CATEGORIZATION BY LOCAL INFORMATION USING SUPPORT VECTOR MACHINES

Diplomarbeit in Computer Science

submitted
by

Mario Joachim Fritz

born 16.1.1978 in Adenau

Written at

Lehrstuhl für Mustererkennung (Informatik 5)
Institut für Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg.

in Cooperation with

Computer Vision and Active Perception Laboratory
Department of Numerical Data Analysis
Royal Institute of Technology
Stockholm, Sweden

Advisor: B. Caputo, Prof. Dr.-Ing. H.Niemann

Started: 1.10.2003

Finished: 1.4.2004

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien des Lehrstuhls für Studien- und Diplomarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Erlangen, den 1.April 2004

Übersicht

Die Objekterkennung ist seit vielen Jahren Gegenstand der Forschung im Bereich Rechnersehen und dabei wurden beeindruckende Ergebnisse erzielt. Jedoch stellt die Aufgabe der Objektkategorisierung immer noch eine große Herausforderung dar. Um dieses anspruchsvolle Problem anzugehen, wird in dieser Arbeit ein kürzlich vorgestellter Ansatz benützt, der lokale Merkmalsrepräsentationen mit Support Vector Machines verbindet. Neben der Fähigkeit zur Generalisierung im Bezug auf die Anzahl der Kategorien und der Anzahl von Beispielen für die einzelnen Kategorien, wird der Ansatz auch unter realen Bedingungen getestet. Typische Probleme die von solchen realen Bedingungen herrühren sind heterogener Hintegrund, partielle Verdeckung und Skalierung. Alle drei Probleme werden in dieser Arbeit angegangen, und verschiedene Methoden werden untersucht, um diese Herausforderungen auf einer Menge nicht trivialer Kategorien zu bewältigen. Darüber hinaus werden Experimente mit Kategorien aus einem Büroumfeld durchgeführt. Die meisten Ansätze, die Erkennungsaufgaben auf der Basis von lokalen Merkmalen durchführen, beruhen auf einem Schritt zum Merkmalsabgleich. Verbesserungen konnten durch die Verwendung von Nebenbedingungen für den Merkmalsabgleich erzielt werden. Deshalb werden zwei neue Methoden zur Durchführung eines Abgleichs mit Nebenbedingung vorgeschlagen und gezeigt, wie diese im Kontext von Support Vector Machines angewendet werden können.

Abstract

Object recognition has been the subject of computer vision research for many years and impressive results have been achieved. However the task of object categorization is still challenging. In this thesis, a recently introduced approach which combines local feature representations with Support Vector Machines is used to tackle this challenging problem. Besides the capabilities to generalize with respect to the number of categories and the number of examples of each category, the approach is tested under real-world conditions. Typical problems arising from these real-world conditions are background clutter, partial occlusion and changes in scale. All three problems are adressed in this thesis and different methods are investigated to cope with these challenges on a set of non-trivial categories. In addition to this, experiments on categories in an office environment are performed. Most approaches that perform recognition tasks based on local features rely on a feature matching step. Improvements have been achieved by using constraints for the matching. Therefore two new methods for performing a constraint matching are proposed and it is shown how they can be applied in the context of SVMs.

Acknowledgements

I would like to thank Prof. Dr.-Ing. Heinrich Niemann and Prof. Jan-Olof Eklundh for making this thesis and my exchange with the KTH in Stockholm possible. In particular, I am indebted to my supervisor Barbara Caputo for her remarkable and dedicated support and guidance of my work and Eric Hayman for an inspiring cooperation and all the help he gave to me. Furthermore, I am grateful to all the people in the computer vision and robotics group at the KTH, who made my stay as pleasant and rewarding as possible by creating a comfortable and stimulating working environment.

Besides this, Christian Wallraven generously supported me in my work with the local kernel and Ivan Laptev and Tony Lindeberg gave helpful advices on scale-space issues. I also want to thank Christoph Gräßl, Oliver Sander, Dominik Stöffel and Amir Akbarzadeh who gave me additional comments on parts of the thesis.

Contents

Chapter 1

Introduction

One of the main topics in computer vision is the development of systems that are capable of recognizing objects from image data. This means, given a set of known objects, the system should be able to make a decision if one out of the set of all objects is visible (*object detection*) or which one of these objects is presented (*object identification*).

Considering a robot in an office environment, such an object recognition system is useful, as it enables the robot to perform tasks like finding a certain object. For example, the robot could react appropriately to a request like “Fetch my cup”, if it has learned the appearance of “my cup”. However, for many tasks, this is not sufficient. If I ask for just “a cup”, it might not be able to solve this task, as it cannot generalize from the class “my cup” to the much larger class “cups”, which we will refer to as the *category* “cups”.

From this simple example, we can already see how natural it is for humans to express commands or statements in terms of categories. As a conclusion, machines that can interpret data in terms of categories can interact more easily with humans.

Although humans can handle categories with ease, current approaches to this topic in computer vision have shown limited progress until now. The reason for this is illustrated in Figure 1.1. Considering the categories shown in Figure 1.1, traditional cues like color, texture and shape are often not able to reliably distinguish between certain category members. This thesis deals with those difficult categories including challenging examples of categories in real-world settings like shown in figure 1.1.

Figure 1.1: Examples of images used in experiments in this thesis. Note that the cluttered background can be distracting for recognition algorithms.

1.1 Contributions of the Thesis

The overall goal of this thesis is to use visual information to recognize object categories such as those shown in Figure 1.1. As the recognition of object categories is an extension to the well researched topic of object recognition, it is known that for achieving good performance in real-world settings one has to tackle the challenges of cluttered background, occlusion, varying lighting conditions and noise. In addition to this we have to handle the severe changes in appearance introduced by the large diversity of the category members.
Therefore we require for a system which recognizes object categories:

- *Robust representation*: The representation has to extract the information which is common to all of the members of a category and discriminate them from members of other categories. Furthermore it has to be robust with respect to signal changes introduced by clutter, occlusion, varying lighting conditions and noise.

- *Robust classification*: The classification algorithm must be able to generalize over the large variety of the category members. In addition to this, it has to face all the challenges which arise from real-world settings that could not be handled by the representations.

Therefore we combine recent progress in robust representation with a state-of-the-art learning technique to meet the requirements for recognizing object categories.

Various data representations have been proposed, which show very good performance in object recognition tasks. However, methods which use a global representation of an image like

in [Mur95] often suffer from cluttered background and occlusion. Local feature representations seem to be a promising solution to this problem, as they rely on local information which is not globally influenced by partial occlusion or background clutter [Sch97], [Low99].

On the other hand, a lot of progress has been made in machine learning by the introduction of Support Vector Machines (SVMs, [Vap96], [Sch01]), that perform extremely well on a broad variety of learning tasks also in computer vision [Cha99], [Roo01].

Recently the robust representation by local features and the excellent generalization capabilities of SVMs were combined via a new kernel function in [Wal03b] called *local kernel*. This combination of local features and SVM seems to be capable of meeting the tough requirements on representation and classification described above. Therefore it is a very promising approach to the categorization problem and is explored in more detail in this thesis.

One of the interesting aspects of [Wal03b] is the introduction of a constraint which helps identify matching features within the local kernel. This thesis contributes two new types of constraints for this.

Furthermore, the whole approach is extensively evaluated with respect to scale, occlusion and background clutter. Different methods are applied to improve performance. Also the influence of the number of categories and number of presented examples of a category is investigated. For the experiments in real-world settings new images, which also include categories commonly found in office environments, were contributed to existing image collections. We consider very challenging categories like cow, horse and dog with great visual similarity, too.

Detailed reviews of the literature on categorization, data representation and classification is given in the Chapters 2, 3, 4 respectively.

1.2 Outline

The rest of the thesis is organized as follows. Chapter 2 discusses definitions of categorization and how it is considered in our work. Chapter 3 describes the local feature representations. Chapter 4 gives an introduction to Support Vector Machines and how they are applied to local feature representations. In Chapter 5 experiments are described and results are presented. The thesis is summarized in Chapter 6.

Chapter 2

Categorization

To be able to think about categorization, first we need a definition of the term “category”. Following the discussion in [Lak87], this chapter reviews both classical and modern theories concerning the nature of categories. We will conclude that for computer vision applications only the classical approach seems applicable. However, also the term “categorization” itself will reveal problems due to its ambiguity. For example it is unclear if categorization includes the finding of categories themselves or only refers to dealing with them, like being able to determine the membership of an observed entity.

2.1 Categories

Intuitively, categories mean classes that were extended in manner that seems natural to us. We introduce the new term categories, as the standard way of combining entities into classes does not seem to account for the greater amount of variability we allow for categories. The purpose of introducing such meta classes at all is clear. Without reducing the complexity of the world as we perceive it, we would be lost in details. Furthermore, abstract entities would be out of the range of our reasoning, as they only exist in the context of categories.

Also in encyclopedias make similar conclusions and state that categories are the basic building block with which we formulate our thoughts, which develop naturally from the way we generalize and think [Bri04].

These are very vague descriptions of the term category. Computational systems require a firmer definition, and for that we must turn to literature in philosophy and science.

From the work of Aristotle [Ed28] to the modern approaches of *prototype theory* [Ros88] the meaning of the term category has undergone considerable changes. Therefore literature dealing

with this subject from a theoretical (philosophy, cognitive science) or computational (e.g. artificial intelligence, computer vision) point of view does not deliver a canonical definition of this term either. A brief review of possible definitions of categories is given below, closely following the introduction given by Lakoff in [Lak87].

Lakoff identifies two basic views on categories. The first he calls the *classical view* which has its roots in the work of Aristotle. Categories were considered to be something abstract and well defined. Boundaries were thought to be defined by shared properties. A more recent approach is the *prototype theory*[Ros88] which extends the classical concept and considers categories to be far more complex. In this context Lakoff summarizes different types of categories, which can be modeled by prototype theory. We will review them from a machine learning point of view.

- **graded categories**: Not all the members of a category have the same degree of membership. This results in fuzzy boundaries and central members. For central members a human observer should be sure about the membership with respect to a certain category, but approaching the border multiple opinions might exist. An example of a graded category is the category "tall man". For sure, there can be no disagreement classifying extremely tall people. Yet for people whose height is not exceptional, but still clearly above average, one cannot be sure about. Methods like fuzzy sets were introduced to handle such situations, since this kind of categories will introduce ambiguities, that are difficult to resolve.

- **categories with clear boundaries**: Even though a category can be internally graded, there is consensus on the membership of an entity with respect to such a category. An example of a category with clear boundaries is the category bird. As there exists a precise definition of what a bird is, the membership and therefore also the boundaries are well defined. For computational systems, such categories seem much more feasible to handle, especially with respect to discriminative classification approaches like the one used in this thesis, as they introduce sharp boundaries by a decision function.

- **basic level categories**: Basic level categories are embedded within more general and more specific categories in a hierarchy. They distinguish themselves from the others as they are basic with respect to the way we perceive or deal with them. An example of a basic level category is the category dog. Although it is embedded in a hierarchy in between more general categories like mammal and more specific ones like sheep-dog, we most likely refer to it as dog first, when we see it. Hierarchical clustering and classification is a widely used technique in machine learning. However, the choice of the right level of generalization is mostly covered by heuristics, as no canonical level can be identified.

- **embodied categories**: Categories are tied to the way humans think and do not exist without a human observer. In particular, at least some categories are embodied, which means that they depend on the environment and the observers' capabilities and experience. An often cited example of embodied categories are categories for colors. They depend on experience and the physics of the visual system. Therefore many categories might be beyond the capabilities of learning approaches as machines lack these human properties.

These examples show that many types of these categories can in principle be modeled by machine learning techniques. Yet how far we can get in practice in forcing methods to model categories perceived by humans is an open question.

In the following an overview of state-of-the-art approaches to the categorization problem in computer vision is given.

2.2 Current State-of-the-Art within Computer Vision

As we are looking at achievements of categorization in computer vision, we restrict our discussion to visual categories. As mentioned above it has at least to be questioned to what extend we can talk of learning categories in terms of machine learning.

Although sometimes not using the term *category*, problems involving categories have been addressed for a long time in computer vision. Recent work was done on detecting cars [Aga02], faces [Sch00c] and humans and horses [For97] in real-world settings, and in [Nel98] images of cups, fighters, snakes, planes and cars were recognized in homogeneous background.

Some of the works with more awareness on the category issue are now reviewed in more detail.

Work of Weber et al. [Web00b] and its extensions in [Web00a] a categorization task is performed in terms of learning object class models from unlabeled and unsegmented images of cluttered scenes. These models consist of constellations of rigid features which are used to represent characteristic parts of the object by selecting them with respect to their distinctiveness on the data. The classification itself is performed in a probabilistic framework by computing joint probability density functions of the feature appearance and their constellation.

Work of Fergus and Fei-Fei et al. [Fer03] and [FF03] are extensions to the work of Weber. The main improvement is the explicit modeling of variations in appearance within a category. Fergus uses the EM-algorithm to learn new categories. The focus of [FF03] is to reduce the

amount of training data required to learn a new category. This is achieved by using priors learned from other categories. Therefore they use a variational Bayesian approach. By this technique the estimation of a model for a specific category is dramatically simplified and can be done based on only a few training examples (1 to 5).

Work of Leibe et al. Although in [Fer03] it is stated that there is broad agreement of the issue of representation - namely object categories are represented as a collection of features or parts, where each part has a distinctive appearance and spatial position, in [Lei03a] global representation besides local representation performed well, too. The author explicitly restricts his investigations to basic-level categories. For a subset of these categories, cars and cows, results are reported in [Lei03b] for a category-specific figure-ground segmentation task. Therefore a statistical model based on a codebook generated from a local feature representation is used.

As far as we know all the work which is related to recognizing categories in real-world settings is limited to 1 to 6 categories. In addition these categories are fairly distinctive. Examples are:

- [Web00a]: cars, leaves
- [Fer03]: motor bikes, airplanes, faces, cars(side), car(rear), spotted cats
- [Lei03b]: cars, cows

2.3 Problem Definition

In this thesis we consider the categories to be explicitly specified by labeled examples of the categories, which enforces the *classical view* of categories. In this context, categorization can be described as object recognition with a dramatically increased variation in the object classes, which by far surpasses the simple visual similarities handled in object recognition. Considering the conclusions drawn above, the automatic detection of categories themselves has inherent problems due to the nature of many categories and is therefore not considered in this thesis. In contrast to recent work, we will attempt to handle categories which are hard to classify due to their visual similarity. Therefore we will also consider categories like cow, horse and dog.

2.4 Summary

As categories are considered to be strongly related to the way humans generalize and think, it is clear how valuable it is representing information about data in terms of categories. Although there is recent progress in describing category phenomena with the concept of *prototype theory*, we restrict ourselves to the *classical view*, as we train models on labeled data sets. Therefore in this thesis categorization is understood as the task of recognizing object categories previously defined by humans, and where the boundaries of categories are clear.

Although there has been some work on categorization in computer vision, there are still clear limitations to what has been achieved. Especially in real-world settings, the number of categories addressed in recent literature does not exceed six and the categories are chosen that the members have a fairly distinctive visual appearance.

Chapter 3

Local Image Features

Recently, *local feature representations* have gained a lot of interest in computer vision. In contrast to *global representations* which are computed on the whole image, local features capture the appearance only at a set of points called the *interest points*.

Global representations like histograms [Sch00a] or eigenspace representations [Mur95] are popular approaches to encode global image characteristics. Although they have shown good performance in many experiments, they usually suffer in the presence of a heterogeneous background and occlusions. This is one of the reasons why local feature representations gained interest as they show very good performance even with clutter and occlusion [Sch96] and therefore they are considered to be a promising solution to these problems.

Another issue why local features are considered to be a good choice is related to the task of recognizing object categories. In works like [Web00a] and [Fer03], the use of local features is motivated by identifying and redetecting parts of an object, which are characteristic for the whole category. For example, in order to describe a car, we are interested in finding tires, headlights, the windshield and so on. These parts have to be described in a manner so that is distinctive, but also accounts for the large variability allowed within a category.

The acquisition of a local feature representation is divided into two steps:

1. *Detection*: An *interest point detector* is used to determine the position of characteristic features in the image.

2. *Description*: A *local descriptor* is computed at each detected interest point to represent the local appearance.

In Section 3.1 a brief summary on scale-space is presented, as it offers theoretical insight to the process of local feature extraction and leads to a scale invariant representation. Then Section

3.2 presents methods for detecting interest points. In Section 3.3 two local feature descriptors are described. Finally a summary of the Chapter is given in Section 3.4. The whole Chapter is a review of techniques.

3.1 Theory of Local Features

Assuming that an image is nothing more than a collection of primitive structures like edges, blobs and corners, one can make the important observation that these structures appear at certain scales, according to their physical extent in the scene, distance to the camera and resolution of the image. To describe how to exploit this, the principals of scale-space notation will be reviewed in Section 3.1.1. In preparation of Section 3.2 where we want to be able to compute derivative based quantities in scale-space, Section 3.1.2 describes the technical basics and Section 3.1.3 explains how to select characteristic scales.

3.1.1 Scale-space

The size of objects in images is influenced by the image resolution, focal length of the camera and other, generally unknown, parameters. This motivates for multi-scale image representations that explicitly represent the image at different scales of observation. Among many alternative approaches to construct such a representation, Gaussian scale-space theory has shown to be a natural choice due to its convenient mathematical properties and close relations to biological vision [You87].

Starting from a set of axioms, [Wit83] and [Koe84] derived a Gaussian scale-space representation L and have shown that it has to satisfy the diffusion equation:

$$\frac{\partial L}{\partial \sigma^2} = \frac{1}{2}\nabla^2 L \tag{3.1}$$

with the initial condition:

$$L(\boldsymbol{x}, 0) = s(\boldsymbol{x}) \ , \tag{3.2}$$

where $L : \mathbb{R}^D \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is the scale-space of a continuous signal $s : \mathbb{R}^D \rightarrow \mathbb{R}$ and σ is a continuous scale parameter. The solution of this diffusion equation can be computed by convolution with a Gaussian kernel:

$$g(\boldsymbol{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} e^{-\frac{\sum_{d=1}^{D} \boldsymbol{x}_d^2}{2\sigma^2}} \tag{3.3}$$

original $\sigma = 1$ $\sigma = 2$ $\sigma = 4$ $\sigma = 8$

Figure 3.1: Slices through the scale-space of the image on the left at different values for σ

Therefore the Gaussian scale-space is given by:

$$L(\boldsymbol{x}, \sigma) = g(\boldsymbol{x}, \sigma) * s(\boldsymbol{x}) \ , \tag{3.4}$$

where $*$ denotes convolution.

As we are interested in a scale-space representation of an image, we restrict ourselves for further consideration to two dimensions. Therefore the scale-space is built by:

$$L(\boldsymbol{x}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\boldsymbol{x}^2+\boldsymbol{y}^2}{2\sigma^2}} * f(\boldsymbol{x}) \ , \tag{3.5}$$

where $\boldsymbol{x} = (x, y)$ specifies a position in the image f. Examples of slices through such a scale-space are presented in Figure 3.1. The image of the car is filtered with Gaussian kernels with increasing σ. Moving to higher scales the image gets more and more blurred. Intuitively, this low-pass filtering can be thought of as simulating the loss of information when moving to higher scales or respectively moving away from an object, which results in changes of image characteristics like derivatives. But even though information of the original image is lost, the scaling reveals new characteristics, which are typical for higher scales. In Section 3.2 we will exploit this fact to build richer models. In contrast to resolution pyramids [Cro82] the images are not down-sampled to preserve the spatial resolution in the feature detection procedure.

Due to quantization and noise there is a lower limit for the scale, called the inner scale. This is basically the scale where the support of the Gaussian lies totally within one pixel. Obviously the result of the convolution will not change by further reducing σ. There is also an outer scale, which means that due to the finite size of the image, the size of structures that can be captured is limited. This limit is reached when the support of the Gaussian is about the size of the whole image. Therefore it makes only sense to perform computations at scales which lie between these bounds.

3.1.2 Derivatives

As will be seen in Section 3.2 and 3.3, popular approaches to find interest points and to describe their local characteristics need the computation of derivatives. The standard method for computing derivatives is to derive a filter kernel by using a finite difference approximation [Pau97]. For the first order derivative the central difference is used:

$$\frac{\partial}{\partial x_i} : \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \tag{3.6}$$

The second order derivative is computed by the consecutive convolution with two finite difference approximations of first order derivatives.

$$\frac{\partial^2}{\partial x_i^2} : \begin{pmatrix} -1 & 1 \end{pmatrix} * \begin{pmatrix} -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} \tag{3.7}$$

Higher order derivatives are computed by combinations of eqn. (3.6) and eqn. (3.7). Computing these derivatives at a certain scale σ in scale-space we get:

$$L_{i_1 \ldots i_m}(\boldsymbol{x}, \sigma) = \underbrace{\frac{\partial^m}{\partial i_1 \ldots \partial i_m} * g(\sigma)}_{\text{Gaussian derivatives}} * f(\boldsymbol{x}) \tag{3.8}$$

As the values of the derivative decreases with increasing scale, in [Lin98] scale normalized derivatives were introduced,

$$D_{i_1 \ldots i_m}(\boldsymbol{x}, \sigma) = \sigma^m L_{i_1 \ldots i_m}(\boldsymbol{x}, \sigma) \tag{3.9}$$

which will become of importance in Section 3.3 to derive a scale-invariant representation. The normalized Laplacian yields:

$$\nabla^2 L_{norm} = \sigma^2 (L_{xx} + L_{yy}) \tag{3.10}$$

Gaussian derivatives develop from the scale-space representation, but have also shown to be more stable than the normal derivative. As the derivations act as high pass filters, the Gaussian smoothing reduces the otherwise amplified noise [For03]. In Figure 3.2 the filter kernels of the relevant Gaussian derivatives and the Laplacian are displayed. These filters have recently gained interest in various fields of computer vision [Hay04] and were also motivated from human perception in psychophysics.

