughly the same probability, and there is a region of lower probability between the two peaks. Starting from the other peak, local optimization will lead us to the global maximum which is the correct solution and also the most probable one. The crux of the problem is that we will not find it simply by following an uphill path from the initial pose of $\theta = 20^\circ$.

Because of the typically difficult shape of the likelihood landscape (such as the one seen in the previous example), local optimization methods are often unable to find the global maximum. A computationally brute force solution for

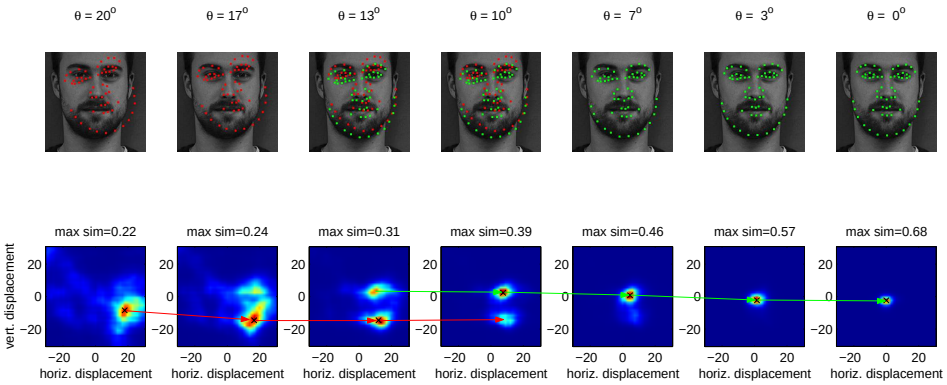


Figure 7.1: Bottom row: Two-dimensional slices of the object likelihood functions with different amounts of plane rotation θ in the matching model. The crosses denote the maxima of the similarity displacement parameters, and the red and green arrows show two local optimization paths advancing from one orientation to another. Average similarity value at the highest maximum are shown on top of each likelihood field. Top row: Feature locations corresponding to the local maxima at each orientation. See text for discussion.

finding the most probable combination of parameters would be to evaluate the target probability distribution at all possible parameter combinations. However, this turns out to be computationally very demanding with only four parameters. If we test 50 possibilities of displacement parameters in horizontal and vertical coordinates at 20 different scales and in 20 different orientations, we need to evaluate the target distribution function $50 \cdot 50 \cdot 20 \cdot 20 = 10^6$ times. In higher-dimensional parameter spaces this approach becomes quickly completely infeasible.

In order to tackle the exponential growth of the parameter space, we will apply Monte Carlo sampling methods which can lessen the computational cost if they manage to evaluate the target distribution mostly only in those regions of the parameters space where the target distribution has significant probability mass. As many of the modes of our target distribution are caused by incorrect partial matches of the object, our ultimate goal to find simply the largest mode of the target distribution. Accordingly, we choose a relatively large value for the parameter β, which controls the steepness of the likelihood function, so that most of the probability mass will be contained in the mode with highest probability density. In this sense our approach has common ground with global optimization methods.

A common problem with convergence of any Monte Carlo method is that depending on the choice of the proposal (jumping) distribution, the methods have a tendency either to converge into some single local mode of the probability distribution and remain stuck in there, or wander aimlessly in the parameter space,

never finding any of the modes with significant probability mass. Depending on the distribution to be simulated, the starting point can cause significant bias in the results, if the Markov chains systematically converge into some single modes of the distribution and are incapable in practice in traveling from one mode to another. These problems are made more severe by the fact that our likelihood function has a difficult shape compared to the distributions typically encountered in statistical inference.

7.3 Comparison of sampling methods

We compare the performance of Metropolis, Gibbs and Population Monte Carlo sampling methods in the in-plane rotation estimation problem. First we consider the case where the feature jets of the matching model are constant with respect to in-plane rotations. This makes the feature likelihood fields remain unchanged even when the orientation parameter changes. To save computational effort, we can precompute the feature likelihoods and use simple table lookup in the sampling stage. All sampling algorithms were initialized using samples from uniform distributions for orientation and displacement with bounds $\theta \in [-60°, 60°]$ and $x, y \in [-10, 10]$ from the image center, and a Gaussian distribution $N(1, 0.1^2)$ for global scale s. The scale is close to the correct value in order to help even the poorly performing sampling methods to converge to the correct solution. Our main interest here is the orientation parameter.

The Metropolis algorithm is easy to implement, and can be considered the baseline method. The proposal distribution is a Gaussian distribution with a diagonal covariance matrix with standard deviations $\sigma_\theta = 0.05$, $\sigma_s = 0.05$ and $\sigma_x = \sigma_y = 3$. The choice of the proposal distribution is crucial for the performance of the Metropolis algorithm, as new proposals should be small enough so that the chain will not turn into a blindly wandering search, but large enough to escape local maxima of the target distribution in search of the global maximum. The values for deviations above appeared to be suitable with respect to these concerns, but admittedly we did not systematically search for the optimal values for fast convergence of the chain.

Gibbs sampling is also very straightforward to implement. The full conditional distributions of the parameters do not have analytical expressions, but since we have already computed the individual feature likelihoods, numerical computation of the full conditional distributions is not computationally too expensive, as we can compute them simply as the products of values from the feature likelihood lookup tables. The ranges of the parameters were limited to $\theta \in [-60°, 60°]$, $s \in [0.7, 1.3]$ and $x, y \in [-20, 20]$, in which we evaluate the target distribution discretized in twenty steps for each parameter. Since the Gibbs sampler moves in orthogonal directions in the parameter space, it is prone

to having problems with traveling from one mode of the distribution to another, and is best applicable to unimodal estimation problems.

