DISS. ETH NO. 15752

Interleaved Object Categorization and Segmentation

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of
Doctor of Technical Sciences

presented by

BASTIAN LEIBE

Dipl. Inform., M. Sc.

born 23^{rd} of April, 1975
citizen of
Germany

accepted on the recommendation of

Prof. Dr. Bernt Schiele, examiner
Prof. Dr. Andrew Zisserman, co-examiner

2004

Abstract

This thesis is concerned with the problem of visual object categorization, that is of recognizing unseen-before objects, localizing them in cluttered real-world images, and assigning the correct category label. This capability is one of the core competencies of the human visual system. Yet, computer vision systems are still far from reaching a comparable level of performance. Moreover, computer vision research has in the past mainly focused on the simpler and more specific problem of identifying known objects under novel viewing conditions.

The visual categorization problem is closely linked to the task of figure-ground segmentation, that is of dividing the image into an object and a non-object part. Historically, figure-ground segmentation has often been seen as an important and even necessary preprocessing step for object recognition. However, purely bottom-up approaches have so far been unable to yield segmentations of sufficient quality, so that most current recognition approaches have been designed to work independently from segmentation.

In contrast, this thesis considers object categorization and figure-ground segmentation as two interleaved processes that closely collaborate towards a common goal. The core part of our work is a probabilistic formulation which integrates both capabilities into a common framework. As shown in our experiments, the tight coupling between those two processes allows them to profit from each other and improve their individual performances. The resulting approach can detect categorical objects in novel images and automatically compute a segmentation for them. This segmentation is then used to again improve recognition by allowing the system to focus its effort on object pixels and discard misleading influences from the background.

In addition to improving the recognition performance for individual hypotheses, the top-down segmentation also allows to determine exactly from where a hypothesis draws its support. We use this information to design a hypothesis verification stage based on the MDL principle that resolves ambiguities between overlapping hypotheses on a per-pixel level and factors out the effects of partial occlusion. Altogether, this procedure constitutes a novel mechanism in object detection that allows to analyze scenes containing multiple objects in a principled manner. Our results show that it presents an improvement over conventional criteria based on bounding box overlap and permits more accurate acceptance decisions.

Our approach is based on a highly flexible implicit representation for object shape that can combine the information of local parts observed on different training examples and interpolate between the corresponding objects. As a result, the proposed method can learn object models already from few training examples and achieve competitive object detection performance with training sets that are between one and two orders of magnitude smaller than those used in comparable systems. An extensive evaluation on several large data sets shows that the system is applicable to many different object categories, including both rigid and articulated objects.

Zusammenfassung

Diese Arbeit beschäftigt sich mit der visuellen Objektkategorisierung, d.h. dem Problem, zuvor noch nie gesehene Objekte zu erkennen, in realen Szenen zu lokalisieren, und die Objekte der korrekten Kategorie zuzuweisen. Diese Fähigkeit ist eine der Kernkompetenzen des menschlichen Sehsystems. Die maschinelle Bildverarbeitung is jedoch noch weit davon entfernt, eine vergleichbare Leistung erbringen zu können. Darüberhinaus hat sich die Forschung in der Vergangenheit hauptsächlich auf das einfachere und speziellere Problem konzentriert, bekannte Objekte unter geänderten Bedingungen wiederzuerkennen.

Die visuelle Kategorisierung ist eng mit dem Figure-Ground Segmentierungsproblem verbunden, d.h. mit der Aufgabe, das Bild in eine Objekt- und eine Hintergrund-Region zu trennen. In der Vergangenheit wurde dieses Problem oft als ein wichtiger und sogar notwendiger Vorverarbeitungsschritt für die Objekterkennung betrachtet. Reine Bottom-up Verfahren haben sich aber bis heute als ungeeignet erwiesen, Segmentierungen von genügender Qualität hervorzubringen, so dass die meisten aktuellen Erkennungsansätze dahingehend entworfen wurden, ohne eine vorausgehende Segmentierung auszukommen.

Im Gegensatz dazu betrachtet diese Arbeit die Objektkategorisierung und Figure-Ground Segmentierung als zwei miteinander verwobene Prozesse, die eng zusammenarbeiten, um ein gemeinsames Ziel zu erreichen. Das Herzstück unseres Ansatzes ist eine probabilistische Formulierung, die beide Fähigkeiten in einem gemeinsamen Rahmen verbindet. Wie wir in unseren Experimenten zeigen, erlaubt die enge Zusammenarbeit dieser beiden Prozesse ihnen, voneinander zu profitieren und ihre Einzelleistungen zu verbessern. Der daraus entstehende Ansatz ermöglicht es, Kategorie-Objekte in neuen Bildern zu detektieren und automatisch eine Segmentierung für sie zu berechnen. Diese Segmentierung trägt dann dazu bei, die Erkennungsergebnisse wiederum zu verbessern, indem sie es dem System ermöglicht, sich auf Objektpixel zu konzentrieren und irreführende Einflüsse von Hintergrundstrukturen zu ignorieren.

Zusätzlich zu der besseren Erkennungsleistung für Einzelhypothesen erlaubt die so berechnete Top-Down Segmentierung es unserem Ansatz ebenfalls, zu ermitteln welche Bildstrukturen eine Hypothese stützen und somit für ihr Zustandekommen verantwortlich sind. Wir verwenden diese Information, um eine auf dem MDL-Prinzip beruhende Verifikationsstufe zu entwerfen, die Konflikte zwischen überlappenden Hypothesen Pixel für Pixel auflöst und somit die Folgen partieller Verdeckungen ausklammert. Insgesamt stellt dieses Verfahren einen neuartigen Mechanismus dar, der es erlaubt, Szenen mit mehreren Objekten auf eine fundierte Weise zu untersuchen. Unsere Ergebnisse zeigen, dass dieser Mechanismus eine Verbesserung gegenüber herkömmlichen Kriterien basierend etwa auf dem Bounding-Box Überlappungsgrad darstellt und dass er genauere Akzeptanzentscheidungen ermöglicht.

Unser Ansatz basiert auf einer sehr flexiblen impliziten Darstellung der möglichen Objektformen, die es gestattet, die Informationen von Objektteilen aus unterschiedlichen Trainingsbeispielen zu kombinieren und zwischen den entsprechenden Objekten zu interpolieren. Als Folge davon kann das vorgeschlagene Verfahren Objektmodelle schon aus sehr wenigen Trainingsbeispielen lernen und gute Detektionsleistungen schon mit Trainingsdatenmengen erzielen die ein bis zwei Grössenordnungen unter denen vergleichbarer Systeme liegen. Eine ausführliche Auswertung auf mehreren grossen Bildersammlungen zeigt, dass das vorgestellte System auf viele verschiedene Objektkategorien, mit sowohl starren als auch artikulierten Objekten, anwendbar ist.

Acknowledgments

I would like to sincerely thank all people that contributed in various aspects to the successful completion of this dissertation.

My foremost thanks go to my advisor, Prof. Bernt Schiele, for supervising the thesis and helping with many ideas, comments, feedback, and support. I would also like to thank Andrew Zisserman for his interest in my work and for agreeing to be the co-examiner of this thesis.

I am grateful to my colleagues from the PCCV group: Julia Vogel, Hannes Kruppa, Nicky Kern, Martin Spengler, Florian Michahelles, Stavros Antifakos, and Edgar Seemann; and to my former colleagues Daniela Hall and Alan Ettlin. They made ETH a fun and exciting place to work. Thanks for providing help in research related discussions and for welcome diversions during group retreats and numerous coffee, lunch, and dinner breaks. Special thanks to Julia for the stimulating discussions during the completion of this thesis.

Thanks also to my diploma thesis and semester project students who have contributed to this thesis through their work: Thomas Hug, Markus Käppeli, Andreas Remund, Michael Steiner, Oliver Bay, Peter Matter, and Dirk Zimmer.

This work was part of the CogVis project, funded by the Commission of the European Union (IST-2000-29375) and the Swiss Federal Office for Education and Science (BBW 00.0617). Thanks to all CogVis members from KTH Stockholm, Max-Planck Institute for Biological Cybernetics, Hamburg University, Leeds University, DIST Genova, ETH, and University of Ljubljana for interesting discussions and highly appreciated feedback during research meetings and mutual visits. Special thanks go to Barbara Caputo, Christian Wallraven, Derek Magee, Daniel Skocaj, and Ales Leonardis for the good collaboration.

In addition, I would like to thank Shivani Agarwal and Dan Roth for providing the UIUC database; Derek Magee, for providing the cow sequences; and Rob Fergus and Pietro Perona's group, for providing the CalTech database images that were the basis for many of our experiments.

Last but not least, I want to thank my family for their love and support, and Andrea Zahnd, for being there.

Contents

Contents **x**

1 Introduction **1**
1.1 Visual Object Categorization 1
1.2 Figure-Ground Segmentation 2
1.3 Contributions 3
1.4 Outline of the Thesis 4

2 State of the Art **7**
2.1 Features 8
2.2 Shape/Structure representations 10
2.3 From Recognition to Top-Down Segmentation 14

3 On the Role of Different Cues **17**
3.1 Object Categorization 18
3.2 Evaluation Database 19
3.3 Recognition Methods 21
3.4 Results 24
  3.4.1 Global Recognition Rates 24
  3.4.2 Confusions 25
3.5 Multi-Cue Combination 27
3.6 Discussion 27

4 Codebook Representations **31**
4.1 A Codebook of Local Appearance 32
  4.1.1 Codebook Generation 33
  4.1.2 Representing Uncertainty 34
  4.1.3 Optimality Criterion 37
4.2 Clustering Methods 37
  4.2.1 K-means Clustering 37
  4.2.2 Agglomerative Clustering 38

4.2.3 Experimental Comparison 40
4.3 Efficient Implementation 41
4.3.1 Standard Algorithm 42
4.3.2 RNN Algorithm 43
4.3.3 Efficient Nearest-Neighbor Search 46
4.3.4 Experimental Results 47
4.4 Discussion 48

5 Interleaved Object Categorization and Segmentation 51
5.1 Shape Representation 52
5.1.1 Learning the Shape Model 52
5.1.2 Recognition Approach 54
5.1.3 Probabilistic Framework 54
5.1.4 Experimental Results 56
5.1.5 Discussion 60
5.2 Figure-Ground Segmentation 62
5.2.1 Theoretical Derivation 62
5.2.2 Implementation 63
5.2.3 Experimental Results 65
5.2.4 Discussion 67

6 Multi-Object Scene Analysis 69
6.1 Performance of the Original Approach 69
6.2 Hypothesis Verification Stage 70
6.2.1 Motivation 70
6.2.2 MDL Formulation 72
6.3 Experimental Evaluation 74
6.3.1 Evaluation on a Car Detection Task 74
6.3.2 Recognition of Articulated Objects 76
6.4 Discussion 82

7 Scale-Invariant Object Categorization 85
7.1 Extended Approach 86
7.1.1 Extended Probabilistic Framework 86
7.1.2 Extended Verification Stage 87
7.2 Influence of the Interest Point Detector 88
7.3 Robustness to Scale Changes 91
7.4 Effect of the Training Set Size 92
7.5 Discussion 94

8 Application to Other Object Categories 97

8.1 Application to Pedestrians 97
  8.1.1 Training Procedure 98
  8.1.2 Evaluation Methodology 99
  8.1.3 Single-Scale Detection Results 100
    Influence of Hypothesis Verification Stage 102
    Influence of Lighting Normalization 103
    Influence of Optimized MSME Kernel Size 104
    Comparison to the Single-Scale Approach 105
  8.1.4 Comparison with Chamfer Matching 106
  8.1.5 Multi-Scale Detection Results 110
    Scenes Containing a Single Object 110
    Scenes with Multiple Objects 111
  8.1.6 Performance with Severe Overlaps 113
  8.1.7 Discussion 114
8.2 Application to Motorbikes 115
8.3 Application to Rear Views of Cars 120
8.4 Extensions 122
  8.4.1 Multi-Cue Integration 122
  8.4.2 Multi-Category Discrimination 124
8.5 Discussion 126

9 Learning Semantic Parts 129

9.1 Learning Object Parts 130
  9.1.1 Visual Similarity 131
  9.1.2 Co-Location Clustering 133
  9.1.3 Co-Activation Clustering 134
9.2 Hypothesis Verification using Parts 135
9.3 Discussion 138

10 Conclusion 139

10.1 Contributions 139
10.2 Additional Remarks 140
  10.2.1 Comparison with Classical Object Detection Approaches . . . 140
  10.2.2 Comparison with the Constellation Model 141
10.3 Biological Relevance 142
10.4 Perspectives 144

A Interest Points 149

A.1 The Harris Operator 149

A.2 The DoG Operator 150
A.3 The Harris-Laplacian Operator 151

List of Figures **155**

List of Tables **157**

Bibliography **159**

1

Introduction

The ability to generalize from examples and categorize objects, events, scenes, and places is one of the core competencies of the human visual system. It is also a necessary capability for any artificial system that shall be employed in real-world environments and perform tasks in an autonomous way. The good human performance is an existence proof that a solution is possible. Yet, the underlying processes are still far from being fully understood.

This thesis considers the problem of visual object categorization from a machine perspective. Biological vision serves as an inspiration. However, while our research is clearly guided by insights from psychophysical and neurobiological studies, the goal is not to imitate a biological vision system in its architectural constraints. Instead, we want to explore which computational mechanisms are needed in order to build a system that can perform categorization tasks. It is our belief that an understanding of these mechanisms can in turn be beneficial for interpreting biological results. In addition, object categorization is an important building block for many computer vision applications, ranging from automatic image analysis to autonomous robotic systems. Just as face detection is currently becoming a standard component in numerous domains, a successful categorization system could fulfill a similar role in the development of future applications.

1.1 Visual Object Categorization

In computer vision, object recognition has reached a level where current approaches can identify a large number of previously seen and known objects. However, the more general task of object categorization, that is of recognizing unseen-before objects of a given category and assigning the correct category label, is less well-understood. Obviously, this task is more difficult, since it requires a method to cope with large within-class variations of object colors, textures, and shapes, while retaining at the same time enough specificity to avoid misclassifications. This is especially true for recognition in cluttered real-world scenes, where objects are often partially occluded and where similar-looking background structures can act as additional distractors. Here, it is not only necessary to assign the correct category label to an image, but also to find the objects in the first place and to separate them from the background.

Object categorization thus includes two different sub-tasks: *discrimination* between multiple categories and *detection* in novel images. Those two tasks have rather different characteristics and necessitate different representations. Object detection is mainly a one-class problem[1], where candidate image regions are evaluated on whether they contain an instance of the object category or not. In contrast, discrimination always requires at least two classes, the decision boundary for which may shift depending on the task.

Throughout this thesis, we will use the term *categorization* for the general problem of recognizing novel objects, in contrast to the *identification* of known objects. The two sub-problems of *multi-category discrimination* and of *detection in cluttered images* will be treated separately. Chapter 3 investigates the role of different cues for multi-category discrimination. The main part of the thesis, Chapters 4–8, is then devoted to building a system that can reliably detect and localize categorical objects. At the end of Chapter 8, we will discuss possible ways how the two aspects can be combined, although a unified solution is out of the scope of this work.

An important question when pursuing visual object categorization is for which categories this task is actually well-defined. In other words, what are the categories that can be learned by purely visual means? Section 3.1 addresses this question by basing the categorization task on a framework grounded in Cognitive Psychology. Results from this discipline show that there is a *basic level* in human categorization at which most knowledge is organized (Rosch et al., 1976). This basic level is mostly defined by visual means and is thus a good starting point for our experiments. We will explicitly not consider *functional categories* (e.g. “things you can sit on”) or *ad-hoc categories* (e.g. “things you can find in an office environment”), since those require a higher level of abstraction and a large degree of world knowledge.

1.2 Figure-Ground Segmentation

The problem of separating objects of interest from the background is generally known as *figure-ground segmentation*. Historically, figure-ground segmentation has often been seen as an important and even necessary precursor for object recognition (Marr, 1982). In this context, segmentation is mostly defined as a data driven, that is bottom-up, process. However, except for cases where additional cues, such as motion or stereo, could be used, purely bottom-up approaches have so far been unable to yield figure-ground segmentations of sufficient quality for object categorization. This is also due to the fact that the notion and definition of what constitutes an object is largely task-specific and cannot thus be answered in an uninformed way. Indeed, recent results from human vision indicate that for humans, recognition and segmentation are heavily intertwined processes (Peterson, 1994; Vecera and O’Reilly,

---

[1] Many approaches nevertheless treat object detection as a two-class problem with an all-encompassing background class in order to draw from the richer pool of machine-learning techniques for two-class problems. While this is perfectly legitimate, we will show in this thesis that the view as a one-class problem is sufficient for the detection of individual categories.

1998; Needham, 2001). It has thus been argued that top-down knowledge from object recognition can and should be used for guiding the segmentation process.

1.3 Contributions

In this thesis, we treat object categorization and figure-ground segmentation as two interleaved processes that closely collaborate towards a common goal. Chapter 5 presents a local approach that integrates both capabilities into a common framework. In particular, we derive a probabilistic formulation of the problem that allows us to incorporate knowledge about the recognized category, as well as the supporting information in the image. As a result, our approach can detect categorical objects in real-world scenes and automatically obtain a segmentation for them, together with a per-pixel confidence estimate specifying how much this segmentation can be trusted. Thus, figure-ground segmentation is addressed as a result and extension of object recognition.

In return, the segmentation information is then used to again improve recognition. Thus, recognition can profit by only aggregating evidence over the object portion of the image and discarding influences from the background. In addition, the knowledge from where in the image a hypothesis draws its support allows to resolve ambiguities between overlapping hypotheses on a per-pixel level. Chapter 6 formalizes this idea in a hypothesis verification criterion based on the Minimum Description Length (MDL) principle. Altogether, this procedure presents a novel mechanism that allows to analyze scenes containing multiple overlapping objects in a principled manner.

These interleaved steps lead to an iterative evidence aggregation scheme that tries to make maximal use of the information extracted from input images. In order to make this process robust to real-world conditions, we pay specific attention to the representation and propagation of uncertainty on all levels. Thus, Chapter 4 investigates how the uncertainty introduced by matching features to an internal representation can be retained. Chapter 5 then extends the resulting formulation to include also the spatial uncertainty of feature occurrences. Finally, Chapter 6 addresses the uncertainty introduced by overlapping hypotheses, and Chapter 7 deals with uncertainty about the object scale.

In addition, our approach makes an important contribution in two further aspects. Many current object detection systems require very large training sets with up to 5–10,000 positive examples per category (Papageorgiou and Poggio, 2000; Schneiderman and Kanade, 2000, 2004; Viola and Jones, 2001, 2004). The effort of manually preparing such large training sets typically restricts their application to a small set of categories. Even when large training sets are available, learning complexity often dictates that objects be represented on very low resolutions, such as 24×24 (Viola and Jones, 2001, 2004), 32×32 (Torralba et al., 2004), or even 15×10 pixels (Vidal-Naquet and Ullman, 2003). Such low resolutions make it difficult to compensate for the effects of overlaps and partial occlusion. In contrast, the flexible nature of our approach allows it to learn category models already from small training sets with only 50–150 examples. In addition, the low learning complexity of our method makes it possible to represent object models at higher resolutions. As our experiments in Chapters 6 and 8 demonstrate, the resulting system can therefore recognize objects in crowded scenes and under significant partial occlusion.

Last but not least, we explore how the semantic structure of an object category can be learned from training data. Starting from a purely visual representation, semantically meaningful object parts are learned by a hierarchy of grouping steps based on non-visual constraints. Chapter 9 proposes two novel grouping principles that can be used for this purpose: co-location and co-activation. The resulting representation forms an intermediate layer that can be used to interface the visual representation with higher-level reasoning mechanisms. As an example of such a mechanism, we show how the learned parts can be integrated in a Bayesian network that verifies hypotheses by reasoning about part configurations.

1.4 Outline of the Thesis

The thesis is structured as follows:

Chapter 2 gives and overview over related work in generic object recognition and object categorization. In particular, we review the different recognition methods, features, and structural representations that are used in current appearance-based approaches. In addition, the chapter documents the recent transition from recognition to top-down segmentation.

Chapter 3 investigates the role of different cues for multi-category discrimination and analyzes how several state-of-the-art object identification methods generalize to this task. In order to arrive at a meaningful definition for the categorization task, we first cast it in a framework grounded in Cognitive Psychology. Based on this definition, we build up a novel evaluation database, specifically tailored to multi-category discrimination, and use it to compare well-known recognition methods based on color, texture, contours, and shape. Our evaluation shows that global and local shape cues are the most important single features for discriminating the given categories. What is more important, though, is that every tested cue is the best choice for at least one category, which highlights the importance of multi-cue combination. We further investigate this potential by evaluating different cue combinations. The results confirm our previous findings that contour and shape cues are most important for discrimination: the total performance drops most when they are not included in the cue combination.

Chapter 4 introduces the codebook representation that is the basis for our object detection approach. In order to represent the appearance variability of an object category, we build up a vocabulary of local appearances that are characteristic for its member objects. This is done by extracting image patches around interest points and grouping them with an agglomerative clustering scheme. We pay particular attention to the question how the matching uncertainty can be represented and propagated to later processing stages. As the initial clustering step will be applied to large data sets, an efficient implementation is crucial. The chapter therefore reviews different clustering methods and describes efficient algorithms that can be used for codebook generation.

Chapter 5 builds on this codebook representation to develop our interleaved categorization and segmentation approach. At the core of our approach is an *Implicit Shape Model*, which extends the uncertainty handling idea further in order to estimate the spatial occurrence distribution of codebook entries on the target category. The resulting representation is highly flexible and can learn object models already from few training examples. In addition, the chapter introduces a probabilistic formulation for the segmentation problem, which integrates learned knowledge of the recognized category with the supporting information in the image. The resulting procedure yields a pixel-wise figure-ground segmentation as result and extension of recognition. In addition, it returns a per-pixel confidence estimate, specifying how much this segmentation can be trusted.

Chapter 6 extends the approach by a hypothesis verification stage that uses the segmentation result to again improve recognition. The segmentation allows to determine from where in the image a hypothesis draws its support. This information is valuable for two reasons. First, it can be used to only aggregate evidence over the object region and discard influences from the background. Second, it allows to resolve ambiguities between overlapping objects and factor out the effects of partial occlusion. We formalize this idea in a criterion based on the Minimum Description Length (MDL) principle. The resulting mechanism presents a fundamental improvement over previous criteria based on bounding box overlap and allows to handle scenes containing multiple objects in a principled manner. An extensive evaluation on two large data sets shows that the system achieves excellent detection and segmentation results for categories as diverse as cars and cows. At the same time, its flexible representation allows it to generalize already from small training sets.

Chapter 7 generalizes our approach to scale-invariant detection and makes it thus usable in real-world situations where the object scale is a-priori unknown. The extension is based on the use of scale-invariant interest points, that is of structures whose size can be reliably estimated under scale changes. Replacing the original single-scale image patches, those structures vote not only for possible object positions, but also for the corresponding object scales. The search for hypotheses with maximal support is formulated in a Mean-Shift framework, which allows to draw parallels to kernel density estimation. We experimentally evaluate the suitability of different interest point operators for use in our system and quantify the robustness of the resulting approach to large scale changes. Our results show that the proposed scheme achieves good detection results while being robust to scale changes of more than a factor of two.

Chapter 8 demonstrates the versatility of the proposed approach by applying it to three additional object categories. The evaluations on pedestrians, motorbikes, and rear views of cars allow to verify previous results also for other scenarios and more difficult scenes. We apply the system to a sequence of test sets of increasing difficulty, culminating in a pedestrian detection task in crowded scenes with severe overlaps. In addition, we compare the results of our method with a reimplementation of the Chamfer matching approach (Gavrila, 1998), as an example of an existing pedestrian detection system. This comparison allows us to assess the potential for a combination with global cues. Finally, we discuss possible extensions to multi-cue integration and multi-category discrimination.

Chapter 9 investigates how the semantic structure of an object category can be learned from training data. We argue that visual appearance alone is not enough for this learning step. Instead, we propose to use a hierarchy of grouping steps based on non-visual constraints, such as the information that the object views in two images are aligned. These constraints lead to two novel grouping principles, co-location and co-activation. By applying each grouping criterion as long as it performs reliably, we arrive at an intermediate layer of semantically meaningful object parts, which can be used to interface the visual information readily available from the image with higher-level reasoning mechanisms. We show how such a mechanism can be implemented by integrating the learned subparts and parts into a Bayesian network for hypothesis verification that improves the recognition results by reasoning about allowed part configurations. While our results demonstrate the feasibility of the structure learning mechanism, the experimental evaluation is not complete, so that this chapter may be seen as a perspective of the thesis.

Chapter 10 concludes by discussing the biological relevance of the developed approach and listing future directions and perspectives of the thesis.

2

State of the Art

Early approaches to "generic" object recognition represented objects by 3D models or by a decomposition into parametric surfaces or volumetric primitives (Roberts, 1963; Binford, 1971; Marr, 1982; Biederman, 1987). A central element of those approaches was the use of an object-centered coordinate system, which should enable view-invariant recognition. However, the difficulty of reliably extracting the postulated geometric representations from real-world images and of finding viewpoint-invariant yet discriminative descriptions restricted their success.

The object-centered paradigm was challenged in the early 90's by the success of view-based approaches, which showed that the use of a viewer-centric coordinate system and fast image-comparison methods made it possible to identify a large number of known objects (Swain and Ballard, 1991; Murase and Nayar, 1995; Rao and Ballard, 1995; Mel, 1996; Schmid and Mohr, 1996; Nelson and Selinger, 1998a; Schiele and Crowley, 2000). A main focus of those efforts was to achieve robustness to deteriorated viewing conditions caused by changes in viewpoint, scale, and image plane rotation; and by the introduction of noise, clutter, and occlusion. However, the robustness to these influences was often just tested for individual objects and under laboratory conditions. In contrast, object detection approaches concentrated on the task of finding instances of a single object class under real-world conditions. Impressive results, both in terms of accuracy and run-time, have been achieved for object classes such as pedestrians, faces, and cars (Rowley et al., 1998; Schneiderman and Kanade, 2000, 2004; Papageorgiou and Poggio, 2000; Viola and Jones, 2001; Viola et al., 2003; Gavrila, 2000), even though often only for single viewpoints.

Still, the more general task of multi-class object categorization under real-world conditions is yet largely unsolved. In recent years, renewed interest in the topic has sparked various new approaches, including (Weber et al., 2000a; Leibe and Schiele, 2003a; Fergus et al., 2003; Li, 2004; Nilsback and Caputo, 2004). Nevertheless, many questions are still open. While it is generally agreed upon that the underlying features should be local in order to cope with noise and occlusion, it is not yet clear which features are best for object categorization, nor in what kind of structural representation they should be combined. The following sections therefore aim to give an overview of the design choices that have been used in various systems so far.

The history of figure-ground segmentation is closely connected to that of recognition. In early approaches, segmentation was often seen as a necessary preprocessing step for recognition, for which a solution could be assumed. While the difficulty of segmentation was generally acknowledged, it was still postulated that a separation into object and non-object regions could be achieved by a series of purely bottom-up grouping steps of corner, line, and region primitives. The general failure to achieve this goal, together with the success of appearance-based methods to provide recognition results without prior segmentation, led to the separation of the two areas and the further development of recognition independent from segmentation. In recent years, however, the areas have converged again by the growing insight that recognition and segmentation are indeed interleaved processes and that intermediate recognition results can be used to drive a top-down segmentation process (Borenstein and Ullman, 2002; Yu and Shi, 2003; Leibe and Schiele, 2003b).

In the following, we give an overview of features and structural representations that are used in current approaches to object categorization. In addition, we document the recent transition from recognition to top-down segmentation, which has been developing into an area of active research. In accordance with the current state of the field, we concentrate on view-based object categorization with a focus on local appearance-based methods.

2.1 Features

The design choice which features to use for recognition can be divided into two separate questions. The first question is which image locations shall be sampled, that is which subset of the available image information shall actually be used. The second question is then how to represent the sampled image information and which descriptors to compute.

Of course, these questions are closely interrelated. Many early appearance-based methods rely on relatively simple features that are sufficiently low-dimensional that they can be computed (and stored) over the full image. Swain and Ballard (1991) represent an object by its color histogram (approximating its color distribution). Objects are identified by matching a color histogram from a test image region with the histograms from training objects. Other authors use multidimensional combinations of derivatives. Rao and Ballard (1995) represent objects (or object patches) by a high-dimensional "iconic" feature vector, consisting of 45 responses of nine oriented Gaussian filters at five different scales ($9 \times 5 = 45$). Using the steerability of Gaussian derivatives, the feature vector is made rotational invariant. Schmid and Mohr (1996) propose instead to describe an image by a nine-dimensional rotational invariant vector of local characteristics based on Gaussian derivatives computed at interest points. Schiele and Crowley (2000) generalize the color histogram approach to represent objects by multidimensional histograms of greyvalue derivatives over multiple scales. Hall et al. (2000) again generalize this approach to include a combination of color and grayvalue derivatives in an eight-dimensional feature vector. In Chapter 3, we analyze the suitability of various simple feature descriptors for the task of object categorization (see also Leibe and Schiele, 2003a). Our results indicate that while the simple features allow for a surprising degree of generalization, the more important information for categorization seems to be the (global or local) object shape.

In contrast to these relatively low-dimensional representations, the following local descriptors are so high-dimensional that they are typically only evaluated at certain specific locations, such as a those returned by interest point detectors. The interest point detectors themselves have a long history. The underlying idea is to search for locations that are distinctive enough that they can be reliably extracted under various image transformations. Depending on the application, this can be boundary concavities or curvature extrema (Lamdan et al., 1988), corner-like structures (Harris and Stephens, 1988), or extrema of local operators optimized for scale-invariant (Lindeberg, 1998; Mikolajczyk and Schmid, 2001; Kadir and Brady, 2001) or even affine-invariant extraction (Tuytelaars and van Gool, 2000, 2004; Mikolajczyk and Schmid, 2002; Matas et al., 2002; Schaffalitzky and Zisserman, 2002). The use of interest point detectors allows to represent objects by a relatively small set of local descriptors, such as the ones described below.

Perhaps the simplest local descriptor for matching interest points is just a raw image patch, as used in (Burl et al., 1998; Weber et al., 2000a,b; Agarwal and Roth, 2002; Ullman et al., 2002; Vidal-Naquet and Ullman, 2003; Fergus et al., 2003; Leibe and Schiele, 2003b). The advantages of such a representation are its simplicity of implementation and the ability to directly visualize what has been matched, which may be helpful during algorithmic design. A disadvantage is the higher dimensionality. However, this problem may be alleviated by performing matching in a truncated eigenspace with a significantly reduced number of dimensions (see Fergus et al., 2003, for an example). Typically, some kind of lighting normalization is also performed prior to matching. The idea to just extract patches from the input image can also be augmented by other preprocessing steps. In addition to raw image patches, Weber et al. (2000b) also propose a representation based on high-pass filtered patches. Nelson and Selinger (1998a,b) go one step further and extract local windows of edge-like structures (based on a robust computation of curvature extrema) as basic features.

Lowe (1999, 2001, 2004) introduces a local feature representation inspired by the response properties of complex neurons in the human visual cortex. His so-called SIFT descriptors are defined as 4×4 grids of localized histograms of gradient orientations computed at multiple scales. Mikolajczyk et al. (2003) extend this idea and compute a similar descriptor based on localized edge direction histograms.

Although a recent study has compared a representative selection of local descriptors for the task of finding exact correspondences between image pairs (Mikolajczyk and Schmid, 2003), it is not yet clear which of them are best suited for object categorization, where a generalization to certain within-class variations is needed. In our experience, however, the exact choice of descriptors is not as important for this task as how their combination is performed and how the object structure is represented.

2.2 Shape/Structure representations

Over the years, the term *shape* has been connotated with many meanings. For this discussion, we therefore find it important to make a clear distinction between the concepts of *shape* and *structure*. Many early papers have argued for the need to find an object representation that is both invariant to variations in the imaging process and mathematically simple to model. In their context, *pure shape* denotes the information which remains when the effects of color, texture, and illumination are discarded. In practice, the term is often equated with the object contour or silhouette, or with an edge-based representation. In order to avoid confusions with this definition, we will use the term *object structure* to denote a set of spatial relations between local appearances[1]. The two concepts represent different strategies for dealing with the inherent problem of imaging variations in computer vision. While shape-based methods try to cope with these variations by building a representation that is invariant to them, structure-based methods are trying to use them to build a more realistic object model. In this section, we will only focus on representations for object structure – for a detailed discussion of shape-based methods, we refer to the excellent review in (Shokoufandeh et al., 2004).

A large class of methods match object structure by computing a cost term for the deformation needed to transform a prototypical object model to correspond with the image. Prominent examples of this approach include *Deformable Templates* (Yuille et al., 1989; Sclaroff, 1997), *Morphable Models* (Jones and Poggio, 1998a,b; Giese and Poggio, 2000), *Shape Context Matching* (Belongie et al., 2001, 2002), or *Combinatorial Geometric Hashing* (Sullivan and Carlsson, 2002). The main difference between them lies in the way point correspondences are found and in the choice of energy function for computing the deformation cost (e.g. Euclidean distances, strain energy, thin plate splines, etc.). Cootes et al. (1998) go one step further and characterize objects by means and modes of variation for both shape and texture. Their *Active Appearance Models* first warp the object to a mean shape, so that the texture variations can be computed on corresponding object pixels. Since there may be correlations between the shape and greylevel variations, they then estimate the combined modes of variation of the concatenated shape and texture models. For matching the resulting AAMs to a test image, they learn the relationship between model parameter displacements and the induced differences in the reconstructed model image. This allows them to learn an inverse mapping from residual matching errors to the parameter changes that lead to a better fit. Provided that the method is initialized with a close estimate of the object's position and size, a good overall match to the object is typically obtained in a few iterations, even for deformable objects. Blanz and Vetter (2003) generalize this approach further to use densely sampled 3D models obtained by a laser scanner, instead of 2D shape models.

Wiskott et al. (1997) propose a different structural model known as *Bunch Graph*.

---

[1] Note that this explicitly includes the above-mentioned effects of color, texture, and illumination. In particular, *structure* can also be defined on top of (local) *shape*, but not vice versa.

The original version of this approach represents object structure as a graph of hand-defined locations, at which local jets (multidimensional vectors of simple filter responses) are computed. The method learns an object model by storing, for each graph node, the set ("bunch") of all jet responses that have been observed in this location on a hand-aligned training set. During recognition, only the strongest response is taken per location, and the joint model fit is optimized by an iterative elastic graph matching technique. This approach has achieved impressive results for face identification tasks, but an application to more object classes has been restricted by the need to hand-model a set of suitable graph locations. A recent generalization of the method, however, alleviates this restriction by automatically learning a suitable graph structure (Loos and v.d. Malsburg, 2002). More recently, (Hall and Crowley, 2003; Hall, 2004) have presented an object categorization method that uses a similar elastic graph matching technique for comparing the spatial arrangement of a set of learned prototypical region detectors.

In contrast to those deformable representations, most classic object detection methods either use a monolithical object representation (Rowley et al., 1998; Papageorgiou and Poggio, 2000) or look for local features in fixed configurations (Schneiderman and Kanade, 2000, 2004; Schneiderman, 2004; Viola and Jones, 2001, 2004). Schneiderman and Kanade (2000, 2004) express the likelihood of object and non-object appearance using a product of localized histograms, which represent the joint statistics of subsets of wavelet coefficients and their position on the object. The detection decision is made by a likelihood-ratio classifier. Multiple detectors, each specialized to a certain orientation of the object, are used to achieve recognition over a variety of poses, including frontal and profile faces and various views of passenger cars. Their approach achieves very good detection results on standard databases, but is computationally still relatively costly. Viola and Jones (2001, 2004) instead focus on building a speed-optimized system for face detection. They achieve this by building a cascade of simple classifiers, each of which is based only on the differences between average grayvalues summed over fixed image regions. The classifiers themselves are simple threshold functions, but their ensemble allows to learn complex appearance variations. In more recent work, Viola et al. (2003) extend this approach to pedestrian detection using a combination of appearance and motion features. In recent years, this class of approaches has been shown to yield fast and accurate object detection results under real-world conditions (Lienhart et al., 2003; Torralba et al., 2004; Kruppa, 2004). However, a drawback of these methods is that since they do not explicitly model local variations in object structure (e.g. from body parts in different articulations), they typically need a large number of training examples in order to learn the allowed changes in global appearance.

One way to model these local variations is by representing objects as an assembly of parts. Early approaches that tried to model objects by a set of hand-defined or postulated geometric parts with a rule-based combination scheme (Marr, 1982; Biederman, 1987; Huang et al., 1997) were not too successful, mainly because of the difficulty of reliably extracting the geometric representations from real-world images. Even though some later approaches had more success with (human and horse) body parts modeled as cylinders (Forsyth and Fleck, 1997), the geometric part definition typically restricts them to a small application domain. For this reason, many current methods use appearance-based parts instead and try to learn as much as possible about the object model instead of postulating it.

Mohan et al. (2001) still use a set of hand-defined appearance parts, but learn an SVM-based configuration classifier for pedestrian detection. The resulting system performs significantly better than the original full-body person detector by Papageorgiou and Poggio (2000). In addition, its component-based architecture makes it more robust to partial occlusion. Heisele et al. (2001) use a similar approach for component-based face detection. As an extension of Mohan et al.'s approach, their method also includes an automatic learning step for finding a set of discriminative components from user-specified seed points. More recently, several other part-classifier approaches have been proposed for pedestrian (Ronfard et al., 2002; Mikolajczyk et al., 2004) or car detection (Kruppa, 2004). In all of those approaches, the parts are manually specified.

Nelson and Selinger (1998a,b) propose to recognize objects by an approach based on the assembly of local "context frames". Their algorithm extracts contour segments from intensity images based on a robust computation of curvature extrema. Extracted segments are stored together with their context, i.e. the relative position of other segments in their surroundings, in a local reference frame, which is brought to a canonical orientation. For recognition, matches between these local context regions are searched, which are then combined to single object views in a global skeleton. Experimental results show that the method achieves good results for object recognition in cluttered scenes and under partial occlusion. Although the original system was only intended for the identification of known, rigid objects, later results indicate also good generalization performance for object categories such as cups, toy cars, toy airplanes, and snakes.

Burl et al. (1998) learn the assembly of hand-selected (appearance) object parts by modelling their joint spatial probability distribution. Weber et al. (2000a,b) build on the same framework, but also learn the local parts and estimate their joint distribution. Fergus et al. (2003) extend this approach to scale-invariant object parts and estimate their joint spatial and appearance distribution. The resulting *Constellation Model* has been successfully demonstrated on several object categories. However, the complexity of the combined estimation step restricts it to a relatively small number of (only 5–6) parts. Li et al. (2003) also extend the Constellation Model to learn a category prior, so that the system can utilize the knowledge from three already learned categories to learn a model for a fourth category from only 3–10 examples. In more recent work, this approach is further generalized to learn category models online and thus avoid the commonly-used batch learning phase (Li, 2004). Moreover, the method is evaluated on an object present/absent task with 101 categories. However, this extension uses an even smaller number of only 4 parts.

Agarwal and Roth (2002) keep a larger number of object parts and apply a feature-efficient classifier for learning spatial configurations between pairs of parts.

However, their learning approach relies on the repeated observation of cooccurrences between the same parts in similar spatial relations, which again requires a large number of training examples. b Ullman et al. (2002) and Vidal-Naquet and Ullman (2003) represent objects by a set of fragments that were chosen to maximize the information content with respect to an object class. Candidate fragments are extracted at different sizes and from different locations of an initial set of training images. From this set, their approach iteratively selects those fragments that add the maximal amount of information about the object class to the already selected set, thus effectively resulting in a cover of the object. In addition, the approach automatically selects, for each fragment, the optimal threshold such that it can be reliably detected. For recognition, however, only the information which model fragments were detected is encoded in a binary-valued feature vector (similar to Agarwal & Roth's), onto which a simple linear classifier is applied without any additional shape model. The main challenge of this approach is that the complexity of the fragment selection process restricts the method to very low image resolutions (e.g. 14 × 21 pixels), which severely limits its applicability in practice.

Wallraven et al. (2003) and Caputo et al. (2004) have recently proposed a different approach for combining the results of local feature detectors. They also extract features around interest points, but use them in a Support Vector Machine (SVM) with a specially developed *local kernel*. This *local kernel* combines the discriminance of SVMs with the flexibility and robustness of local features. It searches for correspondences between features extracted from the test image to stored training cases by applying a greedy matching strategy with an additional location constraint. The results of this matching process are then directly used for SVM classification. The method has been successfully applied to several multi-class categorization tasks (Caputo et al., 2004). However, it currently only categorizes full images and contains no object detection component yet.

Another large group of approaches, the *Geometric Methods*, represent object structure only implicitly for computationally efficient matching. The basic idea behind these approaches is to avoid costly grouping operations of local features in the image space and instead transform them into a space where whole object configurations can be described by single points. In the *Generalized Hough Transform* (Hough, 1962; Ballard, 1981; Grimson, 1990), matching feature pairs between a model and test image are translated into votes for a rigid transformation which would align the two objects under the assumption that the matches are correct. As the same procedure is independently applied to a large number of feature pairs, consistent votes for the same transformation reinforce each other and result in distinct peaks in the voting space, while false votes from random mismatches are uniformly spread over the space of possible transformations. As a result, the Hough Transform is robust to a high percentage of outliers (Lowe, 2004). *Geometric Hashing* (Lamdan et al., 1988; Lamdan and Wolfson, 1988; Wolfson, 1990) follows a similar principle, but stores votes in a one-dimensional hash table. In the original approach, triplets of feature points define an affine basis for calculating the coordinates of all remaining points, which are used as entries to the hash table (together with the associated object label). In an offline learning step, the hash table is precomputed for all such triplets from all training objects. During recognition, only a small number of triplets need to be taken as affine bases for querying the hash table with the transformed coordinates of the remaining points. The method then accumulates the votes from accessed hash table entries to determine the object identity and its location in the scene. Originally, geometric methods have been introduced and motivated for the identification of specific, solid objects. Successful applications for this purpose include (Lowe, 2001, 2004). However, in Chapter 5, we will describe how the Hough Transform can also be generalized to recognize object categories.

2.3 From Recognition to Top-Down Segmentation

The traditional view of object recognition has been that prior to the recognition process, an earlier stage of perceptual organization occurs to determine which features, locations, or surfaces most likely belong together (Marr, 1982). As a result, the segregation of the image into a figure and a ground part has often been seen as a prerequisite for recognition. In that context, segmentation is mostly defined as a bottom-up process, employing no higher-level knowledge. State-of-the-art segmentation methods combine grouping of similar image regions with splitting processes concerned with finding most likely borders (Shi and Malik, 1997; Sharon et al., 2000; Malik et al., 2001). However, grouping is mostly done based on low-level image features, like color or texture statistics, which require no prior knowledge. While that makes them universally applicable, it often leads to poor segmentations of objects of interest, splitting them into multiple regions or merging them with parts of the background (Borenstein and Ullman, 2002).

Results from human vision indicate, however, that object recognition processes can operate before or intertwined with figure-ground organization and can in fact be used to drive the process (Peterson, 1994; Vecera and O'Reilly, 1998; Needham, 2001). In consequence, the idea to use object-specific information for driving figure-ground segmentation has recently developed into an area of active research. Approaches, such as Deformable Templates (Yuille et al., 1989), or Active Appearance Models (Cootes et al., 1998) are typically used when the object of interest is known to be present in the image and an initial estimate of its size and location can be obtained. Examples of successful applications include tracking and medical image analysis.

Borenstein and Ullman (2002) represent object knowledge using image fragments together with their figure-ground labelling (as known from a training set). Class-specific segmentations are obtained by fitting fragments to the image and combining them in jigsaw-puzzle fashion, such that their figure-ground labels form a consistent mapping. While the authors present impressive results for segmenting side views of horses, their initial approach includes no global recognition process. As only the local consistency of adjacent pairs of fragments is checked, there is no guarantee that the resulting cover really corresponds to an object and is not just caused by background clutter resembling random object parts. In more recent work, the approach is extended to also learn the figure-ground labeling of training images in an unsupervised fashion (Borenstein and Ullman, 2004), and to combine the top-down segmentation with bottom-up segmentation cues in order to obtain higher-quality results (Borenstein et al., 2004).

Yu and Shi (2003) also present a parallel segmentation and recognition system. They formulate the segmentation problem in a graph theoretic framework that combines patch and pixel groupings. A set of 15 known objects is represented by local color, intensity and orientation histograms obtained from a number of different viewpoints. During recognition, these features are matched to patches extracted from the image to obtain object-part hypotheses, which are combined with pixel groupings based on orientation energy. A final solution is found using the Normalized Cuts criterion (Shi and Malik, 1997). This method achieves good segmentation results in cluttered real-world settings. However, their system needs to know the exact objects beforehand in order to extract their most discriminant features.

In our application, we cannot assume the objects to be known beforehand — only familiarity with the object category is required. This means that the system needs to have seen some examples of the object category before, but those do not have to be the ones that are to be recognized later. Obviously, this makes the task more difficult, since we cannot rely on any object-specific feature, but have to compensate for large in-class variations.

The following chapter presents a comparison of various simple recognition methods (initially introduced for identifying known objects) when applied to an object categorization task. The evaluation focuses on the role different cues (such as color, texture, contours, and shape) play for this more general problem. Chapters 4–7 then develop a local-feature based approach for detecting categorical objects in real-world images.

3

On the Role of Different Cues

Over the last years, the problem of object recognition has been thoroughly researched, and significant progress has been made for identifying known objects in different poses and under novel viewing conditions (Swain and Ballard, 1991; Rao and Ballard, 1995; Murase and Nayar, 1995; Mel, 1996; Schmid and Mohr, 1996; Nelson and Selinger, 1998a; Lowe, 1999, 2001; Schiele and Crowley, 2000).

However, as yet little is known about the more general task of object categorization. Obviously, this task is more difficult, since approaches do not only have to deal with changing viewing conditions, but also with potentially large intra-class variation. While some impressive results have been achieved for the detection of individual categories, such as faces, cars, and pedestrians (Rowley et al., 1998; Schneiderman and Kanade, 2000; Papageorgiou and Poggio, 2000; Viola and Jones, 2001; Viola et al., 2003), little progress has so far been made on the discrimination of multiple categories, with some notable exceptions (e.g. (Nelson and Selinger, 1998b), and more recently (Li, 2004) and (Torralba et al., 2004)).

As a first step towards this goal, it is necessary to evaluate the status quo and analyze what can already be achieved with the methods that were initially introduced for object identification. Many recognition methods have not been tested on multi-class categorization, so that little is known about their respective capabilities to generalize beyond known and seen objects. Also, it is not clear what the role of different cues such as color, texture, contours, and shape is for categorization. Traditionally, contour and shape based methods are considered most adequate for handling the generalization requirements needed for categorization tasks, but an empirical proof for this view is still lacking.

To address these issues, we have built an evaluation database specifically tailored to the task of multi-category discrimination. It contains 80 objects from 8 categories. Each object is represented by 41 views spaced evenly over the upper viewing hemisphere. Using this database, we analyze the performance of different recognition methods for object categorization. As a segmentation mask is provided for each image, both appearance and contour based methods can be compared in the idealized setting of perfect segmentation. Even though any comparison on a particular database has its limitations, we believe that an evaluation in the same setting can help determine the relative importance of different cues for object categorization and shed some light on their interplay. This analysis will set the context for the development of our object categorization system in the following chapters.

Section 3.1 casts the object categorization problem in a framework founded in Cognitive Psychology. This foundation motivates our object database, introduced in Section 3.2. Different contour and appearance-based methods are then introduced in Section 3.3, and Section 3.4 presents experimental results comparing those methods as well as different cues for object categorization. As expected, different methods and cues have their respective strengths and weaknesses. Therefore, Section 3.5 proposes and discusses the combination of different methods.

3.1 Object Categorization

It is important to emphasize that the notion and the abstraction level of object classes is far from being uniquely and clearly defined. Notably, the question of how humans organize knowledge at different levels has received much attention in Cognitive Psychology (Brown, 1958). Taking an example from Brown's work, a dog can not only be thought of as a *dog*, but also as a *boxer*, a *quadruped*, or in general an *animate being* (Brown, 1958). Yet, *dog* is the term that comes to mind most easily, which is by no means accidental. Experiments show that there is a *basic level* in human categorization at which most knowledge is organized (Rosch et al., 1976). According to Rosch et al. (1976) and Lakoff (1987), this basic level is also

- the highest level at which category members have similar perceived shape.
- the highest level at which a single mental image can reflect the entire category.
- the highest level at which a person uses similar motor actions for interacting with category members.
- the level at which human subjects are usually fastest at identifying category members.
- the first level named and understood by children.

These points are the motivation for us to address multi-level object categorization rather than the less clearly defined problem of object classification. Basic level categorization is easiest for humans. At the next lower levels, subordinate categories and the exemplar level used in object identification can be found. The next higher level, superordinate categories, requires a higher degree of abstraction and world knowledge. It is thus useful to start the generic object recognition task in the framework of basic-level categories, which seem to be a good starting point for visual classification.

Another argument is that the distinction between object classes may be quite arbitrary when drawing strict borders between any two classes. In reality, some classes are inherently more similar than others (e.g. dogs and horses are more similar since they are quadrupeds than dogs and cars). Looking at multiple levels of object categorization rather than individual classes, it becomes a desired property that objects from the same superordinate category, such as quadrupeds, be classified as more similar than objects from different superordinate categories. If the object itself is not correctly recognized, then we want it to be assigned at least to a "similar" category (graceful degradation).

The experiments in this work are restricted to basic level categories. In a first step, we explicitly do not want to model functional categories (e.g. *"things you can sit on"*) and ad-hoc categories (e.g. *"things you can find in an office environment"*) (Barsalou, 1983). Even though those categories are important, they exist only on a higher level of abstraction and require a high degree of world knowledge and experience living in the real world.

It is also important to note that categories do not exist per se in the world; they are a learned representation (Rosch et al., 1976) and therefore depend on experience and education. So, different people may have different entry levels for categorizing certain objects they are specialized in. Similarly, it may not always be possible to find *the* unique basic level for every object. However, there are objects that have become so much part of our daily life that their basic level is well-defined almost all over the world (e.g. apples, horses, cars, etc.). In the following section, we will introduce an evaluation database, which contains some of those categories.

3.2 Evaluation Database

Existing publicly available image databases, like the COIL (Murase and Nayar, 1995), have been very influential. Most directly related to our endeavor, the RSORT database (Nelson and Selinger, 1998b) contains full-sphere views of objects from 5 categories, but only includes grayscale images and no segmentations. In this section, we present a new database for object categorization containing 80 objects from 8 carefully chosen categories, high-resolution color images, and segmentation masks for every image.

In our work, we want to explore categorization for both natural and artificial (human-made) objects. In particular, we include objects from the following basic areas: "fruits & vegetables"; "animals"; "human-made, small (graspable)"; and "human-made, big" (e.g. vehicles). Objects from these areas have different affordances, that is different ways of interacting with the environment, and thus different characteristics. For the first iteration of our database, we chose to include the following objects: apples, pears, and tomatoes for the "fruits & vegetables" area; cows, dogs, and horses for the "animals"; cups for the "graspable", and cars for the "vehicles" supercategory.

In principle, there are two ways how such a database can be built. A category can either be set up by a representative distribution of member objects reflecting their probabilities of occurrence in practice, or by a few prototypes that approximately

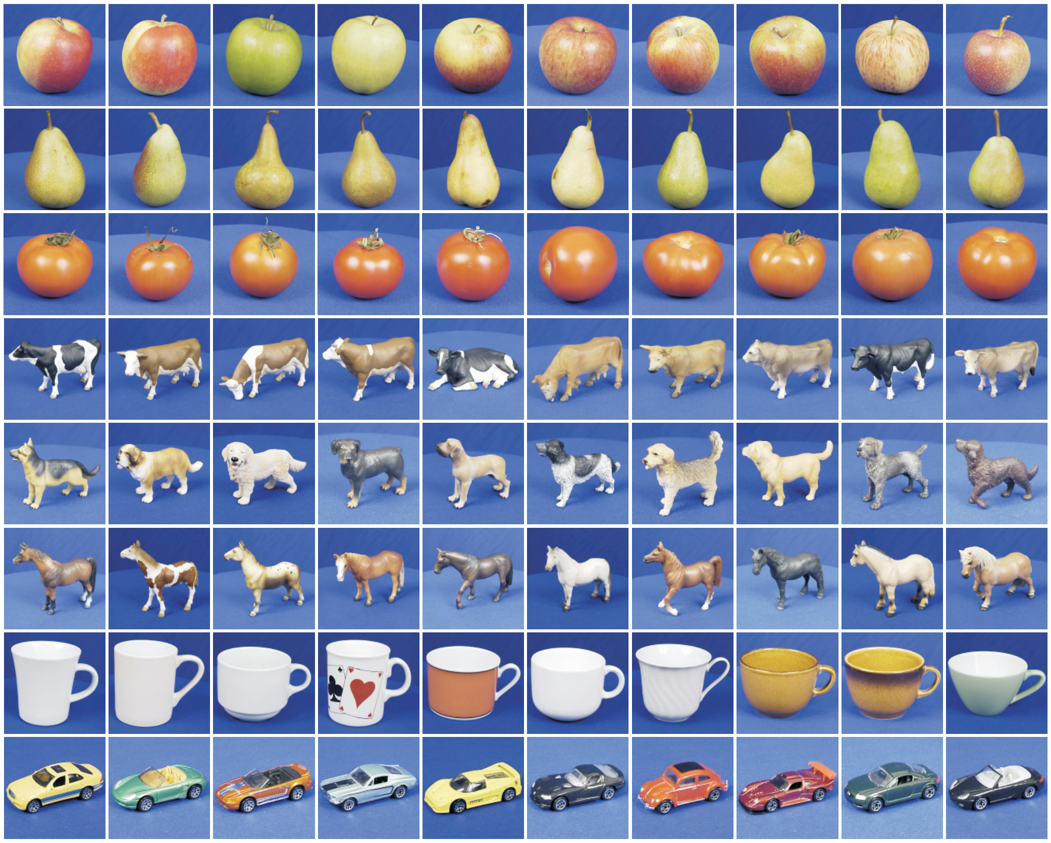


Figure 3.1: *The 8 categories of the ETH-80 database. Each category contains 10 objects with 41 views per object, spaced equally over the viewing hemisphere, for a total of 3280 images.*

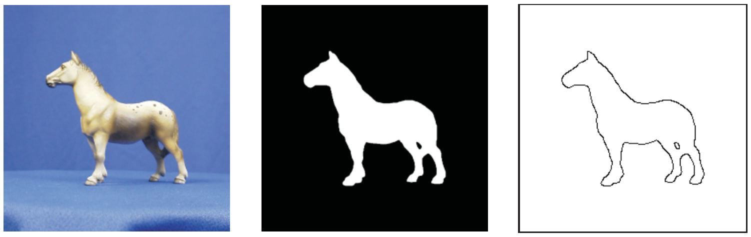


Figure 3.2: *Example database image (left), segmentation mask (middle), and extracted contour (right).*

span the category (Sclaroff, 1997). In light of the difficulty of establishing representative distributions and the effort involved in taking pictures of member objects, we resort to the second option. Figure 3.1 shows the current status of our database (in the following referred to as the CogVis-ETH80 database). For each category, we provide 10 objects that span large in-class variations while still clearly belonging to the category. Each object is represented by 41 images from viewpoints spaced equally over the upper viewing hemisphere (at distances of $22.5 - 26°$). The viewing positions were obtained by subdividing the faces of an octahedron to the third recursion level. For collecting the views, we employed an automated robot setup and a blue chromakeying background for easier segmentation. All images have been taken with a Sony DFW-X700 progressive scan digital camera with 1024×768 pixel resolution and a Tamron 6-12mm varifocal lens (F1.4). For every image, we provide a high-quality segmentation mask (Figure 3.2), so that shape and contour based methods can be easily applied. The full database is made available on our webpage[1].