The success of these approaches is based on capturing different characteristics of an image

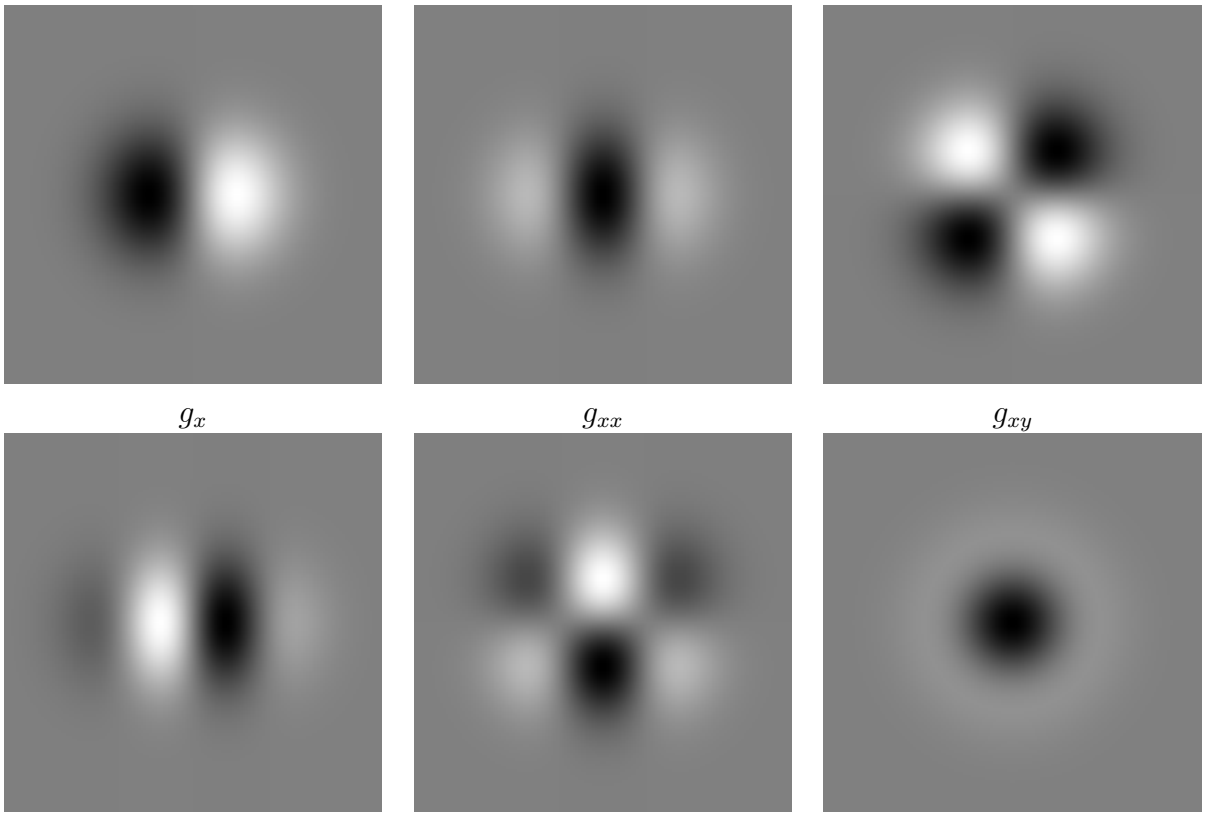


Figure 3.2: Filter kernels for Gaussian derivatives and the Laplacian

by choosing different σs for the computation of derivative based statistics. But as there is no a priori information how to choose σ such approaches can suffer from the increased amount and dimensionality of the data. Therefore in [Lin98] the theory for automatic scale selection has been introduced, which also lead to a scale-invariant representation.

3.1.3 Scale selection

The assumption of scale selection is that there is a characteristic scale at which a function of Gaussian derivatives, called a differential entity, achieves a maximum. If this extremum can be reliably detected, a scale-invariant representation can be constructed and the ambiguity of representing the same image feature at different scales due to the introduced scale parameter σ is eliminated again. In [Lin98] it was shown that functions like $trace(\mathcal{H})$ (Laplacian) and $det(\mathcal{H})$ attain a maximum on synthetic data and real images, where $\mathcal{H}$ is the Hessian matrix in the context

of normalized scale-space derivatives is given by:

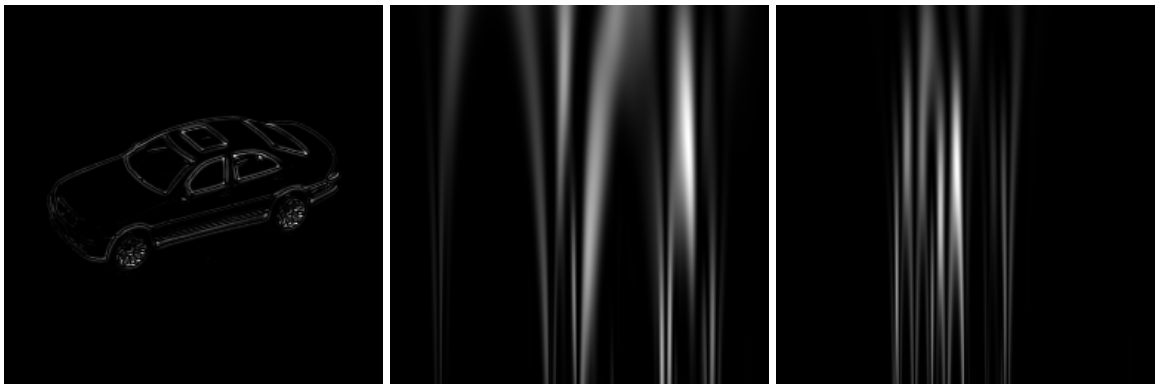
$$\mathcal{H}(\boldsymbol{x},\sigma) = \sigma^2 \begin{pmatrix} L_{xx}(\boldsymbol{x},\sigma) & L_{xy}(\boldsymbol{x},\sigma) \\ L_{xy}(\boldsymbol{x},\sigma) & L_{yy}(\boldsymbol{x},\sigma) \end{pmatrix} \tag{3.11}$$

In experimental evaluations in [Mik01], the Laplacian yields good results for scale-selection. To use the Laplacian to detect maxima in scale-space, we need a scale derivative of the Laplacian. Considering eqn. (3.10), the scale derivative of the normalized Laplacian is given by ([Lap04],[Lin93]):

$$\begin{aligned} \frac{\partial}{\partial\sigma^2}\nabla^2 L_{norm} &= \frac{\partial}{\partial\sigma^2}\sigma^2(L_{xx}+L_{yy}) && (3.12)\\ &= L_{xx}+L_{yy}+\sigma^2\frac{\partial}{\partial\sigma^2}(L_{xx}+L_{yy}) && (3.13)\\ &= L_{xx}+L_{yy}+\frac{\sigma^2}{2}(L_{xxxx}+2L_{xxyy}+L_{yyyy}) && (3.14) \end{aligned}$$

In eqn. (3.12) the normalized derivative from eqn. (3.10) is used and in eqn. (3.13) only the product rule of derivation is applied. For eqn. (3.14) we look at $\frac{\partial}{\partial\sigma^2}(L_{xx}+L_{yy})$ as a scale-space of the the function $L = L_{xx}+L_{yy}$ and replace it with the right-hand side of the diffusion equation (3.1). By using this trick, no additional scales have to be computed, as it would be the case for a finite difference approximation in scale.

In Figure 3.3 two slices through scale-space parallel to the x and y axes of the car image shown in Figure 3.5 are given. Each column of these images is called scale-space signature or scale-space trace in the literature. As predicted by the theory, maxima can be observed on these traces. A more detailed look reveals some typical properties of these traces. First, the traces one would identify by a first glance at Figure 3.3 are no straight lines going from the bottom to the top. Many of them are bent. These bent traces are called deep structure. This property is illustrated in Figure 3.4, where features are extracted at a corner at different scales. The detected location is marked by a cross and the scale is visualized by the size of a circle centered at that position. The typical projection of a deep structure lying approximately on the bisecting line of the corner can be observed. We will refer to methods which deal with this property in Section 3.2.



(a) (b) (c)

Figure 3.3: (a) Laplacian of the input image of size 256x256 ; (b) and (c) slice through scale-space of the Laplacian at $y = 128$; right: slice through scale-space of the Laplacian at $x = 128$ (scale-space displayed from $\sigma = 1$ at bottom of the image to $\sigma = 8$ at the top)

3.2 Detection

As already mentioned, the process of extracting local features is divided into two steps: the interest point detection and description. Interest points are required to be characteristic for the image and robust to redetect. The information content of the descriptor, which is computed in the second step, is highly dependent on the chosen interest points [Mik02a]. A common assumption for a characteristic point is that there is a significant change in the intensity value of the image and therefore a strong derivative at the interest point. A popular choice for interest points are local maxima of a function of derivatives, which will be referred to as an interest function. As a representative, the *Laplace detector* will be reviewed in Section 3.2.1. But edge-like structures can also preserve a high score along a line and often don't show well defined maxima. For this reason more selective functions were defined such as the *Harris detector* [Har88] described in Section 3.2.2. Thereafter, methods for dealing with scale are described - namely the multi-scale approach [Sch97] in Section 3.2.3 and automatic scale selection [Mik01] in Section 3.2.4. Finally Section 3.2.5 briefly reviews experimental comparisons of interest point detectors.

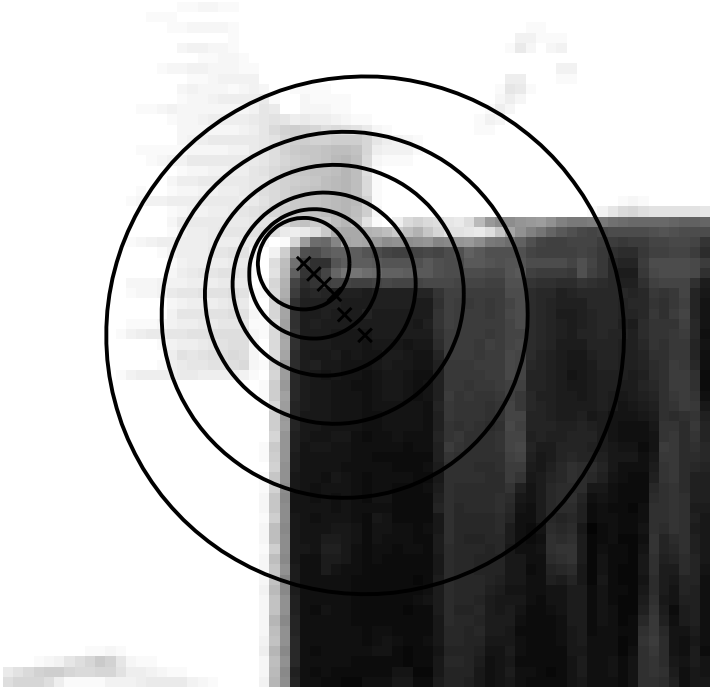


Figure 3.4: Features detected at a corner at multiple scales. The feature positions are marked with a cross and the scale is visualized with a circle.

3.2.1 Laplace detector

A common approach to look for interest points is to look for the maxima of the Laplacian for the Laplacian of Gaussian in a scale-space context with scale normalized derivatives:

$$\nabla^2 L(\boldsymbol{x}, \sigma) = |\sigma^2 L_{xx}(\boldsymbol{x}, \sigma) + \sigma^2 L_{yy}(\boldsymbol{x}, \sigma)| \tag{3.15}$$

The associated filter is displayed in Figure 3.2. Convolving with such a filter leads to an interest function which responds to blob-like structures similar to the filter itself. But also edges will lead to high output, and so this filter is also applied as an edge detector. Figure 3.6 shows two examples of the interest function and the detected points computed by the Laplacian. A high score is associated with the contour which has a strong edge. Of course, lots of these points will not be redetected, as they do not correspond to a characteristic part of the scene. Another observation is that the interest points seem to be distributed randomly on the contour which illustrates the already mentioned problem of there being no well-defined maxima along an edge.

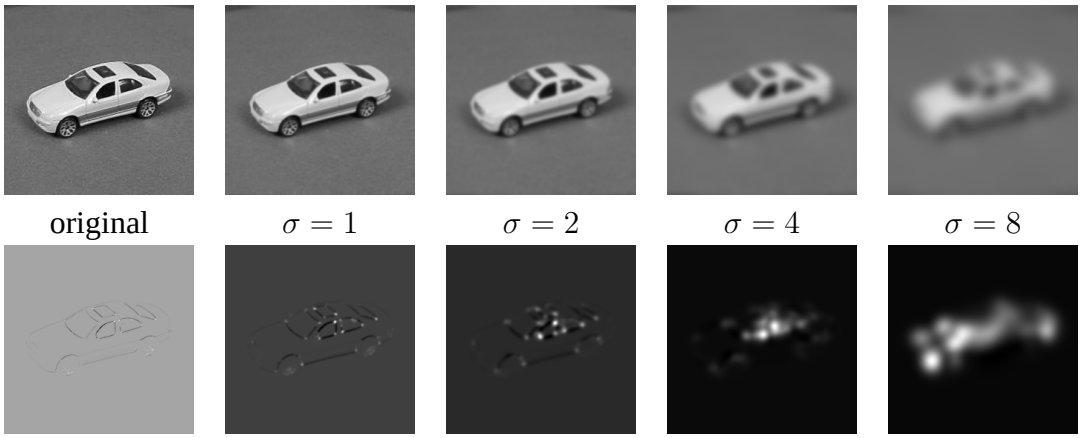


Figure 3.5: first row: image data at different scales; second row: Harris function (normalized for displaying) at different scales

3.2.2 Harris detector

Some of these problems can be solved by a different approach called the Harris corner detector [Har88]. The basic building block of this method is the second moment matrix:

$$\mu(\boldsymbol{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{pmatrix} L_x^2(\boldsymbol{x}, \sigma_D) & L_x L_y(\boldsymbol{x}, \sigma_D) \\ L_x L_y(\boldsymbol{x}, \sigma_D) & L_y^2(\boldsymbol{x}, \sigma_D) \end{pmatrix}, \tag{3.16}$$

where σ_I is the integration scale at which the derivatives are computed and σ_D the detection scale, which can be thought of as an additional smoothing to stabilize the detection of local maxima. The integration scale is determined by the scale at which one wants to detect interest points and a common choice for the detection scale is $\sigma_D = 2 * \sigma_I$. From this matrix the Harris interest function:

$$det(\mu(\boldsymbol{x}, \sigma)) - \alpha(trace(\mu(\boldsymbol{x}, \sigma))^2) \tag{3.17}$$

is computed, where $\alpha = 0.04$ is proposed in [Har88].

Intuitively, the Harris detector looks for points that have strong curvature in the two orthogonal principal directions. This is enforced by seeking local maxima of the determinant of the second moment matrix, which is the product of the eigenvalues belonging to the orthogonal eigenvectors of this symmetric matrix. The term on the right can be thought of as a penalty term for edge-like structures.

Two examples of the interest function and the detected points computed by the Harris detector are shown in Figure 3.6. The interest points are no longer concentrated on the edge and are more spread out over the whole object. The Harris detector has shown good performance in

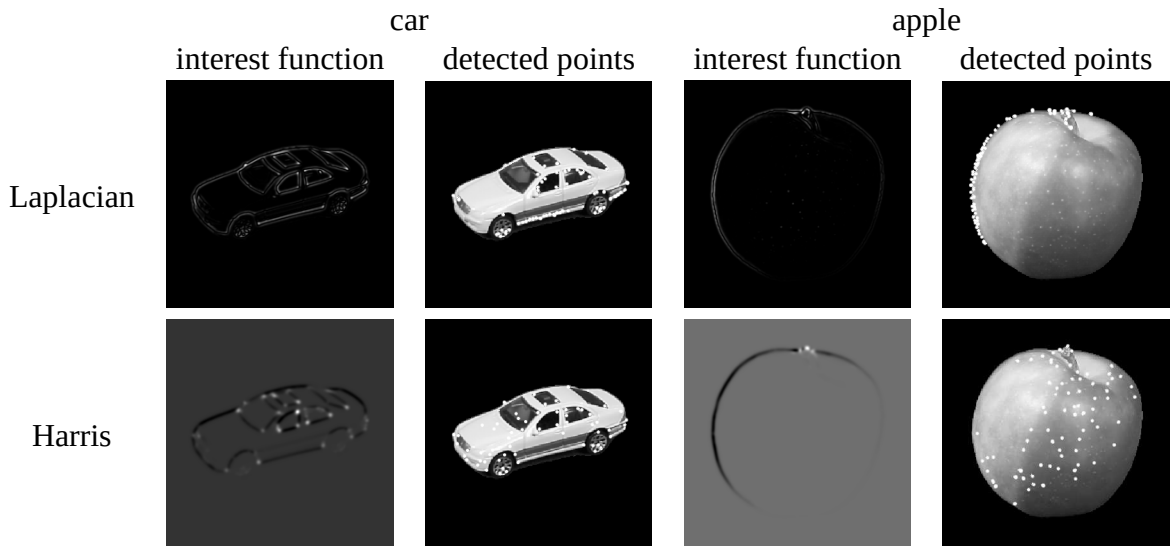


Figure 3.6: Example for interest points computed by Laplace and Harris measure. The corresponding interest functions are shown, too. The weakly textured object apple on black background reveals problems of the Laplacian measure.

comparison with other detectors [Sch98]. In Figure 3.5 the Harris interest function is computed at different scales. Assuming that the Harris detector is useful for detecting characteristic interest points, one can observe that for different scales different structures of the image are considered characteristic. Again this affirms the hypothesis that varying a scale parameter in feature extraction leads to a model capable of capturing characteristics which otherwise would be undetected.

But as already mentioned in Section 3.1.1 the choice of the scale parameter σ is unclear. Therefore two approaches will be given how to take advantage of this scale parameter without running in the problem of storing a lot of redundant information.

3.2.3 Multi-scale

The multi-scale approach, which was applied in [Sch97], samples the scale-space at a set of scales $\sigma_1, \ldots, \sigma_n$. The spacing should be chosen in an exponential manner $\sigma_i = \sigma_0 s^i$ for some fixed $s \in \mathbb{R}$. As for example the size of an object changes in the same way with respect to the distance of the observer, this spacing seems natural when dealing with scale. At each scale interest points are detected. They are gathered into a single set of all interest points across all scales. Of course many of these points will refer to the same structure viewed at different scales especially when the spacing of the sampling is very fine. This will cause redundancy in the model and lead to higher computational costs and larger storage requirements.

3.2.4 Automatic Scale Selection

The redundancy introduced by the multi-scale approach from Section 3.2.3 can be reduced by applying the theory reviewed in Section 3.1.3. This is done by storing the feature at the scale at which it yields the strongest response of the interest function. In [Mik01] a combined Harris-Laplace detector was presented which finds scale-space maxima with an iterative algorithm, and in [Low99] a more exhaustive search for localizing scale-space interest points was used, which is referred to as the SIFT key detector.

Harris-Laplace Detector

In [Mik01] a Harris detector is used for detecting interest points in the spatial domain and a Laplace measure for the scale domain. In addition one has to compensate for the problems mentioned in Section3.1.3. The deep structures we perceive in Figure 3.3 and 3.4 are bent and therefore the detected interest points at different scales differs by several pixels. To solve these problems, an iterative algorithm was introduced in [Mik01], which takes initial detections of points and searches along scale traces for a characteristic scale where the Laplacian attains a maximum. After each iteration the location of the point is redetected with the Harris detector to update the position. A Nassi-Schneiderman diagram of the algorithm is given in Figure 3.7.

Further extensions like affine invariance were not considered, as in [Mik02a] an experimental evaluation showed that with small viewpoint variations between images ($< 20°$), the Harris-Laplace detector showed better repeatability than its affine counterpart.

SIFT Detector

In [Low99] a new type of local features is introduced, called *SIFT*. The interest point detector, which Lowe calls key detector, uses a scale selection mechanism based on differences of Gaussians. The scale-space is built by convolving with Gaussians and down-sampling after each octave, so that a pyramid-like data structure is obtained. The difference of Gaussians are computed by the difference of neighbouring scales. After that, interest points are detected by looking for maxima with respect to the eight bordering pixels. In a second step all the points which represent a maximum in scale-space are selected, by checking the closest pixel at the next higher and next lower scale.