The implementation of the Population Monte Carlo sampler is somewhat more involved. We use the proposal distribution of the Metropolis sampler as a guideline when designing the generating distributions of the particles, so that the differences between results of the methods are not only due to different proposal distributions. In the basic version of the PMC sampler, each particle i of a single generation is generated independently and the proposal distribution of the Metropolis sampler is used as the generating distribution, i.e. a multivariate Gaussian distribution with covariances $\sigma_\theta = 0.05$, $\sigma_s = 0.05$ and $\sigma_x = \sigma_y = 3$.

The PMC sampler can be however made more efficient by clever selection of the generating distributions. Since the choice of the generating distributions for each particle is completely free, we can even use the evaluated values of the target distribution when generating the new proposals. The distribution to be simulated is problematic partly because the strong interconnections of the parameters, and we generate each new particle i of a single generation in an alternative version of the PMC sampler which uses a local feature based pose heuristic as follows:

- Evaluate the probabilities p_{ij} of each feature location j individually
- Generate new rotation angle θ_i^{new} and scale s_i^{new} parameters from $\theta \sim N(\theta_i^{old}, \sigma_\theta^2)$ and $s \sim N(s_i^{old}, \sigma_s^2)$ respectively
- Sample a feature location index j according to their individual probabilities
- Compute the rotated and scaled spatial feature locations, with the feature index j acting as the origin for rotation θ_i^{new} and scaling s_i^{new}
- Generate an additive displacement to the feature locations from the distributions $x \sim N(0, \sigma_x^2)$ and $y \sim N(0, \sigma_y^2)$

The motivation of this scheme is to generate parameter states in which well-matched feature locations are more likely to remain well-matched, as the rotation of the features is performed about a feature which has been matched with a good probability. Each particle is rotated, scaled and displaced independently with a different set of pose parameters, and multiple good candidates for the feature locations typically exist simultaneously and independently in a single generation of particles.

Figure 7.2 shows the samples generated by the three sampling methods. We have chosen a sampling run where all samplers have converged to the same mode of the target distribution. The Metropolis sampler finds very quickly a quite good parameter combination, and the move to the better mode requires a large jump in the orientation parameter. After this only very few of the proposals become accepted. The Gibbs sampler, which accepts every move, moves much more

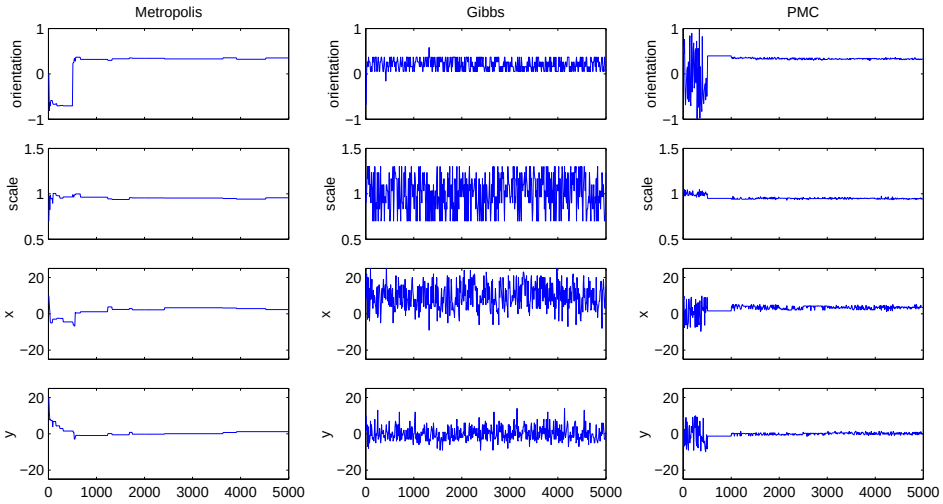


Figure 7.2: Samples generated by the different sampling methods. For clearer visualization, only every tenth sample has been plotted.

aggressively through the parameter space. The PMC sampler evaluates a wide range of possible parameter combinations in its first generation of 500 particles, in a manner not unlike importance sampling, and subsequent generations are merely small improvements on the good parameter combination found already in the first generation.

5000 iterations were performed with all three algorithms. Samples in the beginning of a Markov chain, which are biased by the initial values, are typically discarded when computing the results, a procedure which is called *burn-in*. In order to ensure convergence into a single mode, we used only 500 last samples of the chain when computing the sample averages. 500 particles and ten generations were used in the PMC sampler.

There are large differences in the capability of the samplers to find the mode in the probability distribution with significant mass. To evaluate this, we ran each sampler with all of the DTU images in five different orientations, rotated by -40, -20, 0, 20 and 40 degrees. The average distance of the features from their annotated locations were computed, and the sampler was deemed to have converged into the correct mode if the average distance of the features was less than ten pixels. Because the feature location model is stiff and includes only scale changes as deformations, the ten pixel difference in the locations is not unrealistically large.

Table 7.1 summarizes the results of the matching experiment. When converged, Metropolis, Gibbs and PMC samplers achieve an average distance of a

method	$P_{\pm 40^\circ}$	$P_{\pm 20^\circ}$	P_{0°	D_c (px)
Metropolis	0.51	0.81	0.84	5.79
Gibbs	0.42	0.54	0.62	5.94
PMC, no heuristic	0.92	0.93	0.95	6.22
PMC	0.93	0.97	0.97	5.77

Table 7.1: Probabilities of convergence to the correct mode with Metropolis, Gibbs and PMC methods (without and with the pose heuristic), and the average distance D_c of converged mean model matches from the annotated feature locations.

little under 6 pixels. The PMC sampler without pose heuristic appears to converge slower than the other samplers, and the average distance of features is larger than with the other samplers when the same number of iterations is performed.