The intended test mode is leave-one-object-out crossvalidation. This means we train with 79 objects and test with all views of the one unknown object. Recognition is considered successful if the correct category label is assigned. The results are averaged over all 80 possible test objects. We use the database for a best-case analysis: categorization of unknown objects under the same viewing conditions, with a near-perfect figure-ground segmentation, and known scale. In a practical application, such perfect information is seldomly available. But if an algorithm does not work under these ideal conditions, it is likely to fail in practice.

3.3 Recognition Methods

Using the database presented above, we want to compare different methods for multi-class object categorization. In particular, we want to address the question of what the role of color, texture, and shape is for this task. In this section, we introduce a selection of well-known recognition methods that are prototypical for these cues. Those methods serve as the basis for our experiments.

Color: One of the earliest appearance-based recognition methods is recognition with color histograms (Swain and Ballard, 1991). Using this approach, we collect a global RGB histogram over all image pixels belonging to the object (as specified by the segmentation mask). Two histograms V and Q can be compared using the intersection measurement

$$\cap(Q, V) = \sum_i min(q_i, v_i) \tag{3.1}$$

or the χ^2 divergence

$$\chi^2(Q, V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}. \tag{3.2}$$

The test image is then assigned to the category containing the closest matching histogram. In our experiments, we obtained the best results with a histogram resolution of 16-16-16 for the color channels and using the χ^2 measurement.

---

[1] http://www.vision.ethz.ch/pccv/

Texture: For the texture cue, we use a generalization of the color histogram approach to histograms of local grayvalue derivatives at multiple scales (Schiele and Crowley, 2000). In our experiments, we compare two versions of this approach. The first is a rotation-variant descriptor and uses only first derivatives in x and y direction over 3 different scales. The second uses rotation invariant features, namely the gradient magnitude and the Laplacian, again over 3 scales. Both the $D_x D_y$ and the Mag-Lap version have been applied to the COIL database in the past with 100% recognition rate (Schiele and Crowley, 2000). In our experiments, we obtained best results with the scales set to $\sigma_{1,2,3} = (1, 2, 4)$, 16 histogram bins per dimension, and the χ^2 measurement for histogram comparison. As shown in (Schiele and Crowley, 2000), histogram based approaches can also be used locally to recognize objects from a small set of sample points taken from the test image. In this paper, however, we use only the simpler alternative of matching histograms.

Global Shape: For the shape cues, we make a difference between global and local shape. As representatives for global shape, we use PCA-based methods (Turk and Pentland, 1991; Murase and Nayar, 1995). There are two principal ways of using PCA for recognition. In the traditional method (Murase and Nayar, 1995), one single global eigenspace for all categories is built and the training images are projected into this space. Recognition then becomes a nearest-neighbor search in the eigenspace for the closest training example. The other approach is to build separate eigenspaces for each category and measure the reconstruction error (“distance from feature space” (Turk and Pentland, 1991)), that is the quality by which the class-specific eigenspace can represent the test image. This approach can be generalized even further towards view-specific eigenspaces (Leonardis et al., 2002), which we will leave for future experiments.

The class-specific approach has the advantage that it can be extended easily to a larger number of categories – only the eigenspaces for the new classes have to be recomputed – but it is not yet known how it scales. We have made experiments with both approaches and found no significant differences in their recognition performance. Since our experiments require the recalculation of the eigenspace for every object, and the global eigenspace version takes an order of magnitude longer to compute, we only report results on the version with class-specific eigenspaces.

In two separate experiments, we apply PCA to the raw segmentation masks (“pure” global shape) and to the segmented grayvalue images. For the segmentation masks, the best recognition performance was achieved using only the first 30 eigenvectors; for grayvalue images, best results were obtained using the first 40 eigenvectors. For all PCA experiments, the images are downscaled to a size of 128 × 128 pixels. In contrast to (Murase and Nayar, 1995), we do not adapt the scale for individual views of an object such that its bounding box always fills the whole image. In our experience, the varying scales distort the eigenspace and could potentially hurt recognition performance.

	Color RGB	Texture D_xD_y	Texture Mag-Lap	PCA Masks	PCA Gray	Contour Greedy	Contour DP	Avg.
apple	57.6%	**85.4%**	80.2%	78.8%	**88.3%**	77.1%	76.3%	77.7%
pear	66.1%	90.0%	85.4%	**99.5%**	**99.8%**	90.7%	91.7%	89.0%
tomato	**98.5%**	94.6%	**97.1%**	67.8%	76.6%	70.7%	70.2%	82.2%
cow	86.6%	82.7%	**94.4%**	75.1%	62.4%	86.8%	86.3%	82.1%
dog	34.6%	62.4%	74.4%	72.2%	66.3%	**82.0%**	**82.9%**	67.8%
horse	32.7%	58.8%	71.0%	77.8%	77.3%	**84.6%**	**84.6%**	69.5%
cup	79.8%	66.1%	77.8%	**96.1%**	**96.1%**	**99.8%**	**99.0%**	87.8%
car	62.9%	**98.3%**	77.6%	**100%**	**97.1%**	**99.5%**	**100%**	90.8%
total	64.9%	79.8%	82.2%	83.4%	83.0%	86.4%	86.4%	80.9%

Table 3.1: *Recognition Results for the categorization of unknown objects.*

Local Shape: We have chosen contours as a representative feature for local shape. Over the years, numerous methods have been developed for contour-based recognition, e.g. deformable prototypes (Sclaroff, 1997) or shock graphs (Siddiqi et al., 1999; Macrini et al., 2002), to name but a few. We pick out a method based on the Shape Context proposed by Belongie (Belongie et al., 2001), which has achieved excellent results, for example for handwritten digit recognition.

In this approach, an object view is represented by a discrete set of points sampled regularly along the internal or external contours. For every point, a log-polar histogram, the *shape context*, is computed that approximates the distribution of adjacent point locations relative to the reference point. In order to achieve scale invariance, the outer radius for the histograms is typically set to the mean distance between all point pairs.

Point correspondences between different shapes can be found by matching the log-polar histograms. In their original implementation, Belongie et al. (2001) match shapes by iteratively deforming one contour using thin plate splines. Here, we compare two simpler approaches. In the first method, we search a continuous path around the main object contour using a dynamic programming approach (similar to Dynamic Time Warping). We allow that adjacent points on one contour be matched to the same point on the other contour, and that a mismatching point be skipped; but every point on one of the contours must be matched, and the overall matching order must be kept. The final score is the sum over all individual matching costs. The second approach is just a one-to-one matching between contour points using a greedy strategy. Here, the matching score is also the sum over all individual matching costs. In both cases, best results were obtained using 100 points on the contour, 5 radius and 12 sector bins, and the intersection measurement for comparing shape context histograms.

3.4 Results

In this section, the methods described above are applied to the object categorization task. As all methods depend on a set of parameters, we made a series of preliminary experiments to determine the optimal parameter settings for every method. In the following, we report only the best results.

3.4.1 Global Recognition Rates

Table 3.1 shows the recognition results for the different methods, both averaged over the whole database and broken up per category. As already mentioned in Section 3.2, the test mode is leave-one-object-out crossvalidation. So, the results always show the performance for the categorization of unknown objects. As can be seen, the contour-based methods perform best with 86.4% recognition rate. Next best are the global-shape based PCA variations with 83.41% and 82.99%, respectively. The texture histograms are only slightly behind with 82.23% for the rotation-invariant case, and 79.79% for the rotation-variant one. With only 64.85% recognition rate, color performs worst.

Globally, there is only a slight difference between the two PCA methods. However, on the category level significant differences become apparent. For the apple and tomato categories, the version with grayvalue images outperforms the mask-based version. Here, the global shape is similar for both categories, but the objects in both classes have a characteristic, class-specific texture. As a result, shape ambiguities between the categories can be resolved by additional information from the grayvalue images. For the cow, dog, and horse categories, on the other hand, the mask-based version shows better performance. Here, the global shape is again similar for all three categories. However, the ambiguities cannot be resolved by resorting to the grayvalue information encoded in the eigenspaces, because there is no characteristic texture for those categories. On the contrary, the intra-class variation for texture is so high that using localized grayvalue information actually hurts performance. The behavior of both contour based methods is similar to the one for PCA on mask images, only on a globally higher level. Between the two contour-based methods, there is no significant difference.

For the texture histograms, the rotation invariant version has a better global performance than the rotation variant one. On the per-category level, however, the methods show more distinct behaviors. Rotation variant features seem to be significantly better for the apple, pear, and car categories, that is for those objects where the relative orientation of texture elements or lines is important for recognition. For those categories that contain mainly circular texture elements (like the specularities on most of the tomatoes), or where the relative number of edge pixels on its own is a characteristic feature (as seems to be the case for the animals and cups), the rotation invariant texture descriptor gives the better results.

In general, it becomes clear that no single method is superior for all categories. Interestingly, though, almost all of the above methods are the best choice for at least

Category	Primary Feature(s)	Secondary Feature(s)
apple	PCA Gray	Texture D_xD_y
pear	PCA Gray / Masks	
tomato	Color	Texture Mag-Lap
cow	Texture Mag-Lap	Contour / Color
dog	Contour	
horse	Contour	
cup	Contour	PCA Gray / Masks
car	PCA Masks / Contour	Texture D_xD_y

Table 3.2: *Best primary and secondary features for our categories, as derived from the recognition results.*

one category. For example, the global color distribution, which is in general not a characteristic feature for many basic-level categories, still performs well for cows and tomatoes. From this we can conclude that for multi-class object categorization, we need multiple features and different combinations of features. Table 3.2 shows a list of the most discriminative primary and secondary features for our categories (achieving best and second-best recognition results).

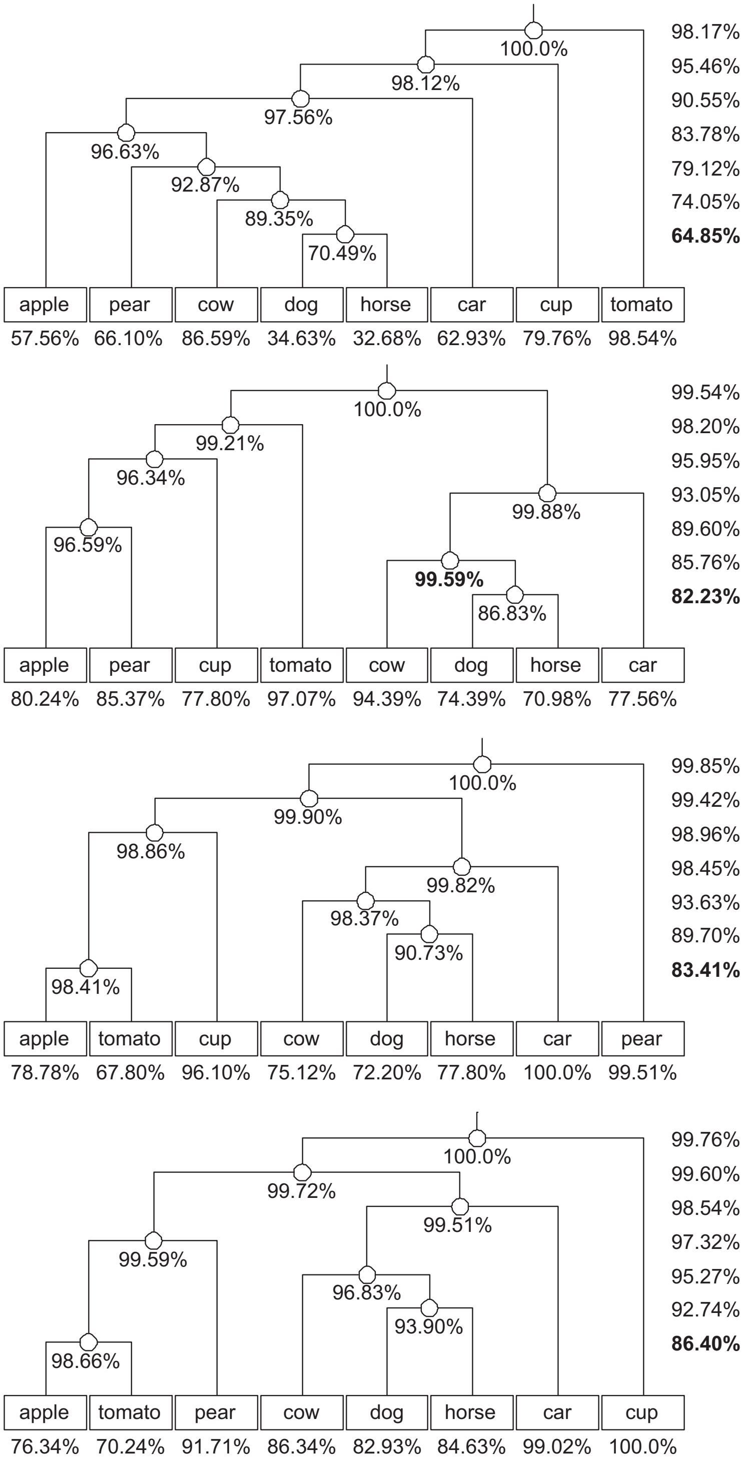
3.4.2 Confusions

In Section 3.1, we have stated the need for graceful degradation of an object categorization system. We therefore want to evaluate which objects are treated as similar or are confused by the different methods. We hope this can shed more light onto how the methods perform and how they may generalize to larger tasks with more categories.

In order to examine this more closely, we look at the confusion matrix for each method. By iteratively grouping together those categories that are confused most often, we obtain a hierarchy of groupings. Figure 3.3 shows the grouping hierarchies for color, rotation-invariant texture, PCA on segmentation masks, and contours. As can be seen from these diagrams, the contour-based method results in the most intuitive hierarchy, grouping together both the fruits and the animals. Both PCA and texture succeed in grouping together the animals, but manage only two of the three fruit categories. Interestingly, those groupings are different for the two cues: apples and tomatoes are treated as similar in terms of global shape; apples and pears in terms of texture. As could be expected, color again performs worst.

The out-of-class confusions that occurred most often in our experiments are cows with cars for the shape and contour cues, and apples with cups for texture. These are mainly degenerate views from above, where a cow has a roughly rectangular outline, or from a medium height, where the cup handle is not visible and only an ambiguous shape remains. In real-world situations and with unconstrained viewpoints, such confusions are likely to appear.

Interestingly, rotation-invariant texture is the cue that best groups the animal



(a) Color RGB

(b) Texture Mag-Lap

(c) PCA Masks

(d) Contour Greedy

Figure 3.3: *Grouping hierarchies for four different cues. The diagrams show, when read from bottom to top, the best groupings for each cue. At each node the local recognition rate for this grouping is displayed. The numbers to the right show the global recognition rate after the groups are split.*

categories together. When all animals are taken for a single class, this cue can recognize them with 99.59% accuracy – significantly better than it is possible with global shape or contours. It only fails when trying to distinguish the individual types of animals.

3.5 Multi-Cue Combination

The results from our experiments stress the need for multi-cue combination. In the following, we want to examine the combination potential in more detail. In particular, we are interested in how much the categorization can profit from adding a certain cue when some other cues are already available. As an evaluation tool, we use a decision tree (Duda et al., 2001) that at each level bases its decisions on one cue only.

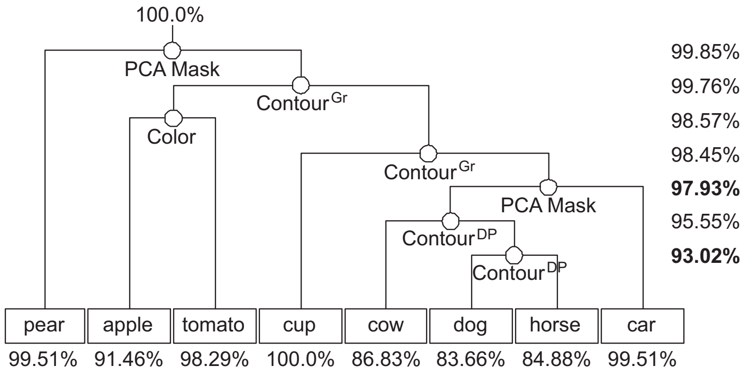
Starting again from the confusion matrices, we seek an optimal partition of the categories that minimizes the number of misclassifications. We then make our decision based on the cue that produces the best partition and iteratively refine the resulting group of categories. For this, we have to recompute the confusion matrices for all cues while leaving out those views that have already been misclassified. In this example, we stop at the category level, but we expect that the results can be improved when the approach is pursued down to a view or aspect level.

Figure 3.4 shows the resulting optimal decision trees for the case where all cues are available, and for the case where local shape is not. The performance for the first case is clearly better, with 93.02% recognition rate compared to 89.97% for the second case. However, both versions are comparable up to the point where the individual animal categories need to be distinguished. Here, the main difference occurs, and 3% performance is lost because the other cues are not as good at separating the animals. Using only color and texture and no shape information at all, the performance is significantly worse with only 86.4% combined recognition rate (not shown). This confirms that both global and local shape are important cues for object categorization.

3.6 Discussion

In this work, we have analyzed the performance of several state-of-the-art recognition methods for the more general task of multi-class object categorization. As basis for our analysis, we have introduced a new database containing several categories and both object appearances and segmentation masks. This allows to evaluate both appearance- and contour-based methods in the same setting and bring the formerly separate communities a bit closer together. That there is a potential for mutual benefit can be seen from our results. Contours proved to be the best single cue for the categories in our database, followed by global shape and (rotation invariant) texture descriptors. What is even more important, though, is that every cue we

(a) With all cues



(b) Without contours

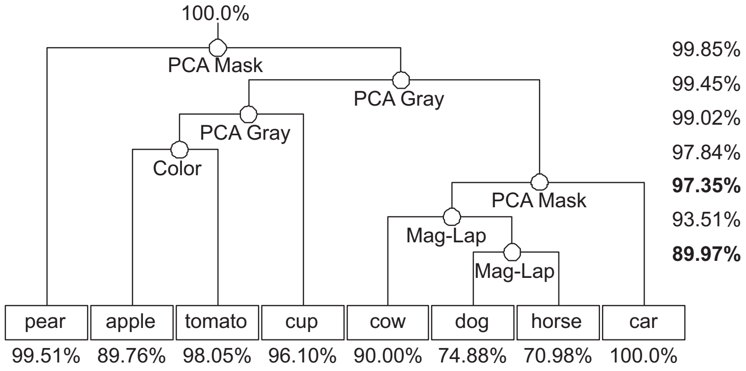


Figure 3.4: *Optimal multi-cue decision trees when all cues are available (top) and when local shape is not (bottom). The numbers to the right of each tree show the global recognition rate after each split. Note that the performance of both trees differs significantly only for distinguishing the animal categories.*

tested turned out to be the best choice for at least one category. This shows that there is significant potential for improvement by using multiple cues.

In the second part of our analysis, we have demonstrated how this potential can be used in the form of a multi-cue decision tree. Using all available cues, we were thus able to improve the global recognition rate from 86.4% to 93%. Contours again played an important role in this improvement. Without them, the recognition rate could only be increased to about 90%, mostly because the remaining cues were not able to distinguish the different animal categories. Without both contours and global shape, recognition performance could only be increased from 83.4% to 86.4% – a performance the contour-based methods achieved on their own. This emphasizes the importance of shape-based cues for object categorization.

It is important to bear in mind that this work shows a best-case analysis. Transferring methods from a lab setting to the real world is not a trivial task, and it may well be that some necessary features cannot be extracted in sufficient quality for a particular method to work. What we can deduct from the experiments is an opposite argument: if a method does not achieve good results under our idealized conditions, it is likely to fail in practice. In that respect, our finding that no single method achieved over 87% recognition rate is an even stronger argument for the necessity of multiple cues.

The results of our study have shown than global appearance and shape cues are important for discriminating between multiple basic-level categories. When working with real-world images, however, these cues are often not available or cannot be extracted reliably. In addition, the objects of interest often take up only a small portion of the image, so that they need to be localized and separated from the background before any global measure can be computed. Local features, on the other hand, can be extracted more reliably from real-world images and are also less vulnerable to clutter and occlusion. They are thus better-suited for detecting categorical objects under real-world conditions. The following four chapters will therefore focus on developing a local approach for object detection and figure-ground segmentation. Chapter 8 will then discuss how the results of this stage can be used to integrate also global cues, which have a higher potential for multi-category discrimination.

4

Codebook Representations

The first task of any local-feature based approach is to determine which features in the image correspond to which object structures. In Computer Vision, this is known as the *correspondence problem*. For detecting and identifying known objects, this translates to the problem of robustly finding exactly the same structures again in new images under varying imaging conditions (Schmid and Mohr, 1996; Lowe, 1999, 2001; Obdrzalek and Matas, 2002; Rothganger et al., 2003; Ferrari et al., 2004). As the ideal appearance of the model object is known, the extracted features can be very specific. In addition, the objects considered by those approaches are often rigid, so that the relative feature configuration stays the same for different images. Thus, a small number of matches typically suffices to estimate the object pose, which can then in turn be used to actively search for new matches that consolidate the hypothesis (Lowe, 1999, 2001; Rothganger et al., 2003; Ferrari et al., 2004).

When trying to find objects of a certain category, however, the task becomes more difficult. Not only is the feature appearance influenced by different viewing conditions, but both the object composition (i.e. which local structures are present on the object) and the spatial configuration of features may also vary considerably between category members. In general, only very few local features are present on all category members. Hence, it is necessary to employ a more flexible representation.

In this chapter, we introduce the first level of such a representation. As basis, we use an idea inspired by the work of Burl et al. (1998), Weber et al. (2000a,b), and Agarwal and Roth (2002). We build up a vocabulary (in the following termed a *codebook*) of local appearances that are characteristic for an object category by sampling local features that repeatedly occur on a set of training images of this category. Features that are visually similar are grouped together in an unsupervised clustering step. The result is a compact representation of object appearance in terms of which novel images can be expressed.

Codebook representations have become a popular tool for object categorization recently, and many approaches use variations of this theme (Burl et al., 1998; Weber et al., 2000a,b; Fergus et al., 2003; Li et al., 2003; Agarwal and Roth, 2002; Borenstein and Ullman, 2002, 2004; Ullman et al., 2002; Vidal-Naquet and Ullman, 2003). However, there are still large differences in how the grouping step is performed, how the matching uncertainty is represented, and how the codebook is later used for

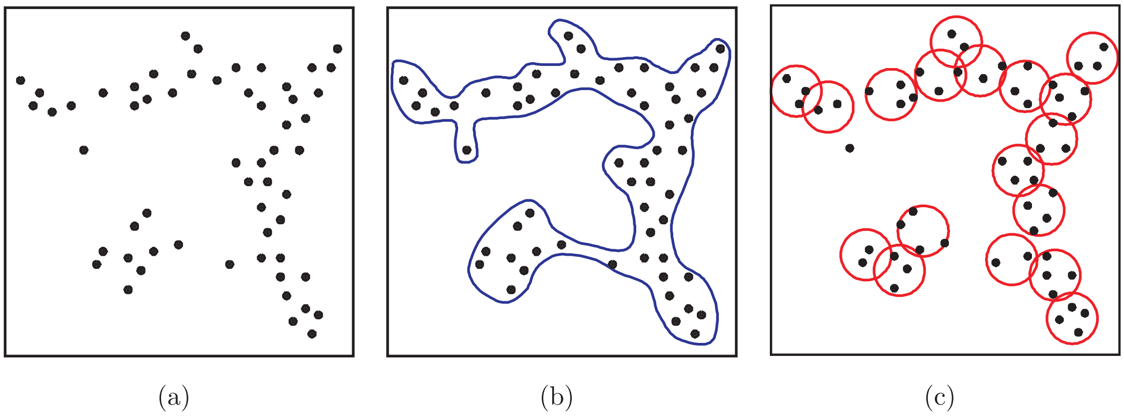


Figure 4.1: *Visualization of the codebook representation. Let the points in (a) be observed samples from the appearance distribution of some object part. Instead of trying to find a complex decision surface that separates all part appearances from non-part appearances (b), the codebook approach represents the appearance distribution by a set of compact prototypes (c). For each individual prototype, the matching decision is a simple distance threshold, and a point is classified as belonging to the appearance distribution if it is close enough to any prototype.*

recognition.

In the following section, we describe our codebook representation in more detail. We pay specific attention to the question how the matching uncertainty can be modelled. This allows us to formulate an optimality criterion based on which the codebook quality can be judged. Section 4.2 then gives an overview of different clustering methods and compares their relative performance. As the clustering step will be applied to large amounts of data (with 5,000–100,000 local features), an efficient implementation is crucial. Section 4.3 therefore describes efficient algorithms that can be used for this problem. A final discussion concludes the chapter.

4.1 A Codebook of Local Appearance

In order to gain a better understanding of the codebook principle, it is helpful to contrast it with part-classifier based approaches. Several recent methods manually define a small set of object parts and then use a complex classifier to learn them (Mohan et al., 2001; Heisele et al., 2001; Ronfard et al., 2002; Mikolajczyk et al., 2004; Kruppa, 2004). The focus of these approaches is on building robust detectors for semantically meaningful parts, so that the presence of an object can already be inferred from few part detections. As those parts are mostly defined by their semantic label, not their visual appearance, they may exhibit complex appearance variations. Consequently, these approaches typically require a large number of positive and negative training examples with precise alignment.

The codebook representation follows a different approach. Its key idea is to automatically learn a relatively large number of simple and compact appearance prototypes and represent the complex appearance distribution in relation to them. Figure 4.1 illustrates this concept. Let the points in Fig. 4.1(a) be observed samples from the appearance distribution of some object part. Instead of trying to find a complex decision surface that separates all part appearances from non-part appearances (Fig. 4.1(b)), the codebook approach represents the appearance distribution by a set of compact prototypes (Fig. 4.1(c)). For each individual prototype, the matching decision is a simple distance threshold, and a point is classified as belonging to the appearance distribution if it is close enough to any of the prototypes.

The possibility to learn prototypes in an unsupervised way by clustering makes it possible to use far more object parts than when the parts need to be manually specified (and annotated) by an expert. This is especially important for object categories where only a small number of semantically meaningful parts can be found. While each individual prototype conveys less information about the object, their greater number and the resulting denser cover of the object can compensate for that.

Another important consequence is that by using multiple prototypes, the appearance can be represented on a finer level, and different appearances of the same semantic object part may be treated differently. For example, some instances of a car wheel might be highly discriminant, while others might be readily mistaken for other object parts. The codebook approach allows to separate these two cases and model the matching uncertainty for each case individually. In the following, we describe how the codebook generation process is implemented.

4.1.1 Codebook Generation

We start by applying an interest point detector to obtain a set of informative locations for each image. By extracting features only from those locations, the amount of data to be processed is reduced, while the interest point detector's preference for certain structures assures that "similar" regions are sampled on different objects. Several different interest point detectors are available for this purpose. Here, we use a simple Harris detector, which prefers corner-type structures (Harris and Stephens, 1988, see also Appendix A).

Around each interest point, we extract an image patch of size 25 × 25 pixels. Throughout this thesis, the extracted image patches are directly used as features (Note, however, that the approach is not restricted to this choice – in principle, any local feature could be used). Figure 4.2 shows the extracted interest points and patches for two example images. As can be seen from those examples, the sampled information provides a dense cover of the object, leaving out only uniform regions. This process is repeated for all training images, and the extracted patches are collected.

Next, we group visually similar features to create a codebook of prototypical local appearances. The similarity between two patches is measured by Normalized

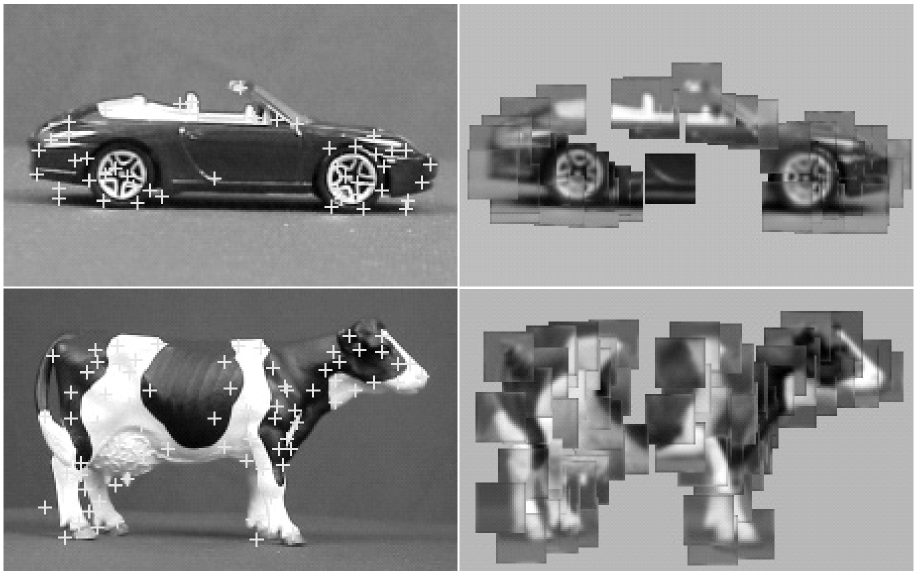


Figure 4.2: *Local information used in the codebook generation process. (left) Harris interest points; (right) patches extracted around the interest points. Altogether,* 56 *patches are found for the car and* 85 *for the cow example.*

Grayscale Correlation (NGC):

$$NGC(p,q) = \frac{\sum_i (p_i - \overline{p_i})(q_i - \overline{q_i})}{\sqrt{\sum_i (p_i - \overline{p_i})^2 \sum_i (q_i - \overline{q_i})^2}}, \tag{4.1}$$

which is equivalent to simple correlation for zero-mean unit-variance normalized data:

$$NGC(p,q) = \tfrac{1}{d} \sum_i p_i q_i \qquad \text{if} \quad \sum_i p_i \overset{!}{=} 0, \quad \tfrac{1}{d} \sum_i (p_i)^2 \overset{!}{=} 1. \tag{4.2}$$

In order to keep the representation as simple as possible, we represent all patches in a cluster by their mean, the cluster center. Of course, a necessary condition for this is that the cluster center is a meaningful representative for the whole cluster. In that respect, it becomes evident that the goal of the grouping stage must not be to obtain the smallest possible number of clusters, but to ensure that the resulting clusters are visually compact and contain the same kind of structure. This is an important consideration to bear in mind when choosing the clustering method. Before we do that, however, we first need to discuss how the matching uncertainty shall be represented in the later system.

4.1.2 Representing Uncertainty

The purpose of the codebook is to serve as an internal vocabulary in terms of which novel images are expressed. By matching the a-priori unknown image content to this vocabulary, the system can take advantage of discriminatory properties and structural relations it has learned for the individual codebook entries.

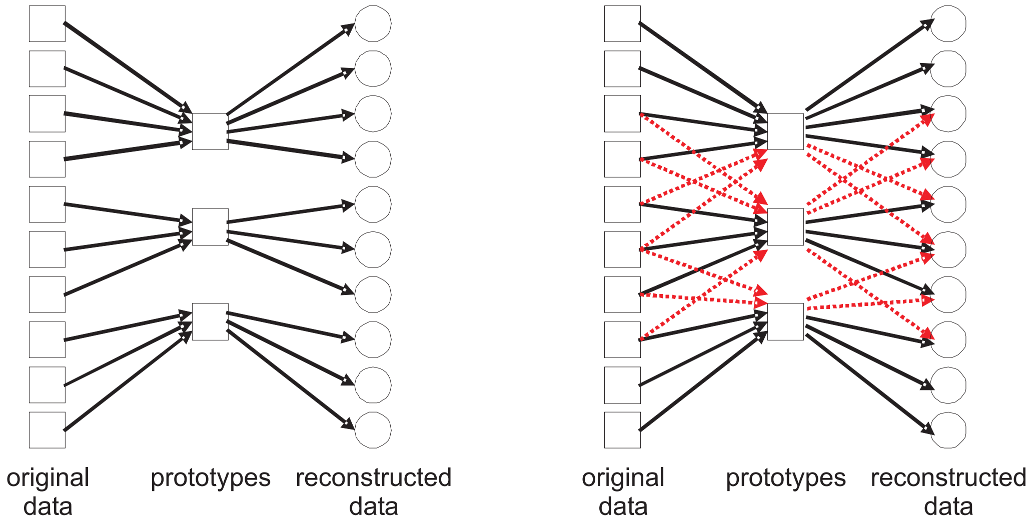


Figure 4.3: *Visualization of the codebook matching process, (left) when each patch is assigned only to the closest matching prototype; (right) when the assignment is interpolated between all sufficiently similar prototypes. The arrows on the left side of each diagram show which training examples are associated with each codebook entry. The arrows on the right then demonstrate the system's response when a codebook entry is activated. As becomes evident from the diagrams, the interpolated version is more stable and allows for robust performance when the patch appearance is slightly altered.*

When pursuing such an approach, however, it is important to represent uncertainty on all levels: while matching the unknown image content to the known codebook representation; and while accumulating the evidence of multiple such matches, e.g. for inferring the presence of the object. As Gibson (1957) put it

> *"...the percept is always a wager. Thus uncertainty enters at* two *levels, not merely one: the configuration may or may not indicate an object, and the cue may or may not be utilized at its true indicative value."*

Applied to our context, this means that we need to represent the uncertainty that is invariably associated with the matching process and propagate it to later stages. In order to use spatial relations later on, we need to know not only that an image patch contains object structure, but also which part of the appearance distribution (i.e. which codebook entrie(s)) it corresponds to.

Figure 4.3 illustrates two different ways how this information can be obtained. In the ideal case, the clustering has been performed in a way that each patch can be uniquely assigned to a single codebook entry. In this case, the logical choice would be to activate only the nearest neighbor (as shown in the left image). However, it is unrealistic to expect such clean data in practical applications. It is far more likely that often several codebook entries are more or less similar to a certain patch. Under those circumstances, a nearest-neighbor matching scheme will quickly become unstable, since small changes in appearance may cause the nearest-neighbor assignment to suddenly switch in favor of a different codebook entry.

The robustness of the matching process can be increased if hard decisions are avoided and an image patch is associated with all sufficiently similar codebook entries (as the right image in Fig. 4.3 demonstrates). In this case, the system response is an interpolation between the responses of all "activated" codebook entries, and each such entry contributes to the final result according to the strength of its activation. Consequently, even when the patch appearance is slightly altered, its codebook activation pattern will change only gradually.

This point can be corroborated also by a different argument. The goal of the whole codebook matching stage is to represent the high-dimensional appearance distribution of object parts. As Edelman (1999, pp. 104–105) argues, the performance of a nearest-neighbor classification scheme for this task depends on the degree to which the memorized examples cover the measurement space. Thus, the necessary number of training examples for a given performance grows exponentially with the dimensionality d of the data, which makes the approach infeasible already for moderate values of d. However, in most cases the relevant data is not spread out over the full space, but it is contained in a relatively low-dimensional and smooth manifold, which an interpolation scheme might take advantage of. By interpolating between several prototypes instead of pursuing a nearest-neighbor approach, the number of examples required for valid generalization can therefore be reduced by orders of magnitude (for some numerical examples, see (Stone, 1982)).

So, instead of activating only the nearest neighbor, we associate each patch with all codebook entries to which it can be matched (where a "match" means that the similarity between patch and codebook entry is greater than a threshold t). If a certain cluster is clear-defined and does not overlap with any other cluster, the corresponding codebook entry will always be activated as the only match. If, on the other hand, several clusters overlap in the appearance space, they will sometimes be activated together. Such cases therefore indicate where confusions are likely to occur.

The interpolation strategy increases the robustness when patches are matched to the codebook. In order to profit from this also for the final recognition stage, we need to propagate the knowledge about possible confusions. For this, we propose the following approach. We record, for each codebook entry, the information for which patches from the training set it was activated. The stored activation records are then used to determine the system's response when the same codebook entry is later found during recognition. Thus, if a training patch matches to two codebook entries (as indicated with the red arrows in Fig. 4.3(right)), each of those codebook entries will keep a record of this match. When any of the two codebook entries is later activated during recognition, it will produce responses for all of its stored activations, including the one from said training patch. This way, codebook entries that are likely to get confused will also overlap in part of their responses. In Chapter 5, we use this property to robustly estimate the spatial occurrence distribution of codebook entries.

4.1.3 Optimality Criterion

The proposed procedure provides a way to compensate for the confusions that invariably occur in real-world codebooks by storing the activation patterns observed on training images. Later experiments will show that this step allows for robust performance even when the clustering is not ideal. However, it is clear that the less a codebook is tuned to the right structures, the more confusions will occur, and the more activations will have to be stored to compensate for them. Since a larger number of stored activations also increases the effort during recognition, the representative quality of a codebook can be judged by the number of activations it produces. The goal is thus to find a grouping such that, for a given threshold t,

- all patches assigned to a cluster can still be matched to the cluster center with a similarity of at least t;
- as few patches as possible that are not assigned to a cluster can be matched to this cluster center with a similarity greater than t;
- the number of clusters is minimized.

The first condition guarantees that no information is lost during the clustering process, while the second tries to enforce codebook specificity and thus reduce the effort of modelling uncertainty. Together, this gives us a measure to compare different clustering algorithms and evaluate their suitability for codebook generation.

4.2 Clustering Methods

A core part of the codebook idea is to find features that are typical and representative for a given object category. This is done by grouping features that occur repeatedly on the training images into a set of appearance prototypes. This section reviews different clustering methods that can be used for this purpose.

4.2.1 K-means Clustering

The k-means algorithm (MacQueen, 1967) is one of the simplest and most popular clustering methods. It pursues a greedy hill-climbing strategy in order to find a partition of the data points that optimizes a squared-error criterion. The algorithm is initialized by randomly choosing k seed points for the clusters. In each following iteration, each data point is assigned to the closest cluster center. When all data points have been assigned, the cluster centers are recomputed as the means of all associated data points. In practice, this process converges to a local optimum within a few iterations.

Many algorithms employ k-means clustering because of its computational simplicity, which allows to apply it to very large data sets (Weber et al., 2000a,b; Sivic and Zisserman, 2003, 2004; Sivic et al., 2004). Its time complexity is $O(Nk\ell d)$, where N is the number of data points of dimensionality d; k is the desired number of clusters; and ℓ is the number of iterations until the process converges.

However, k-means clustering has several known deficiencies. Firstly, it requires the user to specify the number of clusters in advance. Secondly, there is no guarantee that the obtained clusters are visually compact. Because of the fixed value of k, some cluster centers may lie in-between several "real" clusters, so that the mean image is not representative of all grouped patches. Last but not least, the k-means procedure is only guaranteed to find a local optimum, so the results may be quite different from run to run.

4.2.2 Agglomerative Clustering

Other approaches therefore use agglomerative clustering schemes, which automatically determine the number of clusters (Agarwal and Roth, 2002; Leibe and Schiele, 2003b). However, both the runtime and the memory requirements are often significantly higher for agglomerative methods. Especially the memory requirements impose a practical limit, since many agglomerative methods require an $O(N^2)$ similarity matrix to be stored, which means that the typically available main memory on current machines is exceeded already for $N >$ 15–25,000. Therefore, these algorithms are usually used only with small data sets.

All hierarchical agglomerative methods follow the same principle. Starting with each patch as a separate cluster, the two most similar clusters c_r and c_s are merged as long as a similarity criterion between their constituent patches is fulfilled. Different choices for this criterion give rise to the different clustering methods. It has become practice to classify these choices by the way the similarity of a newly formed cluster to all other clusters is updated. Following Lance and Williams (1967), the similarities between a newly-formed cluster $(c_r \cup c_s)$ and an existing cluster c_k can be expressed by the general formula

$$\begin{aligned} sim(c_k, c_r \cup c_s) &= \alpha_r sim(c_k, c_r) + \alpha_s sim(c_k, c_s) + \beta sim(c_r, c_s) \\ &\quad + \gamma |sim(c_k, c_r) - sim(c_k, c_s)|. \end{aligned} \tag{4.3}$$

Table 4.1 gives an overview of the most common hierarchical clustering methods and their choice for the parameters $\alpha_r, \alpha_s, \beta$, and γ[1].

The well-known single-link and complete-link algorithms use the maximum and minimum, respectively, of the similarities between the pairs (c_k, c_r) and (c_k, c_s). However, both extremes tend to favor degenerate cases, which makes them unsuitable for our problem.

The next four methods try to improve on this by using different variations of averaged similarities. They are commonly subsummized as average-link clustering

---

[1] Note that the differing signs for some of the table entries, compared to (Lance and Williams, 1967), are due to our formulation in terms of *similarities* instead of *dissimilarities*.

Clustering Method	α_r	α_s	β	γ
Single-link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Complete-link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
UPGMA (group average)	$\frac{n_r}{n_r+n_s}$	$\frac{n_s}{n_r+n_s}$	0	0
WPGMA (weighted average)	$\frac{1}{2}$	$\frac{1}{2}$	0	0
UPGMC (unweighted centroid)	$\frac{n_r}{n_r+n_s}$	$\frac{n_s}{n_r+n_s}$	$\frac{-n_r n_s}{(n_r+n_s)^2}$	0
WPGMC (weighted centroid)	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
Ward's method (minimum variance)	$\frac{n_r+n_k}{n_r+n_s+n_k}$	$\frac{n_s+n_k}{n_r+n_s+n_k}$	$\frac{-n_k}{n_r+n_s+n_k}$	0

Table 4.1: *Hierarchical clustering methods, according to the Lance-Willams update scheme (Lance and Williams, 1967; Jain and Dubes, 1988). The methods are characterized by the way they express the similarity between a newly-formed cluster* ($c_r \cup c_s$) *and an existing cluster* c_k. *The parameters* $\alpha_r, \alpha_s, \beta$, *and* γ *correspond to the weights for the different terms in Eq. 4.3, and* n_r, n_s, *and* n_k *are the sizes of the respective clusters.*

(Jain and Dubes (1988) also call them *pairwise grouping methods* – therefore the acronym PGM). Differences exist whether the similarity is expressed as *arithmetic average* between all constituent data points (denoted by the suffix A), or as the similarity between the cluster *centroids* (denoted by C). A further distinction is made if the computation is *unweighted* (prefix U), meaning that all points in a cluster are treated equally, or *weighted* (prefix W). While for the Weighted Average and Weighted Centroid criteria, the weight of a cluster depends on its history of previous merging steps, the unweighted criteria can also be written in a compact form:

$$\text{Group Average:} \qquad sim(X,Y) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} sim(x^{(i)}, y^{(j)}) \tag{4.4}$$

$$\text{Unweighted Centroid:} \qquad sim(X,Y) = sim(\mu_x, \mu_y) \tag{4.5}$$

Using this formulation, the two methods can be interpreted more intuitively: clusters are merged as long as the average similarity between their constituent data points or the similarity between their centroids stays above a certain threshold t.

Ward's method (Ward, 1963), finally, pursues a global optimization criterion and tries to minimize the within-cluster variation (Jain and Dubes, 1988)). This method has been shown to be strongly related to deterministic annealing optimization algorithms (Hofmann, 1997).

In this work, we focus on average-link clustering with the Group Average criterion (UPGMA). As we will see later on, this criterion has several nice properties, which allow it to be very efficiently computed. In particular, Section 4.3.2 will present an algorithm that performs average-link clustering with $O(N^2 d)$ time and $O(N)$ space complexity, making it thus applicable to large data sets.

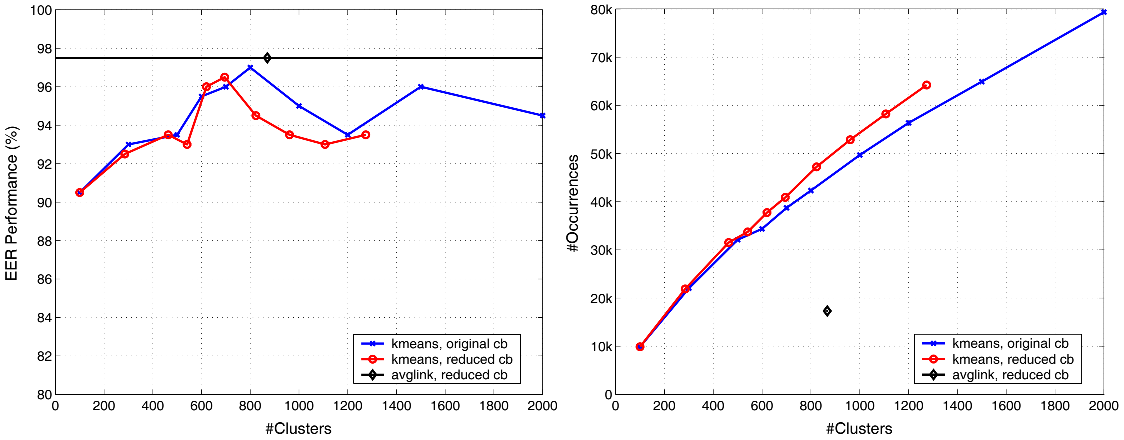


Figure 4.4: *Comparison of codebooks built with k-means and average-link clustering. (left): Recognition performance on a car detection task (the corresponding experiment is described in detail in Chapter 6). (right): Number of stored activations needed to represent the matching uncertainty. As can be seen from the plots, the k-means codebook achieves nearly the same performance as the one built by average-link clustering, but it requires more than twice as many activations to be stored.*

4.2.3 Experimental Comparison

In order to evaluate the different clustering methods, we apply them to the same data set and compare the suitability of the resulting codebook for recognition. The evaluation is based on two criteria. One is the representational quality of the codebook, as judged by our optimality criterion from Section 4.1.3. The other is the recognition performance it allows. As this performance can only be judged in the context of a full recognition system, we skip some steps for the purpose of this experiment and use the algorithm that will be developed in Chapters 5 and 6.

The task is to detect side views of cars in real-world images. Starting from a training set of 100 images containing one car each, a total of 6,413 patches are extracted with the Harris interest point operator. Average-link clustering with the NGC similarity measure and a value[2] of $t = 0.7$ produces 2,104 visually compact clusters. However, 1,241 of these clusters contain only one patch, which means that they do not correspond to any repeating structure. We therefore discard those clusters and keep only the remaining 863 prototypes[3]. In comparison, k-means clustering is executed with different values for k ranging from 100 to 2,000. In addition to the original codebooks, we also try the codebook reduction step and measure the performance when single-patch clusters are removed.

---

[2] We kept this value constant for all our experiments, since it produces good visual clustering results.

[3] Note, however, that although the discarded patches are not explicitly encoded in the codebook, they are in most cases still sufficiently similar to codebook entries, so that they are reflected in the stored activation patterns.

Figure 4.4 shows the results of this experiment. In the left diagram, the recognition performance is plotted as a function of the codebook size. The details of the recognition algorithm and what exactly the performance measure means will be explained in depth in Chapter 6. What is important at this point is that the codebook obtained by k-means reaches approximately the same performance as the one obtained by average-link clustering when k is set to a similar number of clusters. This is the case for both the original and the reduced k-means codebook (without single-patch clusters).

However, as can be seen from the right diagram, the number of activations for this codebook size is more than twice as high for k-means as for average-link clustering. All in all, the k-means codebook with $k = 800$ clusters generates 42,310 activations from the initial 6,413 training patches, while the more specific average-link codebook can represent the full appearance distribution with only 17,281 activations. Since the number of activations determines the efficiency of a later recognition system, the average-link codebook is clearly preferable.

We can thus draw the following conclusions. First, the experiment confirms that the proposed uncertainty modelling stage can indeed compensate for a less-specific codebook. As can be seen from Figure 4.4(left), the recognition performance degrades gracefully for both smaller and larger values of k. This result has important consequences for the scalability of our approach, since it indicates that the method can be applied even to cases where no optimal codebook is available.

Second, the experiment shows that it pays off to spend more effort on the clustering stage, since this will reduce the complexity of a later recognition system. The visually more compact clusters produced by average-link clustering are better suited to our problem than the partition obtained by k-means, as can be seen from the number of activations both codebooks generate. For this reason, we will only consider average-link clustering for all future experiments. The following section shows how it can be efficiently computed.

4.3 Efficient Implementation

Given the large amounts of data that need to be processed, an efficient implementation of the clustering algorithm is not only a nice extension, but indeed crucial for its applicability. This is particularly true for the algorithm's space requirements. The standard average-link algorithm, as found in most textbooks and as described in the following section, requires a quadratic similarity matrix to be stored. In practice, this means that the algorithm is only suitable for up to 15–25,000 input points on today's machines. After that, its space requirements outgrow the size of the available main memory, and the algorithm incurs detrimental page swapping costs.

Fortunately, it turns out that for special choices of the clustering criterion and similarity measure, including the one we are using, a more efficient algorithm is available that runs in $O(N^2 d)$ and needs only $O(N)$ space. Since this algorithm has so far been little known in the Computer Vision community, Section 4.3.2 will

Algorithm 4.1 The standard Average-Link clustering algorithm.

```
// Compute pairwise distances and store them in the matrix $S_{ij}$.
for all pairs of points $(p_i, p_j)$ do
  $S_{ij} \leftarrow sim(p_i, p_j)$
  if $S_{ij} > t$ then
    $P$.insert( $(S_{ij}, i, j)$ )                                (1)

// Perform the clustering
$k \leftarrow N$
while not finished do
  $(s, i_1, i_2) \leftarrow P$.first()
  while not valid($i_1$) or not valid($i_2$) do
    $(s, i_1, i_2) \leftarrow P$.first()                          (2)

  if $s \leq t$ or $P$ is empty then
    finished ← true
  else
    $k \leftarrow k + 1$
    $C_k \leftarrow C_{i_1} \cup C_{i_2}$                          (3)
    declare $C_{i_1}, C_{i_2}$ invalid, $C_k$ valid

    for all clusters $C_i$ with valid($i$) do
      $s \leftarrow 0$
      for all points $p_{i_1} \in C_i$ do
        for all points $p_{i_2} \in C_k$ do
          $s \leftarrow s + S_{i_1 i_2}$                          (4)
      $s \leftarrow s/(|C_i||C_k|)$
      if $s > \theta$ then
        $P$.insert( $(s, k, i)$ )                                 (5)
```

describe its derivation in more detail.

4.3.1 Standard Algorithm

We begin by introducing the standard average-link algorithm, as it can be found in many textbooks. This algorithm can be traced back to Day and Edelsbrunner (1984). It is general in that it works also for cases when the data items do not reside in a vector space (as, for example, is often the case in document retrieval). The only requirement is that a pairwise similarity measure can be computed between data items.

Algorithm 4.1 shows the steps of this clustering procedure. The algorithm starts by computing all pairwise distances between the input points and storing them in a similarity matrix S_{ij}. Simultaneously, it builds up a priority queue P containing all candidate pairs of clusters with a similarity greater than t (1). In each following iteration of the clustering process, the first element of P is retrieved (2), and the corresponding clusters are merged (3). Next, the algorithm recomputes the similarity of the newly-merged cluster to all other clusters (4) and updates P with the new similarities (5). This process is repeated until no candidate pair with a similarity above t is found.

It can easily be seen that the algorithm's time complexity is $O(N^2(d + \log N))$, since a maximum of $O(N^2)$ pairwise similarities are stored in the priority queue and each insertion and removal operation is logarithmic in the priority queue's length[4]. As the similarity matrix is kept in memory for efficient similarity recomputation after a merging step, the algorithm's space complexity is $O(N^2)$ [5].

4.3.2 RNN Algorithm

The main complexity of the standard algorithm comes from the effort to ensure that clusters are merged in the right order. The improvement presented in this section is due to the insight by de Rham (1980) and Benzécri (1982) that for some clustering criteria, the same results can be achieved also when specific clusters are merged in a different order.