3.2.5 Comparison

In the beginning of this Chapter, the assumption was made that local features capture the appearance at characteristic points in the image. That they are characteristic implies that they can be reliable redetected in many views of the scene varying for example the lighting conditions and the viewpoint. Therefore experimental evaluations of the repeatability of interest point detectors under typical transformations and degradations of the image data were made in [Sch00b] and [Mik02b]. The Harris detector and its extension to scale have yielded favourable results.

3.3 Description

After the detection of the interest points the appearance of the local vicinity around the interest point has to be captured. Therefore an interest point descriptor is computed based on a patch around the interest point. On the one hand this descriptor should be discriminative, so that it captures characteristic information of that patch. On the other hand it should be robust or even invariant to changes not of interest in the specific task. A typical example is invariance to rotation. These properties are conflicting, and the question how to balance them is unsolved especially in categorization. For rather continuous changes produced by noise, varying lighting conditions and affine transformations, invariants were introduced. But for changes caused by capturing the same structure of different members of a category, the task of computing invariants seems hopeless. For that reason it is concluded that one has to rely on the learning stage to generalize over such severe signal changes, and try using techniques from object recognition to describe the appearance.

In Section 3.3.1 the definition of the *local jet* descriptor is given while Section 3.3.2 reviews the *SIFT* descriptor. Section 3.3.3 refers to experimental comparisons of local image descriptors.

3.3.1 Local Jet

In [Koe87] a descriptor called the *local jet* was introduced. This approach may be written in a scale-space notation as:

$$J^N[f](\boldsymbol{x}, \sigma) = \{L_{i_1 \dots i_n}(\boldsymbol{x}, \sigma) | (\boldsymbol{x}, \sigma) \in \mathbb{R}^2 \times \mathbb{R}^+, n = 0, \dots, N\} \tag{3.18}$$

In this thesis *local jets* including derivatives up to third order are used to capture the local appearance:

$$\begin{pmatrix} L_x(\boldsymbol{x},\sigma) \\ L_y(\boldsymbol{x},\sigma) \\ L_{xx}(\boldsymbol{x},\sigma) \\ L_{xy}(\boldsymbol{x},\sigma) \\ L_{yy}(\boldsymbol{x},\sigma) \\ L_{xxx}(\boldsymbol{x},\sigma) \\ L_{xxy}(\boldsymbol{x},\sigma) \\ L_{xyy}(\boldsymbol{x},\sigma) \\ L_{yyy}(\boldsymbol{x},\sigma) \end{pmatrix} \tag{3.19}$$

The grey value itself at position $\boldsymbol{x}$ was not considered, as it is very dependent on the lighting conditions. Furthermore, there is a need for a scale-normalized derivatives for the scale invariant representation. A scale normalized descriptor is obtained by:

$$\begin{pmatrix} \sigma L_x(\boldsymbol{x},\sigma) \\ \sigma L_y(\boldsymbol{x},\sigma) \\ \sigma^2 L_{xx}(\boldsymbol{x},\sigma) \\ \sigma^2 L_{xy}(\boldsymbol{x},\sigma) \\ \sigma^2 L_{yy}(\boldsymbol{x},\sigma) \\ \sigma^3 L_{xxx}(\boldsymbol{x},\sigma) \\ \sigma^3 L_{xxy}(\boldsymbol{x},\sigma) \\ \sigma^3 L_{xyy}(\boldsymbol{x},\sigma) \\ \sigma^3 L_{yyy}(\boldsymbol{x},\sigma) \end{pmatrix}, \tag{3.20}$$

where σ is the scale at which the feature was detected.

3.3.2 SIFT

As a starting point for the SIFT descriptor proposed in [Low99], the magnitude M_{ij} and direction R_{ij} of the gradients A_{ij} at a certain scale σ and pixel position i, j are computed:

$$M_{ij} = \sqrt{(A_{ij} - A_{i+1,j})^2 + (A_{ij} - A_{i,j+1})^2} \tag{3.21}$$

$$R_{ij} = \tan^{-1} \frac{A_{ij} - A_{i+1,j}}{A_{i,j+1} - A_{ij}} \tag{3.22}$$

For robustness to illumination changes, the magnitude of the gradient is thresholded. To achieve rotation invariance, canonical directions are computed by detecting maxima in the histogram of local gradient orientations. The magnitude of the gradients is weighted according to the distance to the interest point and the histogram is smoothed, to stabilize the maximum selection.

The SIFT descriptor itself consists of 16 orientation histograms with 8 orientation bins. The area around the interest point is divided into a grid of 4x4 regions. From each of these regions the orientation histograms are computed resulting in a 128 dimensional vector.

3.3.3 Comparison

In [Mik03] Mikolajczyk performed experimental evaluations on the distinctness and robustness of local descriptors under typical transformations and degradations. In these experiments the SIFT descriptor performed best, followed closely by steerable filters [Fre91], which are rotation invariant versions of local jets. Again it has to be mentioned, that these experiments were conducted within the task of recognizing specific exemplars. It is unclear how these results transfer to the task of classifying object categories.

3.4 Summary

After reviewing scale-space theory, different methods for selecting interest points and computing a descriptor at these points have been described. These were extended to handle scale as proposed in the literature. Experimental evaluations from the literature propose the use of the Harris and Harris-Laplace detectors for finding interest points and steerable filters and SIFT descriptors for describing the local features. As we will not consider in-plane rotations of the camera and incorporate rotations in depth in the training, in the following local jets are considered instead of steerable filters.

However one has to note that many assumptions and evaluations were made for object classification. It is unclear how they will transfer to the task of recognizing object categories. Questions like what repeatability of interest points means in the context of categories are still open. For two members of a category there might be structures in the images which refer to the same semantic part, but nothing can be stated about their visual similarity. In fact visual divergence can be an essential part of the category. This holds for example for the category “number plate”.

Concerning variations of the interest point detection and description by influences due to noise and illumination and in particular the shortcomings of modeling intra-category variations, we have to rely on the approach of the learning approach described in the next Chapter.

Harris-Laplace — detection of Harris-Laplace points using scale selection

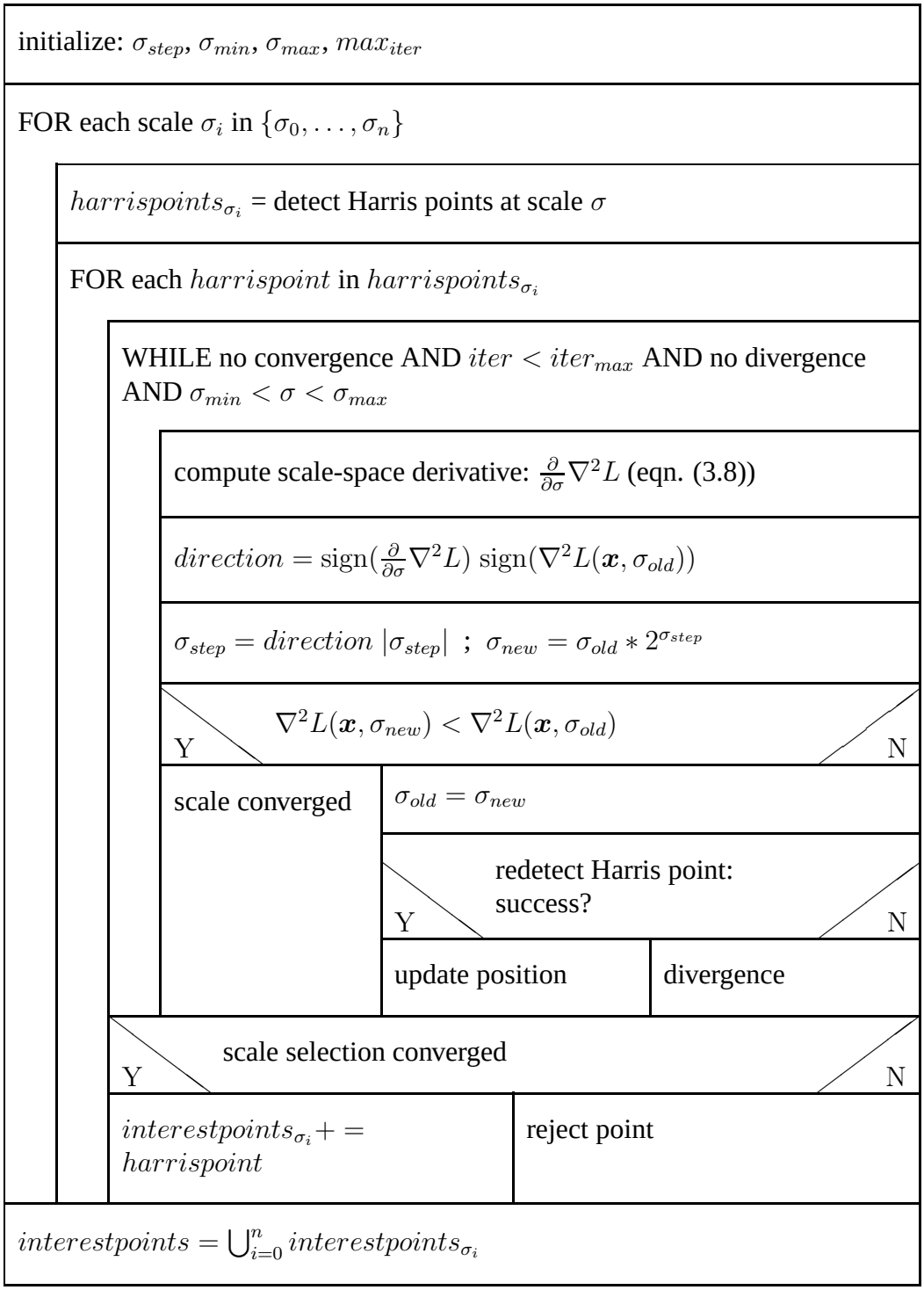


Figure 3.7: Nassi-Schneiderman diagram of the Harris-Laplace detector for finding interest points in the space and scale domain

Chapter 4

Classification

As pointed out in the introduction, besides a robust representation, a method for robust classification is the other essential step in a system for recognizing object categories. Therefore we rely on Support Vector Machines (SVMs) for the learning task, as they have shown excellent generalization performance in pattern recognition, also in various tasks in computer vision.

The task of classification can be described as predicting the class membership of data sample based on features computed from the data. Therefore we distinguish between the feature vector of a data sample $\boldsymbol{x} \in \mathbb{R}^d$ and its label $y \in \mathbb{N}$ (also called target), which determines its class. To solve this task, most proposed methods build a model of the data from a set of examples called the training set $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_l, y_l)\}$ where the label is known. This model is used afterwards to predict the unknown labels $y_1, \ldots, y_m$ of a test set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$. For building such a model there are basically two main approaches. The probabilistic approach with its foundation in Bayesian optimal decision theory [Dud01] and the discriminative approach based on learning theory [Vap96]. The probabilistic approach relies on modeling probability density functions (pdf), from which the likelihood of an observed data sample belonging to a certain class can be computed. Using the probabilistic framework one can take advantage of all the methods developed in the fairly long history of probabilistic calculus [Mac02]. However, choosing a probabilistic approach involves estimating a representative model, which is a more ambitious task than solving the given classification problem [Her01]. This is one of the reasons why discriminative models have recently gained interest. They skip the "overhead" of modeling pdfs by sacrificing the advantages of working in a probabilistic framework, like straight forward formulation of detection and rejection.

Section 4.1 describes Support Vector Machines which are used for classification in this thesis. As we have to deal with multiple classes an overview of heuristic multi-class extension is given

in Section 4.2. To combine the robust representation by local features described in Chapter 3 with this classification method, we make use of a recently proposed type of kernel which is described in Section 4.3. Finally, a summary is given in Section 4.4.

4.1 Support Vector Machines

Support vector machines (SVMs)[Vap96], [Sch01] , which will be the main focus of this Chapter, are discriminative models and have recently raised a lot of interest because of their well-founded theoretical background and very good performance. For describing SVMs the concept of linear discrimination is reviewed in Section 4.1.1 which is extended to optimal separating hyperplanes in Section 4.1.2 and soft margin hyperplanes in Section 4.1.3. Introducing a different formulation using kernels in Section 4.1.4, we will arrive at the concept of SVMs, which allows us to use a certain set of functions described in Section 4.1.5. We closely follow the excellent presentation of this topic given in [Sch01].

4.1.1 Linear Discrimination

The basic idea of a linear decision function is to specify a hyperplane in the input space which separates the two classes. Such a hyperplane is defined by the normal form:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \ , \tag{4.1}$$

where $\boldsymbol{w}$ is the direction normal to the hyperplane and b is the distance of the hyperplane to the origin of the coordinate system.

For every data sample $\boldsymbol{x}$ the distance to this hyperplane can be computed. The sign of the distance tells us on which side of the plane the sample lies. Therefore the decision function is given by:

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b) \tag{4.2}$$

4.1.2 Optimal Separating Hyperplane

The formulation in eqn. (4.2) is not unique, as there are in the case of separable data infinitely many hyperplanes that separate the data. To get to a unique representation, one defines the optimal separating hyperplane, that is also shown to be optimal from a decision theoretical point of view. The optimal separating hyperplane is the plane that has the maximal margin to the data

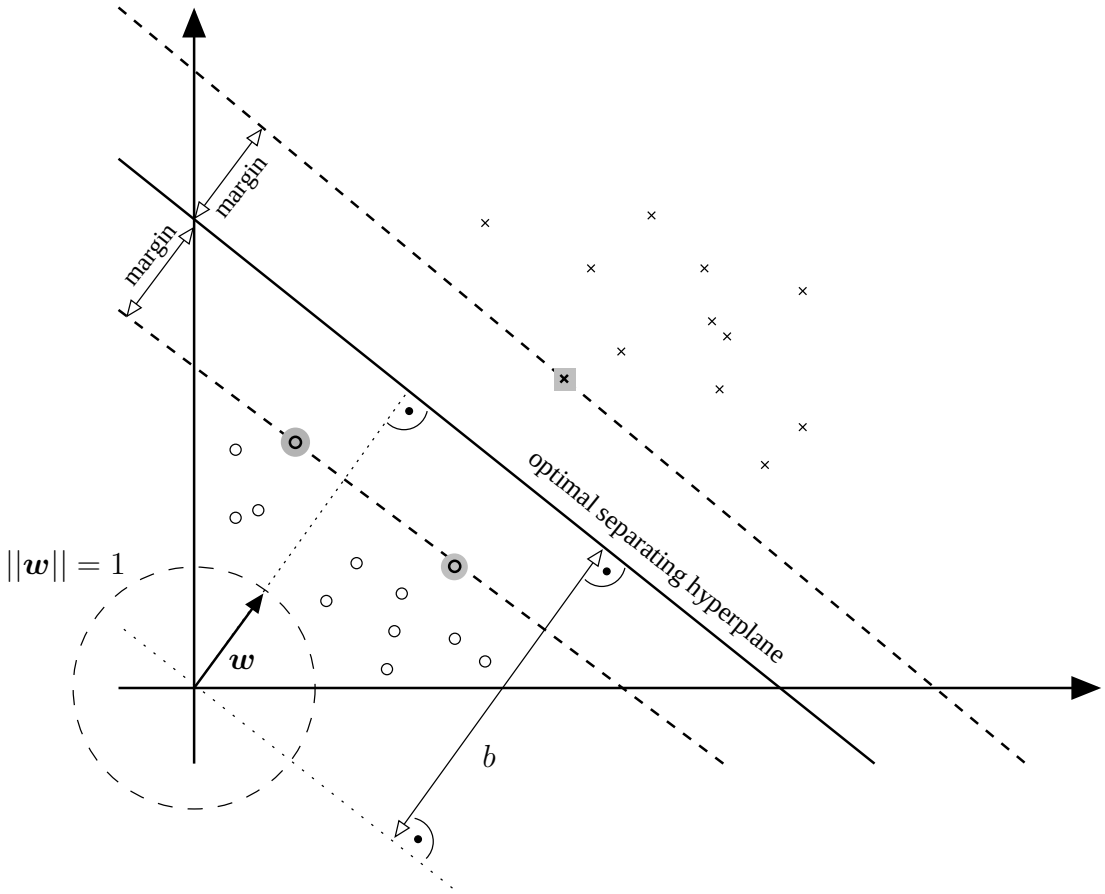


Figure 4.1: Illustration of the normal form of a hyperplane. The orientation of the hyperplane is given by the orthogonal vector $\boldsymbol{w}$ which is constraint to $||\boldsymbol{w}|| = 1$ and the distance from the origin by b. The hyperplane is chosen to separate the two shown classes with a maximal margin. The so-called support vectors are marked in grey.

samples. In order to compute this plane we formulate the following optimization problem:

$$\max_{\boldsymbol{w},b,||\boldsymbol{w}||=1} \{ \min_{i=1,\ldots,l} (y_i(\boldsymbol{x}_i^T \boldsymbol{w} + b)) \} \tag{4.3}$$

This situation is illustrated in Figure 4.1. By normalizing with the length of w, this can be reformulated without the constraint $||\boldsymbol{w}|| = 1$.

$$\min_{\boldsymbol{w},b} \tfrac{1}{2}||\boldsymbol{w}||^2 \tag{4.4}$$

$$\text{subject to} \quad y_i(\boldsymbol{x}_i^T \boldsymbol{w} + b) \geq 1 \ , \ i = 1, \ldots, l \tag{4.5}$$

This is a constrained, convex optimization problem. As a conclusion only one global minimum exists and the solution can be found efficiently. We rewrite the problem by taking the constraints into account via Lagrangian multipliers, obtaining the Lagrange function:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{l} \alpha_i (y_i(\boldsymbol{x}_i^T \boldsymbol{w} + b) - 1) \; i = 1, \ldots, l \; , \tag{4.6}$$

where $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_l$ are the Lagrange multipliers. Therefore the minimum of the optimization problem must satisfy:

$$\frac{\partial}{\partial b} L(\boldsymbol{w}, b, \alpha) = 0 \tag{4.7}$$

and

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \alpha) = 0 \tag{4.8}$$

Substituting eqn. (4.7) and eqn. (4.8) in eqn. (4.6) one obtains the dual optimization problem which is formulated in the variables $\alpha_1, \ldots, \alpha_l$:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^l} W(\boldsymbol{\alpha}) = \textstyle\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{j=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{4.9}$$

$$\text{subject to} \quad \alpha_i \geq 0 \; \forall i = 1, \ldots, l \text{ and } \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.10}$$

As a solution we obtain values for the αs which leads to the following decision function:

$$f(\boldsymbol{x}) = \text{sign}(\sum_{i=1}^{l} y_i \alpha_i \boldsymbol{x}^T \boldsymbol{x}_i + b) \tag{4.11}$$

Computing the αs on typical data sets it turns out that many of αs are 0 and therefore do not contribute to the decision function. The $\boldsymbol{x}_i$ with non-zero αs are called support vectors. To be able to evaluate the model, they have to be stored together with the αs and determine the memory size of the model. In Figure 4.1 the support vectors are marked in grey.

The technique reviewed above solves the problem where the data are linearly separable. Frequently, data we encounter in real-world applications do not show this property. But even if that can be solved, additional problems arise from noise, that can lead to wrong and too complex boundaries. Therefore two extensions were introduced [Vap96], [Sch01]. The *kernel trick* to make the data linear separable by non-linear transformation of the data and the *soft margin hyperplane* which can handle noisy data introducing slack variables with a penalty term.

