LAPPEENRANTA
UNIVERSITY OF TECHNOLOGY

Jarmo Ilonen

SUPERVISED LOCAL IMAGE FEATURE DETECTION

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1383 at Lappeenranta University of Technology, Lappeenranta, Finland on the 7th of December, 2007, at noon.

Acta Universitatis
Lappeenrantaensis
282

Supervisors Professor Heikki Kälviäinen
Docent D.Sc. (Tech) Joni-Kristian Kämäräinen

Machine Vision and Pattern Recognition Research Group
Department of Information Technology
Lappeenranta University of Technology
Finland

Reviewers Ph.D. Krystian Mikolajczyk
Centre for Vision, Speech and Signal Processing (CVSSP)
School of Electronics & Physical Sciences
University of Surrey
United Kingdom

Associate professor Pavel Zemcik
Department of Computer Graphics and Multimedia (DCGM)
Faculty of Information Technology
Brno University of Technology
Czech Republic

Opponent Associate professor Jiri Matas
The Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University, Prague
Czech Republic

Professor Matti Pietikäinen
Department of Electrical and Information Engineering
University of Oulu
Finland

ISBN 978-952-214-466-9
ISBN 978-952-214-467-6 (PDF)
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
Digipaino 2007

Preface

The work presented in this thesis has been carried out in the machine vision and pattern recognition research group at the Laboratory of Information Processing in the Department of Information Technology at Lappeenranta University of Technology, Finland, during the years 2003-2007.

First I would like to thank my supervisors, Heikki Kälviäinen and Joni Kämäräinen. Their roles were quite different, Joni providing much of the scientific supervision and Heikki the financial and practical supervision. Together they created a solid basis (not a frame) for the work.

I would like to thank my co-authors, Pekka Paalanen from our laboratory, Miroslav Hamouz and Josef Kittler from the University of Surrey, and Tuomo Lindh, Jero Ahola and Jarmo Partanen from department of electrical engineering. While not sharing his name with me in any of the articles referenced here, I would also like to thank Ville Kyrki for his part in the earlier development of the methods presented here. Valuable comments from the reviewers, Krystian Mikolajczyk and Pavel Zemcik, were helpful and I want to thank them for their part in improving this thesis.

The good spirit of the laboratory helped me enormously during occasionally frustrating research and writing. The spirit would not exist without the fine people here whom I would like to thank, especially Jani, Jarkko, Leena, Pekka, Saku, Tomi, Toni, Tuomas, and Ville. And of course all the other friends, from irc and real life.

The financial support of the East Finland Graduate School in Computer Science and Engineering (ECSE), the Research Foundation of Lappeenranta University of Technology (LTY:n tukisäätiö) and Tekniikan Edistämissäätiö is gratefully acknowledged.

Finally, I would like to thank my family for supportive yet autonomous atmosphere, giving me free hands to pursue my interests.

"Not only do I not know the answer
I don't even know what the question is",
– Metallica, "My world"

Lappeenranta, November 2007

Jarmo Ilonen

Abstract

Jarmo Ilonen
Supervised local image feature detection
Lappeenranta, 2007
116 p.

Acta Universitatis Lappeenrantaensis 282
Diss. Lappeenranta University of Technology

ISBN 978-952-214-466-9
ISBN 978-952-214-467-6 (PDF)
ISSN 1456-4491

This thesis is about detection of local image features. The research topic belongs to the wider area of object detection, which is a machine vision and pattern recognition problem where an object must be detected (located) in an image. State-of-the-art object detection methods often divide the problem into separate interest point detection and local image description steps, but in this thesis a different technique is used, leading to higher quality image features which enable more precise localization. Instead of using interest point detection the landmark positions are marked manually. Therefore, the quality of the image features is not limited by the interest point detection phase and the learning of image features is simplified.

The approach combines both interest point detection and local description into one phase for detection. Computational efficiency of the descriptor is therefore important, leaving out many of the commonly used descriptors as unsuitably heavy. Multiresolution Gabor features has been the main descriptor in this thesis and improving their efficiency is a significant part. Actual image features are formed from descriptors by using a classifier which can then recognize similar looking patches in new images. The main classifier is based on Gaussian mixture models. Classifiers are used in one-class classifier configuration where there are only positive training samples without explicit background class.

The local image feature detection method has been tested with two freely available face detection databases and a proprietary license plate database. The localization performance was very good in these experiments. Other applications applying the same underlying techniques are also presented, including object categorization and fault detection.

Keywords: Gabor filters, multiresolution filtering, object detection, computer vision, machine vision, pattern recognition

UDC 004.93'1

$I(x, y)$	Intensity image
$D(x, y, \sigma)$	Difference of Gaussians
p	Crossing point between adjacent Gabor filters
k	Scaling factor for Gabor filter frequencies
γ	Gabor filter sharpness along major axis
η	Gabor filter sharpness along minor axis
m	Number of filters in different frequencies
n	Number of filters in different orientations
f_{min}	Tuning frequency of the lowest frequency Gabor filter
f_{max}	Tuning frequency of the highest frequency Gabor filter
f_{high}	The highest frequency included in Gabor filter
a_{sf}	Scaling factor for image
θ	Rotation of Gabor filter
$\psi(t; f_0)$	1D Gabor filter in spatial domain
$\Psi(u; f_0)$	1D Gabor filter in frequency domain
$\psi(x, y; f_0, \theta)$	2D Gabor filter in spatial domain
$\Psi(u, v; f_0, \theta)$	2D Gabor filter in frequency domain
$\xi(x, y)$	Image function
$r_\xi(x, y; f, \theta)$	Gabor responses for image $\xi(x, y)$
$\mathbf{G}$	Simple Gabor feature matrix
$N(t; \mu, \sigma)$	Gaussian (normal) distribution
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$	Multidimensional normal distribution
$G(x; \mu, \sigma)$	Cumulative Gaussian function
$G^{-1}(p; \mu, \sigma)$	Cumulative inverse Gaussian function
x_s, x_f	Envelope endpoints for spatial and frequency domain Gabor filters
$\mathcal{R}$	Confidence region
τ	Pdf value at border of confidence region
κ	Quantile value, confidence $c = 1 - \kappa$
c	Confidence value
$k(x, x')$	Kernel function
σ	The sharpness of RBF kernel
ν	Parameter controlling number of outliers in SVM classifier
α_i	Support vector weights
ρ	Margin to the hyperplane

1D	One dimensional
2D	Two dimensional
EM	Expectation maximization
DFT	Discrete Fourier transform
FFT	Fast Fourier transform
GMM	Gaussian mixture model
HDR	High density region
IFFT	Inverse Fast Fourier transform
LBP	Local binary pattern
MSER	Maximally stable extremal regions
PCA	Principal component analysis
pdf	Probability density function
RBF	Radial basis function
ROC	Receiver operating characteristic
SIFT	Scale-invariant feature transform
SOM	Self-organizing map
SVM	Support vector machine

Chapter I

Introduction

1.1 Object detection and localization

Object detection is a computer vision task where presence and location of an object is determined from an image. Object detection methods are useful in various problems, e.g. license plate detection and recognition [11], face detection [32] and detection of aerial targets [75]. Often methods for specific applications exploit application specific information to succeed, for instance, skin color in face detection. Therefore, object detection has been a disconnected field of study applying a variety of techniques. However, lately object detection approaches have started to converge; many new generic object detection methods use local image features to describe local appearance of an image and combine these local features with a model capturing their geometric relations, together creating a complete object description.

Even with this "parts and structure" approach of object detection there are still many different types of actual methods and some of their central differences are listed here. First is the importance of localization. Some object detection methods concentrate on detecting presence of the object, is it there at all, and exact localization is of secondary importance, while for some other methods accurate localization is important. Second is whether the method tries to detect always the same object or more generally a class of objects. Related to this is whether the method explicitly considers detection of multiple object classes or only a single class at a time. Third is the level of supervision, manual labor, that is required for training the detector.

This thesis presents a supervised method for localizing local image features efficiently and accurately for one object class at a time. A supervised method is used instead of a more fashionable semi-supervised approach to maximize the quality of local image features, and consequently detection performance. Semi-supervised methods start with an interest point detector to detect "interesting" or salient parts of an image, then create local descriptions for the detected parts and finally try to select automatically local features which are shared by all objects of the object class. With the supervised method the task is considerably more simple. Complex combination of interest point detection,

description and model creation is not needed because we *know* the important points and their spatial relationships as they are manually marked. Now, during detection with our supervised method the local descriptor has to perform the function of determining, for instance, "does this point look like an eye", while a method which separates interest point detection and local description first decides "does this point look interesting" and then "is this point similar to some of the image features we know".

In our case the interest points are reliable because we know they really are related to the object class, while the quality of image features determined by the semi-supervised methods is not guaranteed. It is possible that a semi-supervised method returns points commonly found in the background, for example, traffic signs are common in images with cars, therefore a method can decide that the presence of a traffic sign is related to the presence of the car. Furthermore, as human knowledge is used when common points are selected, the selection is not limited by what is deemed interesting by an interest point detection method.

1.2 Contributions and publications

A central contribution is the efficiency improvements of Gabor filtering. Compared to our older implementation the speed has improved by a factor of 50. This work has been published as a comprehensive research report, [35], and a shorter version has been published in a conference, [36]. This thesis also includes novel research on properties of complex-valued Gabor feature space.

Another important contribution is computation of confidence values for Gaussian mixture models (GMM), which converts the arbitrarily scaled probability density function (pdf) values to a probability score. One conference article has been published about this [39], and there is a journal article about GMM's in general [68].

In this thesis a supervised method for local image feature detection is proposed which is based on multiresolution Gabor features and their ranking using Gaussian mixture models. One conference article related to the method proposed in this thesis has been published, [43], and it includes also face detection experiments. There is also a journal submission, accepted with minor changes, about large parts of the complete work [38]. The proposed method has been applied to many important object detection tasks, such as face and license plate detection with excellent results.

Based on the image feature detection method an alternate application was developed for visual categorization of objects. The categorization is based on multiresolution Gabor features and their self-organization and has been published in [34]. Another application of multiresolution Gabor features and GMM classifier was developed for fault detection in electrical motors and has been published as a journal article, [37].

In these publications, the author has made a major contribution to the development and writing in [34, 35, 36, 37, 39], performed experiments, participated in the development and writing in [43, 38] and had a minor writing contribution in [68].

1.3 Thesis outline

Chapter 2 reviews methods related to local image features used in object detection. The chapter is divided to three parts: interest point detection, local image description, and complete object detection methods. The division is natural as many of the detection methods clearly separate interest point detection and local image description; in our own method the local image descriptor operates also as the interest point detector.

Multiresolution Gabor features are the topic of **Chapter 3**. The chapter first introduces Gabor features in both one and two dimensions, and then describes their efficient implementation and an implementation framework. Experimental results related to efficiency improvements are presented here.

Chapter 4 presents information about classification and feature ranking of local image features. One-class classifiers and their requirements are first described and then the chapter describes algorithms for creating local image feature detectors and how they can be used for detection. Also included is information on properties of multiresolution Gabor feature space, as it has been noticed to be occasionally challenging for classifiers.

Chapter 5 presents experiments and applications of multiresolution Gabor filter responses in various tasks. The main experiment has been object detection (face and license plate detection). Other applications are visual categorization of objects based on local image features and their spatial configuration and an approach for fault detection in electrical motors, which is based on 1D Gabor features and their classification.

Finally, **Chapter 6** discusses what was achieved in this thesis, and the strengths and weaknesses of our proposed image feature detection approach.

Chapter II

Local image feature detection

In pattern recognition *features* are numeric or symbolic units of information constructed from measurements by sensors. In case of images *image features* contain information of the image content; the information can represent small parts of the image (local image features), or the whole image (global image features). Global image features, such as gray level histograms, represent information from the whole image, they do not reveal information about local structures. Conversely, local image features represent the local image patches capturing information from the local content of the image. However, when several local image features are combined, their spatial relationships can be useful, revealing larger structures in the image. Local image features are a very large topic; this work concentrates on local image features suitable for object detection. In this field local image features are often represented by local (image) descriptors. In this work distinction between local image descriptors and local image features is defined so that local image descriptor is a numeric feature computed from an image patch and local image feature is a more refined presentation which can be used at the detection phase to localize desired image patches. Before going to the topic of local image description, the workings of the object detection systems are studied first.

2.1 Object detection with parts-and-structure model

State-of-the-art object detection and recognition systems work by dividing the object into smaller parts, and then defining the appearance model and spatial relationship for those parts – "parts and structure". An example is presented in Fig. 2.1. "Parts" are the small image parts characteristic to the object class, and "structure" defines the spatial structure between these parts.

This kind of method was first introduced by Fischler and Elschlager in 1973 [21], but was then largely ignored for two decades until Lades et al. [50] released their paper in 1993. The method has become popular for object detection and localization lately because of many benefits compared to detecting the whole object: description of local image parts can be simpler than description of the whole object, and the occlusion, part of the object

Figure 2.1: An example of object class detection with "parts and structure" model. Same parts – tires, motor and handlebars – of two motorcycles are marked by green circles and their spatial relationships with blue lines.

being hidden by another element in the image, can be naturally handled as well as deformations in the object. Common stages of object class detection systems utilizing "parts and structure" is presented in Fig. 2.2. Foreground images are the images containing objects to be learned and background images are images of basically everything else. First, interest points are found in the foreground training images, descriptions are created for these points and then a model for the object class is learned. Sometimes background images are utilized when the model is learned, while some methods work without explicit examples from the background class.

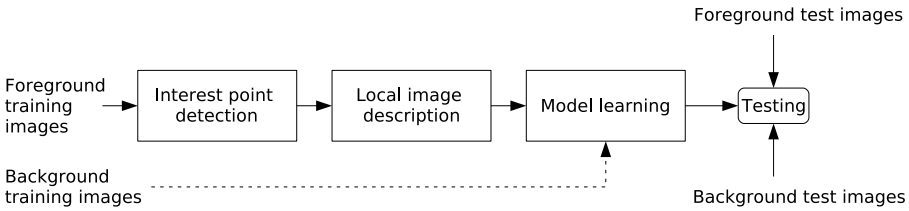


Figure 2.2: A conceptual diagram of learning stages for object class detection with the "parts and structure" model.

One important difference between methods following the approach in Fig. 2.2 is the level of supervision. Supervised systems require more manual help in the interest point detection phase. Manual help can range from segmentation of foreground objects to manually marking interest points. More supervision leads to interest points of assured quality, and the task of model learning becomes easier. Conversely, when the level of supervision is decreased the model learning becomes more complex as the interest points can be found outside of the objects to be learned, and therefore pruning of outlier interest points is required.

In the following some current object detection methods are partitioned by their level of supervision. The basic levels here are unsupervised, semi (weakly) supervised, supervised

and strongly supervised methods, but the division between groups is not clear as the different methods require different levels of supervision in respect to labeling, image alignment and segmentation of training images.

- Unsupervised methods: learning object classes from a set of unlabeled images containing several different object classes. This is yet to be reliably achieved, however, many methods are called unsupervised when they actually belong to the following class, semi-supervised.

- Semi-supervised methods: learning object classes from a set of labeled images. Many studies are concentrating on these kind of methods, some of them are briefly described here.

  Some methods use only a set of image features without a structure (constellation) model, for example, a method using various interest point detectors and local descriptors combined with AdaBoost by Opelt et al. [67], a method utilizing Bayesian learning of image features by Carbonetto et al. [10], and with shape based region detectors and descriptors by Jurie and Schmid [41].

  Many methods also use the structure model, for example, an object detection method using a vocabulary for parts of the object used along with information of their spatial relationships by Agarwal and Roth [1] and various methods from Perona's group, for example, classical parts and structure model learned with EM algorithm [89, 18]. There is a similar method using a star model instead of full constellation model [19] and "One-shot learning of object categories" by Fei Fei et al. [17] which is almost truly unsupervised in the sense that it tries to learn a new object class even from a single image, though knowledge from previously learned classes and background is used. A method by Mikolajzcyk et al. [60] detects multiple object classes (simultaneously), and the training is done from roughly aligned images using a hierarchically formed tree structure of local features (PCA-SIFT).

- Supervised methods: learning from labeled and segmented images. Some examples of these kind of methods are a method by Dorgo and Schmid [15] which selects parts for the object detection using Harris-Laplace and SIFT interest point detection, SIFT description and uses GMM and SVM classifiers without a spatial model, and a face detection method by Viola and Jones [88] which learns a face model from segmented training images using AdaBoost and integral images.

- Strongly supervised methods: labeled training images with manually selected "interest points" or areas. Some examples are detection of humans from a sub-window by detecting head, legs and arms separately by using Haar wavelets and SVM by Mohan et al. [64], and the work this thesis is related to, object classes learned from manually marked interest points using the Gaussian mixture models and multiresolution Gabor filters [43, 30].

Object detection methods can be further divided into another two classes. Some methods only detect whether the object is present in the image or not, not giving the object's precise location or even any kind of guess of its location. Some methods detect also the object's location and pose, however, in many publications the localization performance

is not explicitly measured. Here, the term "object detection" is used for methods which detect an object's presence in an image (is it there or not), and the term "localization" meaning that in addition to detecting the object's presence the method accurately localizes where in the image the object resides. Most of the object detection methods can give an estimate for the object's location, but with unsupervised or semi-supervised methods it is not generally possible to ensure that the features that are learned to distinguish objects really belong to the object itself instead of using some contextual information commonly found in the background, as noted in [67]. An example of this could be in the detection of cars, where an unsupervised method could learn that the presence of traffic signs implies also presence of cars because cars are often found in areas (roads or parking lots) where there are also traffic signs. However, even if the method could localize the object in addition to detecting its presence, it is commonplace to only measure whether the presence was correctly detected, not how precisely the object was localized.

To get an overview of the required level of supervision and suitability to localization, these properties of the object detection methods cited in this section are collated in Table 2.1. The methods are sorted roughly in order of increasing level of supervision. While some methods (e.g., [89]) claim to use "unlabeled" data they still use separate positive and negative training sets, and therefore are counted as using labeled data. Many methods could be used for localization, however, often the localization performance is not explicitly considered or measured, and these methods are marked with "not explicitly" for "localization". Methods which give a rough location estimate are marked with "bounding box".

Table 2.1: A table of object detection methods listing their level of supervision and capability of localization. A * in the segmented/normalized field means that the requirement is implicit in the training set, i.e., training images contain objects in roughly similar settings.

	Labeled	Segmented/aligned	Localization
[67]	Yes	No	No
[10]	Yes	No	No
[41]	Yes	Aligned	Bounding box
[1]	Yes	Aligned and normalized	Bounding box
[89]	Yes	Roughly aligned *	Not explicitly
[18]	Yes	Roughly aligned *	Not explicitly
[19]	Yes	Roughly aligned *	Not explicitly
[17]	Only 1 image	Roughly aligned *	Not explicitly
[60]	Yes	Roughly aligned	Bounding box
[15]	Yes	Segmented	Not explicitly
[88]	Yes	Segmented	Yes
[64]	Yes	Segmented	Bounding box
[43, 30]	Yes	Manually marked keypoints	Yes

Another distinction for methods is whether they are intended for object or object instance detection (object matching). In object detection the object class should be learned in general fashion and the method must not be too selective, otherwise it will be led astray

by intra-class variations (the differences between objects belonging to the same class), but still be able to capture inter-class variations (to distinguish objects from different classes from one another). In object instance detection the same object must be detected in different images. The method must learn details specific to the object so that it can distinguish that object from all others in the image.

Object instance detection methods are often used for matching (for example matching of stereo images) and therefore the method should be highly robust to viewpoint changes. One use is matching differing views of the same object or scene, and some examples of this are the original use of SIFT (Scale Invariant Feature Transform) features by Lowe [55] or maximally stable extremal regions (MSER) by Matas et al. [57]. In general, structure and motion problems do not necessarily require the use of local image descriptors; correct correspondences between interest points obey a geometric constraint, epipolar geometry, which can be solved, for example, by the RANSAC algorithm (e.g. [4]). Local image descriptors become useful when the difference between matched views is large.

The following sections review some of the most widely used interest point detection and local image description methods. In the end of the chapter also some complete object detection methods are shortly described.

2.2 Interest point detection

Interest points are known by many names, among them are distinguished regions [57], affine regions [63] and salient regions [55]. While they are called regions, most of the methods return a specific interest point and not an interest region. Whether the point is deemed interesting depends naturally on what is around it. To be useful the methods have to be invariant, or at least robust, to scale, rotation, noise and illumination changes and possibly for all affine changes; the same points should be found when for example image viewpoint changes or when there are changes in imaging conditions. For object detection they should also be in general robust to intra-class variations. For an example of different types of image changes see Fig. 2.3.

Many methods also determine scale and rotation of the interest point, and that information can be used when local image description is created for the interest point. In the following some of the most known interest point detectors are described shortly.

2.2.1 Harris corner detector

One of the first interest point detectors was a combined corner and edge detector by Harris and Stephens [31], where the main motivation was motion analysis from an image sequence created with a moving camera. The detector is based on local auto-correlation of the signal – the local auto-correlation measures changes when a patch is shifted slightly. A change of intensity for image $I(x, y)$ for a shift (u, v) is

$$E(u, v) = \Sigma_{x,y} w(x, y) \left[I(x + u, y + v) - I(x, y) \right]^2 \tag{2.1}$$

where $w(x, y)$ is a windowing function, usually Gaussian. For small shifts an approximation can be used,

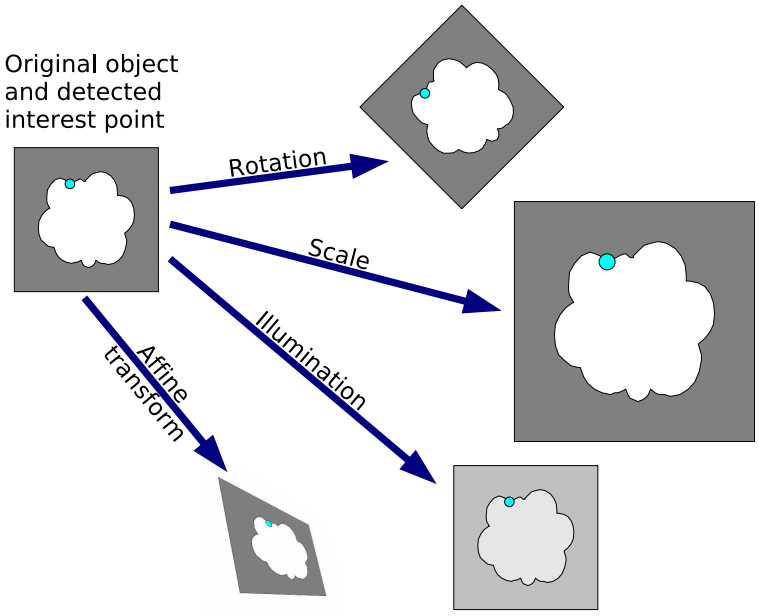


Figure 2.3: An example of types of changes the interest point detector should tolerate.

.

$$E(x,y) = [\Delta x \Delta y] M [\Delta x \Delta y]^T \ , \tag{2.2}$$

where M is a symmetric 2×2 matrix computed from image derivatives as (I_α is the image derivative calculated in direction α)

$$M = \Sigma_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \ . \tag{2.3}$$

Eigenvalues λ_1 and λ_2 of the matrix M are then solved. If both λ_1 and λ_2 are small, image is flat in that point, if both are large there is a corner, and if one is large and the other small there is an edge. Corner response can then be calculated without explicit eigenvalue decomposition,

$$R = \det M - k(\text{trace } M)^2 \ , \tag{2.4}$$

where k is an empirical constant, usually $0.04 \ldots 0.06$. Small $|R|$ means a flat point, $R > 0$ a corner point and $R < 0$ an edge point. The actual selected corner points are the local maxima of the R, so only one point per corner is actually selected. Local minima can also be searched which will isolate edge-points, but this is not as useful as detection of the corner points, because corner points are much more stable for small variations

in the image. The Harris corner detector is invariant to rotation, partially invariant to intensity change (if contrast becomes too low in a corner area, R becomes small and the point is classified as a flat area), but not invariant to scale.

The groundwork for automatic selection of scale of the interest point was investigated by Lindeberg [53], and based on that work the Harris detector has been extended to scale invariance by Mikolajczyk and Schmid [61] – the detector is called the Harris-Laplace detector. The scale invariance is achieved by computing a multi-scale representation for the Harris detector and then selecting points which have a local maximum of normalized image derivatives (the Laplacians).

The Harris-Laplace works by first detecting the Harris corner points in multiple scales. A threshold of $|R|$ is used to remove corner points which are not distinctive enough, as they are not stable for changes. For each point found an iterative algorithm is used to detect scale and the location of the interest point, as the exact location may change when different scales are searched through. The scale of the interest point is detected by finding the maximum for the Laplacian-of-Gaussians response. In a simplified version which is faster to compute the iterative steps are removed and the interest point is rejected if it is not a maximum of Laplacian-of-Gaussians.

In addition to scale invariance, Mikolajczyk and Schmid [61] extended the Harris detector to affine invariance and the detector is called the Harris-Affine. The main addition to Harris-Laplace is the detection of the shape of the interest point. The shape is determined by a rotated ellipse: the rotation is determined from local gradient orientation and the axes of the ellipse are determined from the ratio of eigenvalues of the second moment matrix.

2.2.2 SIFT detector

SIFT (Scale Invariant Feature Transform) by Lowe [55] includes both interest point detector and a local image descriptor. Only the detector is presented here, the descriptor will be discussed in the following section. SIFT works in four major stages:

1. Scale-space extrema detection. Potential interest points are searched in all scales and locations and potential interest points are identified with a difference-of-Gaussian function.
2. Keypoint localization. A model is used to determine location and scale of the interest point and interest points which are not deemed stable are pruned out.
3. Orientation assignment. Local image gradients are used to assign one or more orientations for each keypoint.
4. Keypoint descriptor. Descriptions for keypoints are created.

First, interest points are detected by applying a continuous function of scale: a scale space. The scale space function used here, $L(x, y, \sigma)$, is a product of the variable-scale Gaussian, $G(x, y, \sigma)$ and an input image, $I(x, y)$,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \ , \tag{2.5}$$

where $*$ is the convolution operation and $G(x, y, \sigma)$ is the Gaussian function,

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad . \tag{2.6}$$

Note that $L(x, y, \sigma)$ can be thought of as an image smoothed by a Gaussian kernel. Lowe has proposed to use the extrema of the difference-of-Gaussian function as interest points which can be detected efficiently. Difference-of-Gaussians is defined as

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.7}$$

a difference of two Gaussians on nearby scales separated by constant factor k. This can be efficiently computed from two smoothed images,

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad . \tag{2.8}$$

An example of computation of difference-of-Gaussians can be seen in Fig. 2.4. The image is smoothed with Gaussians which are separated by a constant factor k in scale space – these images, $L(x, y, \sigma)$, form the right image stack. Adjacent images are then subtracted from each other to produce difference-of-Gaussians images, $D(x, y, \sigma)$. Each octave, a doubling of σ, can be handled separately and the computation time can be saved by downscaling the image for every octave. The interest points are located by finding local extrema from the stack of difference-of-Gaussians images, i.e., a point is an interest point if it is the smallest or largest of the $3 \times 3 \times 3$ pixels surrounding it at the same scale level and the levels above and below.

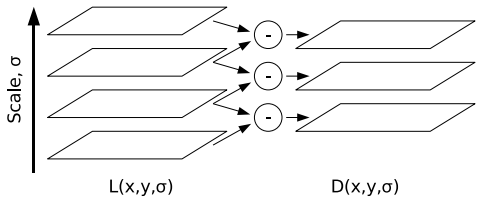


Figure 2.4: Initial image is convolved with Gaussians with different scales, $G(x, y, \sigma)$, producing smoothed images, $L(x, y, \sigma)$. Difference-of-Gaussians, $D(x, y, \sigma)$, can then be computed as difference of two adjacent images.

Next, the exact location of the interest point is measured by fitting a 3D quadratic function to local image points. This calculation also reveals interest points which are in areas with too low contrast; these are removed. The difference-of-Gaussians has a strong response near edges, but edge points are not stable as location along the edge is unstable to small amounts of noise. Therefore, similarly to the Harris detector, the principal curvature is computed for each point by calculating eigenvalues of the Hessian matrix for each interest point. The interest point is accepted only if the ratio between eigenvalues is small enough, and the actual calculation of eigenvalues can be avoided,

again, similarly to the Harris detector. Orientation of the keypoint is determined by computing an orientation histogram for each interest point and its neighborhood. The highest peak of the histogram is defined as the interest point's orientation, but also other peaks higher than 80% of the highest peak are accepted as separate interest points, i.e., one interest point can be split into several different interest points if there are many dominant directions in the orientation histogram.

2.2.3 Entropy based detector

Kadir et al. have developed an interest point detector which is based on an information theoretical approach, the entropy of local image regions [42]. Unlike many other interest point detectors, their detector is explicitly designed while having intra-class variations in mind. The detector works in three steps:

1. Calculate entropy of the local image areas (the entropy of a gray-level or color histogram) in several scales (circles with varying sizes). A flat image area has a histogram with one strong peak and the entropy is low, and an image area with more variations will have a histogram with several peaks or even a flat histogram which has the highest entropy.
2. Select scales which have peaks of entropy.
3. Use an inter-scale unpredictability measure to weight entropy values. Image areas where a specific scale has strong peak get weighted higher than areas where a peak is weak compared to nearby scales. For example, in a very noisy image area, entropy is high at all scales, but there is not one specific scale which has a strong peak.

For instance a bright circle on a black background will have its entropy maximum when the detector's area contains some black area around the circle so that there are approximately an equal number of white and black pixels inside it. Entropy is small if the area is completely inside the circle, and will become smaller when the area size is increased and black pixels start to dominate.

This entropy based saliency measure for interest points is naturally invariant to rotation, translation and small affine transforms: the histogram does not change during these changes. However, it is only invariant to shifts in image intensity, not to contrast changes. Invariance to all affine changes is possible when a circular image scanning window is changed to a ellipse. This increases complexity considerably, because in addition to scale (i.e., the radius of the circle) there is now also rotation and ratio between major and minor axes of the ellipse to search. For that reason a local search strategy is used: first a circular window is used to search for seed points (only position and scale), and then the rotation and shape of the ellipse is iteratively changed to maximize saliency.

2.2.4 Maximally stable extremal regions

MSER (Maximally Stable Extremal Regions) has been introduced by Matas et al. [57]. MSER is based on thresholding and an extremal region is a connected area in a thresholded image. All extremal regions are found by thresholding the image with all possible

thresholds, $[0 - 255]$ for normal gray-scale images, and finding then all connected areas. Maximally stable extremal regions are extremal regions which do not change, or change as little as possible, when the threshold is changed. In practice this means that MSERs are regions with relatively flat intensity surrounded by sharp intensity change.

Regions found by MSER are invariant to all adjacency preserving transformations, which includes scale, rotation and affine transforms as long as the stable region is found in a planar object, invariant to shifts in image intensity but not invariant to large contrast changes.

2.2.5 Performance evaluation

Performance of various interest point detectors (called affine region detectors) was tested in [63]. The performance was tested by checking how many of the same interest points were found in image sets where the viewpoint, scale, rotation or illumination varied, or images were blurred or JPEG-compressed. Accurate homography between images (how the points in one image map to points in the other image) was measured beforehand, and the accuracy of interest point detectors was measured by how many of the found points were matched within certain limit in both images. MSER (Maximally Stable External Regions) [57] and Hessian-Affine [61] were found to perform best overall. The results are not directly applicable to object detection as the tests used images of exact same scenes under various changes: there was no intra-class variations characteristic to object detection problems.

2.3 Local image description

In the following some methods used as local image descriptors are explained shortly. When local image descriptors are used with interest point detectors which detect scale and orientation, and potentially also affine shape of the interest point, the image patch can be normalized before creating the local description. Therefore, in such case the local descriptor itself does not have to be scale or rotation invariant. However, invariance, at least to some degree, to imaging condition changes (lighting changes or noise), and to other small perturbations is important. Invariance to small perturbations is even more important when the descriptor is used in object detection where the descriptor should not be too selective to small variations, otherwise it cannot represent reliably an object class.

