Learning the Compositional Nature of Objects for Visual Recognition

Author(s):
Ommer, Björn

Publication Date:
2007

Permanent Link:
https://doi.org/10.3929/ethz-a-005506634

Rights / License:
In Copyright - Non-Commercial Use Permitted
Learning the Compositional Nature of Objects for Visual Recognition

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by

BJÖRN OMMER

Dipl. Inf., University of Bonn
born October 28, 1979
citizen of Germany

accepted on the recommendation of

Prof. Dr. Joachim M. Buhmann, ETH Zurich, examiner
Prof. Dr. Jitendra Malik, University of California at Berkeley, co-examiner
Prof. Dr. Luc Van Gool, ETH Zurich and University of Leuven, co-examiner

2007
Abstract

This thesis deals with the recognition of object categories in novel visual scenes, be it images or videos. The research is governed by the main hypothesis that structured, hierarchical object representations can be learned in a fully automatic mode. For this purpose, statistical models are devised that can be learned for each category in an unsupervised manner without requiring any localization information.

Learning object representations for detection and recognition poses one of the key challenges of computer vision. This learning problem becomes especially complex and difficult in the limit of unconstrained scenes, large intra-class variations, and weak supervision during training. A central concept for learning object models despite these problems is to exploit the compositional nature of our (visual) world. The compositional nature of visual objects significantly limits their representation complexity and renders learning of structured object models tractable.

In this thesis methods for learning the compositional structure of objects are investigated and an integration in a category-level object recognition system is presented. The approach learns compositions of parts with characteristic relations between them for each category in an unsupervised manner, requiring neither hand segmentations nor object localization during training. Object representations are based on a shared vocabulary of atomic parts that are founded on a new feature detector with favorably low dimensionality and robustness to within-class variations. Moreover, a Bayesian network is presented that comprises all the compositional constituents together with scene context and object shape. Object recognition is then formulated as a statistical inference problem in this graphical model and it yields competitive performance compared to the state of the art.

Furthermore, the approach is successfully extended to near real-time analysis of videos where category-level object recognition, segmentation, and tracking of multiple objects is performed. This investigation shows how the key concept of compositionality can actually be exploited for both, making learning feasible and rendering recognition computationally tractable.
Zusammenfassung

Diese Dissertation befasst sich mit der Erkennung von Objektkategorien in zuvor unbekannten visuellen Szenen, wobei es sich dabei sowohl um Bilder als auch um Videos handeln kann. Die Leithypothese dieser Forschung ist, dass strukturierte, hierarchische Objektrepräsentationen vollautomatisch gelernt werden können. Zu diesem Zweck werden statistische Modelle entwickelt, die für jede Kategorie unüberwacht trainiert werden können, so dass keinerlei Lokalisierungsinformation benötigt wird.


Acknowledgments

My sincere thanks go to all people that have contributed in various ways to the successful completion of this thesis.

First, I would like to thank my advisor Joachim Buhmann for supervising the thesis and for creating a unique research atmosphere in his group. I am particularly grateful that he has given me the opportunity to pursue my studies at ETH Zurich. Also I would like to thank Luc Van Gool and Jitendra Malik for agreeing to be co-examiners of this thesis despite numerous other obligations. Furthermore, I have appreciated the discussions with Ulrich Ramacher and his group from Infineon Technologies at the beginning of this thesis.

Thanks go to all members of the machine learning group of Joachim for valuable research related discussions and for the numerous activities apart from work. These people have made ETH a fun and exciting place to work.

I would like to thank my numerous master thesis and semester project students for contributing to this thesis with their work and for giving me the opportunity to exercise additional skills in supervising them. Working together and assisting them has been very rewarding for me.

I am particularly grateful to all my friends. Thanks for making Switzerland such a wonderful place to be and for making me feel home right away. These friendships and the amazingly beautiful nature of this country with its opportunities for all different kinds of sports have made the last years invaluable for me.

Finally, special thanks go to my family for their love and heartfelt support.
# Contents

## 1 Introduction

1.1 Structured Object Models ................................. 1
1.2 Computer Vision ........................................... 2
1.3 Objectives .................................................. 5
1.4 Current Challenges and Properties of the Proposed Approach ........................................... 6
1.5 Original Contributions ...................................... 11
1.6 Organization of the Thesis ................................. 12

## 2 Key Modeling Decisions for a Visual Recognition System and State of the Art .......................... 15

2.1 Salient Interest Points and Local Image Descriptors ......................................................... 16
  2.1.1 Interest Point Detection ................................ 16
  2.1.2 Local Image Descriptors ............................... 17
2.2 Object Models and Representation Schemes ................................................................. 19
  2.2.1 Model-Based Vision and View-Based Approaches: An Overview ................................ 20
  2.2.2 Towards Generic Object Recognition .............................. 22
2.3 Hierarchical Object Models and Compositionality ...................................................... 32
  2.3.1 The Need for Hierarchical Representations and its Pitfalls ................................ 33
  2.3.2 The Principle of Compositionality .......................... 34
  2.3.3 Perceptual Organization: Retrieving Relevant Compositions ................................... 36
  2.3.4 Summary of the Characteristics of a Compositional Approach .................................. 39

## 2.4 Different Learning Paradigms .............................................. 40

## 2.5 Graphical Models .................................................. 42
  2.5.1 Graph Theoretical Fundamentals ................................ 42
  2.5.2 Undirected Graphical Models ................................ 43
  2.5.3 Directed Bayesian Networks .................................. 44
2.5.4 Learning and Inference in Graphical Models 44

3 Atomic Parts for Compositionality: Localized Feature Histograms 47
   3.1 Requirements on Local Descriptors 47
   3.2 Deriving a Descriptor for Atomic Parts of a Composition System 50
      3.2.1 Canny Edge Detection 51
      3.2.2 Localized Feature Histograms 52
   3.3 A Codebook of Atomic Compositional Parts 54
   3.4 A Recognition System based on Tuple Compositions 55
      3.4.1 Overview over the Processing Pipeline 56
      3.4.2 Forming Tuple Compositions 56
      3.4.3 Modeling Object Shape 58
      3.4.4 Inference of Image Categorization 61
   3.5 Evaluation of Tuple Compositions 63

4 Sharing Compositions among Categories 67
   4.1 Overview over the Processing Pipeline 68
   4.2 Establishing and Representing Candidate Compositions 70
   4.3 Kernel Methods for Multi-Class Classification 71
      4.3.1 Support Vector Machines 73
      4.3.2 Nonlinear Kernel Discriminant Analysis 75
      4.3.3 Solving Multi-Class Problems by Pairwise Coupling 78
   4.4 Learning Relevant Compositions 80
   4.5 A Compositional Shape Model for Binding Compositions 83
      4.5.1 Modeling Objects and Scene Context 83
      4.5.2 Object Localization 84
      4.5.3 Evaluation of the Localization Performance 84
      4.5.4 Object Classification with a Compositional Shape Model 84
   4.6 Experiments 87
      4.6.1 The Caltech-101 Image Database 87
      4.6.2 Baseline Performance without Compositions 89
      4.6.3 Categorization Performance of the Compositional Model 90
      4.6.4 Evaluating Compositions 91

5 Establishing Deep Compositional Hierarchies 95
5.1 Outline of the Approach .............................................. 96
5.2 Perceptual Bottom-Up Grouping ..................................... 98
5.3 Forming Robust Descriptors for Salient Curves ..................... 101
5.4 Learning of Top-Down Grouping .................................... 101
   5.4.1 Salient Base Compositions ..................................... 102
   5.4.2 Approximating Grouping Probabilities Using Initial Groupings
       of Base Compositions ............................................. 103
   5.4.3 Forming a Compositional Hierarchy ............................ 103
   5.4.4 Local Maxima of the Compositional Hierarchy ................. 104
5.5 Recognition based on the Learned Grouping Model ................. 104
   5.5.1 Applying Top-Down Grouping in the Recognition Phase .. 104
   5.5.2 Shape Model for Binding Compositions ....................... 105
5.6 Evaluation ............................................................ 105
   5.6.1 A Baseline Model without Compositionality ................. 105
   5.6.2 Evaluation of the Model based on Deep Compositional Hier-
       archies .............................................................. 106
   5.6.3 Class Hierarchies for Analyzing Categorization ............ 107

6 Combining Segmentation with Recognition ................................................. 109
6.1 Ensembles of Low-level Segmentations .............................. 110
6.2 Experimental Setup ................................................... 111
6.3 Identifying Object Segments ......................................... 112
   6.3.1 Boundary Extraction and String Representation ............. 113
   6.3.2 String Alignments .............................................. 114
   6.3.3 Detecting Foreground Segments ................................ 115
6.4 Object Recognition using Object Boundary Fragments .......... 117
   6.4.1 Direct Approach ................................................ 118
   6.4.2 Combined Approach: Compositional Hierarchies of Object Seg-
       ments .............................................................. 118

7 Learning to Establish Relevant Compositions .............................. 121
7.1 Outline of the Approach .............................................. 122
7.2 Atomic Parts on Multiple Scales ..................................... 124
7.3 Composition Candidates ................................................. 126
7.4 Learning Relevant Compositions of Parts ........................... 127
7.4.1 A Bayesian Criterion for Compositional Relevance . . . . . . . 127
7.4.2 A Statistically Feasible Learning Algorithm for Compositional
Relevance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128
7.4.3 Visualizing Relevant Compositions . . . . . . . . . . . . . . . 129
7.5 Learning Higher Order compositions . . . . . . . . . . . . . . . 130
7.6 Binding Compositions in a Compositional Shape Model . . . . . . . 131
7.6.1 Selecting Relevant Compositions . . . . . . . . . . . . . . . . 132
7.6.2 Object Recognition using Statistical Inference . . . . . . . . . 133
7.7 Evaluation of the Compositional Approach . . . . . . . . . . . . . . 135
7.7.1 Results on Caltech-101 . . . . . . . . . . . . . . . . . . . . . . 135
7.7.2 A Comparison of Feature Descriptors for Atomic Parts . . . . 136
7.7.3 Influence of Color in Localized Feature Histograms . . . . . . 139
7.7.4 Sampling a Compositional Representation from the Generative
Object Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
7.7.5 Inferring Missing Object Components . . . . . . . . . . . . . . 141
7.7.6 Towards Learning Category-level Segmentation . . . . . . . . 142

8 Interleaved Learning of Localization and Classification 145
8.1 Extending the Compositional Recognition Algorithm . . . . . . . . 146
8.2 Learning Object Models from Cluttered, Unsegmented Images . . 148
8.3 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
8.3.1 Classification Performance . . . . . . . . . . . . . . . . . . . . 152
8.3.2 Detection Performance . . . . . . . . . . . . . . . . . . . . . . 155
8.3.3 Analyzing the Performance of the Compositional Model . . . . 157

9 Recognition, Segmentation, and Tracking of Multiple Objects in
Video 159
9.1 Crucial Components of Video Retrieval Systems . . . . . . . . . . 160
9.2 Outline of the Video Composition System . . . . . . . . . . . . . . 162
9.3 Estimation of Optical Flow . . . . . . . . . . . . . . . . . . . . . . 163
9.4 “Atoms” of the Compositional Representation . . . . . . . . . . . 165
9.4.1 Codebook-Based Representation of Atomic Parts . . . . . . . 166
9.4.2 Descriptors for Local Parts . . . . . . . . . . . . . . . . . . . . 166
9.5 Establishing Dynamic Compositions . . . . . . . . . . . . . . . . 167
9.6 Grouping Compositions over Time . . . . . . . . . . . . . . . . . 168
## Contents

9.7 Segmentation Hypotheses for Multiple Objects ........................................... 170
9.8 Model Selection for Identifying Reliable Segmentation Hypotheses ............ 172
9.9 Learning Relevant Compositions ................................................................. 173
9.10 Object Recognition Using the Compositional Shape Model .......................... 174
9.11 Experiments ............................................................................................... 175
  9.11.1 Evaluating the Different Building Blocks of the Composition System ................. 176
  9.11.2 Multi-Object Recognition ...................................................................... 178
  9.11.3 Analyzing the Contributions of Individual Compositions ...................... 178

10 Discussion ...................................................................................................... 181

Bibliography ........................................................................................................ 183

Index .................................................................................................................... 207
Introduction

The mind is not a vessel to be filled, but a fire to be kindled.
—Plutarch, Moralia

1.1 Structured Object Models

Although this world might seem rather chaotic at times it nevertheless exhibits an astounding degree of structure. No matter at what scale we survey our universe, order and regularity are apparent everywhere. On a large scale, orderless clouds of dense matter condense due to gravitational attraction to form stars, stellar systems, and eventually galaxies consisting of billions of stars. On a scale that is directly accessible with our own eyes we can, for instance, observe the complex, ordered patterns exhibited by animals, plants, and non-living matter on this planet. Examples are the symmetry, or even the self-similarity featured by ferns, sea stars, or snowflakes. Finally, on an again much smaller scale, the highly complex DNA molecule controls the development, functioning, and eventually the structure of all living organisms. In the temporal domain we can observe periodical structures such as the hydrologic cycle, the ever repeating seasons of the year, or our own heartbeat.

Considering the second law of thermodynamics, which states that the entropy (the degree of “disorder”) of an isolated system—the universe in the most general case—is monotonically increasing, it is astonishing that such complex, highly ordered structures exist. However, not only the existence of order and structure is astounding but also the robustness is striking which such structures exhibit with respect to
disrupting factors.

Besides its necessity for the existence of life, structure and order play also an important role in human thinking. Man has always been striving for a limited set of simple rules, laws, or relationships that, together with some simple physical entities, would explain complex entities and thereby make the world comprehensible. A prominent example is Isaac Newton’s discovery of universal gravitation [Kee98]. It based on the supposition that the same gravitational force that is pulling objects such as an apple a few meters towards the earth is also valid outside earth and holding the moon in its orbit. This idea resulted in the few laws of classical mechanics that suffice to explain the motion of macroscopic objects anywhere in the universe given some basic premises hold.

When seeking comparably simple representation schemes for complex real-world concepts the human mind again and again exploits the high degree of structure and regularity of our world [Bie87]—the fact that objects are composed of simpler parts with certain characteristic relationships between them, as opposed to random configurations. Such compositional models for real-world objects are based on only few atomic entities and relations between them. Therefore, a small set of fairly simple atoms and rules suffices to build complex representations by recursively reducing them to simpler substructures.

Object recognition, i.e. the detection and classification of a novel object instance, requires some kind of representation—a model—of each potential object class. These models are to be learned from a limited amount of training data and recognition requires to compare a novel instance with all of the previously learned models and to find the most appropriate class. To facilitate robust model learning from few training observations it is favorable to exploit the high degree of structure and regularity of our world. Thereby redundancies and invariance properties can be incorporated for simplifying object representations and, in effect, the learning problem is rendered feasible. To this end compositional representations are favorable as they divide model learning into two easier subtasks: one of learning generic representational atoms and the other of learning characteristic relations between parts that yield object specific compositions.

1.2 Computer Vision

Computer vision pursues the goal to devise algorithms and hardware that enable machines to see and, eventually, understand the visually perceivable world. This
ambitious challenge of automatic image understanding is one of the key problems of artificial intelligence. It is striking that although computers are outperforming human beings in many complex tasks by far, they do not even come close to human performance when it comes to vision. This observation is even more astounding considering the incredible ease with which we are able to understand visual scenes. However, a closer look reveals that computers are mostly superior in those fields where exact, complex calculations are crucial or where large amounts of data have to be stored and processed accurately. In contrast to this, vision requires the processing of a distorted, inaccurate sensory representation of a scene and it relies heavily on vague concepts and large amounts of rather fuzzy side information coming from prior knowledge.

A successful computer vision system requires a number of subtasks to be solved. The initial task is that of image acquisition. Over the last couple of years this field has made significant progress so that nowadays digital cameras and other related equipment is available to the mass market. Second, with the increase of computing power, image and even video processing have become feasible on standard hardware. Image processing is a mandatory prerequisite for other processing steps and deals with image-to-image transformations, not with explicit model building.

**Recognition:** Moving on in the list of subtasks we now come to the key part of vision, to *visual object recognition* which also constitutes a central topic of this thesis. Object recognition can be divided into two closely related subtasks, namely *detection* and *classification*. The former is about answering a “where” question, i.e. it deals with localizing the object position and its extend. The latter one answers a “what” question by labeling an image or an image region as containing an instance of a certain class. In the literature the terms *class* and *category* are often used interchangeably. However, sometimes the latter term is used to emphasize that the object classes are defined on a coarse semantic level (e.g. “shoe” as opposed to “tennis shoe for indoor tennis courts”). *Image categorization* does then refer to the task of deciding what category a whole image belongs two. Therefore, the image is labeled according to its most prominent object. There are also other categorization settings conceivable such as action recognition where an image or video sequence has to be labeled as featuring one of several possible actions. However, this thesis deals with object recognition and, therefore, all terms are used accordingly.

To be able to automatically recognize objects from visual scenes a machine needs some kind of concept of each potential category—a representation. Such a representation is a model of a real world class that reflects the characteristic properties which
distinguish it from other classes. The models need not be completely accurate representations of real world objects (e.g. 3-D models of their physical structure). It is often appropriate to restrict representations so that they only capture the category-discriminative characteristics of objects. However, generative models do also provide for a reconstruction of the initial visual stimulus from the model given values for its free parameters. Therefore, they can be applied to infer missing data and they are open to a deeper analysis of the learned object structures than discriminative models. In conclusion, object classification is a typical pattern recognition problem and it requires models for the different classes.

**Learning:** Now we come to the next problem in our list of subtasks—to the one of learning object models from training data. Certainly, we could require a human supervisor to instruct the computer how specific object classes can be distinguished (for example by selecting some characteristic features or even by designing abstract object models that can be directly used in a classification algorithm). This manual design process is however a tedious, costly task that becomes impractical when the complexity of the recognition problem increases beyond a few classes. Therefore, *machine learning* techniques have to be applied to attack the learning problem which is another key challenge of computer vision.

Furthermore, one has to find a representation scheme that can be used to formalize object models. To this end, statistical models are appealing. As pointed out above, it is also crucial to limited the complexity of representations. For that purpose structured models such as compositional ones are beneficial.

Now we come to final subtask of image understanding—to the problem of reasoning about a scene. Once all constituent objects of a scene have been detected and classified a computer is able to reason about the scene and solve higher level tasks that are depending on the individual application. An autonomous vehicle could for instance detect the presence of pedestrians on the street and plan a lane change maneuver to avoid a collision.

In conclusion, computer vision deals with the inverse problem of computer graphics. As opposed to generating visual representations of scenes given object models, computer vision is all about learning object models to enable a computer to recognize objects in novel scenes.
1.3 Objectives

The goal of this thesis is to study and devise algorithms for automatic learning of structured representations of arbitrary object categories. Particularly, a category-level recognition system is designed that detects and classifies visual objects in images and videos.

On the learning side, the aim is to devise algorithms that automatically discover appropriate models for object categories despite the large within-class variations that they exhibit. This task is aggravated when only few training samples are available and when supervision information is lacking (i.e. object localizations are not provided) as it is the case in this thesis. The availability of only limited training information is in most practical scenarios due to the fact that collecting samples is costly and user supervision is tedious. Especially when there are many object categories to be learned, requiring intense user supervision becomes prohibitive. Therefore, the deficiencies in training data have to be compensated so that learning complex object models becomes feasible. To this end the regularities and structure of our world have to be exploited. As Attneave [Att54] points out, the visual stimulus is highly redundant in the sense that it exhibits significant spatial and temporal interdepen-
Chapter 1. Introduction

Dependencies. In other words, the regularity makes portions of the visual field become predictable given other parts. *Compositionality* (e.g. [GPC02]) is a general principle in cognition and especially in human vision [Bie87] and language that exploits these dependencies. A representational system is compositional if and only if each complex representation is determined by its constituent parts and the relationships between them (cf. [WMS05]). Consequently, a compositional representation decomposes complex models into simpler ones using the relations between them, finally resulting in a hierarchy of recursive compositions. It is eventually based on a small set of simple, atomic parts that are generic and can be used for representing various objects, that is they are shared by all classes. Consider Figure 1.1 for a schematic illustration of this concept. The generic geometric primitives shown in a) can be used to represent various objects, however by themselves they provide only very little information about any specific object. In c) a specific spatial relationship is established between these parts and, as a result, the same unspecific parts as in a) now represent a specific street sign. Consequently we can summarize that compositions bridge the semantic gap between complex objects and the low level percepts (e.g. individual photoreceptive cells on the retina or, equivalently, pixels of a digital camera) by establishing intermediate hidden layer representations. Therefore, complex object models are decomposed into a hierarchy of simpler models so that learning those substructures becomes a feasible problem.

This thesis pursues a statistical approach to object recognition. Therefore, a probabilistic model is sought for disentangling all the data contained in a scene. Recognition can then be formulated as an efficiently solvable inference problem in the underlying statistical model which is especially crucial in the case of near real-time video analysis where one has to deal with large amounts of data.

1.4 Current Challenges and Properties of the Proposed Approach

Learning object models for detection and recognition is one of the key challenges of computer vision. The complexity of this problem depends on several factors that will be discussed in the following. In constraint scenarios (e.g. visual inspection of conveyor belt assembly processes such as semiconductor fabrication), computer vision algorithms have been applied successfully for many years. In contrast to these partial vision solutions, the completely unconstraint, general vision problem is still beyond the frontier of modern research although it has been targeted since the 1960s
1.4. Current Challenges and Properties of the Proposed Approach

Figure 1.2: Intra-class variability of the category *airplane* from the Caltech-101 image database [Cal].

Let us now summarize crucial factors that determine the complexity of the vision problem and see what challenges this thesis deals with.

**Semantic granularity of classes:** Object classes that are to be recognized can be situated on any conceivable level of semantic granularity. Therefore, possible tasks might be anything ranging from exemplar search (finding exactly a given specific query object) to category-level recognition (finding all exemplars of a category such as airplanes). As within-class variations are significantly larger in the latter case this task is a harder problem. This thesis is dealing with exactly this challenge.

**Intra-class variability:** Object categories typically exhibit significant intra-class variations so that learning models for them is aggravated (see Figure 1.2 for some examples of the category *airplane*). The variability can take various different forms. Examples are variations in geometry, particularly among articulated objects or changes in appearance (such as the variability in color of the sheep in Figure 1.3). Even in the case of exemplar search such variability can become a severe problem due to aspect changes, i.e. object appearance can change dramatically under different views. Moreover, many object categories exhibit significant inherent scale variability as can for instance be seen in Figure 1.4.

Another challenge are object categories that are only defined on a functional level as opposed to a visual level: functional categories are characterized by their purpose (e.g. “a chair is a piece of furniture a person can sit on”), whereas visual categories comprise objects that are related by some kind of visual consistency, i.e. by the similarity of their visual representations. The learning of abstracts concepts such as “sitting on something” requires additional training information beyond visual cues. However, since this thesis is only based on visual information it deals only with visual categories.

**Between-class similarity and number of classes:** The complexity of a recognition task depends on the separability of the individual classes. A larger number of classes, or classes that are mutually similar aggravate the task of segregating them
Chapter 1. Introduction

Figure 1.3: Intra-class variability in color.

Figure 1.4: Inherent scale variability within an object category.

and, therefore, increase the complexity of the recognition task. This thesis deals with the recognition of large numbers of categories, i.e. at the order of a hundred classes.

Degree of supervision during training: Generally speaking, user supervision is favorable since it reduces the complexity of the learning task by providing additional information that helps to disambiguate the training data. Incorporating such information helps to simplify the very models that are to be learned. In the training phase of a computer vision system several types of supervision information are possible. First, training images can be segmented by hand into the different objects
1.4. Current Challenges and Properties of the Proposed Approach

that they contain and into background regions. Each of these regions is then labeled with the appropriate class label. Such a figure-ground-segregation is valuable as it removes distracting clutter and provides boundary contours of objects. However, providing hand segmentations as depicted in Figure 1.5(a) is a tedious task that becomes prohibitive when the number of categories increases. Therefore, many datasets approximate segmentations by merely providing bounding boxes, i.e. the rectangular hulls of objects in images (illustrated in Figure 1.5(b)). However, when assembling databases for large numbers of categories it is convenient to only provide an overall category label of the most prominent object in a training image. Such weak supervision becomes especially convenient when setting up datasets for object recognition in videos, because a frame by frame labeling is just too tedious. However, in this setting significantly less information is available and learning becomes harder than in a fully supervised approach (see Figure 1.6 for a caricature that illustrates this problem). The learning algorithm has to find out on its own what the relevant regions of an image or video frame are and it has to ignore the clutter. This thesis follows this last setting and focuses on such methods for learning object models from still images and videos that require as little supervision as possible. Moreover, the number of provided training samples is another critical factor. When approaching the limit of infinite amounts of training data a simple nearest neighbor classifier will achieve optimal recognition performance by merely memorizing the whole visual world at the expense of infinite classification time. However, for complexity reasons we aim at sparse object models that yield sufficient performance when only small sample sizes are available.

**Environmental factors:** Several properties of the environment and of the observing camera have significant influence on the complexity of the recognition task (see

Figure 1.5: Different degrees of supervision. a) Accurate hand segmentation. b) Bounding box. c) No segmentation or localization information. This thesis deals with the last setting depicted in c).
Background clutter: Usually images are not manually preprocessed to cleanly segment objects from the background. Such cluttered regions can contain distracting texture or several homogeneous patches (e.g. sky). However, they might also contain other objects that distract a learning algorithm. In contrast to this, there are situations when background is actually even providing relevant contextual information about the objects in a scene (e.g. a street background is indicative for the presence of cars).

Occlusion: Some parts of an object might be obscured by other objects or even by itself which is therefore named self-occlusion.

Illumination changes: The human visual cortex compensates for brightness and hue changes of light sources (which is called color constancy) so that a human observer, in most situations, is not aware of the specificities of the light source. On its lowest level a computer vision system is based on color measurements at individual image pixels. When the lighting of a scene changes, the colors are either shifted, scaled, or even transformed in a nonlinear manner due to shadows and changes of the position of the light source. When learning object representations such transformations have to be compensated.

Distortion due to viewpoint changes: When the camera is moved relative to the object, the visual appearance can be distorted by several in-plane transformations (translation, scaling, rotation, and skew) and out-of-plane transformations (foreshortening). However, not all viewpoints are equally likely. Such prior
1.5. Original Contributions

The main contributions of this thesis are listed below.

- A hierarchical object model is proposed that is based on compositions of parts and relations between them. The model exhibits competitive performance on standard databases.

- A learning algorithm is devised that is capable of learning compositional representations from image data without requiring hand segmentations or any other localization information.

- Although not trained with localization information the system is capable of inferring bounding box segmentations in the test phase.

- The object models are based on a codebook of atomic parts that is shared by all categories. The underlying feature descriptors are specifically designed to have low dimensionality and to be robust with respect to intra-class variations. They are shown to outperform a standard descriptor as feature for compositionality.

- All compositions, their spatial arrangement, and scene context are combined in a single statistical model, a Bayesian network. Recognition is then formulated as an efficiently solvable statistical inference problem that yields the category posterior of the object. Hence, it also provides a confidence in the categorization and not only a class label.
Chapter 1. Introduction

- Composition candidates are formed by means of perceptual organization and a Bayesian relevance detection algorithm is proposed that automatically retains relevant compositions.

- Category-level object recognition is successfully combined with image segmentation.

- The compositional approach is extended to video analysis where category-level object recognition, segmentation and tracking of multiple objects is performed in near real-time.

- Stability-based model selection forms the basis of multi-object detection and classification in video.

- Neither training nor recognition require any manual interaction. Only the category label for the most prominent object in a complete video sequence has to be provided during training. There is no restriction on the video acquisition process: for evaluation purposes a database of videos is assembled using a hand-held camera in ordinary street scenes with heavy clutter and camera motion.

1.6 Organization of the Thesis

This thesis is organized as follows:

Chapter 2 gives an overview over the various crucial components of visual object recognition systems. Different modeling paradigms are discussed and the state of the art is presented. This chapter reviews the basic concepts needed for the presentation of the composition system in later parts of the thesis.

Chapter 3 derives an invariant, low dimensional descriptor for atomic parts of a compositional recognition system—the localized feature histograms. Based on these features a recognition system is presented that forms tuple compositions.

Chapter 4 extends the compositional object model from the previous chapter so that compositions of arbitrary many parts can be established. It is then investigated how a small vocabulary of compositions can be learned that is shared by all object categories. Learning characteristic compositions works automatically without any user supervision regarding object localization such as hand segmentations. As
another key contribution this chapter presents a compositional shape model that couples all the individual compositions and scene context to obtain a concerted object hypothesis.

Chapter 5 describes how deep compositional hierarchies can be established. Therefore, several grouping laws from perceptual organization are combined to obtain a single bottom-up grouping algorithm. In addition, a top-down grouping procedure is presented that forms compositions recursively so that they are most characteristic for object categories that are present in a scene.

Chapter 6 investigates how image segmentation can be integrated into the recognition process. The goal is to exploit segmentations for forming characteristic compositions without having to devise complex grouping algorithms as in the previous chapter. Since individual segmentations are not reliable, ensembles of low-level segmentations are being established.

Chapter 7 analyzes how relevant compositions can be learned completely automatically rather than requiring the manual modeling of a complex grouping algorithm or computationally intensive segmentations. The compositional structure of objects is learned and represented in a Bayesian network. This generative model can then be used to sample object representations and to explain away background clutter.

Chapter 8 focuses on concurrent detection and classification of objects in highly cluttered scenes with multiple objects. The learning and recognition algorithms from the previous chapter are extended by interleaving localization and classification in an iterative procedure. The goal is to learn object detection without providing any localization information in the training phase.

Chapter 9 extends the compositional approach from still images to video analysis. The goal is category-level recognition, segmentation, and tracking of multiple objects in videos in near real-time. Compositional representations can reduce the complexity of object models significantly and turn learning of structured object models into a feasible problem. However, compositional representations might entail disadvantages during recognition, especially under the high computational demands of videos. This chapter presents a compositional approach to video analysis that performs near real-time and demonstrates how the key concept of compositionality can actually be exploited for both, rendering learning tractable and making recognition computationally feasible.

Chapter 10 concludes with a final discussion.
When building a vision system a number of crucial modeling decisions have to be made. These key decisions in the design process of vision systems can be divided into three broad categories: (i) the choice of appropriate features that capture scene content in an initial representation. (ii) The design of an object model that represents the characteristics of the different object classes. (iii) The development of a learning algorithm that learns object models from training data, preferably with as little user supervision as possible.

Although the human visual system can serve as an inspiration or guideline when tackling the different subproblems of the design process, it is, despite intense efforts, nevertheless far too complex and too poorly understood to be directly reimplemented on a machine. In the following, we will study how important modeling steps have been approached in the literature. As pointed out in the introduction, visual recognition can be pursued on different levels of semantic granularity. One extreme is exemplar detection (e.g. [Low04]), where exactly the same query object is sought in scenes with different environmental properties such as background, lighting, occlusion, viewpoint, etc. The other extreme is category-level object recognition. As opposed to finding a specific exemplar, this setting aims at recognizing all possible instances of a category. To give a concrete example, the first task might be to find only a specific car in an image, whereas the latter could be to find all types of cars. This thesis focuses on category-level recognition which exhibits challenging
intra-class variations.

Typically, current approaches in this field can be reduced to a common denominator as follows. First, images are represented using local appearance features that summarize sets of pixels. All of these descriptors are then comprised in a common model. Given this representation, object recognition typically becomes a classification task. In addition, a more or less accurate segmentation can also be required, depending on the complexity of the involved localization task. Within this coarse fundamental framework related work on (category-level) object recognition will be characterized by the way crucial modeling decisions are made.

2.1 Salient Interest Points and Local Image Descriptors

To be able to reason about an image in any way, the sensory input has to be represented in a form that makes it accessible for later processing. This is a crucial step since the initial representation lies the basis for all higher level tasks. Information that is lost at this stage can hardly be reacquired later on. On the other hand, lossy filtering has to be applied in order to make the representation invariant to artifacts introduced during image acquisition such as measurement errors or irrelevant environmental influences such as variations in illumination.

To find an appropriate image representation two questions have to be answered. First, it has to be decided which image locations to capture and which to dispose of. The second question is then how to represent the previously selected, relevant image regions. To solve the first problem intensive research has been done on interest point detection. These detectors find salient image regions on the basis of local image statistics alone [For94]. That is, they are not based on high level information about image content such as object categories. Therefore, bottom-up interest point detection represents an initial approximation to complex neural attention control processes (cf. [RWKP04, EMB+07, KSJ00]).

2.1.1 Interest Point Detection

Let us now briefly review some popular interest point detectors before covering local image descriptors in the following section.
2.1. Salient Interest Points and Local Image Descriptors

**Difference of Gaussian:** A popular detector is the difference of Gaussian operator introduced by Lowe in [Low99, Low04]. It defines interest points as the extrema of a convolution of the image with a difference of Gaussian filter. For an efficient implementation that works in real-time, the image is repeatedly blurred with Gaussian kernels before taking the difference between successive blurrings.