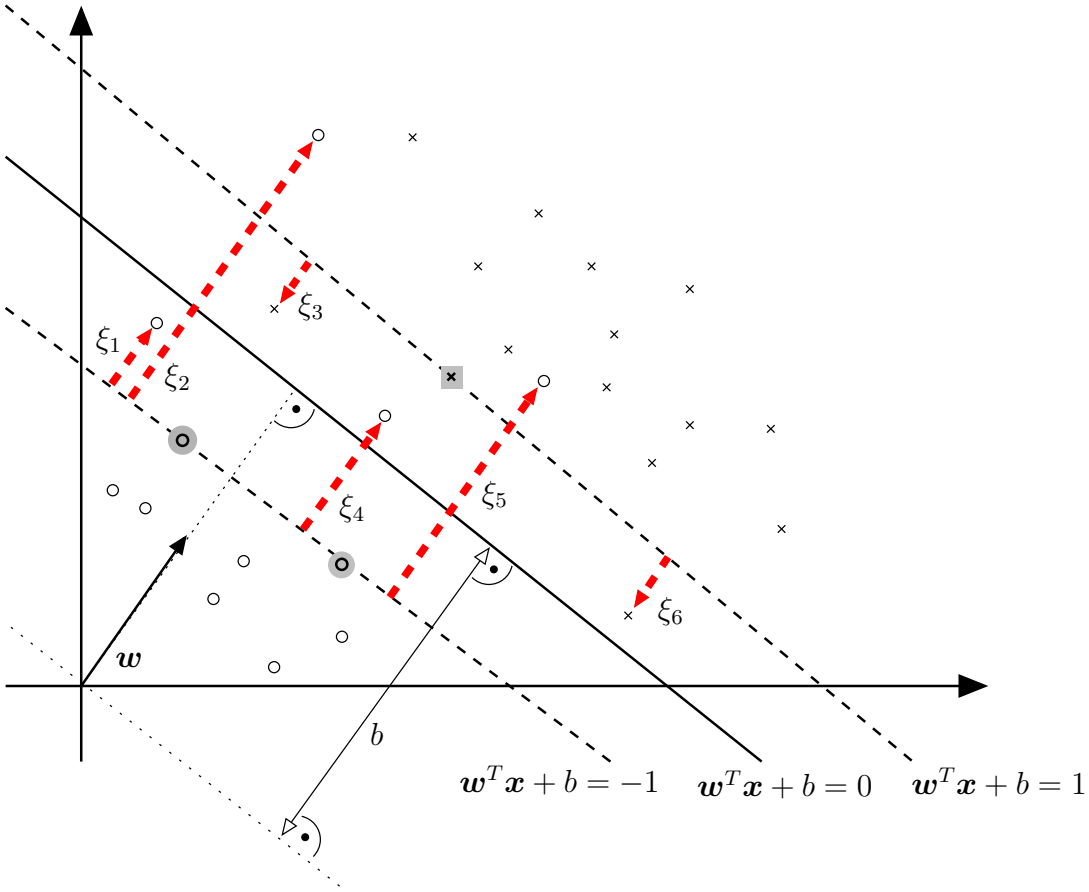


Figure 4.2: Noisy data is handled by slack variables ξ_i which allows some data points to lie within the margin or even on the "wrong" side of the hyperplane.

4.1.3 Soft Margin Hyperplane

To account for data samples that cause the data set to be non-separable, *slack variables* $\xi_i \geq 0$ were introduced changing the constraints to:

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \tag{4.12}$$

As this can introduce classification errors even on the training set, a cost function is added, to penalize for this behavior, to the function with is subject to the minimization in eqn. (4.4):

$$\min_{\boldsymbol{w},b} \tfrac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{l}\xi_i \tag{4.13}$$

$$\text{subject to} \quad \xi_i \geq 0\ ,\ y_i(\boldsymbol{x}_i^T\boldsymbol{w} + b) \geq 1 - \xi_i\ ,\ i = 1, \ldots, l \tag{4.14}$$

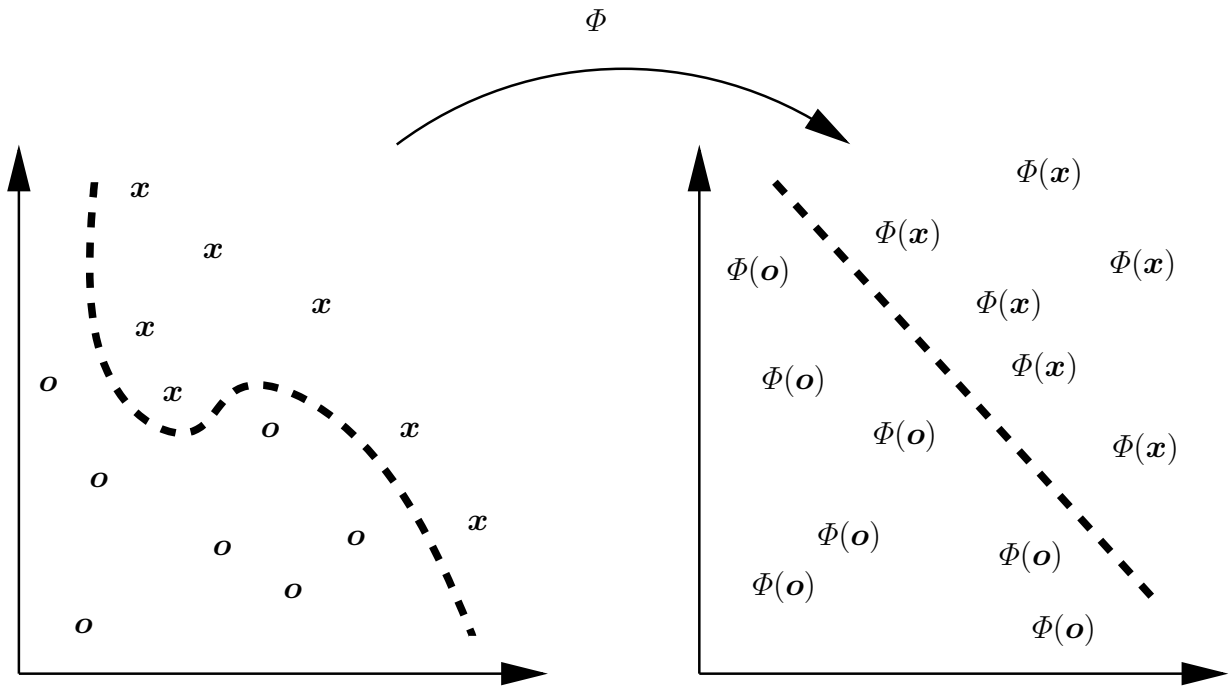


Figure 4.3: Illustration of a non-linear mapping Φ which makes the classes in the data linearly separable.

In Figure 4.2 the slack variables are illustrated. The solution to this optimization problem is obtained analogously to the linearly separable case[Vap96], [Sch01]. There is no canonical way to chose the parameter C. It has to be chosen appropriately depending on the task.

4.1.4 Kernel Trick

The kernel trick is a method to make a linear classifier more generic. The data is transformed into a *feature space* $\mathcal{H}$ by a non-linear mapping Φ which increases the separability of the data:

$$\Phi: \quad \mathbb{R}^d \longrightarrow \mathcal{H} \tag{4.15}$$

$$\boldsymbol{x} \longmapsto \Phi(\boldsymbol{x}) \tag{4.16}$$

This is illustrated in Figure 4.3. The key to the success of this approach lies within eqn. (4.11). Interestingly, the data $\boldsymbol{x}$ and $\boldsymbol{x}_i$ enter eqn. (4.11) only by their scalar product. As the function Φ might be expensive to compute or even map to an infinitely dimensional space, one is more interested in functions which perform both, the mapping and scalar product computation.

Such a function is called a *kernel* k:

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{x}_i) \tag{4.17}$$

To assure that there exists a space in which the kernel computes the scalar product implicitly, one uses kernels that satisfy the Mercer condition - so called *Mercer Kernels*, the definition of which is discussed next.

4.1.5 Mercer Kernels

Dealing with a data set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l\}$, the Mercer condition is fulfilled if the *Kernel matrix*:

$$\boldsymbol{K}_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4.18}$$

is symmetric positive semi-definite. Matrices that have only non-negative eigenvalues are positive semi-definite. For a more detailed presentation of this issue, we refer to [Cri00].

Meanwhile many kernel functions have been proposed often specialized for a certain task and incorporating a priori information about the type of the data [Wal03b], [Cha99]. But there are some commonly known kernel functions which perform well in many relevant learning tasks:

- polynomial kernel: $k(\boldsymbol{x}, \boldsymbol{x}_i) = (\boldsymbol{x}^T \boldsymbol{x}_i)^a \ , \ a \in \mathbb{N}$
- Gaussian kernel: $k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\frac{||\boldsymbol{x}-\boldsymbol{x}_i||^2}{2\sigma^2}) \ , \ \sigma \in \mathbb{R}$
- sigmoid kernel: $\tanh(\kappa \boldsymbol{x}_i^T + \Theta) \ , \ \kappa, \theta \in \mathbb{R}$

New Mercer kernels can be obtained from known Mercer kernels by composition. In [Cri00] an overview of rules is given. Given the Mercer kernels K_1 and K_2, $a \in \mathbb{R}$, a real valued function f, a mapping $\Phi : X \to \mathbb{R}^m$ with a kernel K_3 over $\mathbb{R}^m \times \mathbb{R}^m$, the following are Mercer Kernels:

$$K(\boldsymbol{x}, \boldsymbol{z}) = K_1(\boldsymbol{x}, \boldsymbol{z}) + K_2(\boldsymbol{x}, \boldsymbol{z}) \tag{4.19}$$
$$K(\boldsymbol{x}, \boldsymbol{z}) = aK_1(\boldsymbol{x}, \boldsymbol{z}) \tag{4.20}$$
$$K(\boldsymbol{x}, \boldsymbol{z}) = K_1(\boldsymbol{x}, \boldsymbol{z}) K_2(\boldsymbol{x}, \boldsymbol{z}) \tag{4.21}$$
$$K(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x}) f(\boldsymbol{z}) \tag{4.22}$$
$$K(\boldsymbol{x}, \boldsymbol{z}) = K_3(\Phi(\boldsymbol{x}), \Phi(\boldsymbol{z})) \tag{4.23}$$
$$K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^T \boldsymbol{B} \boldsymbol{z} \tag{4.24}$$

4.2 Multi-Class SVMs

For a multi-class problem we have to make a decision between K classes $\Omega_\kappa, \kappa = 1, \ldots, K$ instead of only two. As already mentioned in the introduction to this Chapter, with SVMs the extension to multiple classes is not trivial. Although so-called *all-together* methods have recently been proposed that formulate and solve the optimization problem described in Section 4.1.2 for the multi-class case [Wes98], [Cra00] are up to now not feasible for large datasets. Therefore we have to rely on heuristic extensions:

- *one-against-all* : K classifiers are trained, where the κ-th classifier discriminates between class κ and all the other classes. Instead of the sign of the distance to the hyperplane in eqn. (4.11), the distance itself is considered and a decision is made by selecting κ with maximum distance to the hyperplane.

- *one-against-one* : $\frac{K(K-1)}{2}$ classifiers are trained, which means one for all possible pairs of classes. This can be thought of as a fully connected graph with the classes as nodes. For each decision made in favour of a specific class, this class gets one vote. The class with the most votes is selected.

- *directed acyclic graph method* : As the name proposes, a directed acyclic graph is set up, in which each node represents one classifier. For making a decision the graph is traversed starting at the root and following the edges according to the decisions made in each node. The leaves of the graph are labeled according to the class to chose.

In [Hsu01] two all-together methods and the mentioned three heuristic approaches were compared. The authors concludes that one-against-one and the direct acyclic graph method are best for most problems. A nice property of these methods is also that they train more, smaller classifiers than in the one-against-all case. As each classifier is easily trained in parallel, these methods profit from a reduced granularity in terms of parallelization.

4.3 Local Kernels

The robust local feature representation described in Chapter 3 consists for each image $\boldsymbol{I}_i, i = 1, \ldots, l$ of a set of local features $\boldsymbol{L}_i = \{\boldsymbol{p}_j(\boldsymbol{I}_i)\}_{j=1}^{n_i}, \boldsymbol{l}_j(\boldsymbol{I}_i)\}_{j=1}^{n_i}$, where n_i is the number of extracted features from image $\boldsymbol{I}_i$ and $\boldsymbol{l}_j$ denotes the descriptor of the j-th feature computed at position $\boldsymbol{p}_j$. In [Wal03b] it is shown that there is no straightforward way to use local features

in SVMs. Most work on local features, for example [Sch96] and [Low99], agree on the fact that a feature matching step is required to establish feature correspondences. Based on these correspondences a decision is made.

Section 4.3.1 reviews a recently introduced kernel, which performs local feature matching in SVMs. Then Section 4.3.2 describes a metric for comparing local features and how it is used in the local kernel together with the constraints proposed in Section 4.3.3.

4.3.1 Definition

A new class of kernels for comparing local feature sets was proposed in [Wal03b]:

$$K_L(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{2}[\hat{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) + \hat{K}(\boldsymbol{L}_k, \boldsymbol{L}_h)] \tag{4.25}$$

with

$$\hat{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \text{argmax}_{j_k=1,\ldots,n_k} \{K_l(\boldsymbol{l}_{j_h}(\boldsymbol{L}_h), \boldsymbol{l}_{j_k}(\boldsymbol{L}_k))\} \tag{4.26}$$

where K_l is a Mercer kernel. The features are implicitly matched by considering feature pairs for which the kernel K_l achieves a maximum. Practical issues on the evaluation of this kernel are reported in the Appendix A.

Although the local kernel achieved excellent results in experiments [Wal03b], the Mercer condition could not be shown. Toy examples have shown that the risk of negative eigenvalues of the kernel matrix can not be excluded. However, recent empirical evaluations of this type of kernel have shown that these problems only occur for parameters that are not relevant to applications, as small number of matches or low local feature dimension [Cap03].

A problem of this kernel is that it permits multiple matches for each feature. However, the kernel may be modified to guarantee a one-to-one matching [Wal03a]. This is done by identifying the best match between two feature sets with respect to the kernel K_l and not considering these two local features for further matches. Thus, eqn. (4.25) is not needed anymore to make the kernel symmetric. This can be formulated in terms of a kernel:

$$K_{one-to-one}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{n} \max_{\Phi \in S_{n_h}, \Psi \in S_{n_k}} \sum_{j=1}^{n} K_l(\boldsymbol{l}_{\Phi(j)}(\boldsymbol{L}_h), \boldsymbol{l}_{\Psi(j)}(\boldsymbol{L}_k)) \tag{4.27}$$

where n is the number of considered matches and S_n are all permutations of possible matches.

We use the following decomposition of the kernel K_l:

$$K_l(\boldsymbol{L}_h, \boldsymbol{L}_k) = K_m(\boldsymbol{l}_{j_h}(\boldsymbol{L}_h), \boldsymbol{l}_{j_k}(\boldsymbol{L}_k))\ K_c(\boldsymbol{p}_{j_h}(\boldsymbol{L}_h), \boldsymbol{p}_{j_k}(\boldsymbol{L}_k)) \tag{4.28}$$

where K_m and K_c are Mercer kernels. According to eqn. (4.21), this operation preserves the Mercer condition. While K_m has to provide a metric to compare local features, K_c deals with position information of the features to improve the matching by introducing a constraint. We consider K_c as optional.

In the following Sections 4.3.2 and 4.3.3, several choices for K_m and K_c are given.

4.3.2 Metrics for Comparing Local Features

The choice of K_m in eqn. (4.28) depends on the type of local features that are used.

Local jets The zero-mean normalized cross-correlation is used in [Wal03b] for comparing the local jet features, which were described in Section 3.3.1. Therefore the following kernel K_m is used for local jets:

$$K_m = K_{jet} = \exp\left(-\gamma\left(1 - \frac{< \boldsymbol{x} - \boldsymbol{\mu_x} | \boldsymbol{y} - \mu_{\boldsymbol{y}} >}{||\boldsymbol{x} - \boldsymbol{\mu_x}||\ ||\boldsymbol{y} - \boldsymbol{\mu_y}||}\right)\right) \tag{4.29}$$

SIFT The L2-norm is proposed for comparing SIFT features in [Low99]. As it is shown in [Bur99] the linear kernel and the Gaussian kernel imply a L2-metric in the feature space. As the Gaussian kernel has shown favourable performance in many cases [Sch01], we decide to use the Gaussian kernel as K_m for the SIFT features. The Gaussian kernel given in Section 4.1.5.

4.3.3 Constraints

To improve the stability of the feature matching various constraints have been introduced in the literature to steer the matching process ([Wal03b], [Sch96], [Tel02], [Pri98]). But most of these methods cannot be formulated in terms of kernels or they are computationally too expensive as the kernel typically has to be evaluated several million times. Therefore we consider a constraint, which has already been used with the local kernel in [Wal03b] and two new approaches [1].

[1] This work is based on an idea of Christian Wallraven

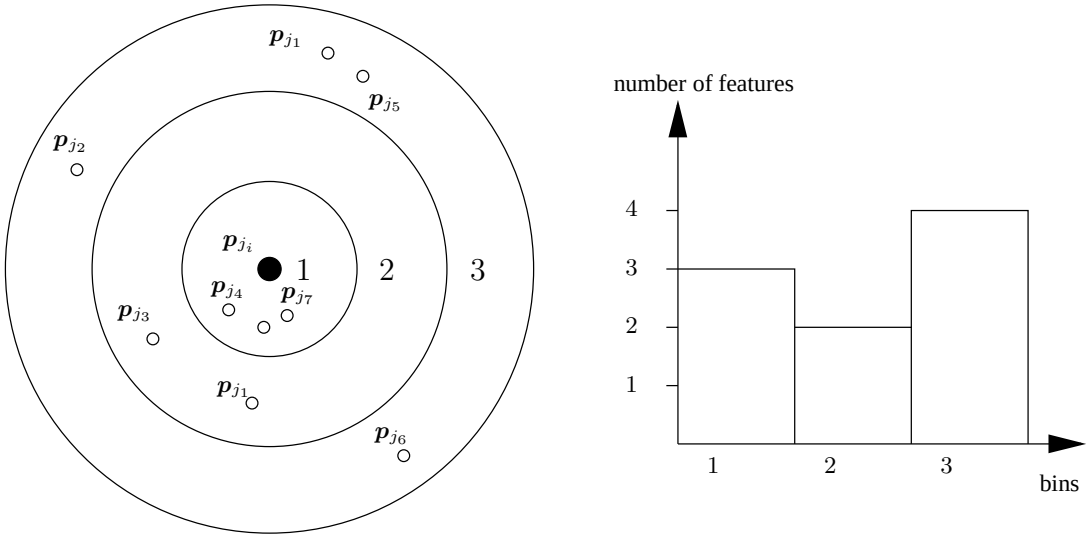


Figure 4.4: Distance histogram: all features lying in a similar distance are binned

Feature distances If the object of interest is always presented at the same position in the image, it is a reasonable assumption, that also corresponding features should occur at similar positions. In practice for example an attention mechanism driven by motion or depth can provide a segmentation of the image, so that for a selected segment the object stays in focus and the assumption holds. In [Wal03b] the following choice for K_c is proposed to exploit this a priori knowledge:

$$K_c = K_{fd}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \exp\left(-\frac{(\boldsymbol{p}_{j_h}(\boldsymbol{L}_h) - \boldsymbol{p}_{j_k}(\boldsymbol{L}_h))^2}{2\sigma^2}\right) \tag{4.30}$$

Intuitively, a Gaussian is placed at the position of a local feature. According to eqn. (4.28) the similarity computed between the two features for a possible match is modified by the value of this Gaussian with respect to the distance between the two candidates.

Distance histograms A constraint which is referred to as a *distance histogram* is motivated by the Belongie descriptor proposed in [Bel01]. The idea is to capture the distribution of features with respect to each feature by a histogram. While the original descriptor uses a histogram with respect to angle and distance, we restrict ourselves to distances, as we are dealing with much less feature points. The computation of such histograms is illustrated in Figure 4.4. One of the standard measures to compare histograms is the χ^2 distance.

$$d_{\chi^2} = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i} \tag{4.31}$$

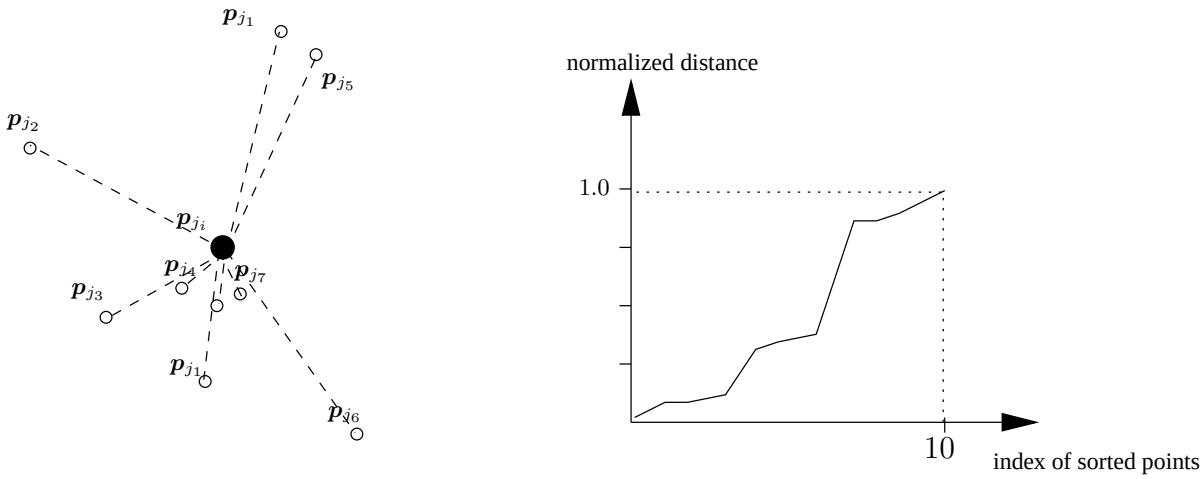


Figure 4.5: Distance profile: the distance to all features is sorted to obtain a profile shown on the right.

We refer to [Pre92] for more details and a nice discussion of this metric. In [Cha99] it is shown how to use a generalized form of the Gaussian kernel with this distance measure. Using the χ^2 in this form yields:

$$K_c = K_{dh}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \exp\left(-\sigma d_{\chi^2}(\text{hist}(\boldsymbol{p}_{j_h}(\boldsymbol{L}_h)) - \text{hist}(\boldsymbol{p}_{j_k}(\boldsymbol{L}_h)))\right), \tag{4.32}$$

where the operation hist refers to the computation of the described histogram from the feature positions. For this approach a number of bins for the histogram has to be specified.