The Gibbs sampler suffers most from the high correlation of the parameters and is often unable to find the mode with the significant probability mass especially in the case of rotated images. The Metropolis sampler performs better and converges to the mode close to the annotated solution with probability 0.84 in the case of unrotated images, but the probability of convergece to the correct solution decreases with rotated images. The PMC samplers, in contrast, are very efficient in finding the strongest mode near the annotated solution, and their performance is almost as good also with highly rotated images. The local feature based pose heuristic of the PMC is quite efficient in directing the sampler quickly to the different modes of the probability distribution. The correct mode is found without the pose heuristic almost equally well.

The presented results do not prove that Metropolis algorithm cannot function effectively in the pose estimation problem, because we have not systematically evaluated the performance with all possible proposal distributions. More effective proposal distributions are likely exist compared to the one we have used. Nevertheless, the PMC algorithm with the same proposal distribution is more effective than the Metropolis algorithm, and also more effective than the Gibbs algorithm. The results suggest that the PMC algorithm is more efficient in general compared to Metropolis and Gibbs algorithms, regardless of the choice of the proposal distributions.

The evolution of the particle generations in the PMC sampler can be seen in Figure 7.3, with 1000 particles in each generation. The descendants in a single generation are typically the offspring of only a handful of particles, due to the difficult spiky shape of the probability distribution. After a few generations, all of the particles in a single generation share a common ancestor.

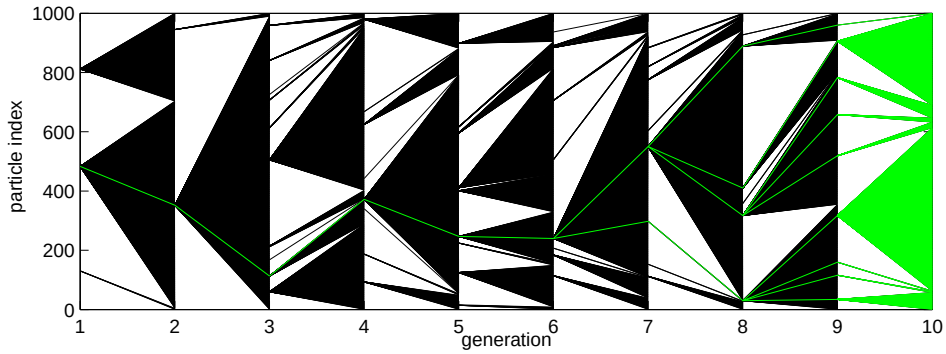


Figure 7.3: Ancestors (green) and descendants (black) of a typical Population Monte Carlo run.

7.4 Rotation-invariance in the feature level

In-plane rotations cause most significant changes to the configurations of the features, but, as noted before, also the filter responses change due to in-plane rotations. In order to account for these effects, we can use steerability to correct the feature responses into any given orientation. Unfortunately as the feature models now change depending on the value of the orientation parameter θ, the feature likelihood fields can no longer be precomputed, and consequently Gibbs sampling becomes computationally infeasible, as its numerical version requires up to a hundred times more evaluations of the target distribution for each random sample.

Figures 7.4 and 7.5 show matching results of three individuals in seven different rotation angles, without and with steering correction in the filter responses, using the PMC sampler. The matching results themselves are highly similar, but if we compute the average similarity values, it can be seen that while without steering correction the similarity values become quite low in the highly rotated orientations, they are nearly equal when steering correction is applied. While the feature detection stage is successful even without steering correction, the invariance of the similarity values in different orientations is crucial for recognition applications. The price of the better quality of the feature similarity values is however the increased computational burden, as we can no longer precompute the feature likelihood fields, and as the steering correction itself takes some time to compute.

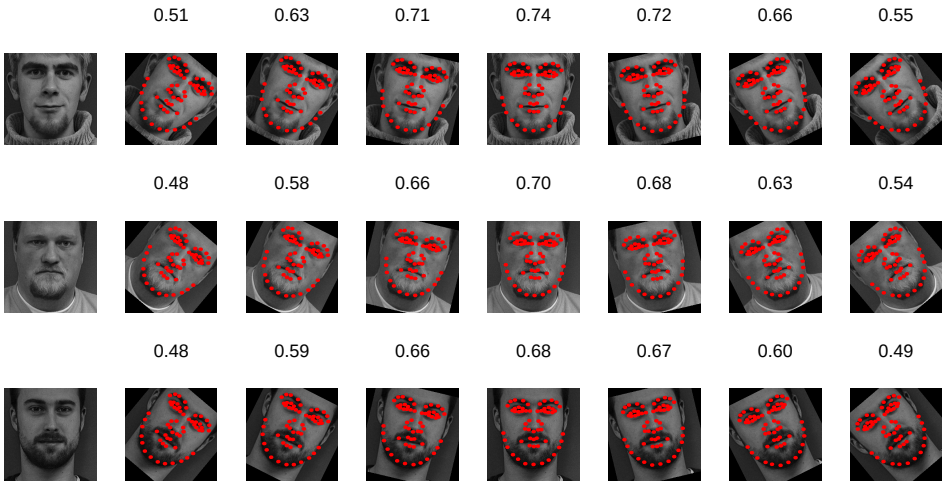


Figure 7.4: Rotation-invariant matching results with Population Monte Carlo without steering, with average similarity scores. The features are matched correctly in all tests, but the similarity scores are lower in rotated poses because the pose variation of features is not modeled. Compare with Figure 7.5.