The algorithm is based on the construction of *reciprocal nearest neighbor* pairs (RNN pairs), that is of pairs of points a and b, such that a is b's nearest neighbor and vice versa (de Rham, 1980; Benzécri, 1982). It is applicable to clustering criteria that fulfill Bruynooghe's *reducibility property* (Bruynooghe, 1977):

$$d(c_i, c_j) \leq \inf(d(c_i, c_k), d(c_j, c_k)) \Rightarrow \inf(d(c_i, c_k), d(c_j, c_k)) \leq d(c_i \cup c_j, c_k). \quad (4.6)$$

The reducibility property effectively states that the agglomeration of a reciprocal nearest-neighbor pair does not alter the nearest-neighbor relations of any other cluster. It is easy to see that this property is fulfilled, among others, for the Group Average criterion (regardless of the employed similarity measure) and the Centroid criterion based on correlation (however, it is not fulfilled for the Centroid criterion based on Euclidean distances).

As soon as an RNN pair is found, it can be agglomerated (a complete proof that this results in the correct clustering can be found in (Benzécri, 1982)). The key to an efficient implementation is thus to ensure that RNNs can be found with as little recomputation as possible.

This can be achieved by building a *nearest-neighbor chain* (Benzécri, 1982). An NN-chain consists of an arbitrary point, followed by its nearest neighbor, which is again followed by its nearest neighbor from among the remaining points, and so on.

---

[4] As presented in Algorithm 4.1, the priority queue may reach in the worst case even a length of $O(N^2 \log N)$ entries, but in practice, this is almost never the case. Although this could be easily reduced to $O(N^2)$ by adapting the algorithm, we have decided against it for the sake of readability.

[5] Even when the more efficient update equations from Table 4.1 are used, the space requirements are still quadratic

Algorithm 4.2 The RNN algorithm for Average-Link clustering with nearest-neighbor chains.

```
// Start the chain $L$ with a random point $v \in V$.
$last \leftarrow 0$
$L[last] \leftarrow v \in V$;   $R \leftarrow V \backslash v$
$lastsim \leftarrow 0$                                              (1)

while $R \neq \emptyset$ do
  // Search for the next nearest neighbor in $R$.
  $(s, sim) \leftarrow \text{getNearestNeighbor}(L[last], R)$      (2)

  if $sim > lastsim[last]$ then
    // No RNNs → Add $s$ to the nearest-neighbor chain
    $last \leftarrow last + 1$
    $L[last] \leftarrow s$;   $R \leftarrow R \backslash \{s\}$
    $lastsim[last] \leftarrow sim$                                  (3)
  else
    // Found RNNs → agglomerate the last two chain links
    if $lastsim[last] > t$ then
      $s \leftarrow \text{agglomerate}(L[last], L[last-1])$
      $R \leftarrow R \cup \{s\}$
      $last \leftarrow last - 2$                                    (4)
    else
      // Discard the current chain.
      $last \leftarrow -1$

  if $last < 0$ then
    // Initialize a new chain with another random point $v \in R$.
    $last \leftarrow last + 1$
    $L[last] \leftarrow v \in R$;   $R \leftarrow R \backslash \{v\}$   (5)
```

It is easy to see that each NN-chain ends in an RNN pair. The strategy of the algorithm is thus to start with an arbitrary point (1) and build up an NN-chain (2,3). As soon as an RNN pair is found, the corresponding clusters are merged (4). The reducibility property guarantees that when this is done, the nearest-neighbor assignments stay valid for the remaining chain members, which can thus be reused for the next iteration. Whenever the current chain runs empty, a new chain is started with another random point (5). The resulting procedure is summarized in Algorithm 4.2.

An amortized analysis of this algorithm shows that a full clustering requires at most $3(N-1)$ iterations of the main loop (Benzécri, 1982). The run-time is thus bounded by the time required to search the nearest neighbors, which is in the simplest case $O(Nd)$. For low-dimensional data, this can be further reduced by employing efficient NN-search techniques (see Section 4.3.3).

When a new cluster is created by merging an RNN pair, its new distance to other clusters could be computed using the update equations from Table 4.1. However, for doing so the previous distances would have to be kept in memory, requiring again $O(N^2)$ space. Applying an idea by Day and Edelsbrunner (1984) and Voorhees (1986), the algorithm's space requirements can be reduced to $O(N)$ if the cluster similarity can be expressed in terms of centroids. In the following, we show that this is the case for Group Average criteria based on correlation or the Euclidean distance.

Theorem 1. *The Group Average clustering criterion based on correlation can be reformulated as*

$$sim(X,Y) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} < x^{(i)}, y^{(j)} > = < \mu_x, \mu_y > . \tag{4.7}$$

Proof. The theorem follows directly from the linearity property of the inner product. □

Theorem 2. *The Group Average clustering criterion based on Euclidean distances can be reformulated as*

$$sim(X,Y) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} (x^{(i)} - y^{(j)})^2 = \sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2 \tag{4.8}$$

Proof. The proof follows by algebraic manipulation:

$$\begin{aligned}
sim(X,Y) &= \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x^{(i)} - y^{(j)}\right)^2 \\
&= \frac{1}{NM}\left[\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x^{(i)}\right)^2 - 2\sum_{i=1}^{N}\sum_{j=1}^{M} < x^{(i)}, y^{(j)} > + \sum_{i=1}^{N}\sum_{j=1}^{M}\left(y^{(j)}\right)^2\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(x^{(i)}\right)^2 - \frac{1}{N}\sum_{i=1}^{N} < x^{(i)}, \frac{1}{M}\sum_{j=1}^{M} y^{(j)} > + \frac{1}{M}\sum_{j=1}^{M}\left(y^{(j)}\right)^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(x^{(i)}\right)^2 - 2 < \mu_x, \mu_y > + \frac{1}{M}\sum_{j=1}^{M}\left(y^{(j)}\right)^2 \\
&= \left[\left(\frac{1}{N}\sum_{i=1}^{N}\left(x^{(i)}\right)^2 - \mu_x^2\right) + \left(\frac{1}{M}\sum_{j=1}^{M}\left(y^{(j)}\right)^2 - \mu_y^2\right)\right. \\
&\qquad \left. - 2 < \mu_x, \mu_y > + \mu_x^2 + \mu_y^2\right] \\
&= \sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2
\end{aligned}$$

□

Using this formulation, the new distances can be obtained in constant time, requiring just the storage of the mean and variance for each cluster. Both the mean and variance of the updated cluster can be computed incrementally:

$$\mu_{new} = \frac{N\mu_x + M\mu_y}{N+M} \tag{4.9}$$

$$\sigma^2_{new} = \frac{1}{N+M}\left(N\sigma^2_x + M\sigma^2_y + \frac{NM}{N+M}\left(\mu_x - \mu_y\right)^2\right) \tag{4.10}$$

In conclusion, we have presented an average-link clustering algorithm with $O(N^2 d)$ time and $O(N)$ space complexity. Among some other criteria, this algorithm is applicable to the group average criterion with a similarity measure of either correlation or the Euclidean distance. As the method relies heavily on the search for nearest neighbors, its expected-time complexity can in some cases further be improved by using efficient NN-search techniques. This will be discussed in the following section.

4.3.3 Efficient Nearest-Neighbor Search

An exhaustive search for the nearest neighbor among N points in d dimensions requires $O(Nd)$ operations. Numerous methods have been proposed for speeding up this search, for example *k-d trees* (Bentley, 1975; Friedman et al., 1977; Robinson, 1981) or the method by Nene and Nayar (1997). Their underlying idea is to approximate the hypersphere containing the nearest neighbors of a point by a hypercube, thus restricting the number of potential matches. For small dimensions (up to about 25–30D), this can provide a speedup compared to an exhaustive search, since the points in the hypercube can be more efficiently retrieved (Nene and Nayar, 1997).

For higher-dimensional spaces, however, this principle is no longer effective. With increasing dimensionality, the differences between the volumes of hypercube and hypersphere grow too large[6], so that no speedup can be obtained.

For this reason, it is advantageous to project the data into a lower-dimensional space first. The optimal choice for this is a truncated eigenspace, since it minimizes the projection error thus made. For very large and high-dimensional data sets, however, the standard PCA algorithm may be problematic, because it is itself rather costly. Computing the sample covariance matrix alone requires $O(\min(Nd^2, N^2 d))$ operations. Solving the eigenvalue problem then may take between $O(\min(d^2, N^2))$ and $O(\min(d^3, N^3))$ operations, depending on the matrix structure. Fortunately, when only the first m leading eigenvectors are needed, we can use the EM-PCA algorithm by Roweis (1997), which can be efficiently computed in $O(mNd)$ without the need to estimate the sample covariance. We can thus expect further performance improvements from this step.

---

[6] A mathematical analysis shows that the volume of a hypercube grows with $(2r)^d$, whereas the volume of a hypersphere reaches a maximum around $d = 5$ and goes rapidly to zero for larger dimensionalities (Weisstein).

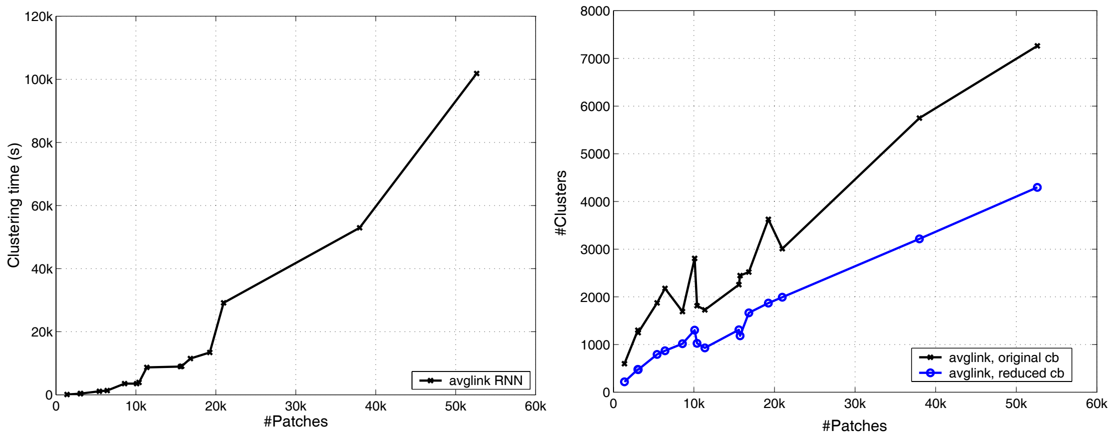


Figure 4.5: *Performance of the RNN algorithm. (left) Run-times for data sets of varying size; (right) Resulting codebook size for those data sets.*

4.3.4 Experimental Results

In order to demonstrate the performance of the RNN algorithm, we present clustering results for various data sets ranging from 1,374 patches from 40 images of a single category up to 52,617 patches from 702 images of 5 visual categories (side and rear views of cars; and side views of cows, motorbikes, and pedestrians). In all cases, the data points have 625 dimensions.

Figure 4.5(left) shows the algorithm's run-time (as yet without any efficient nearest-neighbor search) as a function of the number of patches. All experiments have been performed on a 3GHz Intel Xeon processor. As can be seen from the plot, the run-time scales indeed quadratically with the number of input points and reaches a value of 101,862 seconds (28,3 hours) for the largest data set. For typical single-category codebooks with 8–15,000 patches, the run-time is between 3,500 and 9,000 seconds (1.0–2.5 hours)[7].

Figure 4.5(right) shows the resulting number of clusters, both for the original codebooks and for the reduced versions without single-patch clusters. It can be seen that the number of codebook entries grows much slower than the data set size. Even for the largest data set with 52,617 patches, the original and reduced codebooks only contain 7,362 and 4,295 entries, respectively, which is still a manageable quantity. Of course, the number of codebook entries depends heavily on the data set and its variation, especially for multi-category cases. We therefore contrast the results with the number of clusters when each category is clustered independently. For the 5-category data set, the concatenated single-category codebooks would together contain 11,386 clusters, 5,796 of which consist of more than one patch. From this, it

---

[7] These results were achieved without taking advantage of compiler optimizations. In more recent experiments, an updated implementation of the algorithm successfully clustered data sets of 113,436 patches in 19.6 hours and 157,133 patches in 37.2 hours on an AMD Dual Opteron 1.8GHz processor.

can be seen that there is a large degree of overlap between the individual codebooks. We can thus expect that the number of clusters will not grow indefinitely when more categories are added, but will reach a saturation point.

4.4 Discussion

In conclusion, we have introduced a codebook representation for local-feature based object recognition and categorization approaches. It allows to represent an object's complex appearance distribution by a set of automatically learned local prototypes.

We have paid specific attention to the question how to model the uncertainty that is invariably associated with the codebook matching process and propagate it to later stages of the recognition system. Our solution is based on allowing local features to match to multiple prototypes and interpolating their responses. The knowledge which confusions between prototypes are likely to occur is modelled by storing activation records for each successful assignment of local features to prototypes. This approach permits to use the uncertainty observed on the training examples in order to achieve more robust recognition.

At the same time, the procedure lets us formulate an optimality criterion for the codebook generation process, based on which different clustering methods can be compared. We have performed such a comparison for two different methods: k-means and average-link clustering. Our results are twofold. The experiments show that the proposed uncertainty handling scheme succeeds to compensate for lower-quality codebooks, resulting in nearly the same recognition performance. As a consequence, the exact nature of the codebook is not as important for the achievable recognition performance of a later system. However, the representational quality of the codebook does influence the amount of effort that is needed to compensate for the uncertainty. Since this effort has to be spent during the recognition process, it is advantageous to choose the best available codebook.

In that respect, our results show that k-means is inferior to average-link clustering. For the same level of recognition performance, the k-means codebook requires more than twice as many activation records to be stored. Thus, k-means effectively trades off a faster training phase for a larger effort during recognition.

Further insights can be gained by comparing our codebook representation to the general class of *Vector Quantization* (Gray, 1984) methods, of which k-means is a representative. Vector quantization has been introduced for encoding and compressing data. Its focus is on finding an efficient representation with small reconstruction error for the data distribution in its feature space by matching it to a codebook. However, vector quantization pursues no classification to an object structure – all encountered data points are encoded. If a point does not fit well to the codebook, it is nonetheless represented by the closest prototype (and a correction term is added for the reconstruction error thus made). In our application, we are not interested in representing each data point. Instead, the purpose of our codebook is to only represent object structure and already reject a majority of background patches. For this, it is important to limit the spatial extend of prototypes during the clustering process. This is something that can be better achieved with the Group Average criterion than with k-means, which is one reason for its better performance.

In order to handle the large amounts of data that need to be processed for codebook generation, we have presented an efficient agglomerative clustering algorithm based on reciprocal nearest neighbors (de Rham, 1980; Benzécri, 1982) and we have shown its applicability to the group average criterion with a similarity measure based on correlation or Euclidean distances. Although this algorithm is, in its basic form, already 20 years old, it has so far been little known in the Computer Vision community. With its $O(N^2d)$ time and $O(N)$ space complexity, it is suitable for clustering large data sets, as demonstrated in Section 4.3.4.

As a side note, we want to point out that for the cases considered in our experiments, where the number k of clusters is almost of the same order as N, the k-means algorithm has the same asymptotic time complexity as average-link clustering. Since in our experiments, between 10 and 25 iterations were necessary until the k-means process converged, this number can be combined with the value of k to form a time complexity of $O(N^2d)$. In our experiments, k-means was even slower than average-link clustering with the RNN algorithm.

Altogether, we have thus arrived at a viable framework for generating codebook representations for object categorization. In the following chapter, we will extend the uncertainty handling idea further in order to robustly estimate the spatial occurrence distribution of codebook entries on the object category. The whole procedure will be integrated in a probabilistic framework that allows to recognize categorical objects and, at the same time, generate a probabilistic figure-ground segmentation as a result of the recognition process.

5

Interleaved Object Categorization and Segmentation

The previous chapter has motivated the use of a codebook representation for learning which local structures may appear on objects of the target category. In this chapter, we now use such a representation as basis for the next stage of our system. We learn an *Implicit Shape Model* that specifies where on the object the codebook entries may occur. As the name already suggests, we do not try to define an explicit model for all possible shapes a class object may take, but instead define “allowed” shapes implicitly in terms of which local appearances are consistent with each other. The advantages of this approach are its greater flexibility and the smaller number of training examples it needs to see in order to learn possible object shapes. For example, when learning to categorize articulated objects such as cows, as described in Section 6.3.2, our method does not need to see every possible articulation in the training set. It can combine the information of a front leg seen on one training cow with the information of a rear leg from a different cow to recognize a test image with a novel articulation, since both leg positions are consistent with the same object hypothesis.

This idea is similar in spirit to approaches that represent novel objects by a combination of class prototypes (Jones and Poggio, 1996), or of familiar object views (Ullman, 1998). However, the main difference of our approach is that here the combination does not occur between entire exemplar objects, but through the use of local image patches, which again allows a greater flexibility. Also, the Implicit Shape Model is formulated in a probabilistic framework that allows us to obtain a category-specific segmentation as a result of the recognition process. This segmentation can then in turn be used to improve the recognition results. In particular, we obtain a per-pixel confidence measure specifying how much both the recognition and the segmentation result can be trusted.

The following section defines the shape representation and shows how it is used for recognition. Section 5.2 then derives a probabilistic formulation of the segmentation problem and demonstrates how such a segmentation can be obtained as a result of the recognition process.

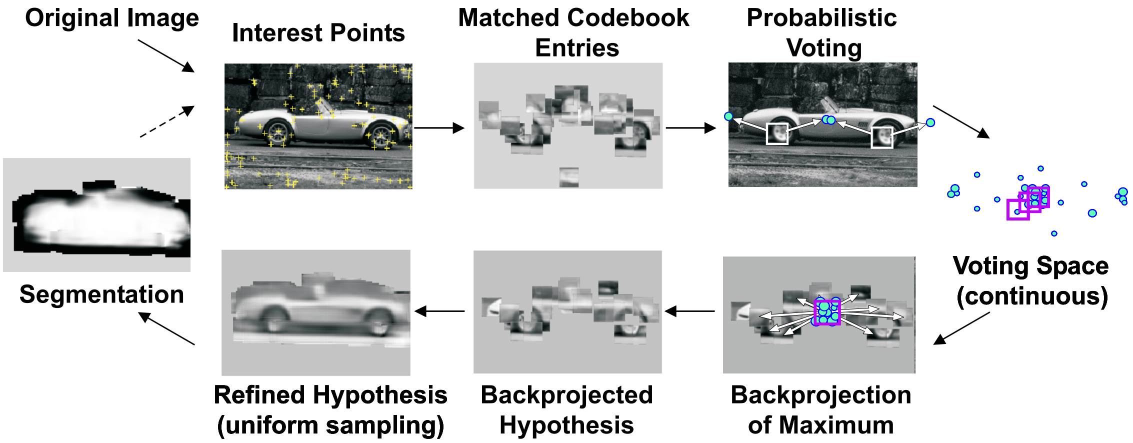


Figure 5.1: *The recognition procedure. Image patches are extracted around interest points and compared to the codebook. Matching patches then cast probabilistic votes, which lead to object hypotheses that can later be refined. Based on the refined hypotheses, we compute a category-specific segmentation.*

5.1 Shape Representation

In our definition, an Implicit Shape Model $ISM(C) = (I_C, P_{I,C})$ for a given object category C consists of a class-specific alphabet I_C (the *codebook*) of local appearances that are prototypical for the object category, and of a spatial probability distribution $P_{I,C}$ which specifies where each codebook entry may be found on the object.

We make two explicit design choices for the probability distribution $P_{I,C}$. The first is that the distribution is defined independently for each codebook entry. This makes the approach flexible, since it allows to combine object parts during recognition that were initially observed on different training examples. In addition, it enables us to learn recognition models from relatively small training sets, as our experiments in Section 5.2.3 and in Chapter 6 will demonstrate. The second constraint is that the spatial probability distribution for each codebook entry is estimated in a non-parametric manner. This enables the method to model the true distribution in as much detail as the training data permits instead of making an oversimplifying Gaussian assumption.

The rest of this section explains how this learning and modelling step is implemented and how the resulting implicit model is used for recognition.

5.1.1 Learning the Shape Model

Let I_C be the learned appearance codebook, as described in the previous chapter. The first step is to learn the spatial probability distribution $P_{I,C}$. For this, we perform a second iteration over all training images and match the codebook entries to the images (again using the NGC measure). Here, we activate not only the best-matching codebook entry, but all entries whose similarity is above t, the threshold

Algorithm 5.1 The training procedure.

```
// Create an appearance codebook $I$.
$E \leftarrow \emptyset$
for all training images do
    Apply the interest point detector.
    for all interest points $(\ell_x, \ell_y)$ and corresponding patches $e_k$ do
        $E \leftarrow E \cup e_k$
Cluster $E$ with $t = 0.7$ and keep cluster centers $I$.

// Compute occurrences $Occ$.
for all codebook entries $I_i$ do
    $Occ[i] \leftarrow \emptyset$
for all training images do
    Let $(c_x, c_y)$ be the object center.
    Apply the interest point detector.
    for all interest points $(\ell_x, \ell_y)$ and corresponding patches $e_k$ do
        for all codebook entries $I_i$ do
            if $sim(I_i, e_k) \geq t$ then
                // Record an occurrence of codebook entry $I_i$
                $Occ[i] \leftarrow Occ[i] \cup (c_x - \ell_x, c_y - \ell_y)$
```

already used during clustering. For every codebook entry, we store all positions it was activated in, relative to the object center.

By this step, we model the uncertainty in the codebook generation process. If a codebook is "perfect" in the sense that each patch can be uniquely assigned to exactly one cluster, then the result is equivalent to a nearest-neighbor matching strategy. However, as argued in Chapter 4, it is unrealistic to expect such clean data in practical applications. We therefore keep each possible assignment, but weight it with the probability that this assignment is correct. It is easy to see that for similarity scores smaller than t, the probability that this patch could have been assigned to the cluster during the codebook generation process is zero; therefore we do not need to consider those matches.

The stored occurrence locations, on the other hand, reflect the spatial distribution of a codebook entry over the object area. This information can be sampled by a kernel density estimator to obtain a non-parametric probability density estimate for $P_{I,C}$. The resulting flexible representation is able to model the true distribution in as much detail as is available from the training data. This is especially important as experiments over several object classes have shown that the spatial distribution of codebook entries can in many cases not accurately be described by a single Gaussian[1]. Algorithm 5.1 summarizes the training procedure.

---

[1] As an example, consider the spatial distributions of car wheels or of texture patterns on a cow's body.

5.1.2 Recognition Approach

Figure 5.1 illustrates the ensuing recognition procedure. Given a new test image, we again apply an interest point detector and extract patches around the selected locations. The extracted patches are then matched to the codebook to activate codebook entries using the same mechanism as described above. From the set of all those matches, we collect consistent configurations by performing a Generalized Hough Transform (Hough, 1962; Ballard, 1981; Lowe, 1999). Each activated entry casts votes for possible positions of the object center according to the learned spatial distribution $P_{I,C}$. Consistent hypotheses are then searched as local maxima in the voting space.

When pursuing such an approach, it is important to avoid all quantization artifacts. In contrast to usual practice (e.g. Lowe, 1999), we therefore do not discretize the votes in a binned accumulator array, but keep the original, continuous votes. Maxima in this continuous space can be accurately and efficiently found using Mean-Shift Mode Estimation (Cheng, 1995; Comaniciu and Meer, 1999)[2].

Once a hypothesis has been selected, all patches that contributed to it are collected (Fig. 5.1(bottom)), therefore visualizing what the system reacts to. Moreover, we can refine the hypothesis by sampling all the image patches in its surroundings, not just those locations returned by the interest point detector. As a result, we get a representation of the object including a certain border area. This refined hypothesis will later serve as the basis for computing a category-specific segmentation, as described in Section 5.2. First, however, we derive a probabilistic formulation for the voting process.

5.1.3 Probabilistic Framework

In the following, we cast the recognition procedure into a probabilistic framework (Leibe and Schiele, 2003b; Leibe et al., 2004). Let $\mathbf{e}$ be our evidence, an extracted image patch observed at location ℓ. By matching it to the codebook, we obtain a set of valid interpretations I_i. Each interpretation is weighted with the probability $p(I_i|\mathbf{e},\ell)$. If a codebook cluster matches, it can cast its votes for different object positions. That is, for every I_i, we can obtain votes for several object identities o_n and positions x, which we weight with $p(o_n,x|I_i,\ell)$. Formally, this can be expressed by the following marginalization:

$$p(o_n,x|\mathbf{e},\ell) \;=\; \sum_i p(o_n,x|\mathbf{e},I_i,\ell)p(I_i|\mathbf{e},\ell). \tag{5.1}$$

Since we have replaced the unknown image patch by a known interpretation, the first term can be treated as independent from $\mathbf{e}$. In addition, we match patches to

---

[2] A similar use of a continuous Hough Transform can be found in (Hameiri and Shimshoni, 2002), although in a different context.

the codebook independent of their location. The equation thus reduces to

$$\begin{aligned} p(o_n, x|\mathbf{e}, \ell) &= \sum_i p(o_n, x|I_i, \ell)p(I_i|\mathbf{e}). \qquad (5.2) \\ &= \sum_i p(x|o_n, I_i, \ell)p(o_n|I_i, \ell)p(I_i|\mathbf{e}). \qquad (5.3) \end{aligned}$$

The first term is the probabilistic Hough vote for an object position given its identity and the patch interpretation. The second term specifies a confidence that the codebook cluster is really matched on the object as opposed to the background. This can be used to include negative examples in the training process. Finally, the third term reflects the quality of the match between image patch and codebook cluster.

The score of a hypothesis $h = (o_n, x)$ is obtained by marginalizing over all patches that contribute to this hypothesis. By basing the decision on single-patch votes, we arrive at the following equation

$$p(o_n, x) = \sum_k p(o_n, x|\mathbf{e}_k, \ell_k)p(\mathbf{e}_k, \ell_k) \qquad (5.4)$$

where $p(\mathbf{e}_k, \ell_k)$ is an indicator variable specifying which patches $(\mathbf{e}_k, \ell_k)$ have been sampled by the interest point detector. In practice, we can neglect this term by computing the sum only over sampled patches. However, in order to be robust to intra-class variation, we have to tolerate small shape deformations. We achieve this by integrating votes over a fixed-size search window $W(x)$ during the Mean-Shift search and thus obtain

$$score(o_n, x) = \sum_k \sum_{x_j \in W(x)} p(o_n, x_j|\mathbf{e}_k, \ell_k). \qquad (5.5)$$

In order to avoid any systematic bias, we require that each sampled patch have the same a-priori weight. From this, it immediately follows that the $p(I_i|\mathbf{e})$ and $p(x|o_n, I_i, \ell)$ should both sum to one. In our experiments, we assume a uniform distribution for both (meaning that we set $p(I_i|\mathbf{e}) = \frac{1}{|I|}$, with $|I|$ the number of matching codebook entries), but it would also be possible, for example, to let the $p(I_i|\mathbf{e})$ distribution reflect the relative matching scores. The complete recognition procedure is summarized in Algorithm 5.2.

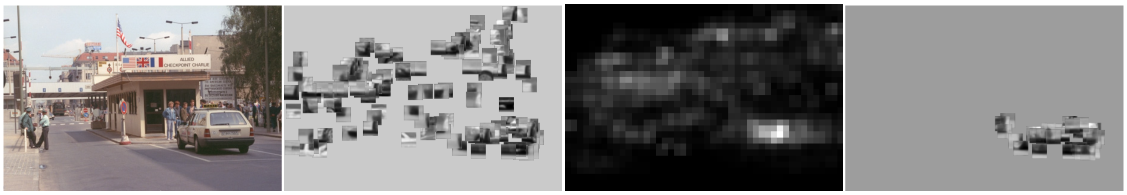
By this derivation, we have embedded the Hough voting strategy in a probabilistic framework. In this context, the mean-shift search over the voting space can be interpreted as a Parzen window probability density estimation for the correct object location. The power of this approach lies in its non-parametric nature. Instead of making Gaussian assumptions for the codebook cluster distribution on the object, our approach is able to model the true distribution in as much detail as is possible from the observed training examples.

Algorithm 5.2 The ISM recognition algorithm.

```
// Produce probabilistic votes.
$V \leftarrow \emptyset$
Apply the interest point detector to the test image.
for all interest points $(\ell_x, \ell_y)$ and corresponding patches $e_k$ do
    $M \leftarrow \emptyset$
    // Record all matches to the codebook
    for all codebook entries $I_i$ do
        if $sim(I_i, e_k) \geq t$ then
            $M \leftarrow M \cup (i, \ell_x, \ell_y)$
    $p(I_i|e_k) \leftarrow \frac{1}{|M|}$
    for all matches $(i, \ell_x, \ell_y) \in M$ do
        for all occurrences $occ \in Occ[i]$ of codebook entry $I_i$ do
            $x \leftarrow (\ell_x - occ_x, \ell_y - occ_y)$
            $p(o_n, x|I_i, \ell) \leftarrow \frac{1}{|Occ[i]|}$
            Cast a vote $(x, w, occ, \ell)$ for position $x$ with weight $w = p(o_n, x|I_i, \ell)p(I_i|e_k)$
            $V \leftarrow V \cup (x, w, occ, \ell)$

// Sample the voting space in a regular grid to obtain promising starting locations.
for all grid locations $x$ do
    $score(x) \leftarrow applyMSMEKernel(W, x)$
// Refine the local maxima using MSME with the kernel window $W$.
for all grid locations $x$ do
    if $x$ is a local maximum then
        // Apply the MSME search
        repeat
            $score \leftarrow 0$, $x_{new} \leftarrow 0$, $n \leftarrow 0$
            for all votes $(x_k, w_k, occ_k, \ell_k)$ do
                if $x_k$ is inside $W(x)$ then
                    $score \leftarrow score + w_k$
                    $x \leftarrow x + x_k$
                    $n \leftarrow n + 1$
            $x \leftarrow \frac{1}{n} x_{new}$
        until convergence
        if $score \geq \theta$ then
            Create hypothesis $h$ for position $x$.
```

5.1.4 Experimental Results

Figure 5.2 illustrates the different steps of the recognition procedure on a real-world example. As in all following examples, the system was trained on 16 views of each of the 10 toy cars shown in Figure 5.3. When presented with the test image, the system applies the interest point detector and extracts a total of 431



(a) orig. image (b) matched patches (c) voting space (d) hypothesis

Figure 5.2: *Intermediate results during the recognition process. (a) original image; (b) extracted patches that could be matched to the codebook; (c) probabilistic votes; (d) support of the strongest hypothesis. (Note that the voting process takes place in a continuous space. The votes are just discretized for visualization).*

patches. However, only 132 of them contain relevant structure and pass the codebook matching stage (Fig. 5.2(b)). Those patches then cast probabilistic votes, which are collected in the voting space. As a visualization of this space in Fig. 5.2(c) shows, only few patches form a consistent configuration. The system searches for local maxima in the voting space and returns the correct detection as strongest hypothesis. By backprojecting the contributing votes, we retrieve the hypothesis's support in the image (Fig. 5.2(d)), which shows that the system's reaction has indeed been produced by local structures on the depicted car.

Figure 5.4 shows some other successful recognition results on difficult real-world test images. Although only trained on toy objects, the system is able to generalize to real cars and find them as the strongest hypothesis in the image. The displayed support shows that in all cases the decision has been made on a correct basis.

In order to obtain a more quantitative assessment of the method's performance, we applied it to a test set of 137 real-world images[3] containing one car each in varying poses. The images were partly taken from the Corel database, partly from the Internet, and partly acquired with a digital camera. All images were manually scaled such that the cars had the same size as our training examples. The envisioned task is to detect and localize the cars in the test images. We define the quality of recognition in terms of four parameters: distance from the hypothesis center to the real object center in x and y direction; coverage of the true bounding box by the hypothesis bounding box; and mutual overlap of the boxes. Since the resolution of our images is roughly twice that of Agarwal & Roth's (Agarwal and Roth, 2002), we double the tolerances used in their evaluation and accept a hypothesis if $\delta_x \leq 56$ pixels, $\delta_y \leq 28$ pixels, and coverage and overlap are both above 0.5.

Figure 5.5 shows the results of this experiment in the form of a rank plot, i.e. in terms of the maximal recognition rate that can be achieved when all hypotheses up to a certain rank are considered. Based on Harris interest points, the system is able to correctly recognize and localize 53.3% of the cases with its first hypothesis. When the first 5 hypotheses are considered, the correct detection is among them in

---

[3] The complete image set is available at `http://www.mis.informatik.tu-darmstadt.de/projects/interleaved/`.

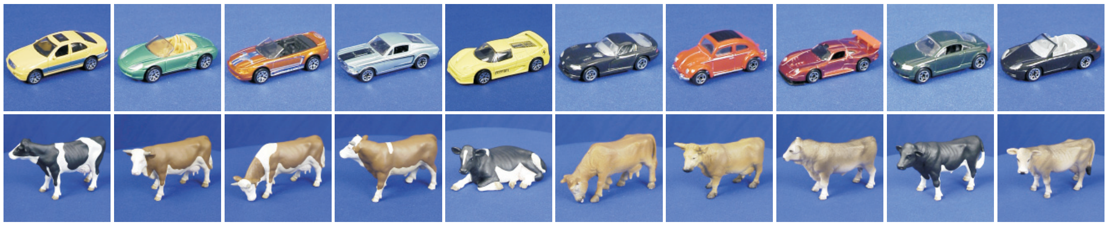


Figure 5.3: *Training objects used for cars and cows (from the CogVis-ETH80 database (Leibe and Schiele, 2003a)). For each object, 16 views were taken from different orientations.*

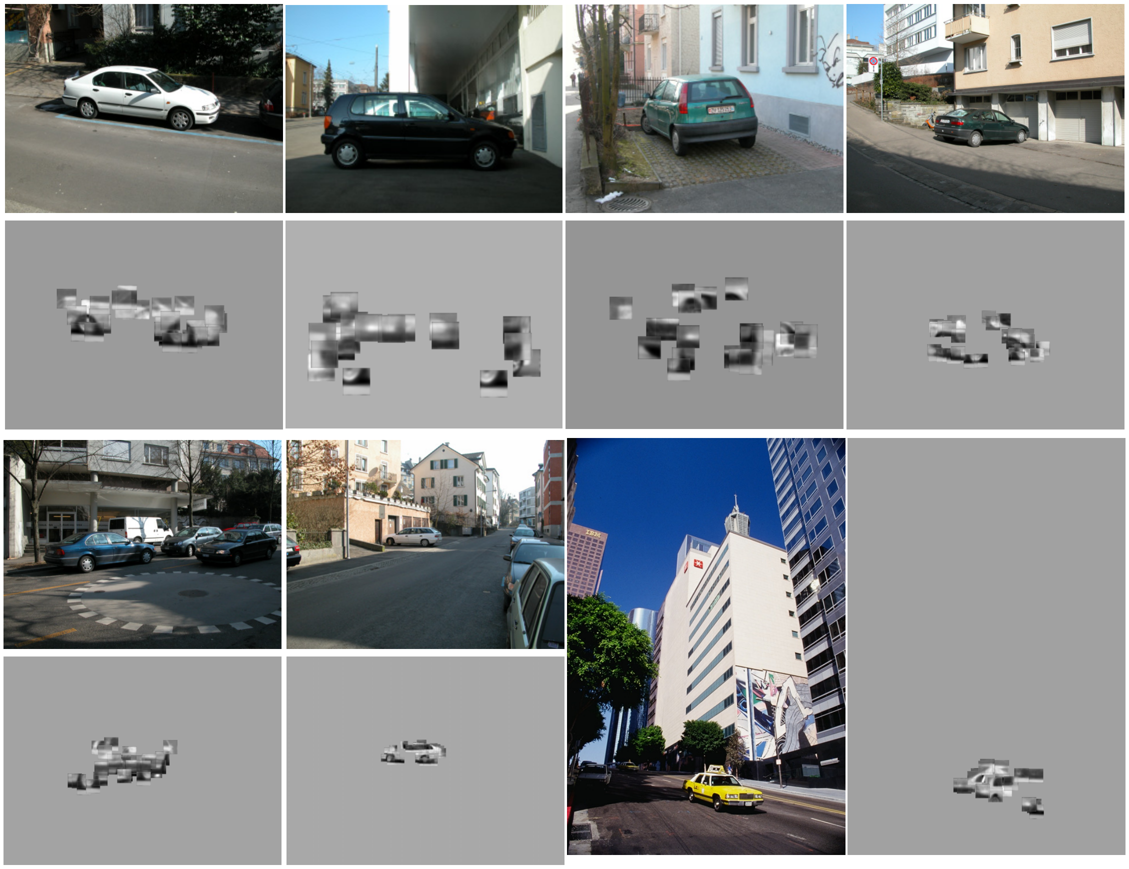


Figure 5.4: *Some more examples of successful recognition results on real-world test images. The images show the support of the system's* 1^{st} *hypothesis.*

75.9% of the cases. Altogether, the method finds a correct hypothesis for 86.1% of the test images. Considering the difficulty of the task, this is a very encouraging result. The results show that our learned representation is indeed representative for cars. Even when only trained on 10 toy cars, the system is able to generalize and detect real-world cars in difficult scenes. In addition, the correct detections are often contained in the system's first hypotheses.

Nevertheless, false positives are still a problem. In order to obtain a better

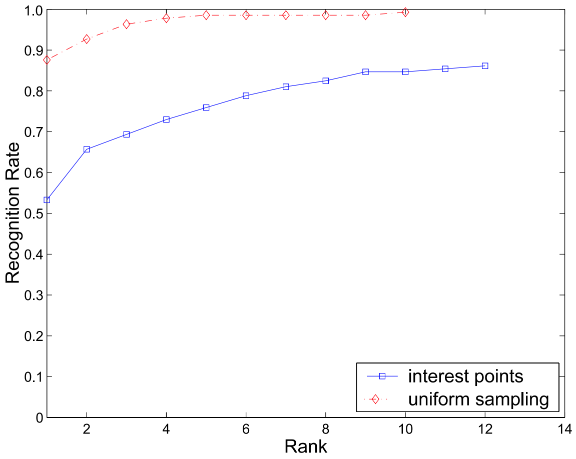
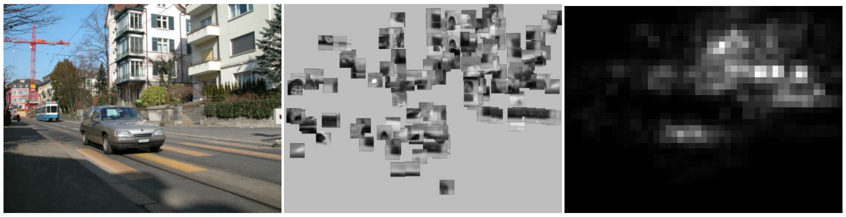
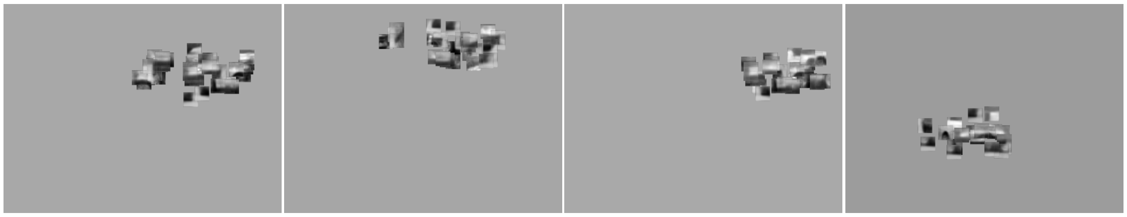


Figure 5.5: *Recognition results on a test set of 137 car images if all hypotheses up to a certain rank are considered.*



(a) orig. image (b) matched patches (c) voting space

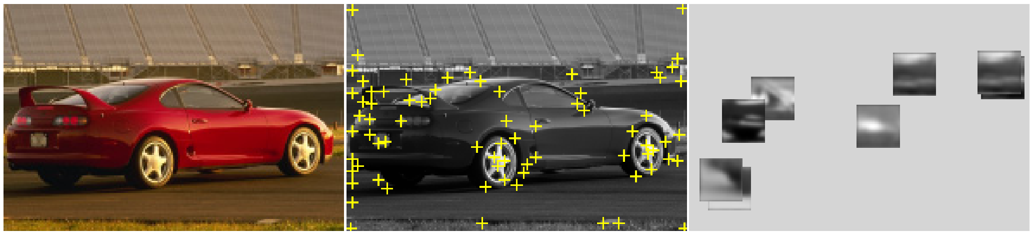


(d) 1st hyp. (e) 2^{nd} hyp. (f) 3^{rd} hyp. (g) 4^{th} hyp.

Figure 5.6: *A case where false positives were introduced due to background clutter.*

understanding of their causes, we look at two typical problem cases. Figure 5.6 shows an image containing high-contrast regular structures in the background. Although the depicted car is correctly found as a local maximum in voting space, the cluttered background causes many spurious image patches to be matched. Even though these do not correspond to valid car configurations, their sheer number leads to random peaks in the voting space, which give rise to several false positives with higher recognition scores.

Figure 5.7, on the other hand, shows an example where recognition fails completely because of bad grayvalue contrast. Although several interest points are found on the depicted car, the extracted patches could not be matched to any learned code-



(a) orig. image (b) interest points (c) matched patches

Figure 5.7: *An example where patch extraction failed completely because of bad (grayvalue) contrast.*

book entry. As a result, the system does not obtain enough support to produce a valid hypothesis.

It is important to note that this failure is not entirely due to the interest point detector. As Figure 5.7(b) shows, enough interest points are found on the object. However, because of the very limited training set, the extracted patches do not correspond to any known structure. We therefore repeat the experiment, but this time taking all available image patches by uniformly sampling the test image in a dense grid. This increases the chances that known structure is found on the test objects, which can be used for recognition. Figure 5.5 shows the results when this is done. As can be seen from the figure, performance improves to 87.6% with the first hypothesis and to 98.5% with the first 5 hypotheses. In total, only 2 out of the 137 test cases could not be solved at all. This confirms our previous results that the proposed implicit representation is suitable for object categorization.

While clearly providing better results, though, the uniform sampling strategy is not unproblematic for other reasons. Taking the example from Figure 5.2, a uniform sampling strategy with a grid spacing of 2 pixels now extracts 31,603 image patches, instead of the mere 431 selected by the Harris detector. The resulting increase in computational complexity of two orders of magnitude may render this option prohibitive for larger scenes. For practical applications, it is thus a better strategy to increase the size of the training set and include more real-world variation.

5.1.5 Discussion

In this section, we have presented a local approach for detecting categorical objects in real-world images. Our approach is based on an Implicit Shape Model, which allows to find consistent configurations of local features. Extracted patches from the test image are matched to a category-specific codebook. Each matched codebook entry then casts probabilistic votes for possible object positions according to a learned spatial distribution. Maximal aggregations of votes correspond to consistent hypotheses, which can be backprojected to the original image to obtain the hypotheses' support. The flexible nature of this representation allows to combine information from different training images. As a result, our approach is able to learn and generalize already from a small number of training examples. Experimental results show the system's ability to categorize objects in a variety of different poses.

Three main sources can be identified for the relatively large number of false positives still observed in the experiments. Foremost is certainly the very limited amount of training data. While the experiments have shown that the Implicit Shape Model can generalize already from the 10 toy objects in the CogVis database, there is no doubt that for robust real-world performance, a larger and more realistic training set needs to be provided.

A second cause for false positives is that the object model used in this evaluation has been trained on several different viewpoints at once. As the results in Figure 5.4 show, this allows to recognize cars in different poses with the same model. However, it also reduces the discriminance of the classifier, since our voting scheme allows that image patches which support different object poses contribute to the same hypothesis. As a result, the occurrence distribution of each codebook entry is more spread-out, and there is a higher chance for random background patches to generate strong hypotheses. For this reason, we will concentrate only on single-viewpoint classifiers in the following experiments.

Last but not least, the experiments have shown that the initial patch sampling strategy can affect the recognition result. Especially with small training sets, the sampled patches may not contain sufficient known structure to recognize the object. High-contrast regions on the background, on the other hand, can result in a large number of spuriously matched patches, which may cause random peaks in the voting space.

In this evaluation, we introduced one possible solution for this problem, namely to choose patches from the whole image by uniform sampling. While this strategy yields better recognition results, the benefit comes at a price of significantly increased computational expense. Since the complexity is increased by approximately two orders of magnitude, the computational requirements may be prohibitive for large scenes. The results should therefore merely be seen as an indicator for the maximal achievable performance. For practical applications, we instead advocate a combined strategy, relying on interest points for initial hypothesis generation and a second stage of hypothesis verification, in particular when the image conditions are not favorable. Chapter 6 will explore this idea in more detail.

In the following, we want to focus on a different aspect of the problem. In the experiments reported above, we have used a hypothesis' support to obtain a rough bounding box of the object. As the sampled patches still contain background structure, however, this segmentation is rather inaccurate. On the other hand, we have expressed the a-priori unknown image content in terms of a learned codebook; thus, we know more about the semantic interpretation of the matched patches for the target object. In the following, we will explore how this information can be used to increase our knowledge about the scene. In particular, we will show how the probabilistic framework can be extended to yield a pixel-wise figure-ground segmentation of the object.

5.2 Figure-Ground Segmentation

5.2.1 Theoretical Derivation

In this section, we describe a probabilistic formulation for the segmentation problem (as derived in (Leibe and Schiele, 2003b)). As a starting point, we take a refined object hypothesis $h = (o_n, x)$ obtained by the algorithm from the previous section. Based on this hypothesis, we want to segment the object from the background.

Up to now, we have only dealt with image patches. For the segmentation, we now want to know whether a certain image pixel $\mathbf{p}$ is *figure* or *ground*, given the object hypothesis. More precisely, we are interested in the probability $p(\mathbf{p} = \mathit{figure}|o_n, x)$. The influence of a given patch $\mathbf{e}$ on the object hypothesis can be expressed as

$$p(\mathbf{e}, \ell|o_n, x) = \frac{p(o_n, x|\mathbf{e}, \ell)p(\mathbf{e}, \ell)}{p(o_n, x)} = \frac{\sum_I p(o_n, x|I, \ell)p(I|\mathbf{e})p(\mathbf{e}, \ell)}{p(o_n, x)} \tag{5.6}$$

where the patch votes $p(o_n, x|\mathbf{e}, \ell)$ are obtained from the codebook, as described in the previous section. Given these probabilities, we can obtain information about a specific pixel by marginalizing over all patches that contain this pixel:

$$p(\mathbf{p} = \mathit{figure}|o_n, x) = \sum_{\mathbf{p}\in(\mathbf{e},\ell)} p(\mathbf{p} = \mathit{figure}|o_n, x, \mathbf{e}, \ell)p(\mathbf{e}, \ell|o_n, x) \tag{5.7}$$

with $p(\mathbf{p} = \mathit{figure}|o_n, x, \mathbf{e}, \ell)$ denoting patch-specific segmentation information, which is weighted by the influence $p(\mathbf{e}, \ell|o_n, x)$ the patch has on the object hypothesis. Again, we can resolve patches by resorting to learned patch interpretations I stored in the codebook:

$$\begin{aligned} p(\mathbf{p} = \mathit{figure}|o_n, x) &= \sum_{\mathbf{p}\in(\mathbf{e},\ell)} \sum_I p(\mathbf{p}{=}\mathit{fig.}|o_n, x, \mathbf{e}, I, \ell)p(\mathbf{e}, I, \ell|o_n, x) && (5.8)\\ &= \sum_{\mathbf{p}\in(\mathbf{e},\ell)} \sum_I p(\mathbf{p}{=}\mathit{fig.}|o_n, x, I, \ell)\frac{p(o_n, x|I, \ell)p(I|\mathbf{e})p(\mathbf{e}, \ell)}{p(o_n, x)} && (5.9)\end{aligned}$$

This means that for every pixel, we build a weighted average over all segmentations stemming from patches containing that pixel. The weights correspond to the patches' respective contributions to the object hypothesis. For the *ground* probability, the result is obtained in a similar fashion.

$$p(\mathbf{p} = \mathit{ground}|o_n, x) = \sum_{\mathbf{p}\in(\mathbf{e},\ell)} \sum_I (1 - p(\mathbf{p}{=}\mathit{fig.}|o_n, x, I, \ell))\, p(\mathbf{e}, I, \ell|o_n, x) \tag{5.10}$$

The most important part in this formulation is the per-pixel segmentation information $p(\mathbf{p} = \mathit{figure}|o_n, x, I, \ell)$, which is only dependent on the matched codebook entry, no longer on the image patch. If we store a fixed segmentation mask for every codebook entry (similar to the approach of Borenstein and Ullman (2002)),

Algorithm 5.3 The top-segmentation algorithm.

```
// Given: hypothesis $h$ and supporting votes $V_h$.

for all supporting votes $(x, w, occ, \ell) \in V_h$ do
  Let $img_{mask}$ be the patch segmentation mask corresponding to $occ$.
  $u_0 \leftarrow (\ell_x - \frac{1}{2}\texttt{patchsize})$
  $v_0 \leftarrow (\ell_y - \frac{1}{2}\texttt{patchsize})$
  for all $u \in [0, \texttt{patchsize}]$ do
    for all $v \in [0, \texttt{patchsize}]$ do
      $img_{pfig}(u - u_0, v - v_0) \leftarrow img_{pfig}(u - u_0, v - v_0) + w \cdot img_{mask}(u, v)$
      $img_{pgnd}(u - u_0, v - v_0) \leftarrow img_{pgnd}(u - u_0, v - v_0) + w \cdot (1 - img_{mask}(u, v))$
```

we obtain a reduced probability $p(\mathbf{p} = \mathit{figure}|I, o_n)$. In our approach, we remain more general by keeping a separate segmentation mask for every stored *occurrence position* of each codebook entry. We thus take advantage of the full probability $p(\mathbf{p} = \mathit{figure}|o_n, x, I, \ell)$. The following section describes in more detail how this is implemented in practice.

5.2.2 Implementation

For learning segmentation information, we make use of a reference figure-ground segmentation mask that is available for each of our training images. We can thus obtain a figure-ground mask for any image patch from the training data. We have experimented with two different ways of integrating segmentation information into the system, corresponding to the different interpretations of the probability $p(\mathbf{p} = \mathit{figure}|o_n, x, I, \ell)$ described above.

In the first approach, as inspired by Borenstein and Ullman (2002), we store a segmentation mask with every image patch obtained from the training images. When the patches are clustered to form codebook entries, the mask coherence is integrated into the similarity measure used for clustering. Thus, it is ensured that only patches with similar segmentation masks, in addition to similar appearance, are grouped together.

Whenever a codebook entry is matched to the image during recognition, its stored segmentation mask is applied to the image. The entry may cast votes for different object identities and positions, but whatever it votes for, the implied segmentation mask stays the same. When an object hypothesis is formed as a maximum in voting space, all patch interpretations contributing to that hypothesis are collected, and their associated segmentation masks are combined to obtain the per-pixel probabilities $p(\mathbf{p} = figure|o_n, x)$.

In the second approach, pioneered in this work and described in Algorithm 5.3, we do not keep a fixed segmentation mask for every codebook entry, but we store a separate mask for every location it occurs in on the training images. With the 2,519

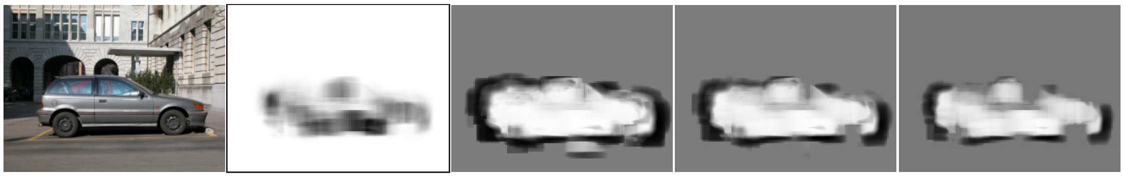

(a) image (b) p(figure) (c) $\theta = 0.1$ (d) $\theta = 0.4$ (e) $\theta = 1.0$

Figure 5.8: *Segmentation results with different confidence[4] levels θ.*

codebook entries used for the car category, we thus obtain 20,359 occurrences, with one segmentation mask stored for each (in practice, this number can be reduced, since only 8,269 of the segmentation masks are different). For the cow category, the codebook contains only 2,244 clusters, but these occur in a total of 50,792 locations on the training images (with 10,378 distinct patch segmentation masks), owing to the larger texture variability on the cow bodies.

Whenever a codebook entry is matched to the image using this approach, a separate segmentation mask is associated with every object position it votes for. Thus, the same vertical structure can indicate a solid area if it is in the middle of a cow's body, and a strong border if it is part of a leg. Which option is finally selected depends on the winning hypothesis and its accumulated support from other patches. In any case, the feedback loop of only taking the votes that support the winning hypothesis ensures that only consistent interpretations are used for the later segmentation.