Distance profiles *Distance profiles* are similar to the distance histograms which were introduced in the previous Section. But for distance profiles one simply computes all the distances of one feature to all the other features and sorts these distances, as shown in Figure 4.5. A reasonable way to compare two such profiles is to calculate the squared error. This suggests again the use of the Gaussian Kernel which is defined in Section 4.1.5.

4.4 Summary

This Chapter describes support vector machines which have shown excellent generalization performance in many machine learning tasks and also in computer vision. To make use of this powerful learning approach, a recently proposed class of local kernels is used. It is extended to perform one-to-one matching which is a desired property when dealing with local features. To

compare different feature types different kernels were plugged into the local kernel. The kernel is additionally extended to handle constraints based on the feature locations, which is a common method to improve the matching process. Besides a simple *feature distance* constraint, two new constraints are proposed. The *distance histogram*, which describes the distances to neighbouring features by a histogram and the *distance profile*, which is a profile of sorted feature distances. In contrast to the *feature distances* constraint, the new constraints do not make the assumption that the object always stays centered in the image.

Chapter 5

Experiments

In this Chapter we present experiments testing the methods for robust data representation and robust classification which were described in Chapter 3 and 4. Their performance is evaluated under real-world conditions. At first, in Section 5.1 the constraint introduced in Section 4.3.3 are evaluated on ground-truth data, to decide which of them should be further considered. After that we present a series of experiments on the CogVis database, to perform multi-category recognition. We conduct experiments to investigate how the approach generalizes with respect to the number of objects in the training set and how many objects are necessary for learning a good model of a category. These are presented in Section 5.2.1. Then we investigate the performance of the approach with respect to different variations which occur naturally in real-world settings:

- **scale**: controlled lab setting with homogeneous background and simulated scale (Section 5.2.2)
- **occlusion**: controlled lab setting with homogeneous background and simulated occlusion (Section 5.2.3)
- **cluttered background**: real-world setting with heterogeneous background (Section 5.2.4)

In the experiments on scale and occlusion, several methods are compared to cope with the posed challenges. The methods which have shown good performance are then applied to the real-world setting with cluttered background. All the experiments are run on extremely challenging sets of categories.

Section 5.3 presents experiments performed on the DIROKOL database. The goal of the experiments is to perform categories recognition on objects commonly found in office environments. We present experiments in homogeneous and heterogeneous background. With respect to

the experiments in cluttered scenes, we use the DIROKOL database and new images recorded at CVAP-KTH. To cope with the low number of objects present for each category in this database a learning approach is tried out to increase the performance.

For the experiments cross-validation is used, if a partitioning of the data is involved.

5.1 Experiment on Constraints

Section 4.3.3 presented three position constraints, that are extensions to the local kernel. These are (i) the *feature distances* which exploits the assumption that corresponding features will reoccur at similar positions, (ii) the *distance histogram* which compares histogram-based statistics on the feature positions and (iii) the *distance profile* which compares profiles obtained by sorted feature distances. To evaluate their usefulness, a matching experiment is performed on the "model house" image sequence [1].

In this sequence a camera is moved around a house keeping the house centered in the image. The advantage of this sequence is that besides the image data also the camera matrices are provided. Thus we use this sequence to test matching robustness with respect to viewpoint changes. Given the camera matrices we can decide if a match made by the kernel is correct or wrong. Matches are considered to be correct if they are consistent with the epipolar geometry defined by the camera matrices. For a correct match, we require the distance of each point to the epipolar line of the matched point to be less than 2 pixels. Although this condition is necessary but not sufficient, we consider it to be precise enough to gather a meaningful statistic on the matches. For details on epipolar geometry we refer to [Har00].

The three mentioned approaches are tested in combination with the metric from Section 4.3.2, varying the settings of the parameters. Besides the number of bins for the distance histogram approach, an additional parameter is considered which is called the neighbourhood. It specifies how many features are used to compute the histogram or the profile. From each of the images 156 features are extracted and only the best 50 matches are considered. The matching experiment was performed for every possible pair of images in the sequence.

In Figure 5.2 the number of correct matches averaged over all image pairs that have the same distance in the image sequence are shown. For each approach, results for the parameters which performed best with respect to subsequent frames in the sequence. This is motivated by the databases which are used in the classification experiments that also have a rather dense sampling of the viewpoints. For distance histograms and distance profiles a neighbourhood of

[1] http://www.robots.ox.ac.uk/∼vgg/data/

Figure 5.1: Samples of the house sequence which is used to evaluate the constraints. Frame $1, 3, 5, 7, 9$ of the 9 images long sequence are shown.

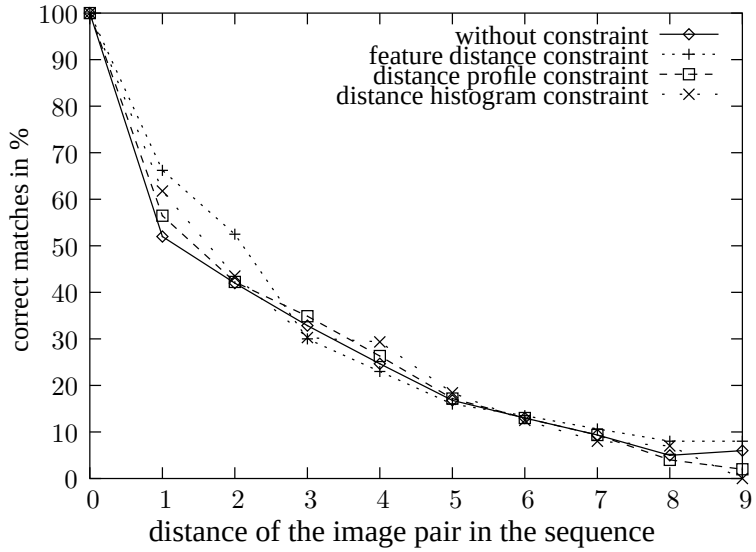


Figure 5.2: The average number of correct matches on the image sequence with respect to the distance of the image pair in the sequence.

40 features yielded the best results. The number of bins was 15. Figure 5.2 shows that for small viewpoint differences the feature distance approach yields the best results. For intermediate viewpoint differences, distance histograms and distance profiles give slight improvements, while the distance histograms once perform worse than the basic algorithm where no constraint is applied.

Overall, this experiment suggests that the feature distance approach is most suitable. However it is clear that this approach is not at all position invariant. Although we are dealing in the experiments with images in which the object is mostly centered, we will also evaluate the distance profile approach, since it should be more robust to position changes of the object and has shown competitive results with respect to the feature distances in informal recognition experiments on images with homogeneous background.

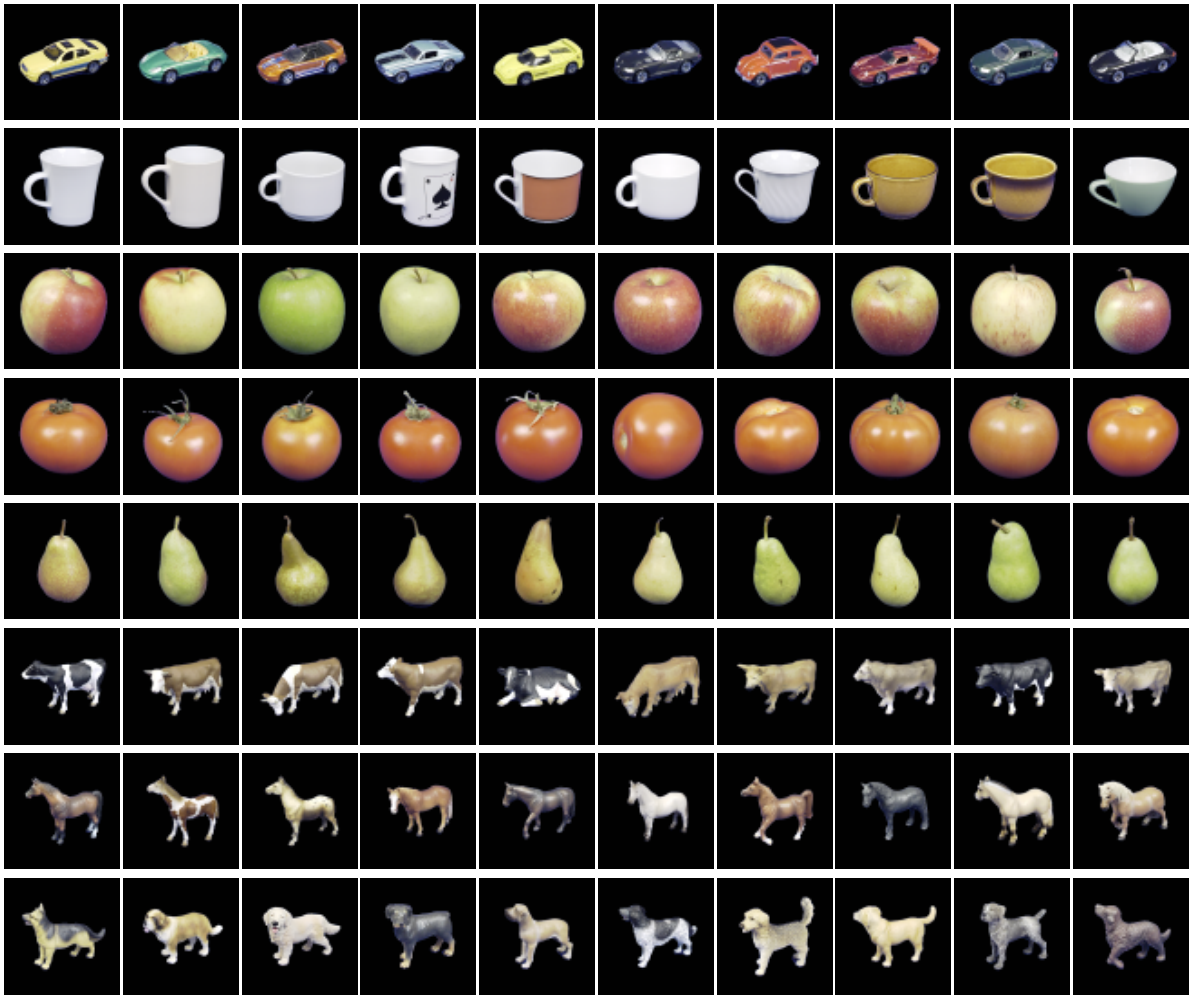


Figure 5.3: CogVis database

5.2 Experiments on the CogVis Database

The CogVis database [Lei03a] is designed to study object categorization. Therefore it contains 80 objects from 8 different categories: apple, pear, tomato, cup, car, horse, cow and dog. For each object 41 views from equidistant points on a hemisphere are taken. The object is always centered in the image, and the background is set to black by a mask which is also provided by the creators of the database. Toy objects are used for the categories car, cow, horse and dog. One view of each object is shown in Figure 5.3.

5.2.1 Experiments with Different Numbers of Objects in the Training Set

An open issue in object categorization is the dimension of the training set for each category. For example, if the task is to recognize cups, how many different instances do we have to show to the algorithm, before it is able to generalize and recognize all cups? In order to partially (and qualitatively) answer this question, we performed preliminary experiments on the categories apple, tomato, dog and horse.

We run 5 different experiments, each corresponding to 5 different partitions of training and test set. The number of objects in the training set is varied from 1 to 9. The remaining objects which are not used for training are used for test. Each object is represented by 16 views equally spaced around the object

As data representation, the multi-scale approach in combination with the local kernel with feature distance constraint as described in Section 5.2.2 is used. In Figure 5.4 the averaged error rates are reported and details are given in Table B.1. Even for a single example for each category an average error rate of less than 14% can be observed. The error rate decrease roughly linearly until 7 instances of each category are included in the training, achieving a error rate below 2%.

A surprising result of this experiment is that even for a single example for each category a good performance is achieved, especially considering the chosen categories with strong visual similarities. We observe a kind of saturation at 7 objects in the training set, where already excellent performance is achieved.

From these experiments and the more extensive experiments recently done in [Cap04], we decide to use 5 objects in the training set for further experiments, as that seems sufficient to learn a model for a category and leaves enough objects for validation and test.

5.2.2 Experiments with Variation in Scale

Variations in scale occur naturally in real-world settings, and humans are very good at handling it. Robustness to varying viewing distance and image resolution is also crucial in many tasks for artificial visual systems. Therefore we test our method for recognizing object categories at different scales. We tried different methods to compensate for the variations introduced by scale.

To make a quantitative evaluation of the performance with respect to scale, we made categorization experiments with artificially scaled images from the CogVis database. All 41 images of each object are used. The challenging categories apple, tomato, horse and dog are used, as they provide category pairs {apple, tomato} and {horse, dog} which show a wide visual similarity.

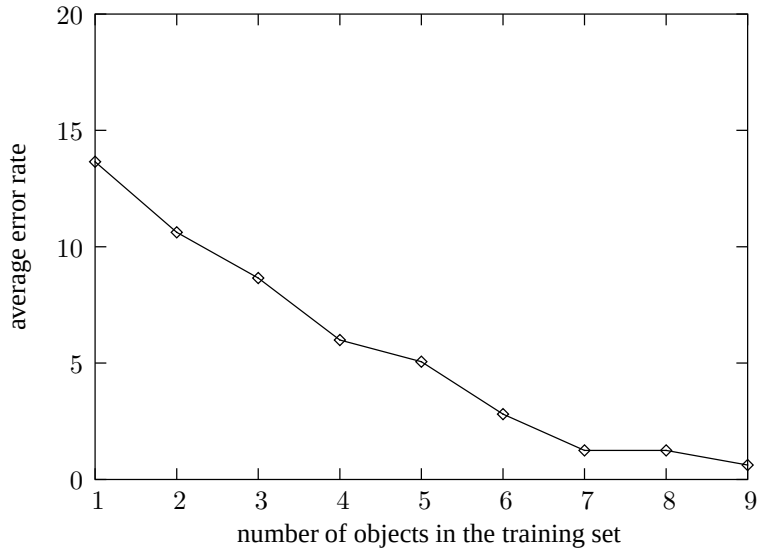


Figure 5.4: Average error rate with respect to number of examples of each category in the training set.

The following scales were considered:

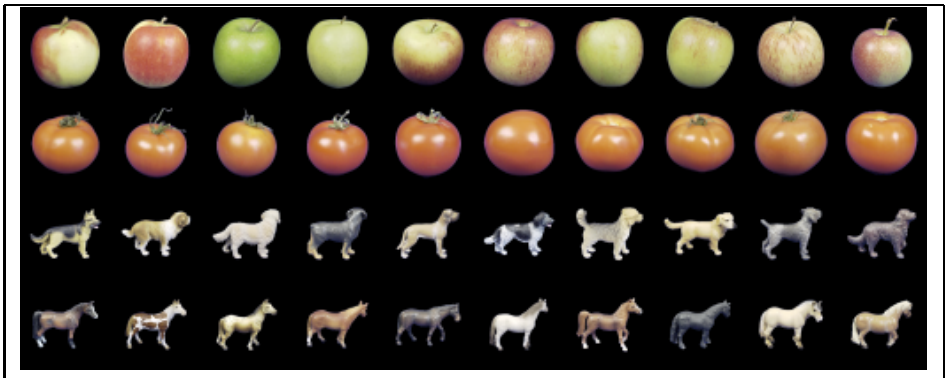
$$\sigma = 2^{-\frac{4}{8}}, 2^{-\frac{3}{8}}, 2^{-\frac{2}{8}}, 2^{-\frac{1}{8}}, 2^{0}, 2^{\frac{1}{8}}, 2^{\frac{2}{8}}, 2^{\frac{3}{8}}, 2^{\frac{4}{8}}, \tag{5.1}$$

where $\frac{1}{\sigma} size_{orig}$ is the size of the scaled image with $size_{orig}$ being the size of the original image. We run 5 experiments on 5 different partitions. Training, validation and test set are described in details in Figure 5.5.

As described in Chapter3, scale can be handled by the data representation. Therefore, we evaluate the following three approaches:

- *multi-scale* (Section 3.2.3): The Harris detector (Section 3.2.2) is used to detect interest points at multiple scales to capture the different appearance of features at different scales. To reduce the computational complexity the number of features is thresholded. In addition, the images are down-sampled to 128^2, as otherwise too many features would be required to represent the image at these fine scales properly. 78 features are extracted by selecting the strongest n_σ features at each scale with respect to the Harris interest function eqn. (3.17). The used scales specified by their σ value and thresholds for these scales are shown in Table 5.2.2. Finally, the local jet descriptor from Section 3.3.1 is used to describe the features.

- *scale selection* (Section 3.2.4): Another way for dealing with scale is to compute a scale



obj1	obj2	obj3	obj4	obj5	obj6	obj7	obj8	obj9	obj10
train1					vali1		test1		
test2			train2				vali2		test2
vali3	test3			train3					vali3
train4	vali4		test4			train4			
train5			vali5		test5			train5	

Figure 5.5: Partitioning of the datasets for scale experiments.

scales σ:	2^0	$2^{0.5}$	2^1	$2^{1.5}$	2^2	$2^{2.5}$	2^3
thresholds n_σ:	25	18	11	9	6	5	4

Table 5.1: From each scale the n_σ strongest features with respect to the Harris interest function are used. The scales and these thresholds n_σ are shown.

invariant representation by applying automatic scale selection (Section 3.2.4). The iterative scheme of the Harris-Laplace detector is used to detect scale invariant interest points, starting with the interest points detected for the multi-scale approach as an initialization. Again, local jets are used to describe the local features.

- *SIFT* (Section 3.2.4): SIFT features are used as their interest point detector also performs scale selection. The implementation of David Lowe is used [2].

All these approaches have been successfully applied to object recognition. However their applicability to categories is unclear. To our knowledge, this is the first systematic experimental evaluation of the performance of these descriptors for categories recognition.

For classification a SVM with the local kernel eqn. (4.27) is used. For the experiments involving local jets, we use the correlation based similarity measure eqn. (4.29) with and without the feature distances constraint eqn. (4.30). For the SIFT features, the Gaussian kernel is used for measuring similarity in the local kernel, as proposed in Section 4.3.2. The SVM parameter C

[2] available at http://www.cs.ubc.ca/∼lowe/keypoints/

overall recognition rate		
	without constraint	with constraint
multi-scale	$90.02\%\pm4.26$	$88.82\%\pm6.09$
scale selection	$87.01\%\pm6.23$	$88.57\%\pm6.27$
vali scales	$89.59\%\pm4.50$	$90.39\%\pm4.67$
train scales	$88.33\%\pm6.95$	$91.44\%\pm4.75$
SIFT	$67.36\%\pm10.00$	$-$

Table 5.2: Overall recognition rates of the experiments on scale. The feature distance constraint is considered optionally.

is set to 100 while the parameters γ for the local kernel and σ for the constraint were determined during the training on the validation set.

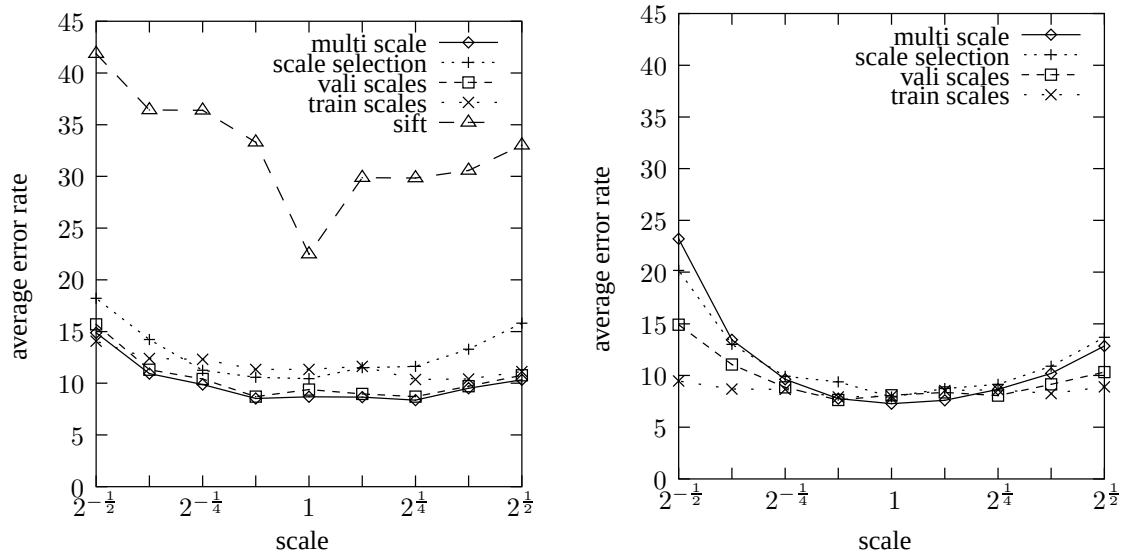
In addition to the three approaches for dealing with scale in the *data representation*, we also test two *pure learning-based* methods . The idea is to learn the variations due to scale by using scaled data in the training set. The use of this method has already been shown in [Hay04].