In addition to the descriptor not having to be scale or rotation invariant, the use of interest point detection as a first step has also the added benefit that the descriptor can be computationally complex as there is only a limited number of descriptors to compute. If an interest point detector is not used, or rather the local image feature combines both interest point detector and local descriptor, the local descriptor has to be used in an exhaustive search and the computational complexity must be low.

Multiresolution Gabor filters are the main local image descriptor used in this thesis, and they are therefore presented in their own section, Section 3.1, in more detail.

2.3.1 Local description by pixel values

The most straightforward idea for local image description is taking a part of the image around the interest point and using the gray-level pixel-values directly as an image descriptor (see Fig. 2.5). If the interest point detector detects scale and rotation of the interest point, the local image area can be scaled and rotated to account for these changes. There are two major problems with this kind of descriptor: high dimensionality of the descriptor (for example, 20×20 area will have a descriptor of length 400) and poor invariance to small perturbations of the image. Both of these problems can be alleviated by reducing the dimensionality, for example by using PCA (principal component analysis). This kind of local descriptor has been used for example by Fergus et al. [18]. A patch of the image based on the scale of the interest point was taken and scaled to size 11×11. The image patch was used as a vector of the gray-level values of length 121 and projected onto 10-15 principal components. Principal components were calculated beforehand based on a large number of interest points.

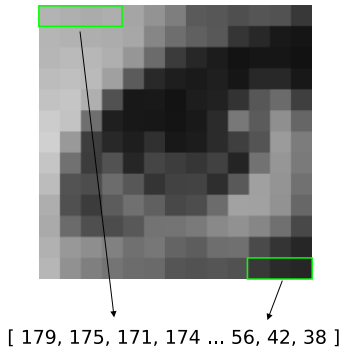


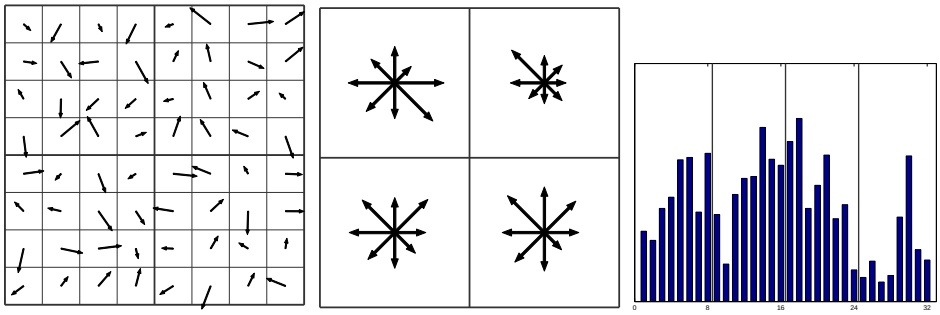
Figure 2.5: Image description by direct pixel values.

2.3.2 SIFT descriptor

SIFT (Scale Invariant Feature Transform) by Lowe [55] includes a local image descriptor based on local image gradients. The descriptor is created for the scale level found by the interest point detector and the rotation of the interest point is also taken into account so that the descriptor is scale and rotation invariant.

Fig. 2.6 shows an example of descriptor creation. The descriptor is created by first calculating image gradients (their magnitudes and orientations) around the location of the interest point. In the example Fig. 2.6(a), gradients for 8×8 points have been calculated. Gradient magnitudes are weighted by a Gaussian so that they become gradually smaller when the distance to the center point increases. Weighting is performed to avoid large changes in the descriptor when the window moves slightly. Then, for each 4×4 subregion, weighted gradients are divided using interpolation to 8 primary directions and then summed (Fig. 2.6(b)), i.e., the gradients pointing roughly to the 8 primary directions are summed together. The actual descriptor is the vector of directional gradient sums from

all subwindows (Fig. 2.6(c)). In the example the descriptor is of length 32 (8 primary directions for 4 sub-windows), but usually an area of 16×16 points is used with 4×4 subregions, therefore creating a descriptor with a length of 128 (8 primary directions with 16 sub-windows). The descriptor is finally normalized to unit length.



(a) Image gradients (b) Weighted sum of gradients (c) Complete descriptor

Figure 2.6: An example of SIFT local descriptor creation.

High dimensionality of the descriptor is a problem for classifiers. Therefore a variant of SIFT has been proposed: PCA-SIFT [46]. In PCA-SIFT the directional sums of gradients for subregions are not calculated, instead the whole patch of local image gradients is used and the dimensionality is reduced by using PCA. The eigenspace is precalculated using a large number of image patches. It was found out that using only the first 20 principal components gave good results; this means that the feature vector is only of length 20, considerably shorter than the descriptor of original SIFT. Another form of combination of SIFT descriptor and PCA has also been used for example by Mikolajczyk et al. [60], where a normal 128 dimensional SIFT descriptor is reduced to 40 dimensions.

2.3.3 Local binary patterns

The original LBP feature is calculated by comparing the value of a center pixel to other pixels in a 3×3 area, and the resulting binary number is the result of the LBP operator (see the example in Fig. 2.7). A 256-bin histogram of LBP-values is formed when the feature is computed over a larger area. The histogram can be used efficiently as a texture descriptor.

The LBP operator has been extended in two ways [66]. LBP operator can operate on different neighborhoods. $\mathrm{LBP}_{P,R}$ refers to a LBP operator which considers P neighbors at the distance of R, for example, $\mathrm{LBP}_{16,2}$ considers 16 neighbors at the distance of 2. $\mathrm{LBP}_{P,R}$ produces 2^P output values which also means that the histogram will be of the length 2^P. The histogram becomes impractically large if P is increased, however, it has been noticed that so called uniform patterns contain more information than the others and the histogram length can be reduced by bundling all non-uniform patterns into a single bin. Uniform patterns include only a limited number of bitwise transitions – from 0 to 1 or the opposite. For example, 00000000 has zero transitions and 00111100 has two transitions. An uniform LBP-operator which bundles the patterns with more than two transitions to a single bin is marked as $\mathrm{LBP}^{u2}_{P,R}$.

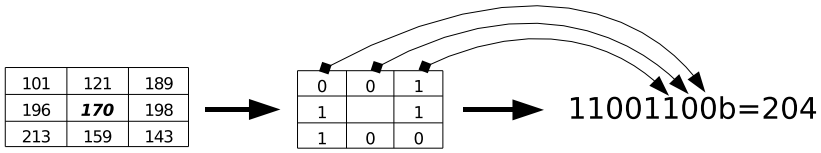


Figure 2.7: An example of LBP calculation. Pixels surrounding the center pixel are thresholded based on the value of the center pixel and a binary feature is formed.

LBP is mainly a texture descriptor and one LBP histogram does not include any information on how the texture changes spatially and therefore normal LBP features are not very useful directly as local image descriptors. For that purpose, as proposed by Hadid et al. in [26], an image patch is divided into smaller patches for which separate LBP histograms are computed. The histograms from adjacent image patches are combined to form an image feature which can describe complex local image areas. Image patches are represented with a combination of $\mathrm{LBP}_{4,1}$ and $\mathrm{LBP}^{u2}_{8,1}$ histograms [26]. A 19×19 image patch is divided into 9 overlapping 9×9 patches. An example can be seen in Fig. 2.8. A $\mathrm{LBP}^{u2}_{8,1}$ histogram is computed for the whole 19×19 image patch and $\mathrm{LBP}_{4,1}$ histograms for smaller 9×9 images. The total length of the combined histograms is $203 - 59$ for the $\mathrm{LBP}^{u2}_{8,1}$ histogram and 16 for each of the 9 $\mathrm{LBP}_{4,1}$ – and it is used directly as the local image feature.

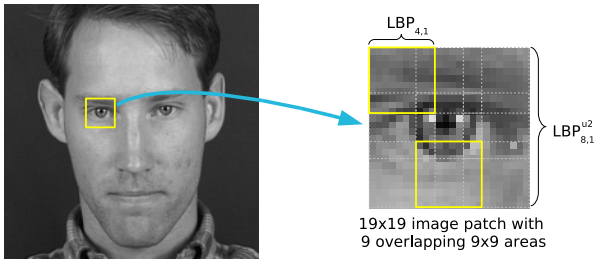


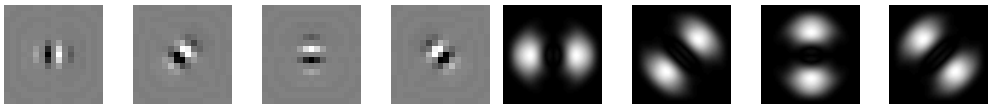
Figure 2.8: Local image patch representation with LBP histograms.

LBP features can be computed efficiently. However, when used as image features in this fashion the resulting feature vectors are long and the efficiency of the classifier may create a bottleneck for total efficiency.

2.3.4 Steerable pyramid

The steerable pyramid is a linear decomposition of image into scale and orientation subbands, and is jointly shiftable in the both orientation and scale [81]. The basis functions for the decomposition transform can be formed by translations, dilations and rotations of a single filter. The transform is constructed as a recursive pyramid. The basis functions

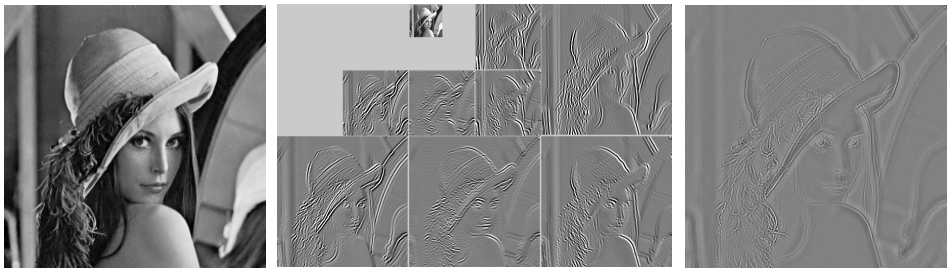
are directional derivative operators, and the number of orientations is defined by the order of the derivative; Nth order derivative has $N+1$ orientations. Examples of oriented bandpass filter kernels are shown in Fig. 2.9.



(a) (b)

Figure 2.9: 3rd order (4 orientations) steerable filters: (a) Spatial domain, (b) Frequency domain.

The pyramid is formed by convolving the input signal with a set of oriented bandpass kernels and a low-pass kernel. To avoid aliasing the bandpass portion is not subsambled, but the low-pass portion is subsambled by a factor of two. The low-pass filtered portion is then used for computing the next level in the pyramid. In addition to the bandpass filtered levels, also the high frequency residual highpass sub-band and the low-pass sub-bands can be stored to be able to reconstruct the original signal. An example of an image decomposed into a 2-level pyramid with 3rd order steerable filters is presented in Fig. 2.10.



(a) (b) (c)

Figure 2.10: Example of image decomposition using steerable filters: (a) Original image; (b) Pyramid level decomposition with the 3rd order steerable filter (possess 4 orientations), 2 pyramid levels and the lowpass sub-band; (c) Highpass residual sub-band.

Steerable pyramid features have been used in object detection and recognition by Ballard and Wixson in [3]. The description of an object is created by using filters with several different orders (number of orientations) and scales. Here, a slightly different approach is taken to be compatible with the approach used with multiresolution Gabor features, and only filters with a specified number of orientations is used in several scales. Only the bandpass filtered levels of the pyramid are used and the highpass and lowpass portions of the image are discarded. The responses can be arranged to a similar matrix form as is used with the multiresolution Gabor features (Section 3.1, (3.8)). Computational

performance of filter response computations is comparable to Gabor filters, however, when filtering in the spatial domain, smaller filters can be used with steerable pyramid filters than with Gabor filters and there are only real values making steerable pyramid responses faster to compute.

2.3.5 Performance evaluation

The performance of local descriptors was tested in [62]. The test builds on the previous performance testing of interest point detectors [63]. First, interest points (or regions) are searched in two images where the viewpoint, scale, rotation or illumination are varied, or images were blurred or JPEG-compressed. Then, the descriptors are evaluated based on how well they can find the correct (same) points among the interest points found in two images. SIFT [55] and its modification made by the authors of the article, gradient location and orientation histogram (GLOH), performed best, but again the performance characteristics in this test cannot be directly applied to object detection.

2.4 Object detection methods

In the following few object detection methods are shortly described. Some of the methods combine interest point detectors and local descriptors to perform detection of complete objects, some use different approaches.

2.4.1 Feature-based affine-invariant detection and localization of faces

The methods presented in the following chapters of this thesis are connected to a face (object) localization method developed by Hamouz et. al, [27, 28, 29, 30]. The discussion here is based mostly on [30]. The paper distinguishes between face detection and localization so that face detection methods estimate the position and pose of the face roughly, for example by a bounding box, and face localization as precise localization of facial features.

The method uses a separate local image feature detection phase to detect and localize facial parts (10 facial parts: eyes, eye-corners, nostrils and sides of mouth) and then another phase to combine them to complete the face localization using a constellation model. Local image features used are multiresolution Gabor features as presented in [48], and the Gaussian mixture model is used for learning the different facial feature classes. During the detection the 200 best candidates for each facial feature is returned.

After facial feature candidates have been detected, a constellation model is used to select which of the found candidates forms a face. The constellation model works by selecting three candidates of different types of facial features (for example two eye-centers and a nostril) and calculating with which kind of affine transform the candidate points fit to the face model formed from the training set. If the required transformation has not been seen in the training set, the three points are probably false positives and do not belong to a face.

After a likely face, or generally an object, hypotheses have been found with the constellation model, an appearance model is used to verify if a real face was found, or if the

hypothesis was badly localized or resides in the background. This step works in the image level and does not use local features: an image patch is extracted and it is classified with a SVM (support vector machine) classifier which gives a score for the patch being a face. Classifier is trained from training data where patches are manually marked and a bootstrapping technique is used to generate negative examples. Two different SVM classifiers and two differently sized image patches are used. First coarse 20×20 patches are used to prune out clearly incorrect face hypotheses and then the most likely hypotheses of those are further verified using 45×60 patches. Multiple steps are used because with the coarse resolution small misalignments cannot be noticed. The localization results were found to be very good.

The differences to the method presented in this thesis are:

- An improved classification method, Gaussian mixture model with confidence information.
- The performance of Gabor filtering has been improved greatly, and is now up to 50x faster.
- Cross-validation for selecting Gabor filter parameters has led to distinctly better results [43].

2.4.2 Distinctive image features from scale-invariant keypoints

SIFT interest point detector was introduced in Section 2.2.2 and SIFT descriptor in Section 2.3.2. Together, they can be used for object recognition [55].

Naturally, the detector is used to find the interest points and their scales and orientations, then the descriptor creates a description for all of these points, called keypoints. For training, the presented method uses a single image for the object which should be recognized, the image should include no other objects and have a clutter-free background. From the training image the keypoints are searched and their descriptions stored in a database, including the spatial relations of the descriptors. The procedure can be repeated for other images with other objects.

During detection, again, interest points are detected and descriptors created. For each of the keypoints a closest match, smallest euclidean distance, in the database is searched. Many of the found interest points arise from the cluttered background or unknown objects, so there may not always be a correct match. Therefore, a threshold is applied. A global threshold does not perform well, but a threshold based on the difference between the closest match and the second closest match found in the database is used instead. The idea is that if the match is correct, the second closest match will be much more distant, and if the match is incorrect the second closest match will likely have a similar distance.

Even after discarding many of the false matches based on thresholds a large part of the remaining interest points will still be incorrect. Hough transform is used to determine if several of the features vote for the same object pose. Each keypoint has parameters for 2D location, scale and orientation, and the database contains the same information for keypoints found in the training image. A Hough transform can be created predicting

location, orientation and scale from the matched keypoints. In addition, each cluster of three or more features found by Hough transform is subjected to a geometric verification procedure to discard keypoints which do not agree with the model accurately enough, or add keypoints which agree but were not found with the Hough transform. Finally, a probabilistic model is used to accept or reject the object hypothesis based on an actual number of matched features.

The method can recognize the trained objects in highly varying poses and when they are heavily occluded. Several objects can be detected at the same time. However, the method is intended for detecting the same object (single object instance): intra-class variations are not considered at all.

2.4.3 Object class recognition by unsupervised invariant learning

The article by Fergus, Perona and Zisserman [18] presents an object class recognition method where object classes are learned and recognized from unsegmented images of the object in cluttered scenes. The method is not completely unsupervised as the training set images are all assumed to contain an instance of the object class, i.e., when training a detector for motorcycles, all training set images must contain a motorcycle.

The method applies the parts and structure model. The object model consists of parts where for each part appearance, relative scale and mutual position with other parts is known. Some parts may also be occluded. The model is generative and probabilistic: parts are modeled with probability density functions, more precisely Gaussians. During learning, interest points and their scales are first searched. From the appearance, scale and mutual position a model is learned so that it gives maximum-likelihood description. Recognition is performed by detecting interest points and their scales in the query image and evaluating found regions in the Bayesian manner applying model parameters found during training.

First N interest points are found with locations $\boldsymbol{X}$, scales $\boldsymbol{S}$, and appearances $\boldsymbol{A}$. The decision is based on likelihood for object presence modeled as

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{S}, \boldsymbol{A}; \theta) &= \sum_{h \epsilon H} p(\boldsymbol{X}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{h}; \theta) \\ &= \sum_{h \epsilon H} p(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{S}, \boldsymbol{h}; \theta)\ p(\boldsymbol{X}; \boldsymbol{S}, \boldsymbol{h}; \theta)\ p(\boldsymbol{X}; \boldsymbol{S}, \boldsymbol{h}; \theta)\ p(\boldsymbol{S}, \boldsymbol{h}; \theta)\ p(\boldsymbol{h}; \theta) \end{aligned}$$

where $\boldsymbol{h}$ is a hypothesis vector of length P, which enumerates which of the detected N interest points belong to the object. Some of them maybe zero, which means that that particular object part is not present. All valid allocations of features to the parts are presented by $\boldsymbol{H}$, which is of $O(N^P)$. From this complexity it can be seen that the number of detected interest points, N, must be relatively low, usually up to 30, and the number of object parts, P, even lower, typically $3 - 7$. The four $p(\cdot)$ clauses represent probabilities for appearance, shape, relative scale and other, the last one handling effects of occlusion. The first three are modeled with Gaussians, and the last one with a Poisson distribution.

Interest point detection is performed with an entropy based detector presented in Section 2.2.3. The local description is performed based on the pixel values, as described in Section 2.3.1: a patch around the interest point is taken, where the size is based on the scale given by the interest point detector. The patch is then scaled to size 11×11, and the resulting vector of length 121 is reduced to $10 - 15$ dimensions using PCA, and this vector is the descriptor for appearance, $\boldsymbol{A}$. From the positions and scales of the interest points also $\boldsymbol{X}$ and $\boldsymbol{S}$ are known. When these are known for the images in the training set, parameters, θ, of the model are learned using expectation-maximization algorithm.

The method performed very well on six diverse datasets, including object classes such as human faces, motorbikes, airplanes and spotted cats.

2.4.4 Rapid object detection using a boosted cascade of simple features

Viola and Jones presented an efficient object detection method in [88]. The method uses simple features based on integral images which are extremely efficient to compute. These simple features are combined by an AdaBoost classifier to create an efficient object detector. The classifier is used in a cascade, if first simple classifiers already determine that there is no object in the image patch, using more complex classifiers is omitted, which further improves the efficiency. The method is supervised, it is trained using segmented images of the training class and background images. During detection the detector goes through the image in a windowed fashion: the image is divided to patches and the detector is used in each of them separately.

The method uses simple rectangular features, examples are shown in 2.11. The value of a feature is computed by taking the sum of pixel values in the white parts of the filter, and subtracting it from the sum of pixel values in the gray parts. The size of the rectangular features are varied. With the base size of 24×24 used for the detector, the complete set of rectangular features is over one hundred thousand. Therefore, efficient computation of feature values is important and they can be computed very efficiently using an intermediate representation of the image, integral image.

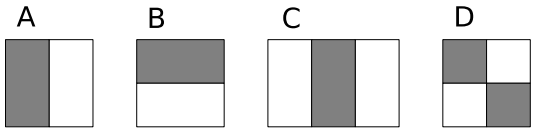


Figure 2.11: Rectangular features work so that the sum of pixel values in white parts of the rectangle are subtracted from the sum of pixel values in gray parts. Features consist of 2 (A and B), 3 (D) or 4 (D) rectangles.

The integral image contains the sum of pixel values above and to the left of the current pixel in the original image. It is computed as

$$I_i(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \ , \tag{2.9}$$

where $I_i(x, y)$ is the integral image and $I(x, y)$ is the original image. The integral image can be computed in one pass over the image:

$$\begin{aligned} s(x, y) &= s(x, y-1) + I(x, y) \\ I_i(x, y) &= I_i(x-1, y) + s(x, y) \ , \end{aligned} \tag{2.10}$$

where $s(x, y)$ holds the cumulative row sum and negative indexes equal to zero. Fig. 2.12 demonstrates how value inside any rectangle can be calculated using only few operations. Values of the actual features (examples in Fig. 2.11) can be computed in a similar fashion.

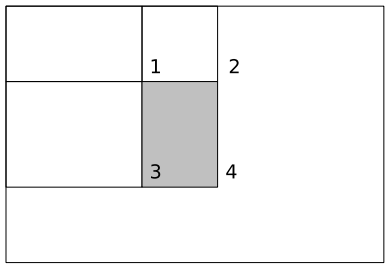


Figure 2.12: Using integral image to calculate sum values inside grayed area can be performed as $4 - 2 - 3 + 1$, i.e., take the value at point 4, deduct the values at points 2 and 3 and then add value at point 1 because previous step deducted its sum twice.

Computation of a single feature is fast, computation of all of them still takes a long time and a small subset of the features should be selected. AdaBoost [22] is a classifier which can select a small number of relevant features and combine them into a powerful classifier: the complete classifier combines several weak classifiers, each weak classifier here operates on one rectangular feature. The AdaBoost classifier selects at each training step one feature which best separates the positive and negative training samples. The weights of the training samples are adjusted so that the next weak classifier will concentrate on the samples which have been wrongly classified with previous weak classifiers. The training error becomes smaller with each added weak classifier.

The detection is performed in sub-windows and a vast majority of them are negative, i.e., the object to be detected is not there. Therefore, rejecting negative samples should be done as fast as possible, which is achieved by using classifiers in a a cascade. The first classifier is very simple, trying to achieve a very small false negative rate, i.e., no positive samples should be rejected. However, the first classifier should reject many of the true negatives. Following classifiers are more complex targeting consecutively smaller false positive rates, i.e., trying to reject more and more of the true negative samples. To be classified as positive, all classifiers must give a positive result and the classification procedure ends if any of the classifiers give a negative result. During training each classifier will get as its input only the samples which passed through the previous

classifier, also the thresholds for false negatives and false positives are modified for later classifiers.

The method was tested in face detection. The results were very good and, at the time, the detector was considerably faster than any of the earlier approaches.

2.5 Summary

This chapter presented an overall model of parts-and-structure type object detection methods. The methods pertaining to this model usually combine an interest point detector which first finds significant points in images, which should stay stable when imaging conditions or even object instances change, and a local image descriptor, which is used to describe the local appearance of the image. Some of the most well-known interest point detection, local image description and complete object detection methods were then presented.

Chapter III

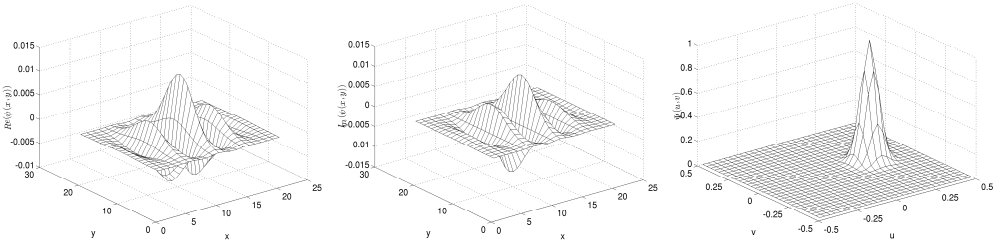
Multiresolution Gabor features

Gabor filters, originally introduced by Dennis Gabor in 1946 for 1D signals [23], have a well-known connection to receptive field receptor profiles of mammalian visual systems [13]. They also are a realization of the general image processing operator proposed by Granlund [25]. Multiresolution structure of Gabor filters is similar to wavelets, but it lacks the important orthogonality property [52]: Gabor filters do not form a basis. They form instead a frame, which is a generalization of the basis without orthogonality and unique dual transform properties.

Gabor filters have been a popular feature extraction method in last few decades, and during the 2000s the activity has actually increased according to IEEE Xplore^{TM} database. The most important reason for the increase is probably the wide success in some application areas, such as biometric authentication. Methods based on Gabor features have been very successful in iris recognition [14], large scale face recognition contests (e.g. 2 best methods in [59]), and provided state-of-the-art accuracies in fingerprint matching [40] and face localization [30]. It can be assumed that Gabor features will have an important role also in the future. However, in the relevant literature a major disadvantage of Gabor features, the computational heaviness, is often overlooked. This Chapter explains construction of Gabor filters and efficient computation of multiresolution Gabor features.

3.1 Constructing Gabor features

Gabor filters are linear filters whose responses are defined by a sinusoidal wave multiplied by a Gaussian function. An example of a 2D Gabor filter is presented in Fig. 3.1. Usually in image processing Gabor filter responses are used in a multiresolution structure: the features are based on responses of Gabor filters on multiple scales and orientations forming a multiresolution Gabor frame structure. While the Gabor filter responses are complex-valued, commonly only response magnitudes are used, but it will be shown later that using the complex values (or magnitude and phase presentation) improves results in many applications.



(a) (b) (c)

Figure 3.1: Gabor filter in 2D with parameters $f_0 = 0.2$, $\theta = 0$, $\gamma = \eta = 1$; (a) spatial domain, real component; (b) spatial domain, imaginary component; (c) frequency domain.

3.1.1 1D Gabor filter

The 1D Gabor filter is presented first since many of the efficiency improvements can be obtained easily in 1D and then generalized to 2D Gabor filters.

A normalized Gabor filter in the time domain can be defined as [44]

$$\psi(t; f_0) = \frac{|f_0|}{\gamma\sqrt{\pi}} e^{-\left(\frac{|f_0|}{\gamma}\right)^2 t^2} e^{j2\pi f_o t} \tag{3.1}$$

where f_0 denotes the filter tuning frequency and γ the filter bandwidth. The filter function can be divided into two parts: a complex sinusoidal wave on the tuning frequency and a Gaussian envelope defining the effective time duration. The effective time duration is inversely proportional to the effective bandwidth via the uncertainty relation.

The corresponding equation in the Fourier domain is [44]

$$\Psi(u; f_0) = e^{-\left(\frac{\gamma\pi}{f_0}\right)^2 (u-f_0)^2} \tag{3.2}$$

where u denotes the frequency.

3.1.2 2D Gabor filter

Similarly to the 1D case, the 2D filter can be divided into an elliptical Gaussian and a complex plane wave. The filter in the 2D spatial domain is [44]

$$\begin{aligned} \psi(x, y; f_0, \theta) &= \frac{f_0^2}{\pi\gamma\eta} e^{-\left(\frac{f_0^2}{\gamma^2}x'^2 + \frac{f_0^2}{\eta^2}y'^2\right)} e^{j2\pi f_0 x'} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{3.3}$$

where the new variable θ denotes the rotation angle of both the Gaussian and plane wave. This is not the most general form of the 2D Gabor filter, but a form whose properties are the most useful for image processing, namely that the filters in different frequencies with the same bandwidth parameters are scaled versions of each other. The bandwidth is controlled by two parameters, γ and η, corresponding to the two perpendicular axes of the Gaussian.

The same filter in the frequency domain is [44]

$$\begin{aligned} \Psi(u, v; f_0, \theta) &= e^{-\pi^2\left(\frac{u'-f_0}{\alpha^2}+\frac{v'}{\beta^2}\right)} \\ u' &= u\cos\theta + v\sin\theta \\ v' &= -u\sin\theta + v\cos\theta. \end{aligned} \tag{3.4}$$

The filter in (3.3) is centered to the origin and its response for an image function $\xi(x, y)$ can be calculated at any location (x, y) with the convolution [44]

$$\begin{aligned} & r_\xi(x, y; f, \theta) = \psi(x, y; f, \theta) * \xi(x, y) \\ & = \iint_{-\infty}^{\infty} \psi(x - x_\tau, y - y_\tau; f, \theta)\xi(x_\tau, y_\tau)dx_\tau dy_\tau \ . \end{aligned} \tag{3.5}$$

3.1.3 Multiresolution structure

A filter bank consisting of several filters needs to be used because relationships between responses provide the basis for distinguishing objects. The selection of discrete rotation angles θ_l has been demonstrated for example in [69], where it was shown that the orientations must be spaced uniformly.

$$\theta_l = \frac{l2\pi}{n} \ \ l = \{0, \ldots, n-1\} \ , \tag{3.6}$$

where θ_l is the lth orientation and n is the total number of orientations to be used. The computation can be reduced by half since responses on angles $[\pi, 2\pi[$ are complex conjugates of responses on $[0, \pi[$ in the case of a real valued input. The frequencies must be selected exponentially [44, 50],

$$f_l = k^{-l} f_{max} \ \ l = \{0, \ldots, m-1\}. \tag{3.7}$$

Common values for k include $k = 2$ for octave spacing and $k = \sqrt{2}$ for half-octave spacing.

Using the features to cover frequencies of interest $f_0, \ldots, f_{m-1}$ and the orientations for desired angular discrimination, one can construct a set of features at an image location (x_0, y_0). The filter responses are arranged into matrix form as

$$\mathbf{G} = \begin{pmatrix} r(x_0,y_0;f_0,\theta_0) & r(x_0,y_0;f_0,\theta_1) & \cdots & r(x_0,y_0;f_0,\theta_{n-1}) \\ r(x_0,y_0;f_1,\theta_0) & r(x_0,y_0;f_1,\theta_1) & \cdots & r(x_0,y_0;f_1,\theta_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0,y_0;f_{m-1},\theta_0) & r(x_0,y_0;f_{m-1},\theta_1) & \cdots & r(x_0,y_0;f_{m-1},\theta_{n-1}) \end{pmatrix} \tag{3.8}$$

where rows correspond to responses on the same frequency and columns correspond to responses on the same orientation. The first row is the highest frequency and the first column is typically the angle 0°.

3.1.4 Feature manipulation for invariant search

Linear row-wise and column-wise shifts of the response matrix correspond to scaling and rotation in the input space, and therefore, invariant search can be performed by simple shift operations: by searching several spatial locations (spatial shift) and by shifting response matrices. With normalization of the response matrix, illumination invariance can also be achieved [44, 48].

Rotating an input signal $\xi(x, y)$ anti-clockwise by $\frac{\pi}{n}$ equals to the following shift in the feature matrix

$$\mathbf{G} = \begin{pmatrix} r(x_0,y_0;f_0,\theta_{n-1})^* & r(x_0,y_0;f_0,\theta_0) & \Rightarrow & r(x_0,y_0;f_0,\theta_{n-2}) \\ r(x_0,y_0;f_1,\theta_{n-1})^* & r(x_0,y_0;f_1,\theta_0) & \Rightarrow & r(x_0,y_0;f_1,\theta_{n-2}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0,y_0;f_{m-1},\theta_{n-1})^* & r(x_0,y_0;f_{m-1},\theta_0) & \Rightarrow & r(x_0,y_0;f_{m-1},\theta_{n-2}) \end{pmatrix} \tag{3.9}$$

where * denotes complex conjugate.