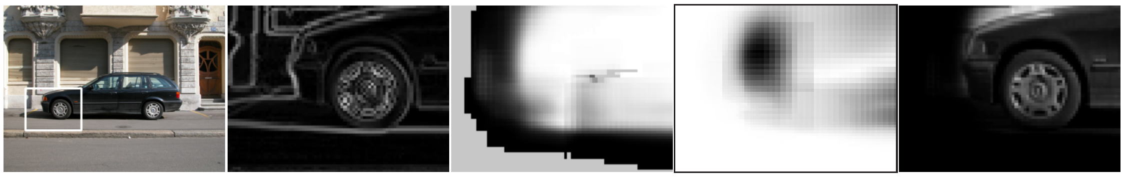
In our experiments, we obtained much better results with the occurrence masks, even when edge information was used to augment matches. In the following, we therefore only report results for occurrence masks. Further, we assume uniform priors for $p(\mathbf{e}, \ell)$ and $p(o_n, x)$, so that these elements can be factored out of the equations. In order to obtain a segmentation of the whole image from the figure and ground probabilities, we build the likelihood ratio for every pixel:

$$L = \frac{p(\mathbf{p} = \textit{figure}|o_n, x)}{p(\mathbf{p} = \textit{ground}|o_n, x)}. \tag{5.11}$$

Figure 5.8 shows example segmentations of a car, together with $p(\mathbf{p} = \textit{figure}|o_n, x)$, the system's confidence in the segmentation result. The darker a pixel, the higher its probability of being *figure*. The lighter it is, the higher its probability of being *ground*. The uniform gray region in the background of the segmentation image does not contribute to the object hypothesis and is therefore considered neutral. By only considering pixels where $\max(p(\textit{figure}), p(\textit{ground})) > \theta$, the computed probability can be used to set a certain "confidence level" for the segmentation and thus limit the amount of missegmentation. Figures 5.8(c)-(e) show segmentation results with different confidence levels[4]. As can be observed, the segmentation with the lowest

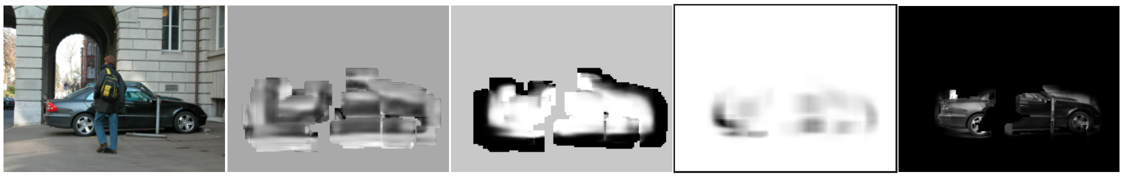
---

[4] The confidences are not in the range $[0, 1]$, because we omitted a normalization factor in the implementation. For better visualization, the images show not L but $\textit{sigmoid}(\log L)$.



(a) orig. image (b) edges (c) segmentation (d) p(figure) (e) segm. image

Figure 5.9: *An example where object knowledge compensates for missing edge information.*



(a) orig. image (b) hypothesis (c) segmentation (d) p(figure) (e) segm. image

Figure 5.10: *Segmentation result of a partially occluded car. The system is able to segment out the pedestrian, because it does not contribute to the car hypothesis.*

confidence level still contains some missegmented areas, while higher confidence levels ensure that only trusted segmentations are made, although at the price of leaving open some uncertain areas. The estimate of how much the obtained segmentation can be trusted is especially important when the results shall later be combined with other cues for recognition or segmentation. It is also the basis for our MDL-based hypothesis verification criterion described in Chapter 6.

5.2.3 Experimental Results

In the following, we present two examples that highlight the advantages a top-down segmentation can offer, compared to bottom-up and gradient-based approaches. The enlargement shown in Figure 5.9 demonstrates one such advantage. At the bottom of the car, there is no visible border between the black car body and the dark shadow underneath. Instead, a strong shadow line extends much further to the left of the car. The proposed algorithm can compensate for that since it “knows” that if a codebook entry matches in this position relative to the object center, it must contain the car’s border. Since at this point only those patch interpretations are considered that are consistent with the object hypothesis, the system can infer the missing contour.

Figure 5.10 shows another interesting case. Even though the car in the image is partially occluded by a pedestrian, the algorithm finds it with its second hypothesis. Refining the hypothesis yields a good segmentation of the car, without the occluded area. The system is able to segment out the pedestrian, because it contributes nothing to the car hypothesis. This is something that would be very hard to achieve

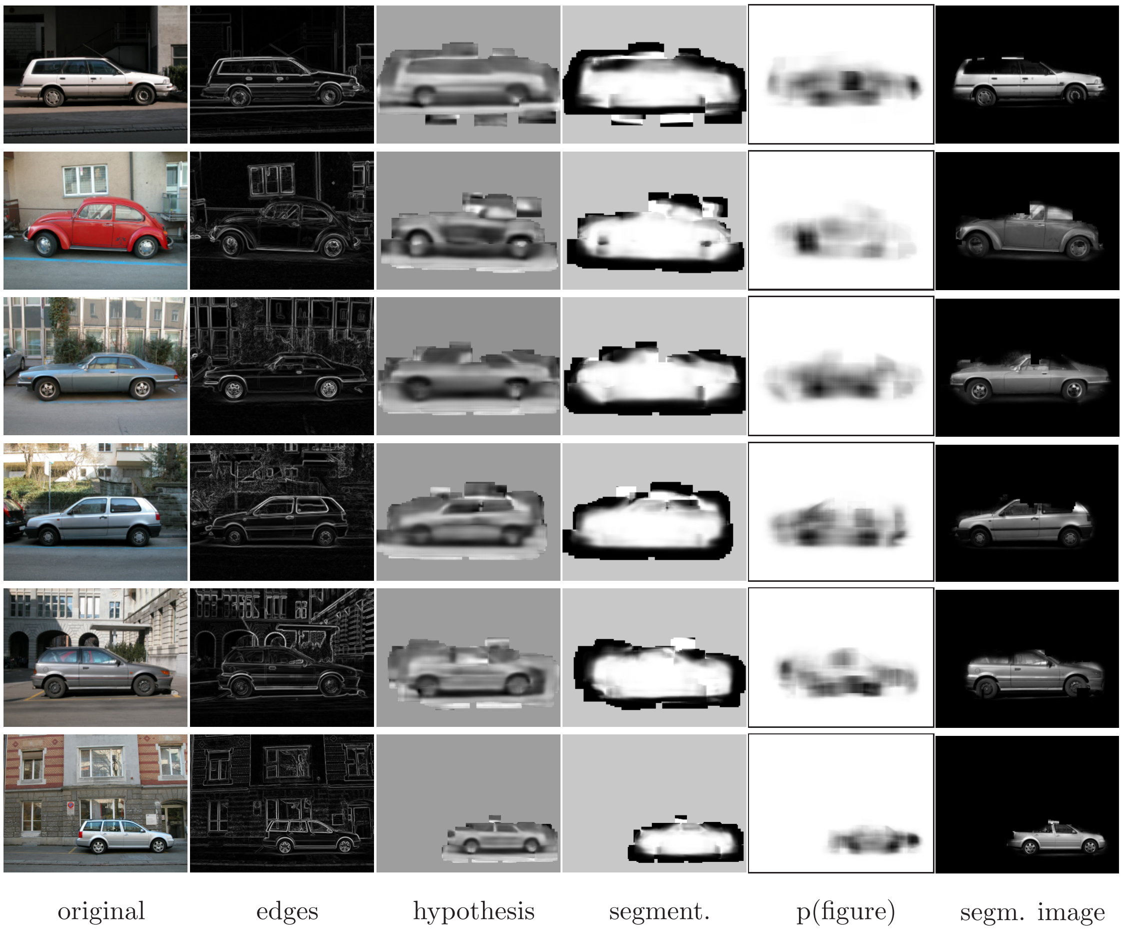


Figure 5.11: *Example segmentation results for car images (when trained on only the 10 toy cars from the CogVis-ETH80 database).*

for a system purely based on pixel-level discontinuities.

More segmentation results for cars and cows can be seen in Figures 5.11 and 5.12. As already in the previous examples, training was done on the toy objects from the CogVis-ETH80 database. In contrast to the recognition experiments in Section 5.1.4, however, the system was only trained on side views of either cars or cows. All depicted cars and the first three cows were correctly found with the recognition system's first hypothesis (the last cow was found with the second hypothesis). Next to each test image, the gradient magnitude is shown to illustrate the difficulty of the segmentation task. Even though the images contain low contrast and significant clutter, the algorithm succeeds in providing a good segmentation of the object. Confidence and segmentation quality are especially high for the bottom parts of the cars, including the cars' shadows (which were labelled *figure* in the training examples). Most difficulties arise with the car roofs and cow heads. These regions contain a lot of variation (e.g. caused by (semi-) transparent windows or different head orientations), which is not sufficiently represented in the training data. What is remarkable, though, is that the cows' legs are captured well, even though no single

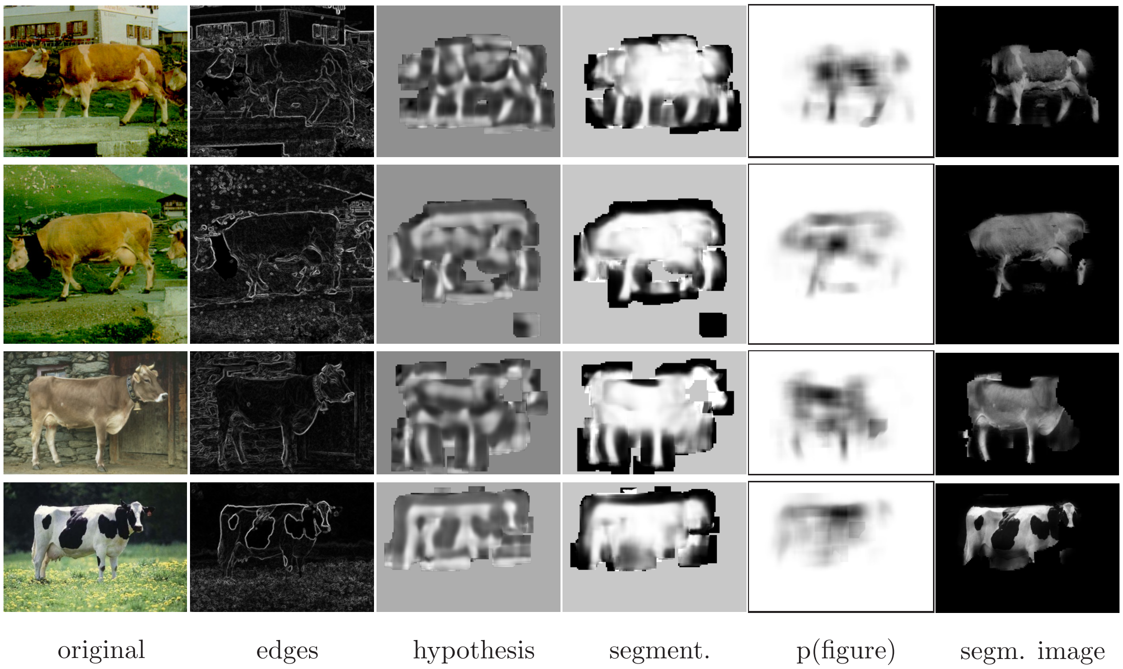


Figure 5.12: *Example results for cow images (when trained on the 10 toy cows from the CogVis-ETH80 database).*

training object contained exactly the same leg configuration. The local approach can compensate for that by combining elements from different training objects.

Another interesting effect can be observed in the cow images 1 and 4. Even though there are strong edge structures on the cows' bodies, no borders are introduced there, since the system has learned that those edges belong to the body. On the other hand, relatively weak edges around the legs lead to strong segmentation results. The system has learned that if a certain structure occurs in this region, it must be a leg. No heuristics are needed for this behavior – it is entirely learned from training data.

5.2.4 Discussion

In this section, we have developed an algorithm that computes a figure-ground segmentation as a result and extension of object recognition. The method uses a probabilistic formulation to integrate learned knowledge about the recognized category with the supporting information in the image. As a consequence, it returns a figure-ground segmentation for the object, together with a per-pixel confidence estimate specifying how much this segmentation can be trusted. We have applied the method to the task of detecting and segmenting unfamiliar objects in difficult real-world scenes. Experiments show that the approach produces good results for categories as diverse as cars and cows and that it can cope with cluttered backgrounds and partial occlusions.

For more accurate segmentation results, obviously, the combination with traditional contour or region based segmentation algorithms is required. The result images show that edges are quite prominent in those regions where our proposed algorithms has problems, such as on the car roofs or cow heads. On the other hand, category-specific knowledge can serve to resolve ambiguities between low-level image structures in those regions where our algorithm is confident. In short, both kinds of methods are mutually beneficial and should be combined, ideally in an iterative process. The probabilistic formulation of our algorithm lends itself to an easy integration with other segmentation methods.

More important than a pixel-accurate segmentation, however, is the fact that the probabilistic formulation gives us the opportunity to determine from where in the image a hypothesis draws its support. This information is valuable for two reasons. Firstly, it can be used to reduce the influence from the background and thus improve the reliability of recognition. Secondly, the per-pixel confidence map allows to resolve ambiguities from overlapping hypotheses and factor out the effects of partial occlusion. The following chapter will derive a hypothesis verification stage based on this idea in order to improve the recognition results. An extensive evaluation on two large databases will show that the resulting system is suitable for robust real-world recognition.

6

Multi-Object Scene Analysis

As presented in the previous chapter, our approach has only been evaluated on small data sets. The goal of this chapter is now to extend it to robust real-world recognition. As a first step, we therefore assess its current performance by evaluating the method on a real-world object detection task. The results from this analysis then serve to design a hypothesis verification stage which improves the method's results.

6.1 Performance of the Original Approach

In order to compare our method's performance to state-of-the-art approaches, we apply it to the UIUC car database (Agarwal and Roth, 2002). This test set consists of 170 images containing a total of 200 side views of cars. The images include instances of partially occluded cars, cars that have low contrast with the background, and images with highly textured backgrounds. In the data set, all cars have approximately the same size.

Together with the test set, Agarwal & Roth provide a training set of 550 car and 500 non-car images. In our experiments, we do not use this training set, but instead train on a much smaller set of only 50 hand-segmented images (mirrored to represent both car directions) that were originally prepared for a different experiment. In particular, our training set contains both European and American cars, whereas the test set mainly consists of American-style sedans and limousines. Thus, our detector remains more general and is not tuned to the specific test conditions. The original data set is at a relatively low resolution (with cars of size 100 × 40 pixels). Since our detector is learned at a higher resolution, we rescaled all test images by a constant factor prior to recognition (Note that this step does not increase the images' information content). All experiments are done using the evaluation scheme and detection tolerances from (Agarwal and Roth, 2002).

Figure 6.1 shows a recall-precision curve (RPC) of our method's performance. As can be seen from the figure, our method succeeds to generalize from the small training set and achieves a good recognition performance with an Equal Error Rate (EER) of 91% (corresponding to 182 out of 200 correct detections with 18 false positives).

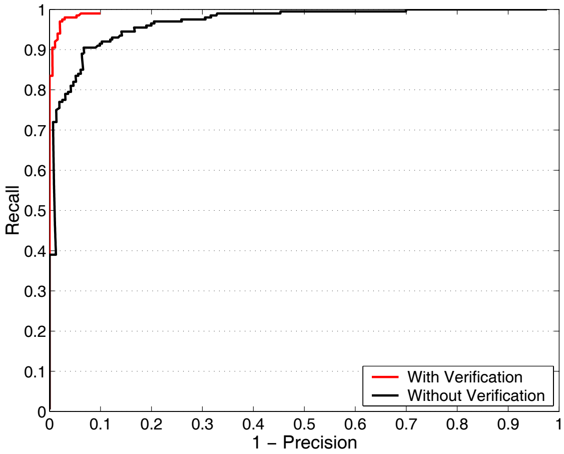


Figure 6.1: *Results on the UIUC car database with and without the MDL hypothesis verification stage.*

Analyzing the results on the test set, we observed that a large percentage of the remaining false positives were due to *secondary hypotheses*, which contain only one of the car's wheels, e.g. the rear wheel, but hypothesize it to be the front wheel of an adjoining car (see Figure 6.2 for an example). This problem is particularly prominent in scenes that contain multiple objects. The following section derives a hypothesis verification criterion which resolves these ambiguities in a natural fashion and thus improves the recognition results.

6.2 Hypothesis Verification Stage

6.2.1 Motivation

Up to now, we have integrated information from all patches in the image, as long as they agreed on a common object center. Indeed, this is the only available option in the absence of prior information about possible object locations. As a result, we had to tolerate false positives on highly textured regions in the background, where many matched patches caused random peaks in the voting space.

Now that a set of hypotheses is available, however, we can iterate on it and improve the recognition results. The previous chapter has shown that we can obtain a probabilistic top-down segmentation from each hypothesis and thus split its support into *figure* and *ground* pixels. The basic idea of this verification stage is now to only aggregate evidence over the figure portion of the image, that is over pixels that are hypothesized to belong to the object, and discard misleading information from the background. The motivation for this is that correct hypotheses will lead to consistent segmentations, since they are backed by an existing object in the image. False positives from random background clutter, on the other hand, will result in inconsistent segmentations and thus in lower *figure* probabilities.

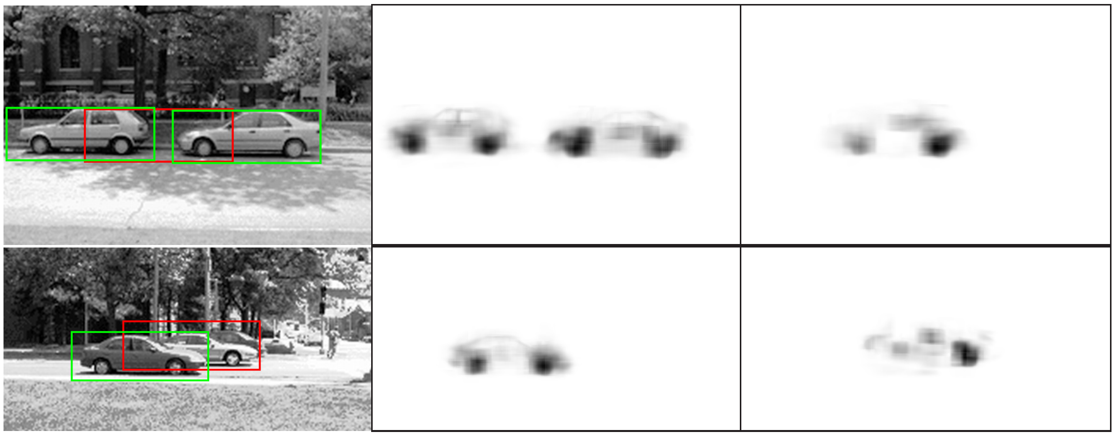


Figure 6.2: *(left) Two examples for overlapping hypotheses (in red); (middle)* $p(\mathbf{p} = \textit{figure}|h)$ *probabilities for the front and (right) for the overlapping hypotheses. As can be seen, the overlapping hypothesis in the above example is fully explained by the two correct detections, while the one in the lower example obtains additional support from a different region in the image.*

At the same time, this idea allows to compensate for a systematic bias in the initial voting scheme. The probabilistic votes are constructed on the principle that each patch has the same weight. This leads to a competitive advantage for hypotheses that contain more matched patches, e.g. because the area was more densely sampled by the interest point detector. Normalizing a hypothesis's score by the number of contributing patches, on the other hand, would not produce the desired results, because the patches can overlap and often contain background structure. By accumulating evidence now over the *figure* pixels, the verification stage removes this bias. Using this principle, each pixel has the same potential influence, regardless of how many sampled patches it is contained in.

Finally, this strategy makes it possible to resolve ambiguities from overlapping hypotheses in a principled manner. As already mentioned in the previous section, a large number of the initial false positives are due to secondary hypotheses which overlap part of the object. This is a common problem in object detection. Generating such hypotheses is a desired property of a recognition algorithm, since it allows the method to cope with partial occlusions. However, if enough support is present in the image, the secondary detections should be sacrificed in favor of other hypotheses that better explain the image. Usually, this problem is solved by introducing a bounding box criterion and rejecting weaker hypotheses based on their overlap. However, such an approach may lead to missed detections, as the example in Figure 6.2 shows. Here the overlapping hypothesis really corresponds to a second car, which would be rejected by the simple bounding box criterion (Incidentally, only the front car is labelled as “car” in the test set, possibly because of that problem).

Again, since our algorithm provides us with an object segmentation together with the hypotheses, we can improve on this and exactly quantify how much support the overlapping region contains for each hypothesis. In particular, this permits us to detect secondary hypotheses, which draw all their support from areas that are already better explained by other hypotheses, and distinguish them from true overlapping objects. In the following, we derive a criterion based on the principle of Minimal Description Length (MDL), inspired by the approach pursued in (Leonardis et al., 1995), which combines all of these motivations.

6.2.2 MDL Formulation

The MDL principle is an information theoretic formalization of the general notion to prefer simple explanations to more complicated ones. In our context, a pixel can be described either by its grayvalue or by its membership to a scene object. If it is explained as part of an object, we also need to encode the presence of the object ("model cost"), as well as the error that is made by this representation. The MDL principle states that the best encoding is the one that minimizes the total description length for image, model, and error.

In accordance with the notion of description length, we can define the *savings* (Leonardis et al., 1995) in the encoding that can be obtained by explaining part of an image by the hypothesis h:

$$S_h = K_0 S_{area} - K_1 S_{model} - K_2 S_{error} \tag{6.1}$$

In this formulation, S_{area} corresponds to the number N of pixels that can be explained by h; S_{error} denotes the cost for describing the error made by this explanation; and S_{model} describes the model complexity. In our implementation, we assume a fixed cost $S_{model} = 1$ for each additional scene object. As an estimate for the error we use

$$S_{error} = \sum_{\mathbf{p} \in Seg(h)} (1 - p(\mathbf{p} = \mathit{figure}|h)) \tag{6.2}$$

that is, over all pixels that are hypothesized to belong to the segmentation of h, we sum the probabilities that these pixels are not *figure*.

The constants K_0, K_1, and K_2 are related to the average cost of specifying the segmented object area, the model, and the error, respectively. They can be determined on a purely information-theoretical basis (in terms of bits), or they can be adjusted in order to express the preference for a particular type of description. In practice, we only need to consider the relative savings between different combinations of hypotheses. Thus, we can divide Eq. (6.1) by K_0 and, after some simplification steps, we obtain

$$S_h = -\frac{K_1}{K_0} + (1 - \frac{K_2}{K_0})N + \frac{K_2}{K_0} \sum_{\mathbf{p} \in Seg(h)} p(\mathbf{p} = \mathit{figure}|h). \tag{6.3}$$

This leaves us with two parameters: $\frac{K_2}{K_0}$, which encodes the relative importance that is assigned to the support of a hypothesis, as opposed to the area it explains; and $\frac{K_1}{K_0}$, which specifies the total weight a hypothesis must accumulate in order to provide any savings. Good values for these parameters can be found by considering some limiting cases, such as the minimum support a hypothesis must have in the image, before it should be accepted.

Using this framework, we can now resolve conflicts between overlapping hypotheses. Given two hypotheses h_1 and h_2, we can derive the savings of the *combined hypothesis* $(h_1 \cup h_2)$:

$$S_{h_1 \cup h_2} = S_{h_1} + S_{h_2} - S_{area}(h_1 \cap h_2) + S_{error}(h_1 \cap h_2) \tag{6.4}$$

Both the overlapping area and the error can be computed from the segmentations obtained in Section 5.2. $S_{area}(h_1 \cap h_2)$ is just the area of overlap between the two segmentations. Let h_1 be the stronger hypothesis of the two. Under the assumption that h_1 opaquely occludes h_2, we can adjust for the error term $S_{error}(h_1 \cap h_2)$ by setting $p(\mathbf{p} = \mathit{figure}|h_2) = 0$ wherever $p(\mathbf{p} = \mathit{figure}|h_1) > p(\mathbf{p} = \mathit{ground}|h_1)$, that is for all pixels that belong to the segmentation of h_1.

The goal of this procedure is to find the combination of hypotheses that provides the maximum savings and thus best explains the image. Leonardis et al. (1995) showed that this can be formulated as a quadratic Boolean optimization problem as follows. Let $m^T = (m_1, m_2, \ldots, m_M)$ be a vector of indicator variables, where m_i has the value 1 if hypothesis h_i is present, and 0 if it is absent in the final description. In this formulation, the objective function for maximizing the savings takes the following form:

$$S(\hat{m}) = \max_m S(m) = \max_m m^T Q m = m^T \begin{bmatrix} c_{11} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{M1} & \cdots & c_{MM} \end{bmatrix} m. \tag{6.5}$$

The diagonal terms of Q express the savings of a particular hypothesis h_i

$$c_{ii} = S_{h_i} = -\frac{K_1}{K_0} + (1 - \frac{K_2}{K_0})N + \frac{K_2}{K_0} \sum_{\mathbf{p} \in Seg(h_i)} p(\mathbf{p} = \mathit{figure}|h_i) \tag{6.6}$$

while the off-diagonal terms handle the interaction between overlapping hypotheses

$$c_{ij} = \frac{1}{2} \left(-(1 - \frac{K_2}{K_0})|O_{ij}| - \frac{K_2}{K_0} \sum_{\mathbf{p} \in O_{ij}} \min_{i,j} p(\mathbf{p} = \mathit{figure}|h) \right) \tag{6.7}$$

where $O_{ij} = Seg(h_i) \cap Seg(h_j)$ denotes the area of overlap between the segmentations of h_i and h_j. As the number of possible combinations grows exponentially with increasing problem size, though, it may become untractable to search for the globally optimal solution. In practice, we found it sufficient for our application to only compute a greedy approximation (see Algorithm 6.1), as also argued in (Leonardis et al., 1995).

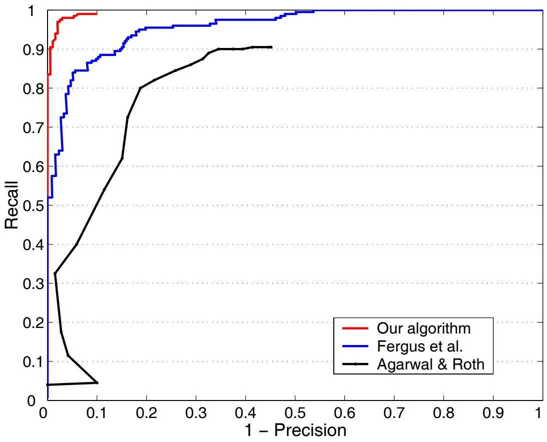
Algorithm 6.1 The greedy MDL verification algorithm.

```
// Given: hypotheses $H$ and corresponding segmentations $\left\{(img^{(i)}_{pfig}, img^{(i)}_{pgnd})\right\}$.

// Build up the matrix $Q = \{c_{ij}\}$
for all hypotheses $h_i \in H$ do
    $sum \leftarrow 0$, $N \leftarrow 0$
    for all pixels $\mathbf{p} \in img$ do
        if $img^{(i)}_{pfig}(\mathbf{p}) > img^{(i)}_{pgnd}(\mathbf{p})$ then
            $sum \leftarrow sum + img^{(i)}_{pfig}(\mathbf{p})$
            $N \leftarrow N + 1$
    $c_{ii} \leftarrow -\frac{K_1}{K_0} + (1 - \frac{K_2}{K_0})N + \frac{K_2}{K_0} sum$

    for all hypotheses $h_j \in H, j \neq i$ do
        $sum \leftarrow 0$, $N \leftarrow 0$
        for all pixels $\mathbf{p} \in img$ do
            if $\left(img^{(i)}_{pfig}(\mathbf{p}) > img^{(i)}_{pgnd}(\mathbf{p})\right) \wedge \left(img^{(j)}_{pfig}(\mathbf{p}) > img^{(j)}_{pgnd}(\mathbf{p})\right)$ then
                $sum \leftarrow sum + \min\left(img^{(i)}_{pfig}(\mathbf{p}), img^{(j)}_{pfig}(\mathbf{p})\right)$
                $N \leftarrow N + 1$
        $c_{ij} \leftarrow \frac{1}{2}\left(-(1 - \frac{K_2}{K_0})N - \frac{K_2}{K_0} sum\right)$

// Search for the best combination of hypotheses
$m \leftarrow (0, 0, \ldots, 0)$, finished $\leftarrow$ false
repeat
    for all unselected hypotheses $h_i$ do
        $\tilde{m} \leftarrow m$, $\tilde{m}(i) \leftarrow 1$
        $S_i \leftarrow \tilde{m}^T Q \tilde{m} - m^T Q m$
    $k = \arg\max_i(S_i)$
    if $S_k > 0$ then
        $m(k) \leftarrow 1$
    else
        finished $\leftarrow$ true
until finished
```

6.3 Experimental Evaluation

6.3.1 Evaluation on a Car Detection Task

Figure 6.1 shows the results on the UIUC car database when the MDL criterion is applied as a verification stage. As can be seen from the figure, the results are significantly improved, and the EER performance increases from 91% to 97.5%. Without the verification stage, our algorithm could reach this recall rate only at the price of a reduced precision of only 74.1%. This means that for the same recall rate,



Method	EER
Agarwal and Roth (2002)	∼79%
Garg et al. (2002)	∼88%
Fergus et al. (2003)	88.5%
Our algorithm	97.5%

Figure 6.3: *Comparison of our results on the UIUC car database with others reported in the literature.*

the verification stage manages to reject 64 additional false positives while keeping all correct detections. In addition, the results become far more stable over a wider parameter range than before. This can be illustrated by the fact that even when the initial acceptance threshold is lowered to zero, the MDL criterion does not return more than 20 false positives. This property, together with the criterion's good theoretical foundation and its ability to correctly solve cases like the one in Figure 6.2, makes it an important addition to the system.

Figure 6.3 shows a comparison of our method's performance with other results reported in the literature. The adjacent table contains a comparison of the equal error rates (EER) with three other approaches. With an EER of 97.5%, our method presents a significant improvement over previous results. Some example detections in difficult settings can be seen in Figure 6.4. The images show that our method still works in the presence of occlusion, low contrast, and cluttered backgrounds. At the EER point, our method correctly finds 195 of the 200 test cases with only 5 false positives. All of these cases are displayed in Figure 6.5. The main reasons for missing detections are combinations of several factors, such as low contrast, occlusion, and image plane rotations, that push the object hypothesis below the acceptance threshold. The false positives are due to richly textured backgrounds on which a large number of spurious object parts are found.

In addition to the recognition results, our method automatically generates object segmentations from the test images. Figures 6.6-6.8 show some example segmentations that can be achieved with this method. Even though the quality of the original images is rather low, the segmentations are reliable and can serve as a basis for later processing stages, e.g. to further improve the recognition results using global methods. In particular the examples in Figure 6.8 show that the system can not only detect cars despite partial occlusion, but it is often even able to segment out the occluding structure[1].

---

[1] In the presented examples, our method is also able to segment out the car windows, since they

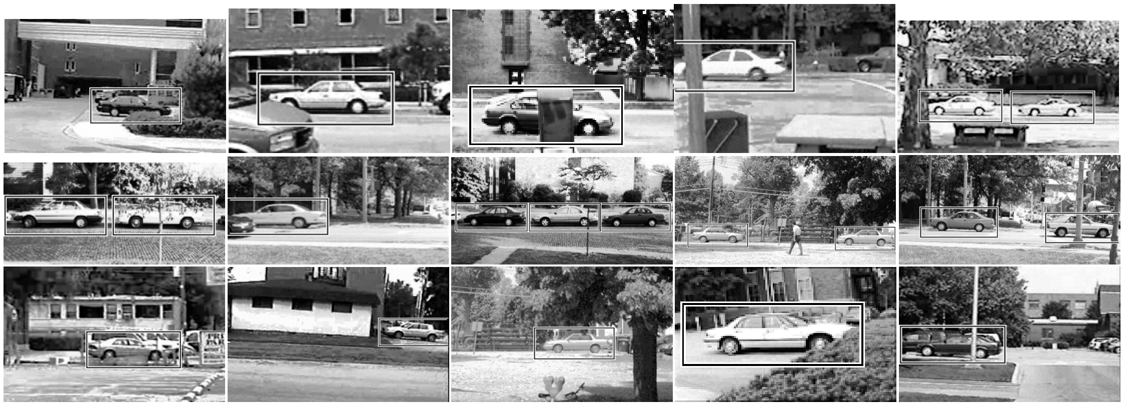


Figure 6.4: *Example detections on difficult images from the test set.*

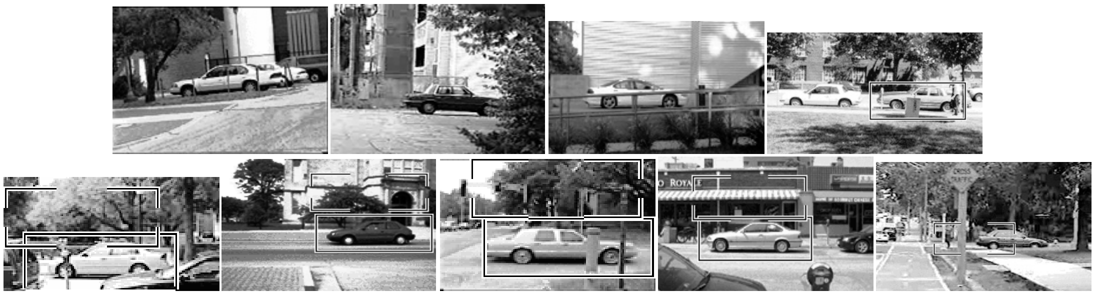


Figure 6.5: *All missing detections (above) and false positives (below) our algorithm returned on the car test set. The last picture contains both a false positive and a missing detection.*

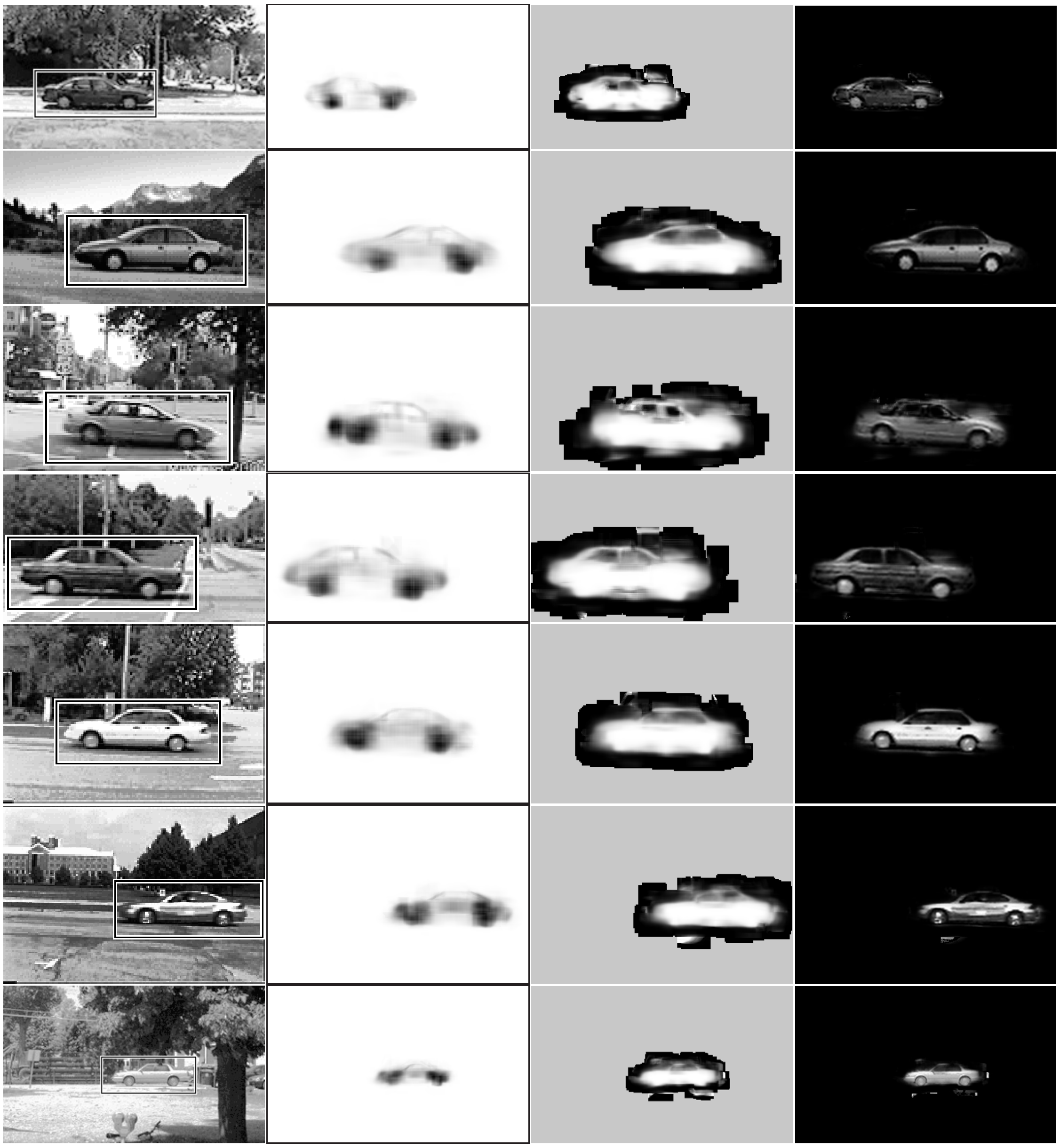
6.3.2 Recognition of Articulated Objects

Up to now, we have only considered static objects in our experiments. Even though environmental conditions can vary greatly, cars are still rather restricted in their possible shapes. This changes when we consider articulated objects, such as walking animals. In order to fully demonstrate our method's capabilities, we therefore apply it to a database of video sequences of walking cows originally used for detecting lameness in livestock (Magee and Boyle, 2002). Each sequence shows one or more cows walking from right to left in front of different, static backgrounds.

For training, we took out all sequences corresponding to three backgrounds and extracted 113 randomly chosen frames, for which we manually created a reference segmentation. We then tested on 14 different video sequences showing a total of 18 unseen cows in front of different backgrounds and with varying lighting conditions. Some test sequences contain severe interlacing and MPEG-compression artifacts and significant noise. Altogether, the test suite consists of a total of 2217 frames, in which 1682 instances of cows are visible by at least 50%. This provides us with a significant number of test cases to quantify both our method's ability to deal with different

---

were labelled *ground* in the training data.

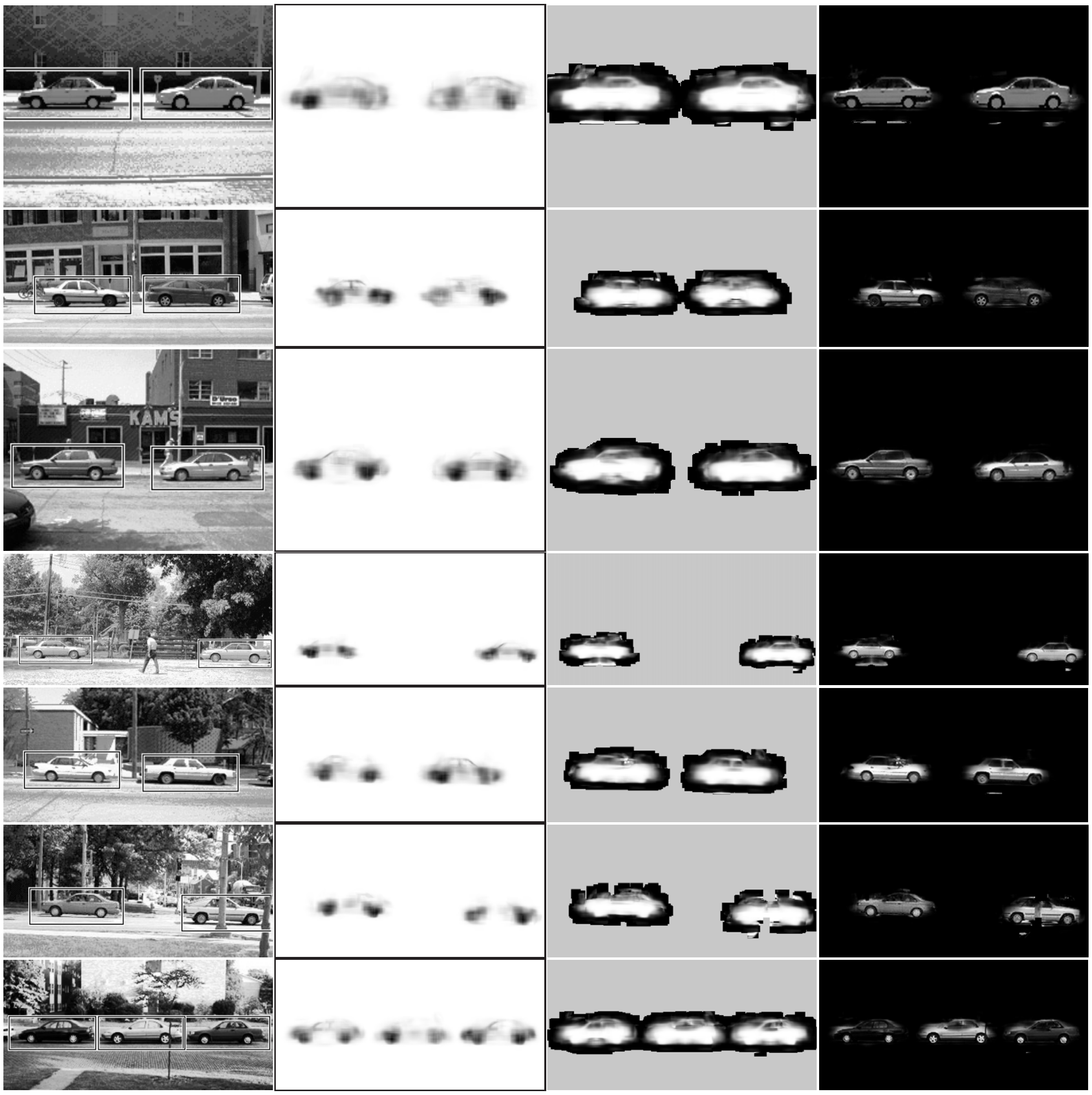


(a) detections (b) p(figure) (c) segmentation (d) segm. image

Figure 6.6: *Example object detections, figure probabilities, and segmentations automatically generated by our method.*

articulations and its robustness to occlusion. Using video sequences for testing also allows to avoid any bias caused by selecting only certain frames. However, since we are still interested in a single-frame recognition scenario, we apply our algorithm to each frame separately. That is, no temporal continuity information is used for recognition, which one would obviously add for a tracking scenario.

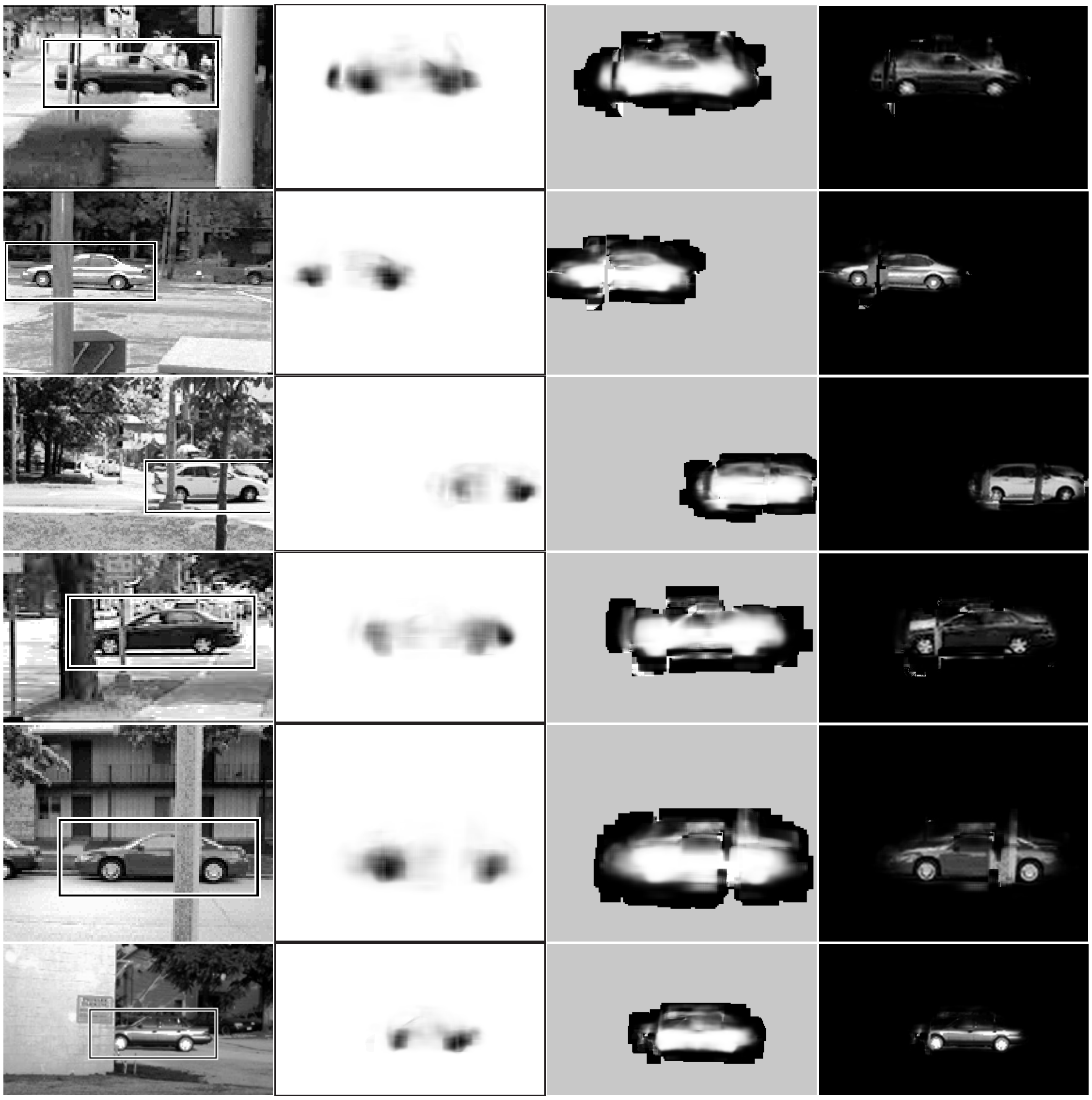
We applied our method to this test set using exactly the same detector settings as before to obtain equal error rate for the car experiments. The only change we



(a) detections (b) p(figure) (c) segmentation (d) segm. image

Figure 6.7: *Object detections, figure probabilities, and segmentations for scenes containing multiple objects.*

made was to slightly adjust the sensitivity of the interest point detector in order to compensate for the lower image contrast. Using these settings, our detector correctly finds 1535 out of the 1682 cows, corresponding to a recall of 91.2%. With only 30 false positives over all 2217 frames, the overall precision is at 98.0%. Figure 6.9 shows the precision and recall values as a function of the visible object area. As can be seen from this plot, the method has no difficulties in recognizing cows that are fully visible (99.1% recall at 99.5% precision). Moreover, it can cope with significant partial occlusion. When only 60% of the object is visible, recall only drops to 79.8%. Even when half the object is occluded, the recognition rate is still at 69.0%. In some



(a) detections (b) p(figure) (c) segmentation (d) segm. image

Figure 6.8: *Object detections, figure probabilities, and segmentations for scenes containing occluding structure.*

rare cases, even a very small object portion of about 20–30% is already enough for recognition (such as in the leftmost image in Figure 6.12). Precision constantly stays at a high level.

False positives mainly occur when only one pair of legs is fully visible and the system generates a competing hypothesis interpreting the front legs as rear legs, or vice versa. Usually, such secondary hypotheses are filtered out by the MDL stage, but if the correct hypothesis does not have enough support in the image due to partial visibility, the secondary hypothesis sometimes wins.

A more detailed analysis of the results can be obtained when looking at the

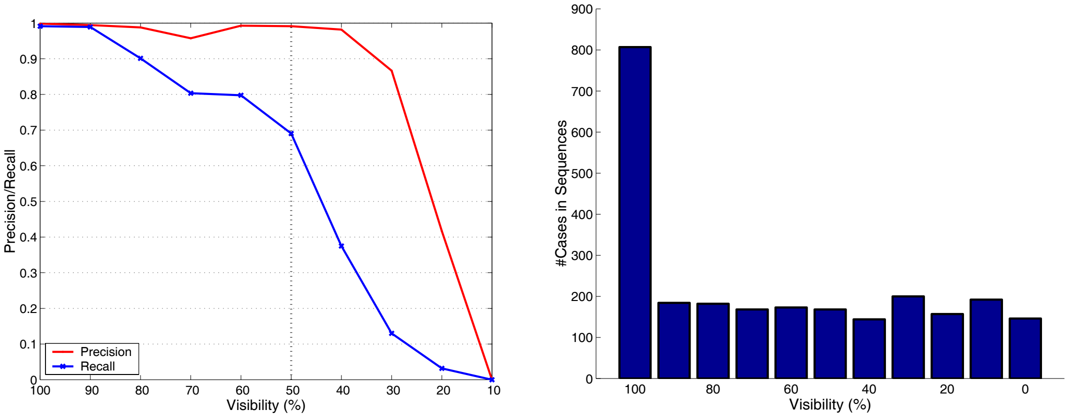


Figure 6.9: *(left) Precision/Recall curves for the cow sequences when x% of the cow's length is visible. (right) Absolute number of test images for the different visibility cases.*

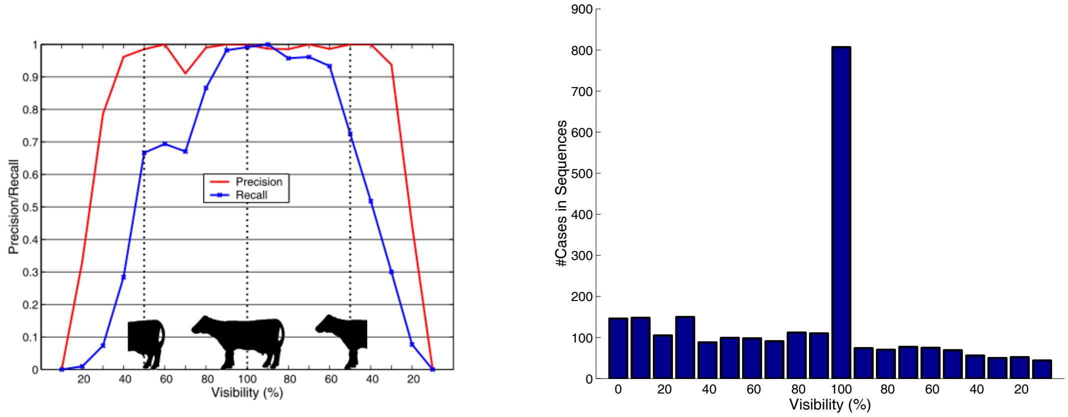


Figure 6.10: *(left) Precision/Recall curves for the cow sequences when x% of the cow's length is visible (and when front and rear cases are distinguished). (right) Absolute number of test images for the different visibility cases.*

information which part of the cow (the front or the rear) is visible. Figure 6.10 shows the precision and recall curves when these two cases are distinguished. It can be seen that the front part is more discriminant, yielding higher recall scores and allowing for recognition under stronger partial occlusion. Indeed, the strongest performance hit in both precision and recall occurs when only the rear 70% of the cow are visible, i.e. when just the head and one front leg are occluded by the image boundary. Again, the reason for this behavior is that a secondary hypothesis misinterprets the visible rear legs as the front legs of an adjoining cow and thus suppresses the correct interpretation.

Figures 6.11–6.13 show example detection and segmentation results for three of the sequences used in this evaluation. As can be seen from these images, the system

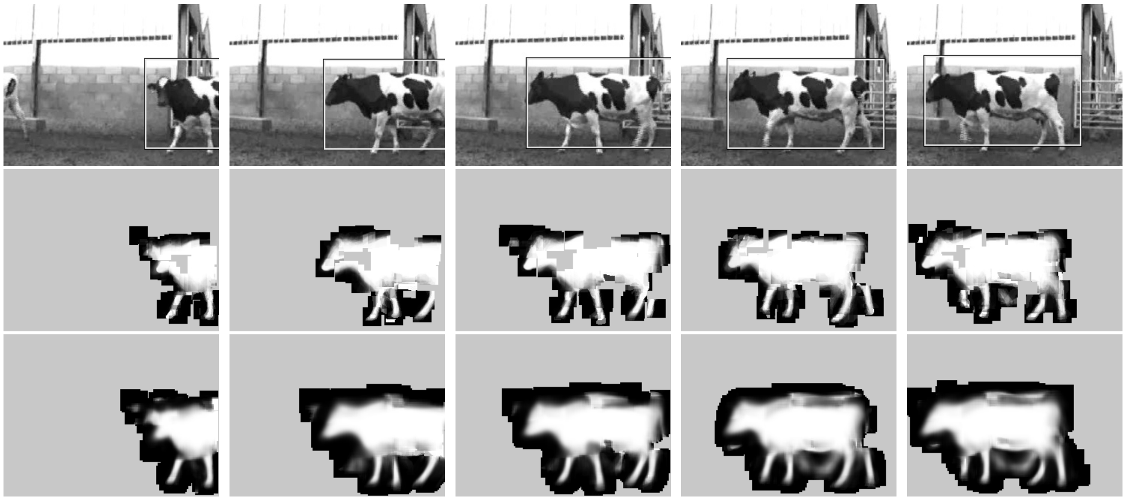


Figure 6.11: *Example detections and automatically generated segmentations from one cow sequence. (middle row) segmentations obtained from the initial hypotheses; (bottom row) segmentations from refined hypotheses.*

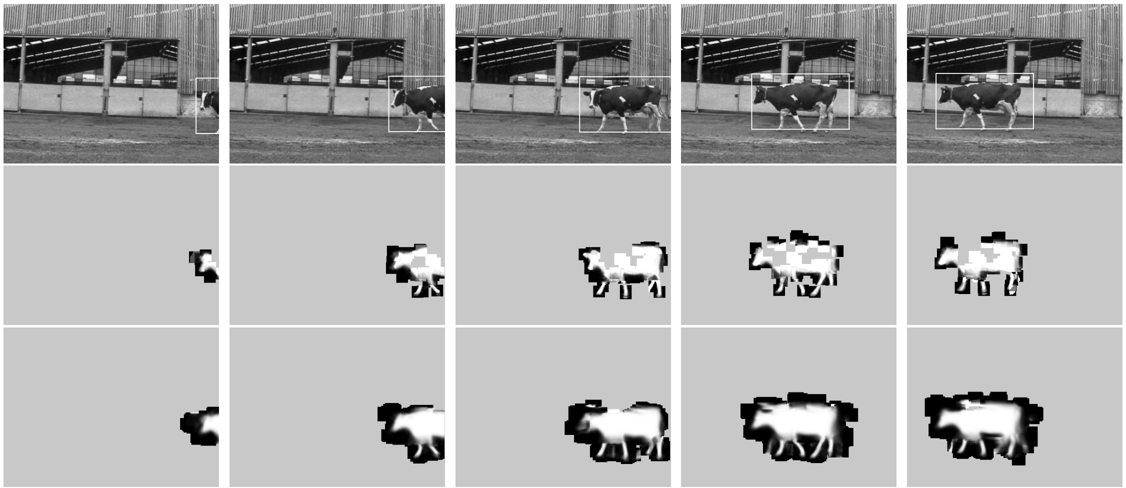


Figure 6.12: *Example detections and automatically generated segmentations from another cow sequence. Note in particular the leftmost image, where the cow is correctly recognized and segmented despite a high degree of occlusion.*

not only manages to recognize unseen-before cows with novel texture patterns, but it also provides good segmentations for them. Again, we want to emphasize that no tracking information is used to generate these results. On the contrary, the capability to generate object segmentations from single frames could make our method a valuable supplement to many current tracking algorithms, allowing to (re-)initialize them through shape cues that are orthogonal to those gained from motion estimates.