**Kadir and Brady:** This detector computes intensity statistics at each point in an image for various scales. Therefore, it establishes intensity histograms in a circular region of varying radius around each point. To find candidate scales for each image point, the entropies of the histograms that correspond to all radii at that point are computed and the local maxima are selected. Thereafter, a saliency measure (see [KB01, Lin98]) is computed for all candidates to yield a sparse saliency map in the scale space. A final clustering of the candidates in scale space does then yield a set of interest points. It turns out that this detector is especially effective at finding circular structures such as motorbike wheels. Therefore, it is particularly successful on datasets that feature few object categories with predominant circular structures (such as the Caltech-4 database [Cal]). However, the manual design of an interest point detector that is specifically tuned to many categories is rapidly becoming impractical as the number of classes increases. Therefore, this thesis uses saliency detection only to reduce computational complexity by avoiding a processing of image regions that are definitely uninformative. A further filtering of distracting image regions does then solely rely on an automatic relevance learning algorithm.

**Multiscale Harris:** Mikolajczyk and Schmid [MS04] have proposed a detector that finds corners in images by adjusting the scale of the region to find an optimal size. In a first stage the Harris corner detector [HS88] is applied to identify interest points. Subsequently, the characteristic scale of each interest point is detected to specify a region. Therefore, the radius of a scale-normalized Laplacian filter is varied and its response is measured at the interest point. The radius with maximal response is then taken as the characteristic scale. This detector is also suitable for real-time video processing.

2.1.2 Local Image Descriptors

**Appearance Patches:** The most straightforward representation of a local image region is to directly build a feature vector of all its pixels. In praxis some extra processing is needed to reduce the dimensionality and insure invariance to at least
limited image transformations. Therefore, image patches (typical sizes are in the range of $20 \times 20$ pixels) are extracted and converted to grayscale. Additionally, the regions can be smoothed using a Gaussian low-pass filter before subsampling them to obtain invariance with respect to small translations. For dimensionality reduction, linear methods such as principal component analysis (PCA) are frequently used. Finally, the features are vector quantized using normalized grayscale correlation as a distance measure which assures invariance to illumination changes of patches as a whole. As a result, a large codebook of typically a few thousand local patch representatives is obtained already for few categories (e.g. Caltech-4). This complexity becomes intractable for large category numbers. Moreover, since the resulting invariances are just established by a global subsampling and intensity normalization, translations or local alterations still have a disproportionately high influence on the complete feature. Additionally, due to the low-pass filtering, only information on the strongest edges is preserved while the remaining patch content is blurred. Nevertheless, the big advantage of this descriptor is that its visualization is simple since it can be displayed directly as an image patch. Probably mostly due to its simplicity, appearance patches have become a popular local feature that has been used widely, for instance in [AAR04, FPZ03, FFFP04, LLS04].

**SIFT Features:** The scale invariant feature transform (SIFT) [Low99, Low04] computes gradients in an image patch. The gradients are then normalized for orientation by rotating the whole region so that its dominant orientation is fixed. Then the region is divided into a regular $4 \times 4$ grid and 8 bin orientation histograms are computed for each cell. As a result a $8 \times 4 \times 4 = 128$ dimensional feature vector is obtained. Although this descriptor is designed for exemplar detection it has also shown acceptable performance in the field of categorization (e.g. [SRE+05]). However, the high dimensionality and the specificity of these features with respect to individual instances of an object require large codebooks when these features are clustered.

**Geometric Blur:** The geometric blur descriptor [BM01], which is an extension of the *shape context* descriptor [BMP02], blurs the region around an interest point with a spatially varying kernel. In [BBM05, ZBMM06] Gaussian kernels with a standard deviation that is directly proportional to the distance from the center of the descriptor are used. The idea is to put more emphasis on the center of the region and suppress areas that are far away using strong smoothing. Signals that are filtered with these kernels are typically the output of an edge detector such as the one of Martin et al. [MFM04]. As a result a high dimensional feature vector is obtained
that describes the image region surrounding an interest point.

**Gabor Filters:** Gabor filters are inspired by the classical simple cells that Hubel and Wiesel found in the primary visual cortex (V1) [HW62]. These filters take the form of Gabor functions [Gab46] which have been shown to be a good model of simple cell receptive fields [JP87]. The impulse response of a Gabor filter is the product of a harmonic function and a Gaussian. Thus the filter is defined by

\[
A_{\lambda, \theta, \sigma, \gamma}(x, y) = \exp \left( -\frac{\hat{x}^2 + \gamma^2 \hat{y}^2}{2\sigma^2} \right) \times \cos \left( \frac{2\pi}{\lambda} \hat{x} \right), \text{ s.t. } (2.1)
\]

\[
\hat{x} = x \cos \theta + y \sin \theta \quad \text{\&} \quad (2.2)
\]

\[
\hat{y} = -x \sin \theta + y \cos \theta . \quad (2.3)
\]

The parameter \( \lambda \) specifies the wavelength of the filter and \( \theta \) is its orientation. \( \sigma \) corresponds to the effective filter width by controlling the size of its envelope, whereas \( \gamma \) specifies the spatial aspect ratio, that is the ellipticity of the filter. Characterized by these parameters, a Gabor filter responds to a texture of specific spatial frequency and orientation. A local feature descriptor of an image is then obtained by combining the responses of an array of different Gabor filters (a filter bank).

Gabor filters are commonly used as an initial representation layer in neurophysiologically inspired vision systems (see for instance [RP99, SWB+07]).

### 2.2 Object Models and Representation Schemes

Interest point detection and local features yield a sparse representation of an image. The fundamental question is then how to combine the individual, distributed stimuli so that they act as a unit when representing an object. This problem is referred to as the *binding problem* in perception [CK90, RN99]. The challenge is therefore to find a model that combines all the local descriptors and captures their co-occurrence as well as any spatial relations between features. Thereby not only local observations are made but also the global spatial structure of an object is represented. This spatial structure or global geometry of an object is often referred to as its *shape* [Tho17, Ken84, Boo86, DM98, Sma96]). Kendall [Ken84] has given an informal definition of shape that has been aptly paraphrased by Dryden and Mardia [DM98]:

“Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.”
Recognition does then require to solve the *correspondence problem*—features of a test image have to be matched against the parts of a learned representation. For an optimal assignment of query features to model features, local descriptor correspondences as well as the global spatial structures have to be matched at the same time [BBM05]. The matching process is based on the assumption that objects do not scatter features arbitrarily in an image. This assumption is in turn founded on the structure and regularity of the visual world.

### 2.2.1 Model-Based Vision and View-Based Approaches: An Overview

Early approaches to object recognition have represented visual objects using models that approximated the *physical object* being imaged. Such *model-based* vision systems are either directly based on 3-D object representations or they decompose them into simpler volumetric primitives or surfaces [Rob63, Bin71, Bro81, BL03, DPR92]. The rationale of model-based vision is to center the representation on the physical objects so that it becomes invariant with respect to view changes. However, there is a trade-off between viewpoint-invariance of descriptors and their ability to discriminate between objects—a compromise are for instance features based on *moment invariants* [Alt62, Hu62]. This trade-off makes it difficult to reliably extract and hypothesize abstract geometric representations.

In contrast to model-based methods that are centered on physical object models, *view-based* approaches are founded on models which are directly centered on the observed view of an object in an image. These methods have become popular in the 1990s. In the simplest case recognition proceeds merely by efficient comparisons of local regions in the image space. This technique is generally referred to as *template matching*, e.g. [PRV98]. A simple way to obtain invariance with respect to object translation are *sliding windows* (e.g. [SK00, VJ01]). The image is divided into (overlapping) regions and each of these segments is matched against the template. Based on the matching score we then can decide which region contains the object of interest. Similarly, invariance to scaling, or rotation can be incorporated by additionally searching over scale and orientation. To compensate for environmental factors such as shadows, or introduction of clutter, the comparison is typically conducted in a feature space that results from projecting the image space using linear or non-linear methods. A popular linear method is *principle component analysis* (PCA) [Jol86, HTF01] which results in eigenspace representations [MN95, HCK97] and particularly in *eigenfaces* [TP91]. However, global image transformations such
2.2. Object Models and Representation Schemes

as translation, scaling, or illumination changes have to be removed in a preprocessing stage before applying a PCA approach.

View-based methods that are based on a holistic object representation suffer from the complexity of these models and their fragility when the spatial structure of objects exhibits variability (e.g. articulated objects). To address the latter problem deformable template matching has been introduced [YHC92, JZL96]. This approach compensates for variations in the spatial structure by applying a global transformation when matching templates. The total deviation of the deformed template from the probe image should be minimized. This distance consists of the local distances between key points on the deformed template and feature points in the image. As an additional constraint, the total deformation is kept bounded so that simple transformations are preferred while minimizing local deviations between template and probe. Based on this principle Chui and Rangarajan [CR03] have proposed a matching algorithm that uses thin-plate splines for regularizing the transformation. This keypoint registration algorithm has served as the basis for recognition systems based on shape matching [BBM05, FJS07].

Another solution that is currently very popular are part-based models [FE73, LVB+93]. These models represent an object as consisting of a number of specific parts that feature characteristic spatial structure. Typically parts refer to local regions of an object. However, it is also possible to construct models that are only based on boundary curve segments and their spatial structure [Fel05, FTG06, OPZ06]. Therefore, these methods are purely based on the shape of objects (cf. [Ken84, Boo86, DM98, Sma96])—their spatial structure—and do not take appearance into account. Additionally, hybrid methods have been presented that combine local appearance descriptors with boundary segments [FPZ04]. In addition to part-based modeling of the object contour, research has also focused on holistic contour representations. Popular parametric models for contours are splines [dB78] and in particular B-splines due to their computational convenience. The seminal work on snakes by Kass et al. [KWT87] has then introduced elastic contours that are fitted to features in an image. In their work, prior information about the shape of a contour is traded against evidence from the image. Moreover, active contours cannot only be matched to a static image but also to image sequences [BI98]. Here the deformation of a curve over time that is necessary to keep track with moving objects is retarded by an inertial force. Although active contours are by their nature only representing the outline of objects an additional measure of the homogeneity of the circumscribed segment can be incorporated as is done by Cremers et al. [CTWS02]. Consequently the resulting snakes also take coarse appearance information about the homogeneity
Chapter 2. Key Modeling Decisions and State of the Art

Figure 2.1: A graphical illustration that contrasts the different modeling paradigms. Each of the four axes represents a different characteristic aspect of object recognition systems with the two antithetic extremals of the respective aspect printed at the ends of that axis.

A summary of the different modeling aspects of object recognition systems is presented in Figure 2.1. Each axis represents a different characteristic and at the ends of each axis the two extremal concepts are printed.

2.2.2 Towards Generic Object Recognition

The previous section has presented an overview over the different general modeling concepts in computer vision. Subsequently, we will look at the individual system design decisions in greater detail to see how they have been approached in the literature.

Geometry Without Appearance

In the early years of vision research, image acquisition and handling was cumbersome and costly. As a consequence, a main stream of research focused on the geometry of objects thereby leaving aside appearance information. The resulting model-based approaches aim at approximating the physical world directly. Therefore, the 3-D geometry of an object is described by decomposing it into simple primitives. In the 1970s and 80s many such shallow hierarchies existed that were based on volumetric primitives such as generalized cylinder-based parts or surfaces such as ribbons and ellipses. David Marr has presented an inspiring view on computer vision in the landmark contribution [Mar82]. The proposed 2-D sketch functions as a connection between the observed image and the model. Classical vision systems from this domain are Brooks’ ACRONYM system [Bro81] and Binford’s SUCCESSOR
system [BLM87, BL03]. Other approaches include the schema system [DCB+89] and interpretation trees by Ikeuchi and Kanade [IK88, GP87]. Initially, models were hand built 3-D CAD models. Moreover, range data was popular [Bin71, AB73] as it provides direct measurements of the depth of a scene and thus simplifies the extraction of 3-D shape. In later years the field focused on methods that work directly on intensity images without requiring range data. However, these methods aimed at recognizing specific instances of a 3-D object from 2-D views rather than tackling the more general problem of category-level recognition.

The recognition algorithms used by model-based approaches typically takes on a single form known as hypothesize and test [FP03].

1. Given a collection of geometric object models (the model base of all candidate objects) a correspondence between a set of query image features and a candidate object model is hypothesized.

2. Based on this hypothesis a projection of the candidate model to the image frame is estimated. Therefore, this backprojection step creates a rendering of the object model for the final verification stage.

3. Finally the rendering is compared with the query image and the hypothesis is accepted if both are sufficiently similar.

There is an exponential number of potential correspondence hypotheses, so an exhaustive search is computationally prohibitive. To improve the robustness of the algorithm with respect to viewpoint changes and to increase its speed geometric invariants have been introduced and geometric hashing (a voting on object geometry) has been performed [RZMF92]. Moreover, greedy alignment techniques have been presented to prune the set of potential hypotheses. An example is Lowe’s SCERPO system [Low87] which uses perceptual grouping for limiting the search space. First, lines are extracted from a query image before grouping them using perceptual grouping rules such as parallelism and co-linearity. A 3-D CAD model is then brought into correspondence with the various groups of lines by solving for the unknown viewpoint. This matching is performed using only subsets of all lines to obtain robustness with respect to occlusion.

Finally, model-based approaches differ from each other by the intrinsic representation they use for 3-D objects. Approaches such as [Low87] are build on a direct 3-D representation whereas Ullman and Basri [UB91] incorporate a mixture of 2-D models that are matched to the image using lines and points. Research of human vision has seen a related debate. Some argue that the brain uses a 3-D representation.
Biederman [Bie87], for instance, proposes *geons*, a set of 3-D primitives like cones, cylinders, or cuboids. His idea is that any object is decomposed into these generic constituents. In contrast to this, a view-based representation has been proposed by Riesenhuber and Poggio [RP00].

**Appearance without Geometry**

In contrast to the geometry-based methods summarized above, a number of appearance-based methods have been presented in the literature. In [PRV98] a template based approach is pursued. Therefore images are subsampled before summarizing the pixels in a vector. Classification is then performed using a linear support vector machine. Moreover, unsupervised dimensionality reduction methods are frequently used. The goal is to obtain a compact image description that is invariant to local image alterations (e.g. noise) so that learning a classifier becomes a statistically feasible problem. A widely used linear technique is *principal component analysis* [Jol86, HTF01]. PCA has been applied to recognition in two different ways. One is to construct a single global eigenspace for all object classes [MN95]. Training samples are then projected into this space and, in its simplest form, recognition becomes a nearest-neighbor search [DHS01] in the eigenspace for the closest training sample. The second approach is to construct separate eigenspaces for each class [TP91]. Recognition is then performed by measuring how good some class specific eigenspace can represent a query image. Whereas the original eigenspace approach was holistic in nature (modeling the appearance of whole image) a smooth transition towards part-based models took place in later research. A paper along that line is the modular eigenspace description technique of Pentland et al. [PMS94] which incorporates salient image parts such as eyes, nose, and mouth. Besides unsupervised dimensionality reduction, discriminative techniques have also been applied in the literature. While PCA seeks a subspace projection that is efficient for representation, *Fisher’s linear discriminant analysis* (LDA) [Fis36, McL92, DHS01] looks for a projection that is efficient for discriminating object classes from each other. Belhumeur et al. [BHK97] have applied LDA for face recognition. Recent examples of appearance only models are the discriminative approach of [DS05] and the AdaBoost based ensemble of appearance descriptors by Opelt et al. [OPFA06].

**Global Image Histograms in Content Based Image Retrieval:** In content-based image retrieval [Eak02] histograms over complete images have been popular [VT00]. These methods integrate features such as color or texture over whole images.
Consequently these representations are invariant with respect to local changes in an image and also to global transformations such as translation at the cost of limited specificity, sensitivity to background clutter, and limited invariance to occlusion. To enhance specificity, a smooth transition towards incorporating feature localization information (and thus global object geometry) has taken place. Approaches along that way are the one of Swain and Ballard [SB91] and that of Schiele and Crowley [SC00]. The latter method establishes joint histograms over local appearances, which are measured by local shape descriptors. The histograms do also incorporate spatial information of the local descriptors. Moreover, Vogel and Schiele [VS07] follow a region-based approach to content-based image retrieval that is based on semantic descriptions of local segments. They incorporate limited spatial information by using a rigid grid of local regions. However, a representation that exhibits better localization is crucial for recognition and especially for detection of individual objects.

**Bag-of-Features, pLSA, and LDA:** Over the last years, *bag-of-words* models from text retrieval [SM83] have become very popular in the field of visual recognition, e.g. [CDF+04, SRE+05, FFFFZ05, ZMLS06, LSP06]). In this context they are commonly referred to as *bag-of-features* approaches. They represent an image (corresponding to a text document in text mining) by a distribution over a codebook of characteristic features (analog to words in text retrieval). Consequently, the representation is a histogram that lists the occurrence frequencies of each prototype in the image. In the training phase, features are collected from all training images and a codebook is inferred by clustering all extracted features. Thereafter, a classifier is trained that maps the bag-of-words vector to a class label. Consequently, the bag-of-features representation disregards the spatial arrangement of the local observations and only captures their co-occurrence. The advantage is a compact representation that can be learned even from few samples in a statistically feasible manner. Moreover, no complicated spatial correspondence problem has to be solved. However, the lacking of spatial information limits the discriminative power of this representation and it also hinders the localization of objects. One way of dealing with the latter problem is to apply *sliding windows* together with some pruning technique that limits the complexity of the search space—a popular pruning method is a cascade of classifiers as in [VJ01]. Additionally, a windowed version of the bag-of-features approach reduces its sensitivity to background clutter [MS06].

Lazebnik et al. incorporate spatial information into the bag-of-features approach by splitting images into a regular grid and describing the individual cells using separate feature bag descriptors. The image is then represented by concatenating all the
individual descriptors. Therefore, the method is basically performing rigid template matching on the level of grid cells. Consequently, this approach relies heavily on the assumption that the spatial structure of objects is fixed with respect to the image. A related technique has also been applied to recognition of objects in videos by Dalal et al. [DTS06].

Based on the bag-of-features approach *latent semantic analysis* (LSA) and related techniques from text retrieval have become popular in object recognition over the last years. Again each image is represented by an occurrence histogram over a fixed codebook. Consequently, the set of all training samples is characterized by a large co-occurrence matrix which is then decomposed using a singular value decomposition. The eigenvectors correspond to different latent topics and the eigenvalues give their relative weighting. Hofmann [Hof99, Hof01] has brought LSA into a probabilistic framework, calling it *probabilistic latent semantic analysis* (pLSA). Given a set $D$ of images $d \in D$ (the documents in text retrieval), each image is represented by a bag-of-features over prototypes $w \in W$ (the words). Consequently, the whole corpus of sample images is described by a $W \times D$ co-occurrence matrix of joint probabilities $P(w,d)$. The idea of pLSA is to introduce a hidden layer of latent topics $z \in Z$ that $d$-separate [Jen01] the random variables $w$ and $d$. The number of latent topics typically equals the expected number of object categories in the dataset. By marginalizing over latent topics, the joint distribution can be decomposed into two simpler conditionals and a document prior

$$P(w,d) = \sum_{z \in Z} P(w,d,z)$$

$$= \sum_{z \in Z} P(w|z)P(z|d)P(d).$$

The conditionals are learned in an unsupervised manner by applying an *expectation-maximization* (EM) algorithm [DLR77, MK97] as described in [Hof01]. Novel images $d$ are classified by running EM with $P(w|z)$ fixed. As a result $P(z|d)$ is obtained which is the mix of topics (corresponding to categories that are not explicitly defined by the user) in that image. Sivic et al. [SRE05] have applied several LSA methods to visual recognition. Moreover, Fergus et al. [FFFPZ05] have incorporated spatial information into pLSA.

Blei et al. [BNJ03] have proposed an extension called *latent dirichlet allocation* (LDA). In this model the multinomial weights over topics and words of pLSA are treated as latent random variables that are distributed according to a Dirichlet distribution. These priors assign probabilities to data that lies outside of the training set. Fei-Fei and Perona [FFP05] show an application of LDA to scene analysis.
2.2. Object Models and Representation Schemes

Figure 2.2: Illustration of different part-based models. The depicted graphical models show which dependencies between parts each type of model establishes. a) Bag-of-features: all parts are assumed to be independent. b) Star graph: there exists one reference part \( f_1 \) on which all other parts are conditioned. c) \( k \)-fan \( (k = 2) \): \( k \) reference parts on which all other parts are conditioned \( (f_1 \text{ and } f_2 \text{ in this example}) \). d) Tree model: parts dependencies form a tree hierarchy. e) Constellation model: each part is depending on all other parts resulting in a fully connected graph. f) Compositional hierarchy: intermediate compositions of parts, the \( g_i \), are established. In contrast to d), parts are only present at the leafs of the hierarchy. The intermediate compositions \( g_i \) are not observed directly but inferred from the parts.

Geometry and Appearance: Part-Based Models

Part-based models are a popular choice for enriching view-based approaches with global object geometry. Several different models for combining local appearance in a common object representation have been proposed. An illustration of the different types is presented in Figure 2.2. The depicted graphical models show which dependencies between parts are established by each type of model. While the purely geometric or purely appearance based methods are typically only suitable for tasks such as exemplar search or recognition of very few object classes (e.g. one class against background), recent part-based models have been successfully applied to recognition problems with more than a hundred categories and large intra-class vari-
ations. Moreover, it is commonly agreed upon that artificial vision systems should eventually incorporate both global object geometry and appearance information. Interestingly enough the simplest part-based models, bag-of-features, without spatial structure have nevertheless become popular in the last few years.

**Parts and Structure Models:** In their 1973 landmark paper Fischler and Elschlager [FE73] proposed the first model that combined local parts and global spatial structure. It consists of local parts represented by small templates that are arranged in some geometric configuration. This type of model is commonly referred to as *parts and structure model*. Recognition does then require to solve a 2-fold correspondence problem. On the one hand local parts should fit to the parts of a model. On the other hand the global deformation from a standard configuration is to be minimized at the same time. The spatial model has the analogy of a set of parts that are connected with springs. Although the authors obtained only limited results with their attempted face recognizer, this part-based model was an important starting point for later research. Inspired by the *dynamic link architecture* for cognitive processes, Lades et al. [LVB+93] followed the same fundamental idea when proposing their face recognition system based on a deformable grid template.

Amit et al. [AG98, AGW97] have investigated principled probabilistic object models with semi-local feature descriptors and applied them to face detection and character recognition. Their approach is based on groupings of edge fragments that exhibit stable relative positions. These grouping are further agglomerated into a joint object representation. During training, edge groupings are sought that discriminate one object class from the rest.

**Biologically Inspired Convolutional Networks:** Several biologically inspired vision systems have been proposed over the years. A first influential model is Fukushima’s Neocognitron [Fuk80]. Inspired by the *simple* and *complex cells* of Hubel and Wiesel [HW62], alternating layers of S and C-type neurons constitute the feed-forward Neocognitron. The S-cells pool the outputs of different C-cells in the previous layer (convolution operation), whereas C-cells perform a sub-sampling by computing a weighted sum of a local patch of S-cells in the previous layer. Thereby S-cells build up feature complexity and C-cells introduce location invariance. The model starts with a 2-D array of pixels as input and presents the classification result by the activation of cells in the output layer.

After the Neocognitron other convolutional networks followed that benefited from improved learning algorithms. LeCun et al. [LBBH98] have proposed large neural
networks that receive pixels from small images as input and return the class label as output. The layers in this convolutional neural network alternate between sub-sampling and convolution operations. Neurons in the output layer compute the distance of their input to a pattern that is stored in the weights. The spatial extend of the receptive fields of neurons increases towards the output layer. Therefore, occluded or corrupted parts of an image have a significantly lower influence on neurons in higher layers (such as the output) than on those of the input layer. In [LBBH98] the network is applied to the problem of handwritten digit recognition on the MNIST dataset [LeC]. During training the network weights are learned using back-propagation on labeled training digits. Due to the large number of weights large training sets (at the order of $10^5$ samples) are required. The method has been shown to yield competitive performance and an extension to a five category dataset of toy objects has been presented in [LHB04].

Another feed-forward model that is inspired by the organization of the ventral stream of the primate visual cortex is the HMAX model [RP99]. Like other convolutional networks this model consists of alternating S and C layers. However, the input layer consists of a Gabor filterbank that is applied to an image rather than taking image pixels as direct input. Moreover, the sub-sampling is performed by computing the maximum over all inputs of a C-cell rather than the weighted sum. Finally, HMAX is a multi-scale model since C-cells pool information not only over locations but also over nearby scales. Later extensions [SWP05, ML06, SWB+07] have modified the initial approach and applied it to multiclass categorization.

**Constellation Models:** Based on the parts and structure model of Fischler and Elschlager [FE73] the constellation model has been developed by Perona et al. [BWP98, WWP00a, FPZ03]. This model represents the joint configuration of all local parts. Each local part is an appearance descriptor of an image region. A constellation model describes the joint distribution of the local appearances and their mutual spatial relationships. When depicting the dependencies between parts in a graphical model, Figure 2.2(e), we therefore obtain a fully connected graph. This indicates that each part is depending on all other parts and no simplifying conditional independence properties are incorporated. A number of models of this type have been published so the following gives a brief overview over the evolution of this concept.

Initially, a face model was manually trained by identifying reference points on a set of training images [BLP95, BWP98]. The model consisted of the statistics of a set of detectors as well as of the joint statistics of their relative locations. Thereafter,
Weber et al. [WWP00a, WWP00b] improved the approach to reduce the amount of supervision information required for training. Their main step was to incorporate an interest point detector and $k$-means clustering to reduce the number of candidate parts. Fergus et al. [FPZ03, FPZ04] have further extended the approach by using a probabilistic framework. Particularly, local appearance is represented with probability density functions in their approach. In [FPZ04] Fergus et al. estimate the joint Gaussian distribution of spatial arrangement, scale, appearance, and edge curves in all detected patches. However, the number of parameters in this model grows exponentially with the number of parts and, therefore, the complexity of the joint model causes only small numbers of parts (around 5 parts) to be feasible. Consequently, the approach is only suitable for object classes that can be characterized with very few, highly specific parts. An object representation that is founded on very few, but highly specific components suffers from the problem that such critical parts can only be detected with limited reliability. In [FFFP03, FFFP06] priors are incorporated into the constellation model to significantly reduce the required amount of training samples to less than 5 images per object category.

**Models with Large Numbers of Appearance Patches:** In contrast to complex constellation models of very few parts, [AR02, AAR04, LLS04, LS04] have investigated models that are based on large numbers of local image patches. In the manner of Weber et al. [WWP00a, WWP00b], Agarwal and Roth [AR02, AAR04] extract appearance patches at interest points and cluster them to find common patches. An image is then represented by all patch prototypes that have been detected and by a coarse set of spatial relations between patches. A sliding window technique, which searches exhaustively over all locations and scales, is used to obtain candidate object hypotheses. Hypotheses that cover background are then filtered out with a sparse network of winnows (SNoW) classifier [CCRR99]. The approach is applied to car recognition and evaluated on the UIUC car dataset [AAR].

Leibe and Schiele [LLS04] have extended the approach of Agarwal and Roth with a star graph like shape representation (cf. Figure 2.2(b))—the implicit shape model. Again appearance patches are clustered to obtain prototypical representatives of object parts. Additionally the relative location of the object center is recorded for each part during training. Moreover, a foreground/background mask is stored. During recognition all detected parts vote for hypothetical locations of the object center by means of a probabilistic Hough voting scheme. The maximum in voting space is then selected as the position of the object center and it is used to discard irrelevant parts in a subsequent backprojection stage. Finally, they follow the jigsaw-
like segmentation approach of Borenstein and Ullman [BU02, BSU04]. Therefore, the foreground/background masks which are associated with each remaining prototypical part are averaged to segment the object from the background. [LS04] extends the approach to deal with objects on multiple scales. The method has been shown to yield appealing results. However, in contrast to the constellation model approach of Fergus et al. [FPZ03] Leibe and Schiele require significantly more supervision during training—each training sample has to be manually segmented. Therefore the approach is limited to small numbers of object categories. Typically, it is applied to discriminate one category from background. Recently, in several publications the model has been applied to pedestrian recognition [LSS05, SLS06, SS06].

Tree and \( k \)-fan Models: Based on the approach of Fischler and Elschlager [FE73] Felzenszwalb and Huttenlocher [FH00, FH05] have proposed part-based models where the spatial relationships between parts are tree-structured for computational reasons, cf. Figure 2.2(d) for an illustration of the dependencies between parts. The model is used for recognizing people in images. Training requires to manually pinpoint the part configuration in sample images—hence only small numbers of parts (5-10) are used. A significant contribution of [FH00] is that they reduce the complexity of the matching algorithm. The matching is linear in the number of possible locations per part so that an exhaustive scan over all locations becomes computationally feasible.

Crandall et al. [CFH05] have compared part models of different complexity. These so called \( k \)-fans (illustrated in Figure 2.2(c)) are graphs that represent dependencies between object parts. Therefore, a subset of \( k \) parts is selected that serve as reference parts. All other parts are then depending only on the reference parts, i.e. non-reference parts are independent conditioned on the reference parts. Therefore, \( k \) specifies the complexity of the model. A 0-fan is a bag-of-features model with no dependencies, star graph models like [FPZ05] are 1-fans, and a fully connected constellation model of \( N \) parts corresponds to an \( N \)-fan.

Instead of using a fixed set of reference parts, Carneiro and Lowe [CL06] have only kept the number of reference parts fixed by conditioning each feature on its \( k \) closest neighbors in the image. To learn this model an on-line learning algorithm has been proposed which is advantageous compared with the batch learning that is employed in most of the other works in this field.

Shape Matching for Solving the Correspondence Problem: Berg et al. [BBM05, Ber05] have posed the correspondence problem of part-based models as an
integer quadratic programming problem. Although, this problem is typically NP-hard they come up with an approximation that is computationally feasible. Inspired by [CR03], thin-plate splines [Wah90, Boo89] are used for regularizing the non-rigid mapping between a query and a trained model. In contrast to the deformation field of [WJ05], which represents deformations by a regular grid of mutually independent shift vectors, this representation of shape has the advantage that it directly regularizes the deformation by modeling the spatial dependencies between image regions. Finally, a nearest neighbor approach [DHS01] that matches a query against all training samples for all categories yields the correct object class. In [ZBMM06] classification is improved by combining the nearest neighbor classifier with a support vector machine (SVM) [Vap98, CST00, SS02].

3-D Geometric Context: Recent work in the field of object recognition mostly focused on view-based object representations without using information on the 3-D nature of the physical objects being imaged. In contrast to this, Hoiem et al. [HEH05a, HEH05b, HEH07] obtain coarse surface orientation estimates from only a single image. By incorporating a-priori assumptions on the imaging process and the resulting 3-D structure of the scene [HEH06], many unlikely object hypotheses can be ruled out. Thereby a further step towards disambiguating object recognition is made.

Another source of information that can be used for simplifying recognition is scene context [Tor03]. The idea is to detect objects exclusively based on contextual cues. To this end it is exploited that objects are correlated with other objects in the scene and with background clutter. Descriptors for scene context that measure the statistics of natural images have been investigated in [TO03]. The idea is to capture those characteristics of a scene that are indicative for the presence of object categories—the gist of the scene [Fri79]—without having to detect and recognize the objects. Instead global scene statistics are computed.

2.3 Hierarchical Object Models and Compositionality

Most model-based approaches of the 1970s and 80s were at least based on shallow hierarchies. In this context, hierarchies have been employed mainly because model-based representations are rather abstract and high level in the sense that they are not focused on the observed images but on the physical, 3-D objects being imaged.
Building such abstract concepts from a concrete image requires a complex abstraction process. To simplify this process, it has typically been divided into several smaller steps resulting in intermediate representations. A classical example is for instance Marr’s \textit{primal} and \textit{$2\frac{1}{2}$-D sketch} \cite{Mar82}.

In contrast to hierarchical model-based systems, view-based approaches perform classification as directly as possible in the image space and are, therefore, typically non-hierarchical. There are good reasons for such shallow representations. Simple view-based models that incorporate very limited geometry or even no spatial structure at all can be represented directly without intermediate stages. Examples are template matching \cite{PRV98}, eigenfaces \cite{TP91}, global image histograms \cite{VT00}, or simple bag-of-features approaches \cite{CDF+04}. However, when spatial structure is to be incorporated into a representation, the complexity rises rapidly with the number of model parts. As can be seen in the case of constellation models, flat approaches become rapidly intractable when the degree of structure is increased.

\subsection{The Need for Hierarchical Representations and its Pitfalls}

A fundamental problem that vision systems face is the large \textit{semantic gap} between the initial object representation and the final model parameters that characterize an object category. The initial representation is constituted by direct measurements that are made in the image (observed features in the pixel domain such as appearance patches) whereas the final model is very condensed (e.g. only the object category label and a parameter for object location and size).