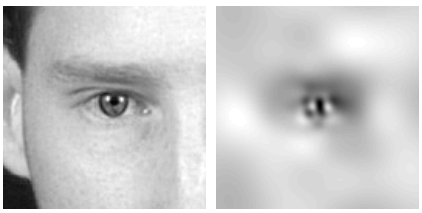
Downscaling the same signal by a factor $\frac{1}{k}$ equals to the following shift in the feature matrix

$$\mathbf{G} = \begin{pmatrix} r(x_0,y_0;f_1,\theta_0) & r(x_0,y_0;f_1,\theta_1) & \cdots & r(x_0,y_0;f_1,\theta_{n-1}) \\ r(x_0,y_0;f_2,\theta_0) & r(x_0,y_0;f_2,\theta_1) & \cdots & r(x_0,y_0;f_2,\theta_{n-1}) \\ & & \ddots & \\ \Uparrow & \Uparrow & & \Uparrow \\ r(x_0,y_0;f_m,\theta_0) & r(x_0,y_0;f_m,\theta_1) & \cdots & r(x_0,y_0;f_m,\theta_{n-1}) \end{pmatrix} \tag{3.10}$$

For this to work new low frequencies f_m must be computed and stored in advance while the highest frequency responses on f_0 vanish in the shift.

3.1.5 Image reconstruction

The original image patch can be reconstructed from multiresolution Gabor features via their bi-orthogonal transform functions [71]. An example of local reconstruction is demonstrated in Fig. 3.2. The reconstruction of a complete object can be performed by combining features from several spatially distant points.



(a) (b)

Figure 3.2: A multiresolution Gabor feature reconstruction example: (a) Original image of an eye; (b) Reconstructed image using responses from a single point (at the center of an eye).

3.1.6 Filter spacing - selection of multiresolution feature parameters

The selection of filter bank values, filter frequencies, bandwidths and number of orientations, is an application dependent problem. It is not, however, necessary to define all parameters separately due to their interdependencies [35, 36]. More detailed information and the analytical solutions are presented in Appendix I.

Filter frequencies

The multi-resolution frequencies f_i are drawn from $f_0 = f_{max}$, $f_1 = f_{max}/k$, $f_2 = f_{max}/k^2$, $f_m = f_{max}/k^{m-1}$ and these equations define the relationships between the other parameters in (3.1), (3.2), (3.3) and (3.4). Table 3.1 can be used to select the multi-resolution feature parameters. Number of filters is denoted by m, scale factor (2 for wavelets) is denoted by k and p denotes the intersection point of two consecutive filters, which for normalized Gabor filters is between 0 and 1. Other parameters are filter bandwidth γ and lowest and highest filter tuning frequencies, f_{min} and f_{max}, respectively.

The most useful approach is to select the filter bandwidth γ based on filter spacing k and overlap p. Equation for this can be found in the first line of Table 3.1, p and k are known and based on them γ is calculated as $\gamma = \frac{1}{\pi}\left(\frac{k+1}{k-1}\right)\sqrt{-\ln p}$.

Table 3.1: Parameter equations for filter frequency spacing.

p	k	γ	m	f_{min}	f_{max}
p	k	$\frac{1}{\pi}\left(\frac{k+1}{k-1}\right)\sqrt{-\ln p}$			
p	$\frac{1+\frac{1}{\gamma\pi}\sqrt{-\ln p}}{1-\frac{1}{\gamma\pi}\sqrt{-\ln p}}$	γ			
$e^{-\left(\gamma\pi\frac{k-1}{k+1}\right)^2}$	k	γ			
	$e^{-\frac{\ln f_{min}-\ln f_{max}}{m-1}}$		m	f_{min}	f_{max}
	k		$-\frac{\ln f_{min}-\ln f_{max}}{\ln k}+1$	f_{min}	f_{max}
	k		m	$\frac{1}{k^{m-1}}f_{max}$	f_{max}
	k		m	f_{min}	$f_{min}k^{m-1}$

Filter orientations

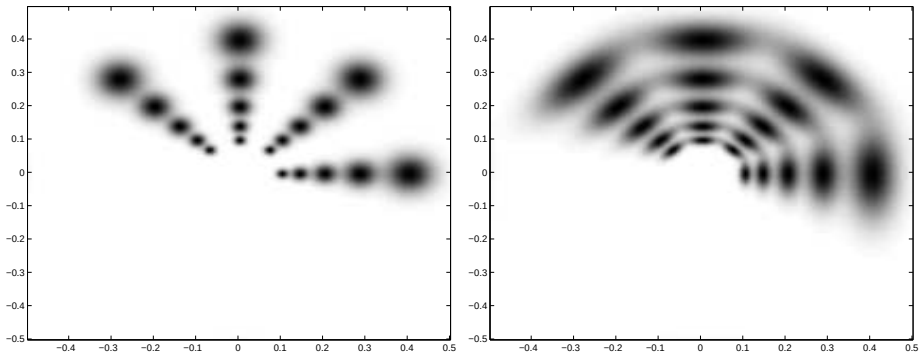
The equations for frequency spacing in Table 3.1 apply to both 1D and 2D filters, but the orientation spacing additionally depends on the number of orientations n and the minor axis bandwidth η. The analytical solutions can be derived and are collected into Table 3.2. Again the most useful equation is the selection of filter bandwidth η based on the filter overlap p and the number of filter orientations n, which gives equation $\eta = \sqrt{-\frac{(\eta\pi^2)^2}{4\ln p}}$. The overlap p here is assumed to be the same for both frequency and orientation spacing, but it can also be different over the orientations. Note that these equations are approximations. To get accurate filter intersection values for orientation spacing the whole elliptical envelope of the filter should be considered, not only its minor axis. However, the cost would be more complex equations because both η and γ would have to be included and the difference to the approximation presented here is generally small.

Table 3.2: Parameters equations for filter orientation spacing.

p	η	n
p	η	$\sqrt{-\frac{(\eta\pi^2)^2}{4\ln p}}$
p	$\frac{1}{\pi}\frac{\sqrt{-\ln p}}{\frac{\pi}{2n}}$	n
$e^{-\left(\frac{\eta\pi^2}{2n}\right)^2}$	η	n

Example of filter spacing

Two filter banks in the frequency space are presented in Fig. 3.3. Only the upper half of the filter bank is needed because responses on the lower half are complex conjugates. In Fig. 3.3(a) filters are closely located in the frequency space ($k = \sqrt{2}$) and therefore in the frequency direction the filters are relatively sharp (γ is large). The same value was used for η and consequently there are large gaps between filters in different orientations, and as a result some structures in the image with specific angles cannot be detected by the filter bank in Fig. 3.3(a). In Fig. 3.3(b) η has been solved based on equations in Table 3.2 and the gaps between filters in different orientations disappear.



(a) $\gamma = \eta \approx 2.35$ (b) $\gamma \approx 2.35$, $\eta \approx 1.03$

Figure 3.3: Examples of filter banks in frequency space, both use $m = 5$, $n = 4$, $p = 0.2$ and $k = \sqrt{2}$; (a) $\gamma = \eta \approx 2.35$; (b) $\gamma \approx 2.35, \eta \approx 1.03$.

3.2 Efficient computation

Gabor filters correspond to linear filters, so the most straightforward technique to filter is via convolution in the spatial domain. Standard convolution with Gabor filters can be improved by utilizing the separability of Gabor filters [8, 65] or their symmetry, antisymmetry and wavelet characteristics to reduce the number of needed multiplications and additions [74]. Certain approximation techniques, such as recursive Gabor approximation [92] or an approximation by decomposition into Gaussians [6] lead to much more efficient computation than normal spatial domain filtering by convolution, but limit the

free selection of filter bank parameters. The approximations also do not guarantee the beneficial feature space properties [44]. Therefore, without having to limit filter bank parameter selection to some special cases, the textbook solution, performing filtering in the frequency domain, often provides the most significant general improvement.

External knowledge on how the features are used is often neglected. The features are typically used in a multiresolution structure utilizing several frequencies and orientations, and the stable numerical support for Gabor filters is provided by a relatively small effective area of the filters. The multiresolution structure is similar to computing Gabor features on an image decomposed to a Gaussian pyramid, but using the Gaussian pyramid approach the selection of the filter frequencies would be limited. Here no limitations are applied as the unrestricted selection of all filter parameters is important for maximizing the usability of Gabor features for all application areas in image and signal processing.

Sometimes Gabor filter banks are optimized, for example, by finding maximal separation between two input classes [7], using a boosting techniques [12], or using a stochastic search method [86], which enables using a fewer number of Gabor filters leading to faster filtering. However, these methods often lead to non-homogeneous parameter sampling, violating (3.6) and (3.7), which in turn make invariant processing difficult because signal rotation and scaling cannot be handled by simple matrix manipulations as in (3.9) and (3.10). It would be possible to start with a filter bank respecting (3.6) and (3.7) and having a large number filter orientations and scales and then optimizing it, i.e., removing some of the filters which are not helpful for classification. However, the impact of this would be reduced because invariant search would still require computation of many of the removed filters. Therefore, these types of filter bank optimizations are not considered here. However, the presented efficiency improvements still apply if such methods are used.

In this section the most important characteristics of Gabor filters and filtering in both domains, spatial and frequency, are discussed in the context of computational complexity.

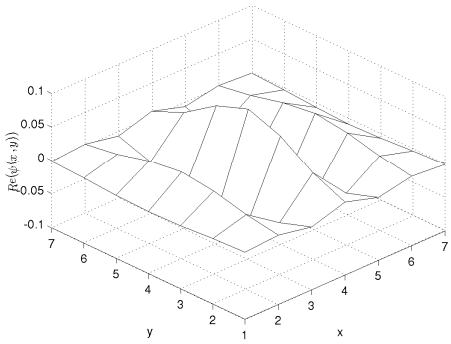
3.2.1 Related research

At the core of every Gabor feature is the filter response in (3.5) computed with the convolution. 1D convolution at one point requires $O(M)$ operations for a filter of length M. For a signal of length N the total complexity is $O(MN)$. For 2D images and filters the complexity becomes $O(M^2N^2)$. Due to these exhaustive computing requirements, efforts to decrease the complexity have been reported.

If a filter G can be expressed as a multiplication of two vectors, $G_{col} * G_{row}$, it is separable. For separable filters the convolution can be computed with two vectors, a column and a row vector, reducing the complexity from $O(M^2N^2)$ to $O(2MN^2)$. Horizontal and vertical Gabor filters are separable, the first vector is a sinusoidal with a Gaussian envelope and the second a Gaussian envelope. The separability can be extended to cover also the angle 45° [65], but for arbitrary orientations the input image must be rotated which increases the computational cost considerably.

Gabor filters also possess a significant degree of symmetry and anti-symmetry which can be utilized to reduce the number of multiplications needed for the convolution [74]. For example in Fig. 3.4 is a Gabor filter where the value 0.0620 or its negation is repeated

four times around the center of the filter. Using generic convolution the computation concerning those four points would be $0.0620v_1 + 0.0620v_2 + (-0.0620)v_3 + (-0.0620)v_4$ which includes four multiplications and three additions. The same can be calculated as $0.0620\,(v_1 + v_2 - v_3 - v_4)$, which reduces the number of multiplications to one. Similar re-ordering can be performed for many of the filter locations. Utilizing these properties for generic Gabor filtering is not sensible because of meager savings and complex implementation, but may be useful for an application using a fixed set of filters.



(a)

-0.0009	0.0031	-0.0065	0.0084	-0.0065	0.0031	-0.0009
-0.0031	0.0108	-0.0228	0.0293	-0.0228	0.0108	-0.0031
-0.0065	0.0228	-0.0483	0.0620	-0.0483	0.0228	-0.0065
-0.0084	0.0293	-0.0620	0.0796	-0.0620	0.0293	-0.0084
-0.0065	0.0228	-0.0483	0.0620	-0.0483	0.0228	-0.0065
-0.0031	0.0108	-0.0228	0.0293	-0.0228	0.0108	-0.0031
-0.0009	0.0031	-0.0065	0.0084	-0.0065	0.0031	-0.0009

(b)

Figure 3.4: A spatial domain Gabor filter: (a) figure of the real part of the filter; (b) real values of the Gabor filter.

The convolution can be performed in the Fourier domain, where it becomes the product between the Fourier transform of the the filter and the Fourier transform of the signal. The computation in the Fourier domain requires forward and inverse Fourier transforms for an input image. The standard discrete Fourier transform (DFT) is not used but the fast Fourier transform (FFT) which has the complexity $O(N \log N)$ for 1D signals. Compared to normal convolution in spatial domain with complexity $O(MN)$, the Fourier domain filtering is always faster unless the filter size, M, is very small. The complexity of 2D FFT is $O(N^2 logN)$. The Fourier domain enhancements are most significant due to their generality and superior overall efficiency. For filtering in both domains savings can be gained by using effective filter envelopes which will be presented next.

3.2.2 Effective filter envelopes

The support of a Gabor filter is infinite, but in the discrete domain the filter size is always limited. An effective filter envelope corresponds to the smallest support area which contains a predefined portion of the total filter energy. Filter coefficients outside the area can be discarded with a negligible effect on the accuracy. The support area of a Gabor filter is defined by the Gaussian part of the function. The support of a Gaussian function is elliptical [85], and therefore, the support area of a Gabor filter is also elliptical. However, elliptical envelopes are not very useful from the computational point of view, and the smallest rectangular envelope encapsulating the elliptical envelope will be used instead. Using filter envelopes, computing time is reduced significantly in spatial domain filtering and a considerable amount of memory is saved in frequency domain filtering.

1D ENVELOPES

The envelope has the standard Gaussian form,

$$N(t;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad . \tag{3.11}$$

The integral of the Gaussian function corresponds to the cumulative distributive function of the normal distribution,

$$G(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \quad . \tag{3.12}$$

The envelope can be solved from the inverse of the cdf: the point x where the cdf value is p,

$$x = G^{-1}(p;\mu,\sigma) = \{x : G(x|\mu,\sigma) = p\} \quad . \tag{3.13}$$

Substitutions for μ and σ can be solved from the filter equations in (3.1) and (3.2) yielding the substitutions $\mu = 0$ and $\sigma = \frac{\gamma}{\sqrt{2}|f_0|}$ in the time domain, and $\mu = f_0$ and $\sigma = \frac{f_0}{\gamma\pi\sqrt{2}}$ in the frequency domain. The envelope resides symmetrically around the mean of the distribution where the density is highest. Therefore, envelope end-points for the spatial domain filter with $e \in]0,1[$ energy are

$$x_s = \pm G^{-1}\left(\frac{1+e}{2}; 0, \frac{\gamma}{\sqrt{2}|f_0|}\right) \tag{3.14}$$

and for the frequency domain filter

$$x_f = f_0 \pm G^{-1}\left(\frac{1+e}{2}; 0, \frac{f_0}{\gamma\pi\sqrt{2}}\right) \quad . \tag{3.15}$$

Note that neither G nor G^{-1} can be solved analytically but effective approximation methods exist and are included in many numerical computing libraries.

2D ENVELOPES

The 2-D analog to the 1-D case is an effective ellipse region, but for the computational reasons the ellipse must be replaced with a rectangle covering the region. The separability of the Gabor filters can be utilized, i.e., it is sufficient to solve two 1-D problems including e_{1d} percent of the filter energy: total filter energy is then e_{1d}^2. The rectangular envelope can be determined by finding the ultimate dimensions of the effective area ellipse (see Fig. 3.5). The generic ellipse equation is $\frac{a^2}{x^2} + \frac{b^2}{y^2} = 1$. For a spatial domain filter a is set to x_s from (3.14) using the major axis bandwidth γ, and b is set to x_s applying the minor axis bandwidth η. To solve the rectangular envelope for a filter in orientation θ, the points in the derivative of the ellipse equation with slopes $c = \tan\theta$ and $c = -\tan(\frac{\pi}{2} - \theta)$ must be solved. These four points, (x_1, y_1), $(-x_1, -y_1)$, (x_2, y_2), and $(-x_2, -y_2)$, lie in the border of the envelope. The points must be rotated in relation to the origin by θ to get the final envelope for the spatial domain filter,

$$A_s = \begin{bmatrix} x_1 & y_1 \\ -x_1 & -y_1 \\ x_2 & y_2 \\ -x_2 & -y_2 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad . \tag{3.16}$$

Now, the smallest and largest x and y coordinates must be selected from A_s. For $\theta = n\frac{\pi}{2}$ $(n = 0, 1, 2, ...)$ one of the slopes goes to infinity and the four points are $(a, 0)$, $(-a, 0)$, $(0, b)$, and $(0, -b)$.

The envelope of a frequency domain filter is solved similarly, but the process is started by setting the values of a and b from (3.15). The frequency domain envelope is not centered at the origin, and therefore, f_0 must be added to the coordinates prior to rotation.

$$A_f = \begin{bmatrix} f_0 + x_1 & y_1 \\ f_0 - x_1 & -y_1 \\ f_0 + x_2 & y_2 \\ f_0 - x_2 & -y_2 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} . \tag{3.17}$$

The actual envelope is again determined by the smallest and largest x and y coordinates of A_f.

3.2.3 The highest necessary frequency

The frequency domain envelope has an important property: the higher bound defines also the highest frequency the filter is attuned to, f_{high}. The frequencies above f_{high} are not relevant for filtering as the filter response does not change whether they are present or not. Two examples can be seen in Fig. 3.6.

The high frequencies are removed when the signal is downscaled: if f_{high} is low enough, the image can be downscaled by a large factor leading to faster filtering. An input image can be downscaled before the filtering by the scaling factor

$$a_{sf} = \frac{0.5}{f_{high}} \quad , \tag{3.18}$$

where 0.5 is the Nyquist frequency. After the downscaling, the filter frequency f must be adjusted, $f_{new} = f_{old}\frac{0.5}{f_{high}} = a_{sf} f_{old}$.

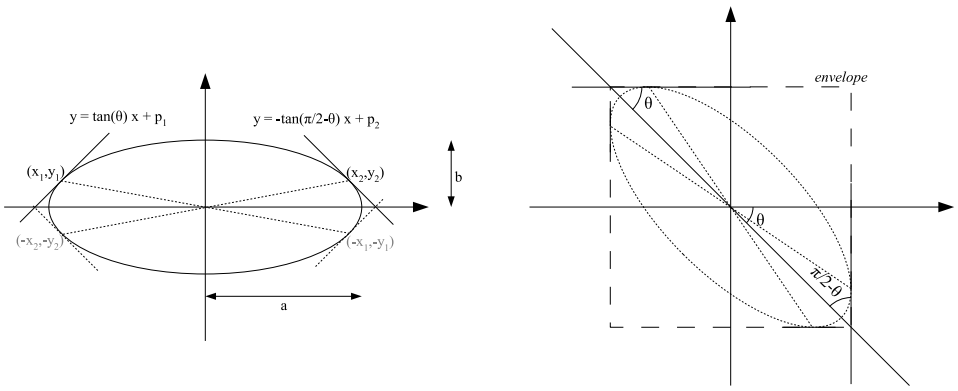


Figure 3.5: Determining the 2D effective envelope in the spatial domain.

Downscaling does not decrease the accuracy of the Gabor responses directly. The only negative effect of downscaling is that the responses are computed more sparsely, for example, if an image is downscaled by a factor $a_{sf} = 4$ the responses can be computed only for every sixteenth pixel in the original image. Loss of this kind of resolution does not matter usually as the responses do not change rapidly when the spatial location of the filter is changed slightly. However, there are some exceptions, for example, even a small deviation from the center of a perfect circle will cause a large change to responses.

To solve f_{high} the maximum distance from the origin to the furthest edge of the frequency domain envelope must be found. The center of the ellipse is located at the point $(f_0, 0)$, where f_0 is the frequency of the filter and its major axis a is directed along the u-axis and minor b along the v-axis (the frequency axes). The values of a and b are set to x_f from (3.15) by respectively applying major and minor axis bandwidths γ and η respectively. The distance from the origin is

$$d(x) = \sqrt{(f_0 + x)^2 + \left(\frac{b}{a}\sqrt{a^2 - x^2}\right)^2}, |x| \leq a \ . \tag{3.19}$$

The concept is illustrated in Fig. 3.7. The lower half of the ellipse can be ignored since it is symmetrical to the upper half. Then, x can be solved from the previous equation,

$$x = -\frac{a^2 f_0}{a^2 - b^2}, b > a, |x| \leq a \ . \tag{3.20}$$

The equation may result to a solution $x > a$ in which case $f_{high} = f_0 + a$, otherwise f_{high} can be found by applying x to (3.19), $f_{high} = d(x)$.

3.3 Optimal implementation framework

This section describes an optimal framework in which the given properties and results are applied to enhance practical computation efficiency. The optimality claim is based on the analytically derived complexities.

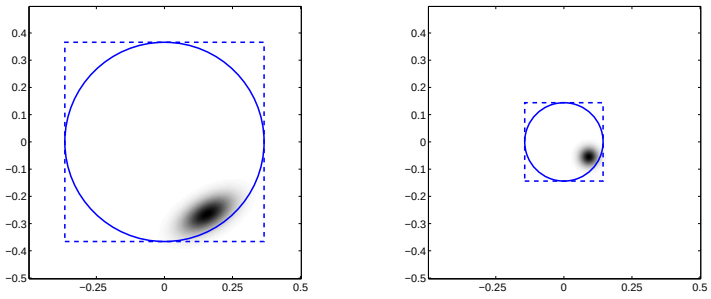


Figure 3.6: Examples of the highest necessary frequency for the filter in the frequency space. Circle marks the highest frequency and frequencies outside of the rectangle can be discarded; (a) $f = 0.3$, $\theta = \pi/3$, $\gamma = 2$, $\eta = 1$ which leads to $f_{high} \approx 0.37$; (b) $f = 0.1$, $\theta = \pi/6$, $\gamma = 1$, $\eta = 1$ which leads to $f_{high} \approx 0.14$.

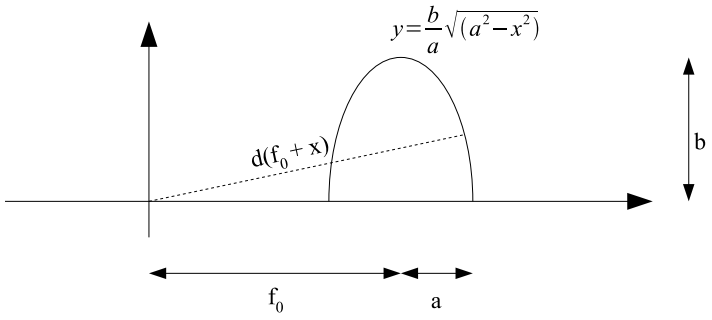


Figure 3.7: Determining f_{high} with the help of an ellipsoidal envelope of a 2D Gabor filter in the frequency domain.

3.3.1 Spatial domain filters

A diagram of the filtering in the spatial domain is presented in Fig. 3.8. The complexity depends directly on the size of the Gabor filter. The complexity for calculating the response at a single location is $O(M^2)$ and for an entire image $O(M^2N^2)$.

Prior to filtering the input image can be downscaled by the factor a_{sf} in (3.18). In practice, only the integer factors, or even more preferably, the power of two factors, are useful since then the downscaling corresponds to an average of a group of pixels and interpolation is not needed to avoid the aliasing effects. The complexity of the averaging is $O(N^2)$ and every pixel participates once to produce N^2/a_{sf}^2 pixels to the result image. When the image is downscaled, the frequency of a filter must be adjusted correspondingly by the factor a_{sf} leading to the filter envelope becoming smaller by the factor $\frac{1}{a_{sf}^2}$. The

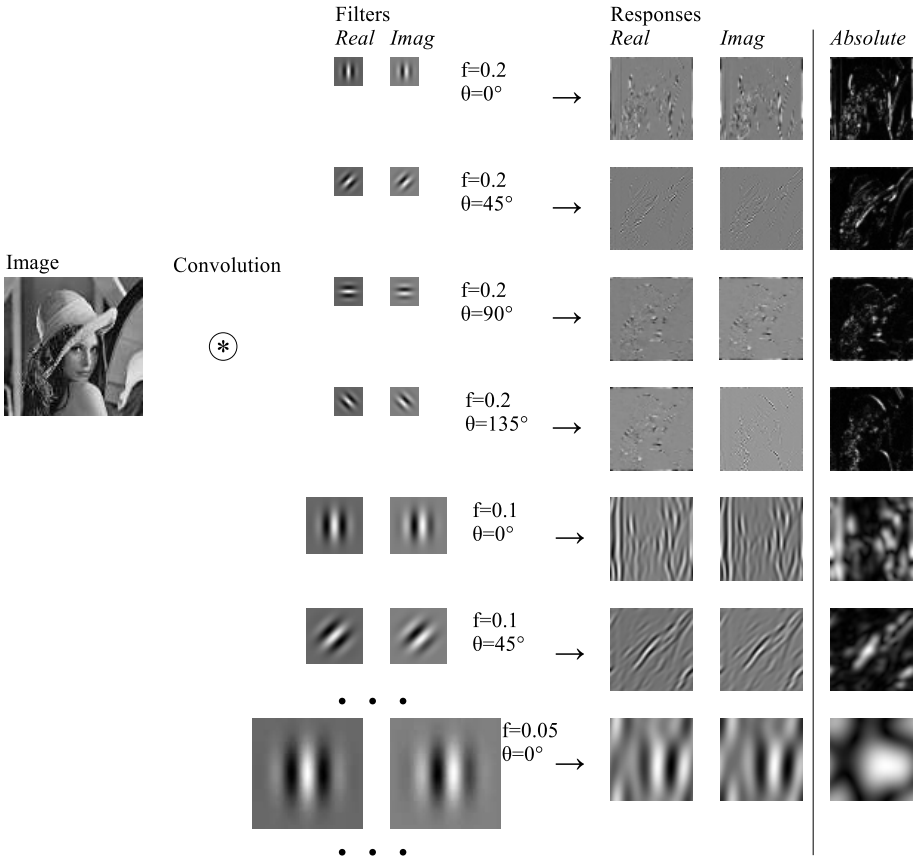


Figure 3.8: Diagram of spatial domain feature extraction.

complexity of computing a single response is now $O(M^2/a_{sf}^2)$ and for the entire image

$$O(\frac{N^2M^2}{a_{sf}^4}) \ . \tag{3.21}$$

The total complexity of computing K filter responses is either (without and with downscaling)

$$O(KM^2) \text{ or } O(N^2 + K\frac{M^2}{a_{sf}^2}) \tag{3.22}$$

depending on whether it is worth to downscale or not. The smaller one of the two complexities can be selected using the actual values of the parameters: in the first case there are very few points to filter (K is small) and downscaling would increase the complexity, and in the second case K is large enough for downscaling to become beneficial.

There are two separate steps in the filtering: the creation of a filter and the filtering itself. An algorithm for the creation of the spatial domain filter is presented in Algorithm 1 and an algorithm for filtering in the spatial domain in Algorithm 2.

Algorithm 1 *Create a spatial Gabor filter with parameters f, θ, γ and η.*

1: *Solve f_{high} using f, γ and η (Section 3.2.3).*
2: *Adjust f by a_{sf}, $f' = a_{sf} f$, from (3.18).*
3: *Solve filter envelope E for a filter with parameters (f', θ, γ, η) (Section 3.2.2).*
4: *Compute the filter g for filter area E with parameters (f', θ, γ, η).*

Algorithm 2 *Filter an image s in the spatial domain with a filter g (scaling factor a_{sf}) at locations $P = \{(x, y)_k\}$.*

1: *Downscale the image s by factor a_{sf}, $s \rightarrow s'$.*
2: ***for** All points p in P **do***
3: *Adjust the point's coordinate, $p' = p/a_{sf}$.*
4: *Compute response $r(p)$ by convolving the image s' in the point p' with the filter g.*
5: ***end for***

The symmetry and separability properties of Gabor filters could be applied here [8, 65, 74], but are neglected since their effect compared to the downscaling or frequency domain filtering would be small and they apply only to some specific filter configurations.

3.3.2 Frequency domain filters

A diagram of the filtering in the frequency domain is presented in Fig. 3.9. The complexity is dominated by FFT and IFFT, which is $O(N^2 \log N)$. The size of the effective envelope is not as crucial in the frequency domain as in the spatial domain since the image must be converted to the frequency domain and back whether the filter envelope is used or not. However, most of the coefficients will be close to zero and can be omitted to minimize memory requirements and also the number of floating point multiplications decreases, but the effect is small compared to the complexity of FFT.

An input image can be downscaled in a similar manner as in the spatial domain. Another option is to perform the downscaling in the frequency domain, which can be faster even though the spatial domain downscaling has lower complexity than the FFT. In the multiresolution structure the image needs to be converted only once to the frequency domain, but if the spatial domain downscaling is performed then the FFT has to be performed for all downscaled images. Downscaling in the frequency domain can be performed by discarding frequencies higher than f_{high} in the frequency space (see Fig. 3.6). The frequency domain downscaling by the scaling factor a_{sf} reduces the IFFT complexity to

$$O(\frac{N^2}{a_{sf}^2} \log \frac{N}{a_{sf}}) \ . \tag{3.23}$$

It should be noted that responses must be multiplied by the factor $1/a_{sf}^2$ to retain the correct response magnitude as compared to the non-downscaled results.

An algorithm for the filter creation is presented in Algorithm 3 and for the filtering in Algorithm 4.

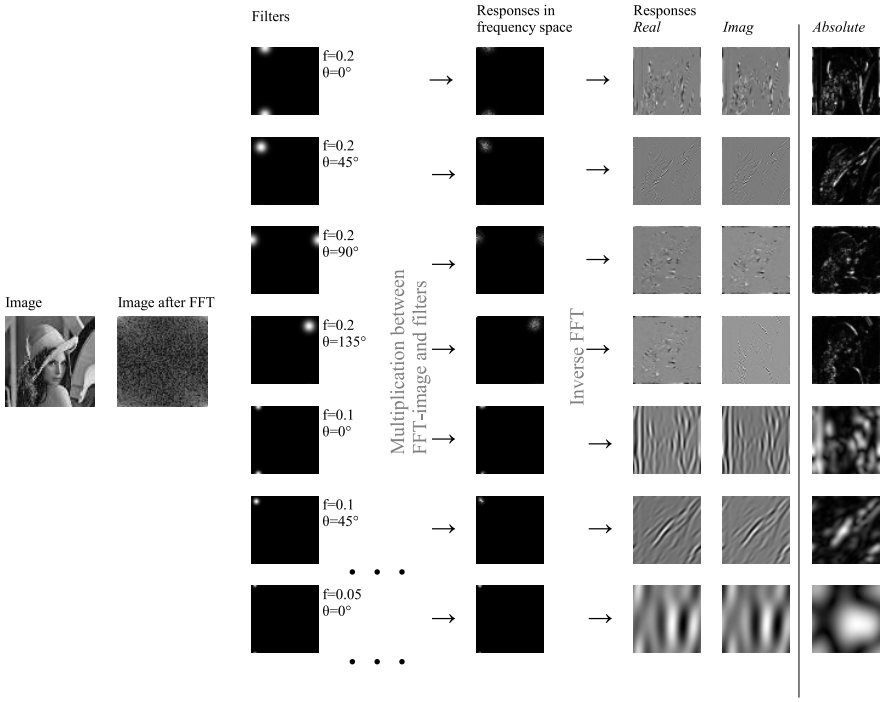


Figure 3.9: Diagram of frequency domain feature extraction.

Algorithm 3 *Create a frequency domain Gabor filter with parameters f, θ, γ and η.*

1: *Solve the filter envelope E with parameters (f, θ, γ, η) (Section 3.2.2).*
2: *Compute the filter g for filter area E with parameters (f, θ, γ, η).*
3: *Solve f_{high} using f, γ and η (Section 3.2.3).*
4: *Solve scaling factor a_{sf}, from (3.18).*

Algorithm 4 *Filter an image s in the frequency domain with a filter g (scaling factor a_{sf} and filter area E).*

1: *Initialize r' to the same size as s and set values to zero.*
2: *Compute FFT of the image, $s' = F(s)$.*
3: *Filter in filter area, $r'(E) = s'(E) * g$.*
4: *Crop frequencies above $\frac{0.5}{a_{sf}}$ out of r'.*
5: *Transform responses back to spatial domain with IFFT, $r = F^{-1}(r')$.*
6: *Scale response magnitudes, $r = r\frac{1}{a_{sf}^2}$.*

3.3.3 Multiresolution filtering

Multiresolution Gabor feature extraction is similar to the Laplacian pyramid [9]. A Laplacian pyramid represents an image as a pyramid of quasi-bandpassed images (see

Fig. 3.10), where the bottom of the pyramid represents the highest frequency content and is sampled densely, and the higher levels low frequency information sampled increasingly more sparsely. Each level of the pyramid reduces the filter's band limit by an octave, and the sample density can be reduced by the same factor, i.e., to half a resolution. The Laplacian pyramid was originally used for image compression, but the multiresolution Gabor features, such as simple Gabor feature space [48], yield to a similar structure. Both the computation time and memory are saved as the responses are computed at lower resolution than the original image.