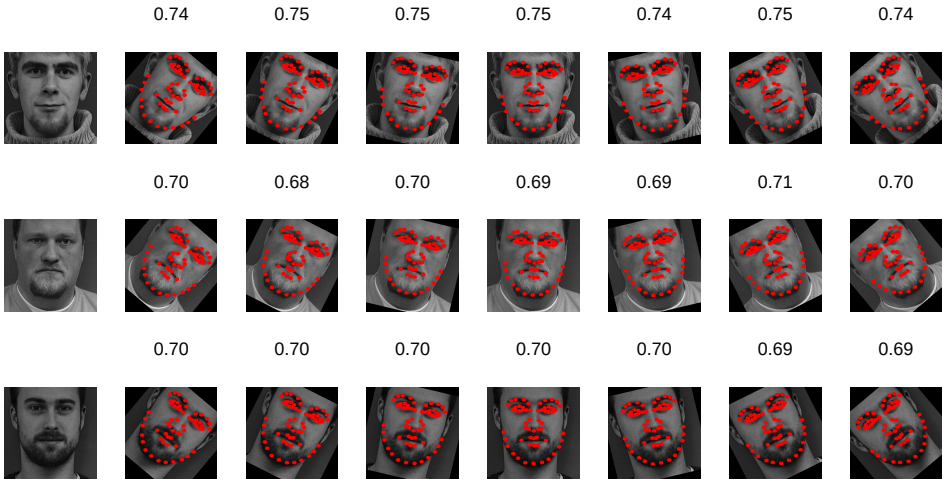


Figure 7.5: Rotation-invariant matching results with Population Monte Carlo using steering-corrected filter jets, with average similarity scores. The similarity scores are equally good in any orientation. Compare with Figure 7.4.

7.5 Object matching with depth rotations

Finally we consider the case with three independent rotation angles. The parameter vector $\boldsymbol{\theta} = [\psi\ \phi\ \theta\ s\ x\ y]$ consists of the azimuth angle ψ, the elevation angle ϕ, in-plane rotation angle θ, global scale s and displacement in the image plane x, y.

The object model M presented in Section 4.7.3 specifies the locations of individual features via the object pose parameters $\boldsymbol{\theta}$. In other words, the locations of individual features are not optimized, similarly to the previous two sections in this chapter. The predictions of the mixture of Gaussians regression model generated from synthetic data in Chapter 6 are used as the feature models which account for the depth rotations. Only the changes in feature locations due to in-plane rotations are modeled, and steering correction is not used.

As initial distributions for the parameters, we used $\psi \sim Unif(0, 2\pi)$, $\phi \sim Unif(-0.1, 0.1)$, $\theta \sim Unif(-0.1, 0.1)$, $s \sim N(1, 0.1^2)$ and $x, y \sim Unif(-50, 50)$, with the angles given in radians and displacement values in pixels. This choice of initial distributions initializes the particles of the PMC sampler so that nearly upright profile, half-profile and frontal poses are present in the initial particle distribution. This strategy is chosen because we cannot cover the whole pose space very well, and the azimuth angle presents most difficulties for the sampler as it is often quite difficult for the sampler to move away from a profile pose. In a well-mixing chain the choice of the initial distribution should not affect the results when the sampler has converged, but we want the chain to converge as soon as possible, and the choice of the initial distribution affects the speed of convergence to some degree.

The generating distributions for the new particles in the PMC sampler were set as follows: $\psi \sim N(\psi_{old}, 0.5^2)$, $\phi \sim N(\phi_{old}, 0.25^2)$, $\theta \sim N(\theta_{old}, 0.02^2)$, $s \sim N(s_{old}, 0.5^2)$ and $x, y \sim x_{old}, y_{old} + Unif(-3, 3)$. The rotations and scaling are again performed in the same manner as in the case of in-plane rotations for each particle separately. We first sample a feature index using the feature probabilities as weights, which acts as the origin for the rotation and scaling, and the displacement values are added to the feature locations after rotation and scaling. The variance in the generating distribution of θ is kept small, because the test images do not have variation in the in-plane angle, and only the angles which produce depth rotations (i.e. azimuth ψ and elevation ϕ) are varied. The other two angles have a sizable variance in their generating distributions in order to facilitate efficient travel of the chain from one mode to another. This is needed because in the case of depth rotations, profile poses with their strong edges often have some probability mass even when the correct solution does not correspond to a profile pose, and the sampler should be able to escape these modes in search of better ones. Ten generations and 500 particles in each generation were used. The sampling process takes approximately five minutes for a single image using

an unoptimized Matlab implementation on an Intel P4 1.7GHz based machine.

Figure 7.6 shows the matching results with a synthetic test model. The matching experiment was repeated ten times and the median graph in terms of the similarity has been plotted with a red line. The sampling variance in a single matching experiment is illustrated by plotting the graphs of one standard deviation estimates of the parameter samples with green lines. The PMC sampler has managed to find the correct solution in all poses.

Because of the probabilistic nature of the matching process, it is possible that the sampler gets stuck in an incorrect mode of the likelihood function. We did not systematically search for good initial distributions for the particles and generating distributions, which could improve the probability of convergence to the correct mode. Another possibility would be simply to increase the number of particles, although this approach is not very elegant.