When the initial measured representation and the final concept representation are far away from each other, a direct modeling requires complex models that are cumbersome and difficult to learn. In other words, learning object models corresponds to learning a mapping from the initial measurements into the space of abstract object representations. When both layers are too far apart it is favorable to replace the complex mapping by a concatenation of simpler ones that correspond to individual layers of an hierarchy. Moreover, it can be argued on information-theoretic grounds that representations of intermediate complexity maximize object class information and are thus optimal for classification \cite{UVNS02}. However, we cannot merely stack an arbitrary number of layers on top of each other and expect a functioning hierarchy. Noise and other disturbances at the feature level can be amplified by successive representation layers as in a high tower of jelly that gets shaken. Moreover, the complexity of the final representation is likely to be huge so that learning model
parameters from limited data becomes intractable. Therefore, the complexity of the individual layers have to be restricted and irrelevant intermediate representations should be discarded to limit the model space to a size that renders model learning statistically feasible. In addition it is favorable to compare the models that are inferred on higher levels with the initial input in a subsequent verification stage.

Recently, several view-based approaches have been presented that show at least shallow hierarchies. For instance the bag-of-features approach, which is suited for large numbers of parts, has been extended by pLSA and LDA which introduce a single hidden layer (e.g. [SRE+05]). However, these models typically lack any spatial structure (exceptions are for instance [FFFPZ05, STFW05]). On the other end of the modeling spectrum are the heavily supervised models of Felzenszwalb and Huttenlocher [FH00]. These are tree-structured representations of only few parts that are modeled by hand. Other examples are [BT05] and the feature decompositions of [EU05].

2.3.2 The Principle of Compositionality

Compositionality refers to the prominent ability of human cognition to represent entities as hierarchies of meaningful and generic parts. Therefore a comparably small number of generic lower-level constituents is used in widely differing contexts to build a (nearly) infinite number of hierarchically constructed entities. The fact that just a few atomic components suffice to enable perception in various situations is fundamental to the flexibility of human cognition: Using a common set of low-level entities that are not characteristic for any category but generic, new scenarios of widely differing nature can be tackled without having to learn a novel low-level representation in order to adapt to new tasks. Moreover, the used entities are usually much simpler than the scenarios described by their compositions (cf. [Bie87]). The relations between the entities—e.g. their spatial arrangement—are fundamental for the meaning of their composition. Consequently, a compositional representation contains more information than what is merely present in its individual parts.\(^1\) Wertheimer sum-

\(^1\)Hoffman [Hof98, pp. 47 ff.] describes the visual limitations of a patient who was diagnosed to have “visual form agnosia” after a poisoning. Though he was able to accomplish most of the basic tasks such as discriminating colors, perceiving small motions, or detecting boundaries, he could not group these stimuli into a greater whole to form objects. As a result he was unable to recognize objects such as body parts, and he could not even identify himself when seeing an image of his face. This disease pattern emphasizes the fundamental idea of Gestalt psychology that groupings are highly important to perception, because they contain more valuable information with respect to the overall semantics of an image than all the components on their own do.
marized this in the phrase “the whole is different from the sum of its parts” [Gol89, p. 194]. Finally, it should be said that the flexibility of human vision that results from compositionality is also fundamental to computer vision, since it forms a basis for general applicability of a vision system.

The Origin of Research on Compositionality

Compositionality is essential for all of cognition, but it is especially evident in the syntax and semantics of language. Using only a very limited number of atomic entities, the letters from the underlying alphabet, infinitely many words and sentences can be formed. These sentences can be used to describe even the most different scenarios that are imaginable. Although Gottlob Frege has highlighted the separability of sentences and thoughts into sub-structures in [Fre23] and [Fre92, Fre93], probably the first contribution that has actually introduced the word “compositionality” is the paper of Katz and Fodor [KF63]. Since then compositionality has been studied in many diverse fields such as linguistics, logic, and neuroscience [WMS05]. To base all of this research on a common ground a universal definition of the principle of compositionality has been sought for (cf. [Hod98]). Werning et al. [WMS05, p. 10] propose a concise formulation that serves as a good approximation to this principle:

“An interpreted representational system $R$ is compositional if and only if for every complex representation $r$ of $R$, the meaning of $r$ is determined by the structure of $r$ and the meaning of the constituents of $r$.”

In linguistics structure is defined by orthographic, syntactic, and grammatical rules. Moreover, there exists a small set of atomic parts or primitives (letters in written language and phonemes in spoken language) that serve as fixpoints for a recursive decomposition. Consequently, recognition based on a compositional representation system bridges the semantic gap by constructing intermediate representations that are situated between the low-level input (represented by a large number of primitives with structure) and the high-level representation (the object class plus model parameters).

Biederman [Bie87] gives an analysis of compositionality in vision. He presents a restricted number of atomic shape primitives, called geons, that constitute the components of more complex objects. Moreover he proposes a hierarchical construction process of entities that can serve as a basis for recognition. The detection of an object is then based on the spatial relationships (the structure) of its components. Biederman states that each geon has certain distinctive properties which are invariant
over different viewpoints. Using non-accidental cues he bases recognition directly on the two-dimensional representation of geons without having to reconstruct a three-dimensional representation. The non-accidental principle [Low85, WT83] assumes that specific regularities in an image correspond directly to related true regularities in the world, since they are not accidentally caused by the imaging process.

### 2.3.3 Perceptual Organization: Retrieving Relevant Compositions

Obviously, even a small set of atomic parts with arbitrary relations can lead to an infinite number of recursively built compositions. This large space of potential compositions poses two related questions. (i) How can a vision system learn compositional representations in a statistically feasible manner in the training phase? (ii) Is there a way to render the search for appropriate compositions during recognition a computationally feasible problem?

What we need is a way to limit the space of all potential compositions. The goal is to obtain a significantly smaller space of candidate compositions for which learning of statistics and inference become feasible. It turns out that Gestalt psychology [BGG96, Mur73] has intensively investigated the very same problem, although in a slightly different setting. This branch of psychology has its origin in the work by Max Wertheimer in the 1920s [Wer22, Wer23]. According to this fundamental idea the human visual system processes visual stimuli by grouping individual percepts to yield meaningful compositions. The underlying process of perceptual organization [Low85, Gol89] exhibits the principle law of Prägnanz, meaning that the “best, simplest, and most stable” groupings, [BGG96, p.109] are preferred. These properties are necessary to obtain compositions that can be memorized efficiently and recognized robustly. However, Prägnanz is an abstract concept that needs a tangible foundation so that it can be actually applied. To this end Gestalt psychology has also proposed numerous Gestalt laws of organization [BGG96, Mur73] that are intended to entail Prägnanz. These laws represent composition rules [GPC02] that impose constraints on the types of admissible constructions, whereby most of the potential combinations are actually ruled out since they are rated meaningless. As a consequence, the space of all compositions is pruned significantly so that only relevant candidates of high Prägnanz have to be considered.
2.3. Hierarchical Object Models and Compositionality

Gestalt Laws of Perceptual Organization

Gestalt psychology proposes a set of visual rules that guide the construction process of groupings and yield compositions of high Prägnanz. The Gestalt laws establish causal relationships between grouping constituents. The most important of them can be summarized as follows (other principles have been proposed, but they are not commonly accepted in psychology):

1. **Proximity**: Objects that are close to each other tend to be grouped together. An illustration is given in Figure 2.4(b). Here the circles are perceived to form horizontal rows.

2. **Similarity**: Objects with similar characteristics are grouped together. The set of attribute that can be used to establish similarities includes:

   (a) **Orientation**: Similarity of the spatial orientation of objects. The orientation is determined by prominent parts of an entity.

   (b) **Shape**: Similarity of the form of objects. Figure 2.4(c), which is perceived as vertical columns, demonstrates this principle.

   (c) **Symmetry**: Components that form a symmetrical composition tend to be grouped. Two contours that are similar when mirrored with respect to some axis are a common instance which this principle refers to. Figure 2.5(b) is perceived as three black patches since the resulting boundaries of each patch are symmetrical. The white area in between is observed as background. Figure 2.5(c) shows the same effects with switched colors.

Figure 2.3: Law of Prägnanz: (a) Original image; (b), (c), and (d) are possible groupings; (b) is the grouping of greatest Prägnanz, [Gol89, p.197].
Chapter 2. Key Modeling Decisions and State of the Art

Figure 2.4: Illustration of Gestalt laws. See text for details. (a) This scenario is ambiguous. (b) Proximity; (c) Similarity of shape, all from [Gol89, pp. 198, 200]. (d) Closure, [Low85, p. 23].

(d) **Color**: Similarity in color of the objects or their direct surroundings. Color information about the local neighborhood is especially relevant when grouping boundary contours. Obviously, similarity in *texture* can be used in the same way.

(e) **Common fate**: Similarity of the motion pattern of objects: A camouflaged animal can be spotted much easier when it moves than while it remains stationary. Another example is the perception of a swarm of birds as a greater whole.

3. **Good continuation**: Objects are grouped in a manner that preserves “smooth continuity” [BGG96, p. 108] rather than yielding sharp bends in the route of contours. According to this idea entities are seen as belonging together when they are arranged on straight lines or “smooth” curves. Obviously a lot of modeling is necessary to transfer this concept into an algorithmic form. However, this law is fundamental to the ability to bridge gaps in contours. The shapes in Figure 2.5(a) are perceived as two contours intersecting in the middle of the figure.

4. **Closure**: Groupings that produce closed figures, especially convex ones, are preferred. Figure 2.4(d) illustrates this preference, since the shapes are perceived as squares.

5. **Inclusiveness**: Groupings that unite a greater number of objects without becoming too complicated are preferable, although they might lose track of some details.

Evidently, the Gestalt laws are only vague principles of descriptive nature. The question of how far these rules have to be fulfilled in order to support a grouping
2.3. Hierarchical Object Models and Compositionality

Figure 2.5: Illustration of Gestalt laws. See text for details. (a) Good continuation, [BGG96, p. 108]. (b), (c) Symmetry, [DW79, p. 255].

is deferred by literature to individual experimentation, see for instance [BGG96, p. 107], [Mur73, p. 137]. In [OB03] and in the more comprehensive presentation of [Omm03] we have shown how various Gestalt laws can be combined in a common statistical framework. Based on this framework a compositional architecture for perceptual organization has been derived that produces perceptual groupings of high Prägnanz. In this thesis perceptual organization will be applied to restrict the space of composition candidates by filtering out irrelevant compositions of low Prägnanz.

2.3.4 Summary of the Characteristics of a Compositional Approach

As opposed to constellation models [FPZ03] that establish a joint model of very few, but highly distinctive parts, the compositional approach builds a hierarchy of intermediate image representations based on a small number of generic atomic parts that are, for that reason, not at all class specific. Other approaches such as Leibe and Schiele [LS04] have made it possible to incorporate large numbers of parts by using less complex shape models than the one of [FPZ03]. However, they are still depending on a large codebook of specific parts. The compositional approach pursues the goal to become independent of the existence of few highly specific but therefore fragile parts that serve as fingerprints of an object. Rather does a compositional approach automatically form relevant groupings of unspecific parts. The information that is necessary for discriminating object classes does not reside in individual parts but in the relations between them. The ultimate goal of a compositional approach is then to learn relevant compositional object structure with as little supervision as possible (as opposed to the manual selection of object parts in [FH00])—ideally without any information by the user regarding the compositional nature of the visual world.

In contrast to our main theme of learning a compositional architecture for object
Figure 2.6: An object representation is the result of a complex mapping that is a concatenation of several sub-processes which are applied to the physical object existing in real world. Additional tasks such as segmentation are not explicitly mentioned in this schema.

recognition, Jin and Geman [JG06] have presented a compositional system with manually built structure for license plate reading. Their study has focused on structural aspects of compositionality and explicitly excluded the question of learning such architectures. In their conclusion they put emphasis on the complexity of the future challenge of learning such a compositional model. This thesis deals with exactly this problem in the even less constraint case of large numbers of natural object classes.

### 2.4 Different Learning Paradigms

All recognition approaches can be divided into two broad categories, depending on the type of learning and classification method they utilize. To start with, the goal of classification in general is to map a representation $x \in \mathbb{R}^d$ of a real world object to an appropriate class label $y \in \{1, \ldots, k\}$. Just as a side note: the representation $x$ is already the result of a complex mapping that is a concatenation of a number of sub-processes such as image acquisition, preprocessing, feature extraction, and additional feature processing (see Figure 2.6 and Section 1.1).

There are two ways of obtaining the optimal class label $y$. Generative classifiers (also called informative classifiers) learn a model of the joint probability density function $p(x, y)$, or equivalently of $p(x|y)$ and $p(y)$. Classification is then performed by applying Bayes’ formula

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$  \hspace{1cm} (2.6)  \\
$$\propto p(x|y)p(y)$$  \hspace{1cm} (2.7)

and selecting the most likely label, i.e. the $y$ that maximizes $p(y|x)$. In contrast to this, discriminative classifiers estimate the posterior $p(y|x)$ directly. Alternatively, they can also learn a direct mapping $y = f(x)$ from inputs $x$ to labels $y$.

When comparing these two alternatives an obvious question arises: Why should
we care about the generative approach when there exists a much more direct way to solve the problem? Actually this is along the lines of Vapnik [Vap98] who says that

“one should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling the class distributions].”

Indeed, both theoretical and empirical investigations [NJ02] have at least presented indications that in the asymptotic case of large training sets discriminative methods yield lower error rates than generative ones. However, there also exist generative approaches (such as kernelized versions of discriminant analysis, cf. [RT01]) that focus on constructing an optimal class decision boundary and only implicitly establish models for the different classes.

An example of a visual object recognition system that learns from large volumes of data using a discriminative method is for instance the face classifier by Viola and Jones [VJ01, VJ04]. However, in the machine vision community generative approaches are nevertheless popular, e.g. [FPZ03, LS04, Sch04]. One reason for this popularity is that generative methods provide a straightforward way of dealing with missing or corrupted features by marginalizing the likelihood $p(x|y)$ over the affected dimensions of $x$. Moreover, there exists evidence [NJ02] indicating that generative methods yield superior performance for small training sets. In the context of limited training data, prior knowledge becomes increasingly important and a generative approach provides an intuitive solution to incorporating such a prior. An example of a computer vision approach that exploits priors to learn object models from few training samples is given by Fei-Fei et al. [FFFP03]. Finally, an important property of generative models is that they allow to reverse the processing pipeline so that representations that are typical for learned object classes can be sampled and visualized. Such studies provide additional insights into the learned object models. Moreover, the generative approach renders it possible to infer missing object parts.

This comparison of generative and discriminative models shows that both have attractive properties. For that reason hybrid approaches have been studied. Jaakkola and Haussler [JH99] have investigated how a generative method can be transformed into a discriminative one using Fisher kernels. Holub et al. [HWP05] have then applied this approach to object recognition.

Finally, it should be noted that almost all current approaches in object recognition are of probabilistic nature. Papers on recognition that have influenced this way of modeling are [BWP98] and [AG98].
2.5 Graphical Models

When formulated as a statistical learning and inference problem, object recognition and other challenging tasks typically involve large numbers of random variables. As already discussed previously, the joint distribution of large sets of random variables becomes quickly intractable since the dimensionality of the state space increases exponentially with the number of variables. Complex models with many parameters entail two significant problems. First, learning such models requires statistical estimations in high dimensional parameter spaces which in turn demand massive training sets that are not available. Second, statistical inference becomes computationally intractable when many parameters are involved.

So how can we nevertheless deal with complex problems that involve models of many random variables? The key idea is to exploit independence properties that are inherently present in the problem. Probabilistic graphical models address this task by representing conditional independencies in an explicit manner [Lau96, Jor04, Cha91]. Therefore, graphs are used to decompose the set of all random variables into small subsets with only local interactions. As a results the joint probability distribution of all variables is simplified as it is divided into factors that only involve variables of the respective subset. This decomposition is crucial for making learning and inference feasible because it significantly reduces the dimensionality of the state space by dividing it up into several smaller subspaces.

2.5.1 Graph Theoretical Fundamentals

Subsequently, graph theory will be discussed briefly as it lays the foundation for graphical models. A more detailed coverage of this topic is for instance given in [Lau96]. A graph $G = (V, E)$ consists of a set of vertices or nodes $V$ and a set of edges $E \subset V \times V$ between them. In directed graphs, an edge $(i, j) \in E$ connects two distinct vertices, a parent node $i \in V$ to its child $j \in V$. The connection between the two nodes is depicted by an arrow (see Figure 3.5). For each node $j$ in the graph this defines a set of its parent vertices

$$\Gamma(j) := \{i \in V : (i, j) \in E\}. \quad (2.8)$$

For undirected graphs an edge $(i, j) \in E$ implies that $(j, i) \in E$ and an edge is visually represented by a connecting line. In this case the set $\Gamma(j)$ is referred to as the set of neighbors of vertex $j$. Finally, there exist chain graphs that are a hybrid model with directed and undirected edges (see [Lau96]).
2.5. Graphical Models

2.5.2 Undirected Graphical Models

Markov Random Fields

Graphical models represent probability distributions

\[ p(X = x) = p(X_1 = x_1, \ldots, X_N = x_N) \]  

over random variables \( X_i \) with state spaces \( \mathcal{X}_i \) (and realizations \( x_i \in \mathcal{X}_i \)) by graphs \( G = (V, E) \). Therefore, each vertex \( i \in V \) is associated to a random variable \( X_i \). The sparse set of edges does then represent conditional dependencies between random variables. In the case of undirected edges, the graph defines a Markov random field [Win03]. The sparsity of edge connections in this graph is directly related to the so-called local Markov property of the Markov random field. This property states that each variable is independent of all other variables conditioned on its neighbors \( X_{\Gamma(i)} := \{X_j : j \in \Gamma(i)\} \):

\[ p(x_i | x_{\{x_i\}}) = p(x_i | x_{\Gamma(i)}) . \]  

(2.10)

Here the shorthand notation \( p(x_i) = p(X_i = x_i) \) has been used. Moreover \( x\backslash\{x_i\} \) are all random variables, except \( x_i \). Finally, for vertices \( i \) without parents (\( \Gamma(i) = \emptyset \)) the conditionals are defined as \( p(x_i | x_{\Gamma(i)}) = p(x_i) \). By exploiting the Markov property any positive joint distribution can be split up into a product of simpler clique potentials. Within an undirected graph, a clique is a fully connected subgraph and a clique is said to be maximal if no vertex can be added without violating the clique property. Let \( \mathcal{C} \subseteq \mathcal{P}(V) \) denote the set of all maximal cliques of \( G \). Moreover, for a clique \( C \in \mathcal{C} \), \( Z_C := \{X_i : i \in C\} \) is the set of random variables that correspond to the nodes in the clique. The joint distribution can then be written as a product of nonnegative potential functions \( \psi_C \) that are associated with the individual cliques \( C \),

\[ p(x) = p(x_1, \ldots, x_N) \propto \prod_{C \in \mathcal{C}} \psi_C(z_C) . \]  

(2.11)

Factor Graphs

A factor graph [Fre98] is a bipartite graph \( G = (V_v \cup V_f, E) \), with \( V_v \cap V_f = \emptyset \) and \( E \subseteq V_v \times V_f \). The graph has a variable node \( i \in V_v \) for each random variable \( X_i \) and factor nodes (also called function nodes) \( \varphi \in V_f \) for local functions \( f_\varphi \). Each local function \( f_\varphi \) is a mapping from a subset of all random variables \( Z_\varphi \subseteq \mathcal{X} \) into a
specific codomain—we will focus on functions that map into \([0, 1]\), that is
\[
\begin{align*}
  f_\varphi : \mathcal{Z}_\varphi &\longrightarrow [0, 1] \\
  z_\varphi &\longmapsto f_\varphi(z_\varphi).
\end{align*}
\] (2.12)

Each local function is connected to the variables on which it depends, i.e. there is an edge that connects a variable node \(i\) with a factor node \(\varphi\) if and only if \(X_i\) is an argument of \(f_\varphi\) (see Figure 3.8(b) for an illustration of a factor graph). A factor graph does then represent the joint probability distribution by a product of local functions
\[
p(x_1, \ldots, x_N) \propto \prod_{\varphi \in \mathcal{V}} f_\varphi(z_\varphi).
\] (2.13)

### 2.5.3 Directed Bayesian Networks

Bayesian networks [Pea88, Jen01] are represented by directed acyclic graphs \(G = (\mathcal{V}, \mathcal{E})\). Consequently, there exist no closed paths when traversing along directed edges. Moreover, edges establish a partial ordering of the random variables [Pea88]. Similar to undirected Markov random fields the joint distribution can again be decomposed. In the undirected random fields potentials over cliques, which are defined by the Markov property, are used. In the case of Bayesian networks conditional probabilities over parent nodes take the place of the clique potentials. The joint probability can then be decomposed, giving
\[
p(x_1, \ldots, x_N) = \prod_{i \in \mathcal{V}} p\left(x_i \mid x_{\Gamma(i)}\right).
\] (2.14)

As a consequence, Bayesian networks define a generative process for efficient sampling of realizations \(x \in \mathcal{X}\). This process is causal since it follows the partial ordering of random variables defined by the directed edges. First samples have to be drawn from nodes without parents before sampling from the conditionals that depend on child nodes.

### 2.5.4 Learning and Inference in Graphical Models

In later chapters we will then discuss how statistical models for real-world object categories (in particular Bayesian networks) can be learned from weakly labeled training data. A general overview over learning in the context of graphical models is given in [Jor99] and in particular in [Hec99]. In the prediction phase, a previously trained model is then used to classify novel data. Therefore, statistical inference has to be
performed on the basis of the underlying graphical model. By this process evidence at observed variables (depicted as shaded nodes in the graphical model of Figure 3.5) is propagated along edges to estimate the posterior marginals, the so called beliefs, of hidden variables. The propagation of beliefs is called message passing since it exchanges information about the distributions of random variables over their connecting edges. In the concrete application of object recognition the observables are image features. This evidence is then to be used to infer compositions. The intermediate compositional representation layers are in turn used to infer object location and object category. In later chapters we will investigate how statistical inference can be formulated as a computationally feasible problem by exploiting the independencies that are expressed by the graphical model. A prominent message passing algorithm for efficient computation of posterior marginals is belief propagation [Pea88] which will be presented in Section 3.4.4.
3

Atomic Parts for Compositionality: Localized Feature Histograms

Compositional object models decompose complex object representations into compositions of simpler sub-structures. Eventually, such a hierarchy is based on an initial layer of elementary features. Subsequently, we will investigate which crucial requirements should be met by a feature descriptor. Based on these properties a descriptor for the atomic parts of the compositional hierarchy will be derived as proposed in [OB05, OB06]. Thereafter, perceptual organization will be applied to form compositions of parts and a shape model for coupling arbitrary numbers of compositions will be presented.

3.1 Requirements on Local Descriptors

The initial layer of the compositional hierarchy consists of parts that appear as atoms to the composition system, i.e. they are not further decomposed and constitute the most basic entities of the hierarchical representation system. These parts capture local properties of an object by summarizing regions of an image. Let us now take a look at crucial properties a descriptors of atomic compositional parts should exhibit.

Descriptiveness: Obviously, a local feature should describe the characteristics of the object in its support region. All information that is not captured in this region can hardly be reacquired later on.

Good Localization: The individual atomic parts cover local image regions. In a later stage a shape model combines all the local evidence into a single global object
hypothesis. Consequently, the better the individual atomic parts are localized the greater is the accuracy of the spatial model. Conversely, features that spread out over large image regions hinder a precise localization of characteristic object details.

Robustness to Local Image Changes: Local regions of an image can be corrupted by noise or other environmental factors during the imaging process. A part-based model can in principle tolerate the corruption of individual parts. This is one of the main advantages over a wholistic representation that is based on only a single descriptor for a whole image. However, changes in individual pixels should not alter a local feature, significantly. Otherwise too many parts will become void.

Low Dimensionality: Image features form the basis for the object models that are to be learned. It is therefore favorable to use concise descriptors that yield a low dimensional feature space. Otherwise learning will become statistically intractable. Moreover, this helps to keep the demands on the number of training samples as low as possible.

Shareable among Object Categories: Another way of keeping the demanded training sets as small as possible is to share atomic parts among object classes [TMF04]. As a consequence, the limited training data of multiple classes can be combined when learning the statistics of local parts.

When reviewing these properties it becomes obvious that they aim in orthogonal directions and they have effects that are mutually working against each other. On the one hand features should be descriptive and extract as much information from the image as possible. On the other hand it is desirable for features to be invariant to local image changes. Obviously, invariance is incorporated at the expense of description power, because invariance means that certain properties are actually not captured by the feature. Similarly, the ability to localize features means that their spatial support has to be restricted which again increases the liability to noise since region statistics are estimated from a small number of observations (pixels).

Two classical extremes are appearance patches on the one hand and global image histograms on the other. Appearance patches (see Section 2.1.2) are basically subsampled image patches that are vector quantized using normalized grayscale correlation to obtain invariance to illumination changes of a patch as a whole. However, since the resulting invariances are just established by a global subsampling and intensity normalization, translations or local alterations still have a disproportionately high influence on the complete feature. Moreover, due to the low-pass filtering, only
3.1. Requirements on Local Descriptors

Figure 3.1: Objects from the car category of the ETH-80 database [LS].

Information on the strongest edges is preserved while the remaining patch content is blurred. An alternative approach at the other end of the modeling spectrum is that of using histograms over complete images (see Section 2.2.2). Thereby, utmost invariance with respect to changes of individual pixels can be obtained. However, no spatial information is retained. In conclusion, the former approach facilitates almost perfect localization while the latter one offers maximal invariance with respect to local distortions. We aim at a representation whose invariance properties are transparently adjusted between these two classical extremes and add the specificity lost by invariance through the relations incorporated in compositions.

We then have to find a trade-off between all of the individual requirements. Ideally, this compromise is chosen in such a way that the representation discards image characteristics that are more likely to be caused by the environment or the imaging process than by the object itself. Such a descriptor concentrates on only few image characteristics that are actually valuable for the task of discriminating object classes from each other. Thereby all the individual requirements are optimally combined.

Finally, the compositional approach offers an important opportunity to add the specificity that is lacking in the individual parts due to invariance. This is achieved by establishing relationships between parts and combining them to form compositions of greater descriptive power. Consider for example two parts that lack spatial information about the regions they cover. By capturing the spatial relationship between those regions a composition can add additional specificity to the information that is already present in its constituent parts.
Figure 3.2: A subset of the 2,519 appearance patch prototypes that have been formed to represent the 10 cars from Figure 3.1. This illustration, which is taken from [LS03], shows the cluster prototypes with their corresponding patches.

3.2 Deriving a Descriptor for Atomic Parts of a Composition System

Let us now analyze the representation found by appearance patches for a simple toy problem from [LS03] consisting of only a single object category, namely 10 toy cars from the ETH-80 dataset [LS] shown in Figure 3.1. To form appearance patch prototypes a set of 160 images corresponding to 16 views around the equator of each object are collected. From these images, Leibe and Schiele [LS03] extract 8,269 patches of size $25 \times 25$ pixels. These patches are then clustered to form a massive codebook of 2,519 prototypes, just for this single category. Of course, an initial representation of such high specificity cannot be generalized to large numbers of categories, but it is still interesting to observe the limitations of this representation. Although this codebook, which is represented in Figure 3.2, is totally category specific it nevertheless mostly captures only the strong edges inside patches. This effect, which will increase dramatically for larger numbers of categories, is not surprising as subsampling and low-pass filtering (for reducing aliasing effects due to subsampling) are discarding texture patterns of high frequency. However, in appearance patches the edges are represented on a 2-D pixel grid which introduces a lot of redundancies: relevant boundary edges are typically smooth contours so that adjacent pixels on the contour are highly correlated.

So can the complexity of the representation be reduced by exploiting the redundancies? A straightforward way is to describe edge curves by the edge normal and by the strength of the discontinuity as we will do in the following.
3.2. Deriving a Descriptor for Atomic Parts of a Composition System

3.2.1 Canny Edge Detection

Before edge curves can be represented by a local descriptor they have to be detected. We therefore follow John Canny’s approach to optimal edge detection which is reviewed in this section. In [Can86] he derives an edge detector based on several performance criteria—a necessary correction for the localization criterion is given in [Td90]. The resulting detector can be said to be optimal for these goals. The criteria Canny uses to detect step edges can be summarized as follows:

1. **Good detection:** The edge detector should be robust to noise. The probabilities of missing real edge points and falsely marking points that do not lie on an edge should both be low. This criterion can be formalized [Can86, Td90] by requiring the edge detector to maximize the output signal-to-noise ratio (SNR) [Hor86] for a given input signal-to-noise ratio.

2. **Good localization:** The detected edge points should be as close as possible to the true edge. This criterion is formalized in [Td90] by requiring that the location of the maximal detector response to an edge exhibits low variance.

3. **Uniqueness of response:** The edge detector should not produce multiple responses to a single edge. This criterion can be formalized by requiring the filter which performs edge detection to have a small spatial width [Td90].

Obviously, the output signal-to-noise ratio can be increased by low pass filtering the input signal. This would, however, increase the spatial width of the overall filter that performs edge detection, as described in [Td90]. Therefore a trade-off between the first and third criterion has to be found. By using additional necessary properties of a filter that performs edge detection (see [Td90]) Tagare and deFigueiredo come to the conclusion that the **derivative of the Gaussian**—a Gaussian smoothing of the image followed by a first derivative—is the optimal detector, given the aforementioned criteria.

**Components of the Canny Edge Detector**

Let \( I \) denote an image consisting of three color channels for red, green, and blue,

\[
I(x, y) = (I_r(x, y), I_g(x, y), I_b(x, y))^T.
\]  

(3.1)

Moreover, \( \hat{I} \) denotes the grayscale version of that image, i.e. its brightnesses. As an initial process of edge detection, the brightnesses \( \hat{I} \) are smoothed by a (discrete)
convolution—denoted by the \(*\) operator—with a two-dimensional Gaussian \(G_c\). The strength of this smoothing is determined by \(\sigma_c\) (in [pixel]),

\[
G_c(x, y) := \frac{1}{\sqrt{2\pi}\sigma_c} \exp \left[ -\frac{x^2 + y^2}{2\sigma^2_c} \right].
\] (3.2)

Thereafter, the gradient of the smoothing, \(\nabla(G_c \ast \hat{I})\), is computed on a discrete grid of image coordinates. Canny [Can86] estimates the normal \(\tilde{n}_c\) to the tangent of the edge contour by

\[
\tilde{n}_c := \frac{n_c}{\|n_c\|_2}, \quad n_c := \nabla \left( G_c \ast \hat{I} \right).
\] (3.3)

Potential edge points are local maxima of the gradient magnitude in the direction of the gradient, \(\tilde{n}_c\). The implementation of the resulting technique, which is called non-maximum suppression, examines the neighbors in gradient direction to both sides of a candidate edge point. This candidate point is only marked as an edge point if its gradient magnitude is greater than that of the neighbors, i.e.

\[
\frac{\partial^2}{\partial \tilde{n}_c^2} G_c \ast \hat{I} = 0.
\] (3.4)

Moreover, the neighbors surrounding the marked edge point are removed by setting their gradient magnitude to zero.

The eight direct neighbors of each marked edge point that is not removed by non-maximum suppression are examined in a postprocessing step using a lower and a higher threshold on gradient magnitude, \(t_{\text{low}}\) and \(t_{\text{high}}\), respectively: Starting on an edge point whose magnitude of the gradient succeeds the higher threshold, all direct neighbors of this point are visited recursively. Each of these points is also marked as an edge point if its gradient has a magnitude that is greater than \(t_{\text{low}}\). The described procedure is called hysteresis thresholding, and it continues recursively by examining the neighbors of all marked edge points. As a result, small accidental gaps on edge contours are bridged.

### 3.2.2 Localized Feature Histograms

After detecting edge pixels in an image and connecting them to curves, each of the edge curves needs to be represented using a descriptor. Since edge contours are typically not straight lines a single normal vector will not be be sufficient for representation. One solution would be to use a chain code that represents the normals along a curve. However, such a descriptor has a high complexity and it is not robust with respect to local corruptions of the curve. Let us therefore consider a much more
3.2. Deriving a Descriptor for Atomic Parts of a Composition System

A concise representation that does however discard the spatial order of pixels on the curve. For that purpose, a curve is represented by a histogram over all its normal directions $\vec{n}_c$ and by an additional histogram over the gradient strengths $\|\vec{n}_c\|_2$. A compromise that captures at least the rough spatial structure of edge curves is then to compute separate histograms for local image regions.

Concretely, this is implemented as follows: To process an image, quadratic patches of size $20 \times 20$ pixels are extracted at interest points (obtained using the multiscale Harris detector of [MS04]). Each patch is divided up into four equally sized sub-patches, $l = 1, \ldots, 4$, with locations fixed relative to the patch center. In each of these subwindows marginal histograms over edge orientation $\vec{n}_c$ and edge strength $\|\vec{n}_c\|_2$ are computed at edge pixels (see Figure 3.3). Each of these histograms has four bins so that $e_i^{(o_l)} \in [0, 1]^4$ represents the edge orientations in the $l$-th subpatch of the $i$-th patch and $e_i^{(s_l)} \in [0, 1]^4$ is the corresponding edge strength histogram.