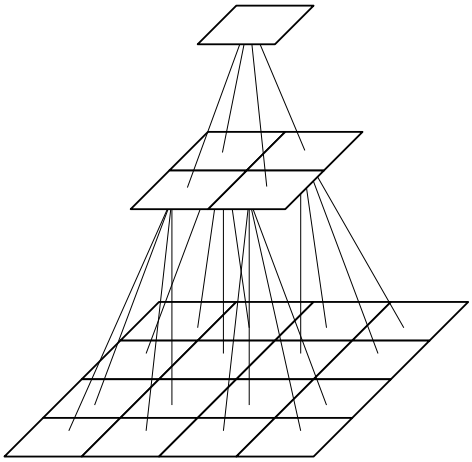


Figure 3.10: The structure of Laplacian pyramid.

Implementing the multiresolution structure with Gabor filters is straightforward. Algorithm 4 can be used as an example: the scaling factor a_{sf} is selected based on f_{high} and the resolution depends directly on the frequency. High frequency responses are sampled densely and lower frequencies increasingly more sparsely. It should be noted that octave spaced filter frequencies ($k = 2$) must be used if a similar structure to Fig. 3.10 is required: four lower level responses correspond to one response in the next level. If another value for k is used then the pyramid structure will not have as clear correspondences between responses at different levels. In that case, the pyramid levels can be downscaled to half size when the frequency is suitable. A similar technique can be found for example in the difference-of-Gaussians structure used in the SIFT interest point detector [55]. For example, with half-octave spacing ($k = \sqrt{2}$) there are always two pyramid levels with the same resolution corresponding to two consecutive filters, but for the third filter the pyramid level can be scaled to half size.

Using the multiresolution structure can be problematic in the following processing steps, for example, classification and object recognition. If the responses in different frequencies must be eventually used with the same resolution, sparse low frequency responses must be upscaled back to the required resolution. In practice, it is preferable to compute all responses directly at the same resolution and omit the upscaling procedure. The base resolution is selected based on the highest frequency filter, so processing time can still

be saved if the highest frequency allows.

3.3.4 Selecting the optimal filtering procedure

The decision whether the filtering should be performed in the spatial or in the frequency domain depends mainly on the number of points to be computed. If the entire image must be filtered, the frequency domain filtering practically always outperforms the spatial domain filtering as is evident from the complexities in (3.21) and (3.23) where the latter log-clause is very likely to be smaller than M^2/a_{sf}^2. When only K points are filtered, the decision of the filtering domain is based on the complexities in (3.22) and (3.23). The optimal decision tree is sketched in Fig. 3.11.

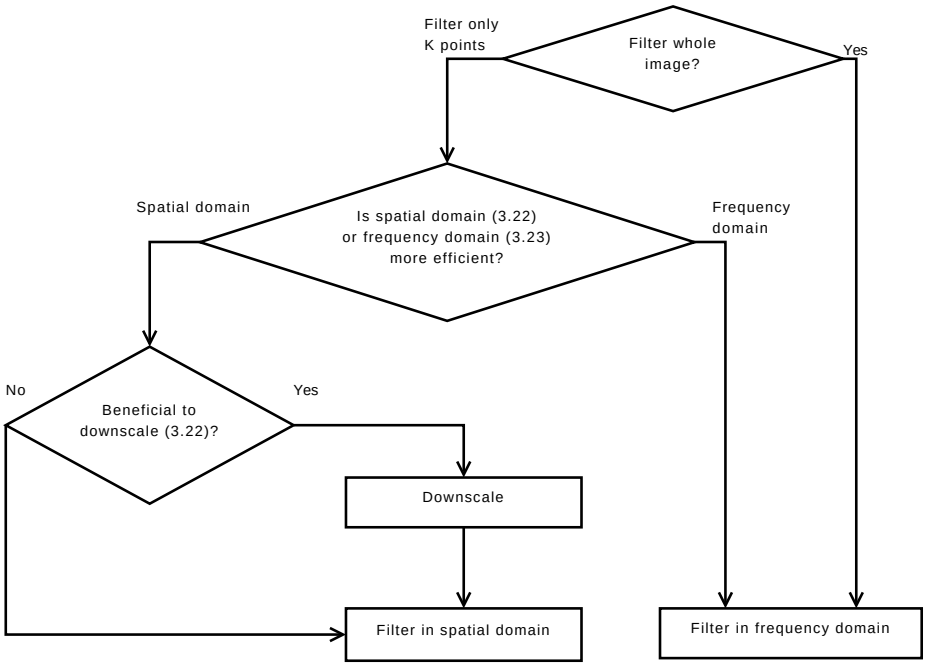


Figure 3.11: Procedure path for optimal Gabor filtering.

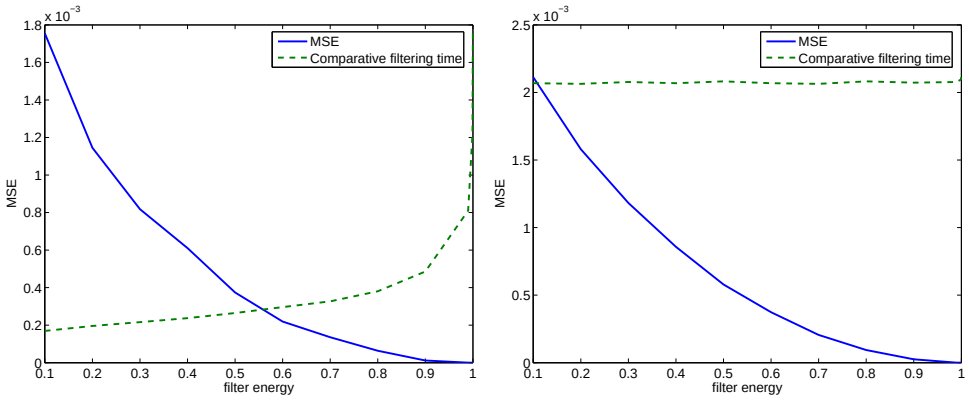
3.4 Results

Selection of the optimal filtering procedure was based on the analytically devised complexity equations, but inaccuracies in the responses induced by the proposed enhancements deserve a more practical treatment addressed in this section.

3.4.1 Error induced by effective envelopes

If effective envelopes are used to reduce the filter size, the required proportion of the filter energy must be selected. The discarded filter coefficients induce error to the responses. To study the behavior of the error, MSE (mean square error) was measured

for images containing Gaussian noise. MSEs as a function of the envelope energy are shown in Fig. 3.12. The effective envelope had a drastic impact on the computing time in the spatial domain, but it also induced a proportional inaccuracy to filter responses (Fig. 3.12(a)). The size of the envelope had practically no computational effect in the frequency domain as was expected because of FFT dominating the computation time, and therefore, the filtering should always be performed with a sufficiently large envelope as it provides better accuracy (Fig. 3.12(b)). Full size filters may be used with small images, but otherwise the envelope energy limit between $0.99 - 0.999$ seems to provide sufficiently accurate results while saving a large amount of memory.



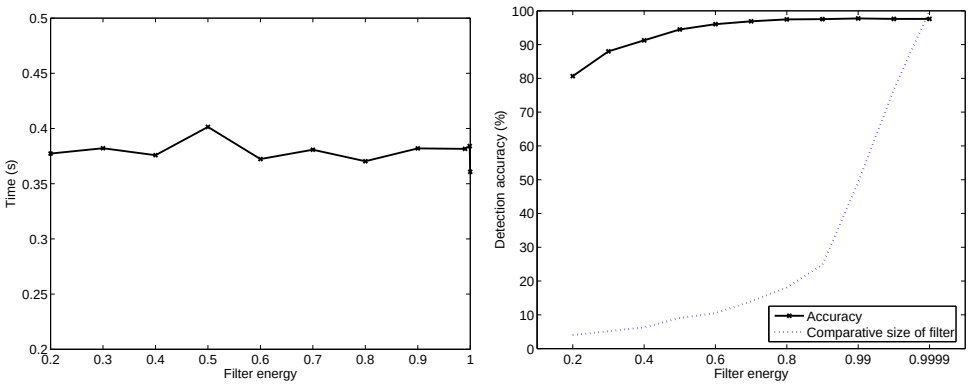
(a) (b)

Figure 3.12: MSE between the responses of the full size filter and the filter with an effective envelope (including relative time complexity): (a) In the spatial domain; (b) In the frequency domain.

3.4.2 Inaccuracy due to the efficiency improvements

This experiment tests the effects of efficiency improvements in a practical and important application: what is the effect to the speed and accuracy on the face detection experiment using the XM2VTS image database [58]. A detailed description of the test is presented in Section 5.2, and the focus here was only to evaluate how the proposed efficient computation methods affected the results and the computation time. Only the frequency domain filtering was tested, since the method always needs features from the whole image and therefore the frequency domain filtering is always faster.

In the frequency domain changing the filter envelope energy, which changes also the size of the filter, has only a small effect on the complexity (see Algorithm 4). Therefore, the speed measurement results in Fig. 3.13(a) present no surprise as there were no speed difference between the filters of different envelope energies. The detection accuracy in Fig. 3.13(b) however shows that the accuracy became steadily better with higher energy. A large frequency domain filter ($0.99 - 0.999$) should be used, but not excessively large as it leads to a waste of memory.



(a) (b)

Figure 3.13: Facial feature detection accuracy (see Section 5.2) and speed with varying filter envelope: (a) The filtering speed; (b) Detection accuracy.

The amount of potential downscaling depends on the filter frequencies, and more specifically, on the highest frequency component, f_{high}, of the highest frequency filter.

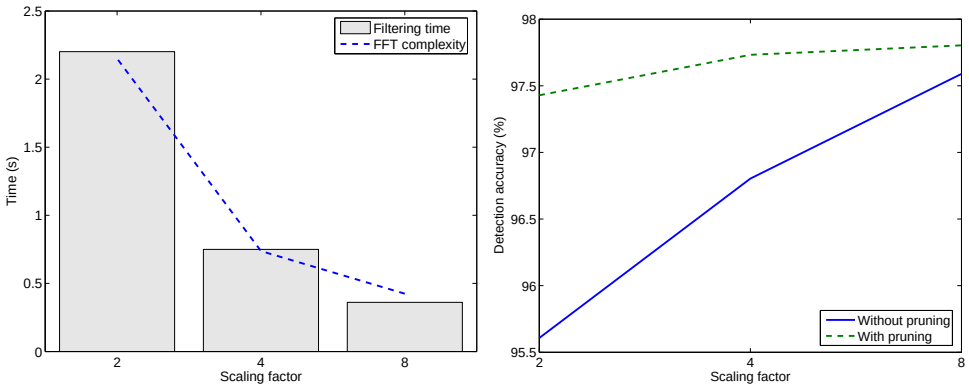
The scaling factor can be calculated by (3.18). If the image is further downscaled, then the frequencies the filter occupies begin to disappear. The maximum scaling factor for the highest frequency filter in this case was $a_{sf} = 8$. The results with various factors are presented in Fig. 3.14. The filtering speed (Fig. 3.14(a)) followed closely the complexity of the 2D FFT, $O(N^2 \log N)$. The detection accuracy results in Fig. 3.14(b) were surprising: the detection accuracy became slightly better when the image was scaled to a smaller size. This was mainly an artifact of the used performance measurement method: with a high resolution a false detection often caused a bunch of false detections in the neighboring pixels, while only one false detection occurred using a low resolution. To confirm this a pruning technique was added, which removes replicates from the vicinity of detected points, and using pruning the results stayed nearly constant over the different scaling factors. Unless the image features must be detected very accurately, the initial downscaling seemed to present no difficulties.

With the proposed efficiency improvements, the same accuracy as reported in [30] was achieved with nearly 1/50 of the original computation time.

3.5 Summary

This chapter introduced the concept of multiresolution Gabor features.

Firstly, computation of a single Gabor feature in 1D and 2D were studied followed by how several filter responses can be combined using a multiresolution structure and how the filter bank parameters should be selected. Secondly, based on this information, an efficient computation method for Gabor features was proposed utilizing an effective filter and the highest tuning frequency. Third, an optimal framework for computing Gabor features in the spatial or frequency domain was presented with information on how the most



(a) (b)

Figure 3.14: Facial feature detection accuracy and speed with image downscaling. (a) Filtering speed per image; (b) Detection accuracy (note the scale of y-axis).

efficient filtering procedure can be selected in a specific setting. Finally, experimental results of induced inaccuracies caused by the proposed optimizations were presented. It was found that the proposed computation improvements are able to significantly increase the feature extraction speed with negligible effect on the accuracy.

Chapter IV

Image feature classification and ranking

Classification and ranking of low-level features is needed for image feature detection and recognition when they are searched from observed images. During training phase, feature classes are created from training images by computing local image descriptors in marked points and then training a classifier for the local image feature. In the detection phase local image descriptors are created for all points in the image, and the classifier determines the best candidates for each class, for example, locations most likely representing eye-centers. To avoid the problem of creating a background class, representing anything else than the local image features which are being searched, one-class classifiers are used to only learn the local image feature classes.

Main local image descriptor in this work is the multiresolution Gabor feature and Gaussian mixture models (GMM) are proposed as the classifier. However, alternative methods can be used and their requirements will be discussed.

This chapter starts with a background and description of the used one-class classification methods. After that the local image feature detection method is presented. Additionally properties of the complex-valued Gabor feature space are studied because they have been noticed to have surprising effects on the classification performance.

4.1 Background and motivation for classification

One-class classification, also called novelty detection, outlier detection, or data description [83], can be used to detect uncharacteristic observations. One-class classification is necessary when samples can be obtained only from a single known class, for example, normal operation mode in motor condition monitoring where all failure modes are not known. One-class classification is also useful when the background class contains enormous variations making its estimation unfeasible, for example, a background class in object detection: the background class should contain everything except the object to be detected. One-class classifiers are used for this reason in this study.

Additional requirement for the classifier in this application is that it must sort the features into ranked order, it is not enough to simply decide whether a feature vector belongs to the class or not. If a classifier is trained for detecting local image features, for example eye-centers, in the detection phase the eye-center candidates must be returned in ranked order where the first one resembles the eye-center the most and the following ones less.

4.2 Gaussian mixture models

Many types of pdfs can be approximated with finite mixture models. Finite mixture models combine several single distribution forms to be able to approximate arbitrarily complex pdfs. The most common distribution function is the normal distribution (Gaussian distribution) because it is a well-understood distribution with useful properties for many application areas [85].

When the density of the data can be estimated, the easiest method for obtaining a one-class classifier is to set a density value threshold to the estimated probability density [82]. Gaussian mixture models (GMM) have been widely used in classification and general density estimation tasks, and they are also suitable for one-class classification. The expectation-maximization (EM) is a general method for estimating the mixture model parameters, and the EM algorithm is proved to converge to the global maximum likelihood estimate if the overlap between the Gaussians in the model is sufficiently small and there is a sufficient amount of data [56].

The multiresolution Gabor feature computed in a single location can be converted from the matrix form in (3.8) to a feature vector as

$$\boldsymbol{g} = [r(x_0, y_0; f_0, \theta_0) \;\; r(x_0, y_0; f_0, \theta_1) \ldots r(x_0, y_0; f_{m-1}, \theta_{n-1})]\,. \tag{4.1}$$

4.2.1 Multivariate normal distribution

The multivariate normal distribution of a D dimensional random variable can be defined as

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{4.2}$$

where $\boldsymbol{\mu}$ is the mean vector and Σ the covariance matrix of the normally distributed random variable $\boldsymbol{X}$. A multivariate Gaussian pdf is an elliptically contoured distribution where the equiprobability surface is a $\boldsymbol{\mu}$-centered hyperellipsoid [85].

The Gaussian distribution in (4.2) can be used to describe the pdf of a real valued random vector ($\boldsymbol{x} \in \mathbb{R}^D$). However, a similar form can be derived for complex random vectors ($\boldsymbol{x} \in \mathbb{C}^D$) as (e.g. [24])

$$\mathcal{N}^{\mathbb{C}}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\pi^D|\Sigma|} \exp\left[-(\boldsymbol{x}-\boldsymbol{\mu})^*\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{4.3}$$

where * denotes the adjoint matrix.

For a multimodal random variable, where values are generated by several randomly occurring independent sources instead of a single source, a finite mixture model can be

used to approximate the true pdf. If the Gaussian form is sufficient for single sources then a Gaussian mixture model (GMM) can be used in the approximation. However, the underlying distributions do not need to be Gaussians as GMMs can approximate any other distribution given a large enough number of components.

The GMM probability density function can be defined as a weighted sum of Gaussians

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_c,\Sigma_c) \tag{4.4}$$

where α_c is the weight of cth component. The weight can be interpreted as *a priori* probability that a value of the random variable is generated by the cth source, and thus, $0 \leq \alpha_c \leq 1$ and $\sum_{c=1}^{C} \alpha_c = 1$. A Gaussian mixture model probability density function is completely defined by a parameter list [16]

$$\boldsymbol{\theta} = \{\alpha_1, \boldsymbol{\mu}_1, \Sigma_1, \ldots, \alpha_C, \boldsymbol{\mu}_C, \Sigma_C\} \ . \tag{4.5}$$

The main problem is how the parameters in (4.5) can be estimated from the training data. The most popular estimation method is the expectation maximization (EM) algorithm. The problem with the algorithm is that it requires the number of Gaussians, C, as an input parameter. The number is often unknown and there is a strong motivation to apply adaptive unsupervised methods, such as that of Figueiredo-Jain [20] or the greedy EM algorithm [87]. The standard EM algorithm has been shown to outperform the adaptive methods if the correct number of mixture components is known, but in the absence of such knowledge the adaptive estimation algorithms give accurate and reliable results [68]. Of the two adaptive methods the Figueiredo-Jain was noted to provide more accurate results and it has been extended to complex values, and can therefore be directly applied to estimation of pdfs of complex multiresolution Gabor feature vectors in (4.1).

4.2.2 One-class classification using confidence with GMM

In our case confidence is used to estimate the reliability of a classification result where a class label is assigned to an unknown observation. If the confidence is low it is more probable that a wrong decision has been made. Intuitively a value of class conditional pdf at an observation corresponds to decision confidence for favor of the corresponding class: the higher the pdf value is, the more class instances appear similar to the observation. However, using pdf values directly can be difficult since they are arbitrarily scaled. Confidence values are always in the range $[0, 1]$.

The most straightforward use of confidence is to find a pdf value threshold for a class [82]. The threshold can be used to decide whether an observation is sufficiently similar to the class in question. The threshold can be selected based on the training data, for example, by selecting a pdf threshold for which half of the training data yields higher pdf values (median). Another possibility is to select the threshold using confidence: finding a threshold which includes a certain proportion of the total probability mass. The pdf type is not limited to a single Gaussian distribution but to a mixture of models with an arbitrary number of components. The selection method can be easily generalized for other types of pdfs.

To be a proper probability measure the confidence value should satisfy $\in [0, 1]$. For any support region $\mathcal{R}$ of the definition space Ω of the pdf, $\mathcal{R} \subseteq \Omega$, it holds that $0 \leq p(\boldsymbol{x}) < \infty$, $\forall \boldsymbol{x} \in \Omega$. The confidence value is defined via value κ which related to a non-unique confidence region $\mathcal{R}$ such that [68]

$$\int_{\Omega \setminus \mathcal{R}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \kappa \ . \tag{4.6}$$

The proposed confidence value is easily interpretable via the confidence region $\mathcal{R}$ which covers a proportion $1 - \kappa$ of the probability mass of $p(\boldsymbol{x})$ because for all probability distributions $\int_{\Omega} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$. It is clear that $\kappa = 1$ for $\mathcal{R} = \emptyset$ and $\kappa = 0$ for $\mathcal{R} = \Omega$. The confidence value has no meaning until the region $\mathcal{R}$ is defined as the minimal volume region. The minimal volume region is called the highest density region (HDR) [33]. For some distribution types the HDR can be non-unique (e.g., the uniform distribution). The proposed confidence value, $1 - \kappa$, corresponds to the smallest region which includes observation $\boldsymbol{x}$ and has a probability mass κ, defined as HDR.

A confidence value corresponds to a proportion of a probability mass in the area $\mathcal{R}_j$ for the class ω_j. In one-class classification the confidence region $\mathcal{R}_j$ can be used instead of the confidence value: a sample vector $\boldsymbol{x}$ is allowed to enter the class ω_j only if $\boldsymbol{x} \in \mathcal{R}_j$. If a sample is not within the confidence region of any of the classes it is classified to a background class. The background class is a special class and samples assigned to the class may need special attention depending on the application. For example, in a two-class problem where data is available only from one class the background class may represent another class with an unknown distribution.

To find the confidence region a reverse approach can be used to find a pdf value τ which is at the border of the confidence region: τ must be equal everywhere in the border, otherwise the region cannot be the minimal volume region [33, 39]. τ can be computed by rank-order statistics using the density quantile $F(\tau)$ (e.g., [33]) and by generating data according to the pdf. It is assumed that the gradient of the pdf is never zero in the neighborhood of any point where the pdf value is nonzero. An example of the confidence region can be seen in Fig. 4.1.

4.2.3 Confidence estimation algorithms

An analytical solution to the GMM confidence region cannot be solved and therefore estimation must be used. Estimation can be based on the GMM training data directly, or it can be based on randomly generated data derived from the estimated pdf. If confidence is determined based on the training data, volume of the confidence region does not necessarily have a direct relation to the confidence value: if a threshold is selected to include 50% of the training data, volume of the region may not be half of the total volume.

A pdf value threshold for $p(\boldsymbol{x})$ can be selected with the help of training data. First, a cumulative pdf value histogram H for the data $\boldsymbol{x}_{1..N}$ is created (Algorithm 5). Second, the threshold can be found using the cumulative histogram H and the required confidence value $c = 1 - F(\tau)$ using Algorithm 6.

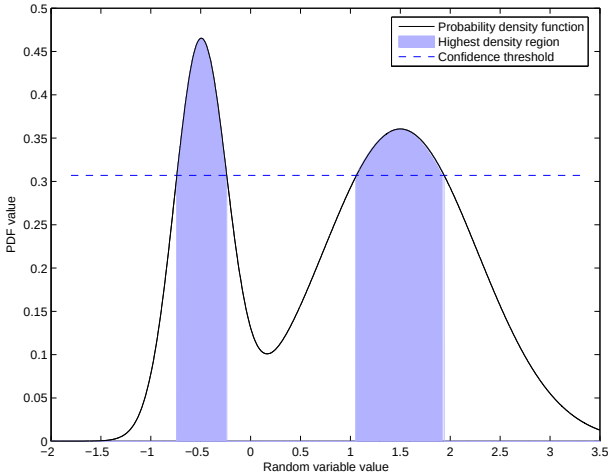


Figure 4.1: The highest density region (HDR) of a two-component GMM pdf and the corresponding threshold in one dimension. The confidence region is not a simple connected set.

Algorithm 5 *Create a cumulative confidence histogram H for pdf $p(\boldsymbol{x})$ with sample vectors $\boldsymbol{x}_{1..N}$ (training data)*

1: ***for*** $k = 1..N$ ***do***
2: *Calculate pdf value for* $\boldsymbol{x}_k$, $H_k = p(\boldsymbol{x}_k)$
3: ***end for***
4: *Sort H in ascending order, H=sort(H)*
5: *Return H.*

Algorithm 6 *Select a pdf threshold value τ for the confidence value c using the cumulative confidence histogram $H_{1..N}$*

1: *Select histogram position, m=round($c * N$)*
2: *Return $\tau = H_m$.*

The confidence value for a new sample $\boldsymbol{x}$ can be calculated using Algorithm 7.

Algorithm 7 *Return confidence value c for a sample vector $\boldsymbol{x}$ using the cumulative confidence histogram $H_{1..N}$ of the pdf $p(\boldsymbol{x})$*

1: *Calculate pdf value for the sample vector* $\boldsymbol{x}$, $p_x = p(\boldsymbol{x})$
2: *Select position of the closest pdf value to p_x in H, $m = argmin_i |H_i - p_x|$*
3: *Return $c = m/N$.*

In Algorithms 6 and 7 interpolation can be used instead of selecting the nearest value.

In the case of Gaussian mixture models, it may be beneficial to use randomly generated data. An algorithm for generating random data for any GMM is presented in Algorithm 8. The algorithm has been extended to multiple components from an algorithm presented in [85].

Algorithm 8 *Generate N random samples, X, for a D-dimensional GMM of C components with weights $\alpha_{1..C}$, mean vectors $\boldsymbol{\mu}_{1..C}$ and covariance matrices $\Sigma_{1..C}$*

```
k=1
for c = 1..C do
  T = chol(Σ_c) {Cholesky decomposition }
  {Number of generated samples depends on the weight of the component, α_c }
  for 1..round(α_c N) do
    Z = randn(1 × D) {Generate D independent normally distributed (μ = 0, σ = 1) random variables}
    X_k = ZT + μ_c
    k=k+1
  end for
end for
```

4.2.4 Experiments using confidence

The first experiment studies the required amount of data for distributions with increasing number of dimensions. The second experiment demonstrates the benefits of the confidence information on an image feature localization problem.

Data generation

The accuracy of the confidence and threshold computation methods with Algorithms 5, 6, 7 and 8 depends only on the amount of data, if the data and the estimated GMM represent the same underlying distribution. If that assumption holds, the only inaccuracy in the confidence values is caused by the limited amount of data. If the distributions deviate slightly from each other, which is typically caused by the GMM parameter estimation, the confidence values may be biased. If there is a large discrepancy between the distributions the confidence values may become completely useless, for example, all become binarized to either 0 or 1.

Here the relationship between data dimensionality and the required number of random samples is studied. To avoid the issue of distribution mismatch, a D-dimensional GMM pdf was generated semi-randomly and data was derived from the generated GMMs. Random data was generated with Algorithm 8 and then a pdf threshold was searched with Algorithms 5 and 6. For each value of D the number of required samples was evaluated repeatedly; each evaluation consisted of creating a semi-random covariance matrix and finding a number of samples at which the standard deviation of the found pdf threshold value for confidence $c = 0.5$ was varying at most by 1% from the mean value. The result is shown in Fig. 4.2.

The number of required samples increased linearly with the data dimensionality. Despite the fact that the size of the covariance matrix increases quadratically, and the number of required samples could be assumed also to grow quadratically, the linear dependency is as expected based on the data generating Algorithm 8: a D-dimensional sample is generated using D random numbers. In practice this means that the data generation is feasible even for high dimensional distributions because the required number of samples grows only linearly.

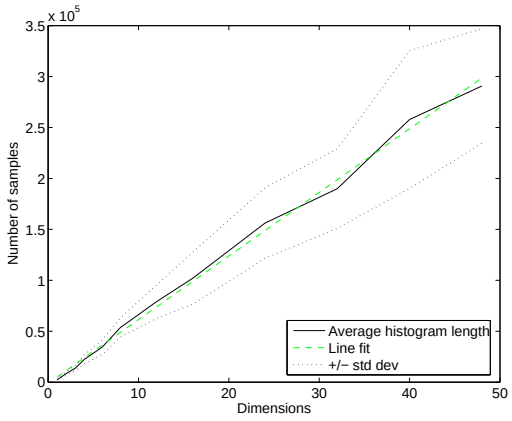


Figure 4.2: Required number of generated samples for a pdf threshold estimate ($c = 0.5$).

Image feature detection with confidence

The detection algorithm presented in Section 4.4 returns a fixed number of highest ranked image features found in the image. One obvious problem is that in the case when there is nothing to detect, a fixed number of points will still be returned. Image feature detection is followed by spatial constellation model search, which will then do useless work if there are spurious local image features. Also, when the object to be detected is present in the image, returning a fixed number of features may not be optimal, but the number should be decided adaptively.

In this example the use of confidence information is demonstrated in the face detection experiment, which is explained with more detail in Section 5.2. This specific example, results presented in Fig. 4.3, is concerned with searching one specific image feature, the left nostril, from an image. Fig. 4.3(a) shows a pdf surface from a GMM pdf trained for recognizing the left (in the image) nostril. Figs. 4.3(b) and 4.3(c) show only the confidence regions corresponding to 0.01 and 0.50 confidence values. The correct image feature location and very little else was included in the 0.50 confidence region, and even the 0.01 confidence region discarded very large part of the image.

4.3 One-class SVM (support vector machine) classifier

The single-class SVM (support vector machine) classifier used in this study was a one-class classifier based on a μ-SVM classifier [79]. The one-class SVM algorithm starts with a set of points and estimates a region with a specified fraction of the points. Several different regions are possible, which region is selected depends on the kernel and used regularization. Internally the algorithm functions by mapping the data to a feature space using a kernel and finding a hyperplane separating the data from the origin with a maximum margin. Fig. 4.4 shows an example. A new data point is classified based on which side of the hyperplane it falls on the feature space.

(a) (b) (c)

Figure 4.3: Example of using density quantile for defining confidence regions: (a) pdf value surface for the left (in the image) nostril class; (b) confidence threshold 0.01 ($F(\tau) = 0.99$); (c) confidence threshold 0.50 ($F(\tau) = 0.50$).