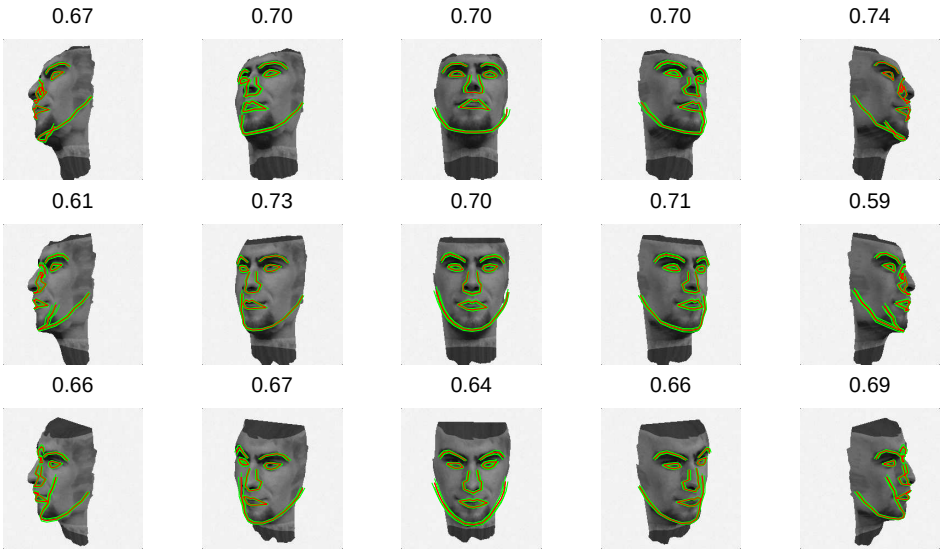


Figure 7.6: Median result of ten repeated matching experiments using the synthetic model, with average similarity scores on top of each pose. Graphs corresponding to one standard deviation of the samples in a single run have been plotted with green line.

The evolution of the PMC particles can be seen in Figure 7.7. At each generation the PMC algorithm first generates the candidate states (top row), evaluates their fitness and resamples them according to their probabilities, obtaining samples from the target distribution (bottom row). In the first generation the most probable particle corresponds to a profile pose. The sampler travels from this mode into the approximately correct solution after a few generations.

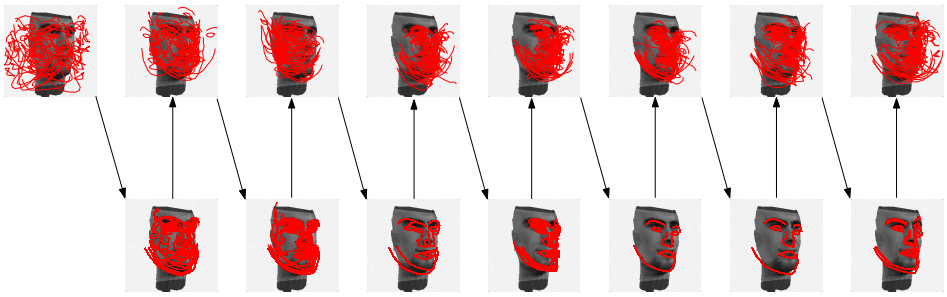


Figure 7.7: Evolution of the particles in pose-invariant object matching with three rotation angles. Leftmost image shows the samples from the initial distribution. The rest of the images in the upper row show particles from the generating distribution at each generation, and the lower row shows the samples obtained from the target distribution at each generation. Because of clarity only every 30th sample in the upper row has been plotted.

The rigid feature location model does not take into account variation in the locations of features due to identity and expression. Thus the locations are probably not accurate enough to serve as a basis for recognition of identity. Modeling of identity variation simultaneously with pose is not trivial, because the largest variations in feature locations due to identity and due to pose can be highly similar in non-extreme poses. However, for example a non-rigid probabilistic object model which allows the feature locations to deviate small distances away from a mean shape should be quite straightforward to implement.

Finally, the presented system is compared to the approach presented by Ba and Odobez (2004), which uses a probabilistic approach for head tracking and pose estimation. Particle filtering, which is closely related to Population Monte Carlo methods, is used for head tracking. For pose estimation, maximum a posteriori estimation is used. The method is tested using the PIE database which contains 13 different head poses. Strictly speaking, Ba and Odobez are performing pose classification, not pose estimation, as both the test data and the model assume that the poses are spaced 22.5° apart. Instead of continuous estimation, the pose is classified into 13 classes. The highest pose classification score reported is 94.8%.

Unfortunately our regression modeling approach is not well suited to data such as the PIE database, which contains a limited number of discrete poses, as the feature models need to be built from continuous data which was available only by using synthetic images, and the feature models do not typically generalize very well between highly different image databases, such as between synthetic and natural images. A possible problem in using the same set of poses for both training and testing the system is that the results can be overly positive compared to test data with continuos pose changes. We would like to emphasize that object pose is

a continuous phenomenon and pose estimation error should be measured so that either the training data, the testing data, or preferably both contain poses which adequately cover the whole pose space. Admittedly, a clear problem limiting the applicability of the presented approach employing regression modeling of the features under depth rotations is that it would be laborious to gather the required dense training data in the case of real-world images, and often one is forced to use only sparse data such as the PIE database.

Using a gallery of 13 test poses similar to the PIE database, but synthetic images from 20 individuals generated according to Section 6.5, the presented system achieves pose estimation error with the mean $\mu_\theta = 4.4°$, $\mu_\phi = -1.7°$ and standard deviation $\sigma_\theta = 18.3°$, $\sigma_\phi = 7.3°$. When the sampling process converges to the correct mode in the probability distribution, the estimates are reasonably accurate with only small bias. The biggest problem in the presented approach is that when the random sampler does not converge to the correct pose, the error in pose angle estimates can be very large.

Although a direct comparison to results obtained with the PIE database is not meaningful, combining the results above with a very simple nearest neighbor classifier based only on the estimated angles, the presented system achieves a pose recognition rate of 85% with the same set of test poses. Profile and half-profile poses are recognized more accurately, and misclassifications are most common in near-frontal poses. This is a direct consequence of the feature locations we have chosen, which remain relatively constant in a frontal nodding movement. A local feature based head pose estimation system would benefit from feature points outside the inner face region, which would make it possible to infer the pose more accurately based on the locations of the inner face features in relation to the head border, for example.