Furthermore, color information of image regions is to be represented in a coarse manner. First color quantization is performed using Floyd-Steinberg image dithering [FS76]. The color information in the $i$-th patch from above is then captured by an eight bin color histogram $e_i^{(c)} \in [0, 1]^8$. In total the image patch is represented by the edge histograms of its subpatches and the color histogram. As a result a 40-dimensional $(4 \times 4 + 4 \times 4 + 8 = 40)$ descriptor of localized feature histograms [OB05, OB06]

$$e_i := \left( e_i^{(o_1)}, \ldots, e_i^{(o_4)}, e_i^{(s_1)}, \ldots, e_i^{(s_4)}, e_i^{(c)} \right)^T \in [0, 1]^{40} \quad (3.5)$$

is obtained.

Compared with other histogram based features such as the 128-D SIFT descriptor or the even higher dimensional geometric blur (cf. Section 2.1.2) this is the lowest dimensional feature that is currently used for image categorization. This low dimensionality is indispensable and beneficial for a statistically feasible learning procedure, since statistical estimation in high dimensional feature spaces requires large training sets—in the literature this problem is commonly referred to as the *curse of dimensionality* [DHS01].

**Performance Evaluation:** In Section 7.7.2 we will investigate how much further the dimensionality of the localized feature histograms can be reduced and performance will be compared to the widely used SIFT features.
3.3 A Codebook of Atomic Compositional Parts

The localized feature histograms represent continuous measurements in the feature space taken at different image locations. Subsequently, a small codebook of atomic compositional parts is established. This set of atoms is shared by all object categories and serves as a common alphabet of low-level parts. It can therefore be compared to the fixed set of 26 letters that suffice to form arbitrary English words and sentences. This codebook is necessary when compositions of different numbers of parts are to be represented in a common feature space as it will be done in later chapters of this thesis (see Section 4.2).

As a prerequisite for learning a common codebook of atomic parts, feature vectors $\mathbf{e}_i$ are detected in all training images of all object classes. All the collected descriptors from all categories are jointly clustered using the $k$-means clustering algorithm [Mac67] yielding a $k$-dimensional codebook (typically less than 300 clusters). To improve robustness we select the best solution of 5 restarts of the clustering algorithm with start seeds that are randomly sampled from the training set. This set of prototypes is then used to vector quantize the atomic parts $\mathbf{e}_i$. However, to increase the robustness of the representation and to make it less susceptible to local minima in the expectation-maximization (EM) iterations [MK97] of $k$-means, parts are not merely represented by the closest codebook entry. Rather, each atomic part is described by a Gibbs distribution [Win03] over the codebook. Let $d_\nu(\mathbf{e}_i)$ denote the squared Euclidean distance of a measured feature $\mathbf{e}_i$ to a centroid $\mathbf{a}_\nu$,

$$d_\nu(\mathbf{e}_i) := \|\mathbf{e}_i - \mathbf{a}_\nu\|_2^2 . \quad (3.6)$$

The local descriptor is then represented by the following distribution of its cluster
Figure 3.4: Sketch of localized feature histograms.

assignment random variable $F_i$,

$$P(F_i = \nu | e_i) := Z(e_i)^{-1} \exp(-d_\nu(e_i)),$$
$$Z(e_i) := \sum_\nu \exp(-d_\nu(e_i)).$$ (3.7)

In other words, atomic parts $e_i$ are represented by probability distributions

$$(P(F_i = 1 | e_i), \ldots, P(F_i = k | e_i)) \in [0, 1]^k$$ (3.8)

over a $k$-dimensional codebook. Due to the low dimensionality of the features small codebooks of not more than 300 prototypes suffice for problems involving some 100 object categories. This is crucial to render density estimation and learning statistically feasible and it presents a sharp contrast to other approaches such as the one by Leibe and Schiele [Lei04] who use several thousand prototypes already for a single object category. An experimental investigation of the dimensionality of the features will be presented in Section 7.7.2.

3.4 A Recognition System based on Tuple Compositions

This section presents a first object categorization system that builds tuple compositions of atomic parts and integrates all parts and compositions in a single graphical model. In the recognition phase objects are then classified and localized in images.
### 3.4.1 Overview over the Processing Pipeline

The following sketches the recognition process (illustrated in Figure 3.4) and states how learning is involved: As described in Section 3.2.2 interest Points are detected using the detector of Mikolajczyk and Schmid [MS04] and regions around the interest points are described using localized feature histograms. Here we use quadratic regions with a side length of 10 to 20 pixel depending on the local scale estimate. The atomic parts are then vector quantized using a codebook of 300 clusters (see Section 3.3). In a next step, relations are detected between the atomic parts and they are being used to infer compositions. This inference is based on previously learned category specific grouping probabilities. All atomic parts and compositions enter into the same Bayesian network (presented in Figure 3.5) that couples all compositions by means of unobserved object shape and object categorization. Statistical inference based on all the established compositions yields then a classification. In addition to a maximum a-posteriori estimate for the category, a confidence in this classification is also computed. Finally, a learned model of object shapes is used to infer the object position in the image based on all compositions and the categorization hypothesis. This spatial probability distribution is in turn used to refine compositions and overall categorization. In the training phase the system is presented images together with an overall category label.

Due to the independence properties represented in the underlying Bayesian network from Figure 3.5, the categorization probabilities factorize and their computation is split into separate parts [Pea88]. Thereby, factorization of the probabilistic model is crucial for turning statistical inference into a feasible problem. Pearl [Pea88] introduces a message-passing algorithm that propagates evidence through polytrees to estimate the belief of unobserved variables, that is their posterior given the evidence. For some random variable $Y$ it is

$$BEL(y) := P(Y = y|E), \quad (3.9)$$

where $E$ denotes the observed evidence. Moreover, it has been widely advocated that this so called sum-product algorithm [KFL01] yields good approximations even for Bayesian networks with loops (cf. [MWJ99]).

### 3.4.2 Forming Tuple Compositions

Subsequently, relations between image regions are taken into account to incorporate additional evidence into the model that goes beyond the information that is captured
3.4. A Recognition System based on Tuple Compositions

Figure 3.5: Illustration of the Bayesian network. The evidence variables (shaded nodes) are \( E = \{ e_i \} \cup \{ r_{ij}, x_{ij} \} \). Where \(< i, j >\) denotes pairs of parts that are agglomerated into compositions—the dotted structure (plate notation [Jor04]) is therefore replicated for all these tuples, see text for details.

by individual local parts. From the various principles of perceptual organization, investigated in [OB03] for grouping processes, we apply \textit{good continuation} (see Section 2.3.3). This concept basically groups those entities together that form a common smooth contour. Hence we consider pairs of patches which lie on a common edge curve and measure their distance. To facilitate a later robust statistical estimation, this distance is discretized into three ranges (i.e. close/medium/far) which depend on a histogram over all these distances measured in the training data. The edge curves are obtained by performing Canny edge detection twice, once with a scale parameter that is the mean of the lower half of all scales detected at the interest points, and once with scale being equal to the mean of the upper half. The edge images are then added up. Now consider two patches at interest points \( i \) and \( j \). If they are observed to lie on the same contour and have a discretized gap of \( r_{ij} \) they establish the relation \( R_{ij} = r_{ij} \). The two parts are then forming a composition which we denote by \(< i, j >\), i.e.

\[
< i, j > \iff \text{part } i \& \text{part } j \text{ form a composition}.
\]  

Since the relations between image regions are observed, all the random variables \( R_{ij} \) enter as evidence into the Bayesian network in Figure 3.5. It should be emphasized that this is a sparse set of nodes—iff both patches lie on a common contour, such a random variable is introduced. In conclusion, a grouping based on such relations incorporates additional edge information that takes compositions beyond a proximity or co-occurrence grouping.

Based on the detected relations the following describes how compositions of parts are formed: Let the random variable \( C_{ij} \) represent a composition of the two image
regions $i, j$. Each such composition is of a certain category. That is, it has a certain state $c_{ij} \in \mathcal{L}_C$, where this state space of compositions is a superset of the set $\mathcal{L} = \{\text{face, airplane, …}\}$ of all categories for images, i.e. $\mathcal{L} \subset \mathcal{L}_C$. Consider the illustrating example of an image that is recognized to contain a motorbike. Then compositions representing subparts such as tires might be added to the set of allowed image categories. In our current implementation both sets differ by an additional category for background that we have incorporated for compositions, i.e. $\mathcal{L}_C = \mathcal{L} \cup \{\text{background}\}$.

The distribution of a composition of the two parts $i, j$ depends only on the representations of the involved parts, their relation, as well as on the categorization of the image, denoted by the random variable $C$, which state $c \in \mathcal{L}$. Thereby, the invariances represented in the Bayesian network from Figure 3.5 are reflected,

$$P(C_{ij} = c_{ij} | F_i = f_i, F_j = f_j, R_{ij} = r_{ij}, C = c).$$  \hspace{1cm} (3.11)

All $C_{ij}$ are assumed to be identically distributed and this distribution is split into

$$P(c_{ij} | f_i, f_j, r_{ij}, c) \propto P(c | c_{ij})P(c_{ij} | f_i, f_j, r_{ij})$$ \hspace{1cm} (3.12)

using Bayes’ formula and by dropping a normalization constant. The first factor models category confusion probabilities which are assumed to be independent from features and relations ($P(c | c_{ij}) = P(c | c_{ij}, f_i, f_j, r_{ij})$) when compositions $c_{ij}$ are given. With no further assumptions on categories, the following choice is made,

$$P(c | c_{ij}) = \begin{cases} |\mathcal{L}|^{-1}, & \text{if } c_{ij} = \text{background} \\ \eta, & \text{if } c = c_{ij} \\ 1 - \eta, & \text{otherwise}. \end{cases} \hspace{1cm} (3.13)$$

In [OB05] $\eta$ is for simplicity set to one. The second distribution in (3.12) is the categorization probability of compositions: The underlying non-parametric model is obtained in the learning stage by processing all the training images as follows: for each detected grouping, the category label of the whole image is taken as $c_{ij}$ and the distribution is estimated from the empirical histogram of all observed compositions. Figure 3.6 and Figure 3.11 visualize the category beliefs of compositions for the different classes.

### 3.4.3 Modeling Object Shape

In the following, a model of the spatial configuration of object components is presented. This model couples object localization with classification so that shape
estimates can be used to refine image categorization. The shape of an object of a
given category is modeled by the displacement $s_{ij}$ of all of its components from its
center $x$. Letting $x_{ij}$ denote the location of a composition (the midpoint between its
components) detected in the training data, its shift is computed as

$$s_{ij} = x - x_{ij}.$$  \hspace{1cm} (3.14)
In the training phase images are given together with a category label $c$, but the object position in the images is unknown. The location of the object center in some training image $I$ from category $c$ is therefore estimated by

$$x = \frac{\sum_{<i,j> \in I} x_{ij} \cdot P(C_{ij} = c | f_i, f_j, r_{ij})}{\sum_{<i,j> \in I} P(C_{ij} = c | f_i, f_j, r_{ij})}.$$  

Consequently, $x$ is the center of gravity of all compositions which are weighted with the probability that they actually belong to the object of category $c$. The underlying assumption is that compositions are distributed around the object center according to a bivariate Gaussian model. The mean of this distribution is $x$. Moreover, all compositions have equal prior probability but different category posteriors $P(C_{ij} = c | f_i, f_j, r_{ij})$ which are used to weight them when computing the object center.

In the recognition phase the location of the object center is predicted by means of a Parzen window density estimation. The probability of a shift, given the features of a composition and the object category, is represented by the following non-parametric model

$$p(S = s | f_i, f_j, r_{ij}, c) = \frac{1}{N} \sum_{l=1}^{N} K_{\sigma_N} \left( \frac{s - s_{ij}^{(l)}}{\sigma_N} \right).$$

Here $K_{\sigma}$ is a Gaussian kernel function with diagonal covariance matrix $\Sigma = \sigma \cdot I$. Moreover, $s_{ij}^{(l)}$ is the $l$-th shift vector found in the training data for a composition of parts represented by $(f_i, f_j, r_{ij})$. Note that the relations $R_{ij}$ have been discretized into three states (i.e. close/medium/far) in Section 3.4.2. The number of shift vectors observed for such a composition in the training set is denoted $N=(f_i, f_j, r_{ij})$. Therefore, the spatial density of the object center conditioned on a single composition is

$$p(X = x | f_i, f_j, r_{ij}, c, x_{ij}) = p(S = x - x_{ij} | f_i, f_j, r_{ij}, c).$$

Using this equation the conditional probability to observe a composition at location $x_{ij}$ can be written as

$$p(x_{ij} | f_i, f_j, r_{ij}, c, x) \propto p(S = x - x_{ij} | f_i, f_j, r_{ij}, c) p(x_{ij} | f_i, f_j, r_{ij}, c).$$

To simplify the representation in the graphical model the locations $x_{ij}$ are discretized on a regular $10 \times 10$ grid. The latter term is then approximated during learning by histogramming over the observed positions of compositions in the training data. Figure 3.7 gives an example of object detection.
3.4. A Recognition System based on Tuple Compositions

3.4.4 Inference of Image Categorization

During recognition, loopy belief propagation is performed using the evidence \( E = \{e_i\}_i \cup \{r_{ij}, x_{ij}\}_{i,j} \) to recognize the object in the scene, i.e. we are interested in the belief of the random variable \( C \). Belief propagation simplifies the complex problem of optimizing a marginalization of the joint distribution over all model variables. This calculation is rendered possible by using the independence properties represented in the graphical model and taking only the resulting local interactions into account. To simplify the computation scheme we transform the Bayesian network from Figure 3.5 into the factor graph (cf. [KFL01]) displayed in Figure 3.8(a). Function nodes represent the conditional probability distributions, whereas the remaining variable nodes correspond to the random variables of the Bayes net. To propagate beliefs, each vertex in this graph has to compute and send messages to its neighbors as follows: Consider some variable node \( v \) that has function node neighbors \( F_v \) and \( F_{w_1}, \ldots, F_{w_m} \) as depicted in Figure 3.8(b). Adjacent to each \( F_{w_i} \) is again some variable node \( w_i \) and \( F_v \) has variable node neighbors \( v \) and \( u_1, \ldots, u_m \). Now an unobserved variable sends messages to its function node neighbors by taking all the incoming messages...
Figure 3.8: (a) Conversion of the Bayesian network from Figure 3.5 into a factor graph representation. The function nodes $\mathcal{F}_i$ represent the posterior of the corresponding random variable, e.g. $\mathcal{F}_{f_1} = P(f_1|e_1)$, see text for details. (b) A simple factor graph used for illustrating belief propagation.

into account [KFL01],

$$\mu_{v \rightarrow \mathcal{F}_v}(v) := \prod_i \mu_{\mathcal{F}_{w_i} \rightarrow v}(v).$$  \hfill (3.19)

If $v$ is an evidence variable that is observed to be in state $v'$ then this message is just $\mu_{v \rightarrow \mathcal{F}_v}(v) = 1\{v = v'\}$, where $1\{\cdot\}$ denotes the characteristic function. Moreover a function node sends the following messages to its neighbors

$$\mu_{\mathcal{F}_v \rightarrow v}(v) := \sum_{u_1, \ldots, u_m} \mathcal{F}_v(v, u_1, \ldots, u_m) \prod_j \mu_{u_j \rightarrow \mathcal{F}_v}(u_j).$$  \hfill (3.20)

The belief of $v$ given all the present evidence $E$ is then the product of all of its incoming messages

$$P(v|E) \propto \mu_{\mathcal{F}_v \rightarrow v}(v) \prod_i \mu_{\mathcal{F}_{w_i} \rightarrow v}(v).$$  \hfill (3.21)

In conclusion, the presented architecture propagates beliefs not only in a bottom-up manner from the observed image features to infer categorization and object location. The system also propagates information backwards in a top-down fashion from object localization and categorization to composition and part hypotheses, $c_{ij}$ and $f_i$ respectively. While the bottom-up feature-driven and the top-down category model driven updates are performed concurrently, hypotheses get improved by finding those which have optimal mutual agreement.
3.5 Evaluation of Tuple Compositions

The proposed architecture based on tuple compositions is evaluated on the Caltech-4 image database [Cal] (4 object classes: airplane, car, face, and motorbike). During learning, some 700 images are presented to the system together with the image category labels as the only additional information. The test scenario is then to classify previously unknown images as belonging to one of the categories. Moreover, a confidence in this categorization is returned.

In order to evaluate the gain of compositionality in categorization, we first investigate a simpler model. It is based on the same image representation but with neither compositions nor a shape model (see Figure 3.9). Therefore the categorizations \( c_{ij} \) of compositions are replaced by the classification \( c_i \) of single parts, where \( P(c_i|f_i, c) \) is empirically estimated from the training data in the same way as the \( c_{ij} \) in Section 3.4.2. Figure 3.10 displays the resulting category confusion matrix. The confidence in a categorization of class \( c \) (shown on the x-axis) of images with a given
Chapter 3. Atomic Parts for Compositionality

Figure 3.11: Categorization belief of compositions. For each composition $c_{ij}$, (b) displays $P(C_{ij} = \text{car} \mid E)$ at the position of patches $i$ and $j$. See Figure 3.6 for details. (c) Shows the posterior for class face, (d) for motorbike, (e) for airplane, and (f) for background. Where the last category facilitates a figure-ground segregation of compositions. (g) Shows the regions that support categorizing the image as car.

The ground truth category (shown on the y-axis) is visualized in this figure. Therefore a row represents the beliefs of the different categories for one test image. This model achieves an overall correct classification rate of 82.5%. However, the categorization confidence is quite low.

Subsequently, this simple model is to be compared with the full approach outlined in Section 3.4. Figure 3.12 displays the category confusion matrix of the model that is based on compositions and their spatial arrangement. When comparing the two...
3.5. Evaluation of Tuple Compositions

Figure 3.12: Category confusion matrix for categorization based on the full model with compositionality and shape. This approach achieves an overall classification rate of 91.7% and has a significantly higher confidence than the previous one. Considering only the last three categories a recognition rate of 94.4% is achieved (see text for details). Compare this with the 93.0% reported in [FPZ03] for the same three categories.

plots it becomes evident that the system with compositionality and shape achieves a significantly increased confidence in the correct categorizations. Moreover, the recognition rate has increased to 91.7%. This illustrates that the relations used for compositions add important information to that already present in the individual parts.

As one can see, most of the error results from falsely classified car images. This is due to the fact that the interest point detector returns only very few votes for the large homogeneous parts of the body of a car. Most detections are in the background or at the outline of the vehicle. This is also apparent in the illustration of compositions in Figure 3.11. Although for this specific dataset the background features would provide good indications for the presence of cars, it is generally not desirable to introduce such dependencies as they are to a great deal database specific and would very likely lead to an overfitting to this image collection. In subsequent chapters we will therefore apply feature selection and investigate strategies for removing irrelevant compositions. When leaving out the car category and considering only the remaining three ones, the approach of Section 3.4 that was originally proposed in [OB05] achieves an overall recognition rate of 94.4%. Previously Fergus et al. [FPZ03] have reported a performance of 93.0% for these three categories. In the meantime, Lazebnik et al. [LSP05] have investigated part-based models of different complexity and based them on complex, category specific features [LSP04]. Due to
the specificity of features for the small set of categories in Caltech-4 their complex models could not improve the performance they achieved with a simple naive Bayes model. In summary their overall recognition rates for these categories are roughly situated between 95% and 98%.
Sharing Compositions among Categories

The object model presented in the previous chapter has utilized the full potential of compositionality only partially. So far the model is restricted to tuple compositions. Subsequently, the compositional object model from Section 3.4 is to be extended in several ways: As opposed to the tuples of parts from the previous chapter the following will investigate how compositions of arbitrary many parts can be formed. Moreover, the goal is to learn a comparably small set of compositions that are shared by the object categories.

Learning compositions is then guided by three modeling decisions: (i) Firstly, it has to be determined which parts to group to form potential candidate compositions. Here we follow the principles of perceptual organization that have been presented in Section 2.3.3. (ii) Secondly, we aim at learning a comparably small set of compositions so that estimating category statistics on the training data becomes feasible. Therefore, the system cannot afford to learn compositions that are observed only rarely in the visual world, i.e. compositions with a low prior. (iii) Thirdly, each composition should be valuable for the task of discriminating sets of categories from another—not necessarily one category from all others. Compositions should be discarded whenever they represent background that is present in many different categories, or in case they capture object specificities that are only present in individual instances of a category. The discriminative relevance of a composition is estimated by the entropy of the category posterior distributions given the composition.

Based on these criteria a single objective function will be derived that measures
the relevance of compositions [OB06]. The underlying training is performed in a
weakly supervised manner using only category labels for whole images.

4.1 Overview over the Processing Pipeline

The model can be best explained by first considering recognition, see Figure 4.1.
Given a novel image, salient image regions are detected using the interest point
detector of [MS04] before describing local regions around these points using localized
feature histograms. This extraction of atomic parts proceeds as described in Section
3.2.2 and Section 3.4.1 and yields part descriptors $e_i$.

In a next step a perceptual grouping of these local part descriptors is conducted to
obtain a set of possible candidate compositions. This grouping leads to a sparse image
representation based on (probably overlapping) subregions, where each candidate
represents an agglomeration of local parts. Consecutively, composition candidates
have to be encoded. Therefore all detected atomic part descriptors are represented as
probability distributions over a codebook which is obtained using vector quantization
in the learning stage (see Section 3.3). This codebook of atomic parts is shared by
all object classes and it models locally typical configurations of the categories. A
composition is then represented as a mixture distribution of all its part distributions.
4.1. Overview over the Processing Pipeline

In a next stage relevant compositions have to be selected, discarding irrelevant candidates that represent distractors such as background clutter. The set of relevant compositions has to be computed in the learning phase from the training data in a weakly supervised manner (see Figure 4.2). As intermediate compositional representations should have limited description length, this learning obeys the following rationale: (i) Firstly, we aim at a set of compositions that occur frequently in the visual world of the categories under consideration. Such compositions are likely to be reusable for representing novel object instances. For that purpose all composition candidates found in all the training images are clustered and the prior assignment probabilities of candidates to these prototypes are estimated. (ii) Secondly, relevant compositions have to support the discrimination of sets of categories from another. Clutter that is present in many different categories or configurations that are only observed in few instances of a category are to be discarded to reduce the model complexity. In order to find a relevance measure the category posteriors of compositions are learned from the training data. The relevance of a composition for discriminating categories is then estimated by the entropy of its category posterior. Finally, a single cost function is obtained by combining the reusability criterion with the criterion that measures the ability of compositions to discriminate categories from another. The resulting cost function guides the selection of relevant compositions.

After discarding the irrelevant compositions from a new test image, the image category has to be inferred based on all the remaining relevant compositions. These compositions are spatially coupled by using a shape model.
4.2 Establishing and Representing Candidate Compositions

Given all detected local part descriptors $e_i$ in an image (see Section 4.1), the categorization system follows the principles of perceptual organization by grouping parts using Gestalt laws to search for possible candidates of compositions. For the sake of simplicity, this chapter presents an approach that uses only the grouping principle of *proximity*. Chapter 5 will then investigate more complex grouping strategies. From the set of all parts detected in an image, a subset of 30 is randomly selected. Each of these parts is then grouped with neighboring parts that are not farther away than 60-100 pixel (depending on the local scale estimate of the interest point detector). Consequently this set $\mathcal{G}$ of compositions sparsely covers salient image regions.

After forming candidate compositions they also have to be represented using a descriptor. As compositions are having different numbers of parts the descriptors of their constituents cannot be merely concatenated to a joint descriptor as has been done for the tuple compositions in Section 3.4.2. As a result compositions with different numbers of parts would live in unrelated feature spaces so that no common model could be learned. Moreover, the resulting descriptors would have a high dimensionality and it is not obvious how a robust ordering of the constituents, which would be necessary for a vector representation, can be obtained.

Let us therefore follow a different approach and represent compositions by probability distributions over the $k$-dimensional codebook of atomic parts from Section 3.3. Since parts are the constituents of compositions, the following briefly reviews the part representation introduced in the last chapter. Each part $e_i$ is represented by the Gibbs distribution (3.7)

$$P(F_i = \nu | e_i) := \frac{\exp(-d_{\nu}(e_i))}{\sum_{\nu} \exp(-d_{\nu}(e_i))}.$$  \hspace{1cm} (4.1)

Thus atomic parts $e_i$ are represented by probability distributions $(P(F_i = 1|e_i),\ldots,P(F_i = k|e_i)) \in [0,1]^k$ over the $k$-dimensional part codebook. Let $\Gamma_j = \{e_1,\ldots,e_m\}$ denote the grouping of parts $e_1,\ldots,e_m$. The number of constituents $|\Gamma_j| = m$ is not predefined and can be different for each composition. It depends on how many parts the grouping algorithm can combine into composition in a certain region of an image. The multivariate random variable $G_j$ does then represent the composition consisting of these parts. Realizations $g_j \in [0,1]^k$ of this random variable are again distributions over the $k$-dimensional part codebook. The distribution that represents
a composition $g_j$ is a mixture of the distributions of all the parts $e_i \in \Gamma_j$,

$$g_j = \sum_{e_i \in \Gamma_j} \frac{1}{|\Gamma_j|} \left( P(F_i = 1|e_i), \ldots, P(F_i = k|e_i) \right)^\top.$$  

(4.2)

This representation is a bag-of-parts model—a histogram over the $k$ codebook vectors.

Finally, each of the $k$ dimensions is independently standardized to zero mean and unit variance across the whole training set, giving z-scores. This mixture model exhibits several advantageous properties. First, it is robust with respect to missed or corrupted individual local parts. Second, it is invariant to shifts of the local parts in the composition. Such shifts can for instance result from local fluctuations of interest points (salient points cannot be expected to mark exactly the same spot on different objects of the same category) or due to perspective distortions during the imaging process. Third, the mixture model exhibits low dimensionality irrespective of the number of constituents that are grouped together.

### 4.3 Kernel Methods for Multi-Class Classification

As already stated in the introduction, object recognition involves a classification task where a function has to be learned that maps visual representations to object categories, cf. Section 2.4. Let $f$ denote the mapping from representations $x \in \mathcal{X} \subseteq \mathbb{R}^d$ to class labels $y \in \mathcal{Y}$, that is

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

$$x \mapsto y = f(x).$$  

(4.3)

The label space $\mathcal{Y}$ is a discrete set of class labels such as $\mathcal{Y} = \{1, \ldots, m\}$ for $m$ classes. Subsequently, we will first consider the two-class case and use $\mathcal{Y} = \{-1, 1\}$ to simplify formulas. A simple linear classification function that solves the binary problem is

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

$$= \text{sign}\left(\sum_i w_i x_i + b\right).$$  

(4.4)

(4.5)

The classifier is defined by the parameters $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

Generally speaking, classifiers partition the feature space $\mathcal{X}$ into regions and associate each region with a class label. The resulting class boundaries in feature space
are also called decision surfaces. Typically, classes are not linearly separable. In other words, there exists no linear decision surface that separates all samples of the different classes from another. Is there nevertheless a way to make the favorably simple linear classifier applicable despite the nonlinearity of the problem? The solution is to transform the data with a nonlinear mapping \( \phi \) into a higher-dimensional space where classes are linearly separable,

\[
\phi : \mathcal{X} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^{d'}, \quad d' \gg d.
\]  

Instead of using a complex, nonlinear classifier to label data points \( \mathbf{x} \) directly, this approach uses a simple linear classifier to classify transformed samples \( \phi(\mathbf{x}) \). However, the problem has not yet been solved but only deferred to specifying an appropriate projection function \( \phi \). For \( d' \rightarrow \infty \) specifying \( \phi \) explicitly becomes impossible and even in the finite case the transformation can be very complex. However, if the classification function can be formulated in such a way that data points \( \mathbf{x}, \mathbf{x}' \in \mathcal{X} \) enter only in the form of scalar products \( \phi(\mathbf{x})^\top \phi(\mathbf{x}') \), then the whole problem can be simplified by substituting the scalar product and the mapping with a new function: There exist so called kernel functions, which can be shown to represent scalar products in high-dimensional spaces. Let there be \( N \) samples \( \mathbf{x}_1, \ldots, \mathbf{x}_N \). Then the symmetric function

\[
k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+
\]

is a kernel if and only if the matrix

\[
\mathbf{K} = \left( k(\mathbf{x}_i, \mathbf{x}_j) \right)_{i,j=1}^{N} \in \mathbb{R}_+^{N \times N}
\]

is positive semi-definite. Since a kernel function \( k \) represents a scalar product in some high-dimensional space \( \mathbb{R}^{d'} \) there exists a nonlinear transformation \( \phi : \mathcal{X} \rightarrow \mathbb{R}^{d'} \), so that

\[
\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} : k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}').
\]

Therefore, the scalar products in the high-dimensional space can be substituted by the kernel function which directly operates in the input space. As a consequence, kernels implicitly project the samples into the space \( \mathbb{R}^{d'} \). However, neither the projection nor the high-dimensional space have to be constructed or represented explicitly. The substitution that is expressed in (4.9) is called the kernel trick, also known as kernel substitution, and it lays the foundation for all kernel methods.

An in depth coverage of pattern classification is presented in [DHS01, Bis06, HTF01], and [SS02, CST00] focus especially on kernel methods.
4.3.1 Support Vector Machines

One of the most prominent kernel methods is the support vector machine (SVM). This discriminative approach to classification projects samples into a high-dimensional space where classes are separated using the linear decision rule of (4.4). In contrast to the perceptron which selects an arbitrary hyperplane between two classes, the SVM seeks to maximize a margin between the samples and the hyperplane [Vap98]. More precisely, the hyperplane is chosen so that the distance to the closest point in each class is maximal. This distance is called the margin.

Let there be training samples $x_1, \ldots, x_N \in \mathcal{X}$ with corresponding labels $y_1, \ldots, y_N \in \{-1, 1\}$. Moreover, there exists some unknown transformation $\phi: \mathcal{X} \rightarrow \mathbb{R}^d$, so that the $\phi(x_i)$ become linearly separable. Training a maximum margin classifier with decision function

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

(4.10)
corresponds to solving the optimization problem

$$\left( w^*, b^* \right) = \arg\min_{w, b} \|w\|_2,$$

s.t. $y_i \cdot \left( w^T \phi(x_i) + b \right) \geq 1, \quad \forall i \in \{1, \ldots, N\}.$

(4.11)

Here it has been exploited that maximizing the margin is equivalent to keeping the margin fixed and minimizing $\|w\|_2$. Moreover, the minimization of $\|w\|_2$ can be replaced by the equivalent problem of minimizing $\frac{1}{2}w^T w$. Thus the corresponding Lagrangian is

$$L(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i \left[ y_i \left( w^T \phi(x_i) + b \right) - 1 \right].$$

(4.12)

Now we differentiate with respect to $w$ and $b$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) = 0 \Leftrightarrow w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i),$$

(4.13)

$$\frac{\partial L(w, b, \alpha)}{\partial b} = - \sum_{i=1}^{N} \alpha_i y_i = 0.$$  

(4.14)

By substituting $w$ in (4.12) with (4.13), the corresponding dual representation is
obtained,

\[ L(w, b, \alpha) = \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j) \]

\[ - \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j) + \sum_{i=1}^{N} \alpha_i \]

\[ = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j). \] (4.16)

The functional (4.16) only depends on the dual parameter vector \( \alpha \). Moreover, the samples \( x_i \) appear only in scalar products after being projected with \( \phi \). Therefore, the kernel trick (4.9) can be applied to obtain the following dual problem [BV04] which requires quadratic optimization

\[ \alpha^* = \arg\max_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \]

s.t. \( \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \forall i : \alpha_i \geq 0 \). (4.17)

The solution for the primal problem (4.11) is given by the optimal hyperplane with parameters

\[ w^* = \sum_{i=1}^{N} \alpha_i^* y_i \phi(x_i), \] (4.18)

\[ b^* = \frac{1}{2} \left( \min_{i:y_i=1} (w^*)^\top \phi(x_i) \right) + \left( \max_{i:y_i=-1} (w^*)^\top \phi(x_i) \right). \] (4.19)

The Karush-Kuhn-Tucker complementary condition states that the optimal solution satisfies

\[ \alpha_i^* \left[ y_i \cdot \left( (w^*)^\top \phi(x_i) + b^* \right) - 1 \right] = 0. \] (4.20)

Consequently, the parameters \( \alpha_i^* \) that maximize the dual problem are only non-zero for training samples \( x_i \) that lie on the margin. Only these samples are actually used for establishing the optimal hyperplane in (4.18). For that reason they are called support vectors. Let \( \mathcal{SV} \) denote the set of indices \( i \) of all samples \( x_i \) that are support vectors. The resulting classifier is sparse since it only depends on these samples

\[ f_{\alpha^*, b^*}(x) = \sum_{i \in \mathcal{SV}} \alpha_i^* y_i k(x_i, x) + b^*. \] (4.21)
4.3. Kernel Methods for Multi-Class Classification

For an SVM with a given complexity (specified by the kernel) the training samples may not be separable. To avoid overfitting by requiring more complex kernel functions the concept of the *soft margin SVM* has been introduced. This classifier tolerates misclassified samples by introducing a *slack variable* \( \xi_i \geq 0 \) for each training sample \( x_i \). The slack variables measure the violation of the margin constraint, that is how far a sample lies on the wrong side of the margin. The derivation of the optimal parameters for this classifier follows along the lines of the hard margin SVM presented in this section. Details are presented in [CST00].