- *vali scale*: To influence the model selection not only the validation images, but also scaled versions of them are used for model selection. For this purpose the coarsest, the original and the finest scale are used: $\sigma = 2^{-\frac{4}{8}}, 2^0, 2^{\frac{4}{8}}$.

- *train scale*: As an extension to the *vali scale* approach, the *train scale* approach includes scaled versions of the original images in the training and the validation set. Again the scales: $\sigma = 2^{-\frac{4}{8}}, 2^0, 2^{\frac{4}{8}}$ are used.

The results with respect to variations in scale are reported in Figure 5.6. From left to right the scale of the test images are varied from the nearest/finest scale $2^{-\frac{1}{2}}$ to the farest/coarsest scale $2^{\frac{1}{2}}$. Scale 1 specifies the scale of images in the database. We report error rates. The standard deviations are given in Table B.2 and Table B.3. The overall recognition rates are given in Table 5.2. Besides the SIFT features, all of the approaches achieve an average error rate of approximately 10% for the reference scale $\sigma = 1$. Changing the scale the error rates increase between 2% for the *train scale* method with constraint and 16% for the *multi-scale* method with constraint. Again the SIFT features show worse results with an increase of nearly 20%. Interestingly, the increase in error for experiments with the *feature distance constraint* is stronger when testing on coarser scales than on finer scales.

Concerning the data representation approach to handling scale variations the multi-scale approach without position constraint gives the best results. The constraint does not help in this case as the basic assumption that features can be detected in a close neighbourhood is violated in the

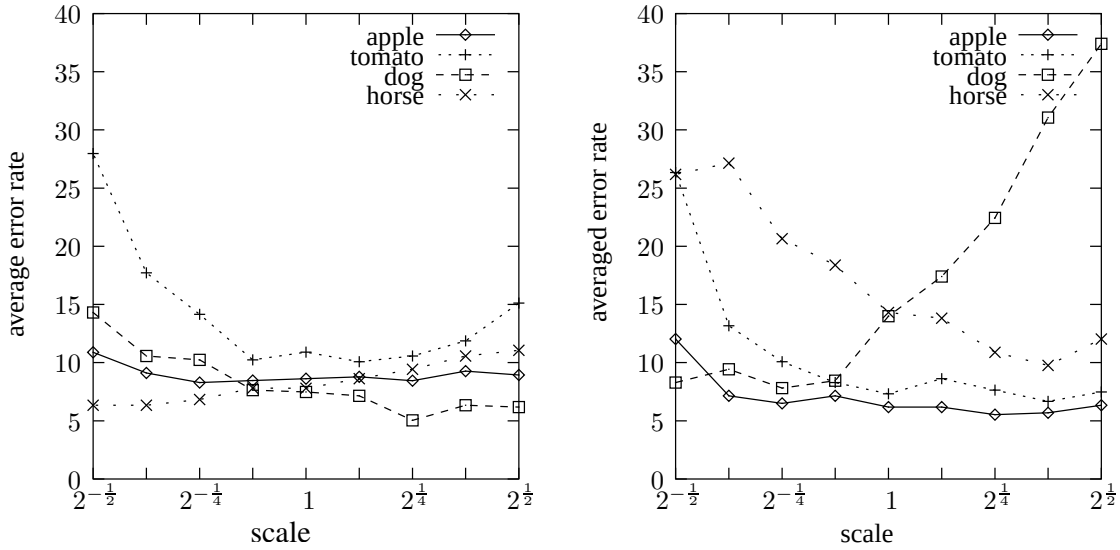


(a) without constraint

(b) with constraint

Figure 5.6: Average error rates of experiment with variation in scale. The feature distance constraint is considered.

case of more severe scale changes. This is also supported by the observation, that the error rate increases faster for coarser scales than for finer scale. The reason for this is that the change in the size of the object is more dramatic when moving to coarser scales. Therefore also the position of the local feature change more, which violates the assumptions for the *feature distance* approach. That the methods based on scale selection do not perform favourably has two reasons. First we have to note that especially the objects apple and tomato are very weakly textured. Therefore they also lack of a sufficient amount of structure that could be detected at a characteristic scale. In the case of the tomato, the SIFT interest point detector only detects 6 features for some views. Although the multi-scale approach does not detect strong interest points either, it at least performs some kind of sampling of the object. Another issue is that scale can also be an important cue. For example, in the chosen set of categories, a dog and a horse can easily be confused if one does not account for the scale. This is also supported by the experiments. Figure 5.7 compares multi-scale approach, which captures scale, and the scale selection approach, which is designed to be scale-invariant. Error rates with respect to the categories are shown. The risk that a horse can be confused with a dog is amplified by changing to the scale invariant representation. Therefore we conclude that scale can provide essential information in distinguishing objects that are visually similar, but have different size or proportions.



(a) multi-scale without constrain

(b) scale selection without constraint

Figure 5.7: Average error rates of experiment with variation in scale for all categories.

With respect to the learning approaches, the *train scale* approach with the *feature distance* constraint performs best. As scaled data is in the training set, the assumption that features reoccur at similar positions seems to hold again, so that we can take advantage of the feature distance constraint. However, this comes at the price of a increased training set which results in increased storage requirements and training time.

In conclusion, the *train scale* approach is the most effective for handling scale among the tested methods. Concerning the data representation approach, the simple multi-scale method performs best. Given the difficult set of categories, with easy to confused pairs {tomato, apple} and {horse, dog}, the results on these images with homogeneous backgrounds are excellent.

5.2.3 Experiments with Occlusion

Another common problem in real-world settings is occlusion. Therefore the robustness of the approach with respect to the loss of information due to partial occlusion is investigated. Occlusion is simulated by successively removing features from left to right according to their position in the image until the desired amount of occlusion is achieved. In contrast to the experiments with scale, we use the categories car, cup, cow, horse and dog and move the first object from the test set to the validation set. The rest of the setup stays the same. This is done to be consistent with the experiments in clutter, as we see these two topics related. In both cases, either decreasing the

number of features or adding distracting features, the method must be able to rely on matched subsets.

For data representation, we use the multi-scale approach with the corresponding local kernel with and without feature distance constraint as described in Section 5.2.2. To investigate the dependence of the performance on the number of categories, we ran four experiments:

- exp1 3 categories: car, cup, cow
- exp2 4 categories: car, cup, cow, horse
- exp3 5 categories: car, cup, cow, horse, dog
- exp4 3 categories: cow, horse, dog

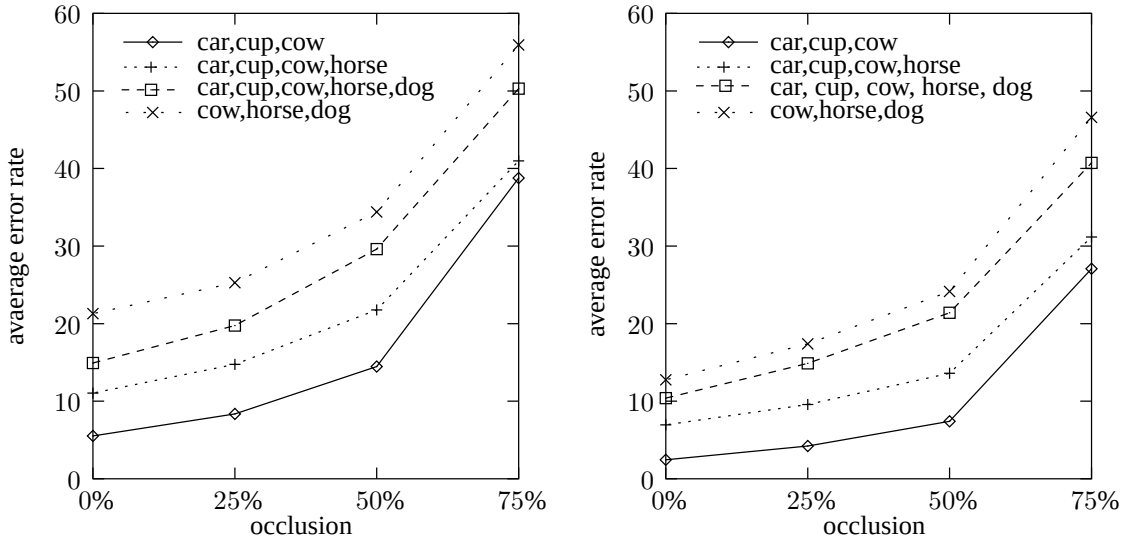
From exp1 to exp3 the number of categories increases and exp4 is used to show how difficult the added categories are. Exp4 consists of 3 animal that show a wide visual similarity.

Results are presented in Figure 5.8. We report error rates. Detailed results for all partitions are reported in Table B.4 and B.5.

Especially for the case, where the feature distance constraint is used, the performance drops only slowly with respect to increased occlusion. Therefore we consider the approach to be robust with respect to occlusion of less than 50%. With respect to the categories, we observe a rather sever loss of performance, when increasing the number of categories. However, this must be considered in the context that the added categories are very difficult to distinguish. This can be seen from the results on exp4, that is in terms of error rates very close to exp3 which includes two more categories than exp4. We conclude that the stability with respect to the number of categories strongly depends on the added categories.

5.2.4 Experiments with Clutter

To evaluate the approach with respect to distractions caused by clutter, we learn models for the categories car, cup, cow, dog and horse on the CogVis database which provides images in homogeneous background and test them on real-world images in heterogeneous background. It has to be noted that for the categories horse, cow and dog, the CogVis database provides images of toy objects, while the images in the test set are pictures of real animals. In Figure 5.9 two examples of each category are presented. The categories are considered to be very challenging due to their visual similarity. Especially the three animals, cow, dog and horse are supposed to be very tricky, as they also consist of similar parts.



(a) without constraint

(b) with constraint

Figure 5.8: Average error rates of experiment with occlusion

The real-world images of cars and cows were obtained from Bastian Leibe [Lei04]. The horse image were download from the homepage of Eran Borenstein [3]. The images of the other two categories dog and cup, are new contributions. The images of the cups were recorded at CVAP-NADA in an office environment. The cups are shown from the side and are approximately centered in the image. The scale can also be considered to be roughly the same for all images. The lighting conditions were uncontrolled. The dog images obtained were partly by a web search. Already a google search gave some results, but large parts were obtained from websites of animal shelters and private pages dedicated to the topic dog. We selected images where a dog is in the center of the image and covers a reasonable size of the image. Other conditions like pose and light are uncontrolled. We use 100 images of each category car, horse, cow, dog and cup. For the car and cow images a region around the object was selected by hand to ensure that the object is centered and approximately at the same scale.

The setup of the datasets is the same as for the occlusion experiment in Section 5.2.3 with two exceptions. Only 16 views from equidistant viewpoints are used for each object and the test set is replaced by the cluttered images.

To cope with these real-world problems, different approach are tried, which have shown favourable performance in the previous experiments. At first a multi-scale approach as described

[3] http://www.wisdom.weizmann.ac.il/∼boren/

Figure 5.9: Samples of the images gathered for experiments in cluttered background.

in Section 5.2.2 is used as a baseline. To compensate for features that are distracted from the object by the background, we increase the number of extracted features from the cluttered views to 156 by raising the thresholds for each scale:

scales:	2^0	$2^{0.5}$	2^1	$2^{1.5}$	2^2	$2^{2.5}$	2^3
thresholds for 156 features:	50	36	22	18	12	10	8

Table 5.3: From each scale the n strongest features are used. The scales and these thresholds n are shown for the case of cluttered views.

As there are still some scale variations present in the images, we try to improve the results by applying the train scale approach, which performed well in Section 5.2.2. From a learning point of view, we try to improve the model selection by splitting the set of cluttered images in two, to use one half for validation and the other half for testing. All these approaches are tried with and without the feature distance constraint. However this constraint relies on strong assumptions about the position of the object. Therefore an additional experiment is performed with the distance profile constraint which we expect to have a much broader applicability.

In Figure 5.4 the overall recognition rates of the experiments are given. An additional experiment using the *distance profile* constraint is reported in Figure 5.5. In the Appendix additional results are given in Table B.6 to B.11. Training times for one model, test times for one image and parameters are given in the Appendix in Table B.12.

We consider the multi-scale experiment as a baseline. The recognition rate varies from 31.6% for five categories to 51.0% for three categories. The use of the *feature distance* constrain gives a consistent improvement of approximately 10% in recognition rate. For this baseline exper-

approach	experiment	without constraint	with feature distances
multi-scale (baseline)	1	$51.00\% \pm 5.34$	$60.47\% \pm 13.06$
	2	$42.30\% \pm 6.43$	$54.45\% \pm 4.99$
	3	$31.64\% \pm 4.76$	$47.68\% \pm 4.47$
	4	$37.00\% \pm 4.39$	$46.00\% \pm 7.32$
validation on clutter	1	$37.73\% \pm 5.96$	$70.67\% \pm 6.91$
	2	$29.10\% \pm 5.72$	$62.40\% \pm 1.52$
	3	$18.08\% \pm 6.85$	$55.44\% \pm 4.98$
train scales	1	$52.40\% \pm 2.42$	$67.67\% \pm 14.54$
	2	$39.10\% \pm 4.67$	$60.40\% \pm 8.09$
	3	$30.64\% \pm 3.47$	$46.44\% \pm 6.01$

Table 5.4: Overall recognition rate of experiments in cluttered background. The feature distance constraint is considered optionally.

iment, we also conducted exp4, which uses the categories cow, horse and dog. It has to be remarked, that for the case with position constraint the recognition rate for this three category experiment is even below the five category experiment exp3. While we get very poor results for the validation on cluttered views without the position constraint, the experiments with the position constraint benefit from an additional increase of approximately 10% with respect to the baseline experiments with position constraint. The last approach in this line of experiments is the *train scale* method which has performed very well on the scale experiments. Also in experiments with clutter improvements of approximately 6% can be observed. Only exp3 does not show the improvement.

Considering the extremely difficult sets of categories and the challenging real-world conditions it is not surprising, that only reasonable performance is achieved. The best results are achieved by using cluttered views in the validation set, which yields recognition rates between 70.67% for the categories car, cup, cow and 55.44% for the categories car, cup, cow, horse, dog. For the baseline setup, a category set 4 with the three animals was used, to show how difficult it is to distinguish between the categories which are successively added to the experiment. It turns out that the performance drops approximately to the level of set 3 with 5 categories. Also the train scales method results in a good improvement which suggests that this method of compensating for scale also works in cluttered background. Comparing the results without and with the feature distance constraint, one can observe considerable improvements of about 10%. This underlines the importance of an appropriate constraint, as it is able to compensate for the increased distraction by the background. Also it shows how generic the issue of scale is in such a real-world setting. Although all the objects are approximately at the same size, an appropriate approach to

approach	set	with distance profiles
multi-scale	1	$53.47\% \pm 2.02$
	2	$43.90\% \pm 1.67$
	3	$32.72\% \pm 4.79$

Table 5.5: Overall recognition rate of experiments in cluttered background with distance profile constraint.

handle scale achieves significant improvements.

An additional experiment is performed using the *distance profile* constraint. The results are reported in Table 5.5. The improvements with respect to the baseline without constraint vary between 1% for exp3 and 2.5% for exp1. These stay far behind the improvements achieved with the *feature distance* constraint.

Even though the performance in this real-world setting cannot compete with the performance of *feature distance* constraint, this has to be seen in relation to the requirements. The *feature distance* makes strong assumptions on the feature positions and the experiments have shown, that their violation have serious effects. The *distance profile* makes less assumptions and should be at least to some extend position invariant by construction. Therefore it remains an interesting alternative to the feature distance constraint.

5.3 Experiments on the DIROKOL Database

The DIROKOL image database [Rei01] consists of 13 objects from the office and health care domain: cola can grey, cola can red, puncher1, puncher2, stapler1, stapler2, cup, cup with plate, fork, spoon, knife, NaCl bottle and medicine box. The objects are shown in Figure 5.10. For each object the database contains 3720 images in homogeneous black background on a hemisphere above the object. The sampling is non-uniform. 3 different lighting settings are used which change from view to view. In addition a second hypersphere with an offset was recorded, but we do not use this data in out experiments . To simulate cluttered background, the database includes 1860 images of each object, where the foreground object was automatically segmented out from the homogeneous background and pasted into new, cluttered images, as illustrated in Figure 5.10. The images have a resolution of 256^2.

Preliminary object classification experiments on the cluttered views of this database yielded results far worse than expected. In object recognition experiments results between 36.57% for all 13 objects and 76.56% for 3 objects (can, stapler, puncher) are achieved using the feature distance constraint. We had anticipated better results since due to the construction of the database, the

objects are perfectly centered, so that the *feature distance* constraint should be able to eliminate most of the wrong matches.

The reason why the recognition rates are so low is that the automatic segmentation procedure evidently performed somewhat poorly on some of the images. Too much of the foreground is removed, implying that the object contains "holes" through which the background is visible. Typical images containing these artifacts are shown in Figure 5.11. The final example in Figure 5.11 illustrates another problem with the automatic extracted foreground masks. Here not all of the black homogeneous background is rejected, inducing "speckles" of black in the new images. These artifacts give rise to a number of spurious detections of features.

As this caused problems especially with the local feature approach, we decided to record new images in cluttered background and to not further consider the heterogeneous views of the DIROKOL database.

5.3.1 Extension to the Database

The DIROKOL database was extended by images in cluttered background for each of the categories can, cup, puncher and stapler. These objects were selected, as they are commonly found in office environments and therefore relevant to robotic tasks in such environments. With a view to classifying object categories, we imaged 10 different cans, 3 staplers, 2 punchers and 5 cups. The views were taken in 11 different background settings. Some examples are shown in Figure 5.12. The object is always centered and approximately at the same scale. The lighting is not controlled. The objects are all recorded from approximately the same viewpoint which has an elevation angle of 25° with respect to the ground. All in all we added 57 images of cans, 56 of cups, 55 of staplers and 55 of punchers.

5.3.2 Experiments in Real-World Settings

For the real-world experiments on the new views in cluttered background, the data representation and the local kernel are the same as for the baseline experiment on the CogVis database described in Section 5.2.3. For training, 121 views of the DIROKOL database were used, covering viewing angles of $25^\circ \pm 15^\circ$. The mapping from the DIROKOL classes to categories is described in Table 5.6.

We performed experiments on two sets. In the first, object 1 is used for training and object 2 is used for validation and in the second experiment object 2 is used for training and object 1 is used for validation. As validation on cluttered views does not seem possible due to the limited

	can	cup	stapler	puncher
object 1	ColaGrau	TasseD	HefterGruen	LocherGruen
object 2	ColaRot	TasseTeller	HefterWG	LocherRot

Table 5.6: Mapping of the DIROKOL objects to categories.

number of views, we tried a different approach to improve model selection. As mentioned earlier, occlusion and clutter can be seen as related topics as in both cases the classification has to rely on subsets of the features. This suggests the use of occlusion in the validation set for an experiment on cluttered data. The occlusion is simulated in the same way as in Section 5.2.3.

In Figure 5.13 the results are shown. For 0% occlusion the standard approach is reported, while the other data points represent results using occlusion in the validation set. Without occlusion in the validation set, the best results were obtained with the feature distance constraint. For set 1 we achieved 61.36% and 46.82%. For set 2 with constraint, increasing the occlusion in the validation set increased also the performance on clutter. However, for set 1 with constraint exactly the opposite can be observed.

The error rates are quite high and the occlusion approach does not show any systematic improvements. Given the challenging cluttered views and only one training example, the error rates can be considered as reasonable. We conclude that we lack sufficient data to train a good model for a category.

5.4 Summary

In this Chapter our approach for object category recognition was evaluated with respect to real-word settings. We addressed the problem visual similar categories and problems like scale, occlusion and clutter. This was done to an extend which it has not been done, to our knowledge.

In the experiments, a pure learning approach to handling scale has shown very good results. Due to the nature of local feature, we observed a good robustness with respect to occlusion up to 50%. Although the real-world experiments with cluttered background have proven very challenging, especially due to the visual similar categories, we obtained reasonable results.

A general observation in the experiment is that many assumptions from standard object recognition approaches do not hold. This causes techniques, which have reported excellent results in object recognition to perform poorly in the categorization context. As an example, we want to mention the scale selection mechanism. Although we experienced more generic problems with weakly textured objects, in particular visual similar categories like horse and dog caused prob-

lems. In this case, the relative scales of the local features are an important cue. Another issue is related to the assumptions made for local feature. In all the experiments the matching without constraints performed quite poorly and the use of constraints yielded significantly better results than without. Therefore we consider the use of constraints for matching local features in the category context as important, as the assumption of visual similarity between corresponding features or object parts is often violated.

We have to note, that the performance of the system is very dependent on the choice of the categories. This makes the comparison of systems for recognizing object categories a very hard task, as it is clear that visual similar categories cause serious problems.