The obtained results show that synthetic data can be applied in learning the feature models, and reasonably accurate pose classification is possible even when the quality of the synthetic models is not very high and contains some spurious effects. It should be noted that the synthetic data which has been used is not necessarily easier to classify than real world data obtained in constrained conditions.

Chapter 8

Conclusions

This thesis has presented a complete rotation-invariant object matching system, employing a local feature based object representation with parameterized models for the changes in features due to rotations, and algorithms using random sampling methods for fitting the models to data. Additionally, the effects of filter shape parameters on both recognition performance and rotation invariance have been studied.

The analytic derivation of steering functions for Gabor-type filters may be considered the most important theoretic result of the thesis, as it enables their use as steerable filters. Gabor filters are very widely employed in various applications, and the derived results give the opportunity to consider in-plane rotation invariance without changing the filters of the system and possibly affecting the performance of the complete system.

The experiments performed with filter parameters suggest that good steerability and recognition performance are conflicting design goals. Best steerability is obtained with Gabor and angular Gaussian filters which have mediocre recognition performance and vice versa. The best filter banks for rotation-invariant recognition of human facial features require many more basis filters than what is necessary for computing the principal orientation of simple edges.

The presented object matching system is able to successfully solve recognition problems involving in-plane rotations. The PMC algorithm was found to be clearly the most efficient, whereas the standard Metropolis and Gibbs sampling algorithms produce only mediocre results, their main problem being the poor probability of convergence to the mode of the probability distribution corresponding to the correct solution. The case of depth rotations is significantly more difficult, and also the PMC algorithm begins to have problems finding the correct solution.

In the object matching scheme presented in Chapter 7 we used a top-down approach, finding the most probable pose parameters of a complete, detailed

object model. Using the same kind of local feature models, it would be possible to first try to detect parts of the object and their pose, and use these to generate efficient proposal distributions for the random sampling algorithms. Filter jets at single locations may be too generic for this purpose, and combinations of several spatially separated jets could be used, as in (Yokono and Poggio, 2004a).

Although perceptually very important, human faces are only one of the numerous object classes humans can distinguish. Good filter parameters for recognition are necessarily dependent on the object class, and it may well be that even different features of the human face would benefit from differently-shaped filters. Thus the aim of using the best generic filtering operation even in the case of a single object class is a computational compromise. For truly optimal performance one might have to employ several filter banks with different shape parameters, and it becomes even more difficult to find the good parameter sets. It is perhaps interesting to note that while it has been known for a long time that the mammalian visual cortex contains a plethora of filters with various orientation and scale sensitivity profiles, computer vision algorithms using Gabor filters typically use only a single profile, usually either a spherical or a biologically motivated $\sigma_y : \sigma_x = 2 : 1$ one. It would be interesting to know if there anything to be gained in considering simultaneously the responses of filters with different orientation profiles, in some sense circumventing the limits which the uncertainty principle poses for a single oriented filter.

The probabilistic approach that has been used in the work provides a unifying theoretical foundation for object matching and merits further research. The Population Monte Carlo class of sampling algorithms is especially interesting since powerful heuristics about the specific problem can be used in choosing the generating distributions, while the samples themselves obtained by the PMC algorithms are guaranteed to follow the target distribution. The pose estimation method presented in this thesis is computationally rather demanding compared to the methods found in the literature such as (Lepetit et al., 2004), requiring at least hundreds of iterations for convergence, and its main use remains currently in theoretical considerations. Nevertheless, probabilistic algorithms using random sampling can be computationally rather efficient in demanding estimation problems, and can be applied even when most other approaches are intractable.

Synthetic data has been applied the work for learning the pose variation in features. Because the visual quality of the synthetic data is not completely lifelike, the feature models do not directly predict features in natural images very well. Significant improvement has occurred in computer graphics even during the course of this work, especially in modeling hair and skin, and the approach to use synthetic learning data in visual tasks is becoming more and more appealing. The primary benefit for doing so instead of using real measured data is that the gathering of training data, which is a major hurdle in computer vision research in

general, is significantly less laborious.

The aims set for the thesis – to develop pose-invariant methods for local feature based object matching systems and analyze their performance – have been achieved, although several unsolved issues remain. In addition, while oriented filters have been studied for some time, their use continues to be a relevant research topic in computer vision.

Appendix A

Image databases

ORL database

Figure A.1: Test images of the ORL database.

BioID database

Figure A.2: Test images of the partial BioID database.

References

Ba, S. O. and Odobez, J.-M. (2004). A probabilistic framework for joint head tracking and pose estimation. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 264–267, Washington, DC, USA. IEEE Computer Society.

Bicego, M., Lagorio, A., Grosso, E., and Tistarelli, M. (2006). On the Use of SIFT Features for Face Authentication. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, pages 35–41, Washington, DC, USA. IEEE Computer Society.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In Rockwood, A., editor, *Proceedings of SIGGRAPH 1999*, volume 33, pages 187–194, Los Angeles. Addison Wesley Longman.

Boukerroui, D., Noble, J. A., and Brady, M. (2004). On the choice of band-pass quadrature filters. *J. Math. Imaging Vis.*, 21(1):53–80.

Bovik, A. C., Clark, M., and Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):55–73.

Bulow, T. (1999). Hypercomplex spectral signal representations for image processing and analysis. PhD thesis, University of Kiel.

Burns, J. B., Weiss, R. S., and Riseman, E. M. (1993). View variation of point-set and line-segment features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):51–68.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698.

Cappe, O., Guillin, A., Marin, J.-M., and Robert, C. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685.

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. on Information Theory*, 36(5):961–1005.

Daugman, J. G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.*, 2(7):1160–1169.