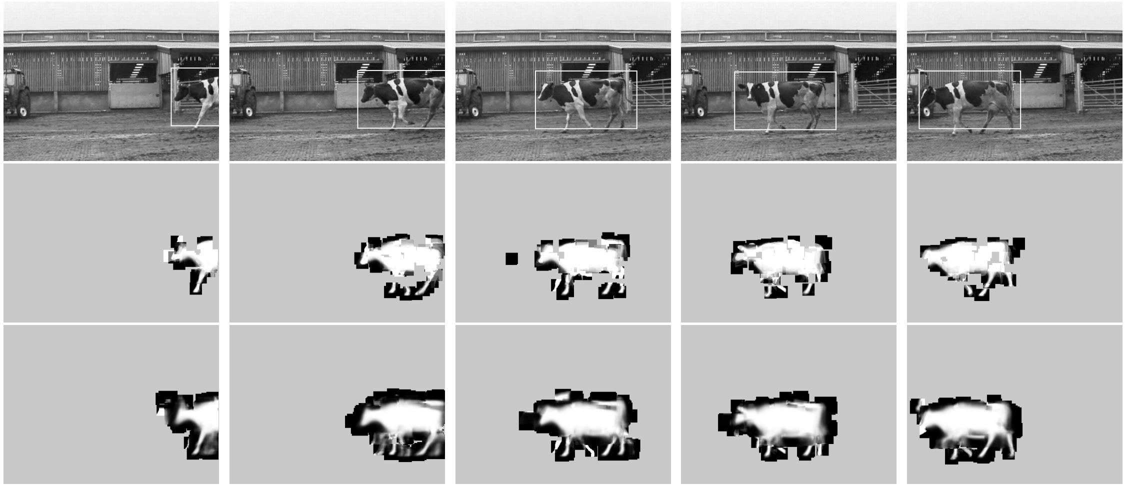


Figure 6.13: *Example detections and automatically generated segmentations from a third cow sequence. Note the low contrast to the background.*

6.4 Discussion

In this chapter, we have introduced a novel hypothesis verification stage that extends our recognition approach. It uses the probabilistic segmentation automatically generated for each hypothesis to aggregate only evidence over the figure portion of the image and discard contributions from the background. In addition, its formulation in an MDL framework allows to resolve ambiguities between overlapping hypotheses and handle scenes containing multiple objects in a principled manner. As a result, the verification criterion significantly improves the method's performance. In addition, we have presented an extensive evaluation on two large data sets for cars and cows. Our results show that the system achieves excellent recognition and segmentation results, even under adverse viewing conditions and with significant partial occlusion. At the same time, its flexible representation allows it to generalize already from small training sets.

Several factors are responsible for our method's good performance. One reason is demonstrated by the probabilities $p(\mathbf{p} = \mathit{figure}|h)$ in Figs. 6.2 and 6.6–6.8. These probabilities correspond to the per-pixel confidence the system has in its recognition and segmentation result. As can be seen from the figures, the cars' wheels are found as the most important single feature. However, the rest of the chassis and even the windows are represented as well. Together, they provide additional support for the hypothesis. This is possible because we do not perform any feature selection during the training stage, but store all local parts that are repeatedly encountered on the training objects. The resulting complete representation allows our approach to compensate for missing detections and partial occlusion.

Another factor is the flexibility of representation that is made possible by the Implicit Shape Model. Using this framework, the method can interpolate between local parts seen on different training objects. As a result, it only needs a relatively small number of training examples to recognize and segment categorical objects in different articulations and with widely varying texture patterns.

The price we have to pay for this flexibility is that local parts could also be matched to potentially illegal configurations, such as a cow with six legs. Since each hypothesized leg is locally consistent with the common object center, there would be nothing to prevent such configurations. In our experiments, however, the MDL criterion effectively solves this problem. Another solution would be to add a global, explicit shape model on top of our current implicit model. Using the obtained object segmentations as a guide, such a model could be learned on-line, even after the initial training stage.

The Implicit Shape Model itself is an instance of a more general functional principle. Real-world images of an object category may contain so much variation that it would be a hopeless endeavor to model all of it in one global representation. Instead, we map small sub-structures onto an internal representation, in our case the appearance codebook. The comparatively small number and local nature of those codebook entries allows us to learn more complex relationships between them, e.g. their spatial probability distribution and their respective figure-ground labels. Hypotheses about a test object's nature are then evaluated in terms of how consistent the matched codebook entries' arrangement is with their learned distribution.

However, when pursuing such an approach, it is important to represent the uncertainty on all levels: while matching the unknown image content to the known codebook representation; and while accumulating the evidence of multiple such matches (Gibson, 1957). In that sense, the reliability and robustness of the observed results would not be possible without the probabilistic codebook matching process presented in Section 5.1.3. In particular, the results would be much less stable if this step were done in a simple nearest-neighbor fashion.

Altogether, we have presented an iterative evidence aggregation scheme, which interleaves the processes of recognition and segmentation to maximize the amount of information that is extracted from novel test images. A natural extension would be to iterate this process further. For example, a stochastic sampling strategy could be pursued to actively sample locations that have not yet been selected by the initial interest point detector, but that would provide additional information (as determined by the *figure* probability map) about the object. This could be used to compensate for an irregular sampling density of the interest point detector, to gain additional evidence for distinguishing true detections from false positives, or to discriminate between multiple object categories

Another extension would be to base the decision on multiple cues. Since the recognition framework is based on the accumulation of probabilistic votes, any local measurement that produces such votes can be used. As long as a cue can be represented in terms of prototypical codebook vectors, it can easily be integrated into the framework. Section 8.4.1 discusses such an extension in more detail.

An important restriction of our approach, as it is described so far, is that it tolerates only small scale changes of about 10–15%. The reason for this is that both the patch extraction stage and the voting procedure only look for structures of a particular size. As our next step, we will therefore extend the approach to multiple scales. The following chapter describes how such an extension can be achieved by using scale-invariant interest points and incorporating a scale component in the patch voting framework.

7

Scale-Invariant Object Categorization

Robustness to scale changes is one of the most important properties of any recognition system that shall be applied in real-world situations. Even when the camera location is relatively fixed, objects of interest may still exhibit scale changes of at least a factor of two, simply because they occur at different distances to the camera. While we can generally assume that in the context of an embodied system, an attention mechanism may provide a rough scale estimate, the expected precision of this estimate will not be too high. It is thus necessary that the recognition mechanism itself can compensate for a certain degree of scale variation.

Many current object detection methods deal with the scale problem by performing an exhaustive search over all possible object positions and scales (Papageorgiou and Poggio, 2000; Schneiderman and Kanade, 2000; Viola and Jones, 2001). This exhaustive search imposes severe constraints, both on the detector's computational complexity and on its discriminance, since a large number of potential false positives need to be excluded.

An opposite approach is to let the search be guided by image structures that give cues about the object scale. In such a system, an initial interest point detector tries to find structures whose extent can be reliably estimated under scale changes. These structures are then combined to derive a comparatively small number of hypotheses for object locations and scales. Only those hypotheses that pass an initial plausibility test need to be examined in detail. In recent years, a range of scale-invariant interest point detectors have become available which can be used for this purpose (Lindeberg, 1998; Lowe, 1999; Mikolajczyk and Schmid, 2001; Kadir and Brady, 2001; Tuytelaars and van Gool, 2000, 2004; Matas et al., 2002).

In this chapter, we apply the second idea to achieve robustness to scale. The chapter contains four main contributions: (1) We extend our approach to multi-scale object categorization, making it thus usable in practice. Our extension is based on the use of scale-invariant interest point detectors, as motivated above. (2) We formulate the multi-scale object detection problem in a Mean-Shift framework, which allows to draw parallels to Parzen window probability density estimation. We show that the introduction of a scale dimension in this scheme requires the Mean-Shift approach to be extended by a scale adaption mechanism that is different from the variable-bandwidth methods proposed so far (Comaniciu et al., 2001; Collins, 2003). (3) We experimentally evaluate the suitability of different scale-invariant interest point detectors and analyze their influence on the recognition results. Interest point detectors have so far mainly been evaluated in terms of repeatability and the ability to find exact correspondences (Mikolajczyk and Schmid, 2001, 2003). As our task requires the generalization to unseen objects, we are more interested in finding similar and typical structures, which imposes different constraints on the detectors. (4) Last but not least, we experimentally evaluate the robustness of the proposed approach to large scale changes. While other approaches have used multi-scale interest points also for object class recognition (Dorko and Schmid, 2003; Fergus et al., 2003; Kadir et al., 2004), no quantitative analysis of their robustness to scale changes has been reported. Our results show that the proposed approach outperforms state-of-the-art methods while being robust to scale changes of more than a factor of two. In addition, our quantitative results allow to draw some interesting conclusions for the design of suitable interest point detectors.

The chapter is structured as follows. The next section describes how our approach can be extended to multiple scales. Section 7.2 then examines the influence of different interest point detectors on the recognition result. Finally, Section 7.3 evaluates the robustness to scale changes, and Section 7.4 explores the effect of the training set size.

7.1 Extended Approach

7.1.1 Extended Probabilistic Framework

A major point of this chapter is to extend recognition to multiple scales using scale-invariant interest points. The basic idea behind this is to replace the single-scale Harris codebook used up to now by a codebook derived from a scale-invariant detector. Given an input image, the system applies the detector and obtains a vector of point locations, together with their associated scales. Patches are extracted around the detected locations with a radius relative to the scale σ of the interest point (here: $r = 3\sigma$). In order to match image structures at different scales, the patches are then rescaled to the codebook size (in our case 25×25 pixels).

The probabilistic framework can be readily extended to multiple scales by treating scale as a third dimension in the voting space (Leibe and Schiele, 2004). If an image patch found at location $(x_{img}, y_{img}, s_{img})$ matches to a codebook entry that has been observed at position $(x_{occ}, y_{occ}, s_{occ})$ on a training image, it votes for the following coordinates:

$$x_{vote} = x_{img} - x_{occ}(s_{img}/s_{occ}) \tag{7.1}$$

$$y_{vote} = y_{img} - y_{occ}(s_{img}/s_{occ}) \tag{7.2}$$

$$s_{vote} = (s_{img}/s_{occ}) \tag{7.3}$$

However, the increased dimension of the voting space makes the maxima search computationally more expensive. For this reason, we employ a two-stage search strategy. In a first stage, votes are collected in a binned 3D Hough accumulator array in order to quickly find local maxima. Candidate maxima from this first stage are then refined in the second stage using the original (continuous) 3D votes. Instead of a simple but expensive sliding-window technique, we formulate the search in a Mean-Shift framework. For this, we replace the simple search window W from equation (5.5) by the following kernel density estimate:

$$\hat{p}(o_n, x) = \frac{1}{nh^d} \sum_k \sum_j p(o_n, x_j | \mathbf{e}_k, \ell_k) K(\frac{x - x_j}{h}) \tag{7.4}$$

where the kernel K is a radially symmetric, nonnegative function, centered at zero and integrating to one; h is the kernel bandwidth; h^d is its volume; and n is the number of points inside the kernel window. From (Comaniciu and Meer, 2002), we know that a Mean-Shift search using this formulation will quickly converge to local modes of the underlying distribution. Moreover, the search procedure can be interpreted as a Parzen window probability density estimation for the position of the object center.

From the literature, it is also known that the performance of the Mean-Shift procedure depends critically on a good selection for the kernel bandwidth h. Various approaches have been proposed to estimate the optimal bandwidth directly from the data (e.g. Comaniciu et al., 2001; Collins, 2003). In our case, however, we have an intuitive interpretation for the bandwidth as a search window for the position of the object center. As the object scale increases, the *relative errors* introduced by equations (7.1)-(7.3) cause votes to be spread over a larger area around the hypothesized object center and thus reduce their density in the voting space. As a consequence, the kernel bandwidth should also increase in order to compensate for this effect. We can thus make the bandwidth dependent on the scale coordinate and obtain the following *balloon density estimator* (Comaniciu et al., 2001):

$$\hat{p}(o_n, x) = \frac{1}{nh(x)^d} \sum_k \sum_j p(o_n, x_j | \mathbf{e}_k, \ell_k) K(\frac{x - x_j}{h(x)}) \tag{7.5}$$

For K we use a uniform spherical or cubical kernel with a radius corresponding to 5% of the hypothesized object size. Since a certain minimum bandwidth needs to be maintained for small scales, though, we only adapt it for scales greater than 1.0.

We have thus formulated the multi-scale object detection problem as a scale-adaptive Mean-Shift search procedure. Our experimental results in Section 7.3 will show that this scale adaptation step is indeed needed in order to provide stable results over large scale changes.

7.1.2 Extended Verification Stage

The key idea of the verification stage is that each hypothesis is judged based on the amount of support it can accumulate over the pixels it rates as *figure*. Obviously, the size of this supporting area depends on the object scale, so we also need to adapt the MDL formulation for the multi-scale case.

Since objects at different scales take up different portions of the image, we can no longer assume a fixed model cost. Instead, we make the model cost dependent on the *expected area* A_s an object occupies at a certain scale. When dealing with only one object category, the true area A_s can be replaced by the simpler term s^2, since the expected area grows quadratically with the object scale and the constant K_1 can be set to incorporate the proportionality factor. However, when multiple categories or different views of the same object category are searched for, the model cost needs to reflect their relative size differences.

By setting the model cost to $S_{model} = A_s$, we obtain the following formulation:

$$S_h \quad = \quad -\frac{K_1}{K_0} + (1 - \frac{K_2}{K_0})\frac{N}{A_s} + \frac{K_2}{K_0}\frac{1}{A_s}\sum_{\mathbf{p}\in Seg(h)} p(\mathbf{p} = \mathit{figure}|h). \tag{7.6}$$

So, a single hypothesis in the image (that does not overlap with any other hypothesis) is now accepted if

$$(1 - \frac{K_2}{K_0})\frac{N}{A_s} + \frac{K_2}{K_0}\frac{1}{A_s}\sum_{\mathbf{p}\in Seg(h)} p(\mathbf{p} = \mathit{figure}|h) \quad \geq \quad \frac{K_1}{K_0}. \tag{7.7}$$

If multiple hypotheses are present in the scene, the known mechanism from Chapter 6 is applied to resolve conflicts and trade off ambiguities between them.

The performance of the resulting approach depends on the capability of the underlying patch extractor to find image structures that are both typical for the object category and that can be accurately localized in position and scale. As different detectors are optimized for finding different types of structures, the next section evaluates the suitability of various scale-invariant interest point detectors for categorization

7.2 Influence of the Interest Point Detector

Typically, interest point detectors are only evaluated in terms of their repeatability. Consequently, significant effort has been spent on making the detectors discriminant enough that they find exactly the same structures again under different viewing conditions. However, we strongly believe that the evaluation should be in the context of a task. In our case, the task is to recognize and localize previously unseen objects of a given category. This means that we cannot assume to find exactly the same structures again; instead the system needs to generalize and find structures that are similar enough to known object parts while still allowing enough flexibility to cope with variations. Also, because of the large intra-class variability, more potential matching candidates are needed to compensate for inevitable mismatches. Last but not least, the interest points should provide a sufficient cover of the object, so that

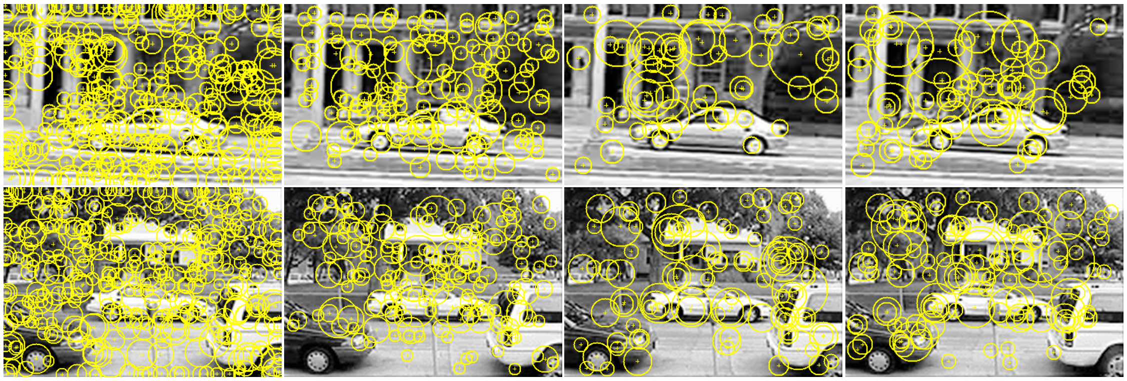


Figure 7.1: *Scale-invariant interest points found by (from left to right) the exact DoG, the fast DoG, the regular Harris-Laplace, and the fast Harris-Laplace detector on two example images (The smallest scales are omitted in order to reduce clutter).*

it can be recognized even if some important parts are occluded. Altogether, this imposes a rather different set of constraints on the interest point detector. As a first step we therefore have to compare the performance of different interest point operators for the categorization task.

In this work, we evaluate two different types of scale-invariant interest point operators: the Harris-Laplace detector (Mikolajczyk and Schmid, 2001), and the DoG (Difference of Gaussian) detector (Lowe, 1999). Both operators have been shown to yield high repeatability (Mikolajczyk and Schmid, 2003), but they differ in the type of structures they respond to. The Harris-Laplace prefers corner-like structures by searching for multi-scale Harris points that are simultaneously extrema of a scale-space Laplacian, while the DoG detector selects blob-like structures by searching for scale-space maxima of a Difference-of-Gaussian (a more detailed description of the two detectors can be found in Appendix A). For both detectors, we additionally examine two variants: a regular and a speed-optimized implementation (operating on a Gaussian pyramid). Figure 7.1 shows the kind of structures that are captured by the different detectors. As can already be observed from these examples, all detectors manage to capture some characteristic object parts, such as the car's wheels, but the range of scales and the distribution of points over the object varies considerably between them.

In order to obtain a more quantitative assessment of their capabilities, we compare the different interest point operators on a car detection task using our extended approach. As a test set, we again use the UIUC database (Agarwal and Roth, 2002). For all experiments reported below, training is done on the same set of 50 segmented images (mirrored to represent both car directions) as in Chapter 6. In a first stage, we compare the recognition performance if the test images are of the same size as the training images. Since our detectors are learned at a higher resolution than the cars in the test set, we rescale all test images by the same factor prior to recognition.

Figure 7.2(left) shows a comparison of the detectors' performances using only the initial patch votes. It can be seen that the single-scale Harris codebook from the

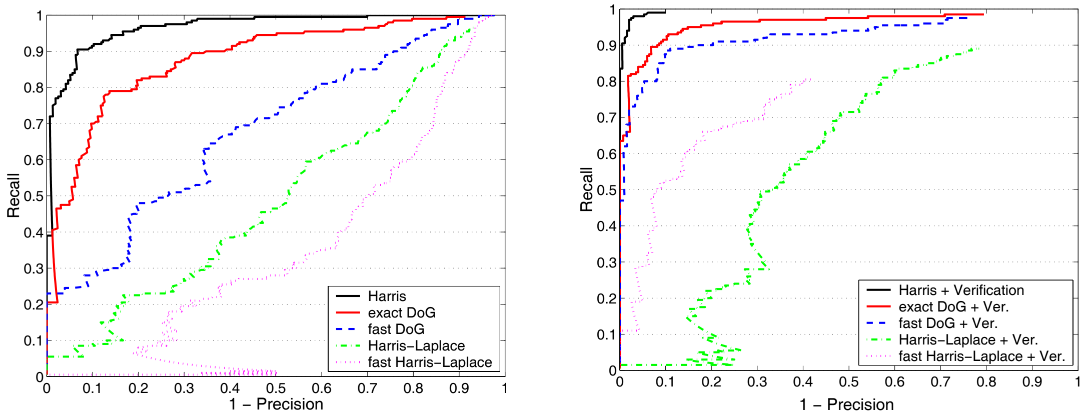


Figure 7.2: *Performance comparison of interest point detectors on the UIUC database. (left) Precision-Recall curves using only the initial patch votes; (right) performance after the hypothesis verification stage.*

previous chapter achieves the best results with 91% equal error rate. Compared to its performance, all multi-scale detectors result in codebooks that are less discriminant. This could be expected, since invariance always comes at the price of reduced discriminance. The exact DoG detector still ranks second with 80.5% EER, but the regular Harris-Laplace and both speed-optimized detectors perform notably worse.

However, all five detectors succeed in finding all cars eventually, which raises hopes that their performance can be improved by adding the hypothesis verification stage. Figure 7.2(right) shows a comparison of the detectors when this is done. Again, the Harris codebook achieves the best performance with 97.5% EER. However, the exact DoG detector reaches an EER performance of 91%, which still compares favorably to state-of-the-art methods (see Fig. 6.3). The fast DoG detector performs only slightly worse with 89% EER. Although the two Harris-Laplace implementations are significantly improved as well (from 48% to 59.5% for the regular, and from 37% to 70% for the speed-optimized version), their performance is still notably inferior.

The main reason for the poorer performance of the Harris-Laplace detectors is that they return a smaller absolute number of interest points on the object, so that a sufficient cover is not always guaranteed. Although previous studies have shown that the Harris-Laplace points are more discriminant individually (Dorko and Schmid, 2003), their smaller number is a strong disadvantage. In the case of the fast Harris-Laplacian, the hypothesis verification stage can still partly compensate for this by considering the consistency of the resulting segmentation, but for the regular Harris-Laplacian, this does not help much.

The DoG detectors, on the other hand, both find enough points on the objects and are discriminant enough to allow reliable matches to the codebook. If only one type of points shall be used, they are thus better suited for our categorization task. For this reason, we only consider DoG detectors in the following experiments.

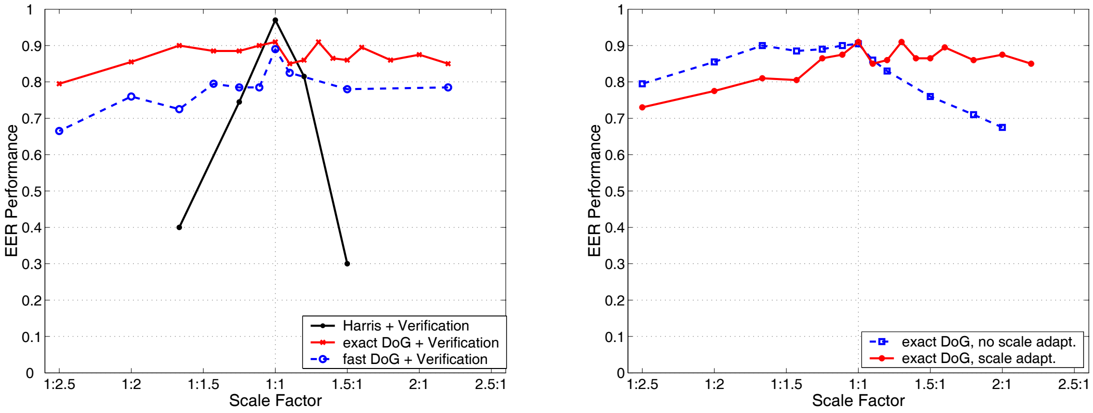


Figure 7.3: *(left): EER performance over scale changes relative to the size of the training examples. While optimal for the single-scale case, the Harris codebook is only robust to small scale changes. The DoG codebook, on the other hand, maintains high performance over a large range of scales. (right): A comparison of the performances with and without the scale-adaption mechanism. As can be seen from the plot, the adapted search window size is necessary for scales greater than* 1.0*, but impedes performance for smaller scales, since a certain minimum search window size needs to be maintained.*

7.3 Robustness to Scale Changes

In the previous section, we have seen that the single-scale Harris codebook performs significantly better than the one constructed with the DoG detector if the test images have the same scale as the training set. We now analyze the robustness to scale changes. In particular, we are interested in the limit to the detectors' performance when the scale of the test images is altered by a large factor and the fraction of familiar image structures is thus decreased. Rather than to test individual thresholds, we therefore compare the maximally achievable performance by looking at how the equal error rates are affected by scale changes.

In the following experiment, the UIUC database images are rescaled to different sizes and the performance is measured as a function of the scaling factor relative to the size of the training examples. Figure 7.3(left) shows the EER performances that can be achieved for scale changes between factor 0.4 (corresponding to a scale reduction of 1:2.5) and factor 2.2. When the training and test images are approximately of the same size, the single-scale Harris codebook is highly discriminant and provides the superior performance described in the previous section. However, the evaluation shows that it is only robust to scale changes up to about 20%, after which its performance quickly drops. The exact-DoG codebook, on the other hand, is not as discriminative and only achieves an EER of 91% for test images of the same scale. However, it is far more robust to scale changes and can compensate for both enlargements and size reductions of more than a factor of 2. Up to a scale factor of 0.6 (corresponding to a scale reduction of 1:1.67), its performance stays above 89%.

Even when the target object is only half the size of those seen during training, it still provides an EER of 85%. For the larger scales, the performance gradation is similar. The fast DoG detector performs about 10% worse, mainly because its implementation with a Gaussian pyramid restricts the number and precision of interest points found at higher scales. Figure 7.3(right) also shows that the system's performance quickly degrades without the scale adaptation step from Section 7.1.1, confirming that this step is indeed important.

An artifact of the interest point detectors can be observed when looking at the performance gradation over scale. Our implementation of the exact DoG detector estimates characteristic scale by computing three discrete levels per scale octave (Lowe, 1999) and interpolates between them using a second-order polynomial. Correspondingly, recognition performance is highest at scale levels where structure sizes can be exactly computed (namely $\{0.6, 1.0, 1.3, 1.6, 2.0\}$, which correspond to powers of $(\sqrt[3]{2})$). In-between those levels, the performance slightly dips. Although this effect can easily be alleviated by using more levels per scale octave, it shows the importance of this design decision.

Figure 7.4 shows a visualization of the range of scales tested in this experiment. Our approach's capability to provide robust performance over this large range of image variations marks a significant improvement over the single-scale version presented in the previous section. Below the car detections, the automatically generated segmentations are displayed for the different scales. As a comparison with the single-scale segmentations in the bottom part of the figure shows, the segmentation quality is only slightly inferior to the quality obtained by the Harris codebook, with noticeable differences only occurring on the fine structures around the windows, which are more accurately localized by the Harris detector. However, the segmentations are stable over a wide range of scales and only degrade slightly for the smallest resolutions, where significantly less information is available from the image.

7.4 Effect of the Training Set Size

Finally, we want to explore the effect of the training set size on detection performance. Up to now, all detectors in this chapter have been trained on the original 50 car images. We now compare their performance when only a subset of those images is considered.

Figure 7.5 shows the resulting performance for different training set sizes from 5 to 50 images. As can be seen from the plot, both the Harris and the DoG codebook reach 90% EER performance already with 20 training examples. When more training images are added, the Harris codebook further improves to the known rate of 97.5%. In contrast, the performance of the DoG detector reaches a saturation point and increases only to 91% for the full training set. Here the advantage of seeing more training images is offset by the increased variance in patch appearance caused by the additional space dimension.

Apart from this evaluation, the figure also compares the performance for the

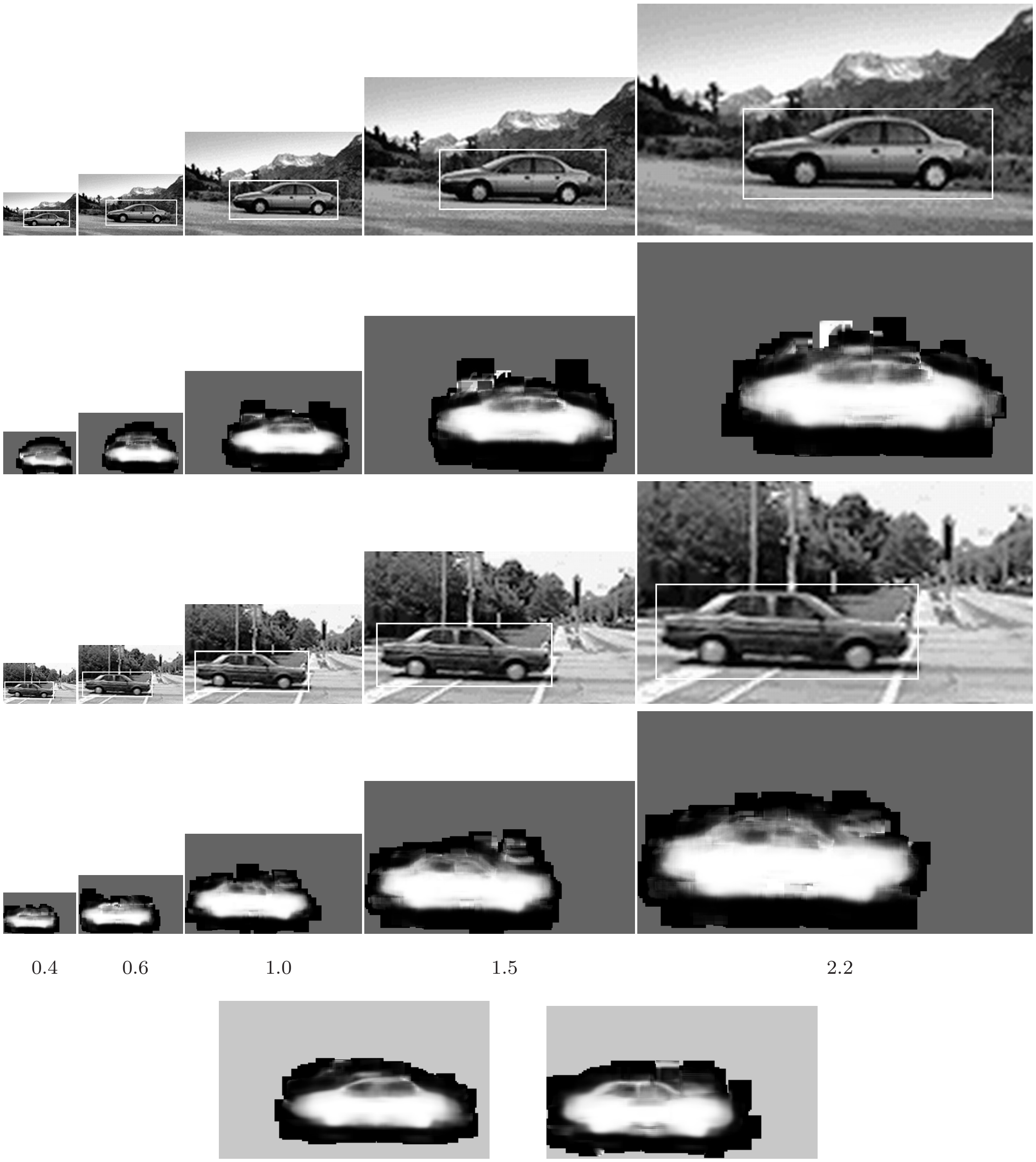


Figure 7.4: *(top) Visualization of the range of scales tested in the experiments, and the corresponding car detections and segmentations. Training has been performed at scale 1.0.; (bottom) For comparison, the segmentation quality for the two cars, as achieved by the Harris codebook.*

original codebooks with the reduced codebooks that are obtained when all single-patch clusters are discarded. It can be observed that the two versions show small differences for the initial voting stage, which however level out when the MDL verification stage is applied. These results confirm our earlier finding that the codebook reduction step does not lead to a decrease in detection performance. Considering

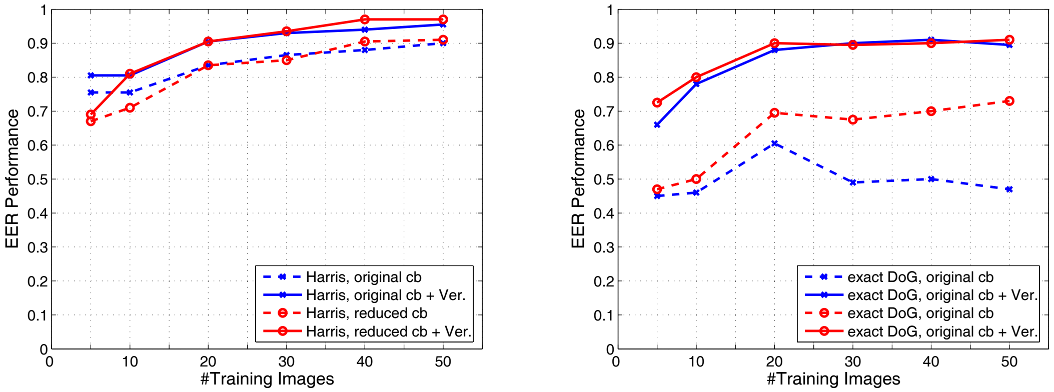


Figure 7.5: *EER performance on the UIUC database for varying training set sizes: (left) for the Harris detector; (right) for the exact DoG detector. The plots show the performance for the original codebooks and for the reduced codebooks when all single-patch clusters are discarded. As can be seen from the plots, both detectors achieve good performance already with 20 training examples. Moreover, the experiment shows that the codebook reduction step does not lead to a decrease in performance.*

that the original codebooks typically contain more than twice as many clusters as the reduced versions, the reduction step can thus be safely advised in order to increase run-time performance.

7.5 Discussion

In this chapter, we have presented a scale invariant extension of the recognition approach that makes the method applicable in practice. By reformulating the multi-scale object detection problem in a Mean-Shift framework, we have obtained a theoretically founded interpretation of the hypothesis search procedure which allows to use a principled scale adaptation mechanism. Our quantitative evaluation over a large range of scales shows that the resulting method is robust to scale changes of more than a factor of 2. In addition, the method retains the capability to provide an automatically derived object segmentation as part of the recognition process.

In order to handle the increased complexity of the higher-dimensional search space, we have proposed an efficient two-step search strategy. The run time of the resulting approach mainly depends on three factors: model complexity (the number of codebook entries and occurrences), image size, and the selected search scale range. Using our current implementation of the car detectors evaluated in this chapter, example run times[1] on two typical test images (with 314×214 and 523×286 pixels) range between 2-3s for our single-scale car detector based on Harris points; 3-6.5s for the multi-scale DoG version with a small scale range of [0.9,1.1];

---

[1] Measured on an AMD Opteron 1.8GHz processor.

and 11-26s for the same detector with a larger scale range of [0.3,1.5] (including detection, segmentation, and MDL verification).

As part of our study, we have also evaluated the suitability of different scale-invariant interest point detectors for the categorization task. One interesting result is that, while found to be more discriminant in previous studies (Mikolajczyk and Schmid, 2001; Dorko and Schmid, 2003), the Harris-Laplacian detector on its own does not detect enough points on the object to enable reliable recognition. The DoG detector, on the other hand, both finds enough points on the object and is discriminant enough to yield good recognition performance. This emphasizes the different characteristics the object categorization task brings with it, compared to the identification of known objects, and the consequent need to reevaluate design decisions.

An obvious extension would be to combine both Harris-Laplace and DoG points in a common system. Since both detectors respond to different image structures, they can complement each other and compensate for missing detections. Consequently, we expect such a combination to be more robust than the individual detectors.

More generally, we can derive the following guidelines for the design of suitable interest point operators from our experimental results:

- The number of interest points returned by the detector plays an important role. While approaches based on stereo matching techniques, such as (Lowe, 1999; Ferrari et al., 2004), can robustly identify known objects already from a small number of matches, this task is much harder for object categorization, where intra-class variability introduces an additional dimension of uncertainty. The interest point operator should thus return enough interest points, so that a sufficient cover of the object is guaranteed.

- Scale interpolation is equally crucial. If the interest point detector estimates characteristic scale on a set of discrete levels (as all detectors in this evaluation did), it is important that the estimated scale be interpolated between adjacent levels. If this is not done, the recognition procedure will inherit the detector's bias for certain scale levels and only produce votes for discrete object scales. In fact, when testing interest point detectors that performed no scale interpolation, we sometimes found it impossible to recognize objects at particular scales, simply because the corresponding region in the voting scale space contained no votes at all.

- Even when scale interpolation is performed, the number of scale levels per octave has a surprisingly strong effect on recognition performance. This effect is acknowledged in (Lowe, 2004) for the case of keypoint matching, but our experimental results show that it is also noticeable on the higher level of object detection. In most applications, the available number of scale levels will be restricted by run-time constraints. Nevertheless, it is important for application designers to be aware of the effect this parameter has on recognition performance.

- Finally, we found that, while speeding up the computation, a Gaussian pyramid implementation of the interest point detector may have a negative effect on the recognition result (as can be observed from the differences between the exact and the speed-optimized detectors' performances in our evaluation). The main reason for this is that the reduced spatial resolution of the upper pyramid levels allows for a poorer localization of detected points. Even if the point locations are accurately interpolated (as advocated in (Lowe, 2004)), the reduced resolution causes less points to be found at larger scales, compared to an exact implementation. Again, the use of a Gaussian pyramid may be dictated by run-time constraints, but it is important to be aware of its effects.

It is important to bear in mind also the restrictions of our evaluation. The UIUC database used for our experiments only contains scenes where all objects have approximately the same scale. While this allowed us to accurately assess the effect of scale changes on the recognition performance, the evaluation is naturally limited as far as the interaction of multiple (potentially overlapping) objects at different scales are concerned. This is especially important as there is a natural tradeoff between the conflicting requirements of accurately detecting objects in small-scale images and excluding false positives in larger images.

In the next chapter, we will therefore evaluate the system on more difficult scenes, containing multiple objects at different scales. In order not to bias the evaluation in favor of a particular object category, we will also apply the system to other categories, such as pedestrians and motorbikes.

8

Application to Other Object Categories

In the previous chapters, we have developed an iterative evidence aggregation scheme which interleaves the processes of object categorization and top-down segmentation. Experiments have shown its capabilities for detecting categorical objects and automatically segmenting them from the background. However, the evaluation has so far been restricted to only two object categories.

In this chapter, we demonstrate the versatility of our approach by applying it to three additional object categories: pedestrians, motorbikes, and rear views of cars. The changed setting allows to verify our previous results also for other scenarios and more difficult scenes. In particular, we evaluate the robustness to scale changes in scenes with multiple overlapping objects at different scales. In addition, the raised difficulty of this task allows a systematic evaluation of several design choices that have not been analyzed in detail before.

The following section applies our approach to pedestrians. Using a sequence of test sets of increasing difficulty, we evaluate its performance for single-scale cases; on scenes containing multiple objects at different scales; and finally on crowded scenes with severe overlaps. Section 8.2 then presents detection and segmentation results for motorbikes, and Section 8.3 shows results for rear views of cars. Finally, Section 8.4 discusses some possible extensions for multi-cue integration and multi-category discrimination.

8.1 Application to Pedestrians

The ability to reliably detect pedestrians in images is interesting for a variety of applications, from video surveillance to automatic driver-assistance systems in vehicles. At the same time, pedestrians are one of the most challenging categories for appearance-based object detection. Since a large percentage of their bodies is covered by different kinds of clothing — which may include a wide range of textures, printed patterns, colors, and color combinations — their appearance may vary considerably, and only few local regions are really characteristic for the whole category. Reliance on global features, on the other hand, is made problematic by the spread of possible articulations and poses, and by a multitude of occluding accessories such as backpacks, briefcases, and hand- or shopping bags, which may perturb a pedestrian's silhouette. Finally, in many applications, several persons may be present in the same image, partially occluding each other and adding to the difficulty.

Previous approaches to pedestrian detection have used either global models, e.g. using full-body appearance (Papageorgiou and Poggio, 2000) or silhouettes (Gavrila, 1998, 2000; Gavrila and Philomin, 1999; Felzenszwalb, 2001); or an assembly of local feature (Viola et al., 2003) or part detectors (Mohan et al., 2001; Mikolajczyk et al., 2004). However, only the last two systems have been demonstrated under partial occlusion, and so far no method has been evaluated for pedestrian detection in crowded scenes with overlaps.

In this section, we apply our approach to the problem of detecting side views of pedestrians. Starting with the single-scale case, we evaluate our method on different test sets of successively increasing difficulty, culminating in an evaluation on crowded scenes with multiple overlapping persons at different scales. In addition, we compare our approach to the Chamfer[1] system (Gavrila, 1998, 2000; Gavrila and Philomin, 1999) and assess the potential for a combination with a global model.

8.1.1 Training Procedure

In order to reduce the training effort, we explore a semi-automatic way for obtaining good training images. We recorded 44 sequences of 35 different people walking parallel to the camera image plane in front of two different backgrounds (see Fig. 8.2(a)). Specific attention was paid to include a wide range of different clothing and accessories, such as backpacks, hand bags, or books. Using the sequences as input, we let the system automatically compute a motion segmentation with the Grimson-Stauffer background model (Stauffer and Grimson, 1999) and manually selected 105 frames for which a good segmentation could be obtained. These 105 images, together with their mirrored versions, served as training set, from which we generated a codebook with 1,024 entries and 89,399 stored occurrences.

Altogether, this procedure resulted in a significant reduction of training effort. Although the obtained segmentation masks are not ideal and still contain some artifacts, their automatic extraction requires far less user intervention than a manual segmentation. Note that the motion segmentation does not have to be perfect for our approach to work — a reasonable segmentation is already sufficient. Moreover, the large amount of data available from video sequences allows us to select only those frames where the Grimson-Stauffer background model produced good results. However, the relatively large number of occurrences already indicates the difficulty of the detection task and the large appearance variations on the pedestrian category.

---

[1] The reimplementation of the Chamfer system and the training data used in this section have been provided by Edgar Seemann.

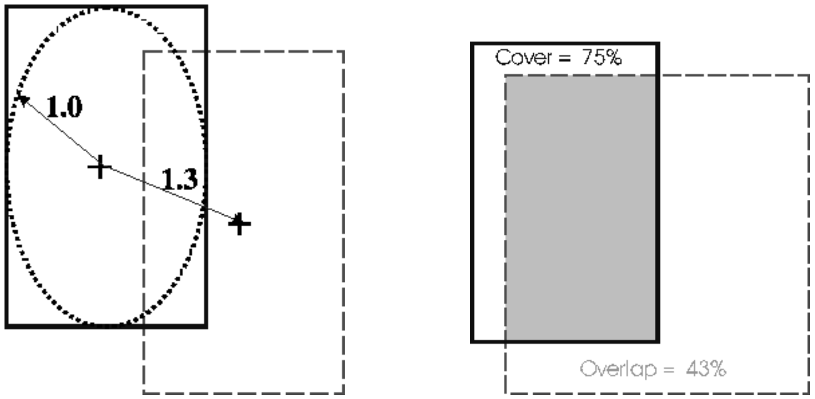


Figure 8.1: *Evaluation criteria for comparing bounding boxes: (left) relative distance; (right) cover and overlap.*

8.1.2 Evaluation Methodology

Previous experiments in this thesis were evaluated according to the scheme by Agarwal and Roth (2002), which is based on absolute distances between bounding boxes. For comparing multi-scale annotation and detection results, this is no longer sufficient. We therefore generalize the evaluation scheme and consider three criteria: *relative distance*, *cover*, and *overlap*. *Relative distance* measures the distance between the bounding box centers in relation to the size of the annotation rectangle (see Fig. 8.1(left)). For this, we inscribe an ellipse in the annotation rectangle and relate the measured distance to the ellipse's radius at the corresponding angle. Points on the ellipse itself have a distance of 1. The relative distance can be computed as follows:

$$d_r = \sqrt{\left(\frac{2 \cdot \Delta x}{w}\right)^2 + \left(\frac{2 \cdot \Delta y}{h}\right)^2} \tag{8.1}$$

where Δx and Δy denote the distances between the bounding box centers in x- and y-direction; and w, h are the width and height of the annotation rectangle. For this evaluation, the annotation rectangle had a fixed aspect ratio of 11:15. *Cover* and *overlap* measure how much of the annotation rectangle is covered by the detection hypothesis, and vice versa (see Fig. 8.1(right)). Together, these criteria allow to compare hypotheses at different scales. In all following experiments, we consider a detection correct if $d_r \leq 0.5$ (corresponding to a deviation up to 25% of the true object size) and *cover* and *overlap* are both above 50%. As before, only one hypothesis per object is accepted as correct – any additional hypothesis on the same object is counted as a false positive.

When analyzing the detection results for unscaled test cases, we observed that the approach often found the target objects correctly, but estimated the object scale either too large or too small, so that the detection was considered "incorrect" by our (rather restrictive) evaluation criterion. The reason for this behavior is that it is very difficult to estimate the true height of a pedestrian from purely local measurements.

One of the most characteristic features for pedestrians appears to be the space just between the two legs during a step. However, as can be seen from Fig. 8.2, a similar feature sometimes also occurs between the knees. In those situations, the person's lower legs can be misinterpreted as the full legs of a smaller pedestrian (and vice versa), with the result that the hypothesized object scale, and thus also its bounding box, are estimated either too small or too large. Despite this uncertainty in the scale estimate, we found that the hypothesized segmentation, obtained as a by-product of the recognition stage, was in most cases still correct. We therefore resolved this problem by computing a "refined" bounding box from the segmentation result, which results in a more robust performance.

In the following, we report the detection results in two different forms: as a *recall-precision* curve (RPC), and as a *recall-false-positive* plot. The latter form has the advantage that it allows to assess also the absolute number of false positives, which is not directly accessible from an RPC diagram.

8.1.3 Single-Scale Detection Results

In this section, we analyze the performance of our approach for the single-scale case. In particular, we compare the performance with and without the verification stage from Section 6.2.2 and the influence of a light normalization step on the recognition results.

For this, we apply the system to two different test sets. Test set 0 is based on the same scenario as the training set (Fig. 8.2(b)). It consists of 197 new images taken from the same sequences already used during training, but containing different articulations and 7 additional, unseen-before people. Each image contains only a single person, and none of them is occluded or only partially visible. In addition, all images are scaled such that all pedestrians have approximately the same height (150 pixels). The purpose of this test set is to verify if enough information has been observed during the training phase to allow for valid generalization.

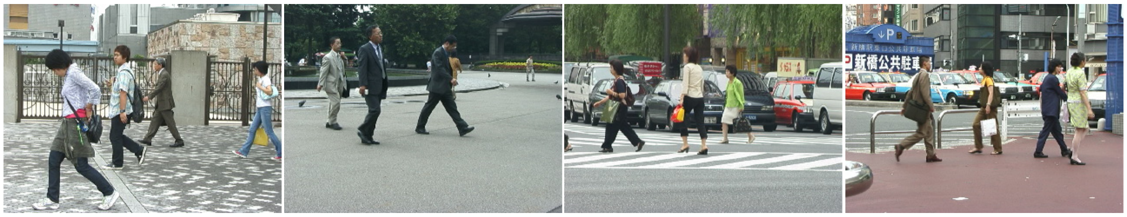
While test set 0 is quite similar to the training set in both content and imaging conditions, test set 1 provides an entirely different scenario. All of its 181 images are taken from a radically different data set containing urban scenes in an Asian metropolis. As a result, the depicted pedestrians, their clothing styles, and the background structures are completely novel (Fig. 8.2(c)). By applying the method to this test set, we can thus verify if the results generalize to a more difficult scenario. Again, each image of test set 1 contains a single, non-occluded pedestrian at a known scale.

In all experiments, we use the multi-scale version of our approach (see Section 7.1) based on scale-invariant interest points obtained by the exact DoG operator. For the single-scale experiments, however, we restrict the scale search to the range $[0.9, 1.1]$.

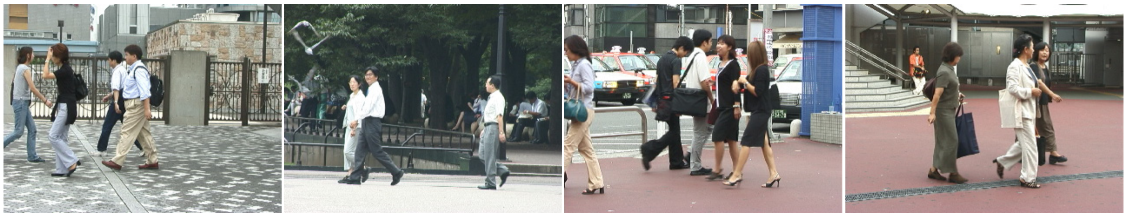
(a) Training set

(b) Test set 0: single-scale, same scenario

(c) Test set 1: single-scale, novel scenario



(d) Test set 2: multiple pedestrians, unknown scales, no overlaps



(e) Test set 3: multiple pedestrians, unknown scales, with overlaps

Figure 8.2: *Example images from the training set and the different test sets for pedestrians used in our evaluation.*

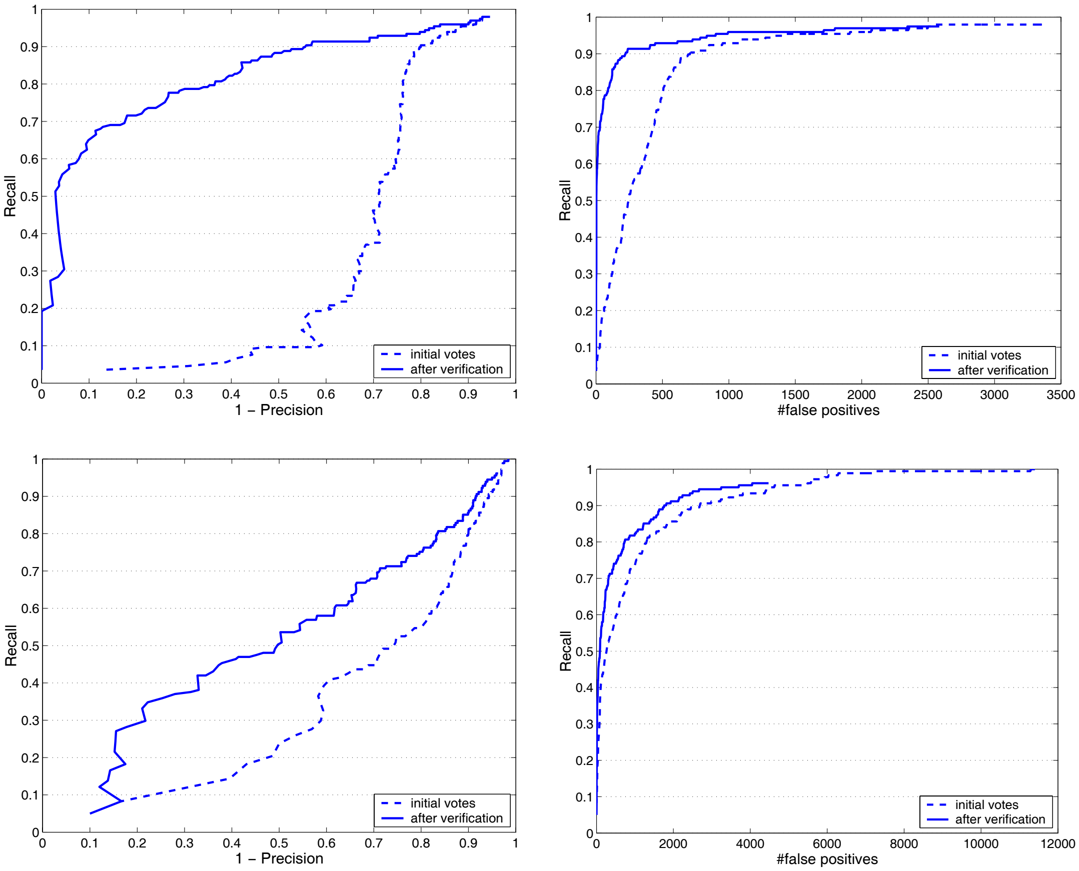


Figure 8.3: *Single-scale pedestrian detection results based only on the initial votes and after the verification stage: (top) for test set 0; (bottom) for test set 1.*

Influence of Hypothesis Verification Stage

Figure 8.3 shows a comparison of the method's performance based only on initial votes and with the additional hypothesis verification stage. It can be seen that the verification stage presents a major improvement. Up to about 65% recall, it returns less than 10% false positives on test set 0. With only the initial votes, the same level of recall can only be achieved at 75% false positives. In absolute numbers, this corresponds to 128 correct detections with 14 compared to 384 false positives. At the 80% recall level, the verification stage achieves a false positive rate of 36.7% (92 false positives for 159 correct detections), while the initial votes result in a nearly unchanged false positive rate of 76.2% (or 510 false positives). For 90% recall, finally, the number of false positives increases to 221 and 707, respectively.

The results for test set 1 are similar. While the absolute performance decreases as a result of the more difficult task, the same qualitative behavior can be observed. As can be seen from Figure 8.3(bottom), the hypothesis verification stage again succeeds to improve the method's performance considerably.

In order to gain a better understanding of the method's performance, we look at the cases where the approach had most difficulties. The missing detections at 90% recall are a good indication for this. Analyzing the images from test set 0, we can make two observations. In 7 of the 19 cases, a hypothesis is correctly centered on the pedestrians, but its bounding box is estimated too large, since the hypothesized segmentation includes spurious responses from background clutter. Those cases are the price we pay for better multi-scale performance later on. Of the 12 remaining problem cases, 9 stem from a single sequence of a girl walking with a long white tube (shown in the rightmost image of Fig. 8.2(b)). Although the tube occludes only a small portion of the girl's body, its elongated shape affects many local measurements sufficiently to distract the recognition algorithm.