4.3.2 Nonlinear Kernel Discriminant Analysis

As described in Section 2.4, generative classifiers model the conditional class densities \( p(x|y) \) and the class priors \( p(y) \). Let \( p_{\theta_y}(x|y) \) denote parametrized models of the likelihoods for each class \( y \). The complete model of all \( m \) classes is then represented by the parameter vector \( \theta = (\theta_1, \ldots, \theta_m)^\top \). A maximum likelihood estimate of the model parameters is then obtained by maximizing the full log-likelihood of all training samples

\[
\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \log p_{\theta_y}(x_i, y_i) \quad (4.22)
\]

\[
= \arg\max_{\theta} \sum_{i=1}^{N} \left( \log p_{\theta_y}(x_i|y_i) + \log P_{\theta_y}(y_i) \right) . \quad (4.23)
\]

In *linear discriminant analysis* (LDA) [Fis36, McL92, DHS01] the classes are modeled by a parameterized multivariate Gaussian model with a common covariance matrix \( \Sigma \) for all classes,

\[
p_{\theta_y}(x|y) = \mathcal{N}(x; \mu_y, \Sigma) . \quad (4.24)
\]

To present a kernelized version of LDA we will first consider the two-class case with labels \( Y = \{-1, 1\} \). To classify a datapoint the log-ratio of the posteriors has to be compared to obtain a classification function

\[
f(x) = \text{sign} \left[ \log \frac{p(Y = +1|x)}{p(Y = -1|x)} \right] \quad (4.25)
\]

\[
= \text{sign} \left[ \log \frac{p_{\theta_{+1}}(x|Y = +1)}{p_{\theta_{-1}}(x|Y = -1)} + \log \frac{p_{\theta_{+1}}(Y = +1)}{p_{\theta_{-1}}(Y = -1)} \right] . \quad (4.26)
\]

Here Bayes’ formula has been applied. Now the Gaussian model for the likelihoods
Chapter 4. Sharing Compositions among Categories

(4.24) can be plugged into above equation (4.26) to obtain

\[ f(x) = \text{sign} \left[ (\mu_{+1} - \mu_{-1})^\top \Sigma^{-1} x \right. \]
\[ \left. - \frac{1}{2} (\mu_{+1} + \mu_{-1})^\top \Sigma^{-1} (\mu_{+1} - \mu_{-1}) \right) \]
\[ + \log \frac{p_{\theta_{+1}}(Y = +1)}{p_{\theta_{-1}}(Y = -1)}. \]

Therefore the resulting classification function has again linear form

\[ f(x) = \text{sign}(w^\top x). \] (4.28)

Here the offset \( b \) from (4.4) has been omitted to simplify notation. The rationale is that samples \( x \) can be augmented with an additional entry of one so that \( b \) is moved into an additional dimension of \( w \). The optimal solution \( w^* \) can then be found by posing LDA as a regression problem, cf. [Bis06]. Therefore, training samples \( x_i \) are regressed against the labels \( y_i \). The matrix

\[ X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{pmatrix} \] (4.29)

contains all the training samples as row vectors and the corresponding labels are summarized in the vector \( y = (y_1, \ldots, y_N)^\top \in \mathcal{Y}^N \). Then the optimal weight is given by the least squares solution

\[ w^* = \arg\min_w \|y - Xw\|_2^2. \] (4.30)

However, this approach selects the model with minimal residual regardless of the complexity of the model. The resulting classifier can therefore be prone to overfitting. In a Bayesian approach this problem is avoided by imposing priors on weights so that smooth discriminant functions are preferred. A Gaussian prior on the weights with diagonal covariance matrix \( \Sigma_w \propto \lambda^{-1} \mathbf{I} \) leads to a ridge regression model [HK70] that regularizes (4.30),

\[ w^* = \arg\min_w \|y - Xw\|_2^2 + \lambda w^\top w. \] (4.31)

Now the weight vector can be expanded using the samples by introducing coefficients \( \alpha_i \),

\[ w = \sum_{i=1}^N \alpha_i x_i = X^\top \alpha. \] (4.32)
4.3. Kernel Methods for Multi-Class Classification

In the next step this expansion is used to substitute the weights in (4.31). Moreover, the kernel function from (4.8) is employed for which it holds that \( K = XX^\top \). The optimization problem of (4.31) can then be reformulated in terms of the expansion coefficients \( \alpha_i \),

\[
\alpha^* = \arg\min_{\alpha} \| y - XX^\top \alpha \|^2_2 + \lambda (X^\top \alpha)^\top X^\top \alpha
\]

\[
= \arg\min_{\alpha} \| y - K\alpha \|^2_2 + \lambda \alpha^\top K\alpha
\]

\[
= \arg\min_{\alpha} y^\top y - 2y^\top K\alpha + \alpha^\top KK\alpha + \lambda \alpha^\top K\alpha.
\]

Differentiating the objective function with respect to \( \alpha \) yields the solution

\[
Ky = K (K + \lambda I) \alpha^*.
\]

The optimal coefficients \( \alpha^* \) fulfill

\[
\alpha^* = (K + \lambda I)^{-1} y.
\]

The inverse \( (K + \lambda I)^{-1} \) always exists for any \( \lambda > 0 \), since \( K = XX^\top \) has non-negative eigenvalues and becomes invertible by adding positive diagonal entries \( \lambda \).

Now the classification function (4.28) is extended to nonlinear problems by projecting samples with a transformation \( \phi \) into a high-dimensional space. Applying the weight expansion (4.32) yields the classification function of nonlinear kernel discriminant analysis (NKDA) [RT01],

\[
f_\alpha(x) = \text{sign}(w^\top \phi(x))
\]

\[
= \text{sign} \left[ \sum_{i=1}^N \alpha_i^* \phi(x_i)^\top \phi(x) \right]
\]

\[
= \text{sign} \left[ \sum_{i=1}^N \alpha_i^* k(x_i, x) \right].
\]

Since \( K + \lambda I \) is symmetric and positive definite, the solution \( \alpha^* \) can be computed with the conjugate gradient method [HS52]. Observe that the minimizer \( \alpha^* \) of the quadratic form

\[
M(\alpha) = \frac{1}{2} \alpha^\top (K + \lambda I) \alpha - y^\top \alpha
\]

fulfills

\[
\nabla_\alpha M = (K + \lambda I) \alpha^* - y = 0.
\]
Figure 4.3: In a one-versus-one approach the 3 classes can be separated with linear two-class classifiers. In contrast to this, a one-versus-rest approach would require a nonlinear classifier.

This solution is exactly in accordance with (4.37). Optimization does then start with an initial vector $\alpha^0$. This vector is successively improved by adding update vectors $u_i$ that are mutually conjugate. After $N$ iterations ($N$ is the number of samples) this yields the optimal solution $\alpha^*$ from (4.37). However, a significantly smaller number of iterations suffices to obtain a satisfying approximation in most practical problems. Therefore large scale problems can be tackled efficiently.

### 4.3.3 Solving Multi-Class Problems by Pairwise Coupling

Many classifiers, such as SVMs, are by design two-class classifiers. Therefore, the question arises how they can be generalized to multi-class problems with $m > 2$ classes. A commonly used approach [Vap98] is to train $m$ two-class classifiers that separate each class from all other $m-1$ ones. Therefore, this procedure is called one-versus-rest approach. The $m$ individual decision rules are then combined into a single one either in a probabilistic manner (e.g. for LDA) or using some heuristic (such as for SVMs). However, this coupling can lead to inconsistent results [Bis06].

A different approach is to train $m(m-1)/2$ two-class classifiers on all possible pairs of classes. This strategy is therefore called one-versus-one approach. One way of obtaining a joint classifier is then to count the votes a novel data point receives from the individual pairwise classifiers and assign it to the class with most votes. Such a voting scheme can again lead to ambiguities. Therefore, Hastie and Tibshirani [HT98] present a pairwise coupling scheme for classifiers with probabilistic outputs. These classifiers do not only return a label for a test sample but also compute a confidence in this prediction. Therefore, it is not only judged on which side of the
decision surface a sample lies, but it is also considered how far a sample is away from the class boundary.

As opposed to the one-versus-rest approach the pairwise strategy has the advantage that the individual classifiers need to solve significantly simpler problems. Consider for instance the toy problem of Figure 4.3. Here a pairwise approach would require only linear classifiers. In contrast to this, the one-versus-all subproblems are not all linearly separable. To find a model that is nevertheless capable of separating the training samples correctly, a nonlinear classifier is needed. Consequently, the one-versus-rest approach may require unnecessarily hard subproblems to be solved. For kernel classifiers the pairwise approach has an additional advantage. For these methods, both computational cost and memory consumption depend mainly on the number of training samples \( N \). For instance, the conjugate gradient optimization of NKDA scales as \( \mathcal{O}(N^2) \) in time and the size of kernel matrices is also \( N \times N \). The advantage of the pairwise approach is then that although a quadratic number (in \( m \)) of subproblems have to be solved each of these problems is significantly simpler as it involves only \( 2N/m \) training samples (assuming equal number of samples per class). Consequently, the reduction of memory consumption is inverse proportional to the square of the number of classes—memory consumption scales as \( \mathcal{O}(N^2/m^2) \). Moreover, it can be shown that the pairwise approach yields for NKDA a reduction in overall computational time that is inverse proportional to the number of classes, [RT01].

Let us now formalize pairwise coupling. There are \( m(m-1)/2 \) two-class classifiers with probabilistic outputs \( \eta_{ij}(x) \),

\[
\eta_{ij}(x) := P(Y = i | x) .
\] (4.43)

\( \eta_{ij}(x) \) represents the conditional probability for assigning a sample \( x \) to class \( i \) when there are only the classes \( i \) and \( j \) to choose from. For a discriminant function \( g_{ij}(x) \) (e.g. \( g_{ij}(x) = w^\top \phi(x) \) for NKDA) between two classes \( i \) and \( j \) the conditionals are

\[
\eta_{ij}(x) = \frac{1}{1 + \exp(g_{ij}(x))} .
\] (4.44)

The goal is then to couple all pairwise probabilities \( \eta_{ij} \) into a multi-class posterior \( \Lambda_i(x) \),

\[
\Lambda_i(x) := P(Y = i | x) .
\] (4.45)

There exists no general solution to this problem but in [HT98] an approximation is presented. Let us introduce auxiliary variables

\[
\mu_{ij} := \frac{\Lambda_i}{\Lambda_i + \Lambda_j} .
\] (4.46)
Now we seek posteriors $\Lambda_i$ so that the corresponding $\mu_{ij}$’s are close to the observed $\eta_{ij}$’s. A suitable distance measure between $\mu_{ij}$ and $\eta_{ij}$ is the Kullback Leibler divergence $D_{KL}(p\|q)$ (cf. [CT06]) which is defined for probability distributions $P(x)$ and $Q(x)$ as
\[
D_{KL}(P\|Q) := \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}. \tag{4.47}
\]
For any $\mu_{ij}$ and the corresponding $\eta_{ij}$ this distance is
\[
D_{KL}(\mu_{ij}\|\eta_{ij}) = \mu_{ij} \log \frac{\mu_{ij}}{\eta_{ij}} + (1 - \mu_{ij}) \log \frac{1 - \mu_{ij}}{1 - \eta_{ij}}. \tag{4.48}
\]
Summing these distances for all pairwise classifiers yields the objective function $L(\Lambda_1, \ldots, \Lambda_m)$ that is to be minimized,
\[
L(\Lambda) = \sum_{i<j} D_{KL}(\mu_{ij}\|\eta_{ij}) \tag{4.49}
\]
\[
= \sum_{i<j} \mu_{ij} \log \frac{\mu_{ij}}{\eta_{ij}} + (1 - \mu_{ij}) \log \frac{1 - \mu_{ij}}{1 - \eta_{ij}}. \tag{4.50}
\]
For an optimal solution $\mu^*_{ij}$ it holds that
\[
\sum_{j: j \neq i} \mu^*_{ij} = \sum_{j: j \neq i} \eta_{ij}, \quad \forall i \in \{1, \ldots, m\} \tag{4.51}
\]
subject to the constraint $\sum_{i=1}^{m} \Lambda_i^* = 1$. The optimal distribution $\Lambda^*$ that minimizes (4.49) can be computed by starting with an initial guess before iterating the update steps of Algorithm 4.1.

In the prediction phase, pairwise coupling yields estimates of the posterior $P(y|x) = \Lambda_y(x)$ and a sample $x$ is classified by maximizing the posterior,
\[
y^* = \arg\max_{y \in \{1, \ldots, m\}} \Lambda_y(x). \tag{4.52}
\]

### 4.4 Learning Relevant Compositions

Given the set $G$ of all candidate compositions $g_j \in G$ (see Section 4.2) a selection has to be performed, retaining only the discriminative ones and discarding clutter. Learning such compositions is divided up into two stages, see Figure 4.2. First those groupings have to be retrieved which are representative for a large majority of objects observed among the considered categories. Thereby, the system avoids to memorize compositions that capture details of only specific instances of a category. Moreover,
4.4. Learning Relevant Compositions

compositions should be shared among different categories. These concepts limit the description length of a compositional image representation and, thereby, reduce the risk of overfitting to specific object instances. In the learning phase the candidate compositions of all training images are therefore clustered (using \( k \)-means) into a comparably large set \( \Pi \) of 1000 prototypes \( \pi_i \in \Pi \). Thereafter, the prior assignment probabilities of candidates to clusters, \( P(\pi_i) \), are computed.

In a second stage those prototypes have to be selected that help in distinguishing sets of categories from another. As later stages of the system combine multiple compositions found in one image, we do not have to solve the harder problem of finding individual compositions that are characteristic for a single category. In contrast to such an approach we pursue the simpler setting of sharing compositions for multiple categories (cf. [TMF07]). Another advantage of this approach is that it supports sharing of training samples among object categories.

To begin with, the category posterior of compositions has to be estimated, i.e. the posterior of a categorization with label \( c \in \mathcal{L} \) (\( \mathcal{L} \) denotes the set of all category labels) given a composition \( \Gamma_j \),

\[
P_{\Gamma_j}(c) := P(C = c | \Gamma_j) .
\]  

(4.53)

This distribution is learned by training probabilistic two-class kernel classifiers on all composition candidates found in the labeled training images. For the two-class classification we choose \textit{nonlinear kernel discriminant analysis} (NKDA) (see Section 4.3.2) and perform a pairwise coupling, Section 4.3.3, to solve the multi-class problem. The combined probabilistic classifier yields an estimate of the posterior (4.53) for the respective image category.
Subsequently the category posterior is used to calculate the relevance of a composition for discriminating sets of categories from another. Groupings that are present in all categories are penalized by this idea, whereas combinations which are typical for only a few classes are fostered. The discriminative relevance measure is then modeled as the entropy of (4.53),

\[ H(P_{\Gamma_j}) = - \sum_{c \in \mathcal{C}} P(C = c|\Gamma_j) \log P(C = c|\Gamma_j). \] (4.54)

Since entropy measures how uniform a random variable is distributed (see Chapter 1) the entropy (4.54) should be minimized.

Finally a cost function can be formulated that measures the total relevance of a compositional prototype \(\pi_i\). It combines the prior assignment probabilities of clusters, \(P(\pi_i)\), with the entropy (4.54),

\[ S(\pi_i) \propto - \log P(\pi_i) + \lambda H(P_{\pi_i}). \] (4.55)

Both constituents of the cost function should be normalized to the same dynamic range, giving rise to an additional additive constant that can be discarded and to the parameter

\[ \lambda = \frac{\max_i \log P(\pi_i) - \min_i \log P(\pi_i)}{\max_i H(P_{\pi_i}) - \min_i H(P_{\pi_i})}. \] (4.56)

This parameter trades the occurrence frequencies of compositions against their discriminative usefulness.

For the 102 object classes in the Caltech-101 image database [Cal] a set of 250 relevant composition prototypes is established. This set is obtained by selecting the prototypes \(\pi_i\) with minimal cost \(S(\pi_i)\). The comparably small size of this set of relevant compositions indicates that compositions are shared among object classes since there are too few compositions to be highly characteristic for individual classes. An image is then represented by retaining only those composition candidates \(g_j \in \mathcal{G}\) formed in Section 4.2 which are closer to one of the relevant prototypes than to any irrelevant one. However, at least the best 5 candidates are retained, thereby ensuring that images from the background category always yield a non-empty representation. The set of relevant compositions is then denoted \(\mathcal{R}\), with \(\mathcal{R} \subseteq \mathcal{G}\).
4.5 A Compositional Shape Model for Binding Compositions

4.5.1 Modeling Objects and Scene Context

Classification of an object requires that all compositions are combined in a single object model so that a concerted categorization hypothesis can be derived. This structured object model should couple (i) the appearance of salient object regions, (ii) object shape (that is the geometry of the local features), and (iii) the scene context in which the object appears. All of this local and global information is then coupled in the compositional shape model [OB06] so that a concerted object categorization hypothesis can be derived during recognition. This Bayesian network is illustrated in Figure 4.4. Later chapters will extend this model to incorporate additional evidence.

The appearance of object regions is covered by compositions, whereas the shape is represented through spatial relations between compositions and the object center. Lastly, scene context is captured by the co-occurrence of all compositions \( g_j \in \mathcal{G} \) that have been established in the image. Therefore, a mixture \( \mathbf{g}^f \) of the distributions of all compositions is computed,

\[
\mathbf{g}^f := \sum_{g_j \in \mathcal{G}} \frac{1}{|\mathcal{G}|} g_j. \tag{4.57}
\]

This variable coarsely captures the gist [Fri79] of the scene—it roughly models the compositions that are present without needing a complex spatial model for them. This descriptor is therefore invariant with respect to individual missing compositions or changes in their spatial arrangement. In essence this representation defines a bag-of-compositions.
4.5.2 Object Localization

Subsequently, all established compositions $g_j \in G$ are used to obtain a first estimate of the object center. Therefore, the descriptor $g^I$ from (4.57) is employed to bind compositions by means of their co-occurrence. To determine the object location $x$, the positions $x_j$ of all compositions $g_j$ are considered (see Figure 4.5). Compositions scatter around the object center and, since no additional information on the object is available, it is assumed that the $x_j$ are distributed according to a normal distribution, $x_j \sim \mathcal{N}(x, \Sigma)$. The first moment of this distribution is the center of mass of the object (we will deal with the second moment in a later part of this thesis). The maximum-likelihood estimate $x^*$ for this unknown variable is the mean of all the $x_j$,

$$x^* = \sum_j x_j \cdot \text{Prob}\{j\text{-th composition observed in the scene}\}$$

$$= \frac{\sum_j x_j \sum_{c \in C} p(g_j|c, g^I) P(c|g^I)}{\sum_{j,c} p(g_j|c, g^I) P(c|g^I)}.$$  \hspace{1cm} (4.59)

The distribution $p(g_j|c, g^I)$ is estimated using Parzen windows [DHS01] and $P(c|g^I)$ is computed using NKDA. In the training phase, when the true category label is available for images, the second sum (over categories) reduces to the true category $c_{true}$ and the distribution over categories degenerates to a discrete Dirac distribution. In this case (4.59) simplifies to

$$x^* = \frac{\sum_j x_j \cdot p(g_j|c_{true}, g^I)}{\sum_j p(g_j|c_{true}, g^I)}.$$ \hspace{1cm} (4.60)

To simplify notation, we will drop the star so that $x$ denotes the maximum-likelihood estimate for the object center in the following.

4.5.3 Evaluation of the Localization Performance

An evaluation on the Caltech-101 database shows that the estimate for the object center in (4.59) deviates from the true center (taking the center of the object bounding box from the hand annotations) by $8.8 \pm 3.8\%$ of the bounding box diagonal (averaged over all categories). This is roughly the size of the atomic parts and, therefore, exact enough to couple compositions in the compositional shape model.

4.5.4 Object Classification with a Compositional Shape Model

Subsequently, all relevant compositions $g_j \in \mathcal{R}$, which have been obtained as described in Section 4.4, are to be coupled to obtain a consolidated object hypothesis.
4.5. A Compositional Shape Model for Binding Compositions

Figure 4.5: Illustration of the different constituents of the compositional shape model.

Dependencies are established between compositions by means of the overall object shape, through scene context, and by a hypothesis on the object category, $c$. These dependencies are represented in the Bayesian network depicted in Figure 4.4. This compositional shape model encodes object appearance by representing object regions with compositions. Object shape is then captured by the spatial structure of the compositions, i.e. by the shifts $s_j$ of compositions from the object center (see Figure 4.5 for an illustration). A composition $g_j$ established at location $x_j$ in the image gives rise to an offset

$$s_j := x - x_j$$

from the object center $x$. Objects can then be classified on the basis of the category posterior

$$P(c|E) := P(c|g^I, x, \{g_j, x_j\}_{j: g_j \in R}).$$

In the abbreviated form $E$ denotes the collected evidence. Let us now derive the category posterior. First Bayes' formula is applied to obtain

$$P(c|g^I, x, \{g_j, x_j\}_{j: g_j \in R}) = \frac{p(\{g_j, x_j\}_{j: g_j \in R}|g^I, x, c) P(c|g^I, x)}{p(\{g_j, x_j\}_{j: g_j \in R}|g^I, x)}. \quad (4.63)$$

Next the independence properties represented in the Bayesian network from Figure 4.4 are to be exploited to factorize the likelihood $p(\{g_j, x_j\}_{j: g_j \in R}|g^I, x, c)$. Individual compositions are conditionally independent, conditioned on the object model parameters $c$ and $x$ and on scene context $g^I$. However, it should be emphasized that this is a conditional independence—the $g_j$ alone are not independent. Moreover, the denominator can be skipped since it is only a normalization constant that has no
influence on the object category \( c \)

\[
P(c|E) \propto P(c|g^I, x) \times p (\{g_j, x_j\}_j|g^I, x, c) \tag{4.64}
\]

\[
= P(c|g^I, x) \times \prod_{j: g_j \in R} p(g_j, x_j|g^I, x, c) \tag{4.65}
\]

Now Bayes’ formula is applied to the individual factors.

\[
P(c|E) = P(c|g^I, x) \times \prod_{j: g_j \in R} \frac{P(c|g_j, x_j, g^I, x) \ p(g_j, x_j|g^I, x)}{P(c|g^I, x)} \tag{4.66}
\]

\[
= \left[ P(c|g^I, x) \right]^{1 - |R|} \times \prod_{j: g_j \in R} P(c|g_j, x_j, g^I, x) \ p(g_j, x_j|g^I, x) \tag{4.67}
\]

Here \(|R|\) denote the cardinality of the set \( R \). The last factor under the product does not depend on \( c \). It is, therefore, a normalization factor that can be omitted. Moreover, object classification should be independent of the absolute position of the object in the image, that is \( P(c|g^I, x) = P(c|g^I) \). Similarly, only the relative shifts \( s_j := x - x_j \) of compositions from the object center are of interest, not their absolute locations. Thus above equation is further simplified

\[
P(c|E) \propto \left[ P(c|g^I) \right]^{1 - |R|} \times \prod_{j: g_j \in R} P(c|S_j = x - x_j, g_j, g^I) \tag{4.68}
\]

\[
= \exp \left( (1 - |R|) \ln \left[ P(c|g^I) \right] \right) + \sum_{j: g_j \in R} \ln P(c|S_j = x - x_j, g_j, g^I) \tag{4.69}
\]

The logarithm is introduced to enhance numerical stability. The first distribution in (4.69) has already been estimated for (4.59). The latter distribution is again estimated using NKDA from the training data.

In conclusion, the category posterior from (4.62) is split up into two terms. The first represents scene context and the second object shape and appearance using compositions. Both terms can be estimated using classifiers with probabilistic outputs. As a result, object classification has been formulated as a statistical inference problem that can be solved efficiently by computing the individual distributions and combining them in a Bayesian network. This model does not only classify an object. It also returns a confidence in this prediction.
4.6 Experiments

4.6.1 The Caltech-101 Image Database

In the following, the compositional approach is evaluated on the challenging Caltech-101 database [Cal] which has been collected by Fei-Fei et al. [FFFP04]. The dataset consists of 101 object categories and a background category and there are 9145 images in total. Classes have varying numbers of samples (between about 30 and 800) and range from photos with clutter to line drawings. The large intra-category variations in this database render object recognition a challenging task. However, there are only limited variations in pose. Figure 4.6 shows example images from all categories of this database.
To evaluate a recognition algorithm, the database is randomly split into two disjoint sets, one for training and the other for testing. Thereafter the model is learned on the training samples before it is finally used to recognize objects in the test images. The overall performance on the test set is summarized in the retrieval rate $\Psi$ which measures the fraction of correctly predicted test images (the true positive rate). It is common practice to compute retrieval rates $\Psi_c$ separately for each class $c$ before averaging these scores in the overall retrieval rate $\Psi$,

$$\Psi := \frac{1}{|\mathcal{L}|} \sum_{c \in \mathcal{L}} \Psi_c, \quad \text{where } \Psi_c \equiv \{\text{true positive rate for category } c\}. \quad (4.70)$$

This training and testing procedure is repeated several times for different random splits of the dataset to obtain error bars by means of cross-validation.

**State of the Art**

Due to the large number of categories, random categorization (labeling an image randomly as containing an object of some class $c \in \mathcal{L}$) would yield a retrieval rate of less than 1%. Berg et al. [BBM05] have calculated a reasonable baseline performance of 16% using a 1-nearest neighbor classifier on texton histograms and 15 training images per category. Moreover, their approach which is based on shape correspondences yields a retrieval rate of 48%. Using a constellation model, Fei-Fei et al. [FFFP04] have reported a performance of 16%. This generative model has been enhanced by means of a discriminative classifier [HWP05] and a fusion of multiple interest point detectors to yield a retrieval rate of 40.1%. Serre et al. [SWP05] have followed an approach based on a neurophysiologically motivated convolutional network (the HMAX model, [RP99]) and achieved a performance of 42%. In [SWB+07] they report an improved retrieval rate of $44 \pm 1.1\%$ on 15 training images per class. Wolf et al. [WBM06] have improved the system by considering Gestalt-like feature detectors and obtained a performance of $51.2 \pm 1.2\%$.

Table 4.1 summarizes the retrieval rates that are achieved by state of the art approaches on this database using 30 training images per category. Note that the top ranked methods exploit the peculiarity of this specific database that the spatial structure of objects is limited in its variation with respect to the image, e.g. [LSP06] split the image into a regular grid and concatenate the individual descriptors to a joint one. In contrast to this, the approach of this thesis aims at *learning* the compositional structure of objects.
Table 4.1: Retrieval rates (in percentage) of current approaches on the Caltech-101 database using 30 training images per category.

- **Approach**
  - [ZBMM06]
  - [LSP06]
  - [GD06]
  - [ML06]
  - [ZMLS06]

- **Retrieval rate**
  - 66.23 ± 0.48
  - 64.6 ± 0.8
  - 58.23
  - 56
  - 53.9

### 4.6.2 Baseline Performance without Compositions

The approach that has been presented in this chapter bases object categorization on an intermediate compositional image representation. The following experiments estimate a baseline performance of the system without this hidden representational layer. Therefore we neglect the compositional representation and consider only the bag-of-features descriptor $g'$ of the whole image, introduced in (4.57). The simple graphical model that represents this approach is depicted in Figure 4.7. It is essentially the non-compositional part of the model from Figure 4.4.

The basic evaluation scenario is as follows: For each class up to 50 training images are randomly selected (the coupled classifiers are weighted to compensate for the unequal priors) and the remainder is taken as test set (minimally 10 images in a class and over 4000 in total). To estimate the retrieval rate and its error 5-fold cross-validation is performed, i.e. the same algorithm is applied to 5 different training and test set compositions. Figure 4.8(b) shows the resulting category confusion table for the case of a feature bag which consists of 100 prototypes. This simple model achieves a retrieval rate of 33.3 ± 0.9%.

To evaluate its dependence on the size of the codebook from Section 3.3 the simple bag-of-features approach is now evaluated with different numbers of clusters. Figure 4.8(a) shows the retrieval rates under varying model complexity. In the case of 1000 prototypes this model yields a retrieval rate of $38.4 \pm 1.3\%$. As the localized feature histograms are fairly low-dimensional descriptors, comparably small codebooks do already yield considerable retrieval rates. This is advantageous for modeling compositions robustly which are obviously consisting of fewer parts than a complete image and justifies the choice of a 100 prototype representation in the full compositional architecture.
4.6.3 Categorization Performance of the Compositional Model

Subsequently, the full compositional model is learned to categorize images. Evaluation under 5-fold cross-validation yields a retrieval rate of \( 53.6 \pm 0.88\% \). Additionally it can be noted that the overall retrieval rate per image without averaging over categories is \( 67.3 \pm 2.1\% \). Figure 4.10(a) depicts the respective category confusion table. When comparing this plot with the one for the simple bag-of-features approach from above it is evident that the number of incorrectly classified images has significantly decreased. The categories with lowest performance are “octopus”, “wildcat”, and “ant”, the best ones are “car”, “dollar bill”, and “accordion”. Amongst the off-diagonal elements the confusions “water-lilly” vs. “lotus”, “ketch” vs. “schooner”, and “lobster” vs. “crayfish” are the most prominent ones (cf. Figure 4.9 for an illustration). All of these confusions are between pairs that are either synonymous or at least semantically very close. To conclude, the observable gain in resolving ambiguities between classes emphasizes the advantage of an intermediate compositional image representation in contrast to a direct categorization.

Multiple Scales

In another experiment images are rescaled to half their original size so that local descriptors on half the original scale are extracted in addition to the standard features.
4.6. Experiments

Figure 4.9: Most confused categories.

Figure 4.10: (a) Category confusion table of the compositional model. The retrieval rate is $53.6 \pm 0.88\%$. (b) Distribution of the number of parts assigned to each composition.

In the training phase the system is therefore trained on two scales and, in effect, becomes less scale dependent—each object is viewed on two scales. This second scale boosts performance to $57.8 \pm 0.79\%$.

4.6.4 Evaluating Compositions

The following evaluates the relevant compositions that have been learned. First, Figure 4.10(b) plots the number of parts that are typically grouped to form a composition. On average there are approximately 57 parts coupled together. This is a significant increase compared to the tuple groupings formed in [OB05].

The next experiment intends to visualize the learned compositions. Since these are agglomerations of localized feature histograms that cannot be displayed directly an indirect method has to be pursued. We therefore plot image regions from the
Figure 4.11: Visualization of compositions: The pictures show the rectangular hulls of test image regions associated with different compositions. Different, abstract concepts captured by compositions: (a) Parts of faces, (b) accordions, and (c) cars, motorbikes. Feature sharing for complex structures of airplanes and schooners in (d), and of boat sails and butterfly wings in (e). (f) Roundish structures. (g) Elongated patterns of chairs and menorah. (h), (i) Texture with and without a sharp edge, respectively. (j) Contours. (k) Drawings. (l) Feet of chairs, pianos, and insects.
4.6. Experiments

Figure 4.12: Relevance of detected compositions (black boxes). Brighter patches than background indicate high relevance, darker ones indicate compositions are not useful.

test images that have been detected to contain compositions that are similar to the relevant compositions learned on the training data. A displayed region is then simply the rectangular hull of all parts that have been agglomerated to a composition. Figure 4.11 visualizes a subset of all learned compositions by showing 3 candidate regions for each. The zones are therefore scaled to equal sizes. Observe that compositions are reflecting quite different, abstract concepts: There are those that nicely correspond to salient structures in a single category Figure 4.11(a)-(c). In the latter case there are however also representatives from another category (motorbike) that show a visually similar pattern. Figure 4.11(d) and (e) exhibit more extended feature sharing. In (d) the triangular structures of airplane rudders and schooners are captured, while (e) combines sails of different boat categories and butterfly wings. The composition in (f) grasps roundish, metallic structures and (g) elongated, repetitive patterns of windsor chairs and menorahs. The next two compositions are an example of textures. The latter however also seems to model the presence of sharp edges, while (j) captures characteristic contours of pianos and staplers. An example for drawings is given in (k), while (l) seems to model the abstract concept of feet of chairs, pianos, and insects. In conclusion various kinds of low level properties are combined to represent fairly abstract concepts that help to discriminate between categories.

Localizing Object Constituents

Subsequently, the relevance of individual compositions for the task of categorizing an image is to be evaluated. Therefore, the relevant object constituents are to be identified and localized. We measure how the categorization performance varies when a single composition is removed. Relevance is then proportional to the decrease in categorization probability of the true category. Figure 4.12 shows examples for the airplane category. This visualization indicates that especially the noses and rudders are particularly relevant.
Establishing Deep Compositional Hierarchies

At its core, compositionality is about finding appropriate compositions of parts that lead to intermediate object representations capturing characteristic substructures of objects and their mutual (spatial) relationships. In Chapter 4 an approach has been presented that uses a proximity based grouping strategy for proposing candidate compositions. The most appropriate candidates are then determined by means of a criterion function that measures the value of compositions for the task of object recognition. As a result a single layer of compositions is obtained. As opposed to this approach the following presents a system that learns to build deep compositional hierarchies.