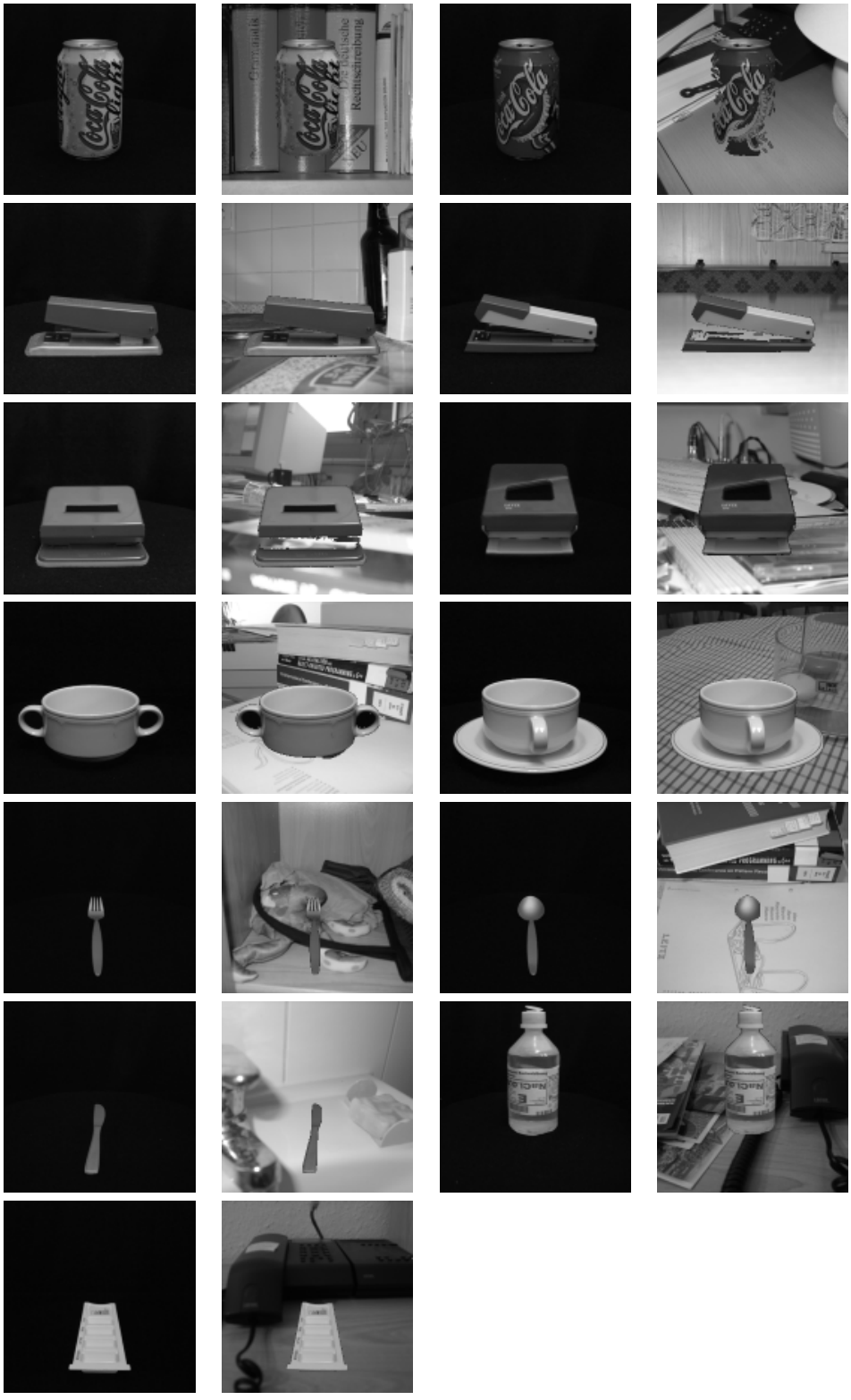


Figure 5.10: Objects in the DIROKOL database. First and third column: objects in homogeneous black background. Second and fourth column: objects in heterogeneous background.

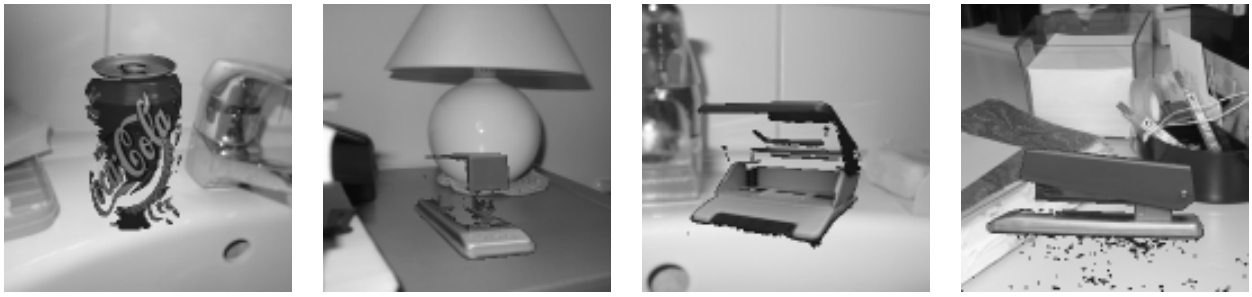


Figure 5.11: Problems on the DIROKOL database due to automatic segmentation.

Figure 5.12: Examples of the recorded image of office object.

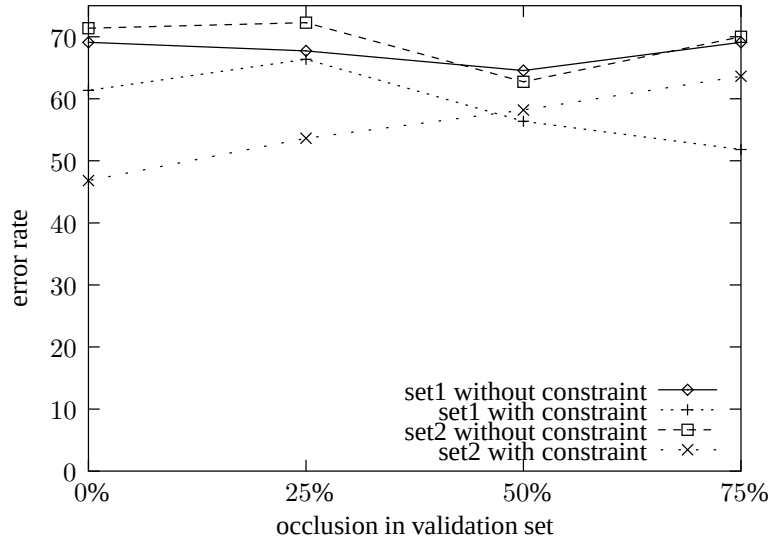


Figure 5.13: Results for the cluttered views with training on DIROKOL with an attempt to improve recognition rates by validating on occluded views.

Chapter 6

Summary

Humans deal naturally with categories, the ways we perceive our environment and formulate our thoughts are based on the concept of categories. With ease we generalize from a specific cup to the category of all cups, although their appearances can be quite diverse.

Although categories are connected to our daily life, their definition is still a matter of debate. As stated in [Lak87], recent approaches argue that categories are far more complex than was initially thought and that they are tied to the person who perceives them. However, for time being, in computer vision we have to stay with the classical definition that describe categories as large classes, but with still exact defined borders. This we refer to as the recognition of object categories.

To set up a system which is capable of recognizing object categories, two main requirements are identified. The first is a robust representation, which can describe what is common to members of a category and what separates them from other categories. The second is a method for robust classification, which is able to build a model for categories taking into account the large variability within a category. To be of use in real-world settings, both steps need to be robust with respect to noise, illumination changes, clutter and occlusion.

In this thesis, state-of-the-art local features are used to fulfill the requirement of a robust representation. We use and compare the multi-scale approach proposed in [Sch96], its extension with the scale invariant Harris-Laplace detector in [Mik02b] and the SIFT features introduced in [Low99].

For robust classification, we use Support Vector Machines (SVMs) [Vap96], which have shown excellent performance in computer vision tasks [Cha99], [Roo01], [Hay04]. Only recently their range of application was extended to local feature data via a new class of kernels in [Wal03b]. All experiments presented in this thesis are performed in this framework. For this

purpose, the new kernel was added[1] to the SVM software library LIBSVM [Cha01].

Most methods for recognition based on local features use a feature matching step to establish feature correspondences. To make this step more reliable, many different constraints were introduced [Sch96], [Tel02], [Pri98]. As the local kernel also performs an implicit matching, this thesis investigated different methods for setting constraints in the matching process. The first constraint is a simple method based on feature distances, suggested in [Wal03b]. In addition, two new approaches[2], namely the distance histogram and the distance profile constraint were evaluated and added to LIBSVM. The performance of these three constraints in a matching experiment on ground truth data and recognition experiments, leads to the conclusion that for images where the object is always shown at the same position in the image, the simple approach based on feature distances gives the best results. One of the new approaches, called distance profiles, yielded a small improvement in a recognition experiment in cluttered background. Although the gain is considerably less than with the simple distance constraint, the new constraint does not make as strong assumptions on the position of the object. Therefore it is an interesting alternative.

To investigate the generalization capability of the approach, a qualitative experiment was performed varying the number of examples of each considered category. Surprisingly good performance is achieved with only a few examples.

To evaluate the approach with respect to real-world conditions typical variation like scale, occlusion and cluttered background are tested individually on the CogVis database and finally in a more uncontrolled setting on real-world images.

The scale experiments are performed on artificially scaled images. To cope with this variation we applied different methods related to the representation and the training step. For the representation that handles scale explicitly, we used a multi-scale approach like in [Sch96], its extension to scale invariance by the Harris-Laplace detector from [Mik02b] and the SIFT features introduced in [Low99]. From the learning point of view, we included scaled images in the validation set and in a second approach also in the training set, as it was done for material classification in [Hay04]. Surprisingly, the scale invariant methods like the one with the Harris-Laplace detector and SIFT did not show the favourable results typically reported for object recognition [Mik02b],[Low99]. We identified two reasons. First, some weakly textured objects violate assumptions on the underlying automatic scale selection process and second, for distinguishing between categories like horses and dogs the overall size of the object and relative scales between features seem to be important cues. This hypothesis is supported by the fact that the method us-

[1] with the generous support of Christian Wallraven

[2] this work is based on an idea of Christian Wallraven

ing the Harris-Laplace detector for scale selection significantly increased the confusion between horses and dogs. The method with training on scales has shown the most stable results with respect to scale.

An other important issue in real-world settings is occlusion. In the experiments occlusion is simulated by successively eliminating features from left to right in the image. Our approach gives good results if the occlusion is less than 50%.

For experiments with cluttered background, images were gathered for the categories car, cup, cow, horse and dog. While the images for cars, cows and horses are gathered from existing collections, the images for dogs and cups are new contributions. The training is performed on the CogVis database which contains only toy objects for cows, horses and dogs. For a baseline experiment, applying a multi-scale representation yields a 60.47% overall recognition rate for a 3 category experiment which drops to 47.68% for a 5 category experiment. This has to be seen in relation to the successively added categories of horses and dogs. Further experiments have shown that these categories are extremely hard to distinguish due to their visual similarities of object parts. Therefore we conclude that how the method scales with the number of categories is strongly related to the actual categories and cannot be answered in general.

Similar experiments on objects in cluttered background were performed with training on the DIROKOL database. Therefore images of cans, punchers, staplers and cups were recorded in an office environment. The best observed recognition rate is 53.18%. An approach to improve the model selection by including occluded views in the validation set did not show any systematic improvements. We conclude that the database does not provide enough different objects of a category to learn a good model.

Another important observation is that the use of constraint matching showed significant improvements in many experiments, especially in presence of cluttered background. Therefore we conclude that constraints are a suitable method for compensation for the distraction caused by the background in the matching process.

However, much work must still be done to even approach the capabilities of humans with regards to recognizing object categories. From the point of view of data representation, the state-of-the-art approach is still to apply methods from object recognition, in the hope that they will show robustness with respect to the increased variations. An extensive study of the use of different descriptions for categories has not been done, but could be of great use to understand categorization better.

Another important issue is to build databases on which experiments on categorization can be performed. For object recognition many databases exist, but they do not offer the variety of

objects required for categorization.

As was shown in this thesis, an approach based solely on local features relies strongly on the process which finds corresponding features in two images. This matching problem is by no means solved. One approach to this problem are constraints as they are used in this thesis. Another approach is to use more cues to make the representation more distinctive. Especially categories like tomatoes which have shown problems due to the lack of features might benefit from a different cue like for example colour. Therefore some categorization problems could possibly simplify dramatically by using the right cue like color, shape, texture or even combining them by an appropriate scheme.

Appendix A

Optimization of the Local Kernel

As described in Section 4.3 a kernel which can handel local features should perform some kind of feature matching. In the case of the local kernel $K_{one-to-one}$ eqn. (4.27) one-to-one matching is used. As mentioned before, this is done by computing unique matches between the local feature of two local features sets L_h and L_k which maximize the sum of the kernel values K_l. In practice this is done by computing a similarity matrix $\boldsymbol{S}$ where $\boldsymbol{S}_{ij}$ is the kernel K_l evaluated for the i-th feature of L_h and the j-th feature of L_k, with $\boldsymbol{S} \in \mathbb{R}^{n_h \times n_k}$, given that L_h consists of n_h and L_k consists of n_k features. The straightforward idea to find the matches is to search for a maximum in $\boldsymbol{S}$ and take this as the first match. Then the row and the column of this maximum are not further considered by setting them to a very low value (poissening). After that the next maximum is searched, until the desired number of matches has been found. As each maximum search has a computational complexity of order $\mathcal{O}(n_h n_k)$ and has to be done for the desired number of matches n, the whole procedure of computing the matches is of order $\mathcal{O}(n_h n_k n)$.

A significant speed-up could be achieved by computing the matches with a different algorithm [1]. In the following we will denote the smaller feature set with L_h. The basic idea is to derive an algorithm from the properties of a match. We consider the feature pair corresponding to the entry $\boldsymbol{S}_{ij}$ in the similarity matrix to be a match, if there is no match which gives a higher value for the kernel K_l with repect to the unmatched features. This is the case when $\boldsymbol{S}_{ij}$ is the best match for the the i-th feature of set L_h:

$$\underset{b=1,\ldots,n_k}{\operatorname{argmax}} S_{ib} = j \tag{A.1}$$

[1] idea of Christian Schuldt

and at the same time best match for the j-th feature of set L_k:

$$\underset{a=1,\ldots,n_h}{\operatorname{argmax}} S_{aj} = i \tag{A.2}$$

This can be assured by checking if $\boldsymbol{S}_{ij}$ is maximum with respect to its column and row.

Therefore the improved algorithm searches row by row for a maximum. If a maximum is found, it is checked if the entry is a maximum for the column, too. If this is the case, a match is found and this column and row are eliminated by setting them to a low number (poissening). This is repeated until $\min(n_h, n_k)$ matches were found. From these matches the n matches which result in the highest kernel value are selected. A diagram of the algorithm is given in Figure A.1.

Fast maximum search — fast computation of one-to-one feature matches in a similarity matrix

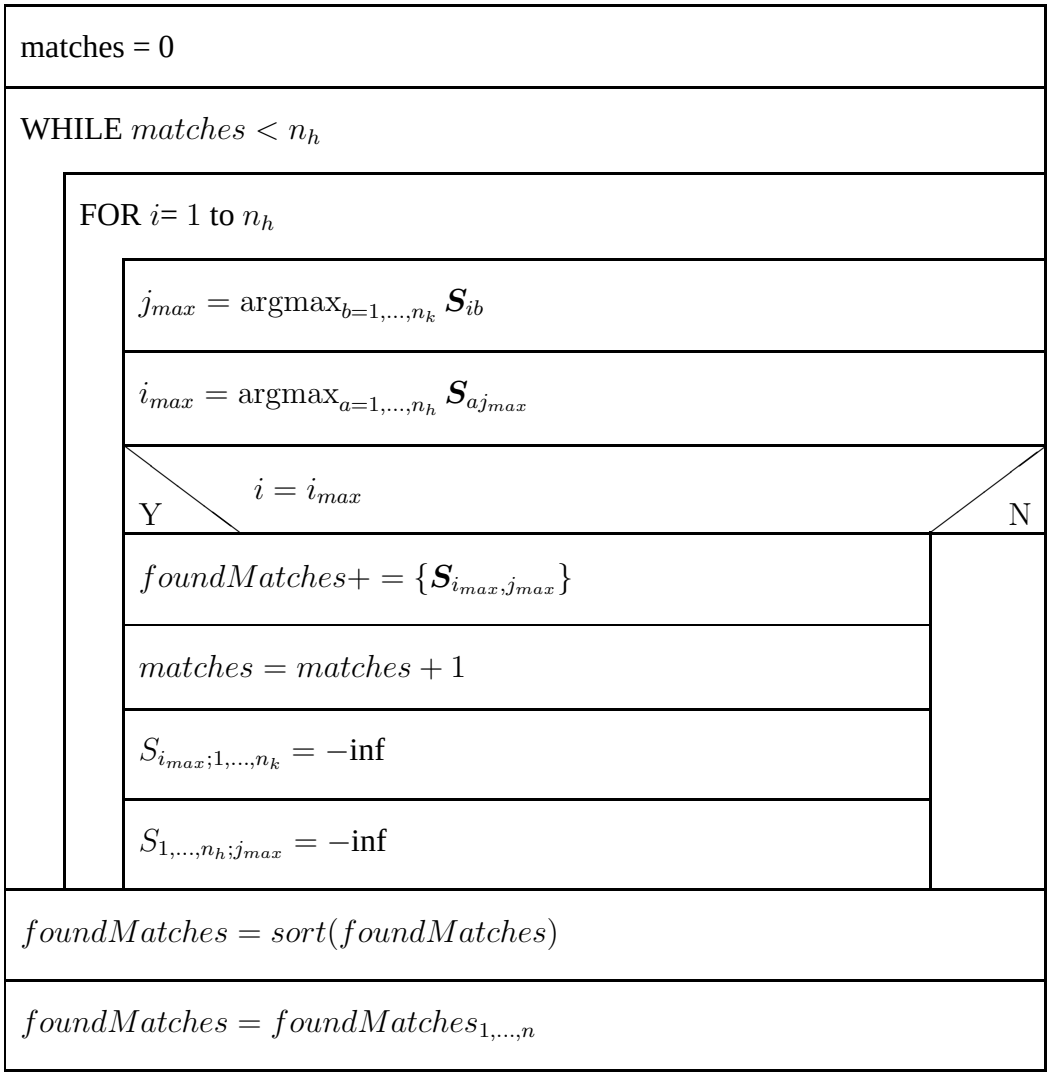


Figure A.1: Algorithm to significantly speed up the one-to-one feature matching used in the local kernel, assuming that L_h is the local feature set with less or equal number of features than L_k

Appendix B

Detailed Experimental Results

B.1 Experiments with Different Number of Objects in the Training Set

objects in train	recognition rates for each partition					average recognition rate
1	89.41%	88.72%	81.08%	91.15%	81.42%	86.35% ±4.74
2	91.02%	86.72%	91.60%	95.12%	82.42%	89.38% ±4.90
3	89.51%	85.71%	98.21%	96.21%	87.05%	91.34% ±5.57
4	90.36%	92.45%	98.18%	91.93%	97.14%	94.01% ±3.44
5	84.38%	97.81%	98.44%	95.31%	98.75%	94.94% ±6.06
6	91.80%	96.88%	99.22%	99.22%	98.83%	97.19% ±3.17
7	98.96%	97.40%	100.00%	100.00%	97.40%	98.75% ±1.31
8	97.66%	98.44%	100.00%	100.00%	97.66%	98.75% ±1.18
9	98.44%	100.00%	100.00%	100.00%	98.44%	99.38% ±0.86

Table B.1: Recognition rate for each partition, varying the number of objects in the training set.