The larger number of false positives in this evaluation confirms the raised difficulty of the task, compared to the detection of rigid objects such as cars, as described in Section 6.3. However, tracing the performance curves to their end, the method succeeds to find all but a handful of the test objects eventually. This raises hopes that its performance can be improved by another verification stage, possibly as a combination with global cues. This possibility will be analyzed in more detail in Section 8.1.4.

Influence of Lighting Normalization

A basic restriction of local approaches is that, since all later stages build on the results of the initial feature matching stage, the total performance can be only as good as the matched features permit. In that respect, a potential problem is that the interest point extraction procedure is inherently sensitive to image contrast. In low-contrast images, typically a smaller number of interest points are found, and the detections become less reliable. In order to compensate for this effect, some object detection systems apply a lighting normalization step prior to the feature matching stage (Schneiderman and Kanade, 2000; Agarwal and Roth, 2002). In previous experiments such as the evaluation on cars presented in Section 6.3, we had obtained good results with *histogram equalization*. In the next experiment, we therefore want to evaluate the effect this choice of lighting normalization has on the recognition results.

Figure 8.4 shows a comparison of the method's performance with and without histogram equalization. As can be seen from the plot, the two curves are very similar for both test sets. Which version is better varies with the desired precision. A more detailed analysis shows, however, that the two options indeed lead to a qualitatively different behavior on some images. In particular, the histogram equalization procedure leads to a slightly better result (and more accurate segmentations) on most images, but it fails completely in some other cases. This can be observed especially for some images with strong contrast differences, where the lighting normalization leads to an overly large number of false positives, while the true detections get very low scores. Because of these effects, we do not pursue the histogram equalization option further for the remaining experiments.

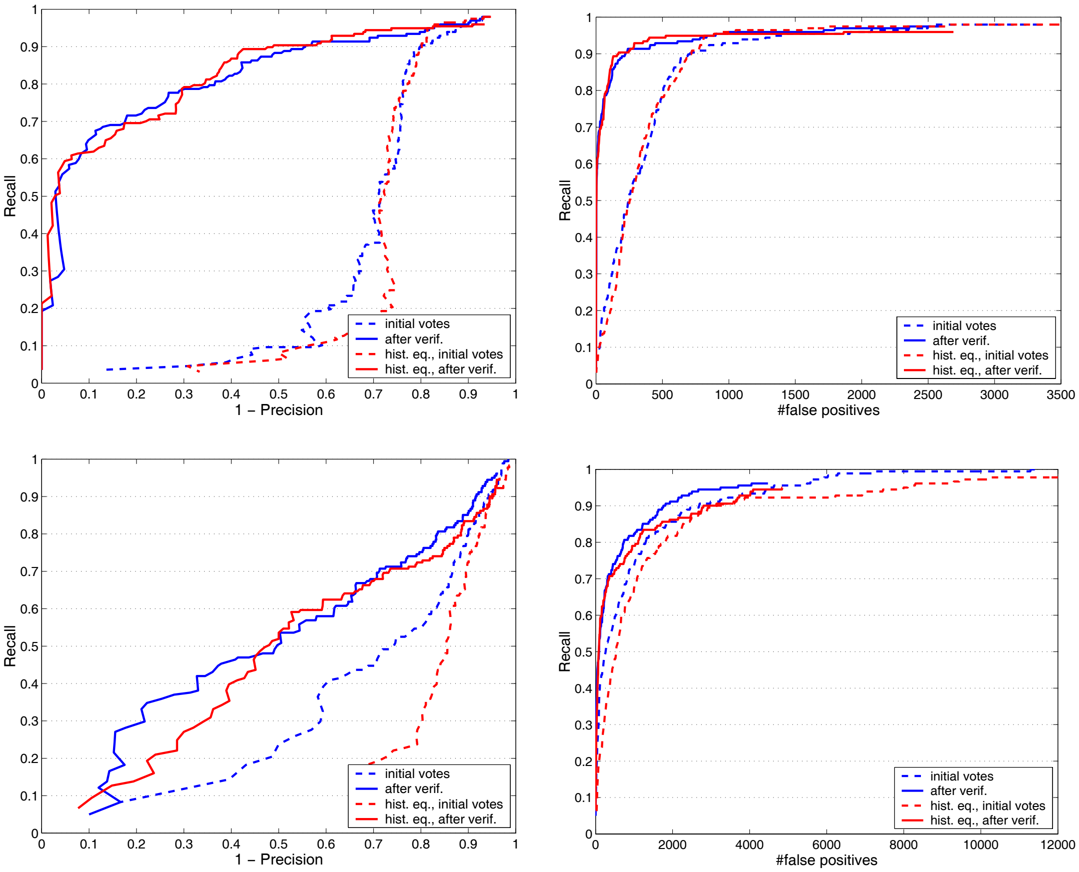


Figure 8.4: *Single-scale pedestrian detection results with and without performing histogram equalization: (top) for test set 0; (bottom) for test set 1.*

Influence of Optimized MSME Kernel Size

So far, we have applied the method with the same basic parameter settings as for the car experiments in Section 6.3. In particular, the size of the MSME search window used for finding local maxima in the continuous voting space was left at a fixed value optimized for cars. Since the search window dimensions should reflect the proportions of the object of interest, this setting is not necessarily optimal. As the next step, we therefore examine the impact an adapted search window, optimized for pedestrians, has on the recognition results. In an extensive test series, we varied the search window size in all three dimensions (x, y, $scale$) independently and determined the optimal kernel dimensions.

Figure 8.5 shows a comparison of the pedestrian-optimized version to the unadapted system. As can be seen from the figure, the adaptation yields a significant performance improvement (of up to 20% precision at 53% recall for test set 1, corresponding to a reduction of the number of false positives from 97 to 41). As a small drawback, the adapted version does not reach 100% recall anymore. Some detections are lost entirely as a result of the tuning process. Because of the ob-

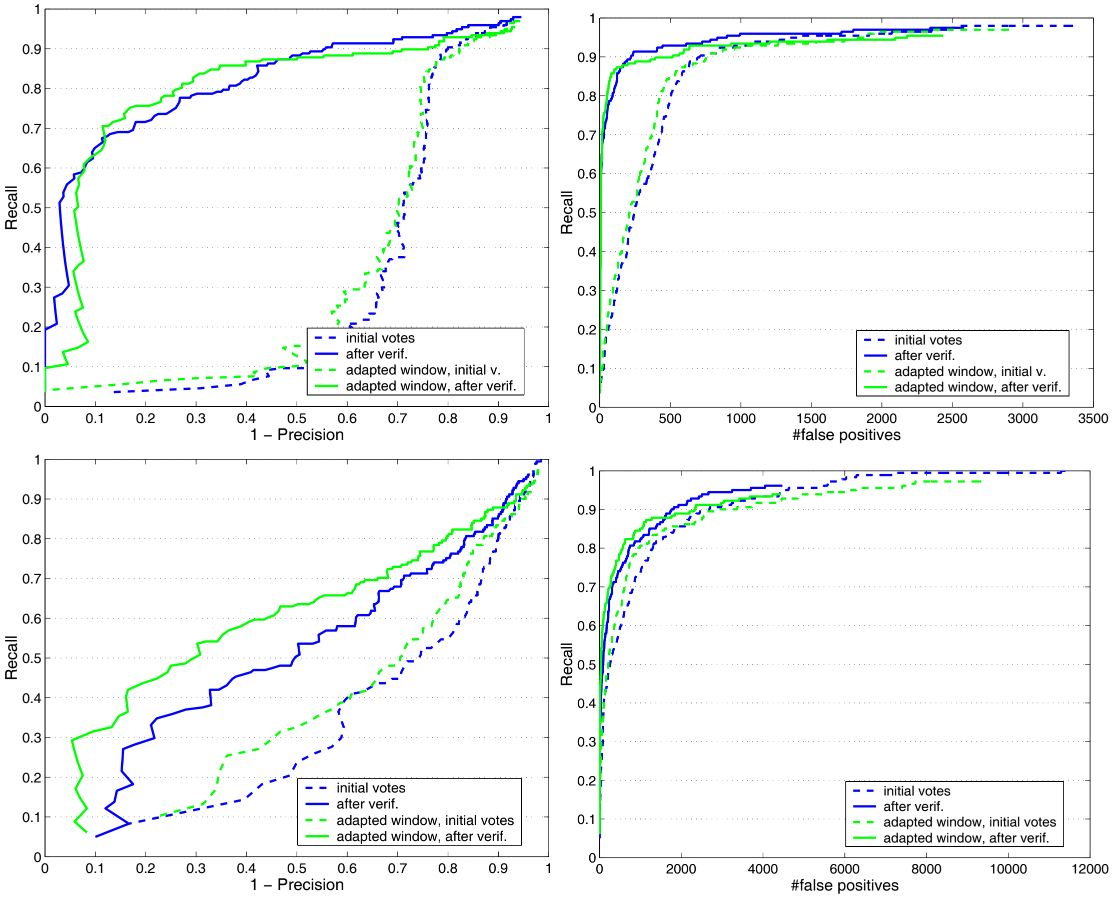


Figure 8.5: *Single-scale pedestrian detection results. Effect of adapting the MSME search window size to pedestrians: (top) for test set 0; (bottom) for test set 1.*

served increase in overall performance, however, we will use the adapted version for all further experiments.

Comparison to the Single-Scale Version with Harris Interest Points

Finally, we want to compare the performance of the scale-invariant approach to our previous single-scale approach based on Harris interest points. Figure 8.6 shows the results of this experiment. Trained on the same image set, the Harris codebook achieves a superior performance, both for the initial votes and after the verification stage. It particular, it manages to reduce the number of false positives significantly. On test set 0, it reaches an EER performance of 82.2%, compared to 75.6% for the scale-invariant version. On test set 1, the improvement is even larger with 71.3% compared to 56.8%.

These results again confirm our previous observation that scale-invariant interest points result in a less discriminant codebook compared to Harris points. The difference can be seen in particular for test set 1, where stronger generalization capabilities

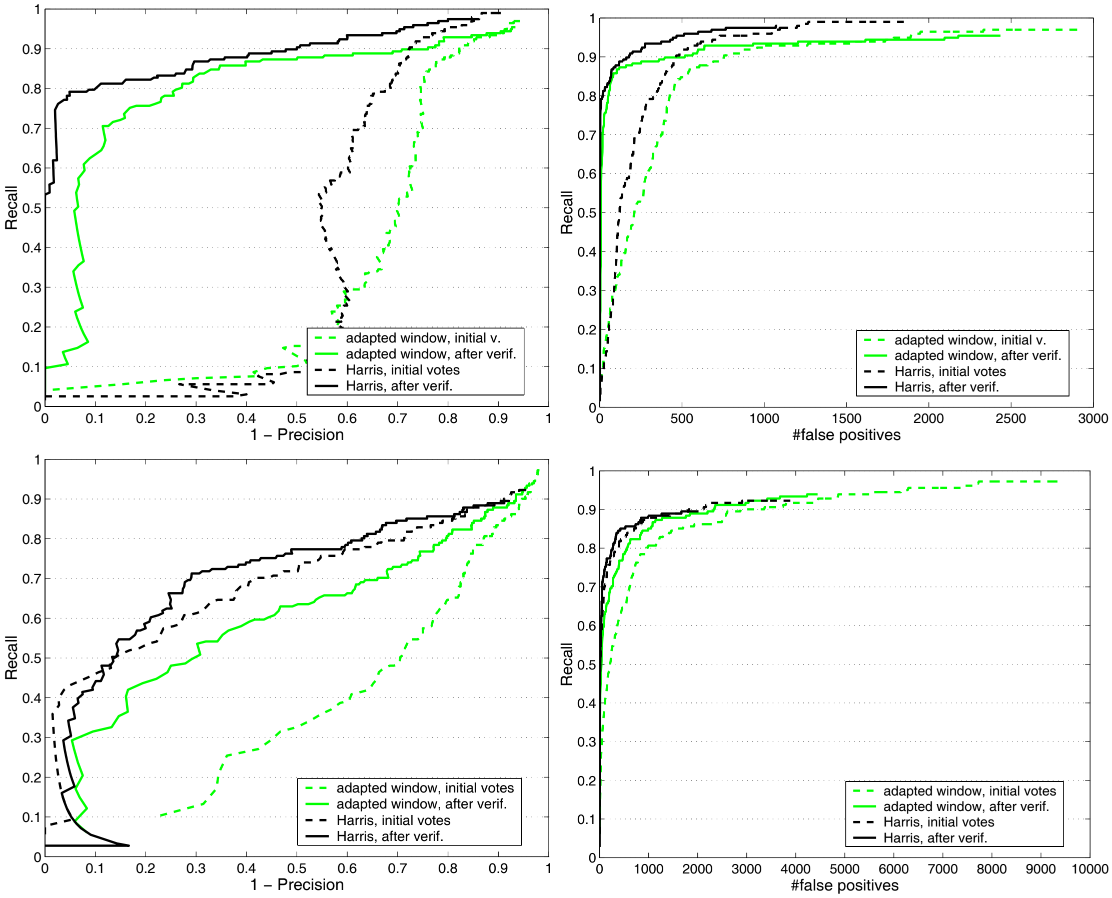


Figure 8.6: *Comparison of the scale-invariant version of our approach (using the exact DoG detector) with the single-scale version (using Harris interest points): (top) on test set 0; (bottom) on test set 1.*

are required. For some applications, it might thus be advantageous to combine the single- and multi-scaled approaches as different stages in a cascade and thus pool their advantages. In such a system, the multi-scale detector would first be applied as a "scale probe" in order to find promising hypotheses, whereupon the relevant image regions are rescaled, so that the more discriminant fixed-scale detector can be used to verify and rank them.

8.1.4 Comparison with Chamfer Matching

As an example of an existing pedestrian detection system, we compare our method to a reimplementation of the Chamfer matching approach (Gavrila, 1998, 2000; Gavrila and Philomin, 1999). The goal of this comparison is not to benchmark performances, but to analyze in which situations which approach is better-suited and whether a combination of the two might be useful.

Chamfer matching tries to detect objects by relying on global shape features.

Given a set of trained shape templates, it searches the image for locations where these templates can best be matched to the image content. In most cases, the templates contain object silhouettes. However, the approach is not restricted to this choice but can use any feature that can be converted into a binary *present/absent* map. Object shapes are compared using a distance transform, which converts a binary feature map into an image where each pixel value denotes the distance to the nearest feature pixel. Matches of a template T to the distance-transformed image I are found by shifting the template over the image and computing, at each location, the average distance value of all pixels that are covered by the template

$$D_{Chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t) \tag{8.2}$$

The advantage of matching a template with the distance-transformed image rather than with the original edge image is that the resulting similarity measure will be smoother as a function of the template transformation parameters (Gavrila, 2000). This makes it possible to speed up the matching process by employing a hierarchical coarse-to-fine search. Moreover, both the Chamfer transform and the later maxima search can be efficiently computed using only integer operations, which makes the method attractive for real-time applications.

In this section, we use our own reimplementation of the Chamfer matching approach and compare its performance on the same test sets as used for our approach. As basis for our experiments, we use a training set of 210 pedestrian silhouettes (plus their mirrored versions) extracted from the same video sequences as described above. However, in order to ensure clean training data, the segmentations have been manually edited. All training silhouettes, as well as the test images, are rescaled such that the pedestrians have a uniform height of 100 pixels.

In addition to the regular version using intensity edges, we also test a modified version based on *color edges*. Instead of applying a Canny edge detector on the gradient magnitude of the intensity image, this version first transforms the image into an $LC_\alpha C_\beta$ color space and takes the magnitude of the three-dimensional gradient vector. As a result, it can compensate for situations with low intensity contrast, which makes it slightly more robust.

The original implementation by Gavrila (1998, 2000) and Gavrila and Philomin (1999) contains a number of other improvements to the method, such as multi-feature matching using several edge orientation planes, the use of a template hierarchy for efficient matching with a large number of exemplars, multi-stage edge segmentation thresholds, and a hypothesis verification stage based on RBF neural networks. It is thus important to note that our reimplementation does most likely not reach the performance of their system. However, we are confident that it allows to gain some insights into the method's behavior and draw conclusions about its potential for a combination with our system.

Figure 8.7 shows a comparison of the two approaches on the single-scale test sets. It can be observed that the Chamfer matching approach manages to find all

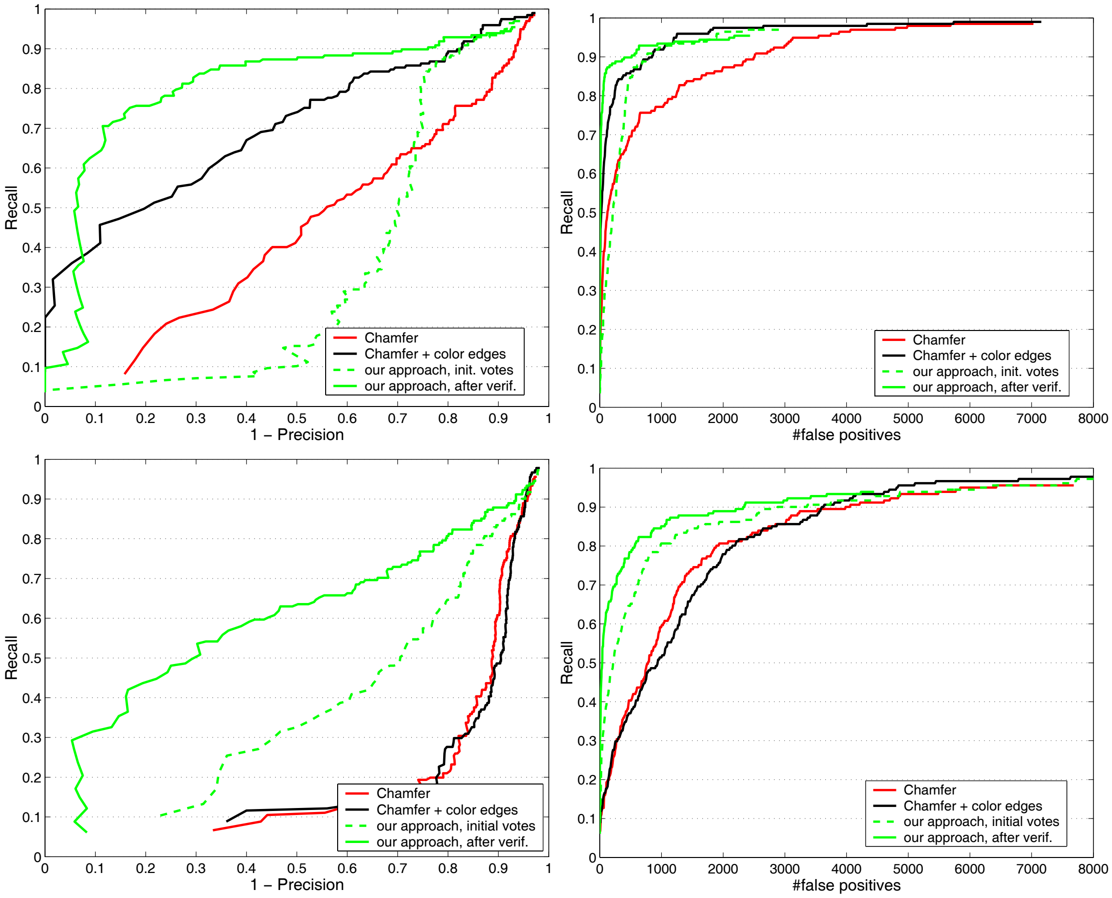


Figure 8.7: *Comparison of our approach to the Chamfer system: (top) on test set 0; (bottom) on test set 1.*

pedestrians eventually, but it returns a relatively large number of false positives. On test set 0, the regular version based on intensity edges achieves an EER performance of 48%. The color-edge based version performs better with 63% EER, but still does not reach the 75.6% of our scale-invariant approach. At the 80% recall level (159 out of 197 correct detections), this performance translates to 238 and 1286 false positives, respectively (compared to 59 for our approach). This is consistent with Gavrila's findings who also reported "a handful of false detection solutions per image" for detection rates in the 60–90% range using the Chamfer System alone (Gavrila, 2000).

The differences are even more pronounced on test set 1. Here, the Chamfer approach surpasses the 1000-false-positives mark already at the 60% recall level, while our Implicit Shape Models can reach this performance with only 86 false positives. It should be said, however, that test set 1 presents a very difficult task for the Chamfer system. The test images contain a large number of edge features, which lead to many high-scoring matches on background structures. In addition, the pedestrians' appearances in test set 1 differ considerably from the training examples.

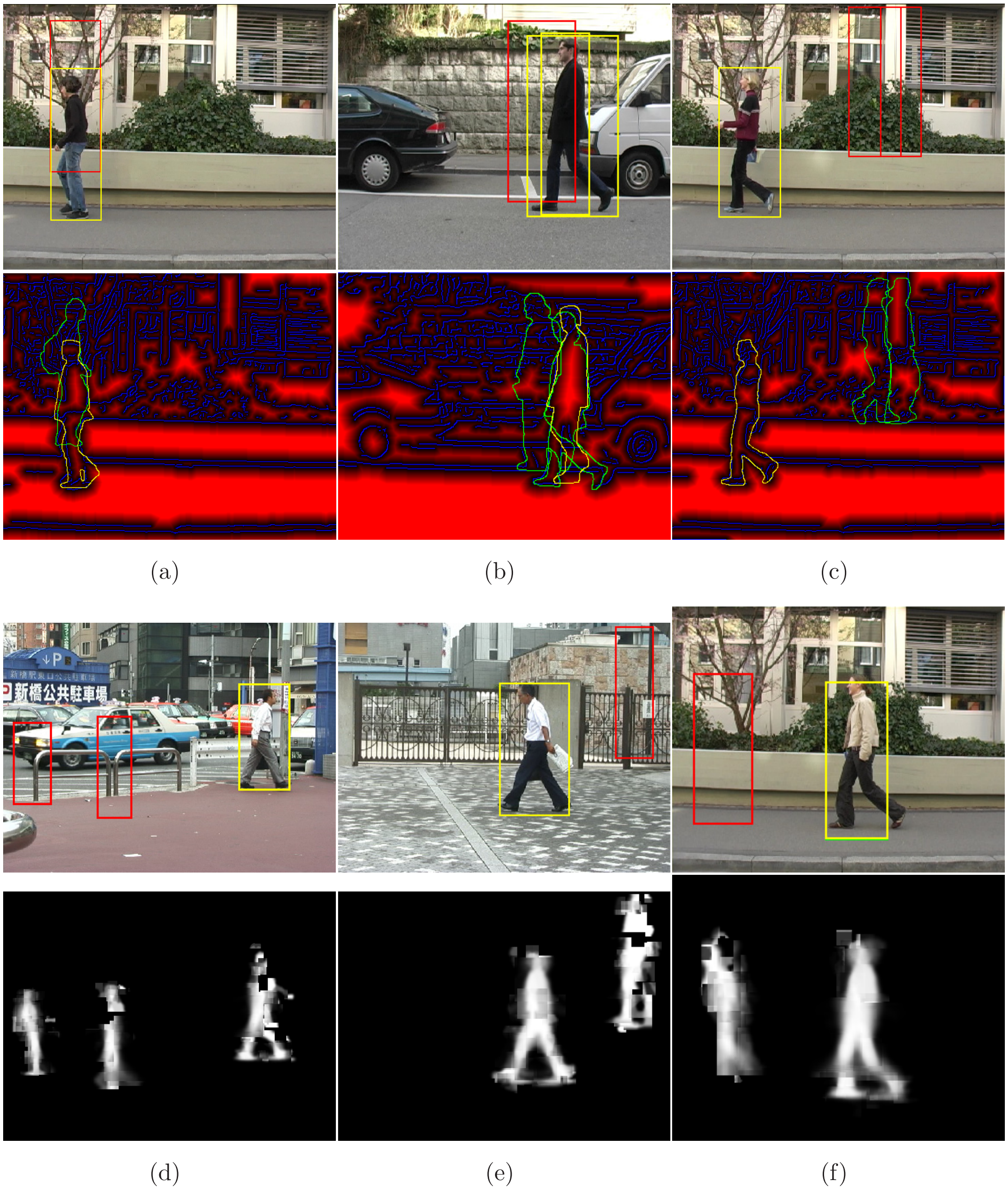


Figure 8.8: *Main reasons for false positives (top) of the Chamfer system; (bottom) of our approach: (a) confusion of upper body with legs; (b) confusion of front and rear edge of silhouette; (c) matches on highly-textured backgrounds; (d,e) thin vertical structures; (f) regions with strong intensity contrasts.*

For instance, the training set contains no examples of running people or of women wearing skirts, which are characterized by different silhouettes. It is naturally harder for a global approach to adjust to this change, and the relatively small number of 210 training examples is clearly not sufficient to compensate for that.

Instead of looking at the raw performance figures, it is thus more useful to compare the two approaches' problem cases qualitatively. Figure 8.8 shows the most frequent causes of false positives for both approaches. For Chamfer matching, the main reasons for spurious detections are (a) confusions of a person's upper body with its legs; (b) confusions of front and rear edge of the silhouette; and (c) matches on highly-textured backgrounds where many edge pixels are found[2]. For our approach, false positives are mainly caused by (d,e) thin vertical structures, such as lamp- or fence posts; and (f) regions with strong intensity contrasts that give rise to many interest points.

In order to understand why thin vertical structures are a problem, consider the corresponding segmentations. As Fig. 8.8(d) shows, the fence posts in this image induce many local matches, which are hypothesized to correspond to a person's front or rear edges. The resulting segmentations are far too slim for a pedestrian, but since only local consistency with a common object center is enforced, the assembly is still taken as a valid hypothesis. Such cases could be rejected by a combination with a global model.

Comparing the missing detections on test set 0, we found that the main reason for missing detections by the Chamfer system were difficulties with the edge extraction stage on dark or low-contrast image regions. 14 out of 20 cases at the 90% recall level could be traced back to this source. The remaining causes were unusual articulations and large occluding objects, such as the guitar case shown in Fig. 8.2(b) (interestingly, this object caused no problems for our approach). Altogether, the overlap between the two methods was only 1 out of the 19 cases at 90% recall, and still only 13 out of 40 cases at 80% recall. This confirms that the two methods are indeed complementary.

8.1.5 Multi-Scale Detection Results

As the next step, we analyze our approach's performance for the multi-scale case. In all following experiments, we apply the method with a scale search range of [0.3,1.5] (corresponding to a search for pedestrians with a height between 45 and 225 pixels).

Scenes Containing a Single Object

For comparison, we first show the performance on test set 1 when the objects occur at their original scales. Figure 8.9 relates the resulting performance to the single-scale case. As can be observed, the method shows the same qualitative performance gradation, but the increased difficulty of this task results in a larger number of false positives.

---

[2] From Gavrila's results, we also know that some rectangular background structures, such as thin columns or windows, cause problems, since a rectangle has on average only a small distance to some pedestrian silhouettes. However, the test images used in our evaluation contain only few such structures.

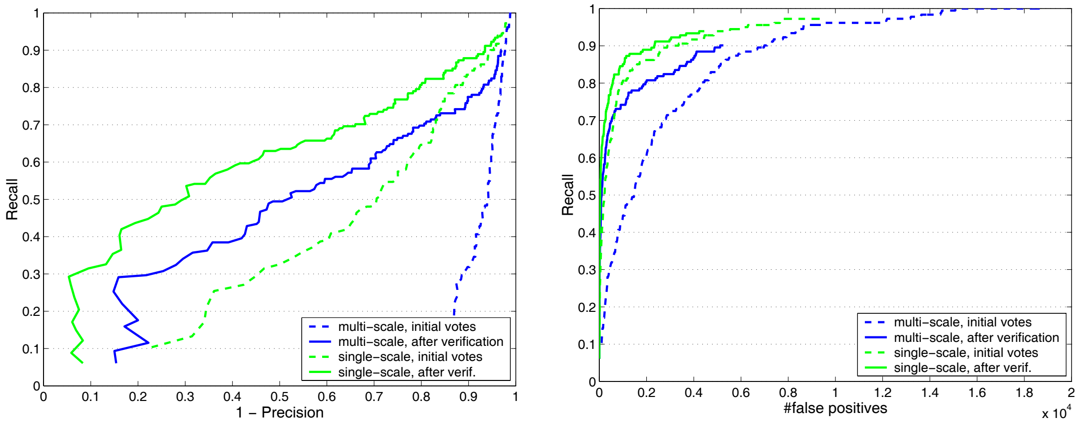


Figure 8.9: *Comparison of the single- and multi-scale results on test set 1.*

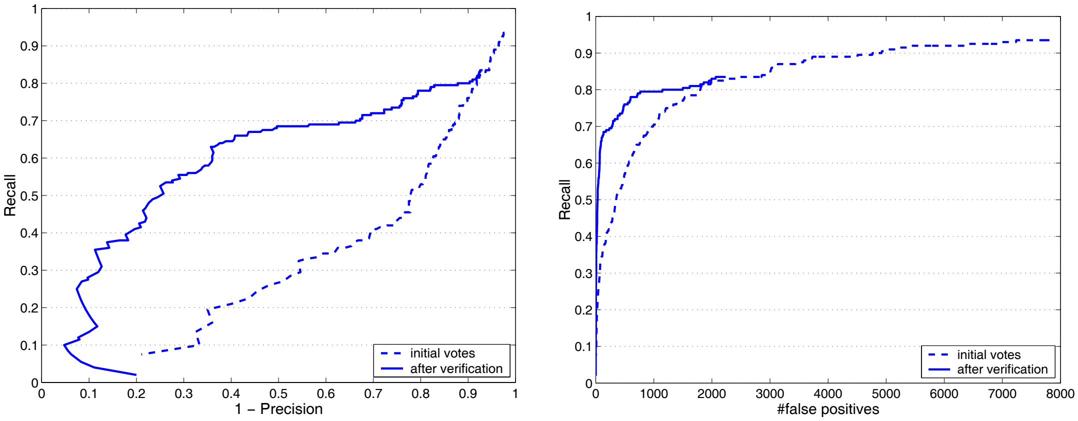


Figure 8.10: *Multi-scale pedestrian detection results of our approach on test set 2.*

Scenes with Multiple Objects

Next, we introduce another level of difficulty by applying the system to a real multi-scale task with scenes containing multiple persons. For this, we use a new test set (in the following called test set 2, see Fig. 8.2(d)) with images taken from the same source as for test set 1. It consists of 75 images containing a total of 200 pedestrians at different, unknown scales. All pedestrians are completely visible in the images, and there are no overlaps between them.

Figure 8.10 presents the results for test set 2. The method scales to this task and achieves an EER performance of 64%. Some example detection results (obtained at the EER) can be seen in Figure 8.11. They show that the method can successfully recognize multiple objects at different scales in the image. Indeed, some of the false positives in those examples correspond to additional correct detections where only the object scale has been estimated incorrectly.

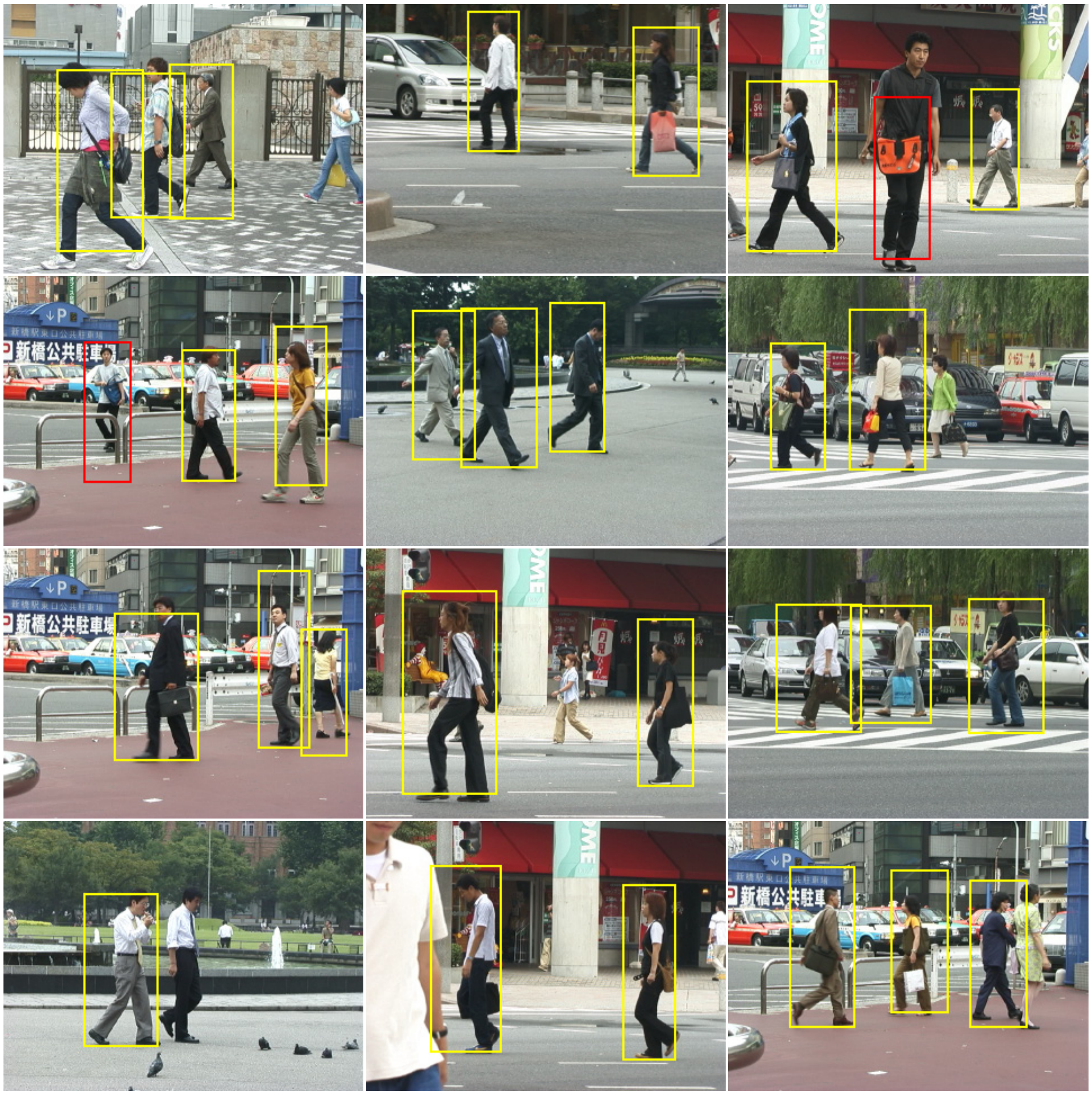


Figure 8.11: *Example multi-scale detections of our approach on test set 2 (at the EER).*

Looking just at the equal error rates, the method's performance on this test set may seem better than its performance on the simpler task in test set 1 (with an EER performance of 64%, compared to 59%). However, the experiments are not directly comparable because of the different image selection. Another factor is that, since more true pedestrians are now in the image, distracting background structures have less effect on the recognition rate. In particular, some problematic background structures that often led to false positives, are less often visible.

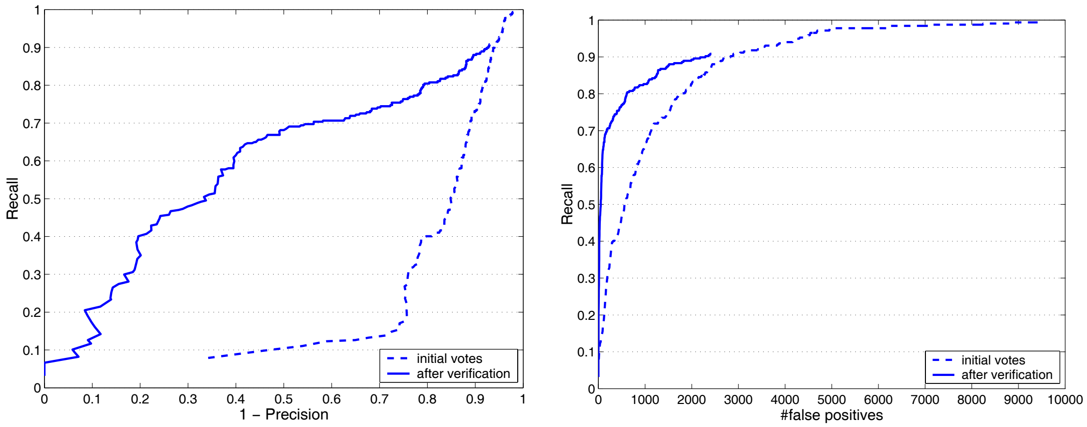


Figure 8.12: *Multi-scale pedestrian detection results of our approach on test set 3 (with overlaps).*

8.1.6 Performance with Severe Overlaps

Finally, we present results on test set 3. This test set constitutes the hardest task in our evaluation in that it contains crowded scenes with multiple overlapping pedestrians. The overlaps present a major obstacle for every global approach, since they have the effect that for many objects, no global contour is present in the image. In contrast, the local nature of our approach makes it still applicable to this scenario. All in all, the test set consists of 131 images with a total of 317 annotated[3] pedestrians (see Fig. 8.2(e)).

Figure 8.12 shows the method's performance on this test set. As can be observed, our approach scales to the difficult task and achieves a respectable EER performance of 60.5%. The quantitative evaluation should be taken with a grain of salt, however, as it is often a matter of interpretation whether a certain partially occluded person should be annotated as "object" or not. As a result, some of the annotated objects are so hard to find that they have not been detected by the algorithm at all, while on the other hand some of the false positives are in reality true detections of objects that had just not been annotated.

In order to get a feeling for the method's capabilities, it is therefore more informative to look at the qualitative results. Figure 8.13 shows some successful detections obtained at the equal error rate. As can be seen, the method is able to find pedestrians even when they are partially occluded or walking in small groups. Even though typically not every single person in such groups is found, the approach manages to detect the subset that was least occluded. Sorting out which limb belongs to which person and thereupon inferring the presence of another object requires a more global interpretation of the image that our local approach with its implicit shape model cannot yield yet.

---

[3] In order not to bias the evaluation, we annotated only pedestrians that were visible in side views and that were fully contained in the image (no boundary occlusion).

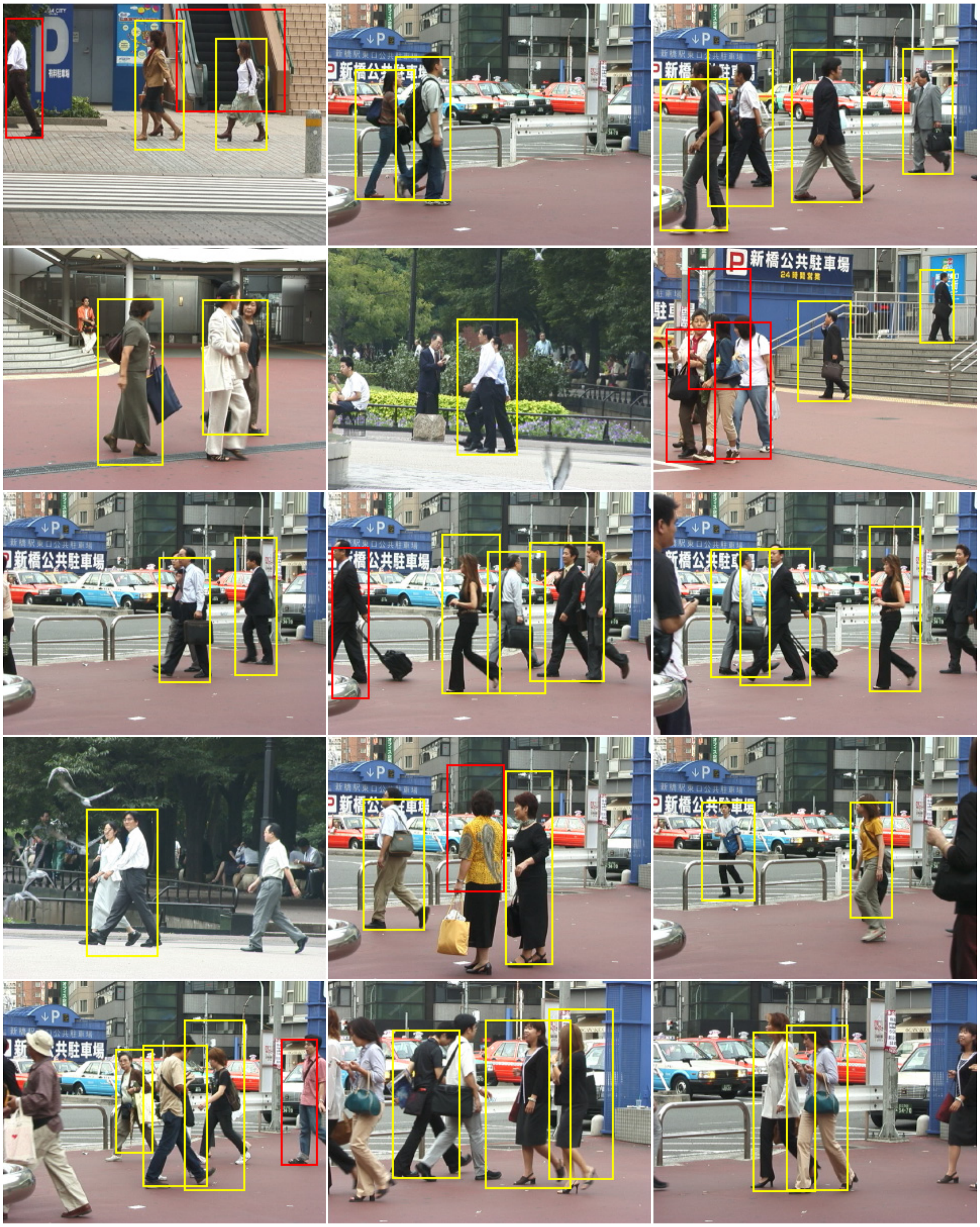


Figure 8.13: *Example multi-scale detections of our approach on scenes with overlaps from test set 3 (at the EER).*

8.1.7 Discussion

In summary, we have shown our method's applicability to the task of pedestrian detection. This task is more difficult compared to the object categories tested in our previous experiments. As a result of their more diverse appearance, only few local regions are really discriminative for pedestrians; in addition, their shapes contain more variation through different articulations and poses.

In the course of our experiments, we have evaluated our approach on different test sets of increasing difficulty. While the images of the initial test set 0 were taken from the same scenario as the training images, the observed results could be reproduced also on several test sets of a different and more difficult scenario. The results show that our local approach can detect pedestrians in novel settings and at different scales, even when they overlap and partially occlude each other.

The raised difficulty of the task allowed to evaluate the effects of several parameters more closely. As an example, we have measured the influence of a lighting normalization step and quantified the performance improvement that can be obtained by adapting the system's parameter settings to pedestrians. In addition, we have compared the single-scale detection performance to Chamfer matching and evaluated the potential for combination. Our results show that the two approaches are indeed complementary in their false positives and missing detections, so that they could profit from each other.

Another promising extension would be to combine the single- and multi-scale approaches as different stages in a cascade. With the multi-scale detector as an initial scale probe, the more discriminative single-scale detector could be applied to a rescaled version of the image for verification.

8.2 Application to Motorbikes

As a second category, we apply our system to side views of motorbikes. For training, we use 153 segmented[4] images of motorbikes from the CalTech database (a subset of the 400 images Fergus et al. (2003) used for training). By clustering the 19,241 patches extracted from this data set, we generated a codebook with 1,869 entries and 94,947 stored occurrences.

In order to compare our approach's performance to results from the literature, we first evaluate it on an *object present/absent* decision task after the scheme described in (Fergus et al., 2003). The test set consists of 400 images containing one motorbike each at an unknown scale and 450 images from a background data set[5]. In order to decide whether or not a test image contains a motorbike, we apply our scale-invariant detector with a scale search range of [0.5,1.5] and accept an image if at least one detection can be found.

Table 8.1 shows the results of this experiment. With an EER performance of 94%, our approach compares favorably to other results reported in the literature (Please note that some of the performance figures shown in Tab. 8.1 are single-scale results, whereas our approach has been evaluated on a multi-scale task). However,

---

[4] The criterion for selecting those 153 images was that they had a roughly uniform background, which made them easy to segment using just the Flood Fill function of standard graphics software. No experiments were undertaken to determine the necessary minimum number of training examples.

[5] The same images were used as in (Fergus et al., 2003).

Method	EER Performance
Weber (2000)	88%
Fergus et al. (2003)	93.3%
Opelt et al. (2004)	92.2%
Thureson and Carlsson (2004)	93.2%
Our algorithm	94.0%

Table 8.1: *Comparison of our results on the CalTech motorbike data set with others reported in the literature. The table shows the EER performances for an object present/absent decision task.*

it is important to point out the differences to a real detection task. If the goal is object detection, the experiment delivers an overly optimistic performance estimate. In order to decide whether an object is present somewhere in the scene, a detection does not need to be too accurate. Moreover, many of the positive test images contain just the target object with little or no background structure, so that localization becomes easy.

In order to demonstrate our approach's performance, we therefore apply it to a more challenging task of detecting objects in an own database of 115 motorbike images collected from the World Wide Web. Each image contains one or more motorbikes at unknown scales and in front of difficult backgrounds. Some images depict larger scenes in which the motorbikes must be localized; others add to the difficulty by containing occluding elements, such as humans sitting on or posing in front of the motorbike.

We present our results on this test set in three stages. First, Figure 8.14 shows example detections on relatively simple images that demonstrate the appearance variability spanned by the motorbike category and the segmentation quality that can be achieved by our approach. As these results show, our method manages to adapt to different appearances and deliver accurate segmentations.

However, the images still show a single object in front of a relatively uniform background. This is different in the next setting. Figure 8.15 presents examples of successful detections in scenes with difficult backgrounds and under partial occlusion. As can be seen from those examples, the method still achieves reliable detection results despite these effects, even though the additional difficulties naturally affect the segmentation quality. Due to the larger appearance variability of motorbikes, however, it is in general not possible anymore to segment out the occluding structure (as was the case for the car category in Chapter 6).

Finally, Figure 8.16 presents detection and segmentation results for larger scenes where the motorbikes take up only a small part of the image. The depicted results show that the method also scales to this task. In addition, the examples demonstrate that the flexible evidence combination scheme can compensate for a certain degree of in-plane and out-of-plane rotation, even though these effects are not explicitly modelled in our system.

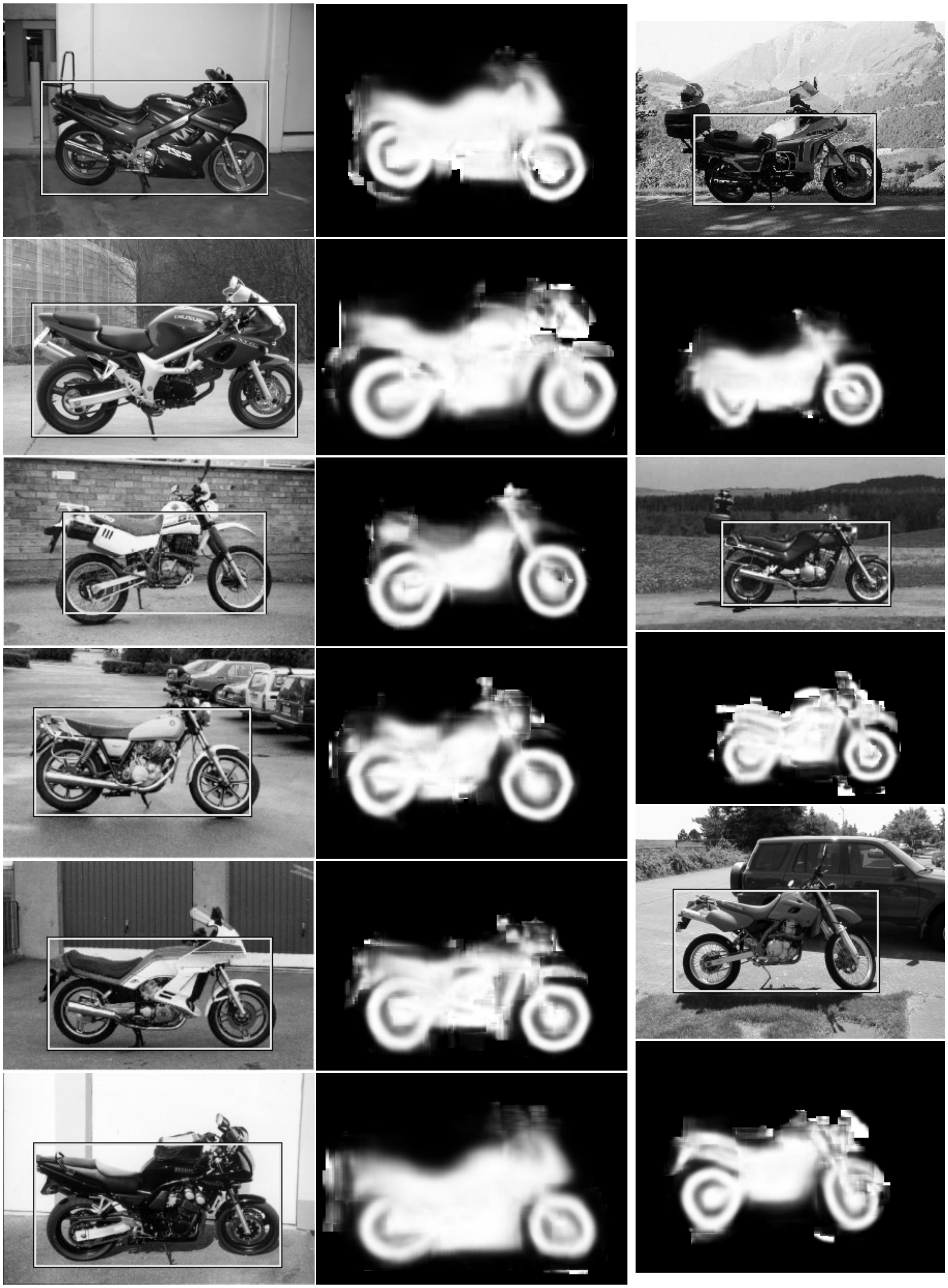


Figure 8.14: *Examples for the variety of motorbike shapes and appearances that are still reliably detected and segmented.*

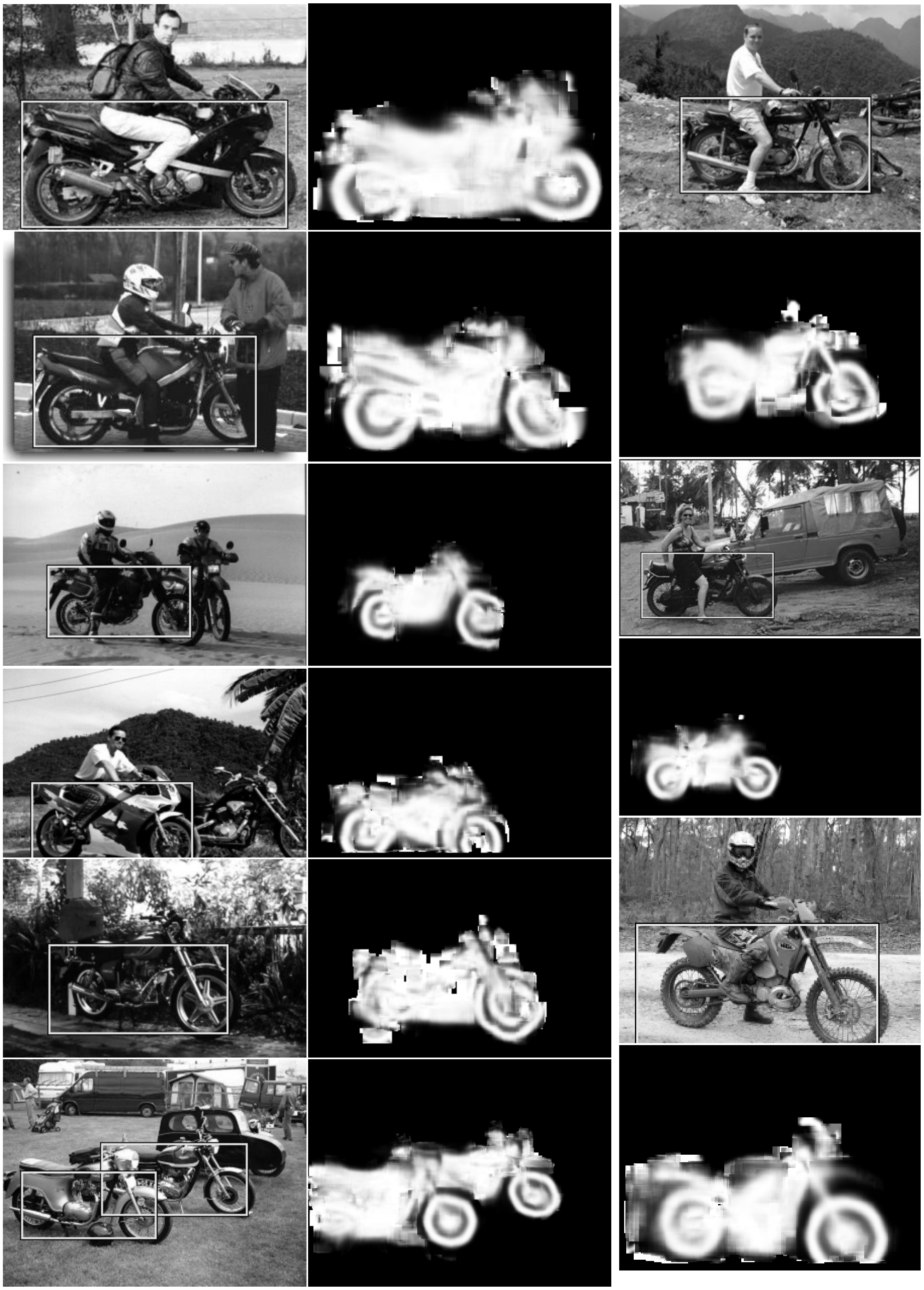


Figure 8.15: *Example detection and segmentation results for motorbike images with partial occlusion and difficult backgrounds.*

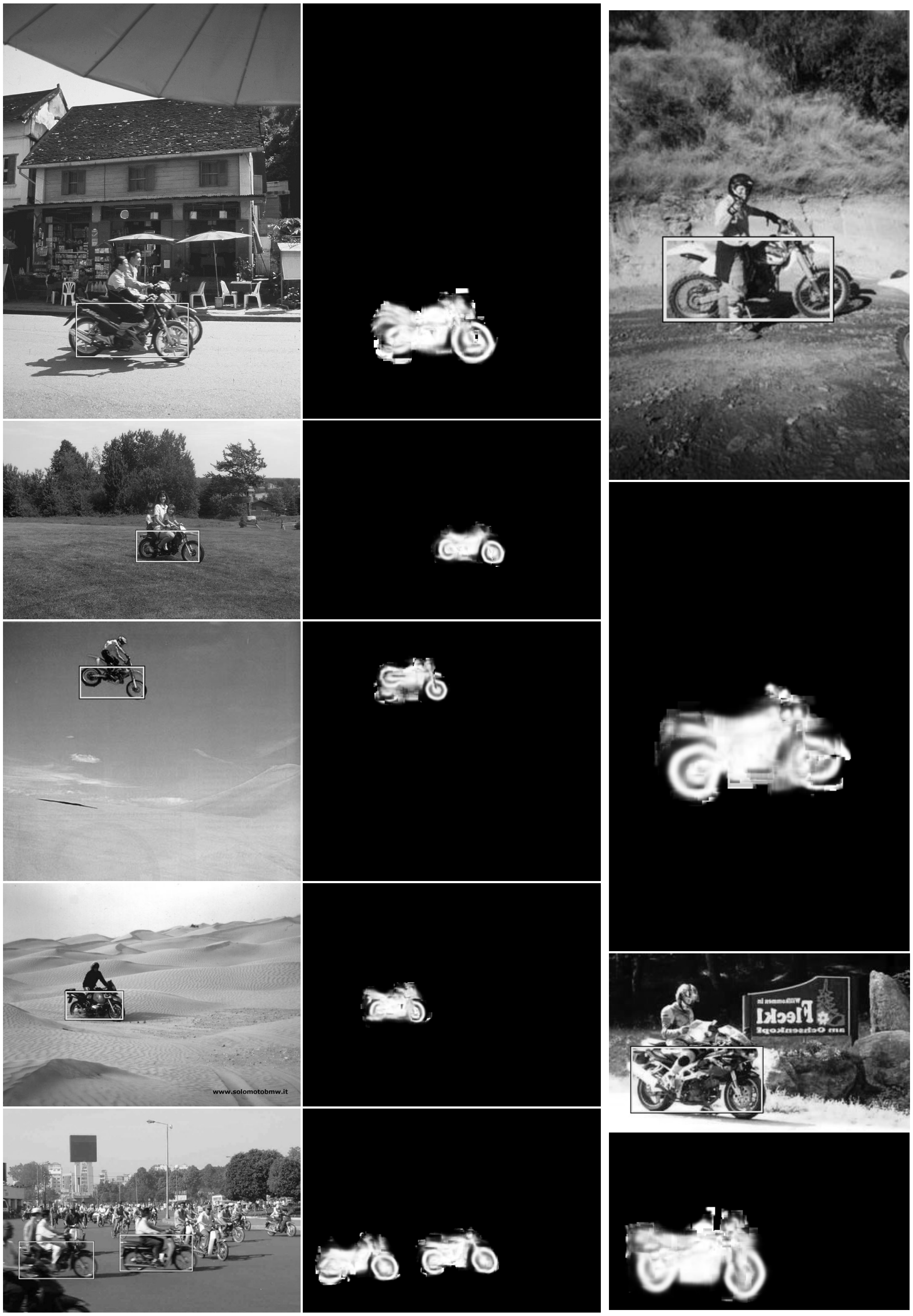


Figure 8.16: *Example detection and segmentation results for motorbikes in difficult real-world scenes.*

Method	EER Performance
Fergus et al. (2003)	90.3%
Our algorithm	93.9%

Table 8.2: *Comparison of our results on the CalTech data set for rear views of cars with others reported in the literature. The table shows the EER performances for an object present/absent decision task.*

8.3 Application to Rear Views of Cars

Last but not least, we apply our system to the detection of rear views of cars, again using the CalTech database. The system is trained on the 126 (manually segmented) images of the `cars-markus` data set, resulting in a codebook of size 559 with 45,774 occurrences. Since no detection results on this category have been reported in the literature so far, we again evaluate our method on an *object present/absent* task using the 526 car and 1,370 non-car images of the CalTech `cars-brad` data set. This data set contains road scenes with significant scale variation. The task is again to decide whether or not there is a rear view of a car in the image.