Learning compositional hierarchies is divided into two subproblems [OSB06]. First, image parts are grouped in a data-driven, bottom-up manner. To this end the Gestalt laws of good continuation, proximity, and convexity are applied. Secondly, these intermediate compositions are grouped in a top-down fashion guided by a grouping procedure that is driven by category models. Whereas the first stage depends on a set of predefined laws of perceptual organization, the grouping in the second stage is controlled by category models that are automatically learned in the training phase. The idea is that bottom-up grouping limits the representation complexity by agglomerating those parts into compositions that are definitely related according to perceptual organization. This grouping stage reduces the great number of initial parts to an intermediate representation layer consisting of few salient compositions. The second stage recursively forms agglomerations of compositions based on their distinctiveness for a category. Such groupings constitute characteris-
tic combinations of dissimilar object parts to cover the heterogeneity of real world objects.

Now it becomes evident how these two processes are complementing each other. The perceptual grouping in the first stage is driven by the perceptual relatedness or similarity of parts (consider for instance a grouping based on good continuation that favors smooth contour curves, i.e. neighboring curvelets have similar orientations). This procedure groups parts to compositions that can be represented with less complexity than an encoding that represents all parts separately. Consequently, perceptual grouping processes can also be understood as following a minimum description length principle. Perceptual grouping tends to agglomerate constituents with similar local statistics into more complex compositions and reduces redundancies in the intermediate compositional image representation. This grouping is in direct contrast to the second stage where compositions of heterogeneous constituents are formed so that they are most characteristic for object categories.

In accordance with the models from the previous chapters this system is again learned using minimal supervision in the training phase. Only category labels for the most prominent object in an image are required, thereby eliminating the need for elaborate hand-segmentations or object localizations. The challenge is then to automatically learn top-down grouping models for large numbers of categories without explicit information about the compositional structure of objects in the training data. This problem is tackled by first approximating category dependent co-occurrence statistics on the training data and using them to form a hierarchy of potential grouping candidates. Using this compositional hierarchy the grouping model is then being refined. In other words, we start with simple and robust category statistics which are then used to guide the system during its investigation of increasingly complex compositions that are in turn utilized to refine the statistics.

5.1 Outline of the Approach

Let us now realize above ideas in a concrete vision system. The approach can be understood best when recognition is considered first. Figure 5.1 shows a sketch of the corresponding processing pipeline. The involved processing steps are then covered in detail by later sections. Given a novel image, Canny edge detection is performed. The resulting edge pixels are grouped using a purely bottom-up, perceptual grouping strategy that yields nearly closed arcs or fairly straight curves. To find a representation for the curves localized feature histograms from Section 3.2.2
5.1. Outline of the Approach

Canny edge detection
\[ \rightarrow \]
decompose curves \[ \rightarrow \] edgels
perceptual grouping
select salient compositions
compositions \[ \rightarrow \] shape model
image categorization
image category
composition selection
bottom-up top-down
feature extraction
building compositional hierarchy

Figure 5.1: Processing pipeline for building deep compositional hierarchies during recognition.

are computed for image patches on a regular grid. A curve is then represented by forming a composition of all the patches that lie on the curve. In a second stage top-down grouping is performed recursively. This step yields agglomerations of curves with increased discriminative power compared to their original constituents. Assume for the moment that groupings which are distinctive for categories have already been learned automatically from the training data. The objective of top-down grouping is then to form a hierarchy of compositions by recursively combining those pairs of constituents whose composition has highest category posterior. Finally, all the groupings are coupled together by means of a shape model.

The final challenge is then to automatically learn and represent models for top-down grouping in the case of large numbers of object classes without extensive user
supervision. In other words, how can the system learn which compositions are relevant for a category without being told about the compositional structure of objects? This problem is tackled subsequently by first estimating category dependent co-occurrence statistics of bottom-up grouped curves in training images of given categories. Using this distribution the curves are then grouped in a recursive manner, thereby giving rise to a hierarchy of compositions. This hierarchy is finally used to update the previously estimated category dependent grouping statistics with probabilities of higher level groupings. Moreover, the global shape model is learned.

5.2 Perceptual Bottom-Up Grouping

The primary objective of bottom-up grouping is to find a comprehensive image representation based on salient edge curves that is yet compact. Processing of an image starts by performing Canny edge detection and finding connected edge curves as illustrated in Figure 5.2 b). This step, however, yields curves of any degree of complexity. To find salient edge curves the contours are first split into fairly simple parts before grouping them again in a perceptually controlled manner.

The complexity of a curve grouping is examined with respect to the following Gestalt laws of perceptual organization (see Section 2.3.3): good continuation (preferring curves with smooth continuity), proximity (avoiding large gaps), and convexity (short curves circumscribing large areas). The underlying idea is to look for curves that remain stable and prominent over different realizations of an object category despite the large intra-category variations. Roughly speaking we are interested in smooth elongated curves (no convexity, but maximal smoothness) or nearly circular arcs (maximal convexity). These two cases constitute the extrema of a criterion function $\zeta(\gamma)$ for curves $\gamma$. Let $A(\gamma)$ denote the area circumscribed by the curve and $l(\gamma)$ be its length then

$$\zeta(\gamma) := \frac{A(\gamma)}{A(\text{circle with perimeter } l(\gamma))} = \frac{4\pi A(\gamma)}{l^2(\gamma)}.$$  

(5.1)

For straight lines, $\zeta$ is zero and for circles it is one. Therefore it is suitable to maximize the criterion function

$$\tilde{\zeta}(\gamma) := \left| \zeta(\gamma) - \frac{1}{2} \right| .$$  

(5.2)

Breaking contours into simple parts is then carried out as follows: Curves with almost maximal criterion function $\tilde{\zeta}$ are kept unaltered as they are already nearly
5.2. Perceptual Bottom-Up Grouping

Figure 5.2: Perceptual bottom-up grouping. a) Original image. b) Connected edge curves from Canny edge detection depicted in the same color. c) Potential high curvature break points. d) Splines fitted to salient curves and illustration of the corresponding image regions.

straight or circular. Otherwise they are split at the point of highest curvature as shown in Figure 5.2 c). The resulting two segments are then in turn processed recursively. As the resulting curvelets will be merged subsequently, a splitting into too short segments is not critical.

A cubic B-spline is fitted to each of the curvelets to remove small wiggles. The spline curves are then ranked in a queue according to $\bar{\zeta}$. For the curve $\gamma_m$ with maximal $\bar{\zeta}$ its pairwise groupings with all other curves $\gamma_n$ are evaluated. Therefore, a spline is fitted to each resulting composition $\gamma_g$ consisting of $\gamma_m$ and $\gamma_n$. Thereafter the score $\bar{\zeta}(\gamma_g)$ is computed. To enforce smoothness of the grouping those compositions will be removed for which at least one of the angles $\alpha_m$ and $\alpha_n$ between the
Figure 5.3: Perceptual bottom-up grouping. a) Original image. b) Connected Canny edge curves. c) Potential high curvature break points. d) Splines fitted to salient curves and visualization of the underlying image regions.

tangent of a curve and the connection between the two curves is greater than 90° as illustrated in Figure 5.4. To follow the principle of proximity, a grouping is also discarded if the gap between the two curves is longer than \( \min\{l(\gamma_m), l(\gamma_n)\} \), the length of the shortest of the two constituent curves. Finally, compositions \( \gamma_g \) are removed if they do not improve the criterion function \( \bar{\zeta} \) in comparison to their constituents, that is if

\[
\bar{\zeta}(\gamma_g) < \min\{\bar{\zeta}(\gamma_m), \bar{\zeta}(\gamma_n)\}.
\]

To summarize, composition candidates \( \gamma_g \) formed from \( \gamma_m \) and \( \gamma_n \) will be discarded in the following cases

\[
\text{discard } \gamma_g \Leftrightarrow \begin{cases} 
\min\{\alpha_m, \alpha_n\} > 90^\circ \vee \\
gap(\gamma_m, \gamma_n) > \min\{l(\gamma_m), l(\gamma_n)\} \vee \\
\bar{\zeta}(\gamma_g) < \min\{\bar{\zeta}(\gamma_m), \bar{\zeta}(\gamma_n)\}.
\end{cases}
\]

The grouping \( \gamma_g \) with maximal \( \bar{\zeta} \) is chosen among the remaining candidates and it is added to the queue while both of its constituents are removed. If the set of
candidates is empty, only $\gamma_m$ will be removed. This curve merging continues with the currently best curve in the queue until there is only one left. In the subsequent stages of the architecture, all created groupings and those curves that could not be merged with another curve are processed further.

5.3 Forming Robust Descriptors for Salient Curves

Each contour that is generated by above bottom-up grouping has to be represented in such a way that curves of varying length and number of constituent curves are possible. To this end localized feature histograms are suitable. The underlying $20 \times 20$ pixel patches are extracted on a regular grid (spacing of 5 pixels). The features $e_i$ describing these patches are then computed as described in Section 3.2.2.

For each pixel on a curve the closest feature patch $e_i$ is selected. All these patches are collected and duplicates are removed (see Figure 5.2 d)). The curve is then represented by combining all these patches in a composition $g_j$. As proposed in Section 4.2 a composition $g_j$ is described by the mixture distribution (4.2) over its parts. Therefore, on Caltech-101 a $k = 200$ dimensional codebook of atomic parts is shared by all the 102 categories of that database.

5.4 Learning of Top-Down Grouping

In order to be able to group parts of a novel image in a top-down manner, the system first has to learn to do so from the training data. Top-down grouping is learned from scratch without having to manually specify any grouping laws. This step is carried out automatically with just the training images and the category label of the most prominent object, but without any further supervision. How can the system then learn what to group without being instructed about the compositional nature of objects? The key idea is to start with an estimate of the co-occurrence statistics of constituents. Based on this information grouping hierarchies are produced in a next step. Thereafter, these groupings are used to update the initial compositional grouping statistics. Figure 5.5 sketches this bootstrapping procedure.
Figure 5.5: Learning to group compositions based on object category models without supervision regarding the compositional structure of objects. Training starts by estimating co-occurrences of base compositions from the training images. Based on these initial estimates, grouping hierarchies are formed. Out of these hierarchies updated grouping probabilities are extracted.

5.4.1 Salient Base Compositions

Processing of a training image starts by selecting a subset of the curves generated in Section 5.2. Therefore, interest point detection is performed (using the scale invariant Harris interest point detector from [MS04]) and all those local patches are marked which cover at least a single interest point (IP). The idea is then to find the most salient curves $\gamma$ using the score function

$$\xi(\gamma) := l(\gamma) \cdot \frac{\text{# patches with IP on } \gamma}{\text{# patches on } \gamma}. \quad (5.5)$$

From the set of all grouped curves we choose the 7 with maximal score $\xi(\gamma)$. From the remainder, at most 4 curves with minimal $\zeta(\gamma)$ (curves that are most circular) are selected; all other curves are discarded. To cover regions not represented by curves, 3 seed points are chosen from the set of all interest points. All patches with interest points that are not farther than 50 pixels from such a seed point are combined to yield 3 additional groupings. The selected curves and additional groupings are collected in the set $\Gamma_{C_0}$ and form base compositions for the subsequent top-down grouping. They are, however, groupings themselves and each is represented by a mixture distribution $g_j$ from (4.2).
5.4. Learning of Top-Down Grouping

5.4.2 Approximating Grouping Probabilities Using Initial Groupings of Base Compositions

For each pair of base compositions $g_i, g_j$ a grouping $g_{ij}$ is established. It is represented by combining the feature histograms (4.2) of its constituents, i.e.

$$g_{ij} = \frac{1}{2}(g_i + g_j). \quad (5.6)$$

The advantage of such a representation is that all compositions are encoded in the same feature space, independently of their level in the compositional hierarchy and the number of atomic patches they cover. Let $\mathcal{L}$ denote the set of all category labels and $c \in \mathcal{L}$ be the category label of the image under consideration. For the initial training step all the groupings $g_{ij}$ which have been formed in all the training images are combined. These samples are then used to learn a first approximation of the category posterior of groupings $g_{ij}$.

$$P(C = c | g_{ij}). \quad (5.7)$$

This distribution is learned by training probabilistic two-class classifiers on all the training samples. For the two-class classification NKDA is chosen again and a pairwise coupling is performed to solve the multi-class problem (see Section 4.3).

5.4.3 Forming a Compositional Hierarchy

The goal is now to recursively form top-down groupings that maximize the posterior (5.7) to obtain a hierarchy as illustrated in Figure 5.6. Firstly a list $\Gamma_C$ of all grouping
candidates is established by inserting all base compositions from $\Gamma_{C_0}$. Moreover, all base compositions $g_i, g_j$ are grouped in a pairwise manner, yielding compositions $g_{ij}$. Among all $g_{ij}$ the grouping with maximal posterior

$$g_{ij}^* = \arg\max_{g_{ij} \in \Gamma_{C}} P(c|g_{ij})$$

is selected and added to the list of candidates $\Gamma_C$ and the constituents are removed

$$\Gamma_C \leftarrow \Gamma_C \cup \{g_{ij}^*\} - \{g_i, g_j\}.$$ (5.9)

Now $g_{ij}^*$ is grouped in a pairwise manner with all remaining elements of $\Gamma_C$. Then recursive grouping continues again with (5.8) to find the next best composition until there is only one element in $\Gamma_C$. In conclusion a hierarchy of groupings in the form of a binary tree is established as illustrated in Figure 5.6. Base compositions constitute the leaves whereas the last remaining element of $\Gamma_C$ forms the root.

### 5.4.4 Local Maxima of the Compositional Hierarchy

Subsequently, a subset of all groupings in the hierarchy is to be selected. This step is necessary since consecutive groupings in the tree are similar. The following procedure filters out such almost duplicates. For each leaf of the hierarchy (illustrated in Figure 5.6) the path to the root is followed and all locally optimal groupings on this path are collected. A grouping is locally optimal if its category posterior (5.7) is greater than that of its predecessor and successor node which lie on the path to the root. Optimality is determined by the category posterior (5.7) of the true category, $P(C = c_{\text{true}} | g_{ij})$. After removing duplicates this processing yields the set $\Gamma_L$ of all compositions with locally maximal posterior.

Finally, the category posterior in (5.7) is updated by training the classifiers with all locally optimal compositions $g \in \Gamma_L$ established for all the training images. This updated posterior guides top-down grouping in the recognition phase.

### 5.5 Recognition based on the Learned Grouping Model

#### 5.5.1 Applying Top-Down Grouping in the Recognition Phase

In the recognition phase a novel probe image is processed by bottom-up grouping and feature extraction as described in Section 5.2 and Section 5.3, respectively. Moreover,
5.6. Evaluation

Salient base compositions are selected as in Section 5.4. Thereafter, the final estimate of the category posterior (5.7) is used to form a hierarchy of compositions as in the previous section. In contrast to training, the correct category label of the image needed for Eq. (5.8) is not given, during recognition. Therefore, (5.8) is replaced by

$$g^*_ij = \arg \max_{g_i, g_j \in \Gamma_c} \max_{c \in \mathcal{C}} P(c \mid g_{ij})$$

(5.10)

in the recognition phase. Similarly, the set of local maxima $\Gamma_L$ can only be found when using $\max_{c \in \mathcal{L}} P(c \mid g)$ as the criterion to maximize.

5.5.2 Shape Model for Binding Compositions

Subsequently, all compositions $g_j \in \Gamma_L$ of an image that have been selected as local maxima in the compositional hierarchy have to be combined to obtain a single, concerted hypothesis for the object category. To this end the compositional shape model from Section 4.5.4 is employed to estimate the category posterior $P(c \mid g^I, x, \{g_i, x_j\}_{j: g_j \in \Gamma_L})$ according to (4.63) and the object is localized as described in Section 4.5.2.

5.6 Evaluation

Subsequently, the compositional approach is evaluated on the Caltech-101 image database [Cal].

5.6.1 A Baseline Model without Compositionality

The presented approach establishes a hierarchy of compositions that is situated between the initial feature representation of a scene and its final categorization. To estimate the gain of compositionality, these hidden representation layers are neglected in the following experiment. Therefore, images are categorized by combining all the descriptors in the single bag descriptor $g^I$ from (4.57). Evaluation is then conducted by randomly collecting 30 training images per category and taking the remainder as test set. The retrieval rate and its error is estimated by performing 5-fold cross-validation, i.e. the same algorithm is run on splits of the data into five different training and test sets. For the $k = 200$ dimensional codebook that is generated as described in Section 3.3 this base model achieves a retrieval rate of $41.3 \pm 0.38\%$. 

Chapter 5. Establishing Deep Compositional Hierarchies

5.6.2 Evaluation of the Model based on Deep Compositional Hierarchies

In the following, the entire compositional approach is evaluated under 2-fold cross-validation. It yields a retrieval rate of $53.0 \pm 0.49\%$ (note that the present approach does not use the multi-scale features of Section 4.6.3). Figure 5.7 shows the corresponding category confusion table after a permutation of the category labels which is described in Section 5.6.3. The observable gain in performance over the baseline model from above emphasizes the advantage of an intermediate compositional representation layer in contrast to a direct categorization. A further investigation of the full model shows that the best performing categories are “car”, “motorbike”, and “pagoda”, the worst ones are “panda”, “strawberry”, and “ant”. The most prominent pairwise confusions are “water-lilly” vs. “lotus”, “crocodile” vs. “crocodile head”, and “panda” vs. “soccer ball”. These confusions are between pairs that are either semantically very close or visually similar.
5.6. Evaluation

Finally, Figure 5.8 shows an evaluation of the sparseness of the image representation induced by the grouped curves. Therefore, the fraction of all local image features that are used to describe the grouped curves in an image is measured over all test images. Averaged over all test images, 7.7% of all local features are used, yielding a fairly sparse representation.

5.6.3 Class Hierarchies for Analyzing Categorization

The categorization which has been established in Section 5.6.2 induces a hierarchical structure among the categories which reveals the degree of relatedness of categories. Therefore, the category confusion probabilities are used to measure the mutual similarities between categories in the database. The final goal is then to establish a hierarchy of categories.

The probability that a test image of category $c_{\text{true}} \in \mathcal{L}$ is classified by our architecture as belonging to class $c_{\text{pred}} \in \mathcal{L}$ is given by $P(c_{\text{pred}}|c_{\text{true}})$. The complete category confusion table is then represented by the matrix

$$M_{c_{\text{true}},c_{\text{pred}}} := P(c_{\text{pred}}|c_{\text{true}}).$$

(5.11)

The matrix is symmetrized by adding its transpose

$$\tilde{M} := \eta E - (M + M^T - 2 \text{diag}[M]).$$

(5.12)

Here $E$ denotes the matrix of only ones, $\eta$ is a constant and $\text{diag}[M]$ is $M$ with its off-diagonal entries set to zero. The resulting matrix $\tilde{M}$ is used as a distance matrix between categories for a subsequent hierarchical clustering of categories. For this step, Ward’s Method with its minimum variance concept is applied. As a result a
hierarchical cluster tree is obtained with categories at the leafs and sets of similar categories at inner nodes. Dissimilar categories are connected by long paths over inner nodes near the root, whereas similar ones are connected over short paths close to the leafs. Figure 5.7 shows the hierarchical cluster tree. Moreover the category confusion table is presented after having permuted both its rows and columns in the same way so that they fit to the leafs of the adjacent hierarchy tree.
Combining Segmentation with Recognition

A basic challenge of the compositional approach is to find groupings of object parts that are characteristic for object categories. In Chapter 5 several Gestalt laws of perceptual organization have been integrated in a criterion function that yields perceptually salient compositions. However, modeling such grouping algorithms is rather involved and it is consequently desirable to replace this stage. Therefore, this chapter investigates how image segmentation can be exploited for generating compositions of object parts. As a result segmentation will be integrated into the recognition process [RO06]. The following study is based on the approach discussed in the previous chapter. The novel, segmentation specific aspects will then be investigated and the integration within the previous recognition system will be presented. However, to avoid repetition the details of the previous approach will not be fully reiterated and so we refer to Chapter 5 for details on that model.

Image segmentation pursues the goal to detect meaningful structures in a cluttered scene. Most current segmentation techniques take a bottom-up approach. An exception is for instance the approach by Borenstein and Ullman [BSU04] that also incorporates object class information. Bottom-up segmentation is based on the assumption that visual similarity (brightness, texture, motion, etc.) is equivalent to semantic relatedness of image regions, i.e. that such regions show parts of the same object. In case this assumption holds, coherent image segments that correspond to the same object can be found by grouping or clustering image parts based on their similarity in specific appearance features. Semantically meaningful image segmentation, however, becomes very difficult for visually heterogeneous objects and under
poor data conditions like shadows, occlusions and noise where above assumption is violated. In such situations, the detected coherent units often do not coincide with our perception of objects in a scene. This discrepancy is mostly due to the fact that the human visual system incorporates a significant degree of side information into the segmentation process. An important source of side information is prior knowledge about object categories which lays the foundation for top-down groupings. Thereby visually dissimilar regions of a scene can still be agglomerated in a common segment when they can be identified with parts of the same, previously seen object.

Despite the generally poor quality of bottom-up segmentations in real-world images, the following experiments demonstrate that it is possible to exploit low-level segmentations for building a powerful object recognition system. The crucial design decision is to use not only one segmentation, but a whole ensemble of segmentations which often captures at least parts of the objects in a scene. Such partial matches of the object boundaries can be successfully used for discriminating between foreground and background segments. A grouping of local features along the object boundaries yields then category specific compositions that capture the boundary shape of objects as well as their appearance.

Based on the candidate foreground segments two different approaches to category-level object recognition will be investigated: the direct approach exclusively relies on the low-level segmentations by computing majority votes over all stable segments in an image, whereas the combined approach uses the predicted foreground segments as input for the hierarchical composition system that has been studied in Chapter 5. The latter learns to group parts of the image foreground segments into a hierarchy of category-specific compositions, and binds them together using a probabilistic shape model to recognize objects in scenes.

6.1 Ensembles of Low-level Segmentations

For image segmentation an adapted version of the algorithm proposed in [RL04] is used which combines both the ideas of partitioning and feature combination/selection. The latter aspect turns out to be important for finding good segmentations, since segment-specific information is often spread over different cues like color and texture. To be more specific, processing starts by extracting feature vectors \( \mathbf{f}_i \in \mathbb{R}^d \) at a set of \( n \) image sites. The feature vectors consist of different cues such as color histograms and texture responses from Gabor filters. Sites are then assigned to image segments by using a Gaussian mixture model with \( k \) mixture components that share
an identical covariance matrix $\Sigma$. The data log-likelihood of this model is

$$
\log p(\{f_i\}_{i=1,\ldots,n}\mid\{\mu_\nu, \Sigma\}_{\nu=1,\ldots,k}) = \sum_{i=1}^{n} \log p(f_i\mid\{\mu_\nu, \Sigma\}_{\nu=1,\ldots,k}) = \sum_{i=1}^{n} \log \left[ \sum_{\nu=1}^{k} \gamma_\nu \cdot N(f_i; \mu_\nu, \Sigma) \right]
$$

and the mixture weights $\gamma_\nu$ sum to one. The model parameters $\mu_\nu$ and $\Sigma$ as well as the segment assignment probabilities can then be found by means of an expectation-maximization algorithm, [MK97]. In [RL04] the M-step is reformulated so that a constrained likelihood criterion is maximized. This criterion assures that only the most relevant features are selected.

In order to find reasonable settings for the free model parameters, a resampling-based model selection strategy is devised which follows largely [LRBB04, RL04]. The key idea is to draw multiple resamples of an image [ZB07]. In other words different image sites are sampled. Thereafter segmentations are inferred on the individual resamples and the resulting segmentations are compared afterwards. As opposed to arbitrary numbers of segments, an appropriate number of segments $k$ is much more likely to yield a stable segmentation that does not fluctuate under resampling. This procedure results in a stability-ranked list of prototypical segmentations.

In addition to selecting these stable segmentations, all individual segmentations are overlaid to compute a probabilistic boundary map that encodes for each pixel its probability of being part of a segment boundary, see Figure 6.1 for a schematic overview. Despite the fact that many individual segmentations are often of rather poor quality, the ensemble approach has two important advantages: (i) Within the subgroup of stable segmentations we often find relatively good partitions; (ii) the aggregated boundary map typically captures many details of the object in the image. To highlight the latter issue, the response of a Canny edge-detector is plotted in the right panel of Figure 6.1. Due to the local character of the edge detection process, the Canny edges are much more noisy than the aggregated segment boundaries.

### 6.2 Experimental Setup

The following sections will investigate ways to integrate image segmentation into the process of object recognition. The resulting object recognition system is then evaluated on a 20 category subset of the Caltech-101 database since the process of establishing ensembles of segmentations is computationally too intensive for the full
Chapter 6. Combining Segmentation with Recognition

Figure 6.1: Ensembles of segmentations. Left: input image and extracted features (top: three texture channels, bottom: LUV color channels). Middle: resampled image sites and corresponding segmentations. Top right: probabilistic boundary map found by overlaying all individual segment boundaries. Bottom right: Canny-edges for comparison.

dataset. The subset is made up of the categories anchor, umbrella, barrel, trilobite, wrench, windsor_chair, tick, stapler, electric_guitar, gramophone, stop_sign, cup, lobster, crayfish, water_lilly, crab, starfish, wild_cat, pyramid, pagoda. As is in the other chapters only very limited supervision is provided in the training phase—training images together with the category label of the prominent object. However, additional information about the location of the objects or a hand-segmentation is not available.

The choice of the 20 category subset was guided by two criteria: the subset should by reasonably small (≈ 1000 images) and it should by sufficiently difficult to reliably evaluate the performance. The chosen categories are a mixture of artificial and natural object classes and they contain some classes that are very difficult to separate like lobster and crayfish. For each class a training set of 25 images is randomly selected. The remaining images are exclusively used for performance evaluation.

6.3 Identifying Object Segments

Based on ensembles of segmentations, the following introduces a method for identifying foreground segments that contain objects of interest. This foreground learning takes place in a pairwise setting. First two object categories are chosen at random. For all training images belonging to these two categories, all segmentations are considered which exceed a certain stability threshold (see section Section 6.1) and we extract the boundary of each segment. On regularly spaced points along these curves, vectors of local image descriptors are extracted. Thus, each connected segment is represented as a string of vectors. The same procedure is applied to the
6.3. Identifying Object Segments

Figure 6.2: Boundary extraction. Left: original image; middle: most stable segmentation; right: three extracted segments (blue) and their boundaries (red).

Training images in the background class. Putting all such strings together we obtain a dataset consisting of $n$ boundary strings from two categories and the background class. We then compute local string alignments for all pairs of these $n$ strings. The final ($n \times n$) matrix of alignment scores is transformed into a valid Mercer kernel. In order to discriminate between fore- and background segments, a Gaussian mixture model is learned with three modes on the boundary strings which are represented by the kernel matrix. The estimated membership probabilities in one of the modes are used for identifying foreground segments: those segments that have a high probability for the correct image category are treated as foreground areas. All steps are now explained in further detail.

6.3.1 Boundary Extraction and String Representation

After the segmentation process, each pixel in an image carries a segment label. In a first step, connected pixels which share the same label are extracted. Such connected groups of pixels are referred to as segments in the sequel. For each of these segments, we compute a chain-code representation of the segment boundary. Such a boundary is called closed if the segment is entirely contained in the image, i.e. if it does not hit the image borders. For such closed segments the boundary chain is extended to two full circulations, which guarantees that the alignment score between two such segments becomes independent of the starting point (note that local alignments are used). If a segment is not closed, we start at the image border and continue the chain until the border is hit again. Figure 6.2 depicts examples of such segment boundaries.

On regular intervals along the segment boundaries, a vector of image descriptors is extracted. The components of such a vector contain three different descriptor types: a shape context histogram, a texture patch and a gray-value patch. The shape context descriptor [BMP01] consists of a log-polar histogram with 60 bins (10 angles, 6 scales) which is centered at the current position along the boundary. Each bin represents the (weighted) sum of those pixels in the map of aggregated segment
Chapter 6. Combining Segmentation with Recognition

Figure 6.3: String representation of segments. Left to right: chain of vectors containing local image descriptors (schematic), segment boundary (red) and vicinity around the segment (green), polar histogram of the shape context descriptor (yellow), polar gray-value patch, polar texture patch.

boundaries which fall into the bin-specific part of the polar histogram and which are “close” to the segment, i.e. which lie in a close vicinity of the segment, see the green tube around the segment in Figure 6.3. The texture- and gray-value patches consist of locally averaged values of Gabor filter responses and image intensities respectively. In analogy to the shape context descriptor, a polar grid is used for defining the areas over which the averaging takes place. This polar geometry has the advantage that we can easily incorporate rotation invariance into the alignment process by simply shifting the indices of the descriptors.

6.3.2 String Alignments

Having extracted a string for each of the \( n \) segments, the \( n \times n \) matrix of pairwise local alignments is computed by way of the Smith-Waterman algorithm [SW81]. Contrary to the typical setting in which this algorithm is used, in the present case there exists no fixed alphabet of symbols for which a predefined scoring table for aligning pairs of these symbols is available. We rather have “strings” which are ordered collections of vectors with real-valued entries. Instead of looking up the symbol-wise scores in a table, in each step of the algorithm, a scoring function is evaluated for two vectors. The components of these vectors consist of 60 bins of a shape context histogram, 60 locally averaged texture measurements and 60 locally averaged gray-values. Thus, a vector is composed of three subvectors,

\[
\mathbf{f} = (\mathbf{f}_{\text{shape}}, \mathbf{f}_{\text{text}}, \mathbf{f}_{\text{gray}})^	op
\]

In the experiments below a simple aggregation of these three cues is employed that combines \( \chi^2 \) distances between shape context histograms with correlation scores for texture and intensity patches: the scoring function for two vectors \( \mathbf{f}_1, \mathbf{f}_2 \) has the form

\[
s(\mathbf{f}_1, \mathbf{f}_2) = a - b \cdot \left( D_{\chi^2}(\mathbf{f}_{1\text{shape}}, \mathbf{f}_{2\text{shape}}) + D_{cc}(\mathbf{f}_{1\text{text}}, \mathbf{f}_{2\text{text}}) + D_{cc}(\mathbf{f}_{1\text{gray}}, \mathbf{f}_{2\text{gray}}) \right),
\] (6.3)
with the $\chi^2$ distance $D_{\chi^2}(f_1, f_2)$,
\[
D_{\chi^2}(f_1, f_2) := \frac{1}{2} \sum_j \frac{(f_1(j) - f_2(j))^2}{f_1(j) + f_2(j)}
\]
(6.4)
and the cross correlation distance $D_{cc}(f_1, f_2) = 1 - |cor(f_1, f_2)|$, with $cor(f_1, f_2)$ being the correlation between the vectors $f_1$ and $f_2$. Note that distances are transformed into similarities, so that a high score means that two strings are similar. The constants $a = 1/2$, $b = 1/3$ were selected empirically.

Since the extracted segments often capture only parts of the objects, the alignment scores are divided by the length of the alignment. In order to avoid high scores for very short random alignments, we consider such length-normalized alignments as valid only if the total alignment length exceed a certain threshold. In the subsequent experiments it is required that two strings must be aligned at more that 15 consecutive positions, otherwise the score is down-weighted by a factor of ten. For a better geometric interpretation, such positions which align to each other are depicted in Figure 6.4 as blue lines.

To further decrease the sensitivity to local segmentation errors, gaps in the alignments are allowed. Such gaps are penalized by a predefined cost value $\eta$. In our experiments we use $\eta = 0.1$ which means that the current alignment score is decreased by 0.1 whenever a position in one string is aligned with a gap in the other. For two strings $x, y$ with lengths $l, l'$ the alignment algorithm recursively fills the $(l \times l')$ matrix $F$:
\[
F(i, j) = \max\{0, F(i - 1, j - 1) + s(x_i, y_j),
F(i - 1, j) - \eta, F(i, j - 1) - \eta\}.
\]
(6.5)

Backtracking from the highest value in $F$ yields the optimal alignment, see [SW81] for details. Recall that the $i$-th position of string $x$ is a vector consisting of shape, texture and intensity, and that $s(\cdot, \cdot)$ denotes the scoring function defined in (6.3). An example alignment matrix for the categories “wrench” and “windsor_chair” is depicted in the right panel of Figure 6.4, which shows a distinct block structure.

### 6.3.3 Detecting Foreground Segments

In the final step of the foreground-detection process a Gaussian mixture model is learned for the $n$ segments. These segments are represented in form of a $(n \times n)$ matrix $K$ of pairwise alignment scores. If this matrix would be positive semidefinite
we could identify it as a Mercer kernel and train a mixture model in the kernel-induced space as proposed e.g. in [RS99]. It is well known that probabilistic alignment models such as pair hidden Markov models produce scores which fulfill the requirements of a valid Mercer kernel. For computational complexity reasons, however, a deterministic alignment model has been used above which might violate the positive-semi-definiteness condition. Moreover, the length-normalization of scores can lead to additional negative eigenvalues of $K$. In practice, however, it can be observed that there are typically only very few negative eigenvalues which are all of small magnitude. In order to transform it into a valid Mercer kernel, the kernel PCA idea [SSM98] can be employed to find a decomposition $K = VAV^\top$ with a diagonal matrix $\Lambda$ of eigenvalues. Discarding all negative eigenvalues a valid kernel $K' = V_+\Lambda_+V_+^\top$ is formed.