The algorithm starts with a set of unlabeled training data

$$X = x_1, \ldots, x_m \subset \mathcal{X}, \tag{4.7}$$

where m is the number of observations and $\mathcal{X}$ is some set, usually $\mathbb{R}^N$. As usual with SVM algorithms, data point x is mapped to a dot-product space, $\mathcal{X} \to \mathfrak{X}$, with $\Phi(x)$ and the feature space is defined so that a simple kernel can be used to evaluate the dot-product (denoted by $\langle \cdot \rangle$),

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle, \tag{4.8}$$

such as the Gaussian, which is often called RBF (radial basis function) kernel,

$$k(x, x') = e^{-\|x - x'\|^2 / \sigma}. \tag{4.9}$$

The algorithm returns value $+1$ for a small region capturing most of the training data and -1 elsewhere. To separate data from the origin, the following quadratic program is solved:

$$\underset{\boldsymbol{w} \in \mathfrak{X}, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}}{\text{minimize}} \qquad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu m} \sum_i \xi_i - \rho \ , \tag{4.10}$$

$$\text{subject to} \qquad \langle \boldsymbol{w}, \Phi(x_i) \rangle \geq \rho - \xi_i, \xi_i \geq 0 \ . \tag{4.11}$$

Here, $\nu \in]0, 1]$, is a parameter which controls the number of outliers and support vectors, ξ_i are the slack variables, and ρ is the the margin to hyperplane. Slack variables ξ_i are used to penalize outliers in the objective function. The decision function is

$$f(x) = \operatorname{sgn}\left(\langle \boldsymbol{w}, \Phi(x) \rangle - \rho\right), \tag{4.12}$$

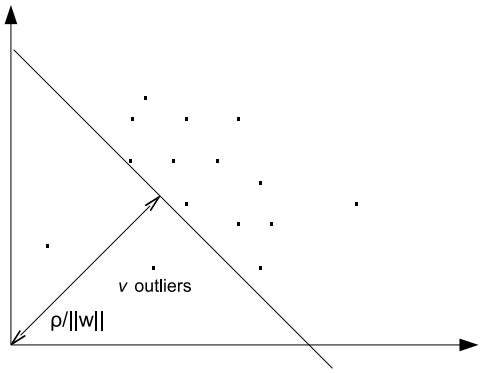


Figure 4.4: A hyperplane separating data from the origin with some outliers.

which is defined so that $\mathrm{sgn}(z)$ equals 1 for $z \geq 0$, -1 otherwise. Using multipliers $\alpha_i, \beta_i \geq 0$ for the weights of support vectors and outliers respectively a Lagrangian is introduced,

$$L(\boldsymbol{w}, \xi, \rho, \alpha, \beta) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu m}\sum_i \xi_i - \rho - \sum_i \alpha_i(\langle \boldsymbol{w}, \Phi(x)\rangle - \rho + \xi_i) - \sum_i \beta_i \xi_i \ . \quad (4.13)$$

Setting the derivatives with respect to the primal variables $\boldsymbol{w}$, ξ and ρ equal to zero, yields to

$$\boldsymbol{w} = \sum_i \alpha_i \Phi(x_i) \ , \quad (4.14)$$

$$\alpha_i = \frac{1}{\nu m} - \beta_i \leq \frac{1}{\nu m}, \sum_i \alpha_i = 1 \ . \quad (4.15)$$

The decision function (4.12) can now be transformed using (4.13) and (4.8) into

$$f(x) = \mathrm{sgn}\left(\sum_i \alpha_i k(x_i, x) - \rho\right) \ . \quad (4.16)$$

The dual problem can be obtained by substituting (4.14) and (4.15) into Lagrangian L, (4.13), and by using the kernel, (4.8),

$$\begin{aligned} \underset{\alpha\in\mathbb{R}^m}{\text{minimize}} \quad & \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(x_i,x_j) \ , \\ \text{subject to} \quad & 0 \le \alpha_i \le \frac{1}{\nu m} \ , \\ & \sum_i \alpha_i = 1 \ . \end{aligned} \tag{4.17}$$

It can be shown that at the optimum the two inequality constraints in (4.11) become equalities if α_i and β_i are not equal to zero. Exploiting any such $\alpha_i > 0$ ρ can be calculated,

$$\rho = \langle \boldsymbol{x}, \Phi x_i \rangle = \sum_j \alpha_j k(x_i, x_j) \ . \tag{4.18}$$

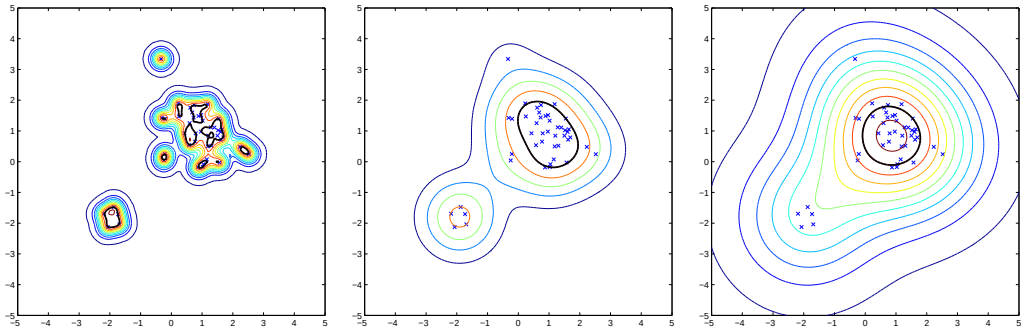
Interpretation of the parameter ν follows. When the parameter approaches zero, the problem becomes a hard margin problem, since the penalization of errors is then infinite. The problem is still feasible but the margin may become negative. Overall, the parameter ν characterizes the fraction of outliers and support vectors. Outliers are the points which are on the wrong side of the hyperplane. As a rule of thumb, ν is the lower bound for the fraction of support vectors and upper bound for the fraction of outliers.

Some examples of the effects of parameters ν and the RBF kernel size σ are presented in Fig. 4.5. The SVM classifier has been created for a simple 2D problem, where there are two distinct sets of points. One problem case can be seen in Fig. 4.5(a), where the kernel size is too small and the classifier is overly complex. On the other hand, a too large kernel size may create a too simple solution, as can be seen Fig. 4.5(c). When the parameter ν is small, see Fig. 4.5(d), only a few outliers are allowed and the distribution may again become too complex. Large ν on the other hand leads to a large number of outliers, see Fig. 4.5(e).

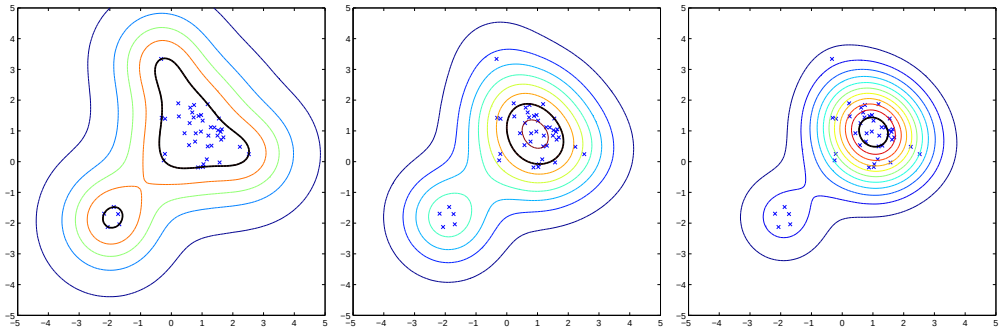
By default this SVM classifier only outputs a binary classification decision based on which side of the hyperplane the point falls. This is not suitable for use in the image feature detection described in this thesis; image feature candidates must be available in ranked order, the most likely candidates having the largest values. While the theory is not as well formulated as in the case of GMM pdfs, this is still possible. The classification decision in (4.16) first computes the distance to the hyperplane, which is positive when the point belongs to the class, and then uses the sgn$(\cdot)$ function to binarize it to either 1 or -1. If the sgn$(\cdot)$ function is omitted, decision is a real valued number, the higher it is the further away on the inclusion side the point is from the hyperplane, and therefore, in the most "dense" part of the distribution. This method is used to rank image features with a one-class SVM classifier in this thesis.

4.4 Supervised image feature detection method

This section presents the supervised image feature detection method, first the training phase and then the detection phase. Requirements for local image descriptors and classifiers are also discussed.



(a) $\nu = 0.5, \sigma = 0.2$ (b) $\nu = 0.5, \sigma = 0.8$ (c) $\nu = 0.5, \sigma = 0.2$



(d) $\nu = 0.1, \sigma = 1.0$ (e) $\nu = 0.5, \sigma = 1.0$ (f) $\nu = 0.9, \sigma = 1.0$

Figure 4.5: Examples of the SVM classifier and effects of ν and RBF kernel-size, σ, parameters. Thick line presents the decision boundary for class inclusion; (a)-(c) Different kernel size σ, constant ν; (d)-(e) constant kernel size σ, varying ν.

4.4.1 Training the detector

The local image feature detector training method is presented in algorithmic form in Algorithm 9 and visualized in Fig. 4.6 (detector for left eye-centers). The eye-centers must be annotated in the training images, and local image descriptors for those areas are computed. After the descriptors have been computed for all annotated positions in the training images, a classifier is trained. In our case, the classifier is a one-class classifier without a background class.

Algorithm 9 *Training a local image feature detector*

1: ***for all*** *Training images* ***do***
2: *Align and normalize image to represent an object in a predefined standard pose*
3: *Compute multiresolution Gabor features at given landmark locations*
4: *Normalize the features*
5: *Store the features to the sample matrix P and their corresponding class labels (class numbers) to the target vector T*

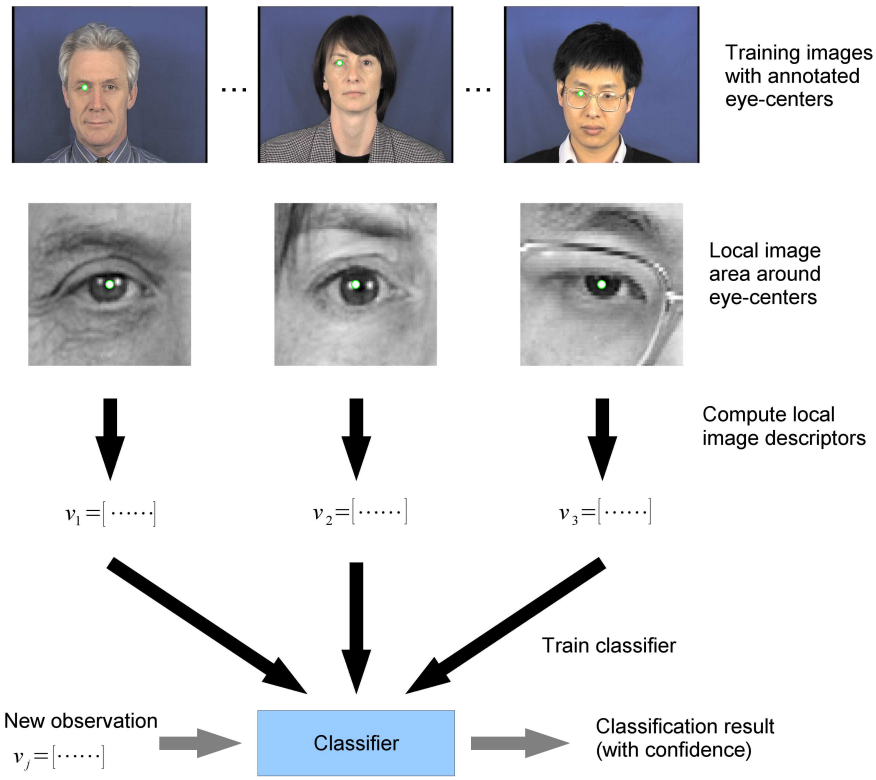


Figure 4.6: A conceptual diagram of local image feature creation for an local image feature detector (left eye-center).

6: ***end for***
7: *Train a one-class classifier using samples in P separately for each class*

In the Algorithm 9 the training images must first be aligned to a standard pose: the pose representing objects in the same scale and orientation. After the images have been transformed to the standard pose, multiresolution Gabor features in (3.8) are computed at annotated landmark image feature locations. Feature matrices can be energy-normalized if a complete illumination invariance is required. Each feature matrix is reformatted into a vector form in (4.1) and stored in the sample matrix P along with the corresponding image feature labels, T. Finally, pdfs (probability density functions) over the complex feature vectors are estimated for each image feature class separately. The standard classifier has been a GMM classifier, but generic requirements for the classifier are presented later.

4.4.2 Detection

Detection is performed as presented in Algorithm 10. The detection procedure is visualized in Fig. 4.7. Local image descriptors are computed and classified separately for all points in the image. Because a one-class classifier is used and there is no background class, the classifier only outputs a likelihood or probability value for a descriptor to belong to the specific feature class. Complete likelihood description (likelihood image) can be computed from the whole image and the highest values can be selected as the most prominent image feature candidates (see Fig. 4.7). The only requirement for the classifier is that the value is higher the more the described point resembles the trained class.

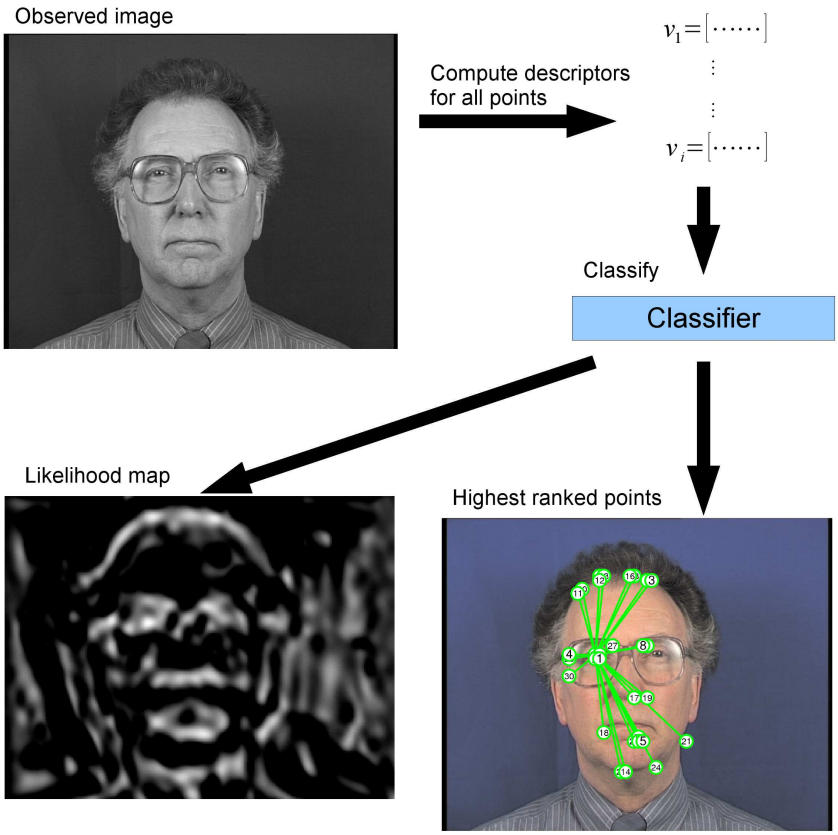


Figure 4.7: A conceptual diagram of image feature detection (left eye-center).

Algorithm 10 *Detect K best image features of each image feature class from image I*

1: *Normalize image if needed*
2: *Compute multiresolution Gabor features $G(x, y; f_m, \theta_n)$ for the whole image $I(x, y)$*

```
3: for all Scale shifts do
4:   for all Rotation shifts do
5:     Shift Gabor features
6:     Normalize Gabor features
7:     Apply the classifier to compute likelihood values for all classes and for all (x, y)
8:   end for
9: end for
10: Sort the the likelihood of image features values for each class
11: Return the K best candidates of each image feature class
```

If the observed images vary heavily in their photometric quality (e.g., large brightness and contrast differences), they can be first normalized. From the normalized image multiresolution Gabor features are extracted at every spatial location and likelihood values for all image feature classes are computed for all invariance shifts. If Gabor features were energy normalized in the training phase the same normalization must be applied here. To save memory, only some predefined portion of the highest likelihood values can be stored instead of storing all likelihood values. After the shifts have been inspected the best image feature candidates are returned and sorted based on the likelihood values. With this approach one location may represent more than one image feature, but each feature can be assigned to one pose only.

4.4.3 Requirements for the local image descriptor

The two algorithms (Algorithm 9 and Algorithm 10) assume that multiresolution Gabor features are used as local image descriptors. However, in general the selection of the descriptor is free: any method can be used, but the method has to be fairly fast during the detection phase since local descriptors must be computed for all points in the image, or at least for a reasonably large portion of all points if sparse sampling is used. Sparse sampling means that a part of the points are omitted either systemically (e.g., handle every second or fourth pixel) or by adaptively sampling more densely in areas where likelihood values have been large. For the same reason also the classifier has to be efficient in processing a large number of feature vectors. Fortunately with most classifiers only the training phase is computationally heavy.

4.4.4 Requirements for the classifier

There are few challenges in classification of multiresolution Gabor features for image feature localization. Firstly, the features are complex-valued which many classifiers do not explicitly consider. Secondly, the localization process is simplified if the background class can be avoided leading to use of one-class classifiers, which are not as completely studied as more typical two-class classifiers. Thirdly, the feature-space of multiresolution Gabor image features can be surprisingly complex for certain types of even simple signals, which can cause problems for some classifiers. The properties of the feature space are studied in Section 4.5. Fourthly, as already mentioned, the classifier should be fast, as exhaustive search over the whole image or at least a large portion of it is performed.

Based on the above mentioned requirements Gaussian mixture model (GMM) classifier was used in this study. Gaussian mixture models can be extended to complex values,

GMM is suitable for single-class classification and are reasonably fast. The remaining unsolved property is the complex behavior of the feature space and GMM's difficulty in estimating it properly from limited training data. The selection of an optimal classification strategy is still an open issue, but fortunately GMM performs well in practice. Additionally, for the GMM classifier *confidence* can be defined from a solid probabilistic background.

A variant of the support vector machine (SVM) was tested as an alternative. For the SVM classifier the main problem is speed: usually quite large number of support vectors is required to achieve sufficient detection accuracy and this leads to slow classification. The classifier can be tuned to use fewer number of support vectors, but the downside is that detection accuracy suffers, and the main benefit as compared to GMM is lost, namely, that the SVM classifier cannot anymore learn the complex feature spaces better.

Gaussian mixture models and multiresolution Gabor features have two interesting, although not unique, properties which are demonstrated in Fig. 4.8. First of all, estimation of GMM is similar to clustering and the found mixture components can be illustrated by searching the closest matches in the training set. This and the fact that images can be reconstructed from multiresolution Gabor features provides a nice property that both the classifier and the image features can be examined visually.

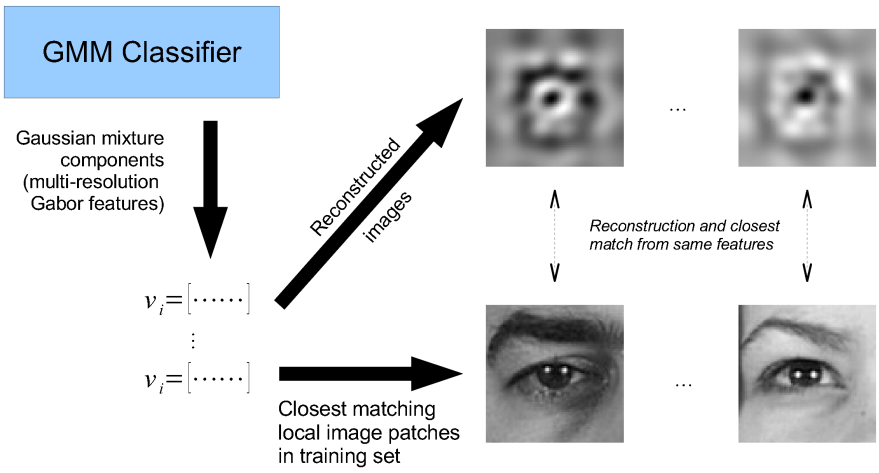


Figure 4.8: Properties of GMM and multiresolution Gabor-features enable visualization of classifier and image feature extraction performance.

4.5 Properties of complex-valued Gabor feature space

If an assumption about statistical properties of multiresolution Gabor features is made, it is usually assumed to be Gaussian [73], [90] or a mixture of Gaussians [80]. It is however easy to show that even for a simple pattern the Gaussian distribution may be surprisingly far from the actual distribution and can lead to non-optimal results. Here,

results from various situations where the responses of a single Gabor filter have non-Gaussian properties are presented with some experimental results with multiresolution structures where the distributions become even more complex.

In this section properties of Gabor features are studied experimentally with a 1D signal without loss of generality in 2D. The experiments demonstrate situations where the Gaussian assumption severely fails.

4.5.1 Sensitivity to small misalignments

In Fig. 4.9 is a signal with two spikes at distance of $2d$, where $d = 50$, and a Gabor filter with frequency $f = 1/d$. The spikes are located one wavelength off the center of the Gabor filter and the filter has a strong response when located exactly between the two spikes. However, when the filter is moved slightly away from the center, there will be large change in the complex responses. To properly interpret the behavior in Fig. 4.9 it must be imagined in 2D Re-Im space which will be considered later.

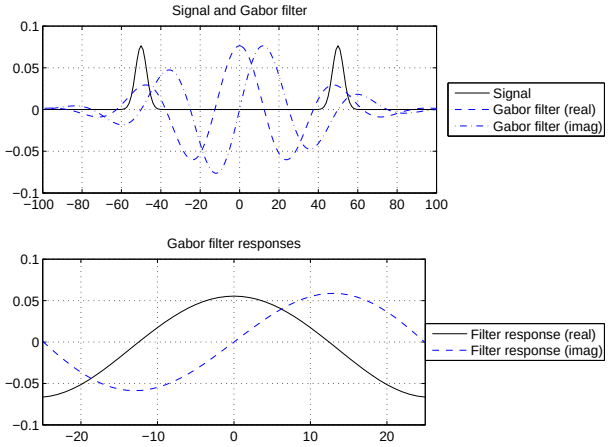


Figure 4.9: The upper figure shows a signal with two spikes at distance $2d$ ($d = 50$) and real and imaginary parts of a Gabor filter with frequency $f = 1/d$ located at the center of two spikes. The lower figure shows the filter's real and imaginary responses as a function of misalignment.

The reason for this non-Gaussian behavior is that the real-valued (cosine) part of the Gabor filter captures symmetric properties of the signal and the imaginary-valued (sine) part the anti-symmetric properties. When the filter is exactly at the center of the signal there is perfect symmetry and a strong real-valued response. The signal does not have any anti-symmetry and correspondingly the imaginary-valued response is zero. When the filter is shifted slightly in either direction, the symmetry starts to wane and the real-value response to decrease slowly, but the anti-symmetry grows rapidly causing a fast change in the imaginary-valued response.

There are infinitely many similar, symmetric or anti-symmetric, configurations where the activated areas of cosine and sine parts are shifted to break the Gaussianity. This

effect can be either seen as a beneficial or a harmful property. On the positive side, the signal can be located exactly because the responses change rapidly as the filter location deviates from the center (Fig. 4.9). On the negative side, for example, object detection may become difficult if Gabor responses change rapidly around a target object. Stability can be increased by using the filter magnitudes instead of complex responses, but this significantly reduces the representative power of the filter, and the problem should be rather avoided by a proper classification method or a similarity measure.

4.5.2 Effect of changes in the signal

Small perturbations in the location can cause surprisingly large changes in the filter responses and misalignments frequently occurs in practice. Combined with other potential changes in signals, multiresolution Gabor feature values may form a very complex structure. The response space from the previous example (Fig. 4.9) is presented in Fig. 4.10(a), where the position of the Gabor filter is changed ± 5 units from the center. There is only a small change in the x-axis (real values), but a large variation in the y-axis (imaginary values).

If the distance between the spikes changes the spikes are not exactly at the distance of the Gabor filter wavelength. The imaginary response stays at zero because the signal stays symmetric. An example of the response space is presented in Fig. 4.10(b) where $d = 30..50$. When the spikes are moved closer to each other, the response will become zero at a distance $2\frac{3}{4}d$ as the spikes are located at points where the real part of the Gabor filter crosses zero. If spikes are moved even closer the response becomes negative.

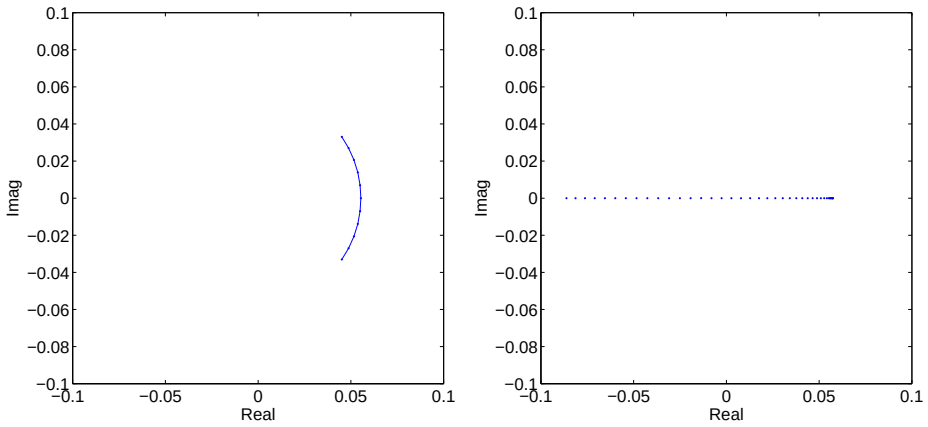
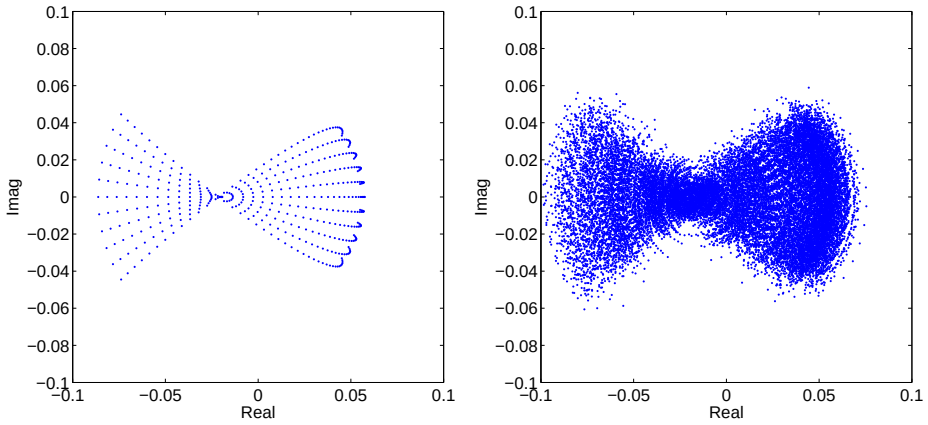


Figure 4.10: Response space of Gabor features with various alterations to the signal in Fig. 4.9: (a) Location of the Gabor filter changed by ± 5 units; (b) Distance between spikes changed, $d = 30...50$.

When both types of alterations are combined, the location of the Gabor filter varies by ± 5 and $d = 30..50$, the response space becomes increasingly complex as can be seen in

Fig. 4.11(a). However, addition of a large amount of Gaussian noise to the signal does not cause a dramatic change (Fig. 4.11(b)). Furthermore, if the signal's contrast changes, i.e., how large its amplitude is compared to the background, the responses will be linearly scaled towards or away from the origin (not presented here).



(a) (b)

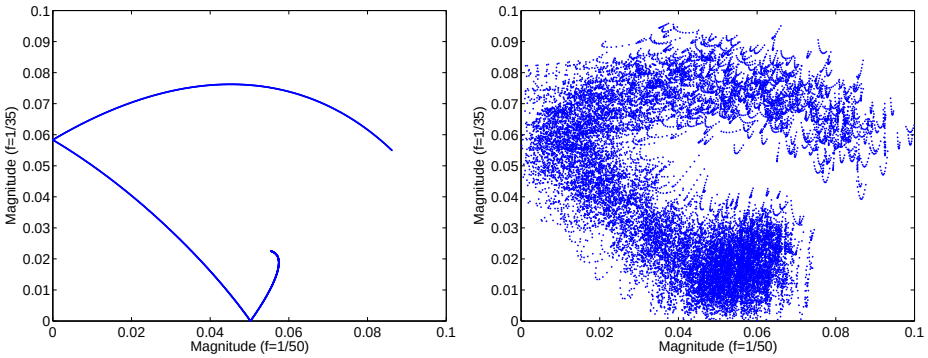
Figure 4.11: Response space of Gabor features with various alterations to the signal in Fig. 4.9: (a) Location of the Gabor filter changed by ± 5 units and distance between spikes changed, $d = 30...50$; (b) Gaussian noise added to the signal.

Previous examples used only one Gabor filter, but with multiresolution Gabor features many filters at many frequencies and orientations are used. With two Gabor filters the response space becomes 4-dimensional (two complex valued responses) which is not easily demonstrated. A simplified demonstration of multiresolution responses is presented in Fig. 4.12(a) where in the x-axis are the response magnitudes of the previous example (complex responses in Fig. 4.10(b)), and in the y-axis magnitudes of filter responses for a filter at $f = 1/35$. The resulting response space is far from Gaussian, and the feature space does not get any less complex when the filter position is changed by ± 5 units and noise is added as shown in Fig. 4.12(b).

4.5.3 Experiments

Experiments with generated and real data were conducted in order to demonstrate difficulties for standard classifiers assuming Gaussianity to classify multiresolution Gabor features as compared to more complicated classifiers. However, simple classifiers are needed because multiresolution Gabor features are often involved in low level processing, where efficiency is important.

Two different one-class classifiers have been used in these experiments, a classifier based on Gaussian mixture models (GMM), presented in Section 4.2, and a one-class support vector machine classifier, presented in Section 4.3.



(a) (b)

Figure 4.12: Response space of magnitudes of two Gabor filters: (a) On x-axis response magnitudes for a Gabor filter with $f = 1/50$ and on y-axis $f = 1/35$ for a signal with distance between spikes $d = 30..50$; (b) Location of Gabor filter changing by ± 5 and Gaussian noise.

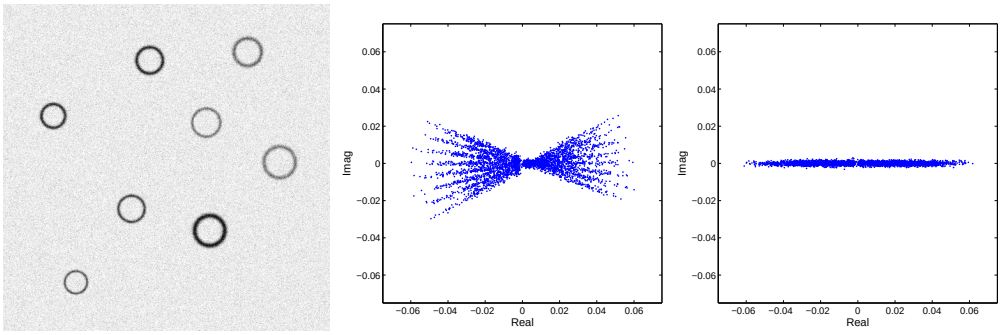
Artificial pollen images

Artificial pollen images resemble real pollen images (Fig. 4.14(a)) which have been used in a detection and identification task [77]. Multiresolution Gabor features were applied for the same task, but surprising problems in classification of the features were observed. A simplified test with artificially created images was conducted to eliminate the effect of various imperfections in images of real pollens. The variables for artificial pollen creation are their radius, the edge width and contrast, and added Gaussian noise (Fig. 4.13(a)).

Multiresolution Gabor features were computed at centers of pollens for 500 training images, each with 5-9 artificial pollens. The images, originally 1024×1024, have been downscaled by factor of 8, because the filters only use low frequencies and computing responses for a low resolution image is much faster. However, the downscaling introduces misalignment. An example of feature space for Gabor filters at frequency $f = 1/50, \theta = 0°$ is presented in Fig. 4.13(b). Variation in the direction of real axis was caused by variations in the radius. The variations along the imaginary axis (anti-symmetry) are very small because the center locations are known precisely and the objects are symmetric, but downscaling to $1/8th$ of the original size causes an imprecision to the center position. When a Gabor filter is not precisely in the center, the pollen is perceived as anti-symmetric, which leads to responses not being purely real. The eight different misalignment positions caused by downscaling can be seen in the response space (see Fig. 4.13(b)) as eight distinct "stripes". If downscaling is not used, leading to a drastic increase in filtering time, the stripes disappear because the center positions are now exactly correct (see Fig. 4.13(c)).

Real pollen images

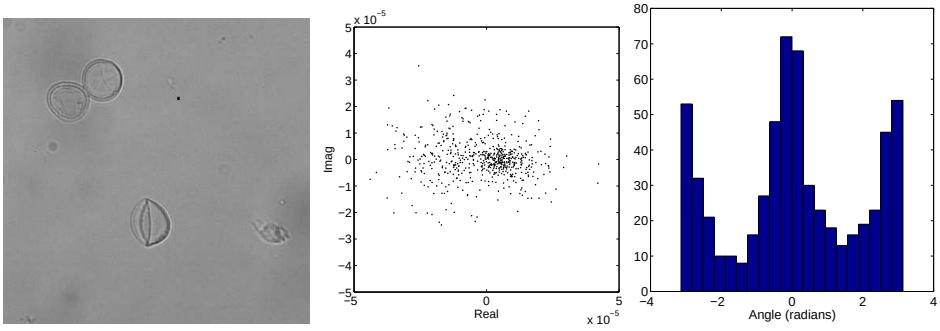
The next experiment involved real pollen images – an example can be seen in Fig. 4.14(a). The radiuses of real pollens varied from 35 to 80 pixels. The real pollens are far from



(a) (b) (c)

Figure 4.13: Artificial pollen experiment. (a) An artificial pollen image (pollens of radius $35 - 50$, varying edge widths and contrasts, and Gaussian noise). (b) Gabor responses from the centers of artificial pollens at frequency $f = 1/50$ and $\theta = 0°$ with downscaling to $1/8th$ size; (c) Without downscaling.

perfect circles and there were also three different types of pollens present in the images. Furthermore, the centers cannot be marked exactly, and thus, it is no surprise that Gabor filter responses do not form as symmetric pattern as with the artificial pollen experiment (Fig. 4.13(b)), but a smoother cloud (Fig. 4.14(b)). Still, most of the responses are located close to imaginary value 0 and vary in the direction of the real axis, where the variations are explained by varying radiuses. Phenomenon can be more easily seen when only angles of the complex responses are observed, see Fig. 4.14(c). Note that only a single feature is present in the figure; the whole feature space is more complex and less Gaussian because of the effects explained previously.