B.2 Experiments with Scale

scale	multi scale	scale selection	vali scales	train scales	sift
$2^{-\frac{1}{2}}$	14.88 % ±4.59	18.21 % ±7.01	15.69 % ±6.22	14.06 % ±7.71	41.87 % ±5.84
$2^{-\frac{3}{8}}$	10.93 % ±4.48	14.23 % ±6.42	11.30 % ±4.65	12.40 % ±8.42	36.42 % ±7.46
$2^{-\frac{1}{4}}$	9.88 % ±4.36	11.26 % ±6.20	10.41 % ±4.31	12.32 % ±8.22	36.38 % ±7.78
$2^{-\frac{1}{8}}$	8.54 % ±2.92	10.57 % ±5.53	8.70 % ±2.95	11.34 % ±7.48	33.30 % ±11.19
2^{0}	8.70 % ±3.42	10.45 % ±5.84	9.39 % ±3.63	11.34 % ±7.25	22.48 % ±7.87
$2^{\frac{1}{8}}$	8.66 % ±3.78	11.50 % ±5.67	8.97 % ±3.76	11.63 % ±7.31	29.88 % ±8.36
$2^{\frac{1}{4}}$	8.37 % ±4.07	11.62 % ±6.01	8.70 % ±3.91	10.37 % ±7.56	29.84 % ±10.83
$2^{\frac{3}{8}}$	9.51 % ±4.15	13.29 % ±6.50	9.72 % ±3.84	10.41 % ±6.52	30.57 % ±11.55
$2^{\frac{1}{2}}$	10.33 % ±5.47	15.81 % ±7.22	10.77 % ±5.47	11.14 % ±7.64	33.01 % ±12.23

Table B.2: Error rates for all methods and 9 scales, averaged over 5 partitions; no constraint is used.

scale	multi scale	scale selection	vali scales	train scales
$2^{-\frac{1}{2}}$	23.21 % ±6.58	20.16 % ±5.95	14.92 % ±6.27	9.47 % ±4.69
$2^{-\frac{3}{8}}$	13.45 % ±2.35	13.01 % ±5.84	11.06 % ±4.90	8.70 % ±4.40
$2^{-\frac{1}{4}}$	9.63 % ±3.38	9.92 % ±6.32	8.86 % ±4.52	8.66 % ±4.90
$2^{-\frac{1}{8}}$	7.76 % ±4.00	9.39 % ±5.79	7.64 % ±3.57	7.93 % ±4.66
2^{0}	7.28 % ±4.28	7.93 % ±4.80	8.09 % ±4.01	8.01 % ±5.12
$2^{\frac{1}{8}}$	7.60 % ±3.85	8.74 % ±5.26	8.37 % ±3.90	8.54 % ±5.37
$2^{\frac{1}{4}}$	8.66 % ±4.00	9.15 % ±5.23	8.05 % ±4.32	8.62 % ±5.82
$2^{\frac{3}{8}}$	10.20 % ±3.77	10.89 % ±5.70	9.15 % ±4.04	8.25 % ±5.64
$2^{\frac{1}{2}}$	12.85 % ±3.95	13.70 % ±5.86	10.33 % ±4.98	8.90 % ±6.19

Table B.3: Error rates for all methods and 9 scales, averaged over 5 partitions; feature distance constraint is used.

B.3 Experiments with Occlusion

set	occlusion	recognition rates for each partition					average recognition rate
1	0%	91.46 %	98.37 %	94.72 %	89.84 %	97.97 %	94.47 % ± 3.81
	25%	84.96 %	97.56 %	93.09 %	85.37 %	97.15 %	91.63 % ± 6.16
	50%	83.74 %	93.90 %	83.74 %	77.64 %	88.62 %	85.53 % ± 6.09
	75%	77.24 %	78.86 %	37.80 %	62.20 %	50.00 %	61.22 % ± 17.63
2	0%	84.76 %	92.68 %	93.90 %	89.94 %	83.54 %	88.96 % ± 4.65
	25%	76.52 %	89.63 %	93.60 %	84.15 %	82.32 %	85.24 % ± 6.61
	50%	69.21 %	84.45 %	84.76 %	71.65 %	81.10 %	78.23 % ± 7.32
	75%	60.67 %	67.38 %	60.37 %	52.13 %	54.57 %	59.02 % ± 5.95
3	0%	81.95 %	87.80 %	90.24 %	83.17 %	82.20 %	85.07 % ± 3.74
	25%	73.66 %	83.41 %	85.61 %	78.29 %	80.24 %	80.24 % ± 4.64
	50%	63.17 %	75.61 %	73.90 %	63.66 %	75.61 %	70.39 % ± 6.41
	75%	49.02 %	54.88 %	48.05 %	43.90 %	52.68 %	49.71 % ± 4.26
4	0%	80.08 %	81.30 %	89.02 %	71.14 %	71.95 %	78.70 % ± 7.38
	25%	73.58 %	76.02 %	84.96 %	69.92 %	69.11 %	74.72 % ± 6.37
	50%	60.98 %	65.85 %	75.61 %	60.57 %	65.04 %	65.61 % ± 6.07
	75%	40.24 %	45.53 %	44.31 %	45.53 %	44.72 %	44.07 % ± 2.20

Table B.4: recognition for all 4 sets of categories and different levels of occlusion. The recognition rates on all 5 partitions and the average recognition rate is reported. No constraint is used.

set	occlusion	recognition rates for each partition					average recognition rate
1	0%	97.97 %	99.59 %	100.0 %	90.65 %	99.59 %	97.56 % ± 3.94
	25%	94.72 %	99.19 %	95.12 %	91.06 %	98.78 %	95.77 % ± 3.33
	50%	90.65 %	98.37 %	91.46 %	87.40 %	95.12 %	92.60 % ± 4.24
	75%	73.98 %	85.77 %	77.24 %	72.76 %	54.88 %	72.93 % ± 11.30
2	0%	93.29 %	97.87 %	96.95 %	88.41 %	88.72 %	93.05 % ± 4.44
	25%	85.98 %	96.65 %	94.82 %	87.80 %	86.89 %	90.43 % ± 4.93
	50%	78.35 %	93.29 %	90.85 %	83.84 %	85.67 %	86.40 % ± 5.90
	75%	67.99 %	71.65 %	71.95 %	68.60 %	64.02 %	68.84 % ± 3.22
3	0%	86.10 %	95.85 %	92.68 %	85.37 %	88.05 %	89.61 % ± 4.51
	25%	75.85 %	92.20 %	89.27 %	81.46 %	86.83 %	85.12 % ± 6.51
	50%	66.10 %	86.83 %	82.20 %	73.66 %	84.15 %	78.59 % ± 8.55
	75%	48.05 %	63.41 %	63.66 %	56.59 %	64.63 %	59.27 % ± 7.04
4	0%	86.18 %	93.09 %	91.46 %	84.96 %	80.49 %	87.24 % ± 5.10
	25%	80.08 %	87.80 %	87.80 %	80.08 %	77.24 %	82.60 % ± 4.89
	50%	70.73 %	80.49 %	79.67 %	72.76 %	75.61 %	75.85 % ± 4.24
	75%	49.59 %	60.16 %	58.13 %	46.75 %	52.44 %	53.41 % ± 5.65

Table B.5: recognition for all 4 sets of categories and different levels of occlusion. The recognition rates on all 5 partitions and the average recognition rate is reported. The feature distance constraint is used.

B.4 Experiments with Clutter

	car	cup	cow
car	58.40%±13.89	0.00%±0.00	41.60%±13.89
cup	40.40%±16.48	6.00%±5.10	53.60%±15.92
cow	52.80%±23.72	0.00%±0.00	47.20%±23.72

	car	cup	cow	horse
car	63.40%±13.28	0.20%±0.40	25.60%±15.81	10.80%±9.20
cup	39.60%±9.85	10.40%±7.34	32.60%±12.37	17.40%±6.80
cow	58.60%±28.39	0.00%±0.00	31.20%±23.18	10.20%±5.91
horse	24.40%±12.37	0.00%±0.00	27.80%±11.65	47.80%±2.04

	car	cup	cow	horse	dog
car	28.80%±8.23	0.20%±0.40	28.60%±18.75	8.60%±7.12	33.80%±19.55
cup	15.00%±5.10	24.40%±9.71	23.40%±12.66	12.40%±6.09	24.80%±14.37
cow	4.00%±2.83	0.00%±0.00	32.40%±22.89	16.00%±7.16	47.60%±29.76
horse	6.40%±2.73	1.40%±1.85	26.40%±12.37	39.00%±6.23	26.80%±15.14
dog	2.80%±2.71	0.80%±1.60	40.40%±13.99	22.40%±3.01	33.60%±11.93

	cow	horse	dog
car	35.00%±23.86	18.40%±6.80	46.60%±28.42
cup	29.60%±13.71	41.20%±7.65	29.20%±14.11
cow	42.60%±14.07	22.60%±3.32	34.80%±11.29

Table B.6: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background. No constraint is used.

	car	cup	cow
car	79.80%±10.78	1.60%±3.20	18.60%±11.13
cup	26.60%±28.64	31.00%±30.23	42.40%±26.65
cow	36.40%±32.65	7.80%±15.60	55.80%±30.98

	car	cup	cow	horse
car	76.00%±17.15	0.00%±0.00	16.60%±14.39	7.40%±4.96
cup	11.40%±8.59	26.40%±16.12	14.40%±13.18	47.80%±6.46
cow	41.60%±31.31	0.00%±0.00	21.40%±23.06	37.00%±27.58
horse	9.40%±6.77	0.00%±0.00	8.00%±8.81	82.60%±10.25

	car	cup	cow	horse	dog
car	74.60%±15.70	0.00%±0.00	15.20%±10.32	9.20%±7.93	1.00%±0.89
cup	6.40%±2.42	35.60%±10.19	10.40%±10.40	30.60%±15.45	17.00%±17.61
cow	23.80%±17.74	0.00%±0.00	32.60%±24.19	28.20%±16.12	15.40%±18.52
horse	8.20%±4.79	0.40%±0.80	7.80%±7.03	70.00%±13.56	13.60%±12.21
dog	5.00%±2.83	0.20%±0.40	10.20%±8.23	59.00%±14.59	25.60%±16.40

	cow	horse	dog
car	37.80%±20.76	28.00%±17.40	34.20%±35.85
cup	11.80%±13.53	71.40%±17.22	16.80%±12.98
cow	18.60%±18.83	52.60%±21.62	28.80%±16.25

Table B.7: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background. The feature distance constraint is used.

	car	cup	cow
car	43.60%±18.06	2.00%±2.10	54.40%±20.10
cup	21.40%±10.63	35.40%±6.05	43.20%±12.66
cow	20.60%±23.85	1.20%±1.47	78.20%±25.31

	car	cup	cow	horse
car	38.00%±5.83	0.00%±0.00	39.00%±23.69	23.00%±19.12
cup	17.40%±2.87	25.80%±9.06	34.20%±14.11	22.60%±13.12
cow	8.40%±6.95	0.00%±0.00	33.80%±25.23	57.80%±20.62
horse	10.20%±4.62	1.00%±0.63	30.00%±15.56	58.80%±12.54

	car	cup	cow	horse	dog
car	27.00%±6.81	0.00%±0.00	27.20%±21.02	9.20%±8.61	36.60%±20.11
cup	14.60%±5.54	23.80%±3.43	21.20%±13.29	12.00%±5.02	28.40%±13.54
cow	4.40%±5.08	0.00%±0.00	24.40%±23.36	21.80%±8.13	49.40%±27.35
horse	7.40%±4.22	1.00%±0.63	23.40%±15.63	42.80%±4.40	25.40%±14.35
dog	2.40%±3.83	0.00%±0.00	35.00%±15.02	27.40%±5.46	35.20%±12.48

Table B.8: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background using the train scales method to compensate for scale changes. No constraint is used.

	car	cup	cow
car	83.40%±7.74	0.60%±1.20	16.00%±8.17
cup	18.20%±18.17	55.00%±21.95	26.80%±17.46
cow	32.40%±37.61	3.00%±4.10	64.60%±38.08

	car	cup	cow	horse
car	81.40%±11.93	0.00%±0.00	14.00%±9.96	4.60%±2.87
cup	12.80%±8.06	47.20%±19.65	19.40%±18.23	20.60%±3.38
cow	27.80%±33.85	1.40%±1.96	48.60%±37.82	22.20%±12.48
horse	5.40%±2.58	2.40%±2.94	27.80%±17.08	64.40%±14.44

	car	cup	cow	horse	dog
car	79.40%±12.82	0.00%±0.00	14.80%±11.75	3.80%±2.23	2.00%±1.67
cup	14.60%±7.23	37.20%±11.75	16.40%±12.08	20.40%±8.91	11.40%±12.08
cow	23.20%±17.63	0.20%±0.40	33.60%±26.88	20.00%±13.64	23.00%±26.82
horse	11.80%±9.68	1.00%±2.00	22.40%±14.85	55.20%±7.68	9.60%±9.39
dog	5.80%±3.76	0.80%±0.75	22.00%±12.95	44.60%±9.00	26.80%±17.17

Table B.9: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background using the train scales method to compensate for scale changes. The feature distance constraint is used.

	car	cup	cow
car	20.00%±33.08	2.40%±4.80	77.60%±37.87
cup	11.20%±14.29	13.20%±22.51	75.60%±36.23
cow	6.00%±12.00	14.00%±28.00	80.00%±40.00

	car	cup	cow	horse
car	26.00%±37.42	0.00%±0.00	64.40%±32.58	9.60%±7.94
cup	27.60%±36.97	0.80%±1.60	62.80%±32.31	8.80%±5.15
cow	20.00%±40.00	0.00%±0.00	64.40%±33.12	15.60%±10.98
horse	20.80%±38.67	0.00%±0.00	54.00%±28.59	25.20%±14.01

	car	cup	cow	horse	dog
car	40.40%±48.67	0.00%±0.00	9.60%±10.91	0.40%±0.80	49.60%±41.09
cup	36.00%±44.25	4.00%±6.20	11.20%±13.72	2.40%±3.88	46.40%±38.52
cow	39.60%±48.50	0.40%±0.80	8.80%±8.45	3.60%±5.43	47.60%±39.89
horse	38.40%±47.10	1.60%±3.20	20.40%±19.86	9.60%±14.39	30.00%±25.11
dog	39.60%±48.50	0.40%±0.80	25.20%±27.76	7.20%±9.68	27.60%±25.25

Table B.10: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background. Validation is done on cluttered views. No constraint is used.

	car	cup	cow
car	70.40%±8.33	2.80%±2.04	26.80%±9.52
cup	6.40%±5.43	73.20%±7.00	20.40%±7.31
cow	19.20%±20.96	12.40%±12.92	68.40%±25.18

	car	cup	cow	horse
car	60.80%±9.68	3.60%±3.44	23.20%±5.46	12.40%±4.27
cup	4.80%±0.98	63.60%±8.80	3.20%±3.92	28.40%±4.08
cow	14.40%±9.41	8.80%±4.83	62.80%±15.52	14.00%±9.21
horse	6.00%±2.19	14.00%±6.32	17.60%±4.27	62.40%±8.71

	car	cup	cow	horse	dog
car	66.40%±9.50	1.60%±1.50	23.20%±8.06	8.00%±4.00	0.80%±0.98
cup	5.60%±0.80	52.80%±7.33	4.80%±0.98	13.60%±10.98	23.20%±7.44
cow	13.20%±9.68	8.00%±3.10	62.00%±16.49	8.40%±6.74	8.40%±5.99
horse	5.60%±1.50	8.80%±2.04	17.60%±4.08	56.80%±4.66	11.20%±8.26
dog	5.20%±1.60	2.40%±2.33	12.00%±1.26	41.20%±5.15	39.20%±8.06

Table B.11: Averaged confusion matrices over 5 partitions for the experiments on the CogVis database with testing images in cluttered background. Validation is done on cluttered views. The feature distance constraint is used.

exp	feature distance constraint	average train time for one model	average test time for one image	parameter γ	parameter σ
1	without	3 min	1.6 s	10	-
	with	3 min	1.6 s	10	25
2	without	5 min	3.0 s	15	-
	with	6 min	3.0 s	10	20
3	without	10 min	3.6 s	50	-
	with	12 min	4.0 s	10	25

Table B.12: Parameters, training time for one model given a certain parameter set, testing time for one image for the real-world experiments with training on CogVis. If the feature distance constraint is used, the parameter σ is specified. Time measured on a SunBlade 100 with an UltraSPARC II processor with 400Mhz

List of Figures

List of Tables

Bibliography

[Aga02] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of European Conference on Computer Vision*, page IV: 113 ff., 2002.

[Bel01] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proceedings of International Conference on Computer Vision*, pages I: 454–461, Vancouver, BC, 2001.

[Bri04] Encyclopædia Britannica. Category, 2004.

[Bur99] Christopher J. C. Burges. *Geometry and invariance in kernel based methods*. MIT Press, 1999.

[Cap03] B. Caputo, C. Wallraven, O. Chapelle, and B. Schoelkopf. Private communication, 2003.

[Cap04] B. Caputo, C. Wallraven, and M.E. Nilsbacken. Object categorization via local kernels. In *Proceedings of International Conference on Pattern Recognition*, Cambridge, UK, 2004. to appear.

[Cha99] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, May 1999.

[Cha01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Cra00] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *Computational Learing Theory*, pages 35–46, 2000.

[Cri00] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, 2000.

[Cro82] J.L. Crowley. *A Representation for Visual Information*. PhD thesis, Carnegie-Mellon University, Robotics Institute, Pittsburgh, Pennsylvania, 1982.

[Dud01] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

[Ed28] W. D. Ed. *The Works of Aristotle, Volume 1: Logic*. Oxford University Press, 1928.

[Fer03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of Computer Vision and Pattern Recognition*, pages II: 264–271, Madison, WI, 2003.

[FF03] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of International Conference on Computer Vision*, pages 1134–1141, Nice, France, 2003.

[For97] D.A. Forsyth and M.M. Fleck. Body plans. In *Proceedings of Computer Vision and Pattern Recognition*, pages 678–683, San Jaun, PR, 1997.

[For03] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2003.

[Fre91] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[Har88] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conference*, pages 147–151, Manchester, 1988.

[Har00] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK, 2000.

[Hay04] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. In *Proceedings of European Conference on Computer Vision*, Prague, Czech Republic, 2004. to appear.

[Her01] Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2001.

[Hsu01] C. Hsu and C. Lin. A comparison of methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

[Koe84] J.J. Koenderink. The structure of images. *BioCyber*, 50:363–370, 1984.

[Koe87] J J Koenderink and A J van Doom. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987.

[Lak87] George Lakoff. *Women, Fire, and Dangerous Things: What categories reveal about the mind*. The University of Chicago Press, Chicago, 1987.

[Lap04] I. Laptev. Private communication, 2004.

[Lei03a] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object-categorization. In *Proceedings of Computer Vision and Pattern Recognition*, pages II: 409–415, Madison, WI, 2003.

[Lei03b] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *Proceedings of British Machine Vision Conference*, Norwich, UK, 2003.

[Lei04] B. Leibe, A. Leonardis, and B. Schiele. Combining object categorization and segmentation with an implicit shape model. In *Proceedings of European Conference on Computer Vision*, Prague, Czech Republic, 2004. Workshop on Statistical Learning in Computer Vision, to appear.

[Lin93] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, December 1993.

[Lin98] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, November 1998.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, 1999.

[Mac02] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[Mik01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of International Conference on Computer Vision*, pages I: 525–531, Vancouver, BC, 2001.

[Mik02a] K. Mikolajczyk. *Detection of Local Features Invariant to Affine Transformation*. PhD thesis, Institute National Polytechique de Grenoble, Grenoble, 2002.

[Mik02b] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of European Conference on Computer Vision*, page I: 128 ff., Copenhagen, Denmark, 2002.

[Mik03] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages II: 257–263, Madison, WI, 2003.

[Mur95] H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[Nel98] R.C. Nelson and A. Selinger. A cubist approach to object recognition. In *Proceedings of International Conference on Computer Vision*, pages 614–621, Bombay, India, 1998.

[Pau97] D. Paulus and J. Hornegger. *Pattern Recognition of Images and Speech in C++*. Advanced Studies in Computer Science. Vieweg, Braunschweig, 1997.

[Pre92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[Pri98] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proceedings of International Conference on Computer Vision*, pages 754–760, Bombay, India, 1998.

[Rei01] M. Reinhold, Ch. Drexler, and H. Niemann. Image Database for 3-D Object Recognition. Technical Report LME-TR-2001-02, computer science 5 (pattern recognition department), University of Erlangen-Nuremberg, may 2001.

[Roo01] D. Roobaert, M. Zillich, and J.O. Eklundh. A pure learning approach to background-invariant object recognition using pedagogical support vector learning. In *Proceedings of Computer Vision and Pattern Recognition*, pages II:351–357, Kauai, HI, 2001.

[Ros88] E. Rosch. Principles of categorization. In A. Collins and E. E. Smith, editors, *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pages 312–322. Kaufmann, San Mateo, CA, 1988.

[Sch96] C. Schmid and R. Mohr. Combining grey value invariants with local constraints for object recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 872–877, San Francisco, CA, 1996.

[Sch97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[Sch98] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *Proceedings of International Conference on Computer Vision*, pages 230–235, Bombay, India, 1998.

[Sch00a] B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, January 2000.

[Sch00b] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Proceedings of International Journal of Computer Vision*, 37(2):151–172, June 2000.

[Sch00c] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proceedings of Computer Vision and Pattern Recognition*, pages I: 746–751, Hilton Head, SC, 2000.

[Sch01] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[Tel02] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proceedings of European Conference on Computer Vision*, page I: 68 ff., Copenhagen, Denmark, 2002.

[Vap96] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1996.

[Wal03a] C. Wallraven. Private communication, 2003.

[Wal03b] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of International Conference on Computer Vision*, pages 257–264, Nice, France, 2003.

[Web00a] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proceedings of Computer Vision and Pattern Recognition*, pages II: 101–108, Hilton Head, SC, 2000.

[Web00b] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of European Conference on Computer Vision*, pages I: 18–32, Dublin, Ireland, 2000.

[Wes98] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, 1998.

[Wit83] A.P. Witkin. Scale-space filtering. In *Proceedings of the 8th Int. Joint Conference of Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.

[You87] R.A. Young. The gaussian dervative model for spatial vision. *Spatial Vision*, 2:273–293, 1987.