Based on this kernel matrix $K$ a Gaussian mixture model with 3 mixture modes is learned in a next step. For initialization all segments in an image are labeled according to the overall category label of the image, despite the fact that some segments might belong to the background class. During further iterations of the EM algorithm, these membership probabilities in one of the three classes (two categories + background) are re-estimated for each segment. It is interesting that the selection of foreground segments does not vary significantly if different pairs of categories or if more than two categories are selected. Examples of detected foreground segments are depicted in Figure 6.5.

To predict foreground segments in novel test images the size of the training set is first of all reduced by extracting from each training image per category only the two highest scoring foreground segments. In the following experiments 25 training images are randomly selected per category and the 20 category subset of the Caltech-101
6.4 Object Recognition using Object Boundary Fragments

In order to exploit the results of the foreground identification for the purpose of object recognition, the following uses the classifier that was learned on the basis of the training images as described above to predict foreground segments in novel test images. For this purpose each segment in a test image is aligned with all \( n = 1000 \) training segments. The resulting 1000-dimensional alignment vector is projected onto the set of eigenvectors \( V^+ \) of \( K' \). Appropriate scaling by the eigenvalues (see [SSM98]) yields a vectorial representation \( \mathbf{x} \) of the test segment, for which the classifier predicts a set of membership probabilities in each of the 20 image categories. Segments that
can be clearly assigned to one of the categories (i.e. which have a high membership probability) are considered as hypothetical foreground segments in a test image.

### 6.4.1 Direct Approach

These hypotheses are now used for predicting the category labels of the test images in two different ways: the direct approach computes a weighted majority vote over all segments in a test image. When assigning each image the most probable category label, the average retrieval rate of the direct approach is 58.3%. Among the two most probable categories, the correct one can be found in $\approx 71\%$, and among the three most probable in $\approx 79\%$. Taking into account that the direct approach only uses low-level segmentations and that for roughly 1/4 of all images it seems to be very difficult to find any good segmentations, these retrieval rates are surprisingly high. For comparison: the implementation of the approach from Chapter 5 achieves an average retrieval rate of 61.8% when trained exclusively on these 20 categories. For analyzing the effect of using many segmentations per image (the experiments use 100 segmentations per image), the whole processing pipeline is repeated with only 5 segmentations per image. In this setting, the average retrieval rate drops down to 26% which effectively demonstrates the advantage of using large ensembles.

In an additional experiment a bag-of-features approach is applied to the hypothetical foreground segments. This baseline approach yields only 49% retrieval rate.

### 6.4.2 Combined Approach: Compositional Hierarchies of Object Segments

The combined approach uses the boundaries of the predicted foreground segments as input for the compositionality-based recognition system from Chapter 5. Therefore, the segment boundary contours are first split at points of high curvature. As outlined in Section 5.3 salient base compositions are formed by grouping all local patches in the neighborhood of a curve. In the training phase top-down grouping probabilities are learned as described in Section 5.4. In the recognition phase, the top-down grouping from Section 5.5.1 is then applied to the compositions at segment boundary curves to obtain a compositional hierarchy. Finally, the established compositions enter into the compositional shape model as outlined in Section 5.5.2. This model predicts the object category by estimating the category posterior conditioned on the boundary curves that result from the observed segmentations.

An evaluation of this combined approach yields an average retrieval rate of 62.3%
which is at least competitive to the reference model from Chapter 5 (however, the increase in performance is probably not statistically significant). Figure 6.6 shows the corresponding category confusion table. This result shows that despite the difficulties of low-level segmentation, it is possible to exploit the information contained in ensembles of segmentations for building state-of-the-art recognition systems.

![Category Confusion Table](image)

Figure 6.6: Category confusion table of the compositional model with 62.3% retrieval rate.
Learning to Establish Relevant Compositions

In the preceding chapters different methods for establishing compositions have been investigated. Whereas Chapter 5 has studied a grouping algorithm based on a number of perceptual Gestalt laws, Chapter 6 has used ensembles of segmentations for that purpose. The first approach requires the manual design of a grouping function that combines several perceptual criteria and it is, therefore, a rather laborious modeling process. This problem has been avoided by the approach based on ensembles of segmentations in Chapter 6. However, the large number of segmentations that are necessary for this approach lead to unfavorable computational costs.

Hence, one may ask for a computationally feasible method to establishing compositions that automatically learns how to form compositions rather then requiring intensive manual modeling. However, in the training phase no supervision regarding the compositional nature of objects is available—only images with an overall class label are given. So the question is: How can a system automatically learn the compositional structure of complex, real world object categories without being instructed about compositions in the training phase? This chapter addresses this challenge by following a purely Bayesian approach. The resulting Bayesian criterion function retrieves relevant compositions that are characteristic for categories. Moreover, a statistically feasible learning algorithm will be presented.

Subsequently, methods for learning the compositional structure of objects are investigated and they are integrated into a category level object recognition system (cf. [OB07b]). The approach learns characteristic compositions of atomic parts for each category in an unsupervised manner, requiring neither hand segmentations nor
object localization during training. In the same way higher level compositions of compositions are learned. Finally, a Bayesian network serves as a coherent model that comprises all the compositional constituents together with object shape. Inference based on this probabilistic model yields a decomposition of a scene into a hierarchy of relevant compositions and, finally, enables localization and recognition of objects. Moreover, the underlying generative model can be used to sample object representations and to explain away background clutter.

The compositional object recognition model realizes feature sharing on the lowest level on which robust statistics are available. To this end, edge and color distributions of small image patches are computed as described in Section 3.2.2. A generic set of atomic parts that is shared among categories is established by forming a small codebook of these features. Category specific relations between parts are used to build compositions that are represented by probability distributions over their constituent parts, i.e. distributions over atomic parts yield compositions, distributions over compositions yield higher level compositions of compositions, and so on. Finally, a statistical, hierarchical scene representation is obtained by capturing the spatial arrangement and coupling all compositions in a single probabilistic model, i.e. the compositional shape model. Therefore, compositions are coupled by means of (i) their spatial arrangement, (ii) by forming relations between compositions that yield higher level compositions, and (iii) by the co-occurrence of all compositions which roughly describes the context of the scene.

7.1 Outline of the Approach

The following gives a brief overview over the approach to compositional scene analysis (illustrated in Figure 7.1). A detailed account of the individual processing steps of the algorithm presented in Algorithm 7.1 is then given in later sections. Moreover the learning of the underlying model will also be explained later on. As in the previous chapters, processing of a novel image starts by extracting a set of small patches $e_i \in \mathcal{E}(\sigma)$. Here these patches are, however, obtained on several different scales $\sigma$ at interest points. They serve as atomic parts in the compositional hierarchy and for all of them localized feature histograms (Section 3.2.2) are computed. Each image region is then represented by a discrete probability distribution over a small codebook of feature prototypes which is shared by all objects (see Section 3.3). As patches are only local features and the codebook is shared by all categories, these atomic parts alone are far from being category specific. Therefore, compositions $g_j \in \mathcal{G}(\sigma)$ of these
7.1. Outline of the Approach

The aim of the presented approach is to avoid a manual modeling of a set of grouping laws that lead to characteristic compositions (e.g., using perceptual grouping). Rather, a learning strategy that automatically learns to establish relevant compositions is to be developed. Hence, only a simple grouping, which is based on the proximity of constituents, is employed. The intention of this step is to limit the search space for the subsequent relevance learning by clustering parts to form candidate compositions $\mathcal{G}^{(\sigma)}$. This grouping is a very crude filtering process after which there is still a great deal of clutter present among the candidates. Out of these candidates, relevant compositions are selected by a relevance model that has been learned during the training phase. The relevant compositions enter into a Bayesian network—the compositional shape model. In this statistical model, all compositions are coupled by means of the co-occurrence and spatial distribution of all compositions so that the individual object hypotheses that they induce can be brought into accordance.

At this stage a large fraction of all possible object categorization hypotheses can already be rejected with high confidence. That is, given the accumulated evidence many categories are very unlikely to be present in the scene. Conditioned on each of the remaining hypotheses a set $\tilde{\mathcal{R}}$ of relevant compositions of compositions (the con-
7.2 Atomic Parts on Multiple Scales

To establish the atomic parts of the compositional hierarchy the feature extraction described in Section 3.2.2 is again followed. This time however, the image is rescaled to scale $\sigma$ before extracting the $20 \times 20$ pixel patches at interest points. The approach that is discussed in this chapter employs three scales $\sigma \in S = \{1, \frac{1}{2}, \frac{1}{4}\}$. For $\sigma = \frac{1}{2}$,
7.2. Atomic Parts on Multiple Scales

ObjectRecognition(I) \triangleright I: a novel test image

1. \(P \leftarrow \text{INTEREST POINTS}(I)\)
2. \(\triangleright \text{extract atomic parts and form composition candidates:}\)
3. for all scales \(\sigma \in \mathcal{S}\)
4. \(E^{(\sigma)} \leftarrow \text{ATOMIC PARTS}(I, P, \sigma)\)
5. \(G^{(\sigma)} \leftarrow \text{COMPOSITION CANDIDATES}(E^{(\sigma)})\)
6. \(G \leftarrow \bigcup_{\sigma \in \mathcal{S}} G^{(\sigma)}\)
7. \(g' \leftrightarrow \frac{1}{|G|} \sum_{g_j \in G} g_j\)
8. \(\triangleright \text{context descriptor from (4.57)}\)
9. \(\triangleright \text{estimate from (7.1)}\)
10. \(x \leftarrow \frac{\sum_{E \in G} \sum_{c \in \mathcal{C}} p(E|g') P(E|g')}{\sum_{E \in G} \sum_{c \in \mathcal{C}} p(E|g') P(E|g')}\)

11. \(R \leftarrow \text{RELEVANT COMPOSITIONS}(G, x)\)
12. \(\triangleright \text{recognize object category:}\)
13. \(P(c|E) \leftarrow \text{COMPOSITIONAL SHAPE MODEL}(R, g', x, \emptyset)\)

\[\tilde{G} \leftarrow \text{COMPOSITION}^2 \text{CANDIDATES}(G)\]
\[\tilde{R} \leftarrow \text{RELEVANT COMPOSITIONS}^2(\tilde{G}, P(c|E))\]
\(\triangleright \text{update previous object category hypothesis by using comp. } R\)
\(\text{and higher order comp. } \tilde{R} \text{ for recognition:}\)
\(P(c|E) \leftarrow \text{COMPOSITIONAL SHAPE MODEL}(\tilde{R}, g', x, \tilde{R})\)
\(\text{return } P(c|E), x\)
\(\triangleright \text{object category posterior and location}\)

Algorithm 7.1: Algorithm for compositional scene analysis.

for instance, this corresponds to subsampling to half the original scale. The patches
are then represented using localized feature histograms yielding feature vectors \(e^{(\sigma)}_i\)
according to (3.5). All features on the same scale are summarized in a set \(E^{(\sigma)}\) (see
line 3 of Algorithm 7.1).

In the training phase a codebook of atomic parts from all scales is learned that
is shared by all categories (see Section 3.3). Atomic parts are then represented by
means of the Gibbs distribution from (3.7).
7.3 Composition Candidates

Different methodologies for establishing category specific compositions are conceivable. In Chapter 5 several grouping laws which are rooted in perceptual organization have been combined in a carefully designed, complex grouping algorithm. Subsequently, an antithetic approach is pursued that automatically learns to build category specific compositions without requiring any prior knowledge regarding the compositional nature of objects. Therefore, a large number of potential candidate compositions are formed by means of a simple proximity based grouping procedure. A learning algorithm is then applied to remove irrelevant compositions and, thereby, the tedious task of retrieving relevant compositions is solved automatically as will be shown in Section 7.4.

From all local descriptors that have been extracted as described in Section 7.2, a random subset of 40 is selected on each scale \( \sigma \) of an image making it 120 in total per image. Thereafter, a proximity grouping is applied that clusters each of these parts with all the parts in its local neighborhood and, as a result, composition candidates are obtained for each scale. Therefore, a grouping diameter is chosen that has the width of 3 local patches. This size offers a good trade-off between localization of compositions (small compositions support exact localization of image particularities) and the statistical robustness of their estimation during training (large compositions integrate over a large number of samples so that they become robust against outliers).

According to Section 4.2, compositions are again represented by a mixture distribution (4.2) over their parts. On each scale \( \sigma \) candidate compositions \( g_j \) are obtained that are pooled in the set \( G^{(\sigma)} \). This concludes the explanation of line 4 of Algorithm 7.1.

Object Localization

After all candidate compositions have been extracted out of an image, the object is to be localized. In a first step, scene context is captured in the co-occurrence descriptor \( g^I \) as presented in (4.57). Thereafter, the object position \( x \) in the image is estimated as described in Section 4.5.2,

\[
    x = \frac{\sum_{j : g_j \in \Phi} x_j \sum_{c \in L} p(g_j | c, g^I) \cdot P(c | g^I)}{\sum_{j : g_j \in \Phi, c \in L} p(g_j | c, g^I) \cdot P(c | g^I)}.
\]  

(7.1)
7.4 Learning Relevant Compositions of Parts

A significant number of composition candidates do actually only capture clutter such as background or other unspecific regions of the scene that are not characteristic for the object category that is present. The goal is therefore to discard such distractors that would otherwise sidetrack the compositional model. Subsequently, an approach will be developed that automatically learns to retrieve those compositions which are actually relevant for our task of recognizing object categories—this implies that categories have to be distinguished from another.

7.4.1 A Bayesian Criterion for Compositional Relevance

From a Bayesian point of view a composition $g_j$ that is drawn from an image $I$ is relevant for representing objects of some category $c$ if it has a high likelihood $p(g_j | \chi_c)$. The indicator function $\chi_c(I) \in \{0, 1\}$ specifies whether an image $I$ contains an object of category $c$,

$$
\chi_c(I) := \begin{cases} 
1, & I \text{ contains an object of category } c, \\
0, & \text{otherwise}.
\end{cases}
\quad (7.2)
$$

By applying Bayes’ theorem the likelihood factorizes

$$
P(\chi_c | g_j) = \frac{p(g_j | \chi_c) P(\chi_c)}{p(g_j)}. \quad (7.3)
$$

Since there should be no bias on the prior probabilities of categories, i.e. all categories are a priori equally likely, $P(\chi_c)$ can be dropped and the likelihood becomes proportional to

$$
p(g_j | \chi_c) \propto P(\chi_c | g_j) p(g_j). \quad (7.4)
$$

Now the estimate of compositional relevance from (7.4) can be refined by incorporating the estimate of the object center $x$ as well as the position $x_j$ of a composition,

$$
p(g_j | \chi_c, x_j, x) = p(g_j | \chi_c, S_j = x - x_j) \quad (7.5)
\propto P(\chi_c | g_j, S_j = x - x_j) p(g_j | S_j = x - x_j). \quad (7.6)
$$

Here we exploit the fact that compositions depend on their shift relative to the object center, $S_j = x - x_j$, but not on their absolute location in the image.
7.4.2 A Statistically Feasible Learning Algorithm for Compositional Relevance

Compositional relevance as defined in (7.4) and (7.6) factorizes into two distributions that measure the discriminative power and detection reliability of compositions. The first factor expresses how powerful a composition $g_j$ is in discriminating object categories from another whereas the second factor indicates how reliably $g_j$ can be detected. While the first distribution can be estimated using discriminative learning, the second one requires a statistically intractable density estimation in a high dimensional feature space. This problem is now tackled by means of an approach that is based on cross-validation (see illustration in Figure 7.2).

The idea is to learn the posterior distribution $P(\chi_c|g_j, s_j)$ on one part of the training data and use it to predict the relevance of compositions in the other part. Unfavorable compositions with low prior $p(g_j|s_j)$ have a low probability to also appear in the validation set. As a consequence, validation prevents the learning algorithm from overfitting to the compositions extracted from the training set.

The learning algorithm which is summarized in Algorithm 7.2 starts by randomly splitting $T_c^{(+)}$, the training images of category $c$, into two disjoint subsets $T_c^{(1)}$ and $T_c^{(2)}$ of equal size. Moreover, a set of irrelevant compositions $T_c^{(0)}$ has to be established by taking a random sample of compositions from all categories other than $c$. In the absence of a proper background set, these compositions act as a negative set against which the relevant ones are learned. Thereafter the problem of discriminating compositions in $T_c^{(1)}$ from the irrelevant ones in $T_c^{(0)}$ is tackled by training a probabilistic 2-class classifier (NKDA is used for that purpose) on these sets. This procedure yields an estimate of the distribution $P(\chi_c|g_j, s_j)$. To discard erroneously detected, irrelevant compositions in $T_c^{(1)}$ (e.g. background clutter or other objects) this classifier is employed to predict the relevance of compositions from the other half of the training samples $T_c^{(2)}$ which, thereby, act as a validation set. Given this ranking, the subset $R_2 \subset G_2$ of cardinality $\rho$ with highest relevance is selected from all compositions on the validation set ($\rho$ is set to retain 50% of the original compositions).

In conclusion, this validation step avoids overfitting to outliers in the first half of the training samples by validation on the second. This procedure is then repeated with the roles of $T_c^{(1)}$ and $T_c^{(2)}$ interchanged. Finally, the relevant subsets of each half of the training data are merged to train a single NKDA classifier that can be used to predict the relevance of compositions in novel test images. In Section 7.4.3 a visualization of the learned compositions is presented.
7.4. Learning Relevant Compositions of Parts

\begin{algorithm}
\begin{algorithmic}[1]
\State \textbf{RelevanceLearning}(\mathcal{T}_c^{(+)}, \mathcal{T}_c^{(0)})
\State \Comment{\(\mathcal{T}_c^{(+)}\): set of training images for category \(c\), \(\mathcal{T}_c^{(0)}\): irrelevant images of categories other than \(c\)}
\State \Comment{collect compositions for all images in each training subset:}
\For {\(i \in \{0, 1, 2\}\)}
\State \(\mathcal{G}_i = \emptyset\)
\For {\(I \in \mathcal{T}_c^{(i)}\)}
\State \(\mathcal{G}_i \leftarrow \mathcal{G}_i \cup \text{AllCompositionCandidates}(I)\)
\EndFor
\EndFor
\Comment{Lines 1 – 5 of Alg. ObjectRecognition}
\For {\(i \in \{1, 2\}\)}
\State \(p(\chi_c|\mathbf{g}_j, \mathbf{s}_j) \leftarrow \text{LearnProbClassifier}(\mathcal{G}_0, \mathcal{G}_i)\)
\Comment{Train on \(i\)-th train set}
\State \(p(\chi_c|\mathbf{g}_j, \mathbf{s}_j)|_{\mathbf{g}_j \in \mathcal{G}_{\{1,2\}\neg i}} \leftarrow \text{Predict}(p(\chi_c|\mathbf{g}_j, \mathbf{s}_j), \mathcal{G}_{\{1,2\}\neg i})\)
\Comment{Predict on other half}
\State \(\mathcal{R}_{\{1,2\}\neg i} \leftarrow \text{subset (cardinality } \rho \text{) of } \mathcal{G}_{\{1,2\}\neg i} \text{ with highest } p(\chi_c|\mathbf{g}_j, \mathbf{s}_j)|_{\mathbf{g}_j \in \mathcal{G}_{\{1,2\}\neg i}}\)
\Comment{Learn compositional relevance on both fractions of the train set:}
\State \(\mathcal{R} \leftarrow \mathcal{R}_1 \cup \mathcal{R}_2\)
\State \(p(\chi_c|\mathbf{g}_j, \mathbf{s}_j) \leftarrow \text{LearnProbClassifier}(\mathcal{G}_0, \mathcal{R})\)
\State \textbf{return} \(p(\chi_c|\mathbf{g}_j, \mathbf{s}_j), \mathcal{R}\)
\EndFor
\end{algorithmic}
\end{algorithm}

7.4.3 Visualizing Relevant Compositions

This experiment analyzes the learning of relevant compositions. Therefore, all compositions from the training data that have been predicted to be relevant for a category \(c\) are clustered using histogram clustering (to this end the clustering approach of [PHB99] is employed). The relevances of all compositions that are assigned to a centroid are averaged and the centroids with highest relevance for the category are presented in Figure 7.3. To obtain this visualization there is still a difficulty that has to be solved. Compositions cannot be visualized directly since they are represented by distributions over a codebook of atomic parts. Therefore, each centroid is depicted by plotting the three closest representatives that have been assigned to it during clustering.
Figure 7.3: Clustering of relevant compositions. For each category, the two centroids with highest relevance are illustrated by visualizing the three closest compositions to that prototype. a) airplanes, b) bass, c) crayfish, d) dragonfly, e) faces, and f) hawksbill.

### 7.5 Learning Higher Order compositions

To incorporate additional information into object models, direct dependencies between compositions can be captured by learning groupings of compositions. To build these higher order compositions, random tuples of compositions \((g_k, g_l)\) are estab-
7.6 Binding Compositions in a Compositional Shape Model for Object Classification

For object classification all compositions are combined in a single object model so that a concerted categorization hypothesis can be derived. Figure 7.4 illustrates the different pieces of information that are to be integrated in the model. This

\[ p(g_k, g_l, r_{kl} | \chi_c) \propto P(\chi_c | g_k, g_l, r_{kl}) p(g_k, g_l, r_{kl}). \]  

This score is then be plugged into Algorithm 7.2 so that learning the relevance of higher order compositions proceeds again according to Section 7.4.
structured object model should couple (i) the appearance of salient object regions, (ii) object shape (that is the geometry of the local features), and (iii) the scene context in which the object appears. To this end the compositional shape model from Section 4.5.4 is extended. This Bayesian network is illustrated in Figure 7.5. The appearance of object regions is covered by compositions, whereas the shape is represented through spatial relations between compositions and the object center. Moreover higher order compositions, which establish spatial relationships between pairs of compositions, contribute to shape representation. Lastly, scene context is captured by the co-occurrence of all compositions \( g_j \in G \) that have been established in the image. Therefore, a mixture \( g^I \) of the distributions of all compositions is computed as presented in (4.57).

All of above evidence on appearance, shape, and context is then coupled in the Bayesian network from Figure 7.5 to yield a joint hypothesis on the object category \( c \in L \) which is expressed by the category posterior

\[
P(c|g^I, \{g_j, x_j\}_{j: k_j \in R}). \tag{7.8}
\]

### 7.6.1 Selecting Relevant Compositions

Here the set of relevant compositions \( R \) is again formed by retaining those 50% of all compositions \( g_j \in G \) that have highest relevance score (7.6), i.e.

\[
R := A : A \subset G
\]

\[
\wedge |A| \approx \frac{1}{2} |G| \tag{7.9}
\]

\[
\wedge \forall g_j \in A, g_{j'} \in G - A : \max_{\substack{c \in \mathcal{L}}} P(\chi_c|g_j, s_j) \geq \max_{\substack{c \in \mathcal{L}}} P(\chi_c|g_{j'}, s_{j'}).\]

This concludes line 8 of Algorithm 7.1.

Relevant higher order compositions are selected by means of a top-down grouping. Candidate composition tuples are formed by randomly drawing pairs of compositions from \( G \times G \). The most relevant candidates from this set \( \tilde{G} \subset G \times G \) are then selected by maximizing the relevance score (7.7). This selection proceeds along the lines of (7.9). This time, however, we can make use of the category hypotheses that have been established based on single compositions (7.8). Therefore, only those composition tuples \((g_k, g_l) \in \tilde{G} \times G\) are sought that are most relevant for the ten categories with highest posterior (7.8). Let \( L_R \subset \mathcal{L}, |L_R| = 10 \) denote the ten most likely categories
given the posterior (7.8).

\[
\tilde{R} := \mathcal{A} : \mathcal{A} \subset \mathcal{G} \times \mathcal{G}
\]

\[
\land |\mathcal{A}| \approx \frac{1}{2} |\mathcal{G}|
\]

\[
\land \forall (g_k, g_l) \in \mathcal{A}, (g_{k'}, g_{l'}) \in (\mathcal{G} \times \mathcal{G}) - \mathcal{A} : \max_{c \in \mathcal{L}} P(c|g_k, g_l, r_{kl}) \geq \max_{c \in \mathcal{L}} P(c|g_{k'}, g_{l'}, r'_{k'l'}) .
\] (7.10)

Consequently, singleton compositions are used to establish an initial object hypothesis that helps to focus the search for higher order compositions. These higher order compositions are then used to refine the initial hypotheses and confirm the correct one. For Caltech-101 a subset of ten categories has been chosen because the correct class is among this set in more than 90% of all cases.

### 7.6.2 Object Recognition using Statistical Inference

Now the category posterior conditioned on all compositions and higher order compositions is derived—the posterior for singletons is then a special case of this distribution. Let us abbreviate the posterior by using \(E\) as a shorthand notation for the collected evidence. We start by applying Bayes’ formula,

\[
P(c | E) = P(c | g^I, x, \{g_j, x_j\}_{g_j \in \mathcal{R}}, \{g_k, g_l, r_{kl}\}_{(g_k, g_l) \in \tilde{R}})
\]

\[
= \frac{p(\{g_j, x_j\}_{g_j \in \mathcal{R}}, \{g_k, g_l, r_{kl}\}_{(g_k, g_l) \in \tilde{R}} | g^I, x, c) P(c | x)}{p(\{g_j, x_j\}_{g_j \in \mathcal{R}}, \{g_k, g_l, r_{kl}\}_{(g_k, g_l) \in \tilde{R}} | g^I, x)}
\] (7.11)

Now the evidence in the denominator can be neglected as it is independent of \(c\). Moreover, the nominator is factorized by exploiting the conditional independence properties expressed in the Bayes net of Figure 7.5, i.e. compositions are independent conditioned on the object model that is specified by classification \(c\) and localization \(x\).

\[
P(c | E) \propto p(\{g_j, x_j\}_{g_j \in \mathcal{R}}, \{g_k, g_l, r_{kl}\}_{(g_k, g_l) \in \tilde{R}} | g^I, x, c) P(c | x)
\]

\[
= p(\{g_j, x_j\} | x, c) \cdot p(\{g_k, g_l, r_{kl}\} | x, c) \cdot p(g^I | x, c) P(c | x)
\]

\[
= P(c | x) \cdot p(g^I | x, c) \times \prod_{g_j \in \mathcal{R}} p(g_j, x_j | x, c) \prod_{(g_k, g_l) \in \tilde{R}} p(g_k, g_l, r_{kl} | x, c)
\] (7.12)
Applying Bayes’ rule to the individual likelihoods yields
\[
P(c|E) = P(c, g^I|x) \times \prod_{g_j \in \mathcal{R}} \frac{P(c|x, g_j, x_j) \cdot p(g_j, x_j|x)}{p(c|x)}
\times \prod_{(g_k, g_l) \in \tilde{\mathcal{R}}} \frac{P(c|x, g_k, g_l, r_{kl}) \cdot p(g_k, g_l, r_{kl}|x)}{p(c|x)}. \tag{7.13}
\]

As we are computing the category posterior, factors that are independent of \(c\) can be neglected. Moreover, it is exploited that the object class is independent of the absolute position \(x\) of the object in the image. Only the relative positions of compositions with respect to the object center are retained since these shifts \(s_j = x - x_j\) represent object shape. Thus we obtain,
\[
P(c|E) \propto P(c|g^I, x) \cdot p(g^I|x) \times \prod_{g_j \in \mathcal{R}} P(c|x, g_j, x_j)
\times \prod_{(g_k, g_l) \in \tilde{\mathcal{R}}} P(c|x, g_k, g_l, r_{kl}) \tag{7.14}
\]
\[
\propto \exp\left[\ln P(c|g^I) + \sum_{g_j \in \mathcal{R}} \ln P(c|g_j, S_j = x - x_j)
+ \sum_{(g_k, g_l) \in \tilde{\mathcal{R}}} \ln P(c|g_k, g_l, r_{kl})\right]. \tag{7.15}
\]

For numerical stability the logarithm has been introduced in the last equation. Computing (7.8) is then a special case of this formula where the last sum over \(\tilde{\mathcal{R}}\) is to be omitted.

In conclusion, the category posterior from (7.11) is split up into three terms. The first represents scene context, the second object shape using singleton compositions, and the third higher order compositions. The individual distributions of (7.11) are all estimated using the probabilistic NKDA classifier. Consequently, classification has been formulated as a statistical inference problem that can be solved efficiently by computing the individual distributions and combining them in a closed-form solution. This model does not only classify an object. It also returns a confidence in this prediction.

### Computational Cost of the Approach

In terms of computational cost, training the algorithm on roughly 3000 Caltech-101 images and testing on the remaining 6000 images requires about 15 hours on
a Pentium 4 with 3 GHz. This is a competitive speed that is astounding for a structured object model.

7.7 Evaluation of the Compositional Approach

7.7.1 Results on Caltech-101

Subsequently, the compositional approach is evaluated on the challenging Caltech-101 database. Again, we follow the standard evaluation protocol: For training a random subset of equal size is drawn for each category. The approach is then tested on the remaining images. Moreover, 5-fold cross-validation is performed to obtain error bars, i.e. the same algorithm is run on 5 different splits of the data into training and test set.

Gain of Compositionality over a Baseline Model

In the following experiments, different aspects of the presented compositional approach will be investigated. To evaluate the gain of compositionality, we start with a model that discards the compositional structure completely and uses features from a single scale. For this baseline experiment the categorization system is restricted to the bag representation \( g^I \). Recognition is then based on maximizing \( P(c|g^I) \). This model achieves a retrieval rate of \( 35.3 \pm 0.8\% \) for 30 training images. In contrast to this, the full compositional model increases performance to \( 58.8 \pm 0.9\% \) and, thereby, emphasizes the value of a compositional object representation.

Multi-Scale Approach and Different Codebook Sizes

In the previous experiment recognition has been conducted only on a single scale. When processing images on 3 scales \( \sigma_1 = 1, \sigma_2 = 1/2, \) and \( \sigma_3 = 1/4 \) (where \( \sigma = 1 \) corresponds to the original image scale) the retrieval rate is further improved to \( 60.7 \pm 0.8\% \) (see Figure 7.6 for other training set sizes). The performances on the individual scales are \( 58.8 \pm 0.9\% \) when only \( \sigma_1 \) is used, \( 56.3 \pm 1.3\% \) for \( \sigma_2 \) alone, and \( 52.5 \pm 0.7\% \) for \( \sigma_3 \). The additional gain in retrieval rate of the multi-scale model shows that it effectively combines image representations from multiple scales to boost performance.

In Figure 7.7 the effect of different codebook sizes for representing compositions is investigated. Increasing the codebook from 200 in the previous experiments to 300
Figure 7.6: Retrieval rates of the full, multi-scale compositional approach for different training set sizes and a 200 dimensional codebook of atomic parts (retrieval rate for 30 training images is 60.7 ± 0.8%).

prototypes improves the performance to 61.3 ± 0.9%.

Analyzing the Established Category Hierarchy

Let us now analyze the final compositional model by investigating the resulting category confusion table. From the category confusion probabilities a measure of the mutual similarities (as estimated by the compositional system) between categories can be derived. As a result the established categorizations induce a hierarchical structure among the categories—a class hierarchy—which reveals the degree of relatedness of categories, as judged by the system. Computing this class hierarchy which is displayed in Figure 7.8 follows along the lines of Section 5.6.3.

The categories that are judged by the system to be most similar are “water-lilly” and “lotus”, “ketch” and “schooner”, and “crocodile” and “crocodile head”. These similarities are intuitive, since they are established between pairs that are either synonymous or at least semantically closely related.

7.7.2 A Comparison of Feature Descriptors for Atomic Parts

In Section 3.2.2 localized feature histograms have been discussed. In particular, it has been pointed out that the low dimensional representation is crucial to render the learning of compositional object models statistically feasible. Subsequently, this descriptor is contrasted with a common representation, David Lowe’s SIFT features [Low04]. Both representations are compared by plugging either of them as local feature $e_i$ into the single scale version (running on scale $\sigma_2 = 1/2$, using a 200
7.7. Evaluation of the Compositional Approach

Figure 7.7: Retrieval rates of the full compositional approach for different sizes of the part codebook. The algorithm is trained on 30 images per category.

Figure 7.8: Category hierarchy and category confusion table permuted to fit to the class tree. The retrieval rate is $61.3 \pm 0.9\%$.

dimensional codebook and 30 training images per category) of the compositional system.