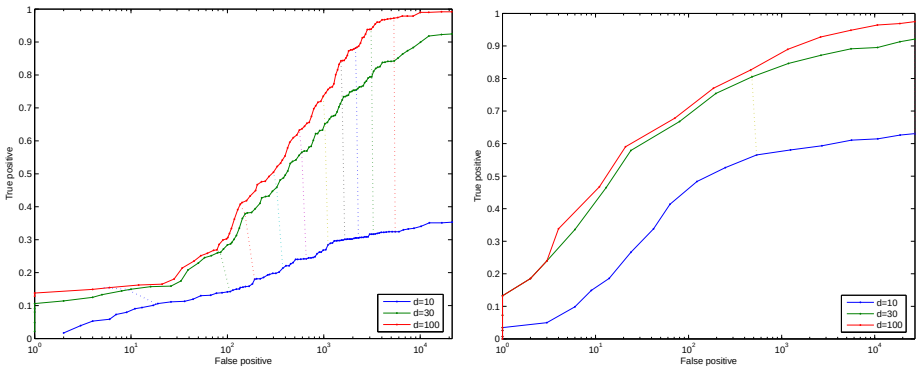


(a) (b) (c)

Figure 4.14: (a) A part of a real pollen image. (b) Gabor responses from the centers at frequency $f = \frac{1}{45\sqrt{2}}, \theta = 0°$; (c) Histogram of complex angles.

Pollen detection results for real pollen images in Fig. 4.15(a) (GMM) and in Fig. 4.15(b) (SVM) as ROC (Receiver Operating Characteristic) curves. There were 606 pollens in

the training set and 352 pollens in the testing set. Figures show results for different distances of location accuracy: in the best case the pollens would be found exactly without generating false positives. A distance of 30 is an acceptable accuracy as the minimum radius was 35. Results with the GMM classifier were bad as a huge number of false positives was found before a significant number (90%) of real pollens. At the point where 100 false positives were found, only 30% of the real pollens (approx. 100) were found. With SVM the result is clearly better at the same point, 65% (approx. 230).



(a) (b)

Figure 4.15: Detection of real pollens. (a) ROC curve for the GMM classifier; (b) ROC curve for the SVM classifier.

4.6 Summary

This chapter presented an image feature localization method based on multiresolution Gabor features and Gaussian mixture models. The method can use other local image descriptors or classifiers, and generic algorithms were presented where the individual parts can be changed at will, given they fulfill the requirements presented also in this chapter. The chapter continued with motivations for using one-class classifiers and usage of confidence with a Gaussian mixture model classifier. An alternative classifier, ν-SVM was presented as related to the problem, non-Gaussianity of Gabor features. In some cases the complexity of the feature space causes problems for GMM estimation given the limited training data (to learn very complex features spaces the estimation requires impractically large amount of training data) and the alternative SVM classifier proved to be able to learn complex feature spaces (from pollen images) more effectively. However, in the next chapter the experimental results shows that multiresolution Gabor features and GMM is still a powerful combination for accurately detecting image features.

Chapter V

Experiments and applications

In this chapter image feature detection and localization is demonstrated using the proposed combination of multiresolution Gabor feature and Gaussian mixture models using 3 different datasets in two challenging problems. Furthermore, to confirm validity of combination of Gabor features and GMM, the tests are repeated by replacing the descriptor and the classifier with other well-known methods which show no significant improvements. The method is further demonstrated in a categorization experiment where the results appear very natural. Finally, a similar approach is applied to a distinctly different subject as a side study, a fault detection problem with 1D signal.

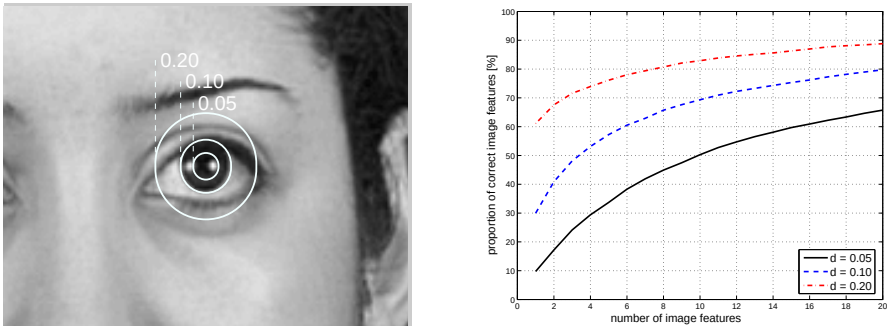
5.1 Accuracy measure

For detection of complete objects for example bounding boxes or ellipses are used, e.g., [62]. A "box" is drawn around the detected object and a union is taken between the detected box and manually marked the groundtruth box. The accuracy measure is then calculated as the area of the union divided by the area of the groundtruth box. This type of measure is problematic because it does not consider pose variations at all and the results vary greatly depending on how tightly or loosely the bounding boxes are drawn around the object. Overall, for measuring accuracy of detecting separate image features the bounding box model is not very suitable. The used accuracy measure, based on ranked order of image feature candidates and normalized error distance, is described next.

Before generating the result graph the image feature candidates are ranked in order, i.e., the most likely image candidate is first. The accuracy of local image feature detection is presented as a cumulative graph where the x-axis is the number of detected image features and the y-axis is the proportion of how often the correct image has been found among them. Localization accuracy is rarely pixel-perfect, so an image feature is deemed as correctly detected if it is within some pre-determined radius around the correct image feature position. For face detection tests the accuracy is measured by normalizing the distance between the eyes to $d_{eye} = 1.0$ and various accuracies are measured based on

the normalized distance (Fig. 5.1(a)). This type of measure is considered as the most appropriate for evaluating localization methods in [76].

An example of the accuracy measure graph is shown in Fig. 5.1(b). From the graph it can be seen, for example, that only every tenth of the highest ranked feature candidates was in the correct position within a distance of 0.05 (of d_{eye}). However, when the allowed distance was increased to 0.20 six of the first ten candidates were correct. Using the ten highest ranked image feature candidates the results improve so that the correct one was among them half of the time for a distance of 0.05 and over eight times out of ten for a distance of 0.20. A perfect result would be one where the first (highest ranked) image feature is always correct and the resulting graph would have a straight line at 100%. This is seldom the case and the graph stays below 100%. In general, if there is a large difference between graphs of different distances, like there is in the example graph (Fig. 5.1(b)), it means that the detection method cannot determine the correct location very exactly but is quite good at detecting it approximately. Another commonly seen variant is that the graphs of different distances are tightly bunched. In that case, the detection method can detect the correct location very accurately when it finds it at all.



(a) (b)

Figure 5.1: Measuring localization performance. (a) Demonstration of the used accuracy distance measure, d_{eye}; (b) An example result graph.

For detecting single image features this kind of accuracy measure is very natural.

5.2 Face detection

5.2.1 XM2VTS face database

The XM2VTS facial image database is a publicly available database for benchmarking face detection and recognition methods [58]. The frontal part of the database contains 600 training images and 560 test images of size 720×576 (width $\times$ height) pixels. The images are of excellent quality and the lightning conditions are stable, and therefore, face detection methods should perform very well with this database. To train the proposed image feature detectors a set of salient face regions were selected.

The selected image features should be discriminative: they can be reliably found in the images, and they can be used to distinguish the object category from other categories and backgrounds. Ten specific facial regions (see Fig. 5.2 for example images from the database with annotated image features) have been shown to have favorable properties to act as image features [30]. To capture visual information of local image patches around the marked locations local image descriptors are used.

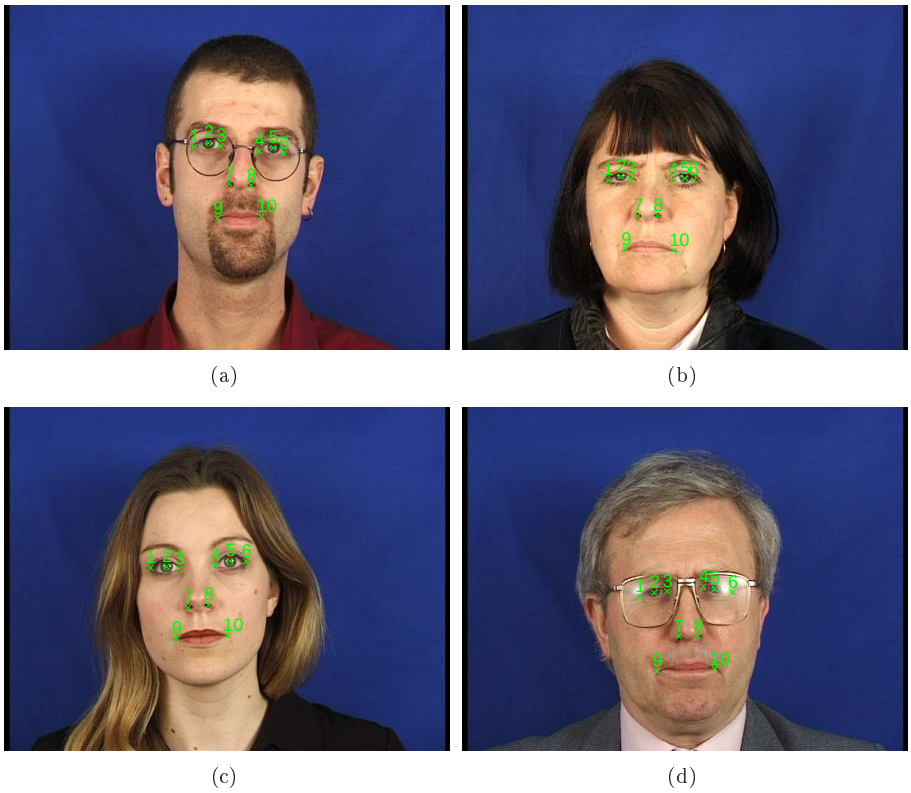


Figure 5.2: Training images with 10 manually marked and annotated image features.

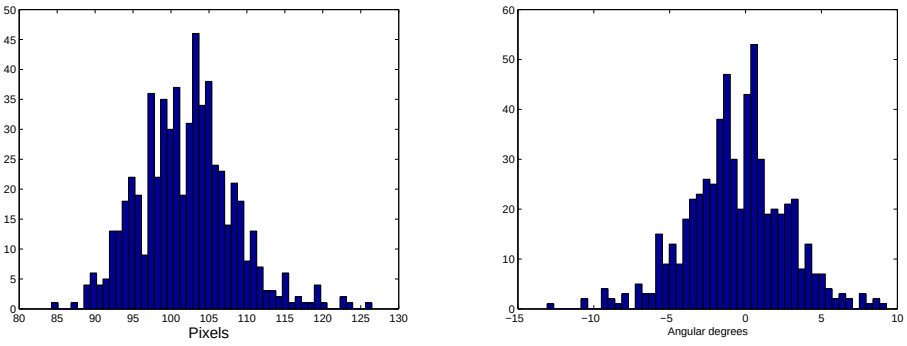
Three different image descriptors were used: multiresolution Gabor features (Section 3.1), local binary patterns (Section 2.3.3) and the steerable pyramid (Section 2.3.4). In addition the SIFT descriptor with and without PCA was tested, but the results were very weak and therefore not included. It was assumed that the high selectivity of SIFT features is not suitable to a task where generalization is needed and additionally their high dimensionality causes problems for the GMM classifier. Testing was limited to these descriptors because other presented descriptors are computationally too heavy for the exhaustive search; the number of points for which the descriptors are needed would have to be limited somehow, for example, by using interest point detectors. The main local descriptor has been multiresolution Gabor features, and their use is described here first.

Gabor feature parameter selection

The parameters of multiresolution Gabor features must be selected, either manually using some heuristics or by optimization, e.g., crossvalidation. Both approaches were applied in the experiments, so called "old parameters" have been selected manually, and are included as a comparison to results in a previously published article [43], and "tuned parameters" have been selected by crossvalidation. Heuristic selection is also explained to clarify properties of multiresolution Gabor features.

In the XM2VTS database all faces are in standard pose, everyone is looking at the direction of camera from the same distance. Naturally, there are still some variations in the distance between the eyes (Fig. 5.3(a)) and the angle between the eyes (Fig. 5.3(b)). The variations are small enough that invariant searches are not necessary and the filter bank parameters should be selected to cover the variations. For angular variations, which are limited approximately to $\pm 10°$, up to eight filter orientations can be used, $n \leq 8$, angular discrimination being then $22.5°$.

Filter frequency spacing, k, should be selected so that a single filter includes information from all scales present. The scale of objects here is presented approximately by the eye distance, the largest eye distances are around 120 pixels and the lowest around 90 pixels. Filters should include therefore scale variations in the order of $k \geq \frac{120}{90} \approx 1.33$. A slightly larger value can be used to assure that one filter covers suitable scales, and a natural choice is $k = \sqrt{2}$, i.e., half-octave scaling of frequencies. For the filter frequencies in the filter bank, defined by the selection of frequency of the highest filter, f_{high}, and the number of filter frequencies, m, no clear guidelines can be given. However, number of filter frequencies $m \geq 3$ should provide enough discriminative frequency information.



(a) (b)

Figure 5.3: Scale and orientation contents of XM2VTS training data computed using coordinates of the left and right eye centers: a) distribution of eye center distances; b) distribution of eye center rotation angles.

These heuristically selected "old parameters" were $n = 3$, $m = 4$, $k = \sqrt{2}$, $f_{high} = 1/30$. The parameters called "tuned parameters" were selected experimentally by a crossval-

idation procedure and were $n = 4$, $m = 6$, $k = \sqrt{3}$, $f_{high} = 1/40$. In both cases filter sharpness parameters γ and η have been selected using equations presented in Section 3.1.6. The tuned parameters differ from the old parameters in two ways: overall, the number of filters have been doubled from $3 * 4 = 12$ to $4 * 6 = 24$, and because of the lower f_{high}, larger filter spacing k and larger number of frequencies, much lower frequencies are included in the filter bank.

Training the GMM classifier

The classifier was trained with the Figueiredo-Jain method as presented in the Section 4.4. The multiresolution Gabor filter responses were computed for all 10 image feature locations in all images in the training set. The responses were then arranged as feature vectors which can be used with any one-class classifier (provided that they work with complex numbers). A classifier based on GMM has been used in these tests. During the evaluation Gabor filter responses were computed in all locations of the image and classified to each of the classes, and for each of the 10 classes, a number of the highest ranked feature locations were selected as potential image features.

Results for original images

After classification image features are processed in ranked order and an image feature was considered to be correctly classified if it was within a pre-set distance limit from the correct location. The distances are normalized to a distance between the eyes, $d_{eye} = 1.0$ (Fig.5.1(a)). The results for the XM2VTS database are presented in Fig. 5.4(a) for the old parameters and in Fig. 5.4(b) for the tuned parameters. With the old parameters using the tightest distance limit, 0.05, approximately 32% of the cases the highest ranked feature was the correct one. By increasing the distance limit to 0.10, which is still very good, the correct one was ranked highest in approx. 63% of the cases. By using the 10 highest ranked features from each class, the correct features were among them in 71% of the cases for the distance limit of 0.05 and 86% of the cases for the distance limit 0.10. Increasing the distance limit further to 0.20 leads to a small improvement. Similarly for the tuned parameters, the highest ranked feature was correct in 41% of the cases for a distance limit of 0.05, and 86% for the limit 0.10. With the 10 highest ranked features the results were 93% and 98% for distance limits 0.05 and 0.10 respectively.

It should be noted that the results with the tuned parameters are approaching the natural variation of the manual marking by different humans, meaning that the results cannot be significantly improved in this test.

The accuracy difference between the old and tuned parameters is demonstrated in Fig. 5.5. With old parameters the highest ranked features were spread all over the image to many false locations, while usually a few of them were found in the correct places. With the tuned parameters the highest ranked features were found compactly around the correct locations. The tuned parameters included lower frequencies and recognized the image feature locations based on a larger neighborhood, and therefore, did not lead to image feature candidates in false locations as easily.

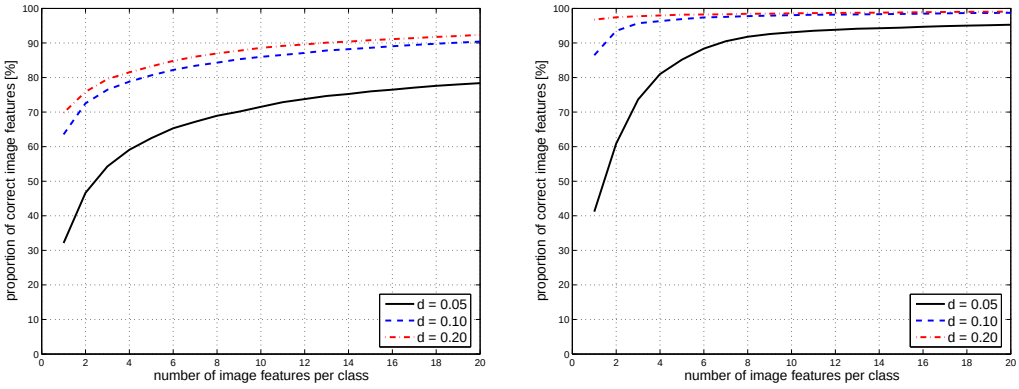


Figure 5.4: Accuracy of image feature extraction from XM2VTS test images: (a) Old parameters; (b) Tuned parameters.

Complex vs. magnitude responses

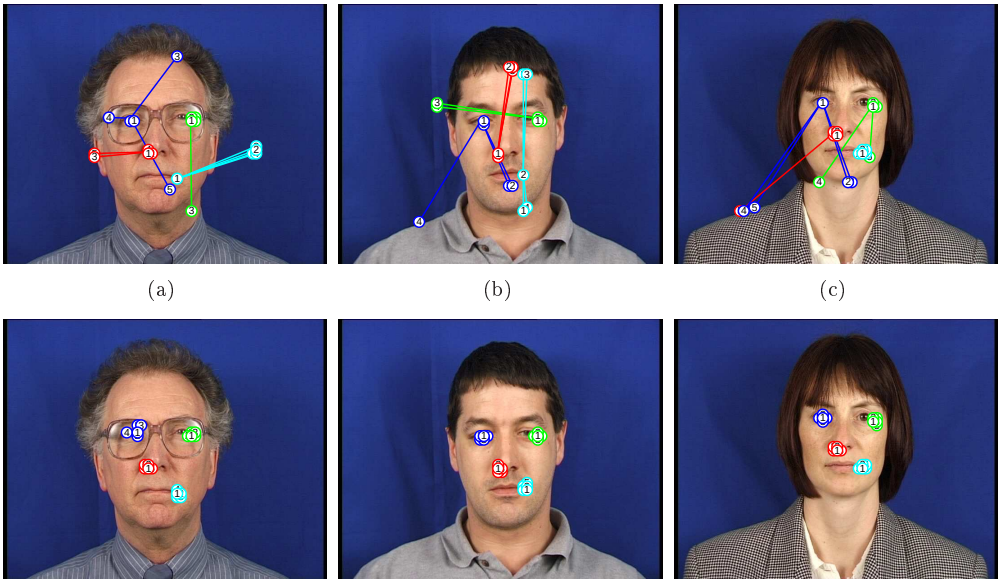
The multiresolution Gabor features described in Section 3 use naturally complex valued feature responses. Still, a majority of studies using Gabor features utilize only magnitude information instead of the complex representation. Using magnitude information is computationally easier and the results may be satisfactory even with only magnitude information. Previous experiments were performed by using only response magnitudes instead of complex numbers and the results are shown in Fig. 5.6, which are clearly not as good as the results with complex values (Fig. 5.4). The results demonstrate that removing the phase information, which is implicitly included in the complex values, leads to clear degradation of localization results.

The advantage of using complex numbers, which include implicitly both magnitude and phase information, instead of magnitude-only representation can be clearly seen in Fig. 5.7. The figure shows responses of a single filter for the left and right eye corners. In the complex plot the two classes are clearly separable, but completely overlap in the magnitude-only plot.

Results for artificially rotated and scaled images

A problem of the XM2VTS data set is that the images do not cover different scales or rotations as the faces are almost always near the standard pose. Invariance properties of the image feature localization cannot therefore be verified using the database. To be able to test the invariance properties, the evaluation images of the database were randomly rotated between ± 45 degrees and up-scaled factor between 1 and $\sqrt{2}$.

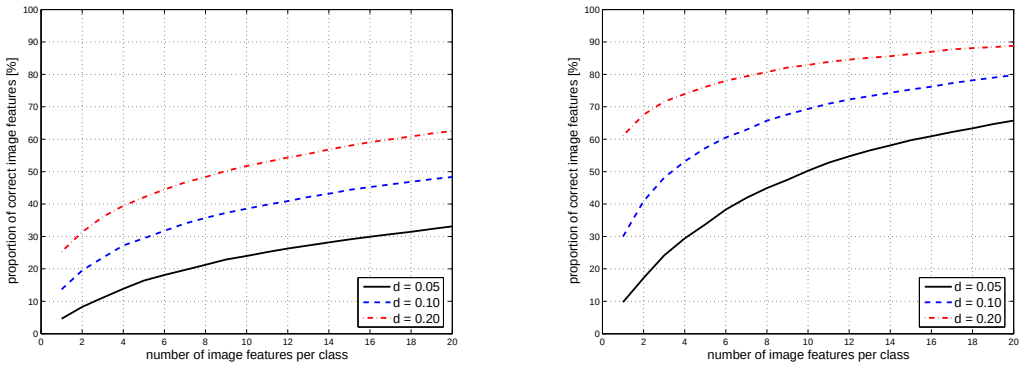
The image features were first searched for without using scale or rotation invariance manipulations. The results for old parameters are in Fig. 5.8(a) and for tuned parameters in Fig. 5.8(b). In the second phase, the detection was performed using one scale-shift and



(a) (b) (c)

(d) (e) (f)

Figure 5.5: Examples of extracted image features (left eye center: blue, right eye outer corner: green, left nostril: red, right mouth corner: cyan, 5 highest ranked features for each, numbered from 1 to 5): (a),(b),(c) Old parameters from [43]; (d),(e),(f) Tuned parameters.



(a) (b)

Figure 5.6: Accuracy of image feature extraction from XM2VTS test images (response magnitudes used instead of complex responses): (a) Old parameters; (b) Tuned parameters.

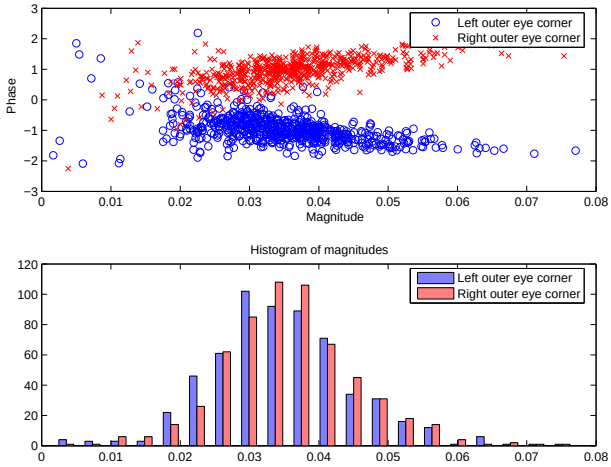


Figure 5.7: Scatter plots of Gabor filter responses for left and right eye corners.

two orientation shifts (± 1 step). For the old parameters this means that the scale-shift was $\sqrt{2}$ and orientation shifts $\pm 45°$, and for the tuned parameters the scale-shift was $\sqrt{3}$ and the orientation shifts $\pm 30°$. The results are presented in Fig. 5.8(c) for the old parameters and in Fig. 5.8(d) for the new parameters. Scale and rotation shifts gave clearly better results, and the difference is most noticeable using the tuned parameters as evident between Fig. 5.8(b) and Fig. 5.8(d).

5.2.2 Banca face database

In this experiment a significantly more challenging BANCA face database was used [2]. Only the English section of the database was used which includes 6240 test images of varying quality and background (see examples in Fig. 5.9). The training set consisted of XM2VTS and worldmodel images from English, Spanish, Italian and French BANCA sections, leading to 1600 as total number of training images.

A different data set required changing the parameters from the previous test with XM2VTS to get the best performance, the new "tuned" parameters were $n = 3$, $m = 6$, $k = \sqrt{3}$, $f_{high} = 1/25$. The differences to the settings used with XM2VTS are that higher frequencies are used (f_{high} has been increased from 1/40) and one fewer frequency is used. There are higher variation in the scales in the BANCA database and the filter bank must be tuned to the smallest scales, hence the higher frequencies. The number of filter frequencies must be decreased to prevent filter bank "seeing" too wide an area, including a cluttered background. The results are presented in Fig. 5.10, one scale-shift has been used and no rotation shifts. It is clear from the results that the BANCA database is considerably mode difficult than XM2VTS. At a distance 0.10 only 51% of the highest ranked features were correct (86% for XM2VTS), and with the 10 highest ranked features 68% (95% with XM2VTS). The spatial search may still succeed if at least three correct features are found.

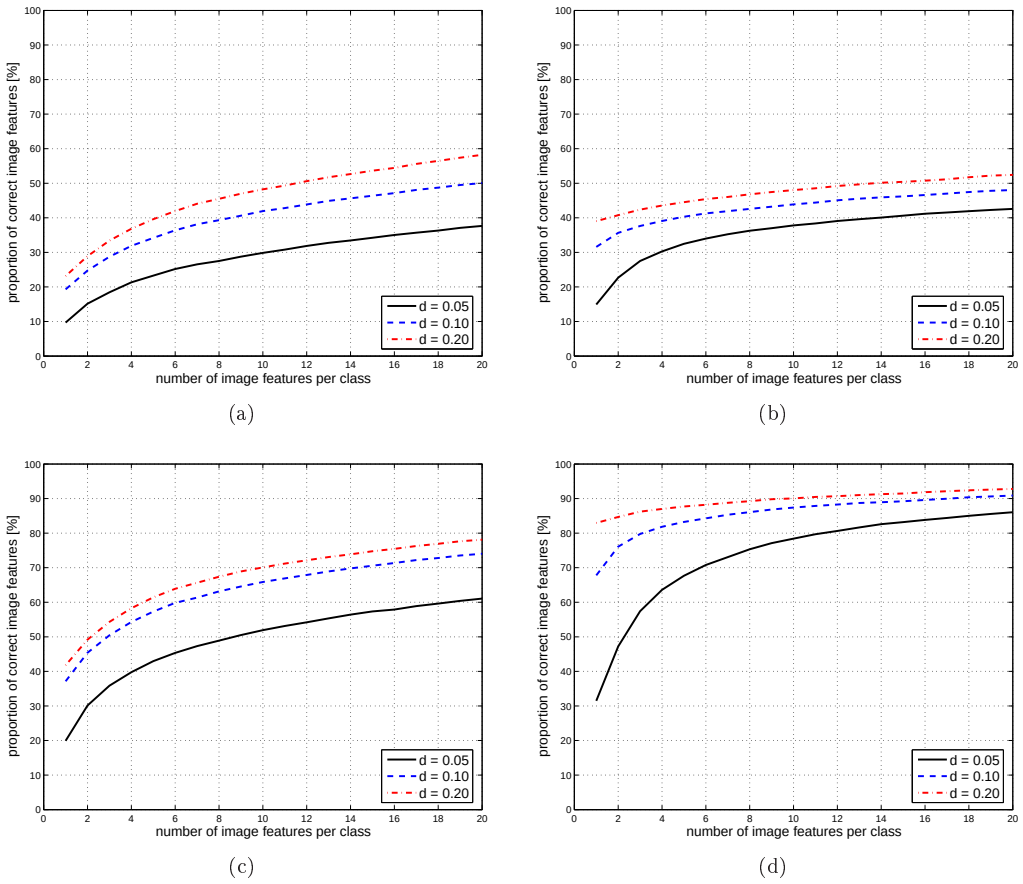


Figure 5.8: Accuracy of image feature extraction from artificially rotated and scaled images from XM2VTS test set; (a) old parameters – no invariance shifts; (b) tuned parameters – no invariance shifts; (c) old parameters – invariance shifts; (d) tuned parameters – invariance shifts.

(a) 1013_f_g1_s01_1013_en_4.ppm (b) 1013_f_g1_s12_1013_en_5.ppm

Figure 5.9: Example images from BANCA database.

5.2.3 Comparison to other local image features

Here the localization performance is tested with three different local image features are. All three image feature types were treated equally. Feature vectors were computed for all image points, and class conditional Gaussian mixture model probability densities were used to find the 100 best candidates. The probability densities were estimated using the training set.

The LBP features were computed as defined in Section 2.3.3. The image features were collected from 19×19 patches around the feature locations in the training set. The images were first downscaled by a factor of 1.5; the downscaling factor was determined manually to give the best localization results. The formed feature (containing several concatenated LBP histograms) is a real-valued vector of length 203. The images in the test set were again downscaled by a factor of 1.5 and the detection was performed as an exhaustive search in all 19×19 image areas. An example of training image with marked image patches is presented in Fig. 5.11.

For the steerable pyramid, Sec. 2.3.4, 4 levels and 6 orientations (5th order filter) were used (24 real values). The images were downscaled to half the size during both training and detection. Several settings were tested to find the parameters giving the best results.

Results of image feature localization of the 10 different face features using the different low-level features are shown in Fig. 5.12. The results for multiresolution Gabor features are repeated here because the testing method had to be changed slightly to treat all the features equally, namely with multiresolution Gabor features in the previous results, in Fig. 5.4, images were downscaled as aggressively as possible to maximize the detection speed. Now, downscaling was done at most to half a resolution to be comparable to the other image features, changing the results slightly. With all the image features, only the local maxima of the pdf were selected and values around them in a radius of 4 pixels were taken out. This improves the results because instead of several detections in the neighboring pixels there is now only one. With aggressive downscaling this is not needed.

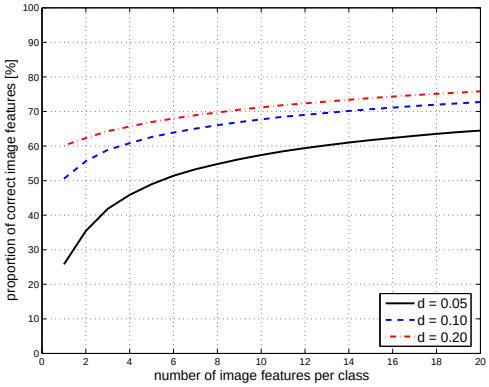


Figure 5.10: Accuracy of image feature extraction from the English section of BANCA database (only tuned parameters).