Daugman, J. G. (1988). Complete discrete 2-D Gabor transformations by neural networks for image analysis and compression. *IEEE Trans. ASSP*, 36(7):1169–1179.

Davis, R. A., McNamara, D. E., Cottrell, D. M., and Campos, J. (2000). Image processing with the radial Hilbert transform: theory and experiments. *Optics Letters*, 25:99–101.

Dunn, D. and Higgins, W. (1995). Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7):947–964.

Edelman, S. (1996). Why have lateral connections in the visual cortex? In Sirosh, J., Miikkulainen, R., and Choe, Y., editors, *Lateral Interactions in the Cortex: Structure and Function*, chapter 12. The UTCS Neural Network Research Group, Austin, TX, http://www.cs.utexas.edu/users/nn/web-_pubs/htmlbook96/. Electronic book, ISBN 0-9647060-0-8.

Faugeras, O. D. and Hebert, M. (1986). The representation, recognition, and locating of 3-d objects. *Int. J. Rob. Res.*, 5(3):27–52.

Fei-Fei, L., Fergus, R., and Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA. IEEE Computer Society.

Felsberg, M. and Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4(12):2379–2394.

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *Int. J. Comput. Vision*, 5(1):77–104.

Fleet, D. J., Jepson, A. D., and Jenkin, M. R. M. (1991). Phase-Based Disparity Measurement. *CVGIP: Image Understanding*, 53:198–210.

Fliege, N. J. (1993). *Multiraten-Signalverarbeitung. Theorie und Anwendungen.* Teubner Verlag.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Trans. PAMI*, 13(9):891–906.

Gabor, D. (1946). Theory of communication. *J. Inst. Elec. Eng*, 93:429–457.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Capman & Hall/CRC, second edition.

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

Gong, S., McKenna, S., and Collins, J. J. (1996). An Investigation into Face Pose Distributions. In *Second International Conference on Automated Face and Gesture Recognition*, Killington, Vermont.

Granlund, G. H. (1978). In search of a general picture processing operator. *Computer Graphics and Image Processing*, 20:155–173.

Greenberg, S., Aladjem, M., and Kogan, D. (2002). Fingerprint image enhancement using filtering techniques. *Real-Time Imaging*, 8(3):227–236.

Greenspan, H., Belongie, S., Perona, P., Goodman, R., Rakshit, S., and Anderson, C. (1994). Overcomplete steerable pyramid filters and rotation invariance. In *CVPR94*, pages 222–228.

Haley, G. M. and Manjunath, B. S. (1995). Rotation-invariant texture classification using modified Gabor filters. In *Proc. Second IEEE international conference on image processing (ICIP'95),*, volume 1, pages 262–265.

Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., and Kim, M. (1989). Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446.

Harris, C. and Stephens, M. (1988). A Combined Corner and Edge Detector. In *4th ALVEY Vision Conference*, pages 147–151.

Hel-Or, Y. and Teo, P. C. (1998). Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9(1):83–95.

Huang, J., Heisele, B., and Blanz, V. (2003). Component-based face recognition with 3d morphable models. In Kittler, J. and Nixon, M. S., editors, *AVBPA*, volume 2688 of *Lecture Notes in Computer Science*, pages 27–34. Springer.

Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.

Jacob, M. and Unser, M. (2004). Design of steerable filters for feature detection using Canny-like criteria. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1007–1019.

Kalliomäki, I. and Lampinen, J. (2003). Modeling of pose effects in oriented filter responses for head pose estimation. In Bigün, J. and Gustavsson, T., editors, *SCIA*, volume 2749 of *Lecture Notes in Computer Science*, pages 156–162. Springer.

Kalliomäki, I. and Lampinen, J. (2005). Approximate steerability of gabor filters for feature detection. In *Image Analysis: 14th Scandinavian Conference, SCIA 2005*, pages 940–949.

Kalliomäki, I. and Lampinen, J. (2007). On steerability of Gabor-type filters for feature detection. *Pattern Recognition Letters*, 28(8):904–911.

Knutsson, H. and Granlund, G. H. (1983). Texture Analysis Using Two-Dimensional Quadrature Filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, Pasadena.

Knutsson, H., Wilson, R., and Granlund, G. H. (1983). Anisotropic Non-stationary Image Estimation and its Applications — Part I: Restoration of Noisy Images. *IEEE Trans on Communications*, COM–31(3):388–397. Report LiTH-ISY-I-0462, Linköping University, Sweden, 1981.

Kovesi, P. (1999). Image features from phase congruency. *Videre: A Journal of Computer Vision Research. MIT Press*, 1(3).

Krüger, V. (2001). *Gabor Wavelet Networks for Object Representation*. PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel.

Kruizinga, P. and Petkov, N. (1999). Non-linear operator for oriented texture. *IEEE Trans. on Image Processing*, 8(10):1395–1407.

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

Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311.

Lee, T. S. (1996). Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971.

Lepetit, V., Pilet, J., and Fua, P. (2004). Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 2, pages 244–250.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer.

Liu, Z. (2003). A fully automatic system to model faces from a single image. Technical Report MST-TR-2003-55, Microsoft Research.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110.

Lowe, D. G. (1989). Fitting parameterized 3D models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450.

Marzban, C. (2004). The roc curve and the area under it as a performance measure. *Weather and Forecasting*, 19(6):1106–1114.

Maurer, T. and von der Malsburg, C. (1995). Single-view based recognition of faces rotated in depth. In *Proceedings of Internationational Conference on Automatic Face and Gesture Recognition, 1995*, pages 248–253.

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.

Michaelis, M. and Sommer, G. (1995). A Lie Group approach to steerable filters. *Pattern Recognition Letters*, 16(11):1165–1174.