Table 8.2 shows the results of this experiment. Our approach achieves an EER performance of 93.9%, which is again superior to previously reported results. The better performance compared to (Fergus et al., 2003) can be explained by the larger number of parts our approach is able to use. Since the car views mainly consist of uniform regions, it is hard for any local approach to find enough discriminative features. Consequently, the six parts used in the Constellation Model are not as distinctive in their appearance as for other object categories and cannot provide as much evidence. Our approach has the same problem of finding good features, but its larger codebook allows it to compensate for the codebook entries' lower individual discriminance.

Figure 8.17 presents some examples of correct detections on the test set. As can be observed, the approach is able to find a large variety of car appearances at different scales in the images. Some typical problem cases are shown in Figure 8.18. The first two concern the detection bounding boxes. As the car's shadow proves to be an important feature for detection, a displaced shadow sometimes leads to a misaligned bounding box (Fig. 8.18(a)). Also, similar structures on the rear window and trunk may cause the object center to be estimated at a wrong location (Fig. 8.18(b)). A third cause for incorrect detections is the limited scale search range of [0.3,1.5] used in this evaluation. Some of the cars are too large for this range, so that their size cannot be properly estimated (Fig. 8.18(c)). However, even if the search range is extended, very large objects still cause problems, since only a small number of interest points is found on those scales[6]. Finally, some spurious detections are found on regions with similar image structure (Fig. 8.18(d)).

---

[6] Since the DoG detector searches for interest points with circular support, the number and location of extracted patches that still fit inside the image is naturally restricted at larger scales.

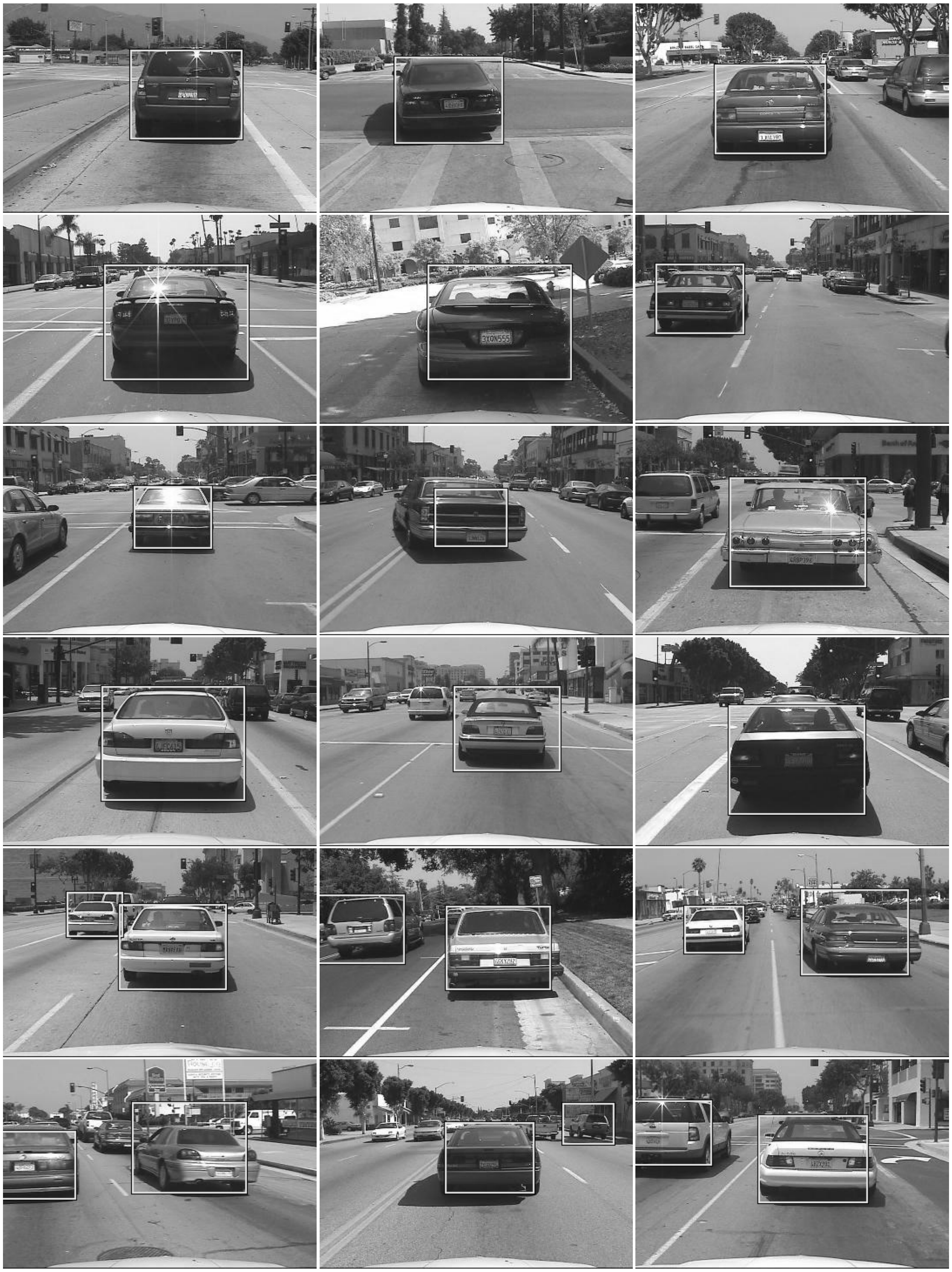


Figure 8.17: *Examples for correct detections of rear views of cars on the CalTech data set.*

When trying to evaluate the approach's detection performance also quantitatively, we encountered a similar problem as for the pedestrian experiments with overlaps. In many cases, it is a matter of interpretation whether a certain object should be annotated in the test images or not.

Three factors are responsible for this uncertainty. One is that each detector is tuned to a specific task, e.g. to detect rear views of passenger cars. Often, there are borderline cases the detector was not explicitly trained for, but which are sufficiently close to the target category. In our example, this applies to rear views of trucks or buses. Even though a good detector will respond to some such cases, we cannot reasonably expect it to detect all of them.

A similar argument can be made about the scale search range. An advantage of local approaches compared to global methods is that they are more robust to partial occlusion. However, they depend on object parts to be visible at a sufficient resolution in the scene and are thus confined to larger object scales. As has been pointed out by several researchers, it is therefore useful to employ different detectors at different scale ranges (Mikolajczyk et al., 2004; Kruppa, 2004). Hence, a local approach can only be expected to detect objects above a certain minimum scale. However, if it exceeds expectations by detecting also some object instances at smaller scales, those should not count as false detections.

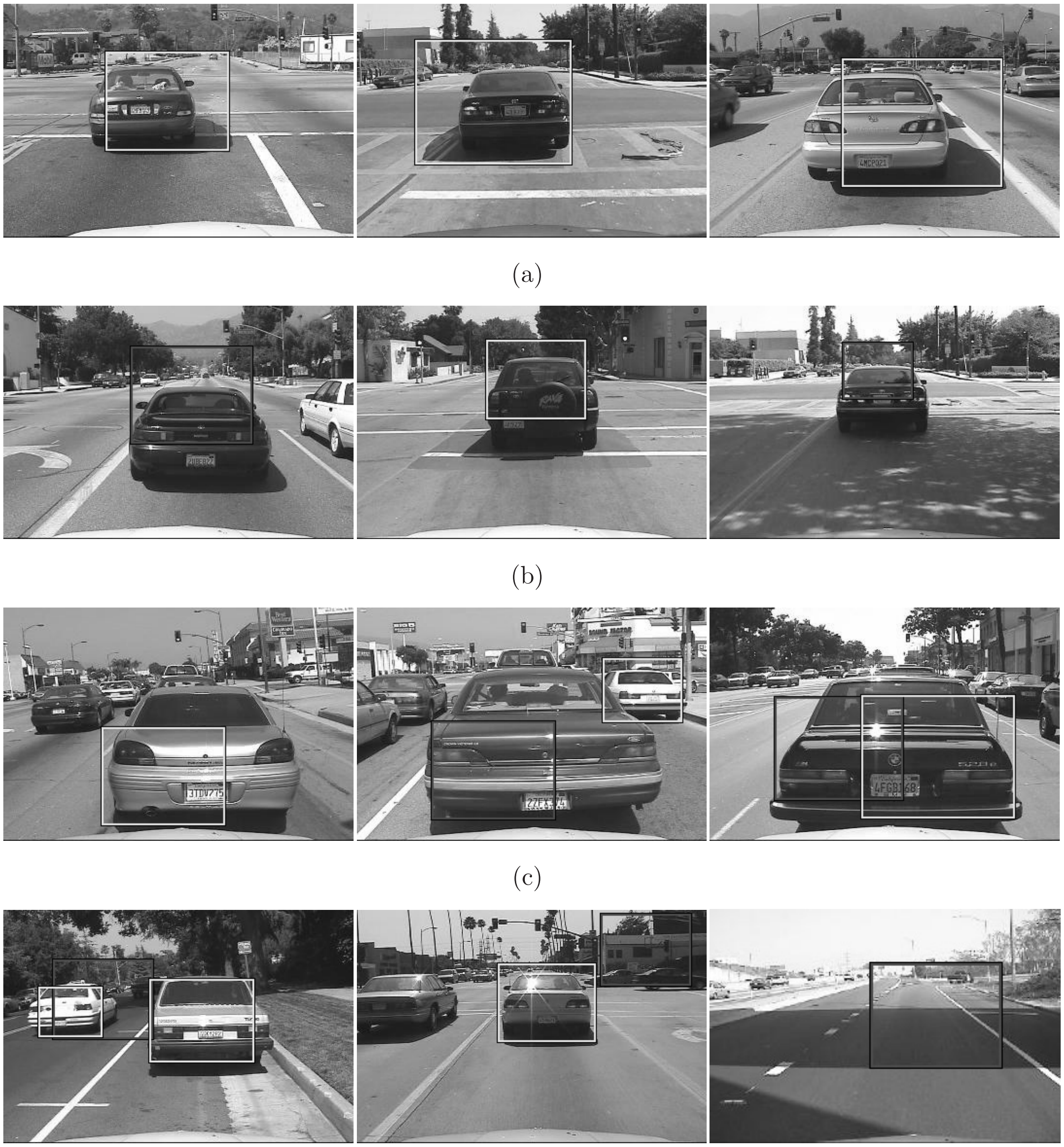
Finally, a detection system can only be expected to tolerate a limited amount of overlap and occlusion. As can be seen from examples such as the middle image of Figure 8.18(c), it is not obvious where to draw the border. If all objects are annotated, the results will be overly pessimistic; if only unoccluded objects are counted, some true detections will be lost.

The resulting dilemma is intrinsic to evaluations in real-world situations. In our opinion, a solution of simply omitting all ambiguous images from the evaluation is not an option. We therefore propose to solve the problem by using a two-level annotation and distinguishing between cases that an approach *must detect* and those that it *could detect*, but which do not count as errors if they are missed. This way, a quantitative evaluation can focus on the cases an algorithm was designed for and treat additional detections as "bonus material". In the future, we plan to perform such an evaluation on the pedestrian and car data sets, but at the time of writing, the corresponding annotations were not yet available.

8.4 Extensions

8.4.1 Multi-Cue Integration

As already mentioned in Chapter 6, additional local cues can be integrated into our system via the voting scheme. Since the initial hypothesis generation stage is based on probabilistic votes for possible positions of the object center, it does not matter which cues these votes were derived from, as long as the confidence in the cue is reflected in the vote weights. Thus, the probabilistic framework can be readily extended to accommodate multiple cues with different confidences. However,



(a)

(b)

(c)

(d)

Figure 8.18: *Typical problem cases observed for rear views of cars: (a) alignment of the detection bounding box on the car's shadow; (b) wrong estimation of the object center due to similar structures on the rear window and trunk; (c) effects of limited scale search range; (d) spurious detections caused by similar image structures.*

our chosen implementation of representing the spatial probability distribution of codebook entries by their stored occurrence locations is optimized for local features with well-localized responses. If a new cue does not yield localized responses on

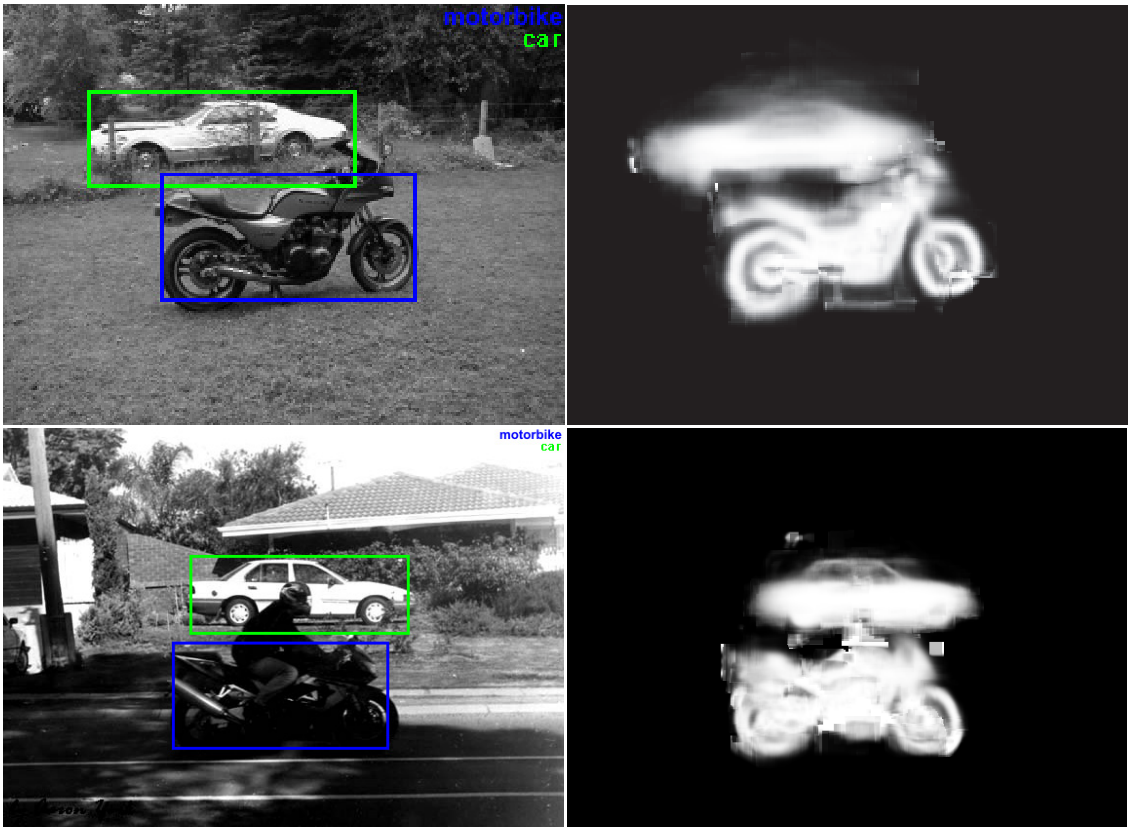


Figure 8.19: *Example detection and segmentation results for scenes containing multiple objects of different categories.*

the object category, a different representation might be necessary for this step, e.g. using a parametric model.

A combination with global cues, on the other hand, should take place on the hypothesis level. Our approach's top-down segmentation can provide an initial localization of the object of interest, so that global measures can be more reliably extracted. Conversely, global cues can enforce a higher level of consistency between local measurements and thus compensate for our approach's weaknesses. Indeed, our earlier comparison with Chamfer matching has shown that silhouette cues are complementary to our local approach and could thus be a profitable extension.

8.4.2 Multi-Category Discrimination

Up to now, we have only considered one category at a time. When objects of multiple categories shall be detected simultaneously, we have to distinguish two cases. If the categories are sufficiently distinct, it is possible to simply execute several detectors in parallel. Figure 8.19 shows some examples where this is done for (side views of) cars and motorbikes. In those cases, it is possible to use either a separate codebook for each individual detector, or, more efficiently, to combine them into a single

Category	EER Perf.
car	91.0%
cow	92.5%
motorbike	80.0%

	car #170	cow #557	motorbike #400
car	-	0.07	0.18
cow	1.00	-	1.05
motorbike	1.07	0.29	-

Table 8.3: *(left) EER object detection performance for three single-category detectors on their respective test sets; (right) cross-category confusions (false positives per test image) when the detectors are applied to each other's test sets.*

codebook with separate occurrence distributions for the different categories. Each detector is operated at its desired level of precision. As long as the hypotheses do not overlap, there is no difference to the single-category case. If there are overlaps, the MDL hypothesis verification scheme from Chapter 6 can be used to resolve the ambiguity. However, when this is done, it is important to weight the hypotheses with the relative object sizes, as discussed in Chapter 7.

If, on the other hand, some very similar categories need to be distinguished (such as for example cows and horses), the task becomes more difficult. Our evaluation from Chapter 3 has shown that in order to discriminate between such cases, it is important to look at specific object details and consider different cues. Which details and cues are best-suited for this task depends on the pair of categories to be distinguished. Therefore, this task necessitates a discriminative model, in contrast to the representative model used so far for detection. Finding the best way to integrate the two models will be a topic of future work.

In order to measure the discriminance of our existing single-category detectors, we evaluate the cross-category confusions for three visually dissimilar categories[7]. The detectors are trained on side views of cars, cows, and motorbikes using the training sets introduced before. All of them use exact DoG interest points and are operated in a scale range of [0.3,1.5] of the respective training scale. We apply all detectors to three different test sets: the UIUC single-scale test set for cars; the CalTech test set for motorbikes; and for cows a set of 556 test images from the same stock as in Section 6.3.2. Each detector is first evaluated on its "own" test set, producing the EER performances shown in Table 8.3(left). We then measure the cross-category confusions on the other test sets when the detectors' parameters are left at the EER point. Table 8.3(right) displays the results of this experiment in terms of false positives per test image. As can be seen from those numbers, the car detector is very discriminant and achieves low false-positive rates also on the other test sets. The cow and motorbike detectors, on the other hand, are less specific and yield higher false positive rates (the relatively large number of false positives on the car images can be partially explained by the fact that those images are about twice as large as the images of the other categories). To a certain extent, this

---

[7] This experiment has been performed in collaboration with Mario Fritz.

could be expected, since the cow and motorbike categories contain more variation in appearance. However, the results also motivate the use of a discriminative model as an additional verification stage.

8.5 Discussion

In this chapter, we have demonstrated the versatility of our approach on three new categories: pedestrians, motorbikes, and rear views of cars. Our results show that the system is applicable to all of them without requiring changes to the method nor the underlying features. The increased difficulty of the pedestrian detection task allowed us to evaluate the effects of several parameters more closely and confirm the method's robustness to scale changes also for a more challenging scenario. In addition, the pedestrian experiments underline our approach's ability to deal with crowded scenes containing multiple overlapping objects.

In order to speed up the training phase, we have experimented with different strategies for minimizing the amount of manual segmentation labor. For the pedestrian category, this could be achieved by recording video sequences and using motion cues to automatically generate training segmentations. For motorbikes, we picked out training images with roughly uniform backgrounds and segmented them using just the Flood Fill function of standard graphics software. Both methods significantly reduced the training effort while producing satisfactory recognition results. However, when choosing the second option, care must be taken to include backgrounds of different brightness levels, so that learned codebook entries are not biased towards a particular background color.

Altogether, the results demonstrate our approach's usefulness for real-world detection tasks. Nevertheless, some extensions can still be beneficial. For very large scale changes such as the ones encountered in the experiments with rear views of cars, it can be advantageous to work on several rescaled versions of the image. One reason for this is simply computational efficiency. Interest points and objects can be searched faster at smaller scales. However, another reason is that most interest point operators, although called "scale invariant", are usually still optimized for a certain scale range (as also argued by Ferrari et al. (2004)). For example, the way the exact DoG detector is designed, it filters the image by a series of Gaussians whose scales differ by a constant multiplicative factor. As a result, the sampled scale levels are spaced further apart with increasing scale, making scale interpolation less reliable and providing less support for hypotheses.

Other possible extensions include the integration of multiple cues and the combination of several detectors for multi-category discrimination. In this chapter, we have discussed several strategies for these combinations and pointed out how the system can be adapted to accommodate them.

In the course of our experiments, we have also identified a methodical problem for the evaluation of object detection algorithms in crowded scenes. In uncontrolled situations, it is often not obvious which objects should be annotated because of partial occlusion, limited scale search range, and borderline cases for the category membership. Even though a good detector will be able to detect some such cases, we cannot reasonably expect it to detect all of them. To our knowledge, this problem has so far not been treated in the literature — possibly because it does not occur for face detection — but it is intrinsic to real-world evaluations of many other object categories with a larger spatial extent. We have proposed to solve this dilemma by using a two-level annotation and distinguishing between cases an algorithm *must detect* and those that it *could detect*, which are not counted as false positives if they are found, nor as errors if they are missed.

An important advantage of our approach compared to other object detection methods is that its flexible representation allows it to learn object models already from a relatively small number of training examples. In the experiments reported above, the recognition performance could already be achieved using training sets with only 100–150 images. However, as only local consistency with a common object center is enforced, this flexibility has the disadvantage that local parts could also be matched to potentially illegal configurations. Consequently, a pedestrian with three legs would be considered a valid hypothesis by our system, since the system has no semantic interpretation of detected object parts and no higher-level knowledge how they should fit together. The next chapter will therefore explore how such a semantic interpretation can be learned from training images and how it can be interfaced with higher-level reasoning mechanisms.

9

Learning Semantic Parts

Approaches encoding local and global appearance of objects have proven successful for the identification of known objects and the detection of single object categories in real world scenes. Appearance based approaches, however, are often criticized for being purely data-driven or "bottom-up" and therefore not allowing a sensible high-level interpretation. Also, since instances of a particular object category may vary substantially in their visual appearance, similarity in visual appearance alone is often not sufficient for object categorization. Recognition by parts, on the other hand, has been promoted due to the possibility to incorporate high-level knowledge and top-down reasoning. Early approaches based on parts (Biederman, 1987) had limited success, mainly because the parts were often postulated and could not be reliably extracted from real-world images.

In order to ground them in reality, the parts should be learned from training data. Many recent approaches therefore learn the appearance of a hand-defined set of parts by training specific part classifiers (Mohan et al., 2001; Heisele et al., 2001; Ronfard et al., 2002; Mikolajczyk et al., 2004; Kruppa, 2004). However, this still requires manual supervision on the part of the designer. Success depends on how well the chosen parts represent the category. Moreover, some characteristic features are likely to be missed if there exists no human-level name for them. It is thus desirable to learn not only the part appearance, but also which parts to use. This chapter investigates the question how the semantic structure of an object category can be learned automatically.

One of the main points we make in this chapter is that visual similarity alone is not enough for this learning step. Even on a part level, the same semantic structure can give rise to many different appearances, e.g. due to intra-class variability, changed lighting conditions, or poor alignment. Creating one single appearance model encompassing all those variations is both difficult and problematic, since it will invariably incur a high rate of false positives. Our approach is instead to use a hierarchy of grouping steps, each based on a different criterion.

In the first stage, we learn a large number of simple and visually compact local appearances (an "appearance codebook", as described before), which can be reliably extracted and matched. The later stages then introduce weak top-down constraints, such as the information that the object views in two images are aligned. These

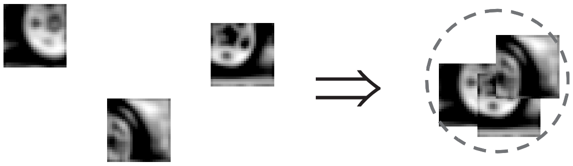


Figure 9.1: *Grouping of patches into parts.*

constraints lead to two grouping principles, *co-location* and *co-activation*, which are used to further group the appearance clusters. Our implementation of these additional grouping steps is built on a statistical modelling of mutual predictability between local features based on the $minCP$ criterion (Edelman et al., 2001), which has been shown to play an important role in human learning of visual structure. The resulting clusters often correspond to physically and semantically meaningful parts of objects, such as trunk, windshield, and wheels of a car.

During recognition, the appearance codebook is used to generate initial object hypotheses, which are then verified by a Bayesian network that encodes the overall topology of the learned subparts and parts. Experiments show that the method can operate on real-world scenes and recognize categorical objects in novel settings. At the same time, the overall structure of the multi-stage categorization approach lends itself to a sensible semantic interpretation.

The chapter is structured as follows. The next section introduces the guiding principles co-location and co-activation and shows how they can be used to learn sub-parts and parts of objects. Finally, Section 9.2 shows how the learned subparts and parts can be used to verify object hypotheses in a Bayesian network.

9.1 Learning Object Parts

Clearly, there exists the problem of a semantic gap between the visual information present in the image and our human-level interpretation. Consider for example the three patches shown in Figure 9.1. They are visually very dissimilar, so that they cannot be grouped based on their visual appearance alone. Yet, humans know that they contain similar information and that they are all sub-parts of a wheel. If we wanted to learn spatial relations between semantic structures directly on the patch level, we would need to collect statistics for many combinations of such wheel parts, only some of which may be present in any one image. This may be possible for one object view, but generalizing it to all possible views of an object category would require a considerable amount of training data.

On the other hand, if we could bridge the gap and infer, upon seeing any of the above patches, the presence of a wheel, this would allow us to learn spatial relations on the part level. The benefits would be the need for less training data and increased robustness since the system would no longer need to observe exactly one particular patch feature – any wheel patch would suffice.

The difficult questions are how to learn this semantic structure and what constitutes a part. In the context of object categorization, we can assume that the parts we are interested in have a certain spatial extend, but are localized on the object. Patches are therefore probable to be semantically close (meaning they belong to the same object part) if they repeatedly occur in a close spatial neighborhood. For learning parts, we have identified three factors to be important, namely *visual similarity*, *co-location*, and *co-activation*.

- *Visual similarity* is used to group local appearance features and obtain a compact representation of what can be observed on the objects of the target class. The result of this step is a codebook of appearance clusters, i.e. a class-specific vocabulary in terms of which images can be described. In order not to compromise the following steps, it is important that the resulting clusters be compact, i.e. that they do not lose their specificity in the clustering process.
- *Co-location* means that codebook entries should further be grouped together if they reliably occur at the same location on the object. The goal of this stage is to group codebook entries corresponding to, for example, a black and a white car trunk, which are visually dissimilar and will never be activated in the same image. In the following, we refer to these intermediate groupings as *subparts*. This stage is also necessary as a pre-step for the final principle, co-activation.
- *Co-activation* expresses that subparts should be grouped if they are consistently activated in a certain spatial neighborhood. This can be used, for example, to group local features from the upper and lower regions of a wheel, or to learn the layout of neighboring object parts.

The main reason why we introduce these separate steps instead of directly learning the layout of features on the whole object is learning complexity. Given the large appearance variability of even a relatively simple and well-defined object category such as cars, a large number of training examples would be needed in order to obtain reliable statistics for spatial relations between bottom-level local features. By separating the process into a series of clustering steps based on different criteria, the subproblems to solve are smaller, so that less training examples are required. In addition, this approach allows to exercise tight control over the clustering parameters, so that we can make sure that grouping is only pursued as long as the respective criterion performs reliably.

We have thus arrived at a viable definition for object parts. In the following sections, we show how we can use this definition to learn parts based on those principles.

9.1.1 Visual Similarity

The first step is to group image patches that are *visually similar*. In our approach, this is achieved by the initial codebook clustering stage from Chapter 4, which automatically determines the number of clusters and guarantees that the resulting

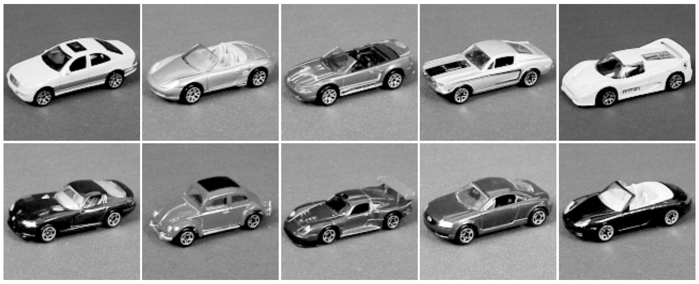


Figure 9.2: *Training objects used for cars (from the CogVis-ETH80 database (Leibe and Schiele, 2003a)). For each object, 16 views were taken from different orientations.*

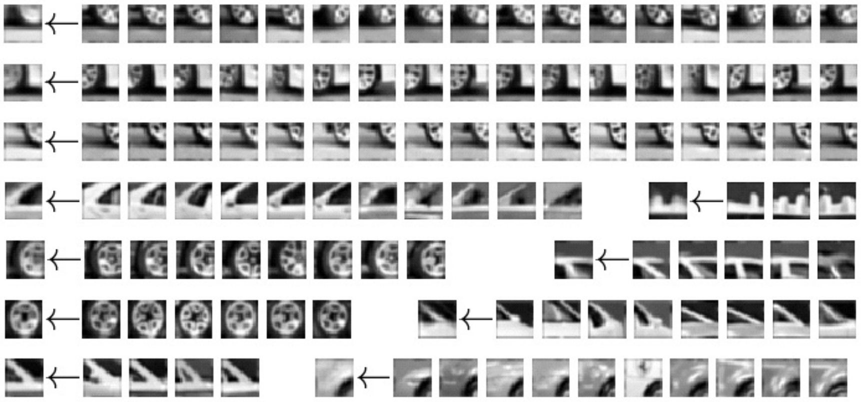


Figure 9.3: *Example codebook clusters with their corresponding patches. Only the cluster centers are stored.*

clusters stay visually compact. The result of this step is a codebook of visual appearance, i.e. a compact summary of the visual information observed on the training images. Based only on visual clustering, the codebook contains no semantic information yet, but forms an overcomplete representation with several entries for different appearances of the same object part. Semantic groupings are then achieved by application of the other two principles.

As a running example, we show results on a car data set. For training, we use a set of 160 images corresponding to 16 views around the equator of each of the 10 objects shown in Figure 9.2. The initial patch extraction stage generates a total of 8,269 patches from those images, which are then reduced to a codebook of size 2,519. Figure 9.3 shows some of the codebook entries, together with the patches they were derived from. While the resulting number of clusters is still high, the most interesting property of the clustering scheme is that all clusters are compact and only contain image patches that are visually similar.

In a second pass over all training images, we record where the stored clusters may occur, just as described in Section 5.1.3. For this, we again extract image patches around interest points and match them to the codebook. We activate all entries whose similarity is above t, the threshold already used during clustering. Thus, we model the full appearance distribution and avoid any quantization artifacts that could potentially be introduced by the clustering step. For every codebook entry, we store its "occurrence locations", that is all the positions it was activated in, relative to the object center. This information is used in the next step for learning subparts.

9.1.2 Co-Location Clustering

Once the initial codebook has been created, we can derive subparts by grouping based on co-location. We want to group the codebook clusters A and B if the occurrence of A in a certain location can be explained by the occurrence of B in a similar location relative to the object (in any training image showing the same object pose), and vice versa. This can be expressed by the $minCP$ criterion (Edelman et al., 2001), measuring the predictability of A from B and vice versa:

$$minCP = min(p(A|B), p(B|A))$$

If $minCP = 1$, then A and B are perfectly predictive of each other and should be merged. Psychophysical experiments by Edelman et al. have shown that the human visual system uses a criterion based on $minCP$ alongside the well-known MDL criterion for learning of visual structure (Edelman et al., 2001).

We thus need a measure how well an occurrence of B can be modelled by the probability distribution of A. In general, we are searching for localized parts. However, certain structures, like wheels, may occur in several locations on the object. It is therefore not appropriate to assume a uni-modal distribution, such as a single Gaussian, for the occurrence locations of A. Instead, we model the distribution of A by laying a small Gaussian kernel $G(x, \sigma) = e^{x^2/2\sigma^2}$ around each occurrence a of A. We can then estimate $p(B|A)$ by calculating the distance $d(b, A)$ of each $b \in B$ to the nearest occurrence of A, weighted by G. This corresponds to a weighted variation of the Hausdorff distance. The conditional probability then becomes

$$p(B|A) \approx \frac{\sum_{b \in B} G(d(b, A), \sigma)}{\#occurrences(A)}.$$

For the implementation, we can take advantage of the fact that our training images are aligned. Using the probabilities derived above, we apply the same agglomerative clustering scheme as in Chapter 4 to obtain subparts. Codebook entries are grouped as long as the $minCP$ value of their occurrence locations is above a certain threshold.

Figure 9.4 shows some typical subparts we obtained with a radius of $\sigma_1 = 7.0$. With this setting, we are able to reduce our initial 2,519 codebook clusters to 745 subparts. As can be seen from the figure, the method succeeds in obtaining good groupings of visually different, but semantically similar structures, like differently-colored fenders, windshields, or trunks. These groupings are clearly more than what could be achieved based on appearance alone.

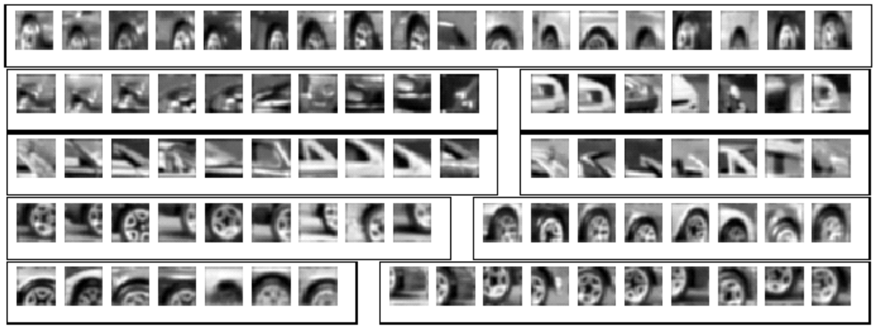


Figure 9.4: *Example subpart groupings (of codebook clusters).*

When experimenting with the parameters, we found the enforced locality to be very important. The main reason is that even on our aligned training data, some part locations may still vary considerably between object instances. Restricting the spatial extend of subparts thus serves as a guarantee that no accidental matches occur. As a consequence, however, the resulting grouping is not complete yet. For many semantic entities, there are still several instances left which cannot be further grouped due to the locality constraint (therefore the term "subparts"). Bringing together those instances is the aim of the next step, grouping based on co-activation.

9.1.3 Co-Activation Clustering

As motivated above, we want to group subparts into larger-scale structures based on their co-activation. Again, we group the subparts A and B if the occurrence of A can be fully explained by a nearby occurrence of B and vice versa, but this time, we additionally demand that the occurrence be *in the same image*. The motivation behind this is that cooccurrence in the same image allows us to compensate for parts that may occur in different locations on the object. As a result, we can relax the locality constraint and accept activations in a larger radius.

We use the same formalism as above for the subparts, only with the additional co-activation constraint. Again, we apply the agglomerative clustering scheme to obtain a grouping of subparts into parts. Figure 9.5 shows a selection of the resulting parts we obtained by setting the coactivation radius to $\sigma_2 = 14.0$. With this setting, we achieve a reduction from 745 subparts to 243 parts. As can be seen from the figure, we obtain good groupings of semantic parts that are activated over a larger area than the subparts alone. For every part, the figure also shows its main activation areas, i.e. the areas where the part occurs. It can be observed that the learned groupings are tuned to localized object parts in specific object poses.

In addition to semantically obvious parts, we also find parts for which we have no distinct human-level name. An example is the shadowed area between the car's wheels, which proves to be also a good feature for recognition. These kinds of parts

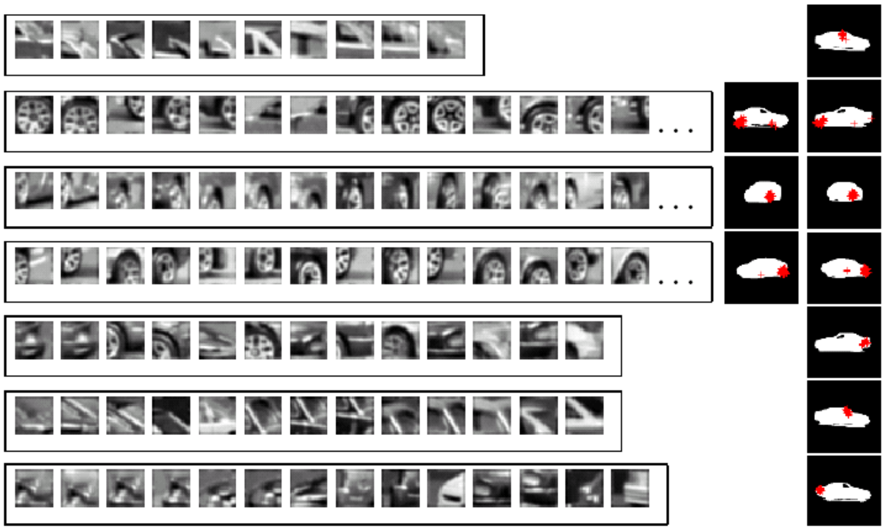


Figure 9.5: *Example part groupings (of codebook clusters) and their main activation areas (in red).*

are often overlooked when system designers hand-select ensembles of parts, even though they might be more reliable to extract from real-world images. With our method, such parts are learned automatically.

At every level of the grouping hierarchy, we have kept tight control over one parameter. This is important to make sure that only those structures are grouped for which the current criterion performs reliably. As a result, we obtain an overcomplete set of parts. In the following section, we show how the learned parts can be used for hypothesis verification.

9.2 Hypothesis Verification using Parts

The parts learned in Section 9.1.3 provide an interface between the visual information readily available from the image and its semantic content. With their help, it is possible to incorporate top-down knowledge and use inference mechanisms such as Bayesian networks (Yow and Cipolla, 1997; Ioffe and Forsyth, 2001; Pham et al., 2002), to assist with the recognition process.

As a proof of concept, we describe a part-based hypothesis verification stage for side views of cars using a Bayesian network. Rimey and Brown (1994) proposed a Bayesian network structure called Composite Net for selective perception in scene understanding, which is the basis for our Bayesian network. Here we use two of its sub-networks, namely the *Part-of Net* and *Expected-Area Net*. Both networks represent the object as a hierarchy of multiple semantic groups of object elements adhering to a topological structure. This structure naturally follows the decomposition of the object generated by its parts and subparts learned in the previous section.

The observable elements in both networks are the codebook entries constituting a subpart. This leads to a three-tiered network structure, with the subparts on the lowest level, an intermediate level representing parts, and an overall object node as root, as shown in Figure 9.6. For the experiments described below we have selected 5 parts from the parts learned in the previous section (corresponding to front wheel, rear wheel, front bumper, trunk, and windshield). In our example, those parts are composed of 3 to 11 subparts. The subpart groupings are obtained by considering only the codebook clusters that were activated in a 16-pixel radius around the part center, resulting in 3 to 13 codebook clusters per subpart.

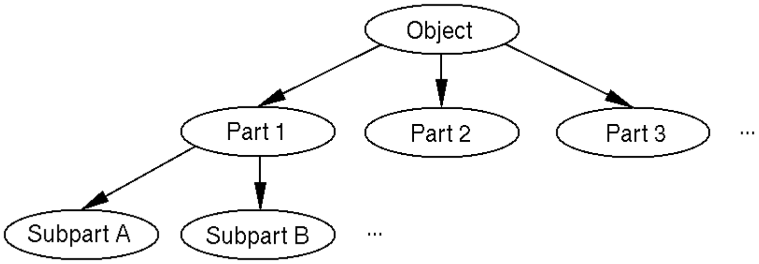
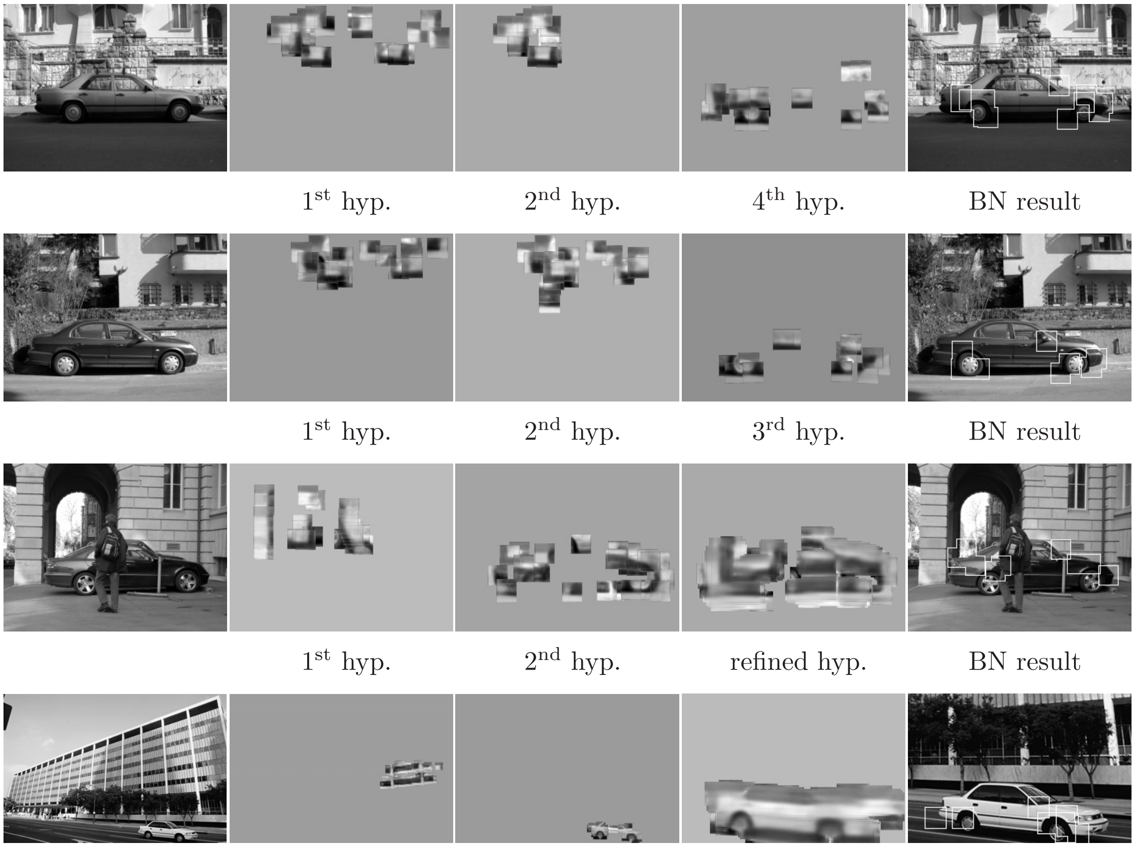


Figure 9.6: *Generic structure of the Bayesian networks used for verifying hypotheses. The leaf nodes are the observable subparts belonging to the parts. All parts together compose the object represented by the root node.*

The Part-of Net contains information about the presence and classification of the object elements. A subpart node in this network contains a discrete probability distribution over the possible instantiations of the object subpart it describes. Since the subpart nodes correspond to the observable object elements, the possible classes are the codebook clusters corresponding to the subpart that node represents. Additionally, the node can take the value *notDetected*. The part and object nodes only have two possible values: *present* and *notPresent*. Currently only the information which subpart has been detected is used, namely by summing all values other than *notDetected*. In a future scenario, the detection of particular clusters could be used to account for different detection reliabilities of the respective patch clusters.

The second Bayesian network is the Expected-Area Net, with the same three-tiered structure, where each node contains a 2D discrete probability distribution over possible locations of the corresponding element of the object. The conditional probability distribution of a node given its parent specifies where, relative to the parent location, the child node is expected to be found. These distributions have been generated by approximating the patch occurrences within the training pictures by Gaussians. Given an Expected Area, a *search area* can be computed where the corresponding element is expected to be located with a certain level of confidence.

For hypothesis verification the Bayesian networks are initialized with the object node located at the hypothesis coordinates. This results in Expected Areas getting generated for all parts and subparts which compose the object. Within the Search Areas of the subparts, an attempt is made to match the image with one of the patch



1st hyp. 2nd hyp. 4th hyp. BN result

1st hyp. 2nd hyp. 3rd hyp. BN result

1st hyp. 2nd hyp. refined hyp. BN result

1st hyp. 10th hyp. ref. (enlarged) BN 3rd hyp.

Figure 9.7: *Examples of the Bayesian network verification stage. Initial hypotheses and hypothesis ranked first by the Bayesian network, together with their extracted parts.*

clusters composing the subpart. Whenever such a match is achieved, this evidence is inserted into the Composite Net, thus rendering the hypothesis more probable and also providing more information about the positions of elements which have not been detected yet. By using the semantically expected positions of parts relative to a hypothesis, the consistency of local matches can be enforced. A further advantage is that the verification is independent from interest points, which leads to improved robustness of recognition.

Figure 9.7 shows some typical results of the part-based verification stage. The Bayesian network succeeds in discarding the wrong hypotheses and ranks the correct ones first. As a by-product of verification, the Bayesian network is able to extract the object parts it based its decision on from the image. First results indicate that the Bayesian network can improve the initial voting result based on interest points. In future work, we will also combine it with the MDL-based verification stage.

9.3 Discussion

In this chapter, we have argued for the necessity of an intermediate layer of semantic object parts, which can be used to interface the visual information with high-level reasoning mechanisms. In order to ensure that the used parts are grounded in reality, it is important that they be learned from visual data instead of being merely postulated. In this paper, we have presented an algorithm by which this learning step can be achieved.

In addition to grouping based on visual similarity, we have introduced two other grouping principles, co-location and co-activation, and have shown how these principles can be used to learn semantically meaningful subparts and parts from statistical data. By performing the part learning not in one step, but using a hierarchy of grouping stages, we can keep tight control over the grouping parameters and ensure that only meaningful groups are generated. Each stage uses a different criterion, and the hierarchical process guarantees that each criterion is only used as long as it performs reliably. The resulting structures generalize beyond the appearance of single objects, and their activations are often tuned to semantically meaningful object parts localized in specific object poses.

While the first step of visual clustering is purely unsupervised, the succeeding steps rely on weak top-down constraints, such as alignment (for co-location) or object constancy (for co-activation). Our results from Chapter 6 show that good initial hypotheses can already be obtained from visual information alone. The learned subparts and parts can then be used to verify those hypotheses on a semantic level.

The same framework can be applied also with other grouping constraints. Example constraints that have been used in the literature so far include consistent motion tracks (Sivic et al., 2004), cyclic motion with similar periodicity (Peternel and Leonardis, 2004), and temporal consistency between local fragments that are delineated by tracked feature points (Bart et al., 2004; Bart and Ullman, 2004). In all those cases, the grouping principles can be used to express a non-visual similarity measure between local appearances. Applying our framework, these similarities would be collected over a set of training examples, whereupon agglomerative clustering would be performed on the resulting pairwise similarity matrix.

As a proof of concept for the combination with high-level reasoning, we have implemented a Bayesian network for hypothesis verification, which succeeds in improving our recognition results. The learned part hierarchy intuitively translates into an efficient tree-shaped network structure, and expected areas are automatically created for all subpart and part locations. During hypothesis verification, the network tries to search for missing object parts, guided by the evidence that has already been observed. As a result, it can compensate for missing part detections and larger variability in object shape, thus improving the recognition results. While the results show the feasibility of the approach the next step will be to automatically learn a Bayesian network also for multiple object poses.

10

Conclusion

In this thesis, we have investigated the topic of visual object categorization. While an initial study was concerned with discriminating multiple object categories in a laboratory environment, the main part of our work has focused on building a system that can reliably detect and localize categorical objects in real-world scenes.

The developed approach integrates the capabilities of object detection and figure-ground segmentation into an iterative process. It creates initial hypotheses without prior segmentation, then derives a top-down segmentation from the recognition result, and uses this segmentation to again improve recognition. As shown in our experiments, the resulting approach can learn object models already from few examples; achieves competitive performance on several standard data sets; and is robust to intra-class variation, noise, and partial occlusion. Moreover, we have generalized the approach to scale invariant detection and have discussed how it can be extended to higher-level semantic parts. Quantitative experimental results on five different object categories, including articulated objects such as walking cows and pedestrians, confirm the method's applicability in practice.

10.1 Contributions

A main contribution of our work is the integration of object category detection and figure-ground segmentation into a common probabilistic framework. As shown in our experiments, the tight coupling between those two processes allows both to profit from each other and improve their individual performances. Thus, the initial recognition phase not only initializes the top-down segmentation process with a possible object location, but it also provides an uncertainty estimate of local measurements and of their influence on the object hypothesis. In return, the resulting probabilistic segmentation permits the recognition stage to focus its effort on object pixels and discard misleading influences from the background. Altogether, the two processes collaborate in an iterative evidence aggregation scheme which tries to make maximal use of the information extracted from the image.

In addition to improving the recognition performance for individual hypotheses, the top-down segmentation also allows to determine exactly where a hypothesis's support came from and which image pixels were responsible for it. We have used this property to design a mechanism that resolves ambiguities between overlapping hypotheses in a principled manner. This mechanism constitutes a fundamental novelty in object detection and results in more accurate acceptance decisions than conventional criteria based on bounding box overlap.

The core part of our approach is the Implicit Shape Model defined in Chapter 5. This implicit representation is flexible enough that it can combine the information of local object parts observed on different training examples and interpolate between the corresponding objects. As a result, our approach can learn object models already from few examples and achieves competitive object detection performance with training sets that are between one and two orders of magnitude smaller than those used in comparable approaches.