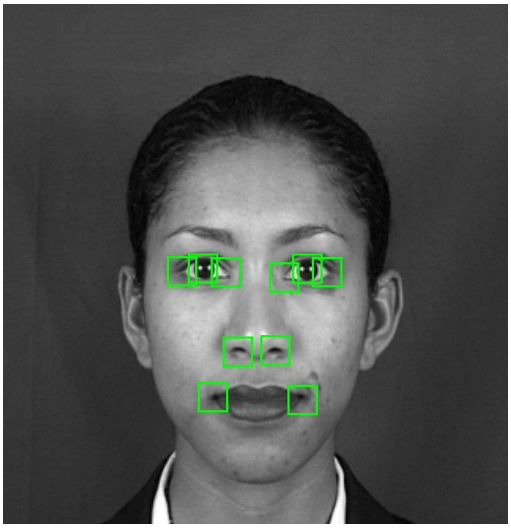
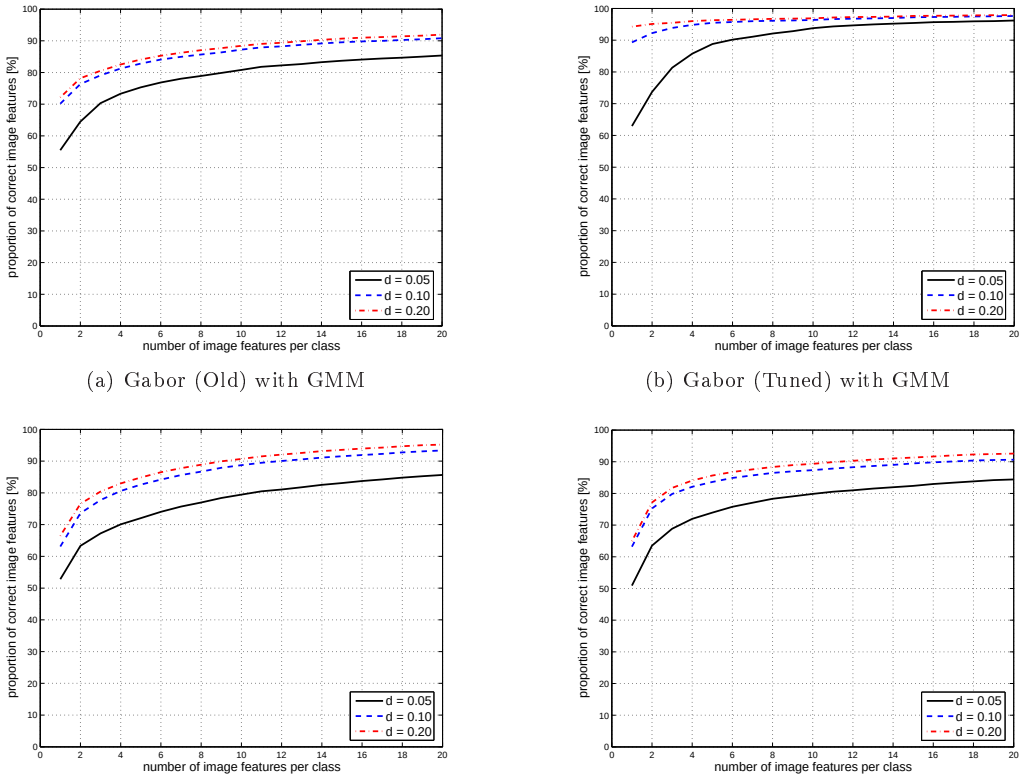


Figure 5.11: Example of 19×19 image patches used with LBP based localization method.

The results for LBP and the steerable pyramid features are similar to each other, steerable pyramid being slightly ahead and nearly equal to multiresolution Gabor features with the old parameters. The multiresolution Gabor features with tuned parameters provided clearly the best results. Better results with both LBP features could likely be achieved by tuning how the detection method is utilized or by changing the classifier as the Gaussian classifier may not be the best choice for the high dimensional LBP features. Unfortunately, tuning the steerable pyramid method is limited by octave spaced scaling levels. The LBP based method can be changed more freely as there are an unlimited number of combinations of different window sizes, how the window is broken into smaller partitions, and which LBP operators are used. However, tuning these settings is unintuitive – the effect to the classification result is difficult to estimate beforehand – and the features are already very long.



(a) Gabor (Old) with GMM

(b) Gabor (Tuned) with GMM

(c) Steerable pyramid with GMM

(d) LBP (tuned) with Gaussian

Figure 5.12: Image feature localization results for 10 different face features for the XM2VTS database with different local image features.

The computation times between these descriptors itself are similar. Like Gabor features, steerable pyramid features were computed in the frequency domain making their complexity essentially equal. However, steerable pyramid features can be computed slightly more efficiently in the spatial domain because the filters are small. LBP histograms are

fast to compute, but with these high dimensional features the classifier becomes much slower, making the LBP features slowest by a large margin of the tested descriptors.

5.2.4 Comparison to SVM classifier

The results with the original images of XM2VTS with the GMM classifier were also compared to results with the SVM classifier presented in Section 4.3. The results for SVM classifier are presented in Fig. 5.13. With the SVM classifier the complex numbers were converted to separate real and imaginary parts because the classifier implementation did not support complex values. An RBF kernel was used. Compared to results with GMM classifier (Fig. 5.4) the results are fairly close: with old parameters, the SVM classifier outperformed the GMM classifier for less accurate measurement distances 0.10 and 0.20, but was slightly worse for the most accurate distance 0.05. For the tuned parameters the results were practically the same for both classifiers.

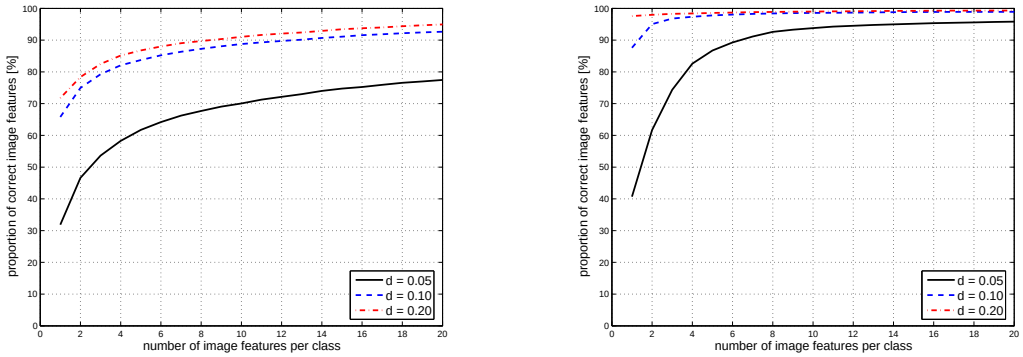


Figure 5.13: Accuracy of image feature localization from XM2VTS images with SVM classifier: (a) Old parameters; (b) tuned parameters.

The SVM classifier has few problems compared to the GMM classifier. First of all, it has more parameters than GMM classifier, which has only one parameter, selecting the number of mixture model components, or not even that with adaptive estimation algorithms like Figueiredo-Jain method.

The SVM classifier however needs at least the kernel type to be selected. The RBF kernels are often used and the size of the kernel must be selected based on the data: too wide kernels cannot learn true class boundaries, and too small easily overfit to the training data. With the normalized multiresolution Gabor features RBF kernel size $\gamma = 0.3$ was found to be a suitable value.

Another important parameter is the parameter controlling the number of outliers and support vectors. With the used SVM classifier this parameter is $\nu \in [0, 1]$, where ν denotes the lower bound for fraction of support vectors and upper bound for fraction of outliers. In these experiments $\nu = 0.1$ was used.

The most severe problem is the speed: the SVM classifier was considerably slower than GMM classifier. The speed depends on the number of support vectors. For the results presented here there were over 200 support vectors for each of the 10 classes and the detection phase was approximately 50 times slower with the SVM classifier than with the GMM classifier with similar detection accuracy. Some of the difference is likely to be explained by more optimal GMM classifier implementation, but the SVM classifier with RBF kernels remains computationally too heavy when there are many support vectors.

5.3 License plate detection

Description

The image feature localization was tested further with a commercial license plate database. The testing methods were similar to the main tests with XM2VTS and BANCA face databases presented in the previous section.

Data and methods

A commercial database was used due to a lack of license plate databases being available to the general public. The training set consisted of 157 images from randomly selected date and the landmark points were manually annotated for all of them. The annotated points were the four corners of license plates (Fig. 5.14). Multiresolution Gabor features were extracted from annotated locations in the training images and the GMM classifier was trained for each of them. In the testing Gabor features were computed for all points and the classifier used to select the highest ranked candidate points. Similarly to the face detection experiment the localization accuracy was normalized, this time by the average distance from a corner to the opposite corner of the license plate. The accuracy measure is illustrated in Fig. 5.14(c) as small circles in the upper left corner of the license plate.

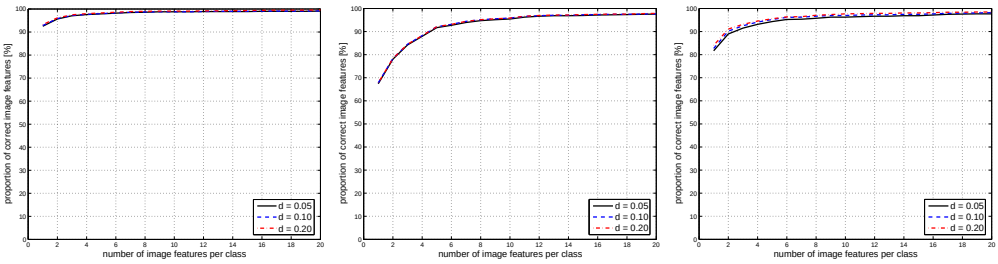
(a) (b) (c)

Figure 5.14: (a)-(b) Example images from license plate database with image feature positions, license plate corners, are marked with green circles; (c) Demonstration of accuracy measure for license plate localization measure (green circles in the upper left corner corresponding to distances 0.05, 0.1 and 0.2, and the large circle demonstrates distance 1.0).

The evaluation set consisted of 247 images from different randomly selected date. The results with multiresolution Gabor features are presented in Fig. 5.15(a). The tuned filter bank parameters were $m = 2$, $n = 4$, $k = \sqrt{3}$, $f_{high} = 1/24$. The corner points provided very easily recognizable image features and with the tuned parameters, which included only two frequencies and four orientations, the results were very good, 93% of the first ranked image features were correct at the distance 0.05. Examples of detected license plate features are shown in Fig. 5.16.

The tests were repeated with steerable pyramid and local binary pattern features and the results are shown in Fig. 5.16(b)-(c). Toth features were used similarly to the face detection experiment, but the settings giving best results were searched for this test separately. Gaussian mixture model classifier was used in all cases.

The steerable pyramid (Section 2.3.4) used 3 levels and 6 orientations (5th order filter) yielding to 18 real values. The images were downscaled to $\frac{1}{2.6}$th size during both training and detection. The LBP features were computed as in Section 2.3.3: the image features were collected from 19×19 patches around the feature locations in the training set. No downscaling was needed this time. The formed feature (containing several concatenated LBP histograms) is a real-valued vector of length 203. The detection was performed as exhaustive search over all 19×19 image areas.

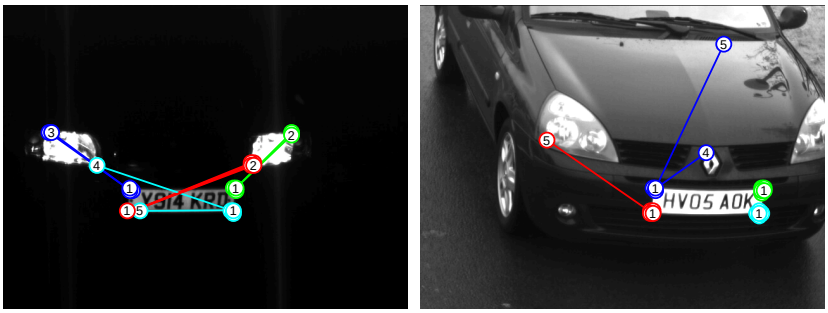


(a) (b) (c)

Figure 5.15: Accuracy of image feature (license plate corner) localization: (a) Multiresolution Gabor features; (b) Steerable pyramid; (c) LBP. With multiresolution Gabor features for $d = 0.05$, the accuracy reaches 93% with only one (highest rank) image feature extracted.

Conclusions

As a conclusion for this test, multiresolution Gabor features were able to provide very good results. The presentation power of the steerable pyramid appeared to be be considerably lower, it needed more filter frequencies and still the detection performance was poor. Increasing the number of filter frequencies, and therefore also the lowest frequencies, were not helpful in this test because the interesting object, license plate, is quite small and its neighborhood is not very helpful in detecting it. Local binary pattern features gave better results than the steerable pyramid, but were still behind multiresolution Gabor feature results. The results with LBP features could likely be improved further by changing the structure of the features, i.e., using other area size than 19×19



(a) (b)

Figure 5.16: Examples of extracted features with multiresolution Gabor image features (left upper corner: blue, right upper corner: green, left lower corner: red, right lower corner: cyan, 5 best features for each class numbered from 1 to 5): (a) night scene; (b) day scene.

and using different pattern of LBP histograms extracted from the area. However, tuning those settings is unintuitive and classifying long feature vectors would still be inefficient.

5.4 Visual object categorization using self-organization

Description

Visual object categorization was studied in [34]. The object categorization was performed using the same data used in face detection in Section 5.2.

The categorization was based on multiresolution Gabor features computed at annotated locations and the spatial relationships between annotated locations. Multiresolution Gabor features can accurately capture local image information and in addition, original information can be reconstructed (see Section 3.1.5). A complete object can be represented by combining several local desctiptors and their spatial locations. A spatial constellation model together with multiresolution Gabor features provides a basis for analyzing the visual appearance and its variation over any real objects. The self-organizing map provides a tool for unsupervised categorization of objects. The main goal was to study the proposed model in the context of automatic categorization and visualization, and investigation of model capability to explain visual similarities of natural objects, human faces.

Related research

Since the introduction of the basic self-organizing map (SOM) method by Kohonen [47] its characteristics have been under active research. Self-organization has been offered as a solution for explaining the organization of information processing in the brains. The same hypothesis could apply to visual information and its processing in the human visual

system. However, one of the main problems has been a lack of robust representation of visual appearance which could allow meaningful organization.

Due to the Gabor filter's correspondence to the human visual system and the SOM's ability to self-organize information hypothetically similarly to human brain, it is not a surprise that combining Gabor filters and SOM have been proposed before. Gabor filters in a multiresolution stack have been utilized to represent visual information and used with the SOM in several studies [93, 72], but they have treated spatial information very coarsely. A large amount of receptive field responses covering a whole image has been claimed to overcome the problem of poor spatial localization of each response. The problem however is severe, a small spatial change may appear as a large change in feature values (e.g., a misaligned face). Moreover, a degree of allowed spatial changes and local distortions cannot be restricted without a spatial constellation model.

Object categorization with the self-organization over visual appearance is the basis of the well-known PicSOM method [49], but the PicSOM also suffers from the same basic problem. The PicSOM utilizes global feature histograms making the method unable to account for spatial changes.

Suitability of Gabor based receptive field responses in categorization by self-organization have been demonstrated by Lampinen et al. [51]. They however discarded the phase information of (complex) Gabor filters which is important for local appearance, and their study considered the categorization only for local object parts, not complete objects.

Methods

The representation of local appearance is based on multiresolution Gabor features, Chapter 3, which are utilized similarly to the object detection and localization method presented in Chapter 4. Multiresolution Gabor features have an useful property, they can be used to reconstruct the original image.

Successful categorization requires information of both the local appearance variation and the global spatial variation of local parts. To form a proper input for the SOM the local appearance descriptions and their spatial constellation must be combined to a fused feature structure. A feature vector can be constructed by concatenating responses of local parts (the feature matrix in (3.8)) into a vector. The spatial information can be fused by simply adding the coordinates of the corresponding parts into the same vector. Spatial normalization is needed, i.e. a fixed origin must be defined, in order to prevent distortions due to the variation in location, scale and orientation (see Fig. 5.17). However, the scale variation can be natural in certain cases (e.g., large vs. small faces).

The basic SOM method is based on the Euclidean distance, and therefore, scaling of all variables should be similar, or otherwise variables with large values dominate the self-organization process. The problem can be solved by normalization. The Gabor responses and coordinates by themselves do not require normalization, but when they are concatenated into one feature vector their relative weights must be adjusted. The weighting depends on total combined magnitude of the local receptive field responses and the scaling of coordinates.

Results

The object categorization method was tested using frontal face images from the XM2VTS database (see Section 5.2 for further details). The method should categorize faces so that similar faces are near each other. This can be visualized in two ways: use of the raw neural weight information and the reconstruction property of Gabor features or find the closest matching faces.

Multiresolution Gabor features were computed using 5 different frequencies down from the highest frequency $f_{max} = \frac{1}{10}$ with a scaling factor $k = \sqrt{3}$, and 6 orientations forming a 6×5 feature matrix, (3.8). The matrices from 10 image points were concatenated to a single feature vector of 300 dimensions. In addition, locations of the 10 image features, a constellation model, were added to the vector resulting in a total of 320 dimensions representing visual appearance of each face. The location coordinates were normalized since faces can be in any location and pose. A straightforward method was used by normalizing the coordinates to a form where the middle point between eyes is located in the origin, $(0, 0)$, and the rest of the coordinates are scaled and rotated in order to set the eye centers to the coordinates $(-0.5, 0)$ and $(0.5, 0)$. An example of the normalized spatial configuration can be seen in Fig. 5.17.

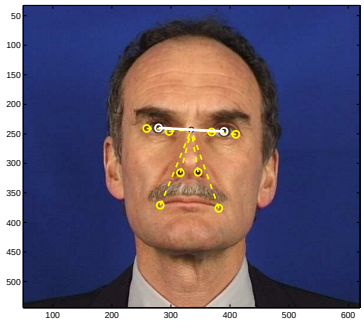


Figure 5.17: Example of the spatial model configuration normalized by the distance and angle between eye centers. Frontal face with 10 salient image features (left and right outer eye corners, left and right inner eye corners, left and right eye centers, left and right nostrils, and left and right mouth corners).

The SOM method was applied to feature vectors of all 600 training images. A rectangular SOM of size 17×13 was used. The unified distance matrix of the SOM is visualized in Fig. 5.18. Large values (light colors) in the distance matrix represent large changes between neighboring cells of the map. Respectively, the dark colors denote relatively similar values in the corresponding area of the map.

Due to the reconstruction property of the appearance model it was possible to visualize raw faces formed by the SOM. In Fig. 5.19(a) reconstructed faces from the SOM are shown; note that only every fourth cell is presented. In Fig. 5.19(b) are shown the closest matching faces from the XM2VTS database. The connection between the distance matrix in Fig. 5.18 and the both reconstructed and closest matching faces can be clearly seen: the distance matrix is dark (i.e., the changes are small) for the most part except for

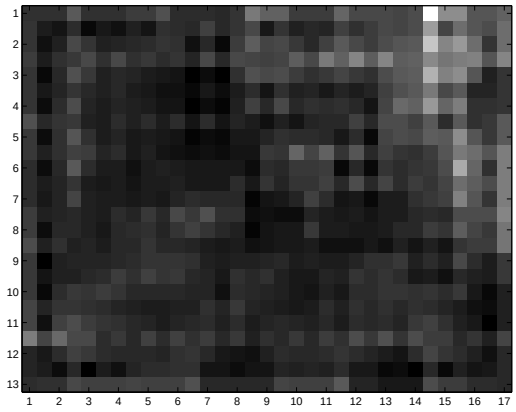


Figure 5.18: Unified distance matrix of 17×13 SOM. Bright shades denote large changes in map values.

the top right corner where the bearded men seem to belong. Overall, there is a clear trend that feminine faces are in the bottom left corner and there is a gradual change to masculine faces in the top right corner.

Next, the SOM was used to categorize individuals to similar visual appearance classes. The local image features and spatial constellations were calculated for all images in the XM2VTS test set and the best matching SOM units were searched. In Fig. 5.20 are shown faces belonging to the same best matching unit, i.e, faces having a similar visual appearance. Note that the examples may include the same person several times since the XM2VTS includes several images of each person. The formed categories can be easily interpreted: the category in Fig. 5.20(a) includes older men with eye-glasses and/or beards, and the category in Fig. 5.20(c) includes only women. However, the gender is not the discriminative factor in all cases as can be seen in Fig. 5.20(b), where the category includes both women and clean-shaven men with the common factor that none have eye-glasses.

Conclusions

This experiment addressed the problem of finding categorical similarity between visual appearance of real objects. The similarity enables automatic categorization of visual observations: formation of object groups via their natural self-organization. Multiresolution Gabor features were used to extract and represent local appearances, and the the local appearances were connected by a spatial constellation model. A fused representation was formed which combines both appearance at a local level and global spatial variation to a single feature structure. The representational power of the proposed structure was investigated by performing a self-organization with the self-organizing map method. From the experimental results on face images it can be seen that the structure encapsulates the visual appearance leading to a natural self-organization of visually similar objects.

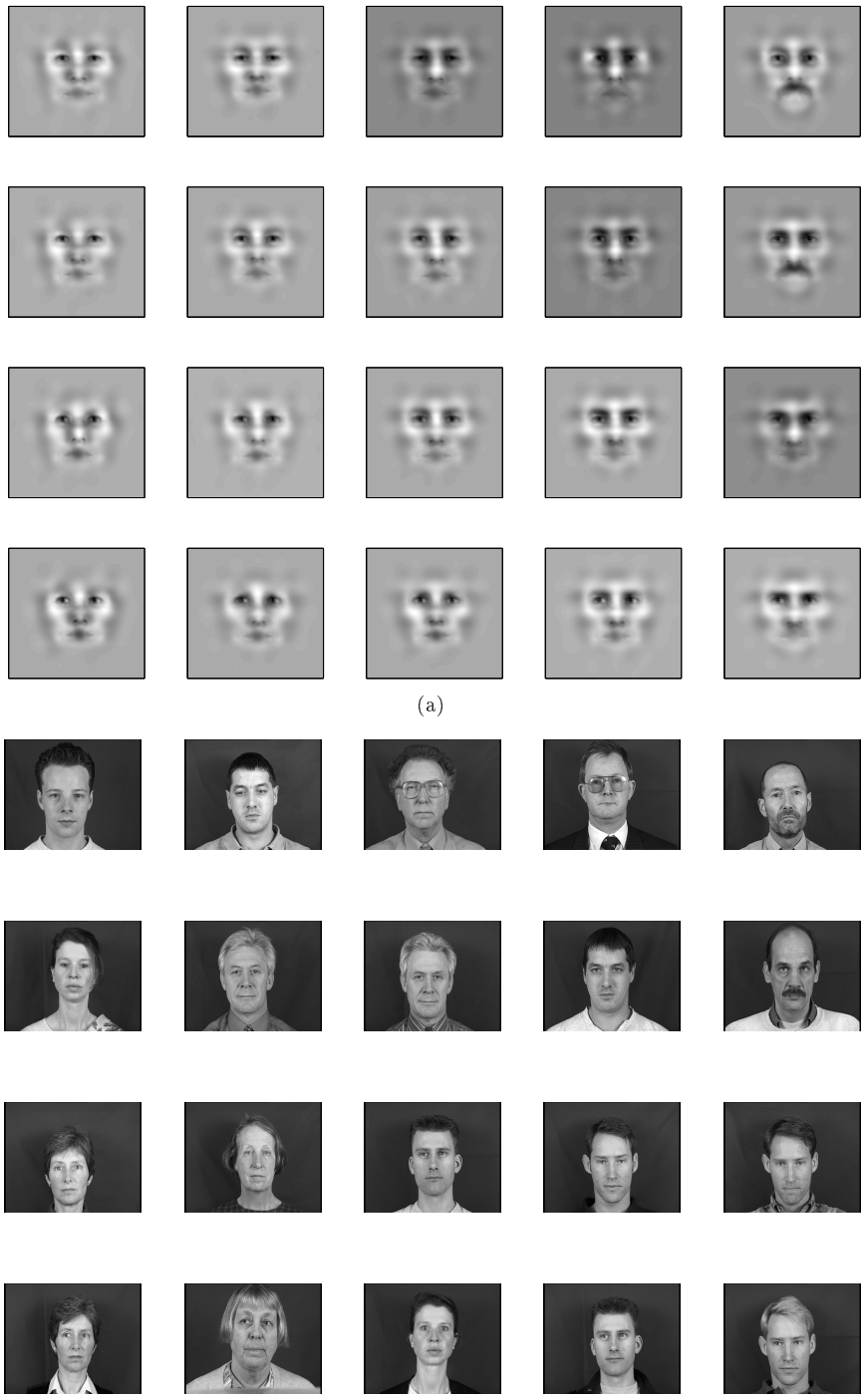


Figure 5.19: (a) Reconstructed raw visual appearances from cells of the 17×13 SOM (only every fourth face is shown); (b) The closest matching faces from the database.

(a)

(b)

(c)

Figure 5.20: Examples of unsupervisedly found visual appearance categories (face classes). (a) Older men with eye-glasses or beards; (b) Women and clean-shaven men without eye-glasses; (c) Women.

5.5 Fault detection in electrical motors

Description

Automatic fault detection was studied in [37]. This study was not directly related to the more general topic of image features, but multiresolution Gabor features were used with 1D signals for fault detection in electric motors. Two-classes of signals, normal and damaged, were used during training and new signals were to be classified in these classes. Measurements were very noisy and damage was visible only at some frequencies. To ease the work of the classifier, the Gabor filters with maximal separation between normal and damaged classes were first searched.

Automatic condition monitoring and diagnosis are important in industrial installations where a high degree of automation is desired. Automatic monitoring can be used to detect and recognize system faults, such as motor failures, where an early warning could prevent escalation of the problem. This is the case for example in motor bearing damage detection [78, 91].

A diagnosis method was proposed to find discriminative regions, bands, from frequency content of the two classes of signals (normal/damaged) and to classify new measurements to these classes. The proposed method is useful in cases where there are measurements, but the physical characteristics of failures are not known. A sufficient amount of measurements from the both normal and damaged classes are needed in order to find the most discriminative features, but the case where there are measurements mainly from the normal class is of special importance. In practice, measurements from failure conditions cannot be comprehensive because measuring signals from various failure modes is too expensive to realize.

Methods

Two sets of signals, $x_k(t)$ and $y_k(t)$, represent examples from two classes, C_1 and C_2, respectively. The sub-index k denotes a measurement number, $k = 0, 1, \ldots, N_1 - 1$ for C_1 and $k = 0, 1, \ldots, N_2 - 1$ for C_2. It is assumed that the signals are measured during a stationary system mode, i.e., system parameters such as rolling speed and load are constant. Now, the discriminative information should be present at some frequency band and it is sufficient to apply a band-pass filter $\psi(t)$. In a stationary system mode the time information can be ignored and a global feature, such as a power spectrum can be utilized. The selection of the best features is reduced to finding the optimal values for the central frequency f and bandwidth γ of a band-pass filter. The normalized Gabor filter (3.1) was used as the band-pass filter.

If there are several frequency bands where the contents of the classes C_1 and C_2 are dissimilar, then the band where the separation of the classes is most evident should be selected. The first-order statistics approach is not sufficient since it simply selects the frequency band where the distance between the expectations is largest, but neglects the variance information, and a significant overlap of the class probabilities may exist [54, 45].

It was assumed that the features are extracted from signals measured during a constant operation mode where variance in the measurements is supposed to be caused by a

large number of unknown independent sources. The form of the probability distributions is therefore assumed to be Gaussian and the classes can be uniquely defined by their expectations, μ_x and μ_y, and variances, σ_x^2 and σ_y^2. For Gaussians Fisher's discriminant ratio (FDR) can be used to measure the distance between the distributions [70]

$$FDR(p_x(n), p_y(n)) = \frac{(\mu_x - \mu_y)^2}{\sigma_x^2 + \sigma_y^2} \quad . \tag{5.1}$$

Using the divergence measure in (5.1) the discriminative energy function can be defined as

$$E = \frac{1}{2}\left(\frac{(\mu_x - \mu_y)^2}{\sigma_x^2 + \sigma_y^2}\right)^2 \quad . \tag{5.2}$$

Using the discriminative energy function (5.2) the frequency f and bandwidth γ of the band-pass filter in (3.1) can be optimized. Single or several frequencies can be selected. For the two classes GMMs (Section 4.2) are estimated and then Bayesian classification is used [84]. However, the quality or number of failure measurements is not usually sufficient or does not cover all failure states. In that case the classification should be based only on the probability distribution of normal condition measurements. Therefore, one-class classification with only the GMM of the normal class was used.

Results

Induction motor bearing damages

Induction motors have been a widely studied subject of condition monitoring [91, 5]. An important sub-category of induction motor failures are bearing damages, which can be detected from vibration, acoustic noise, temperature, or stator current signals. Bearing damages are attractive for evaluating the proposed method since characteristic frequencies of damage appearance can be analytically solved and compared to automatically found frequencies.

Bearing damage detection based on the stator current

The stator current data consisted of stator current signals measured from motors in a normal condition (C_1) and motors with bearing damage (C_2). The measurements contain two cases: no load connected to motors and with a full load. In these experiments the classification was performed using the Bayesian classifier, which requires examples from both classes, and using one-class classification with the confidence based limit (Section 4.2.2), when failure measurements are not needed. The pdf limit was calculated from the normal class training data so that the whole training set was accepted, and the pdf values lower than that were classified as a failure.

For motors with no load discriminative energy E and classification results are presented in Fig. 5.21. The discriminative energy had its maximum near the first harmonic (202 Hz) of the characteristic frequency (101 Hz). Also both classification schemes had the maximal accuracy at the same frequency band.

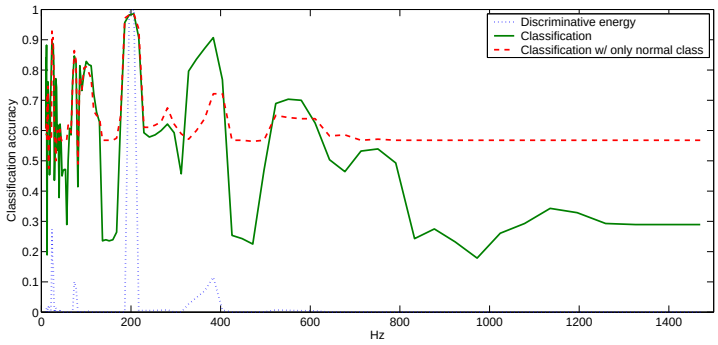


Figure 5.21: Discriminative energy and classification accuracies for motors with no load.

For motors with a full load, results are shown in Fig. 5.22. This was a more difficult situation since the full load caused various disturbances, but still, the characteristic frequency (101 Hz) and some of its harmonics contained discriminative information. Classifications succeeded at the same frequencies, but due to disturbances the accuracy decreased.

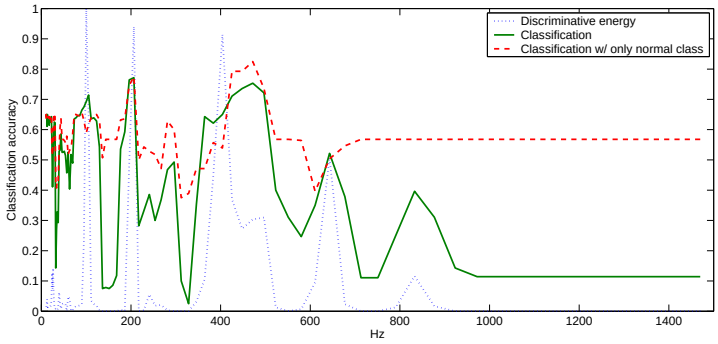


Figure 5.22: Discriminative energy and classification accuracies for motors with full load.

Using both the Bayesian classifier and the one-class classifier the same classification accuracy was achieved at the most discriminative frequencies.

The test was repeated for the full load dataset using six of the most discriminative frequencies and Gaussian mixture models to estimate class pdfs. The results for both the Bayesian classifier and one-class classifier are presented in Fig. 5.23 as an ROC (receiver operating characteristic) curve for different confidence levels. Only normal condition measurements were used to form a pdf and confidence was used to decide between normal and failure conditions. From the curve it can be seen that by decreasing the confidence more normal condition measurements were correctly identified (true positives), but also an increasing number of failure conditions were considered as normal (false positives).

The optimal trade-off depends on the application. On the other hand, there was only a minor difference comparing the results where also the failure condition pdf was used in Bayesian classification (a priories were estimated from the training set, which does not correspond to real situations).

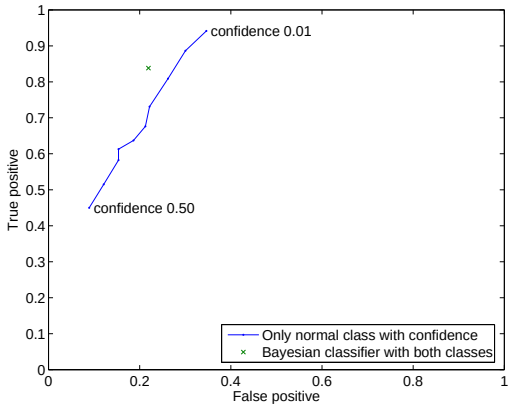


Figure 5.23: Receiver operating characteristic (ROC) curve for using confidence for classification of motor failures.