Mikolajczyk, K., Leibe, B., , and Schiele, B. (2006). Multiple object class detection with a generative model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

Moreno, P., Bernardino, A., and Santos-Victor, J. (2005). Gabor parameter selection for local feature detection. In Marques, J. S., de la Blanca, N. P., and Pina, P., editors, *IbPRIA (1)*, volume 3522 of *Lecture Notes in Computer Science*, pages 11–19. Springer.

Ng, R., Lu, G., and Zhang, D. (2005). Performance study of gabor filters and rotation invariant gabor filters. In *Proceedings of the 11th International Multimedia Modeling Conference*.

Okada, K. and von der Malsburg, C. (2001). Analysis and synthesis of human faces with pose variations by a parametric piecewise linear subspace method. In Proc. of IEEE Conf. on CVPR, Kauai.

Oppenheim, A. V. and Lim, J. S. (1981). The importance of phase in signals. *IEEE Proceedings*, 69:529–541.

Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-time signal processing (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and Modular Eigenspaces for Face Recognition. Proc. of IEEE Conf. on CVPR, Seattle, WA.

Perona, P. (1995). Deformable kernels for early vision. *IEEE Trans. PAMI*, 17(5):488–499.

Ro, Y. M., Kim, M., Kang, H. K., Manjunath, B. S., and Kim, J. (2001). Mpeg-7 homogenous texture descriptor. *ETRI Journal*, 32(2):41–51.

Ronse, C. (1993). On idempotence and related requirements in edge detection. *IEEE Trans. PAMI*, 15(5):484–491.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2003). 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 272–277.

Ruderman, D. and Bialek, W. (1994). Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–818.

Schenk, V. U. B. and Brady, M. (2003). Improving phase-congruency based feature detection through automatic scale-selection. In *CIARP*, pages 121–128.

S.E. Grigorescu, N. P. and Kruizinga, P. (2002). Comparison of texture features based on gabor filters. *IEEE Trans. on Image Processing*, 11(10):1160–1167.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Trans Pattern Anal Mach Intell*, 29(3):411–426.

Shen, L. and Bai, L. (2006). A review on gabor wavelets for face recognition. *Pattern Anal. Appl.*, 9(2):273–292.

Shi, B. E. (1999). 2D focal plane steerable and scalable cortical filters. In *MICRONEURO '99: Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, pages 232–239, Washington, DC, USA. IEEE Computer Society.

Simoncelli, E. P. and Adelson, E. H. (1990). Subband transforms. In Woods, J. W., editor, *Subband Image Coding*. Kluwer Academic Publishers, Norwell, MA.

Simoncelli, E. P. and Farid, H. (1995). Steerable wedge filters. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 189–194. IEEE Computer Society.

Simoncelli, E. P. and Freeman, W. T. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA.

Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. (1992). Shitfable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607.

Smith, J. O. (2003). *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, http://www.w3k.org/books/.

Sommer, G., Michaelis, M., and Herpers, R. (1998). The SVD approach for steerable filter design. In *Proc. Int. Symposium on Circuits and Systems 1998*, volume 5, pages 349–353, Monterey, California.

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, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.

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

Tae-Kyun, K. and Kittler, J. (2005). Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):318–327.

Tamminen, T. (2005). *Models and Methods for Bayesian Object Matching*. PhD thesis, Helsinki University of Technology. Report B52.

Tamminen, T. and Lampinen, J. (2006). Sequential monte carlo for bayesian matching of objects with occlusions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):930–941.

Teo, P. C. (1998). *Theory and Applications of Steerable Functions*. Ph.D. thesis, Technical Report CS-TR-98-1604, Stanford University.

Teo, P. C. and Hel-Or, Y. (1999). Design of multi-parameter steerable functions using cascade basis reduction. *IEEE Trans. PAMI*, 21(6):552–556.

Vetter, T. (1996). Learning novel views to a single face image. In *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, pages 22–27, Washington, DC, USA. IEEE Computer Society.

Weyrauch, B., Huang, J., Heisele, B., and Blanz, V. (2004). Component-based face recognition with 3d morphable models. In *The First IEEE Workshop on Face Processing in Video (FPIV 2004)*, pages 1–5, Washington, D.C, USA. IEEE.

Williams, C. K. I. (2005). How to pretend that correlated variables are independent by using difference observations. *Neural Comput.*, 17(1):1–6.

Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. (1999). Face recognition by Elastic Bunch Graph Matching. In Jain, L. C., Halici, U., Hayashi, I., and Lee, S. B., editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter 11, pages 355–396. CRC Press.

Wu, P., Manjunath, B., Newsam, S., and Shin, H. (2000). A texture descriptor for browsing and similarity retrieval. *Journal of Signal Processing: Image Communication*, 16(1-2):33–43.

Yokono, J. and Poggio, T. (2004a). Evaluation of sets of oriented and non-oriented receptive fields as local descriptors. Technical Report AI Memo 2004-007, CBCL Memo 237, Massachusetts Institute of Technology.

Yokono, J. and Poggio, T. (2004b). Oriented filters for object recognition: an empirical study. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 755–760.

Yu, W., Daniilidis, K., and Sommer, G. (2001). Approximate orientation steerability based on angular Gaussians. *IEEE Transactions on Image Processing*, 10(2):193–205.

Zhang, B., Shan, S., Chen, X., and Gao, W. (2007). Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. *IEEE Trans. on Image Processing*, 16(1):57–68.

Zhang, Z. (2001). Image-based modeling of objects and human faces. Proceedings of SPIE vol. 4309, Videometrics and optical methods for 3D shape measurement, pp. 1-15.

ISBN 978-951-22-8995-0 (printed)
ISBN 978-951-22-8996-7 (PDF)
ISSN 1455-0474
Picaset Oy, Helsinki 2007