Taking a broader view, this implicit model can be seen as a further generalization of the Hough Transform to work with uncertain data. In our approach, we have used this capability to represent the uncertainty from intra-class variation, but it would also be possible to use it with different sources of uncertainty, e.g. for the identification of known objects under lighting variations.

Finally, we have explored how the hitherto purely visual representation can be extended towards semantically meaningful object parts. We have proposed a learning strategy by a hierarchy of grouping steps based on non-visual constraints such as co-location and co-activation. The resulting representation forms an interface between the visual information readily available from the image and higher-level reasoning mechanisms such as Bayesian networks. We have shown how the learned subparts and parts can be combined in an efficient tree-shaped Bayesian network that reasons about part configurations for hypothesis verification.

10.2 Additional Remarks

In the following, we draw parallels and highlight conceptual differences between our method and several other approaches from the literature. In particular, we will examine complementary elements and discuss possible combinations.

10.2.1 Comparison with Classical Object Detection Approaches

Classical object detection approaches such as the methods by Viola and Jones (2001, 2004) and Schneiderman and Kanade (2000, 2004) achieve location and scale invariance by performing an exhaustive search over scales. They shift a search window over the image and evaluate a cascade of local features at each window location. This procedure is optimized for the sequential processing bottleneck in current computer systems. The goal is to evaluate one search window at a time with minimal effort and a minimal number of feature evaluations.

Our approach can be seen as a dual view of this procedure. Instead of shifting a search window over the image and evaluating features relative to this window, we compute features once for the whole image, then let them agree on best-ranking window locations. When applied to the same underlying features, the result would be the same for both strategies. However, our approach only needs to evaluate each feature once, regardless of how many search windows it can contribute to. (Incidentally, Schneiderman (2004) has recently proposed a similar improvement to his approach for recycling features between search windows).

An important consequence of this dual view is that similar improvements as in those approaches can also be used in our system and vice versa. One example is the organization of the codebook in a cascade, which our approach could profit from by matching features to the most discriminative codebook entries first. Conversely, it would be interesting to integrate our probabilistic segmentation step into e.g. Viola and Jones's system (which would, however, require their approach to represent objects at a larger resolution).

However, the comparison also shows a major difference. In contrast to the above-mentioned approaches, our method is optimized for parallel and localized processing. In such a framework, it does not matter if potentially unneeded features are evaluated, as long as all processing is done in parallel. Thus, our recognition approach can be realized by a relatively small number of local units working in parallel. Each unit evaluates features from a local image region on its own, and the contributions of different units are only combined when they cast votes for a common object center. In Section 10.3, we will discuss this idea in more detail down to a potential neural implementation.

10.2.2 Comparison with the Constellation Model

The Constellation Model by Weber et al. (2000a,b) and Fergus et al. (2003) has been introduced for unsupervised learning of object categories. It represents objects by estimating a joint appearance and shape distribution of their parts. Thus, object parts can be characterized either by a distinct appearance or by a distinct location on the object. As a result, the model is very flexible and can even be applied to objects that are only characterized by their texture (such as the "spotted cats" category shown in (Fergus et al., 2003)). However, the method is typically restricted to a small set of only 5–6 parts. As discussed in Chapter 2, this restriction is the result of two conceptual differences to our approach: the use of an explicit model for representing object shape, and the joint estimation of appearance and shape parameters. The explicit model carries the assumption that it is possible to find a consistent set of parts that are present in every image, while the joint estimation results in a rapidly increasing number of model parameters for every additional part.

In contrast, our implicit approach does not estimate a joint distribution, but treats each codebook part independently. As a result, only a small subset of the parts need to be present in the image, and the influence of a part on the final model is not decreased if more parts are added. In addition, the appearance and shape distributions are estimated sequentially, so that the complexity of the learning step is reduced. Finally, since sampled image patches are not exclusively assigned to the best-matching codebook entry but to all sufficiently similar entries, parts do not compete with each other for training data. Together, these properties allow our method to use a far larger number of parts, which is one reason for its better performance on some of the test sets.

As a restriction compared to (Fergus et al., 2003), though, our approach needs the object positions and sizes to be known during training[1] and cannot learn category models in a fully unsupervised way. Which approach is better-suited for a certain application thus depends on the task.

If unsupervised learning is desired, a combination of the two methods could be advantageous. In such a combined system, the Constellation Model would be used to learn an initial object representation in a purely unsupervised fashion. The learned model could then in turn be used to locate and align the training objects, whereupon our approach would be applied to augment the model by additional parts. In this respect, it is interesting to note that the occurrences of our automatically-learned subparts and parts are often Gaussian distributed and can thus be described well by a parametric model.

10.3 Biological Relevance

The architecture of the human visual cortex is fundamentally different from the architectures of current computer systems. It is a massively parallel dynamic system that relies mainly on local interactions. Moreover, it knows neither pixels nor a random-access shared memory; it cannot perform any global operations on the full image; and most of its cells have fixed receptive fields (Edelman and Intrator, 2004). These differences have important consequences for the way visual functions are implemented.

First of all, since the vision system is active (meaning it can direct its gaze at will), translational invariance is only of secondary importance. When trying to recognize objects, it is therefore not necessary to search the whole image, but only a relatively small area around the fixation center. Next, local operations are cheap, since they can be executed in parallel. Thus, instead of performing a search over parameters, it is often more efficient to hard-code several discrete parameter settings by implementing them as tuned cells and interpolating between their graded responses. For example, rather than searching for scale-invariant features, it would be more efficient in a neural system to evaluate several features with different support and interpolate between their responses. Finally, in a dynamic system the first result does not have to be final. The human visual system contains many feedback loops and can refine its results in an iterative process by applying selective reinforcement or inhibition.

---

[1] Fully segmented training images are only required when a top-down segmentation shall be computed.

These constraints lead to a complementary view of our recognition procedure. The codebook entries used so far are similar to complex cell responses in area V4 and the anterior inferotemporal (IT) cortex of the human visual system (c.f. also Ullman et al., 2002). However, instead of sampling many image locations and matching each with the whole codebook in order to vote for an object's whereabouts, the hypothesized object location is more or less fixed at the center of fixation. Thus, each sampled location only needs to be compared to features that are compatible with its relative position. In this setting, the formerly global act of matching the image content to the codebook can be seen as the first stage of an RBF neural network with component cells tuned to specific appearances occurring in a fixed receptive field. The core part of our approach, the probabilistic framework, is still applicable to this situation. The $p(I|\mathbf{e})$ probabilities model the activation potential of the corresponding RBF cell. Together with the feature's contribution $p(o_n, x|I, \ell)$ to the object hypothesis, they form a probabilistic vote with weight $p(o_n, x|\mathbf{e}, \ell)$, which is passed to the next stage of the recognition system.

In our approach, the evidence of local votes was aggregated by a Generalized Hough Transform. In a neural system, the same effect can be achieved by a retinotopic map of "accumulator cells" that are connected in a winner-takes-all (WTA) fashion. When an RBF cell votes for a certain object location, the vote is passed to the corresponding accumulator cell, and, to a lesser degree, also to its immediate neighbors. Accumulator cells with locally maximal support are then singled out by suppressing the weaker responses of their neighbors. This interpretation of the recognition process is compatible with the Selective Tuning Model for visual attention (Tsotsos, 1990; Tsotsos et al., 1995; Cutzu and Tsotsos, 2003), which is well-established in the biological vision literature. However, as Tsotsos (1990) points out, the complexity of WTA circuitry makes it necessary to execute the maxima search not in one global step, but through a hierarchy of successive processing stages.

In the Selective Tuning Model, this WTA hierarchy is then traversed in a top-down, coarse-to-fine manner to locate the original stimulus in the visual field and suppress or attenuate connections in an annular inhibitory zone. Our probabilistic figure-ground segmentation scheme describes a similar feedback loop that reinforces contributions from the *figure* area and suppresses the influence of surrounding *ground* regions. Again, the probabilities in our scheme can be correlated to possible neural paths. Finally, the third stage of the Selective Tuning Model postulates a re-propagation of the selected stimulus through the network without the distracting stimuli of background regions. This is exactly the same mechanism we are using as the basis of our hypothesis verification stage.

When working with many object categories, it would be problematic to assume that a separate WTA hierarchy exists for each category. Instead, it is more plausible that the intermediate levels of the hierarchy perform also a function of grouping higher-level features or larger-scale object parts that are shared by multiple object categories. The higher-level parts learned by our semantic grouping mechanism from Chapter 9 are possible — though certainly not exclusive — candidates for this.

Thus, we can conclude that although the implementation details differ due to the underlying hardware, our scheme is compatible with a possible neural mechanism. Moreover, it is consistent with an established model for visual attention and could be used to extend this model also to object categorization. In this context, our contribution is twofold. Our probabilistic framework formalizes the equations that govern the modelled mechanism, and our experimental results provide a computational feasibility proof that the resulting scheme is suitable for object categorization. Whether it really corresponds to one of the mechanisms that are active in the human visual system cannot yet be ascertained. But our formalization can be the basis for psychophysical experiments to verify this hypothesis.

10.4 Perspectives

There are several natural extensions to this work:

Multi-Cue Integration. As already briefly discussed in Chapter 8, it would be worthwhile to integrate also other types of cues. While all object categories tested in our experiments could be successfully detected using just raw patches as features, other categories might require different features. Candidate local features that have been used in the literature so far include high-pass filtered patches (Weber et al., 2000a), SIFT features (Lowe, 1999, 2001; Csurka et al., 2004; Bileschi et al., 2004), "edge probes" (Carmichael and Hebert, 2003), SIFT-like edge descriptors (Mikolajczyk et al., 2003), and annular shape descriptors (Jurie and C.Schmid, 2004). As argued before, such local descriptors can be easily integrated into our framework, and multiple local cues can be combined via the voting scheme. In order to complement the features used so far, it would be especially interesting to also include local features with non-circular support.

A combination with global cues, on the other hand, would be useful in order to enforce consistency between local measurements. Global cue extraction can profit from our approach's top-down segmentation, and a combination could take place on the hypothesis level. In our experiments, we have identified silhouette cues, such as the ones used in Chamfer matching, as promising candidates for cue combination.

Multi-Category Discrimination. For many real-world applications, it is also desired to discriminate between multiple categories. In Chapter 3, we have examined such a discrimination task in a laboratory environment. In real-world scenes, this task becomes much harder, since the objects need to be localized first before a category label can be assigned. It would thus be interesting to integrate this capability with the current object detection system.

When pursuing such a combination, it is important to bear in mind the different characteristics of the two tasks. While our current system employs a purely representative model that draws its power from integrating evidence over the whole object area, the multi-class categorization task requires a discriminative model that considers specific object details. Finding a good way to integrate those two models will be a topic of future work.

Multi-View Recognition. While Chapter 5 has demonstrated the principal capability of our approach to recognize objects from different viewpoints, the main part of this thesis has only dealt with the problem of detecting single views. This restriction was motivated by the increased robustness that can be achieved by a single-view detector. Still, many real-world applications require that objects be recognized from multiple viewpoints or aspects. In the literature, this problem is typically addressed by training several distinct classifiers on different aspects of the object category and pooling their responses (Schneiderman and Kanade, 2000; Mikolajczyk et al., 2004). However, as Torralba et al. (2004) argues, this solution is suboptimal, since it does not take advantage of common features that can be shared between viewpoints. Applied to our recognition framework, the main challenge therefore lies in finding a way to combine the common parts of several single-view models while keeping their discriminance properties.

Active Sampling. Currently, our approach just relies on the output of an interest point detector for sampling image patches. This may lead to an irregular sampling density of the initial recognition stage and thus to an unwanted bias for certain structures. While a uniform sampling strategy circumvents this problem, it is computationally too expensive for many applications. A better strategy would be to actively sample the image for locations that have not yet been selected by the interest point detector, but that would provide additional evidence for a hypothesis (as determined by the hypothesis's *figure* probability map). The exhaustive search method of Felzenszwalb and Huttenlocher (2005) would be applicable for this purpose, since our model is very similar to a star graph.

Use of Context Information. Recognition and categorization tasks can be greatly simplified by using contextual cues, which narrow down the number of object categories, scales, and positions that need to be considered (Torralba and Sinha, 2001; Torralba, 2003). This becomes especially important when a large number of object categories shall be be detected and disambiguated at the same time. Simply combining the outputs of many individual detectors would result in an overly large number of false positives. Moreover, in many real-world situations the available stimuli are so degraded (e.g. due to low resolution, partial occlusion, or motion blur) that objects of interest cannot be recognized without using contextual information at all (Kruppa, 2004). It would thus be useful to extend the formulation of our recognition system by a contextual prior. Such contextual priors can be obtained by considering holistic information about the scene (Torralba and Sinha, 2001; Torralba, 2003), or by modeling its semantic content (Vogel, 2004).

Combination of Top-Down and Bottom-Up Segmentation. A pure top-down segmentation approach is restricted in that it can only correctly segment object details it has seen before on the training examples. If a target object deviates too much from this learned appearance, e.g. due to a novel articulation, the resulting segmentation may be incomplete. However, objects in real-world images are often delineated by intensity or texture discontinuities, which bottom-up methods can take advantage of. A combination of top-down and bottom-up segmentation methods can thus result in an improved segmentation quality. Two examples for such a combination can be found in (Yu and Shi, 2003; Borenstein et al., 2004), but other formulations are also conceivable.

Online Learning. For humans, learning is a continuous process that goes on for our whole life. Yet, computer vision systems are typically constructed with a fixed training phase in which all learning takes place. In order to build embodied vision systems that operate in and interact with the real world, it is important that the ability to learn and adapt is also kept throughout the system's lifetime.

A possible implementation of this principle could be to start the system with a relatively small initial training set and let it improve its representation by adding information from correctly detected (and segmented) objects in novel images. Since a correct detection can usually be achieved already from few measurements, the remaining object area can add useful information. However, when this is done, it becomes important to also employ bottom-up segmentation cues, so that the system can compensate for incorrectly hypothesized object regions.

With increasing system lifetime, the scalability of the employed representation also becomes an issue. Our current nonparametric implementation is motivated by the small number of training examples used in our experiments. In contrast, the structure learning process from Chapter 9 needs a critical mass of training examples for semantically-motivated grouping steps. Thus, the two processes can be used to complement each other in different phases of the system's lifetime. In the beginning, the Implicit Shape Model would be employed to learn object models from few training examples. When enough information is available for structure learning, the model could then be augmented by semantically meaningful parts in order to arrive at a more compact representation.

Connection with Biological Vision. Finally, it will be rewarding to pursue the connection to biological vision research. The previous section has shown that our probabilistic framework is compatible with a possible neural mechanism. However, the implications for a neural implementation need to be worked out in more detail in order to formulate specific predictions that can be verified by psychophysical or neurobiological experiments. In any case, our method remains a computational existence proof for a mechanism that achieves figure-ground segregation as a result and extension of object categorization. Our experiments have shown that the resulting feedback loop allows a significant increase in recognition performance. There is thus a strong motivation to investigate if a comparable mechanism can be found also in mammalian visual systems.

A

Interest Points

A.1 The Harris Operator

The popular Harris/Förstner operator (Förstner and Gülch, 1987; Harris and Stephens, 1988) was explicitly designed for geometric stability. It defines keypoints to be "points that have locally maximal self-matching precision under translational least-squares template matching" (Triggs, 2004). In practice, these keypoints often correspond to corner-like structures. The Harris detector proceeds by searching for points $\mathbf{x}$ where the autocorrelation matrix $\mathbf{C}$ around $\mathbf{x}$ has two large eigenvalues. The matrix $\mathbf{C}$ can be computed from the first derivatives in a window around $\mathbf{x}$, weighted by a Gaussian $G(\mathbf{x}, \tilde{\sigma})$:

$$\mathbf{C}(\mathbf{x}, \sigma, \tilde{\sigma}) = G(\mathbf{x}, \tilde{\sigma}) \star \begin{bmatrix} L_x^2(\mathbf{x}, \sigma) & L_x L_y(\mathbf{x}, \sigma) \\ L_x L_y(\mathbf{x}, \sigma) & L_y^2(\mathbf{x}, \sigma) \end{bmatrix} \tag{A.1}$$

Instead of explicitly computing the eigenvalues of $\mathbf{C}$, the following equivalences are used

$$\det(\mathbf{C}) = \lambda_1 \lambda_2 \tag{A.2}$$

$$\operatorname{trace}(\mathbf{C}) = \lambda_1 + \lambda_2 \tag{A.3}$$

to check if their ratio $r = \frac{\lambda_1}{\lambda_2}$ is below a certain threshold. With

$$\frac{\operatorname{trace}^2(\mathbf{C})}{\det(\mathbf{C})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r} \tag{A.4}$$

this can be expressed by the following condition

$$\det(\mathbf{C}) - \alpha \operatorname{trace}^2(\mathbf{C}) > t. \tag{A.5}$$

In all experiments reported in this work, we used the following parameters for the Harris detector:

$$\begin{aligned} \sigma &= 3.0 \\ \tilde{\sigma} &= 2.0 \\ \alpha &= 0.06 \\ t &= 100.0 \end{aligned}$$

A.2 The DoG Operator

While shown to be remarkably robust to image plane rotations, illumination changes, and noise (Schmid et al., 1998, 2000), the locations returned by the Harris detector are only repeatable up to relatively small scale changes. For scale invariant point extraction, it is necessary to detect structures that can be reliably extracted under scale changes.

This can be achieved by building up a *scale space* (Witkin, 1983) of the responses produced by the application of a local kernel with varying scale parameter σ. It can be shown that the only operator that fulfills all necessary conditions for this purpose is the scale-normalized Gaussian kernel $G(\mathbf{x}, \sigma)$ and its derivatives (Lindeberg, 1994, 1998). Based on these results, Lindeberg (1998) proposes a detector for blob-like features that searches for scale space extrema of a scale-normalized Laplacian.

Following Lowe (1999, 2004), this Laplacian can be approximated by a difference-of-Gaussian (DoG) $D(\mathbf{x}, \sigma)$, which can be more efficiently obtained from the difference of two adjacent scales that are separated by a factor of k:

$$D(\mathbf{x}, \sigma) \quad = \quad (G(\mathbf{x}, k\sigma) - G(\mathbf{x}, \sigma)) \star I(\mathbf{x}) \tag{A.6}$$

Lowe (2004) shows that when this factor is constant, the computation already includes the required scale normalization. Similar to his approach, we choose this factor by dividing each scale octave into an equal number K of intervals, such that $k = 2^{1/K}$ and $\sigma_n = k^n \sigma_0$.

For more efficient computation, the resulting scale space can be implemented with a Gaussian pyramid, which resamples the image by a factor of 2 after each scale octave. The *fast DoG* detector used in Chapter 7 is based on such a pyramid implementation, while the *exact DoG* version eschews this speedup for more accurate localization of maxima. As this design choice entails the application of Gaussian filters at large scales, the *exact DoG* detector uses a recursive implementation of the Gaussian filter (Deriche, 1993), whose run-time is independent of the selected value of σ.

DoG interest points are defined as locations that are simultaneously extrema in the image plane and along the scale coordinate of the $D(\mathbf{x}, \sigma)$ function. Such points are found by comparing the $D(\mathbf{x}, \sigma)$ value of each point with its 8-neighborhood on the same scale level, and with the 9 closest neighbors on each of the two adjacent levels. Since the scale coordinate is only sampled on discrete levels, it is important to interpolate the responses at neighboring scales in order to increase the accuracy of detected keypoint locations. In our implementation, this is done by fitting a second-order polynomial to each candidate point and its two closest neighbors. Brown and Lowe (2002) have recently introduced a more exact approach that simultaneously interpolates both the location and scale coordinates of detected peaks by fitting a 3D quadric function, which was not yet used in our implementation.

Finally, those points are kept that pass a threshold t and whose estimated scale falls into a certain scale range $[s_{min}, s_{max}]$. The resulting interest point operator reacts to blob-like structures that have their maximal extend in a radius of $\sqrt{2}\sigma$ of the detected points (as can be derived from the zero crossings of the modelled Laplacian). In order to capture also some of the surrounding structure, we sample the image in a radius of $r = 3\sigma$ around the detected points. In addition, we used the following parameters in our implementation:

$$\begin{aligned}\sigma_0 &= 1.0\\ \sharp\text{octaves} &= 5\\ K &= 3\\ [s_{min}, s_{max}] &= [1.0, 32.0]\\ t &= 10.0\end{aligned}$$

A.3 The Harris-Laplacian Operator

The Harris-Laplacian operator (Mikolajczyk and Schmid, 2001, 2002) was proposed for increased discriminance compared to the Laplacian or DoG operators described so far. It combines the Harris operator's specificity for corner-like structures with the scale selection mechanism by Lindeberg (1998). The method first builds up two separate scale spaces for the Harris function and the Laplacian. It then uses the Harris function to localize candidate points on each scale level and selects those points for which the Laplacian simultaneously attains an extremum over scales.

The resulting points are robust to changes in scale, image rotation, illumination, and camera noise. In addition, they are highly discriminant, as several comparative studies show (Mikolajczyk and Schmid, 2001, 2003; Dorko and Schmid, 2003). As a drawback, however, the Harris-Laplacian detector typically returns a much smaller number of points than the Laplacian or DoG detectors. This is not a result of changed threshold settings, but of the additional constraint that each point has to fulfill two different maxima conditions simultaneously.

For our experiments, we used the code implemented by the original authors[1] with a fixed standard set of parameters. This implementation is also available in two variants: as a *regular*, and as a *speed-optimized* version. While we thus did not have full control over all internal parameters, the results from Chapter 7 are consistent with our observations on an own reimplementation of the detector.

It should be noted that in the meantime a new version of the detector is available[2] which uses a less strict criterion. Instead of searching for *simultaneous* maxima, it selects scale maxima of the Laplacian at locations for which the Harris function also attains a maximum *at any scale*. As a result, this modified detector yields more interest points, and it can be expected that recognition performance will improve accordingly. For the evaluation in Chapter 7, this detector was not yet available, though.

---

[1] publicly available at `http://www.inrialpes.fr/movi/people/Mikolajczyk`

[2] publicly available at `http://www.robots.ox.ac.uk/∼vgg/research/affine/detectors.html`

List of Figures

3.1 The 8 categories of the ETH-80 database. 20
3.2 Example database image and segmentation mask. 20
3.3 Grouping hierarchies for four different cues. 26
3.4 Optimal multi-cue decision trees. 28

4.1 Visualization of the codebook representation. 32
4.2 Local information used in the codebook generation process. 34
4.3 Visualization of the codebook matching process. 35
4.4 Comparison of codebooks built with k-means and average-link clustering. 40
4.5 Performance of the RNN algorithm. 47

5.1 The recognition procedure. 52
5.2 Intermediate results during the recognition process. 57
5.3 Training objects used for cars and cows. 58
5.4 Some more examples of successful recognition results on real-world test images. 58
5.5 Recognition results on a test set of 137 car images if all hypotheses up to a certain rank are considered. 59
5.6 A case where false positives were introduced due to background clutter. 59
5.7 An example where patch extraction failed completely because of bad (grayvalue) contrast. 60
5.8 Segmentation results with different confidence levels θ. 64
5.9 An example where object knowledge compensates for missing edge information. 65
5.10 Segmentation result of a partially occluded car. 65
5.11 Example segmentation results for car images. 66
5.12 Example results for cow images. 67

6.1 Results on the UIUC car database with and without the MDL hypothesis verification stage. 70
6.2 Two examples for overlapping hypotheses. 71

6.3 Comparison of our results on the UIUC car database with others reported in the literature. 75
6.4 Example detections on difficult images from the test set. 76
6.5 All missing detections and false positives on the UIUC database. . . . 76
6.6 Example object detections, *figure* probabilities, and segmentations automatically generated by our method. 77
6.7 Example object detections, *figure* probabilities, and segmentations for scenes containing multiple objects. 78
6.8 Example object detections, *figure* probabilities, and segmentations for scenes containing occluding structure. 79
6.9 Precision/Recall curves for the cow sequences when x% of the cow's length is visible. 80
6.10 Precision/Recall curves for the cow sequences when x% of the cow's length is visible. 80
6.11 Example detections and automatically generated segmentations from one cow sequence. 81
6.12 Example detections and automatically generated segmentations from another cow sequence. 81
6.13 Example detections and automatically generated segmentations from a third cow sequence. 82

7.1 Scale-invariant interest points found by the different detectors. 89
7.2 Performance comparison of interest point detectors on the UIUC database. 90
7.3 EER performance over scale changes relative to the size of the training examples. 91
7.4 Visualization of the range of scales tested in the experiments. 93
7.5 EER performance for varying training set sizes. 94

8.1 Evaluation criteria for comparing bounding boxes. 99
8.2 Example images from the training set and the different test sets for pedestrians 101
8.3 Single-scale pedestrian detection results based only on the initial votes and after the verification stage. 102
8.4 Single-scale pedestrian detection results with and without performing histogram equalization. 104
8.5 Single-scale pedestrian detection results. Effect of adapting the MSME search window size to pedestrians. 105
8.6 Comparison of the scale-invariant version of our approach with the single-scale version. 106
8.7 Comparison of our approach to the Chamfer system. 108
8.8 Main reasons for false positives of the Chamfer system/of our approach.109

8.9 Comparison of the single- and multi-scale results on test set 1. 111
8.10 Multi-scale pedestrian detection results of our approach on test set 2. 111
8.11 Example multi-scale detections of our approach on test set 2 (at the EER). 112
8.12 Multi-scale pedestrian detection results of our approach on test set 3 (with overlaps). 113
8.13 Example multi-scale detections of our approach on scenes with overlaps from test set 3 (at the EER). 114
8.14 Examples for the variety of motorbike shapes and appearances that are still reliably detected and segmented. 117
8.15 Example detection and segmentation results for motorbike images with partial occlusion and difficult backgrounds. 118
8.16 Example detection and segmentation results for motorbikes in difficult real-world scenes. 119
8.17 Examples for correct detections of rear views of cars on the CalTech data set. 121
8.18 Typical problem cases observed for rear views of cars. 123
8.19 Example detection and segmentation results for scenes containing multiple objects of different categories. 124

9.1 Grouping of patches into parts. 130
9.2 Training objects used for cars. 132
9.3 Example codebook clusters with their corresponding patches. 132
9.4 Example subpart groupings (of codebook clusters). 134
9.5 Example part groupings (of codebook clusters) and their main activation areas. 135
9.6 Generic structure of the Bayesian networks used for verifying hypotheses. 136
9.7 Examples of the Bayesian network verification stage. 137

List of Tables

3.1 Recognition Results for the categorization of unknown objects. 23
3.2 Best primary and secondary features for the CogVis-ETH80 categories. 25

4.1 Hierarchical clustering methods, according to the Lance-Willams update scheme. 39

8.1 Comparison of our results on the CalTech motorbike data set with others reported in the literature. 116
8.2 Comparison of our results on the CalTech data set for rear views of cars with others reported in the literature. 120
8.3 EER performances and cross-category confusions for three single-category detectors. 125

Bibliography

S. Agarwal and D. Roth (2002), Learning a Sparse Representation for Object Detection, in *Seventh European Conference on Computer Vision (ECCV'02)*, pp. 113–130.

D. Ballard (1981), Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition*, vol. 13(2), pp. 111–122.

L. Barsalou (1983), Ad-hoc Categories, *Memory and Cognition*, vol. 11, pp. 211–227.

E. Bart, E. Byvatov, and S. Ullman (2004), View-invariant recognition using corresponding object fragments, in *Eigth European Conference on Computer Vision (ECCV'04)*, pp. 152–165.

E. Bart and S. Ullman (2004), Class-based matching of object parts, in *Workshop on Image and Video Registration (WIVR'04)*, Washington, DC.

S. Belongie, J. Malik, and J. Puchiza (2001), Matching Shapes, in *Eigth International Conference on Computer Vision (ICCV'01)*.

S. Belongie, J. Malik, and J. Puchiza (2002), Shape Matching and Object Recognition Using Shape Contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24(4).

J. Bentley (1975), Multidimensional Binary Search Trees Used for Associative Searching, *Communications of the ACM*, vol. 18(9), pp. 509–517.

J. Benzécri (1982), Construction d'une Classification Ascendante Hiérarchique par la Recherche en Chaîne des Voisins Réciproques, *Les Cahiers de l'Analyse des Données*, vol. 7(2), pp. 209–218.

I. Biederman (1987), Recognition by Components: A Theory of Human Image Understanding, *Psychol. Review*, vol. 94, pp. 115–147.

S. Bileschi, B. Leung, and R. Rifkin (2004), Towards Component-Based Car Detection, in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pp. 75–89, Prague, Czech Republic.

T. Binford (1971), Visual perception by computer, in *IEEE Conference on Systems Science and Cybernetics*.

V. Blanz and T. Vetter (2003), Face Recognition Based on Fitting a 3D Morphable Model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(9), pp. 1063–1074.

E. Borenstein, E. Sharon, and S. Ullman (2004), Combining Top-Down and Bottom-Up Segmentations, in *IEEE Workshop on Perceptual Organization in Computer Vision (POCV'04)*, Washington, DC.

E. Borenstein and S. Ullman (2002), Class-Specific, Top-Down Segmentation, in *Seventh European Conference on Computer Vision (ECCV'02)*, LNCS 2353, pp. 109–122.

E. Borenstein and S. Ullman (2004), Learning to Segment, in *Eigth European Conference on Computer Vision (ECCV'04)*.

M. Brown and D. Lowe (2002), Invariant Features from Interest Point Groups, in *British Machine Vision Conference (BMVC'02)*, pp. 656–665, Cardiff, Wales.

R. Brown (1958), How Shall a Thing Be Called?, *Psychological Review*, vol. 65, pp. 14–21.

M. Bruynooghe (1977), Méthodes Nouvelles en Classification Automatique des Données Taxinomiques Nombreuses, *Statistique et Analyse des Données*, vol. 3, pp. 24–42.

M. Burl, M. Weber, and P. Perona (1998), A Probabilistic Approach to Object Recognition using Local Photometry and Global Geometry, in *Fifth European Conference on Computer Vision (ECCV'98)*.

B. Caputo, C. Wallraven, and M. Nilsback (2004), Object Categorization via Local Kernels, in *International Conference on Pattern Recognition (ICPR'04)*.

O. Carmichael and M. Hebert (2003), Shape-Based Recognition of Wiry Objects, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pp. 401–408.

Y. Cheng (1995), Mean shift mode seeking and clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17(8), pp. 790–799.

R. Collins (2003), Mean-Shift Blob Tracking Through Scale Space, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

D. Comaniciu and P. Meer (1999), Distribution Free Decomposition of Multivariate Data, *Pattern Analysis and Applications*, vol. 2(1), pp. 22–30.

D. Comaniciu and P. Meer (2002), Mean Shift: A Robust Approach Toward Feature Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24(5), pp. 603–619.

D. Comaniciu, V. Ramesh, and P. Meer (2001), The Variable Bandwidth Mean Shift and Data-Driven Scale Selection, in *Eigth International Conference on Computer Vision (ICCV'01)*.

T. Cootes, G. Edwards, and C. Taylor (1998), Active Appearance Models, in *Fifth European Conference on Computer Vision (ECCV'98)*.

G. Csurka, C. Dance, L. Fan, J. Willarnowski, and C. Bray (2004), Visual Categorization with Bags of Keypoints, in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pp. 59–74, Prague, Czech Republic.

F. Cutzu and J. Tsotsos (2003), The Selective Tuning Model of Attention: Psychophysical Evidence for a Suppressive Annulus around an Attended Item, *Vision Research*, vol. 43, pp. 205–219.

W. Day and H. Edelsbrunner (1984), Efficient Algorithms For Agglomerative Hierarchical Clustering Methods, *Journal of Classification*, vol. 1, pp. 7–24.

C. de Rham (1980), La Classification Hiérarchique Ascendante Selon la Méthode des Voisins Réciproques, *Les Cahiers de l'Analyse des Données*, vol. 5(2), pp. 135–144.

R. Deriche (1993), Recursively Implementing the Gaussian and its Derivatives, Tech. Rep. TR-1893, INRIA Sophia Antipolis.

G. Dorko and C. Schmid (2003), Selection of Scale Invariant Parts for Object Class Recognition, in *Ninth International Conference on Computer Vision (ICCV'03)*.

R. Duda, P. Hart, and D. Stork (2001), *Pattern Classification*, Wiley, New York, 2nd edn.

S. Edelman (1999), *Representation and Recognition in Vision*, MIT Press.

S. Edelman, B. Hiles, H. Yang, and N. Intrator (2001), Probabilistic Principles in Unsupervised Learning of Visual Structure: Human Data and a Model, in *Neural Information Processing Systems (NIPS'01)*.

S. Edelman and N. Intrator (2004), Unsupervised Statistical Learning in Vision: Computational Principles, Biological Evidence, in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pp. 1–14, Prague, Czech Republic.

P. Felzenszwalb (2001), Learning Models for Object Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*.

P. Felzenszwalb and D. Huttenlocher (2005), Pictorial Structures for Object Recognition, *International Journal of Computer Vision*, vol. 61(1).

R. Fergus, A. Zisserman, and P. Perona (2003), Object Class Recognition by Unsupervised Scale-Invariant Learning, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

V. Ferrari, T. Tuytelaars, and L. van Gool (2004), Simultaneous Recognition and Segmentation by Image Exploration, in *Eigth European Conference on Computer Vision (ECCV'04)*. To appear.

W. Förstner and E. Gülch (1987), A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features., in *ISPRS Intercommission Workshop*, Interlaken.

D. Forsyth and M. Fleck (1997), Body Plans, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 678–683.

J. Friedman, J. Bentley, and R. Finkel (1977), An Algorithm for Finding Best Matches in Logarithmic Expected Time, *IEEE Transactions on Mathematical Software*, vol. 3(3), pp. 209–226.

A. Garg, S. Agarwal, and T. Huang (2002), Fusion of Global and Local Information for Object Detection, in *International Conference on Pattern Recognition (ICPR'02)*, pp. 723–726.

D. Gavrila (1998), Multi-feature Hierarchical Template Matching Using Distance Transforms, in *International Conference on Pattern Recognition (ICPR'98)*, vol. 1, pp. 439–444.

D. Gavrila (2000), Pedestrian Detection from a Moving Vehicle, in *Sixth European Conference on Computer Vision (ECCV'00)*, pp. 37–49.

D. Gavrila and V. Philomin (1999), Real-Time Object Detection for Smart Vehicles, in *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 87–93.

J. Gibson (1957), Survival in a World of Probable Objects, *Contemporary Psychology*, vol. 2, pp. 33–35.

M. Giese and T. Poggio (2000), Morphable Models for the Analysis and Synthesis of Complex Motion Patterns, *International Journal of Computer Vision*, vol. 38(1), pp. 59–73.

R. Gray (1984), Vector Quantization, *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 1(2), pp. 4–29.

E. Grimson (1990), *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA.

D. Hall (2004), A System for Object Class Detection, in H.-H. Nagel and H. Christensen (eds.), *Cognitive Computer Vision. Proceedings of Dagstuhl Seminar 03441*. To appear.

D. Hall and J. Crowley (2003), Computation of Generic Features for Object Classification, in *Scale-Space'03, Isle of Skye, UK*, LNCS 2695, pp. 744–756, Springer.

D. Hall, V. C. de Verdiere, and J. Crowley (2000), Object Recognition Using Colored Receptive Fields, in *Sixth European Conference on Computer Vision (ECCV'00)*, pp. 164–177.

E. Hameiri and I. Shimshoni (2002), Estimating the Principal Curvatures and the Darboux Frame from Real 3D Range Data, in *IEEE Inter. Symp. on 3D Data Proc. Visual. Trans.*, pp. 258–267.

C. Harris and M. Stephens (1988), A combined corner and edge detector, in *Alvey Vision Conference*, pp. 147–151.

B. Heisele, T. Serre, M. Pontil, and T. Poggio (2001), Component-Based Face Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pp. 657–662.

T. Hofmann (1997), *Data Clustering and Beyond – A Deterministic Annealing Framework for Exploratory Data Analysis*, Ph.D. thesis, Universität Bonn.

P. Hough (1962), Method and Means for Recognizing Complex Patterns, U.S. Patent 3069654.

C.-Y. Huang, O. Camps, and T. Kanungo (1997), Object Recognition Using Appearance-Based Parts and Relations, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 877–883.

S. Ioffe and D. Forsyth (2001), Human Tracking with Mixtures of Trees, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*.

A. Jain and R. Dubes (1988), *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs.

M. Jones and T. Poggio (1996), Model-Based Matching by Linear Combinations of Prototypes, MIT AI Memo 1583, MIT.

M. Jones and T. Poggio (1998a), Multidimensional morphable models, in *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 683–688.

M. Jones and T. Poggio (1998b), Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes, *International Journal of Computer Vision*, vol. 29(2), pp. 107–131.

F. Jurie and C.Schmid (2004), Scale-invariant shape features for recognition of object categories, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

T. Kadir and M. Brady (2001), Scale, Saliency, and Image Description, *International Journal of Computer Vision*, vol. 45(2), pp. 83–105.

T. Kadir, A. Zisserman, and M. Brady (2004), An affine invariant salient region detector, in *Eigth European Conference on Computer Vision (ECCV'04)*.

H. Kruppa (2004), *Object Detection Using Scale-Specific Boosted Parts and a Bayesian Combiner*, Ph.D. thesis, ETH Zurich.

G. Lakoff (1987), *Women, Fire, and Dangerous Things – What Categories Reveal about the Mind*, Univ. of Chicago Press, Chicago.

Y. Lamdan, J. Schwartz, and H. Wolfson (1988), Object Recognition by Affine Invariant Matching, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'88)*, pp. 335–344.

Y. Lamdan and H. Wolfson (1988), Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, in *Second International Conference on Computer Vision (ICCV'88)*, pp. 238–249.

G. Lance and W. Williams (1967), A General Theory of Classificatory Sorting Strategies: II. Clustering Systems, *Computer Journal*, vol. 10, pp. 271–277.

B. Leibe, A. Leonardis, and B. Schiele (2004), Combined Object Categorization and Segmentation with an Implicit Shape Model, in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pp. 17–32, Prague, Czech Republic.

B. Leibe and B. Schiele (2003a), Analyzing Appearance and Contour Based Methods for Object Categorization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI.

B. Leibe and B. Schiele (2003b), Interleaved Object Categorization and Segmentation, in *British Machine Vision Conference (BMVC'03)*, pp. 759–768, Norwich, UK.

B. Leibe and B. Schiele (2004), Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search, in *DAGM'04 Annual Pattern Recognition Symposium*, Springer LNCS, Vol. 3175, pp. 145–153, Tuebingen, Germany.

A. Leonardis, H. Bischof, and J. Maver (2002), Multiple Eigenspaces, *Pattern Recognition*, vol. 35(11), pp. 2613–2627.

A. Leonardis, A. Gupta, and R. Bajcsy (1995), Segmentation of Range Images as the Search for Geometric Parametric Models, *International Journal of Computer Vision*, vol. 14, pp. 253–277.

F.-F. Li (2004), Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories, in *Workshop on Generative-Model Based Vision (GMBV'04)*, Washington, DC.

F.-F. Li, R. Fergus, and P. Perona (2003), A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories, in *Ninth International Conference on Computer Vision (ICCV'03)*.

R. Lienhart, A. Kuranov, and V. Pisarevsky (2003), Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, in *DAGM'03 Annual Pattern Recognition Symposium*, pp. 297–304, Magdeburg, Germany.

T. Lindeberg (1994), Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales, *Journal of Applied Statistics*, vol. 21(2), pp. 224–270.

T. Lindeberg (1998), Feature Detection with Automatic Scale Selection, *International Journal of Computer Vision*, vol. 30(2), pp. 79–116.

H. Loos and C. v.d. Malsburg (2002), 1-Click Learning of Object Models for Recognition, in *2nd International Workshop on Biologically Motivated Computer Vision (BMCV'02)*, LNCS 2525, pp. 377–386, Springer, Berlin.

D. Lowe (1999), Object Recognition from Local Scale Invariant Features, in *Seventh International Conference on Computer Vision (ICCV'99)*.

D. Lowe (2001), Local Feature View Clustering for 3D Object Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*.

D. Lowe (2004), Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, vol. 60(2), pp. 91–110.

J. MacQueen (1967), Some Methods for Classification and Analysis of Multivariate Observations, in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

D. Macrini, A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker (2002), View-Based 3-D Object Recognition using Shock Graphs, in *International Conference on Pattern Recognition (ICPR'02)*.

D. Magee and R. Boyle (2002), Detecting Lameness using 'Re-sampling condensation' and 'Multi-stream Cyclic Hidden Markov Models', *Image and Vision Computing*, vol. 20(8), pp. 581–594.

J. Malik, S. Belongie, T. Leung, and J. Shi (2001), Contour and Texture Analysis for Image Segmentation, *International Journal of Computer Vision*, vol. 43(1), pp. 7–27.

D. Marr (1982), *Vision*, W.H. Freeman, San Francisco.

J. Matas, O. Chum, U. Martin, and T. Pajdla (2002), Robust wide baseline stereo from maximally stable extremal regions, in *British Machine Vision Conference (BMVC'02)*, pp. 384–393.

B. Mel (1996), SEEMORE: Combining Color, Shape, and Texture Histogramming in a Neurally Inspired Approach to Visual Object Recognition, in *International Conference on Pattern Recognition (ICPR'96)*.

C. Mikolajczyk, C. Schmid, and A. Zisserman (2004), Human Detection Based on a Probabilistic Assembly of Robust Part Detectors, in *Eigth European Conference on Computer Vision (ECCV'04)*, LNCS 3021, pp. 69–82, Springer.

K. Mikolajczyk and C. Schmid (2001), Indexing based on Scale Invariant Interest Points, in *Eigth International Conference on Computer Vision (ICCV'01)*, pp. 525–531.

K. Mikolajczyk and C. Schmid (2002), An Affine Invariant Interest Point Detector, in *Seventh European Conference on Computer Vision (ECCV'02)*, pp. 128–142.

K. Mikolajczyk and C. Schmid (2003), A Performance Evaluation of Local Descriptors, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

K. Mikolajczyk, A. Zisserman, and C. Schmid (2003), Shape Recognition with Edge-Based Features, in *British Machine Vision Conference (BMVC'03)*, pp. 779–788.

A. Mohan, C. Papageorgiou, and T. Poggio (2001), Example-based Object Detection in Images by Components, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(4), pp. 349–361.

H. Murase and S. Nayar (1995), Visual Learning and Recognition of 3D Objects from Appearance, *International Journal of Computer Vision*, vol. 14, pp. 5–24.

A. Needham (2001), Object Recognition and Object Segregation in 4.5-month-old infants, *Journal of Experimental Child Psychology*, vol. 78(3), pp. 3–24.

R. Nelson and A. Selinger (1998a), A Cubist Approach to Object Recognition, in *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 614–621.

R. Nelson and A. Selinger (1998b), Large-Scale Tests of a Keyed, Appearance-Based 3-D Object Recognition System, *Vision Research*, vol. 38(15).

S. Nene and S. Nayar (1997), A Simple Algorithm for Nearest Neighbour Search in High Dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17.

M. Nilsback and B. Caputo (2004), Cue Integration through Discriminative Accumulation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

S. Obdrzalek and J. Matas (2002), Object Recognition Using Local Affine Frames on Distinguished Regions, in *British Machine Vision Conference (BMVC'02)*.

A. Opelt, M. Fussenegger, A. Pinz, and P. Auer (2004), Weak Hypotheses and Boosting for Generic Object Detection and Recognition, in *Eigth European Conference on Computer Vision (ECCV'04)*.

C. Papageorgiou and T. Poggio (2000), A Trainable System for Object Detection, *International Journal of Computer Vision*, vol. 38(1), pp. 15–33.

M. Peternel and A. Leonardis (2004), Visual Learning and Recognition of a Probabilistic Spatio-Temporal Model of Cyclic Human Locomotion, in *9th Computer Vision Winter Workshop (CVWW'04)*, Piran, Slowenia.

M. Peterson (1994), Object Recognition Processes Can and Do Operate before Figure-Ground Organization, *Current Directions in Psychological Science*, vol. 3, pp. 105–111.

T. Pham, M. Worring, and A. Smeulders (2002), Face Detection by Aggregated Bayesian Network Classifiers, *Pattern Recognition Letters*, vol. 23(4), pp. 451–461.

R. Rao and D. Ballard (1995), Object Indexing Using an Iconic Sparse Distributed Memory, in *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 24–31.

R. Rimey and C. Brown (1994), Control of Selective Perception Using Bayes Nets and Decision Theory, *International Journal of Computer Vision*, vol. 12(2/3), pp. 173–207.

L. Roberts (1963), *Machine Perception of 3D Solids*, Ph.D. thesis, MIT Cambridge, MA.

T. Robinson (1981), The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes, *ACM SIGMOD*, pp. 10–18.

R. Ronfard, C. Schmid, and B. Triggs (2002), Learning to Parse Pictures of People, in *Seventh European Conference on Computer Vision (ECCV'02)*, pp. 700–714.

E. Rosch, C. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem (1976), Basic Objects in Natural Categories, *Cognitive Psychology*, vol. 8, pp. 382–439.

F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce (2003), 3D Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

S. Roweis (1997), EM Algorithms for PCA and SPCA, in *Neural Information Processing Systems (NIPS'97)*, pp. 626–632.

H. Rowley, S. Baluja, and T. Kanade (1998), Neural Network-based Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(1), pp. 23–38.

F. Schaffalitzky and A. Zisserman (2002), Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?", in *Seventh European Conference on Computer Vision (ECCV'02)*, pp. 414–431.

B. Schiele and J. Crowley (2000), Recognition without Correspondence using Multidimensional Receptive Field Histograms, *International Journal of Computer Vision*, vol. 36(1), pp. 31–52.

C. Schmid and R. Mohr (1996), Combining Greyvalue Invariants with Local Constraints for Object Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96)*.

C. Schmid, R. Mohr, and C. Bauckhage (1998), Comparing and Evaluating Interest Points, in *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 230–235.

C. Schmid, R. Mohr, and C. Bauckhage (2000), Evaluation of Interest Point Detectors, *International Journal of Computer Vision*, vol. 37(2), pp. 151–172.

H. Schneiderman (2004), Feature-Centric Evaluation for Efficient Cascaded Object Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

H. Schneiderman and T. Kanade (2000), A Statistical Method of 3D Object Detection Applied to Faces and Cars, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pp. 746–751.

H. Schneiderman and T. Kanade (2004), Object Detection Using the Statistics of Parts, *International Journal of Computer Vision*, vol. 56(3), pp. 151–177.

S. Sclaroff (1997), Deformable Prototypes for Encoding Shape Categories in Image Databases, *Pattern Recognition*, vol. 30(4).

E. Sharon, A. Brandt, and R. Basri (2000), Fast Multiscale Image Segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pp. 70–77.

J. Shi and J. Malik (1997), Normalized Cuts and Image Segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 731–737.

A. Shokoufandeh, Y. Keselman, F. Demirci, D. Macrini, and S. Dickinson (2004), Many-to-Many Feature Matching in Object Recognition, in H.-H. Nagel and H. Christensen (eds.), *Cognitive Computer Vision. Proceedings of Dagstuhl Seminar 03441*. To appear.

K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker (1999), Shock Graphs and Shape Matching, *International Journal of Computer Vision*, vol. 30, pp. 1–24.

J. Sivic, F. Schaffalitzky, and A. Zisserman (2004), Object Level Grouping for Video Shots, in *ECCV04*.

J. Sivic and A. Zisserman (2003), Video Google: A Text Retrieval Approach to Object Matching in Videos, in *Ninth International Conference on Computer Vision (ICCV'03)*.

J. Sivic and A. Zisserman (2004), Video Data Mining Using Configurations of Viewpoint Invariant Regions, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

C. Stauffer and W. Grimson (1999), Adaptive background mixture models for real-time tracking, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 246–252.

C. Stone (1982), Optimal Global Rates of Convergence for Nonparametric Regression, *Annals of Statistics*, vol. 10, pp. 1040–1053.

J. Sullivan and S. Carlsson (2002), Recognizing and Tracking Human Action, in *Seventh European Conference on Computer Vision (ECCV'02)*, pp. 629–644, Springer Verlag.

M. Swain and D. Ballard (1991), Color Indexing, *International Journal of Computer Vision*, vol. 7(1), pp. 11–32.

J. Thureson and S. Carlsson (2004), Appearance Based Qualitative Image Description for Object Class Recognition, in *Eigth European Conference on Computer Vision (ECCV'04)*.

A. Torralba (2003), Contextual Priming for Object Detection, *International Journal of Computer Vision*, vol. 53(2), pp. 153–167.

A. Torralba, K. Murphy, and W. Freeman (2004), Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

A. Torralba and P. Sinha (2001), Statistical Context Priming for Object Detection, in *Eigth International Conference on Computer Vision (ICCV'01)*, pp. 763–770.

B. Triggs (2004), Detecting Keypoints with Stable Position, Orientation and Scale under Illumination Changes, in *Eigth European Conference on Computer Vision (ECCV'04)*.

J. Tsotsos (1990), Analyzing Vision at the Complexity Level, *Behavioral and Brain Sciences*, vol. 13, pp. 423–469.

J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo (1995), Modeling Visual Attention via Selective Tuning, *Artificial Intelligence*, vol. 78, pp. 507–545.

M. Turk and A. Pentland (1991), Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86.

T. Tuytelaars and L. van Gool (2000), Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions, in *British Machine Vision Conference (BMVC'00)*, pp. 412–422, Bristol, UK.

T. Tuytelaars and L. van Gool (2004), Matching Widely Separated Views based on Affinely Invariant Nighbourhoods, *International Journal of Computer Vision*, vol. 59(1), pp. 61–85.

S. Ullman (1998), Three-Dimensional Object Recognition based on the Combination of Views, *Cognition*, vol. 67(1), pp. 21–44.

S. Ullman, M. Vidal-Naquet, and E. Sali (2002), Visual features of intermediate complexity and their use in classification, *Nature Neuroscience*, vol. 5(7), pp. 682–687.

S. Vecera and R. O'Reilly (1998), Figure-Ground Organization and Object Recognition Processes: An Interactive Account, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 24(2), pp. 441–462.

M. Vidal-Naquet and S. Ullman (2003), Object Recognition with Informative Features and Linear Classification, in *Ninth International Conference on Computer Vision (ICCV'03)*.

P. Viola and M. Jones (2001), Rapid Object Detection Using a Boosted Cascade of Simple Features, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pp. 511–518.

P. Viola and M. Jones (2004), Robust Real-Time Face Detection, *International Journal of Computer Vision*, vol. 57(2), pp. 137–154.

P. Viola, M. Jones, and D. Snow (2003), Detecting pedestrians using patterns of motion and appearance, in *Ninth International Conference on Computer Vision (ICCV'03)*, pp. 734–741.

J. Vogel (2004), *Semantic Scene Modeling and Retrieval*, Ph.D. thesis, ETH Zürich, Switzerland.

E. Voorhees (1986), Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval, *Information Processing & Management*, vol. 22(6), pp. 465–476.

C. Wallraven, B. Caputo, and A. Graf (2003), Recognition with Local Features: the Kernel Recipe, in *Ninth International Conference on Computer Vision (ICCV'03)*.

J. Ward (1963), Hierarchical Grouping to Optimize an Objective Function, *Journal of the American Statistical Association*, vol. 58, pp. 236–244.

M. Weber (2000), *Unsupervised Learning of Models for Object Recognition*, Ph.D. thesis, California Institute of Technology, Pasadena, CA.

M. Weber, M. Welling, and P. Perona (2000a), Towards Automatic Discovery of Object Categories, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*.

M. Weber, M. Welling, and P. Perona (2000b), Unsupervised learning of object models for recognition, in *Sixth European Conference on Computer Vision (ECCV'00)*.

E. Weisstein (), Hypersphere, From MathWorld–A Wolfram Web Resource. Http://mathworld.wolfram.com/Hypersphere.html.

L. Wiskott, J. Fellous, N. Krueger, and C. von der Malsburg (1997), Face Recognition by Elastic Bunch Graph Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(7), pp. 775–779.

A. Witkin (1983), Scale-Space Filtering, in *International Joint Conference on Artificial Intelligence*, pp. 1019–1022, Karlsruhe, Germany.

H. Wolfson (1990), Model-Based Object Recognition by Geometric Hashing, in *First European Conference on Computer Vision (ECCV'90)*, LNCS, 427, pp. 526–536, Springer-Verlag, Berlin.

K. Yow and R. Cipolla (1997), Feature-Based Human Face Detection, *Image and Vision Computing*, vol. 15(9), pp. 713–735.

S. Yu and J. Shi (2003), Object-Specific Figure-Ground Segregation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

A. Yuille, D. Cohen, and P. Hallinan (1989), Feature Extraction from Faces Using Deformable Templates, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'89)*.

Curriculum Vitae

Bastian Leibe

Date of birth:	April 23, 1975
Place of birth:	Waiblingen, Germany
Citizenship:	German

Education:	2001–2004	Doctoral student at the Swiss Federal Institute of Technology (ETH) Zurich, Perceptual Computing and Computer Vision Group.
	1998–1999	Studies of Computer Science, Georgia Institute of Technology, Atlanta, Georgia, USA. Graduation with the degree *M.Sc. in Computer Science*.
	1995–2001	Studies of Computer Science, University of Stuttgart, Germany. Graduation with the degree *Dipl.Inform.*.
	1981–1994	Primary School and High School in Stuttgart, Germany.

Professions:	2001–2004	Research and Teaching Assistant, Perceptual Computing and Computer Vision Group, ETH Zurich.
	1999–2000	Teaching Assistant, Institute for Computer Science, University of Stuttgart.
	1994–1998	Part-time Employment, unicomputer gmbh, Stuttgart.
	1994–1995	Civil Service, Neckartalwerkstätten, Stuttgart.