Comparative results

The experiments were repeated by utilizing the analytically calculated characteristic frequencies as reported by Yazici and Kliman in [91]. For the characteristic frequencies and our approach the results are shown in Table 5.1. Classification was done using the Bayesian classification. The three most discriminative frequencies were used for classification both separately and combined, and the results were compared to the classification results using the characteristic frequencies. Characteristic frequencies provided an accuracy of 97.5% correct classification for a motor with no load. The most discriminative frequency of E provided the same accuracy, but 100% accuracy was achieved with a combination of the three most discriminative frequencies. For a motor with a full load classification with the characteristic frequencies provided an accuracy of only 66.8% while the three most discriminative frequencies provided a slightly better classification result, 72.1%.

Utilizing the most discriminative frequencies of E a better accuracy was achieved than with the computed characteristic frequencies as used in the literature. It seems that some of the harmonics include noise which harms the classification.

Conclusions

Here, automatic motor condition diagnosis was studied: methods on how to automatically select the most discriminative features and how to classify new signals were investigated. In addition to this, the case where the amount of failure condition measurements

Table 5.1: Classification results using calculated characteristic frequencies and the three most discriminative frequencies of E.

	No load		Full load	
	Freq.	Correct	Freq.	Correct
Char. freq.		97.5%		66.8%
1st peak	206.3 Hz	97.5%	100.1 Hz	69.3%
2nd peak	23.6 Hz	87.9%	206.3 Hz	77.9%
3rd peak	383.6 Hz	91.4%	403.9 Hz	68.6%
Combined		100.0%		72.1%

is not sufficient was considered and one-class classification was used. The classification succeeded using the discriminative discriminative frequencies with both Bayesian classification with two classes and with one-class classification using only the normal class.

5.6 Summary

This chapter presented the experiments. First was the main experiment, face detection, with two different datasets. The experiment was about localization of landmark positions in the face, eye-centers and corners, nostrils and sides of the mouth. The experiment was repeated with two different filter bank settings of multiresolution Gabor features, local binary patterns and steerable pyramid filters. As a classifier, the Gaussian mixture model based one-class classifier and the ν-SVM one-class classifier were used. Results were good and the combination of the multiresolution Gabor features and GMM classifier gave the best results in nearly all tests. Another similar experiment was localization of the corners of license plates and the results were very good for the proposed method. Two different but related applications were also presented. The first was about visual categorization of objects using self-organization and the second about fault detection in electical motors.

Chapter VI

Discussion

The objective of this thesis was to study and develop local image features usable in an object detection and localization method. Many of the currently popular methods are semi-supervised, they require only labeled training images to learn object class. However, semi-supervised methods cannot guarantee good localization performance, and therefore a supervised approach was the main interest in this thesis. The current methods are often based on separate interest point detection and local image description steps, and they both can be considered separately. Complete object detection methods combining interest point detection and local image description were introduced and briefly experimented.

The method presented in this thesis is based on an approach which combines the interest point detection and local description into one step, complete image feature detection. We proposed the combination of multiresolution Gabor features and a one-class classifier based on Gaussian mixture models (GMM). The method can be trained with manually annotated landmark positions. The local image feature detection method was tested in two main applications: face detection and license plate detection. Face detection provided excellent results with XM2VTS image database, and good results with a much more challenging BANCA database. For license plate detection a commercial database was used, and the results were almost perfect.

One of the main problems has been the low computational efficiency of the multiresolution Gabor features, and therefore a major objective and contribution was to study efficient implementation of multiresolution Gabor filtering. Improving the efficiency also provided better results, since tuning the parameters became feasible in the experiments.

Other possible local description methods were also tested as an alternative for multiresolution Gabor features. The limiting factor for many descriptors remains computational heaviness: with this image feature localization method exhaustive search is used and the descriptor should be quick to compute and to classify. Usually the descriptors are used after interest point detectors, in which case the computational complexity is of less importance. Two alternatives were tested, steerable pyramid and local binary pattern based

features. The flexibility of multiresolution Gabor features provided better localization results in both face and license plate detection tests.

An alternative to a GMM classifier was tested. The limitation here is that a one-class classifier is used to be able to omit the background class, the class representing everything else but the searched image features, and one-class classifiers are not as completely studied field as normal two-class classifiers. A support vector machine (SVM) based one-class classifier was used as an alternative, as one of the main problems of the GMM classifier can be its ability to represent occasionally very complex distributions of multiresolution Gabor features, given limited training data. The SVM classifier was able to surpass the results of the GMM classifier in some of the tests, but with the price of being slower and the difference in classification performance was not dramatic.

The data used for supervised object detection and localization experiments was also used for object categorization. Local image description and the spatial relationships between marked landmark positions was used to categorize face images to visually similar clusters using the self-organizing map. The categorization method was able to create natural categorization of similar faces.

Gabor features and one-class classification were also applied to a completely different application area, fault detection in electric motors. In this application a failure in an electric motor must be noticed based on measurements of stator current. There are various failure modes and they all cannot be reliably measured and included in the training data, and therefore one-class classification is useful. Tests were performed with data gathered from motors in normal condition and with a bearing failure. The classification results were good and in accordance with the theoretical results.

Overall, the proposed image feature detection and localization method performed very well. However, this thesis did not include one important part of a complete object detection method, the spatial model combining detected local image features, only the performance of local image feature detection was studied. The requirement of manually marked landmark positions in the training data is also a severe constraint for a generic object detection method. This requirement is not easily removed without giving up the main benefit, ability to exactly locate objects by using high quality local image features learned from manually marked landmark positions. Still, by combining our proposed image feature method with a spatial constellation model the localization accuracy challenges that of the current state-of-the-art methods.

[1] Agarwal, S., and Roth, D. Learning a sparse representation for object detection. In *Proceedings of the European Conference on Computer Vision* (2002), pp. 113–130.

[2] Bailly-Bailliere, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Marietho z, J., Matas, J., Messer, K., Popovici, V., Poree, F., Ruiz, B., and Thiran, J.-P. The BANCA database and evaluation protocol. In *Proceecings of the International Conference on Audio- and Video-Based Biometric Person Authentication* (2003), pp. 625–638.

[3] Ballard, D., and Wixson, L. Object recognition using steerable filters at multiple scales. In *Proceedings of IEEE Workshop on Qualitative Vision* (1993), pp. 2–10.

[4] Beardsley, P., Torr, P., and Zisserman, A. 3d model acquisition from extended image sequences. In *Proceedings of the 4th European Conference on Computer Vision* (1996), vol. 2, pp. 683–695.

[5] Benbouzid, M., Vieira, M., and Theys, C. Induction motors' faults detection and localization using stator current advanced signal processing techniques. *IEEE Transactions on Power Electronics 14*, 1 (1999), 14–22.

[6] Bernardino, A., and Santos-Victor, J. A real-time Gabor primal sketch for visual attention. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis* (Estoril, Portugal, 2005).

[7] Bodnarova, A., Bennamoun, M., and Latham, S. Optimal Gabor filters for textile flaw detection. *Pattern Recognition 35* (2002), 2973–2991.

[8] Braithwaite, R., and Bhanu, B. Hierarchical Gabor filters for object detection in infrared images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1994), pp. 628–631.

[9] Burt, P., and Adelson, E. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4 (1983), 532–540.

[10] Carbonetto, P., Dorko, G., and Schmid, C. Bayesian learning for weakly supervised object classification. Tech. Rep. 427, INRIA Rhône-Alpes, Grenoble, France, 2004.

[11] Chang, S.-L., Chen, L.-S., Chung, Y.-C., and Chen, S.-W. Automatic license plate recognition. *IEEE Transactions on Intelligent Transportation Systems 5*, 1 (2004), 42–53.

[12] Cheng, H., Zheng, N., and Sun, C. Boosted Gabor features applied to vehicle detection. In *Proceedings of the 18th International Conference on Pattern Recognition* (2006), vol. 1.

[13] Daugman, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A 2*, 7 (1985), 1160–1169.

[14] Daugman, J. G. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE 94*, 11 (2006), 1927–1935.

[15] Dorgo, G., and Schmid, C. Selection of scale-invariant parts for object class recognition. In *Proceedings of the 9th IEEE International Conference on Computer Vision* (2003), vol. 1, pp. 634–639.

[16] Everitt, B., and Hand, D. *Finite Mixture Distributions*. Monographs on Applied Probability and Statistics. Chapman and Hall, 1981.

[17] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 4 (2006), 594–611.

[18] Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2003), vol. 2, pp. 264–271.

[19] Fergus, R., Perona, P., and Zisserman, A. A sparse object category model for efficient learning and exhaustive recognition. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), vol. 1, pp. 380–387.

[20] Figueiredo, M., and Jain, A. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 3 (Mar 2002), 381–396.

[21] Fischler, M., and Elschlager, R. The representation and matching of pictorial structures. *IEEE Transactions on Computers C-22*, 1 (1973), 67–92.

[22] Freund, Y., and Schapire, R. E. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 1 (1997), 119–139.

[23] Gabor, D. Theory of communication. *Journal of Institution of Electrical Engineers 93* (1946), 429–457.

[24] Goodman, N. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *The Annals of Mathematical Statistics 34*, 1 (March 1963), 152–177.

[25] Granlund, G. H. In search of a general picture processing operator. *Computer Graphics and Image Processing 8* (1978), 155–173.

[26] Hadid, A., Pietikäinen, M., and Ahonen, T. A discriminative feature space for detecting and recognizing faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 797–804.

[27] Hamouz, M. *Feature-based affine-invariant detection and localization of faces.* PhD thesis, University of Surrey, 2004.

[28] Hamouz, M., Kittler, J., Kamarainen, J.-K., and Kälviäinen, H. Hypotheses-driven affine invariant localization of faces in verification systems. In *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication* (2003), pp. 276–284.

[29] Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., and Kälviäinen, H. Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model. In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition* (2004), pp. 67–72.

[30] Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., Kälviäinen, H., and Matas, J. Feature-based affine-invariant detection and localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 9 (2005), 1480–1495.

[31] Harris, C., and Stephens, M. A combined corner and edge detector. In *Proceedings of The 4th Alvey Vision Conference* (1988), pp. 147–151.

[32] Hsu, R.-L., Abdel-Mottaleb, M., and Jain, A. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 696–706.

[33] Hyndman, R. J. Computing and graphing highest density regions. *The American Statistician 50*, 2 (May 1996), 120–126.

[34] Ilonen, J., and Kamarainen, J.-K. Object categorization using self-organization over visual appearance. In *Proceecings of the International Joint Conference on Neural Networks* (Vancouver, Canada, July 2006), pp. 4549–4553.

[35] Ilonen, J., Kamarainen, J.-K., and Kälviäinen, H. Efficient computation of Gabor features. Research report 100, Department of Information Technology, Lappeenranta University of Technology, 2005.

[36] Ilonen, J., Kamarainen, J.-K., and Kälviäinen, H. Fast extraction of multi-resolution gabor features. In *Proceedings of the International Conference on Image Analysis Processing* (Modena, Italy, September 2007), pp. 481–486.

[37] Ilonen, J., Kamarainen, J.-K., Lindh, T., Ahola, J., Kälviäinen, H., and Partanen, J. Diagnosis tool for motor condition monitoring. *IEEE Transcations on Industry Applications 41*, 4 (2005), 963–971.

[38] Ilonen, J., Kamarainen, J.-K., Paalanen, P., Hamouz, M., Kittler, J., and Kälviäinen, H. Image feature localization by multiple hypothesis testing of gabor features. *IEEE Transactions in Image processing* (accepted with minor changes).

[39] Ilonen, J., Paalanen, P., Kamarainen, J.-K., and Kälviäinen, H. Gaussian mixture pdf in one-class classification: computing and utilizing confidence values. In *Proceedings of International Conference on Pattern Recognition* (Hong Kong, August 2006).

[40] Jain, A., Chen, Y., and Demirkus, M. Pores and ridges: High-resolution fingerprint matching using level 3 features. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (2007), 15–27.

[41] Jurie, F., and Schmid, C. Scale-invariant shape features for recognition of object categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2004), vol. 2, pp. 90–96.

[42] Kadir, T., Zisserman, A., and Brady, M. An affine invariant salient region detector. In *Proceedings of the European Conference on Computer Vision* (2004), pp. 228–241.

[43] Kamarainen, J.-K., Ilonen, J., Paalanen, P., Hamouz, H., Kälviäinen, H., and Kittler, J. Object evidence extraction using simple Gabor features and statistical ranking. In *Proceedings of the 14th Scandinavian Conference of Image Analysis* (Joensuu, Finland, 2005), pp. 119–129.

[44] Kamarainen, J.-K., Kyrki, V., and Kälviäinen, H. Invariance properties of Gabor filter based features - overview and applications. *IEEE Transactions on Image Processing 15*, 5 (2006), 1088–1099.

[45] Kamarainen, J.-K., Kyrki, V., Lindh, T., Ahola, J., and Partanen, J. Statistical signal discrimination for condition diagnosis. In *Proceedings of the Finnish Signal Processing Symposium* (Tampere, Finland, 2003), pp. 195–198.

[46] Ke, Y., and Sukthankar, R. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of the Computer Vision and Pattern Recognition* (2004), vol. 2, pp. 506–513.

[47] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43* (1982), 59–69.

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

[49] Laaksonen, J., Koskela, M., Laakso, S., and Oja, E. PicSOM - content based image retrieval with self-organizing maps. *Pattern Recognition Letters 21*, 13–14 (2000), 1199–1207.

[50] Lades, M., Vorbrüggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R., and Konen, W. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers 42*, 3 (1993), 300–311.

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

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

[53] Lindeberg, T. Feature detection with automatic scale selection. *International Journal of Computer Vision 30*, 2 (1998), 79–116.

[54] Lindh, T., Ahola, J., Kamarainen, J.-K., Kyrki, V., and Partanen, J. Bearing damage detection based on statistical discrimination of stator current. In *Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives* (Atlanta, Georgia, USA, 2003), pp. 177–181.

[55] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[56] Ma, J., and Fu, S. On the correct convergence of the EM algorithm for Gaussian mixtures. *Pattern Recognition 38* (2005), 2602–2611.

[57] Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide-baseline stereo from maximally stable regions. In *Proceedings of the British Machine Vision Conference* (2002), pp. 384–393.

[58] Messer, K., Matas, J., Kittler, J., Luettin, J., and Maitre, G. XM2VTSDB: The extended M2VTS database. In *Proceedings of the 2nd International Conference on Audio and Video-based Biometric Person Authentication* (1999), pp. 72–77.

[59] Messer et al., K. Face authentication test on the BANCA database. In *Proceedings of the 17th International Conference on Pattern Recognition* (2004), vol. 4, pp. 523–532.

[60] Mikolajczyk, K., Leibe, B., and Schiele, B. Multiple object class detection with a generative model. In *Proceedings of the Computer Vision and Pattern Recognition* (2006), vol. 1, pp. 26–36.

[61] Mikolajczyk, K., and Schmid, C. Scale & affine invariant interest point detectors. *International Journal of Computer Vision 60*, 1 (2004), 63–86.

[62] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 10 (2005), 1615–1630.

[63] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. A comparison of affine region detectors. *International Journal of Computer Vision 65*, 1 (2005), 43–72.

[64] Mohan, A., Papageorgiou, C., and Poggio, T. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 4 (2001), 349–361.

[65] Nestares, O., Navarro, R., Portilla, J., and Tabernero, A. Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions. *Journal of Electronic Imaging 7*, 1 (1998), 166–173.

[66] Ojala, T., Pietikainen, M., and Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24* (2002), 971–987.

[67] Opelt, A., Pinz, A., Fussenegger, M., and Auer, P. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 3 (2006), 416–431.

[68] Paalanen, P., Kamarainen, J.-K., Ilonen, J., and Kälviäinen, H. Feature representation and discrimination based on Gaussian mixture model probability densities - practices and algorithms. *Pattern Recognition 39*, 7 (2006).

[69] Park, H. J., and Yang, H. S. Invariant object detection based on evidence accumulation and Gabor features. *Pattern Recognition Letters 22* (2001), 869–882.

[70] Peterson, D., and Mattson, R. A method of finding linear discriminant functions for a class of performance criteria. *IEEE Transactions on Information Theory 12*, 3 (1966), 380–387.

[71] Pötzsch, M., Maurer, T., Wiskott, L., and Malsburg, C. Reconstruction from graphs labeled with responses of Gabor filters. In *Proceedings of the ICANN 1996* (1996).

[72] Pujol, A., Wechsler, H., and Villanueva, J. Learning and caricaturing the face space using self-organization and hebbian learning for face processing. In *Proceedings of the International Conference on Image Analysis and Processing* (2001), pp. 273–278.

[73] Raghu, P., and Yegnanarayana, B. Segmentation of Gabor-filtered textures using deterministic relaxation. *IEEE Transactions on Image Processing 5*, 12 (1996), 1625–1636.

[74] Ranganathan, N., Mehrotra, R., and Namuduri, K. An architecture to implement multiresolution. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (1991), vol. 2, pp. 1157–1160.

[75] Reed, I., Gagliardi, R., and Stotts, L. Optical moving target detection with 3-d matched filtering. *IEEE Transactions on Aerospace and Electronic Systems 24*, 4 (1988), 327–336.

[76] Rodriguez, Y., Cardinaux, F., Bengio, S., and Mariéthoz, J. Measuring the performance of face localization systems. *Image and Vision Computing 24* (2006), 882–893.

[77] Rodriguez-Damian, M., Cernadas, E., Formella, A., and Gonzales, A. Automatic identification and classification of pollen of the urticaceae family. In *Proceedings of Advanced Concepts for Intelligent Vision Systems* (2003), pp. 38–45.

[78] Schoen, R., Habetler, T., Kamran, F., and Bartfield, R. Motor bearing damage detection using stator current monitoring. *IEEE Transactions on Industry Applications 31*, 6 (1995), 1274–1279.

[79] Schölkopf, B., and Smola, A. *Learning with kernels.* MIT Press, 2002.

[80] Shah, S., and Aggarwal, J. A bayesian segmentation framework for textured visual images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1997), pp. 1014–1020.

[81] Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D. Shiftable multiscale transforms. *IEEE Transactions on Information Theory 38*, 2 (1992), 587–607.

[82] Tarassenko, L., Hayton, P., Cerneaz, N., and Brady, M. Novelty detection for the identification of masses in mammograms. In *Proceedings of the 4th International Confenrence on Artificial Neural Networks* (1995), pp. 442–447.

[83] Tax, D., and Duin, R. Support vector data description. *Machine Learning 54*, 1 (2004), 45–66.

[84] Theodoridis, S., and Koutroumbas, K. *Pattern Recognition.* Academic Press, 1999. ISBN 0-12-686140-4.

[85] Tong, Y. *The Multivariate Normal Distribution.* Springer Series in Statistics. Springer-Verlag, 1990.

[86] Tsai, D.-M., Wu, S.-K., and Chen, M.-C. Optimal Gabor filter design for texture segmentation using stochastic optimization. *Image and Vision Computing 19*, 5 (2001), 299–316.

[87] Verbeek, J. J., Vlassis, N., and Kröse, B. Efficient greedy learning of Gaussian mixture models. *Neural Computation 5*, 2 (Feb 2003), 469–485.

[88] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001), vol. 1, pp. 511–518.

[89] Weber, M., Welling, M., and Perona, P. Unsupervised learning of models for recognition. In *Proceedings of European Conference on Computer Vision* (2000), vol. 1, pp. 18–32.

[90] Weldon, T., and Higgins, W. Integrated approach to texture segmentation using multiple gabor filters. In *Proceedings of the International Conference on Image Processing* (1996), vol. 3, pp. 955–958.

[91] Yazici, B., and Kliman, G. An adaptive statistical time-frequency method for detection of broken bars and bearing faults in motors using stator current. *IEEE Transactions on Industry Applications 35* (1999), 442–452.

[92] Young, I., van Vliet, L., and van Ginkel, M. Recursive Gabor filtering. *IEEE Transactions on Signal Processing 50*, 11 (2002), 2798–2805.

[93] Zhi, Y., and Ming, G. A SOM-wavelet networks for face identification. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (2005).

I Analytical solutions for filter spacing formulas

The parameter equations for filter frequency spacing and filter orientation spacing were presented in Table 3.1 and Table 3.2. How these equations were solved is presented here. Analytical solutions to filter spacing formulas are for a multiresolution Gabor filter bank using parameters shown in Table 1. Note that while p is used for filter overlap in both filter frequency and filter orientation formulas, the value does not have to be the same.

Table 1: Parameters of a multiresolution Gabor filter bank.

Parameter	Description
p	Crossing point between adjacent filters
k	Scaling factor for filter frequencies
γ	Filter sharpness along the major axis
m	Number of filters at different frequencies
f_{min}	Tuning frequency of the lowest frequency filter
f_{max}	Tuning frequency of the highest frequency filter
η	Filter sharpness along the minor axis
n	Number of filters in different orientations

Filter frequencies

Using equation for 1D Gabor filter in frequency domain, (3.2), a point u_a can be solved where the value of the equation is p. Consecutive filters cross in a place where both of their values are equal to p (see Fig. 1).

$$\begin{aligned} \Psi(u) &= e^{-\left(\frac{\gamma\pi}{f_0}\right)^2 (u_a - f_0)^2} = p \\ &\Rightarrow \quad u_a = f_0 \left(1 \pm \frac{1}{\gamma\pi}\sqrt{-\ln p}\right), \end{aligned} \tag{1}$$

which corresponds to two adjacent filters at frequencies f_0 and f_0/k. Therefore,

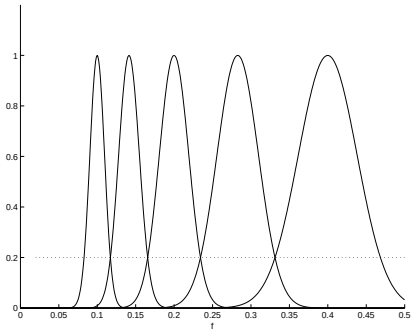


Figure 1: Fixed frequency factor $k = \sqrt{2}$ and overlap value $p = 0.2$.

$$\begin{aligned} f_0\left(1 - \frac{1}{\gamma\pi}\sqrt{-\ln p}\right) &= \frac{f_0}{k}\left(1 + \frac{1}{\gamma\pi}\sqrt{-\ln p}\right) \\ \Rightarrow \quad k &= \frac{1 + \frac{1}{\gamma\pi}\sqrt{-\ln p}}{1 - \frac{1}{\gamma\pi}\sqrt{-\ln p}}. \end{aligned} \tag{2}$$

On the other hand k and p, filter frequency scaling factor and crossing point between adjacent filters, are specified, γ can be solved from (2):

$$\gamma = \frac{1}{\pi}\left(\frac{k+1}{k-1}\right)\sqrt{-\ln p}. \tag{3}$$

Also p can be solved from (2) when γ and k are known as

$$p = e^{-\left(\gamma\pi\frac{k-1}{k+1}\right)^2}. \tag{4}$$

Additionally, we might want to solve k when $f_0 = f_{max}$, $f_{m-1} = f_{min}$ and m are given:

$$\begin{aligned} f_{min} &= \frac{1}{k^{m-1}} f_{max} \\ \Rightarrow \quad k &= e^{-\frac{\ln f_{min} - \ln f_{max}}{m-1}}. \end{aligned} \tag{5}$$

Also an indicative value for m can be solved from (5) based on f_{max}, f_{min} and k,

$$m = -\frac{\ln f_{min} - \ln f_{max}}{\ln k} + 1. \tag{6}$$

The exact value returned by the equation is not usable directly because m is an integer.

Filter orientations

The minor axis sharpness of a 2D Gabor filter, η, can be calculated based on the number of orientations and required overlap. In Fig. 2 a diagram of two Gabor filters in the frequency space is shown. Note that these filter overlap equations are approximations. To get an accurate overlap value, 2D equation of Gabor filters should be used as the whole elliptical filter envelope affects the overlap. However, the overlap equations would be then more complex as both filter bandwidth values, η and γ, are needed. Therefore, as the difference between results of accurate and approximate equations is not large in general, these approximate equations considering only minor axis bandwidth η are used.

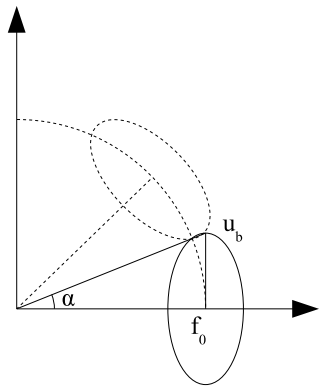


Figure 2: Two Gabor filters with different orientations in the frequency space.

Solving η is based on (3.2) with a crossing point p between two filters in adjacent orientations:

$$\begin{aligned} \Psi(u) &= e^{-\left(\frac{\eta\pi}{f_0}\right)^2 u_b^2} = p \\ &\rightarrow \eta = \frac{f_0}{\pi}\frac{\sqrt{-\ln p}}{u_b} \end{aligned} \tag{7}$$

Now, u_b can be solved from $u_b = \tan\left(\frac{\pi}{2n}\right) f_0$, where n is the number of filter orientations. However, this creates needlessly wide filters when the number of filter orientations is small, $n < 4$. Another possibility is to use an approximation for u_b by dividing the circumference of a circle by a number of filters, $u_b = \frac{\pi f_0}{2n}$. Therefore, η can be solved to either

$$\eta = \frac{1}{\pi}\frac{\sqrt{-\ln p}}{\tan\left(\frac{\pi}{2n}\right)} \quad \text{or} \quad \eta = \frac{1}{\pi}\frac{\sqrt{-\ln p}}{\frac{\pi}{2n}}. \tag{8}$$

When the number of orientations, n, is large, η calculated by both of the equations approaches the same value, but with a small n, the first solution for u_b leads to needlessly

wide filters, so the latter equation is preferred. With an approximate for u_b p can be solved from (7) as

$$p_2 = e^{-\left(\frac{\eta\pi^2}{2n}\right)^2}. \tag{9}$$

Additionally an indicative value for n can be solved based on p and η,

$$n = \sqrt{-\frac{(\eta\pi^2)^2}{4\ln p_2}}. \tag{10}$$

The actual value must be an integer.

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

236. DUFVA, KARI. Development of finite elements for large deformation analysis of multibody systems. 2006. Diss.

237. RITVANEN, JOUNI. Experimental insights into deformation dynamics and intermittency in rapid granular shear flows. 2006. Diss.

238. KERKKÄNEN, KIMMO. Dynamic analysis of belt-drives using the absolute nodal coordinate formulation. 2006. Diss.

239. ELFVENGREN, KALLE. Group support system for managing the front end of innovation: case applications in business-to-business enterprises. 2006. Diss.

240. IKONEN, LEENA. Distance transforms on gray-level surfaces. 2006. Diss.

241. TENHUNEN, JARKKO. Johdon laskentatoimi kärkiyritysverkostoissa. Soveltamismahdollisuudet ja yritysten tarpeet. 2006. Diss.

242. KEMPPINEN, JUKKA. Digitaaliongelma. Kirjoitus oikeudesta ja ympäristöstä. 2006.

243. PÖLLÄNEN, KATI. Monitoring of crystallization processes by using infrared spectroscopy and multivariate methods. 2006. Diss.

244. AARNIO, TEIJA. Challenges in packaging waste management: A case study in the fast food industry. 2006. Diss.

245. PANAPANAAN, VIRGILIO M. Exploration of the social dimension of corporate responsibility in a welfare state. 2006. Diss.

246. HEINOLA, JANNE-MATTI. Relative permittivity and loss tangent measurements of PWB materials using ring resonator structures. 2006. Diss.

247. SALMELA, NINA. Washing and dewatering of different starches in pressure filters. 2006. Diss.

248. SISSONEN, HELI. Information sharing in R&D collaboration – context-dependency and means of governance. 2006. Diss.

249. PURANEN, JUSSI. Induction motor versus permanent magnet synchronous motor in motion control applications: a comparative study. 2006. Diss.

250. PERÄLÄ, KARI. Kassanhallintakäytännöt Suomen kunnissa. 2006. Diss.

251. POUTIAINEN, ILKKA. A modified structural stress method for fatigue assessment of welded structures. 2006. Diss.

252. LIHAVAINEN, VELI-MATTI. A novel approach for assessing the fatigue strength of ultrasonic impact treated welded structures. 2006. Diss.

253. TANG, JIN. Computational analysis and optimization of real gas flow in small centrifugal compressors. 2006. Diss.

254. VEHVILÄINEN, JUHA. Procurement in project implementation. 2006. Diss.

255. MIROLA, TUULI. Impacts of the European integration and the European Union membership on Finnish export industries – Perceptions of export business managers. 2006. Diss.

256. RAUMA, KIMMO. FPGA-based control design for power electronic applications. 2006. Diss.

257. HIRVONEN, MARKUS. On the analysis and control of a linear synchronous servomotor with a flexible load. 2006. Diss.

258. LIU, JUNHONG. On the differential evolution algorithm and its application to training radial basis function networks. 2006. Diss.

259. LAITINEN, RISTO. Development of LC-MS and extraction methods for the analyses of AKD, ASA, and rosin sizes in paper products. 2006. Diss.

260. KUISMA, PETRI. Seinärakenteen infrapunakontrastin pienentäminen käyttäen ilmajäähdytystä ja säteilysuojausta. 2007. Diss.

261. ELLONEN, HANNA-KAISA. Exploring the strategic impact of technological change – studies on the role of Internet in magazine publishing. 2007. Diss.

262. SOININEN, AURA. Patents in the information and communications technology sector – development trends, problem areas and pressures for change. 2007. Diss.

263. MATTILA, MERITA. Value processing in organizations – individual perceptions in three case companies. 2007. Diss.

264. VARTIAINEN, JARKKO. Measuring irregularities and surface defects from printed patterns. 2007. Diss.

265. VIRKKI-HATAKKA. TERHI. Novel tools for changing chemical engineering practice. 2007. Diss.

266. SEKKI, ANTTI. Successful new venturing process implemented by the founding entrepreneur: A case of Finnish sawmill industry. 2007. Diss.

267. TURKAMA, PETRA. Maximizing benefits in information technology outsourcing. 2007. Diss.

268. BUTYLINA, SVETLANA. Effect of physico-chemical conditions and operating parameters on flux and retention of different components in ultrafiltration and nanofiltration fractionation of sweet whey. 2007. Diss.

269. YOUSEFI, HASSAN. On modelling, system identification and control of servo-systems with a flexible load. 2007. Diss.

270. QU, HAIYAN. Towards desired crystalline product properties: In-situ monitoring of batch crystallization. 2007. Diss.

271. JUSSILA, IIRO. Omistajuus asiakasomisteisissa osuuskunnissa. 2007. Diss.

272. 5th Workshop on Applications of Wireless Communications. Edited by Jouni Ikonen, Matti Juutilainen and Jari Porras. 2007.

273. 11th NOLAMP Conference in Laser Processing of Materials Lappeenranta, August 20-22, 2007. Ed. by Veli Kujanpää and Antti Salminen. 2007.

274. 3rd JOIN Conference Lappeenranta, August 21-24, 2007. International Conference on Total Welding Management in Industrial Applications. Ed. by Jukka Martikainen. 2007.

275. SOUKKA, RISTO. Applying the principles of life cycle assessment and costing in process modeling to examine profit-making capability. 2007. Diss.

276. TAIPALE, OSSI. Observations on software testing practice. 2007. Diss.

277. SAKSA, JUHA-MATTI. Organisaatiokenttä vai paikallisyhteisö: OP-ryhmän strategiat institutionaalisten ja kilpailullisten paineiden ristitulessa. 2007. Diss.

278. NEDEOGLO, NATALIA. Investigation of interaction between native and impurity defects in ZnSe. 2007. Diss.

279. KÄRKKÄINEN, ANTTI. Dynamic simulations of rotors during drop on retainer bearings. 2007. Diss.