However, let us first analyze the effective dimensionality of both types of descrip-
Chapter 7. Learning to Establish Relevant Compositions

Figure 7.9: Eigenvalue spectrum of a) localized feature histograms and b) SIFT features. See text for details.

 tors by randomly drawing 10000 features from all of the 102 Caltech-101 categories and applying principle component analysis (PCA) to the resulting merged feature set. Figure 7.9 shows the cumulative of the eigenvalue spectrum for both descriptors. Therefore, eigenvectors have been sorted according to ascending eigenvalue before plotting the cumulative sum of the data variance explained by each eigenvector. For the 40 dimensional localized feature histograms a small subspace of 10 dimensions already yields roughly 80% of the total variance and 20 dimensions capture more than 90%. For SIFT this plot shows a similar behavior. A fourth of the original dimensionality suffices to capture 80% of the data variance and half of all eigenvalues—that is roughly 60 dimensions—represent 90% of the variability. This comparison shows that the effective dimensionality of localized feature histograms is significantly lower than the dimensionality of SIFT features.
7.7. Evaluation of the Compositional Approach

<table>
<thead>
<tr>
<th>PCA on Localized Feature Hists.</th>
<th>10 dims</th>
<th>20 dims</th>
<th>Orig. 40 dims</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieval rate in %</td>
<td>55.5 ± 1.0</td>
<td>56.8 ± 1.0</td>
<td>56.3 ± 1.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCA on SIFT Features</th>
<th>20 dims</th>
<th>40 dims</th>
<th>60 dims</th>
<th>Orig. 128 dims</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieval rate in %</td>
<td>38.8 ± 0.6</td>
<td>39.6 ± 0.7</td>
<td>40.8 ± 1.1</td>
<td>40.6 ± 0.2</td>
</tr>
</tbody>
</table>

Table 7.1: Performance of the compositional approach (single scale $\sigma_2 = 1/2$, 200 dimensional codebook, 30 training images per category) using different local descriptors and after reducing their dimensionality with PCA.

Now either descriptor type is plugged into the compositional system and the results are summarized in Table 7.1. This evaluation underlines that localized feature histograms are effective in capturing local object particularities in a low dimensional representation. Moreover, they clearly outperform SIFT features as a representation basis for atomic compositional parts that are shared among categories.

7.7.3 Influence of Color in Localized Feature Histograms

Subsequently, it is analyzed how much color contributes to the performance of localized feature histograms. Therefore, the compositional approach from Section 7.7.2 (scale $\sigma_2 = 1/2$ and 200 dimensional codebook) is again used. Two different versions of local features $e_i$ are set up. The original 40 dimensional features described in Section 3.2.2 and a variant where the 8 bin color histogram has been replaced by a 4 bin histogram over the gray-scale channel, thereby yielding an overall 36 dimensional feature vector. Figure 7.10 compares the retrieval rates both approaches achieve for different training set sizes. On average, the color version yields a retrieval rate that is roughly 2% higher than that of the grayscale version. Therefore, the difference in performance between localized feature histograms and SIFT is only to a small extend caused by the additional color information.

7.7.4 Sampling a Compositional Representation from the Generative Object Model

During recognition, inference propagates information from local image features over intermediate compositions to an object category label. However, the graphical model from Figure 7.5 can also be applied in a generative manner: Given object category $c$ and object position $x$, compositions and, finally, image patches can be inferred. To obtain the image representation in a region around $x_j$, compositions $g_j$ have to be
Figure 7.10: Performance of the compositional approach (scale $\sigma^2 = 1/2$ and 200 dimensional codebook) based on the original 40-D localized feature histograms that use color information and based on a 36-D variant of these features that discards all color information.

Figure 7.11: Compositional image puzzles obtained by sampling compositions for a) grand piano and b) ferry. Given the position of the image center and a category label, compositions are sampled from the generative model. Image patches corresponding to the inferred compositions are then displayed.

The denominator can be dropped since it only depends on evidence variables ($c$ is an observed variable in this experiment). Moreover, all candidate compositions that

$$p(g_j|c, x, x_j) = \frac{P(c|g_j, S_j = x - x_j) \cdot p(g_j|S_j = x - x_j)}{P(c|x, x_j)}.$$  (7.16)
7.7. Evaluation of the Compositional Approach

Figure 7.12: Inferring compositions for a) a cougar face and b) an elephant. Given only the composition displayed in the box at the bottom left and the true category label, image patches corresponding to the inferred compositions are shown. The location of the conditioned composition is marked by a cross in the inferred image.

are established in the training images are distributed according to the composition prior, \( g_j \sim p(g_j|S_j = x - x_j) \). Compositions can therefore be sampled by evaluating the category posterior \( P(c|g_j, S_j = x - x_j) \) (which has been learned for (7.15)) on compositions \( g_j \) that have been drawn from the training data,

\[
p(g_j|c, x, x_j) \propto P(c|g_j, S_j = x - x_j) |_{g_j \text{ from training}}. \tag{7.17}
\]

The resulting compositional image puzzles in Figure 7.11 provide insights into this generative process. Here compositions have been inferred at points \( x_j \) on a regular grid (5 compositions have been drawn at each point). The sampled compositions are allowed to shift a short distance by performing gradient ascent on the likelihood (7.17) over \( x_j \) in a local neighborhood to reduce artifacts that result from sampling on a regular grid. This experiment reveals that the composition system has learned relevant compositions and their spatial relation to the object, and that it can be used as a generative model for inferring compositional representations.

7.7.5 Inferring Missing Object Components

The higher order compositions which have been introduced in Section 7.5 can be used to infer missing compositions of an object. In other words, based on only a tiny fraction of an object the remainder of the image is hallucinated from the learned
category models. Given a composition \( g_k \) the rest of an object can be inferred by drawing compositions \( g_j \) from the likelihood
\[
p(g_j|g_k, x_k, c, x_j) = \frac{P(c|g_j, g_k, r_{jk}) \cdot p(g_j|g_k, r_{jk})}{P(c|g_k, x_k, x_j)}.
\] (7.18)

Following the same line of reasoning as in Section 7.7.4 yields
\[
p(g_j|g_k, x_k, c, x_j) \propto P(c|g_j, g_k, r_{jk})|_{g_j \text{ from training}}.
\] (7.19)

In Figure 7.12, a single composition is given together with the object category label. This information is used to infer a maximum likelihood solution on the basis of compositions derived from the training set. In principle the algorithm constructs the rest of the image conditioned on the only observed composition \( g_k \) and the previously learned compositional object model.

As already done in Section 7.7.4, compositions are shifted in a local neighborhood using gradient ascent to reduce the artifacts from sampling on a regular grid. The spatial structure of objects which can be observed in the reconstructions demonstrates that the compositional model has learned characteristic relationships between compositions.

When looking at the inferred cougar face in Figure 7.12 a) it becomes obvious that the localization ability decreases with the distance from the only observed composition (the left ear). Whereas the region close to that ear is accurately inferred the localization precision at the other ear is significantly lower and the algorithm is not sure where to exactly place the other ear. The reason for this behavior is that variations in \( r_{ij} \) linearly decrease with the distance between \( g_j \) and \( g_k \).

7.7.6 Towards Learning Category-level Segmentation

Subsequently, the relevance of individual compositions for categorizing a test image is evaluated. Therefore the category posterior of the true category,
\[
P(c|g^l, x, g_j, x_j, g_k, g_l, r_{kl})|_{c=\text{True Category}}
\] (7.20)
is computed for individual pairs of compositions. In Figure 7.13 the resulting probability is then encoded in the opaqueness of the underlying image parts, i.e. alpha blending is used for visualization. As a consequence, this score computes a probabilistic segmentation of a scene based on the relevance of compositions. The visualization shows that the support region for recognizing objects (this region consists of all the
relevant compositions) covers meaningful object parts. Note that this segmentation of relevant compositions is learned from \textit{unsegmented} training images. Consequently, the system learns to segment images without any supervision information regarding segmentations in the training phase.
8

Interleaved Learning of Localization and Classification

This chapter extends the approach from Chapter 7 so that it can successfully localize multiple objects in heavily cluttered scenes by providing object bounding boxes. As in the rest of this thesis, learning of object models proceeds in an unsupervised manner for each object category. In particular, no localization information is required during training. For each category only a set of images has to be provided that feature an object of that category among other objects and clutter.

When processing complex scenes, a visual system is confronted with two challenges: it has to decide what object categories are present (classification) and where the instances of these classes are located (detection). Both tasks are strongly interdependent. For an incorrect localization hypothesis the system cannot reliably classify the object. Similarly, with an incorrect object classification, detection cannot be performed correctly since shape models are class specific.

The problem becomes even more challenging when there is no segmentation or other localization information provided during training and when there are multiple objects of different classes in the training images. This is exactly the scenario we are dealing with. So how can a vision system automatically learn a model for an object category without any information on where to find the relevant training samples in a mass of clutter? To answer this question we will extend the compositional approach from the previous chapter so that it can successfully localize and classify objects in the PASCAL Visual Object Classes Challenge 2006 (VOC’06) image database [EZWVG06]. In contrast to the Caltech-101 dataset [Cal], the VOC’06 database contains significantly more clutter, multiple objects per image, and large scale and
viewpoint variations among object instances. However, in contrast to the training protocol of VOC’06 (bounding box segmentations for objects in the training images) we continue to follow the significantly harder theme of the previous chapters and aim at learning category models without any localization information.

The key idea is to extend the approach from the previous chapter by iterating localization and compositional relevance selection. The localization task, which is equivalent to estimating the parameters of object bounding boxes, depends on which compositions are judged to be relevant. Similarly, the relevance of compositions depends on the estimated bounding box. During training this means that the generation of an object bounding box hypothesis is alternated with the training of a classifier for compositional relevance. Similarly, in the recognition phase the generation of bounding box hypotheses is alternated with the classification task of relevance estimation.

8.1 Extending the Compositional Recognition Algorithm

The PASCAL VOC’06 dataset is a 10 category database. Images in this collection feature multiple object instances of several object classes surrounded by background clutter. The recognition task does therefore differ from that of Caltech-101 where only the most prominent object class is sought for each image. The multi-object evaluation scenario of VOC’06 requires a vision system to provide bounding boxes together with a classification and a confidence in this classification. As an approximation to this problem, the following implicitly assumes that there are multiple objects in each image, but that there is maximally one instance of each category. Wherever this assumption is violated by the database object instances will be missed, leading to erroneous classifications. However, it turns out that this already provides performance that is competitive to the state of the art. By incorporating an additional model selection stage (as will be done in Chapter 9) the number of instances per category could then also be estimated.

Let us now extend the recognition algorithm from Chapter 7 to make it applicable to the recognition task defined by the VOC’06 dataset (the resulting algorithm is presented in Algorithm 8.1). Later on we will then see how the corresponding learning algorithm has to be updated. In the first stage of recognition, atomic parts are extracted out of a novel test image and composition candidates are formed. These steps are retained unchanged from Algorithm 7.1. Subsequently, the system has
to analyze whether an object of some category $c$ is actually present in the image and where it is located. Therefore, bounding box hypotheses are inferred for each category and a confidence in each hypothesis is computed—this is carried out in the main loop starting at line 7. Categories that are not present should then be given low confidence and an arbitrary bounding box. In contrast to this, object categories that are actually present should be localized by an appropriate bounding box and classified with a high confidence score.

As localization and classification are alternated, some initialization is needed to get started. Therefore, an initial bounding box estimate is computed by weighting the location of each composition with the compositional relevance $P(\chi_c|g_j)$ from (7.4). This probability does not depend on an object localization and can therefore serve as a starting point of the iteration. The bounding box is a square which is determined by its center $B^c_x \in \mathbb{R}^2$ and by its side length $2 \cdot B^c_{\sigma} \in \mathbb{R}_+$. These parameters are defined as the weighted mean and weighted standard deviation of the locations of compositions,

$$B^c_x = \frac{\sum_{j: g_j \in G} x_j \cdot P(\chi_c|g_j)}{\sum_{j: g_j \in G} P(\chi_c|g_j)},$$

$$B^c_{\sigma} = \sqrt{\frac{\sum_{j: g_j \in G} \|B^c_x - x_j\|^2 \cdot P(\chi_c|g_j)}{\sum_{j: g_j \in G} P(\chi_c|g_j)}}. \quad (8.1)$$

Given an initial estimate of $B^c$ it can be determined which compositions are relevant conditioned on the object category $c$ and the bounding box estimate $B^c$ for that category. The relevance score is computed according to (7.6) which in this case has the form $p(g_j|\chi_c, S_j = \|B^c_x - x_j\|)$. Based on the set $\mathcal{R}$ of relevant compositions a new estimate of the bounding box can be obtained. Therefore, all compositions that have been judged to be relevant, $g_j \in \mathcal{R}$, are taken into account. Their locations $x_j$ are weighted with the full compositional relevance $P(\chi_c|g_j, S_j = \|B^c_x - x_j\|)$ from Section 7.4, giving an updated estimate of $B^c$ (see lines 13 f).

This estimation of compositional relevance $\mathcal{R}$ and of the bounding box $B^c$ is then repeated iteratively. In this update scheme the computation of $\mathcal{R}$ depends on the previous estimate for $B^c$ and vice versa.

The remainder of the recognition algorithm proceeds then according to Algorithm 7.1. First, candidates for higher order compositions are established based on the estimated bounding box. In contrast to Algorithm 7.1 we now have an estimate of where the object should be located. Therefore higher order compositions $(g_k, g_l) \in \tilde{\mathcal{G}} \subset \mathcal{G} \times \mathcal{G}$ are only established of constituents that lie within the bounding box.
After that those candidates are discarded which are irrelevant for \(c\), the category under consideration.

Finally, the object localized by the bounding box is to be recognized. The underlying inference algorithm has been derived in Section 7.6.2. To make it applicable to the extended approach of this chapter only the shifts \(S_j\) have to be updated with their new formulation. As a result the following category posterior is obtained

\[
P(c|g^I, x, \{g_j, x_j\}_{g_j \in \mathbb{R}}, \{g_k, g_l, r_{kl}\}_{(g_k, g_l) \in \mathbb{R}}) \\
\propto \exp \left[ \ln P(c|g^I) + \sum_{g_j \in \mathbb{R}} \ln P\left(c|g_j, S_j = \|B_c^x - x_j\|_B\right) \right. \\
\left. + \sum_{(g_k, g_l) \in \mathbb{R}} \ln P(c|g_k, g_l, r_{kl}) \right].
\] (8.3)

Within the VOC’06 evaluation framework the object hypothesis (category \(c\), with bounding box \(B_c^x\)) is then rated with the confidence score

\[
\varphi_c := P(C = c|E).
\] (8.4)

In conclusion, the main loop of the algorithm (line 7) computes a bounding box hypothesis for each category and a confidence in the corresponding recognition/localization.

### 8.2 Learning Object Models from Cluttered, Unsegmented Images

To enable the system to learn object category models from cluttered training images without segmentation we have to find a way to automatically localize objects in the training phase. This task is challenging since there are no object models provided in this phase. The approach can be outlined as follows: Learning starts by establishing object hypotheses for all training images. These hypotheses are then all combined to train category models. Based on these models the objects can in turn be localized in the training images, thereby yielding updated object hypotheses. Consequently, this results in an alternating update scheme.

Algorithm 8.2 shows how objects are localized in the training data and how relevant compositions are learned. The algorithm accepts a set of images \(T_c^{(1)}\) that show objects of some category \(c\). Moreover, a set of irrelevant images \(T_c^{(0)}\) is required.
8.2. Learning Object Models from Cluttered, Unsegmented Images

<table>
<thead>
<tr>
<th>Algorithm 8.1: Algorithm for compositional scene analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ObjectRecognition</strong>($I$) \hspace{1cm} \triangleright I: a novel test image</td>
</tr>
<tr>
<td>1. $P \leftarrow \text{InterestPoints}(I)$ \hspace{1cm} \triangleright extract atomic parts and form composition candidates:</td>
</tr>
<tr>
<td>2. for all scales $\sigma \in S$</td>
</tr>
<tr>
<td>3. do $E^{(\sigma)} \leftarrow \text{AtomicParts}(I, P, \sigma)$</td>
</tr>
<tr>
<td>4. $G^{(\sigma)} \leftarrow \text{CompositionCandidates}(E^{(\sigma)})$</td>
</tr>
<tr>
<td>5. $G \leftarrow \bigcup_{\sigma \in S} G^{(\sigma)}$ \hspace{1cm} \triangleright context descriptor from (4.57)</td>
</tr>
<tr>
<td>6. $g' \leftarrow \frac{1}{</td>
</tr>
<tr>
<td>7. for all $c \in L$</td>
</tr>
<tr>
<td>8. do \hspace{1cm} \triangleright obtain initial estimate of bounding box $B^c$:</td>
</tr>
<tr>
<td>9. $B^c_x \leftarrow \frac{\sum_{j: g_j \in R} x_j^c P(\chi_c</td>
</tr>
<tr>
<td>10. $B^c_\sigma \leftarrow \sqrt{\frac{\sum_{j: g_j \in R}</td>
</tr>
<tr>
<td>11. update $R$ and $B^c$ alternatingly for category $c$:</td>
</tr>
<tr>
<td>12. repeat \hspace{1cm} \triangleright select relevant compositions for given $B^c$:</td>
</tr>
<tr>
<td>13. $R \leftarrow \text{RelevantCompositions}(G, B^c, c)$</td>
</tr>
<tr>
<td>14. update $B^c$ for given $R$:</td>
</tr>
<tr>
<td>15. $B^c_x \leftarrow \frac{\sum_{j: g_j \in R} x_j^c P(\chi_c</td>
</tr>
<tr>
<td>16. $B^c_\sigma \leftarrow \sqrt{\frac{\sum_{j: g_j \in R}</td>
</tr>
<tr>
<td>17. until convergence</td>
</tr>
<tr>
<td>18. draw compositions of compositions for category $c$:</td>
</tr>
<tr>
<td>19. $\tilde{G} \leftarrow \text{Composition^2Candidates}(G, B^c)$</td>
</tr>
<tr>
<td>20. $\tilde{R} \leftarrow \text{RelevantCompositions^2}(\tilde{G}, c)$</td>
</tr>
<tr>
<td>21. \hspace{1cm} \triangleright compute confidence in category hypothesis $c$ for object located within bounding box $B^c$:</td>
</tr>
<tr>
<td>22. $P(c</td>
</tr>
<tr>
<td>23. $\varphi_c \leftarrow P(C = c</td>
</tr>
<tr>
<td>24. return ${\varphi_c, B^c}_{c \in L}$ \hspace{1cm} \triangleright object category posterior and location</td>
</tr>
</tbody>
</table>

This set is built by randomly selecting samples from images that show categories other than $c$. The learning algorithm starts by extracting candidate compositions
out of all the training images. In line 5 the distribution \( P(\chi_c|g_j) \) is learned using NKDA based on all established compositions \( g_j \). This estimation is based on a two-class classification task—distinguishing compositions from category \( c \) from the others. Now for each image in \( T_c^{(1)} \) an initial estimate of the object location is computed (for-loop starting in line 6). Estimating these bounding boxes \( B^c \) works as described in Section 8.1 and it uses \( P(\chi_c|g_j) \) to weight the individual compositions.

Given these initial bounding box estimates for each image an alternating update of bounding boxes and of the relevance weights of compositions (which are part of the object models) starts. First, the relevance of compositions, \( P(\chi_c|g_j, S_j = \|B^c_x - x_j\|) \), is estimated using NKDA based on the current bounding box estimates (line 12). Thereafter, the bounding boxes for each training image in \( T_c^{(1)} \) are updated based on these relevance weights.

The bounding boxes that result from this alternating update are then used to select the relevant compositions out of all compositions from category \( c \). For that purpose, Algorithm 7.2 is employed. This time the relevance learning algorithm is however provided with the previously computed bounding boxes for all images in \( T_c^{(1)} \). It can therefore discard all compositions that lie outside of a bounding box—the ones which are judged to be clutter. This function returns the distribution \( p(\chi_c|g_j, s_j) \) and a set \( R \) of relevant compositions for category \( c \).

Based on the set of relevant compositions and the estimated bounding boxes, the remaining distributions in (8.3) can be learned using multi-class NKDA as has been previously done in Chapter 7. At this point, all the distributions that define object category models have been learned and training is finished.

### 8.3 Evaluation

Subsequently, the system is evaluated on the challenging PASCAL VOC’06 database [EZWVG06]. This ten category dataset contains multiple objects per image; object instances feature large variations in scale and viewpoint. Moreover, there is a lot of background clutter. The database consists of 5,304 images (2,618 training images and 2,686 test images) with a total of 9,507 labeled objects. Thus, each image shows at least one object, but there can also be more than one object per class as well as multiple classes. The standard training protocol of VOC’06 stipulates that bounding box segmentations together with the category label of the circumscribed object are used for training. In contrast to that we continue to follow the significantly harder
8.3. Evaluation

Learning Detection and Relevance ($T_c^{(1)}, T_c^{(0)}$)

$\triangleright$ $T_c^{(1)}$: set of training images for category $c$,
$\triangleright$ $T_c^{(0)}$: irrelevant images of categories other than $c$
$\triangleright$ collect compositions for all images in each training subset:

1. for $i \in \{0, 1\}$
2. do $\mathcal{G}_i = \emptyset$
3. for $I \in T_c^{(i)}$
4. do $\mathcal{G}_i \leftarrow \mathcal{G}_i \cup \text{ALLCOMPOSITIONCANDIDATES}(I)$

$\triangleright$ Lines 1 – 5 of Alg. OBJECTRECOGNITION

5. $P(\chi|g_j) \leftarrow \text{LEARNPROBCLASSIFIER}(\mathcal{G}_0, \mathcal{G}_1)$
6. for $I \in T_c^{(1)}$
7. do $\triangleright$ obtain initial estimate of bounding box $\mathcal{B}^c$:
8. $\mathcal{B}^c_x(I) \leftarrow \frac{\sum_{j : g_j \in \mathcal{G}} x_j \cdot P(\chi|g_j)}{\sum_{j : g_j \in \mathcal{G}} P(\chi|g_j)}$
9. $\mathcal{B}^c_\sigma(I) \leftarrow \sqrt{\frac{\sum_{j : g_j \in \mathcal{G}} \|x_j - \bar{x}\|^2 \cdot P(\chi|g_j)}{\sum_{j : g_j \in \mathcal{G}} P(\chi|g_j)}}$

$\triangleright$ alternating update of relevance weights and bounding box:
10. for $h = 1$ to 3
11. do
12. $P(\chi|g_j, S_j = \frac{\|x_j - \bar{x}\|}{\|\bar{g}_j\|}) \leftarrow \text{LEARNPROBCLASSIFIER}(\mathcal{G}_0, \mathcal{G}_1, \mathcal{B}^c)$
13. for $I \in T_c^{(1)}$
14. do
15. $\mathcal{B}^c_x(I) \leftarrow \frac{\sum_{j : g_j \in \mathcal{G}} x_j \cdot P(\chi|g_j, S_j = \frac{\|x_j - \bar{x}\|}{\|\bar{g}_j\|})}{\sum_{j : g_j \in \mathcal{G}} P(\chi|g_j, S_j = \frac{\|x_j - \bar{x}\|}{\|\bar{g}_j\|})}$
16. $\mathcal{B}^c_\sigma(I) \leftarrow \sqrt{\frac{\sum_{j : g_j \in \mathcal{G}} \|x_j - \bar{x}\|^2 \cdot P(\chi|g_j, S_j = \frac{\|x_j - \bar{x}\|}{\|\bar{g}_j\|})}{\sum_{j : g_j \in \mathcal{G}} P(\chi|g_j, S_j = \frac{\|x_j - \bar{x}\|}{\|\bar{g}_j\|})}}$

$\triangleright$ selection of relevant compositions using Alg. 7.2:
17. $(p(\chi|g_j, s_j), \mathcal{R}) \leftarrow \text{RELEVANCELEARNING}(T_c^{(1)}, T_c^{(0)}, \{\mathcal{B}^c(I)\}_{I \in T_c^{(1)}})$
18. return $p(\chi|g_j, s_j), \mathcal{R}, \{\mathcal{B}^c(I)\}_{I \in T_c^{(1)}}$

Algorithm 8.2: Algorithm for learning relevant compositions.

theme of the previous chapters and learn category models without any localization information. That is, for the training images only category labels are provided, but no bounding boxes are used. The learning algorithm described in Section 8.2 does then automatically localize relevant objects in the training images.
The *PASCAL Visual Object Classes Challenge 2006* defines two tasks, a classification and detection problem.

### 8.3.1 Classification Performance

The first task of VOC’06 is a classification problem. For each of the ten classes it has to be decided whether at least one object of that class is present in a test image or not. The real-valued confidence scores are then used to rank the individual predictions so that a receiver operator characteristic (ROC curve) [Faw06] can be drawn.

Let us briefly review *signal detection theory* to explain the concept of ROC curves. A two-class classification problem (distinguishing positive from negative samples with labels $p$ and $n$, respectively) gives rise to the $2 \times 2$ confusion matrix or contingency table displayed in Table 8.1.

<table>
<thead>
<tr>
<th>predicted class</th>
<th>$p$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>true positive (TP)</td>
<td>false negative (FN)</td>
</tr>
<tr>
<td>$n$</td>
<td>false positive (FP)</td>
<td>true negative (TN)</td>
</tr>
</tbody>
</table>

Table 8.1: Confusion matrix of a two-class classification problem.

The correct classifications appear on the diagonal of this table whereas the off-diagonal entries show the two types of errors. Let us use the abbreviations $TP$ for true positives, $FN$ for false negatives, and so on. Now two characteristics can be derived, the false positive rate and the true positive rate,

\[
FPR := \frac{FP}{FP + TN}, \quad TPR := \frac{TP}{TP + FN}.
\] (8.5)

The true positive rate is also called sensitivity or hit rate. To generate a ROC curve the discrimination threshold of the classifier is varied so that the false positive error rate is traded against the true positive rate. The ROC curve is a graphical plot that shows how $TPR$ and $FPR$ vary while the free parameter is changed. Ideally a classification system would yield a maximal $TPR = 1$ for all $FPR$’s. Typically, however, the $TPR$ becomes smaller as $FPR$ is decreased. The same curve can be obtained by plotting sensitivity (= $TPR$) against $(1 - SPC)$, where $SPC$ denotes specificity,

\[
SPC := \frac{TN}{TN + FP}.
\] (8.6)
Figure 8.1: Classification performance. (a) ROC curves for all categories of the PASCAL database, computed according to the guidelines of the PASCAL competition. The legend provides the categories names and the corresponding area under ROC curve (AUC). (b) The bars show the AUCs for all ten categories. The lines indicate the average performance over the 20 submission of the PASCAL 2006 challenge. These methods include highly supervised approaches trained with bounding box information.

To simplify a comparison of different ROC curves it is common practice to compute a single score that characterizes the curve—the area under curve (AUC). This measure is merely the area underneath the ROC curve.

Results

Figure 8.1(a) shows the ROC curves for all ten categories—the compositional approach is trained on all the designated training images and tested on the test set. The corresponding AUCs are in the range of 0.81 to 0.96 except for one outlier, category person with an AUC of 0.66. This class is particularly complicated for a number of reasons: it features panoramic pictures that show full persons as well as closeup views that show only parts of a person. Moreover, this class is articulated and appearance exhibits significant variations due to different clothing. Altogether, the model is competitive compared with the average of all submissions to the PASCAL 2006 challenge which is displayed in Figure 8.1(b). Since the compositional approach
is not specifically tailored to any of these classes but is inherently a multi-class model and does—unlike most of the PASCAL competitors—not use location information in training, achieving comparable performance to these models is remarkable. The largest performance gap can be observed for category person for which a lot of specifically designed approaches exist. The manual tuning towards this category and the rich supervision information used by these methods explain their advantage.

The PASCAL challenge has only specified a single-class classification task. However, unlike models that are specifically designed or adapted for detecting and recognizing a specific object class, the compositional approach learns a single multi-class model. Hence, we cannot only predict the probability of a class present or absent in a test image. We can also specify which is the most likely of the ten classes in the image. This task is significantly harder as it involves a decision between ten potential outcomes instead of only two. Let us now quantify this multi-class performance by computing the confusion table for test images that contain only objects of one category—however, several instances of a single category can still be present. The confusion table is presented in Figure 8.2 and it shows an overall retrieval rate of 46.3%. The best performance is achieved for man-made objects, whereas retrieval rates for the animal and person categories are generally lower. This bifurcation is mainly due to the fact that living creatures are highly articulated (that is also why sheep with their less dominant articulation perform best among all animals). Furthermore, it is interesting to study the confusions between pairs of categories. The dominant off-diagonal terms are confusions between categories cat and dog (roughly 23%) and cows mistaken for sheep (22% of all cows). This implies that confusions typically occur between visually (and conceptually) similar categories.
8.3. Evaluation

8.3.2 Detection Performance

The second task of VOC’06 is object detection, that is the localization accuracy of the system is evaluated. Therefore, the bounding boxes of objects are to be predicted in the test images. For each bounding box the system has to provide a confidence score that is used to weight this hypothesis so that a precision recall curve (PR curve) can be drawn.

In information retrieval, precision denotes the proportion of retrieved samples (e.g. documents) that are relevant. In this context relevant means belonging to the class of positives $p$. In our two-class setting this translates to

$$P_{REC} := \frac{TP}{TP + FP}.$$  \hspace{1cm} (8.7)

The other characteristic is the recall rate

$$RC := \frac{TP}{TP + FN}.$$  \hspace{1cm} (8.8)

In information retrieval this proportion indicates how many relevant samples are retrieved out of all available ones. A recall of one can be trivially achieved by classifying all samples as positives. This shows that recall is traded against precision which measures how many of the retrieved samples are actually relevant.

The predicted bounding box hypotheses are tagged with a confidence score that is to be used in the following. Sorting the hypotheses according to this score and selecting the best $m$ ones yields a precision and recall rate for these top $m$ hypotheses. Varying $m$ over its whole range gives then a PR curve. A principle measure for comparing different PR curves is the average precision (AP). In VOC’06 the precision is interpolated at 11 thresholds on recall $REC = 0, 1, \ldots, 1$ and the arithmetic mean is computed.

Finally, it has to be defined when a bounding box hypothesis is correct and when it is wrong. In VOC’06 the following criterion is used. For a correct detection, the area $A(\bullet)$ of overlap between a predicted bounding box $B^c$ and a ground truth bounding box $B^{gt}$ must be more than half the union of both areas,

$$\text{bounding box hypothesis } B^c \text{ correct } \iff \frac{A(B^c \cap B^{gt})}{A(B^c \cup B^{gt})} > \frac{1}{2}.$$  \hspace{1cm} (8.9)

Moreover, multiple detection of the same object are considered as false detections.

Results

Figure 8.3(a) shows the precision recall curves for the ten categories. In Figure 8.3(b) a comparison with all PASCAL entries is presented. Although these methods
Figure 8.3: Detection performance. (a) Precision recall curves for detection as defined in the PASCAL challenge. The legend provides next to each category label the average precision of that category. (b) The bars show the average precision (AP) for each category. The colored markers denote the performance of entries in the PASCAL competition which are mostly specifically designed for only a few categories and trained on heavily supervised data.

Use heavy supervision information and are mostly tailored to just a few categories the compositional approach outperforms all of them for category *bus* and achieves a performance that is well within the range of competition entries for categories *dog*, *bicycle*, and *horse*.

Moreover, we have made some simplifying assumptions that limit the performance of the compositional approach. The current implementation assumes that although multiple objects might be present in an image, there is only one instance per class. Thus, the maximally achievable true positive rate is limited from the outset, which explains why no recall beyond .5 could be reached. As average precision is computed by averaging at 11 equally spaced points on the recall axis on the full interval [0, 1], more than half of these values are zero and the AP is dragged down. Another limitation is that bounding boxes are represented by only two parameters, their center and side length. Consequently, boxes are assumed to be squares, which causes errors for elongated objects. It remains for future work to resolve these limitations and we expect significant performance gain from such an extension. Anyway, it is remarkable that a model trained without any localization information on cluttered training data is nevertheless competitive to state of the art detection algorithms that have been trained in a supervised manner.
8.3. Evaluation

An example that depicts object localization is presented in Figure 8.4. The blue bounding box on the left is the ground truth whereas the green one in Figure 8.4(b) is the predicted box. The probability map for category dog that is shown in the right figure visualizes $P(\chi_c=\text{dog} | g_j, S_j = \frac{\|B_c - x_j\|}{B_0})$. For all compositions $g_j$ that have been established in the image, this probability is computed. The image region that is covered by a composition is then filled with its probability. Where multiple compositions are overlapping their individual probabilities are averaged. The probability map in Figure 8.4(b) shows that compositions on the object have a high confidence for dog and faithfully cover the animal. Already this example shows that the choice of a bounding box localization may under-represent the model’s true capabilities, which provides object localization on the significantly finer granularity level of compositions.

8.3.3 Analyzing the Performance of the Compositional Model

In Section 7.7.1 it has been shown that the compositional approach significantly outperforms a baseline bag representation. Let us now further investigate the performance gain achieved by compositionality over the baseline model. For that purpose the test scenario is restricted to only the classification task: the correct object bounding box is given and only the category label of the object is queried. Now we can compare the performance of the full compositional model from Section 8.1
Chapter 8. Interleaved Learning of Localization and Classification

Figure 8.5: Comparing the performance of a bag representation $g^I$, with that of a simple compositional approach without spatial information (8.10), and the full compositional model from Section 8.1.

with that of the baseline model. Recognition in the case of the bag representation $g^I$ is based on maximizing $P(c|g^I)$. As an additional, intermediate experiment we neglect the spatial information in the compositional model so that the gain achieved by localization information can be measured separately. Therefore, recognition is based on maximizing

$$P(c|g^I, \{g_j\}_{g_j \in R}) \propto \exp \left[ \sum_{g_j \in R} \ln P(c|g_j) \right].$$

(8.10)

Figure 8.5 compares the retrieval rates achieved by the three approaches. Since the evaluation scenario is restricted to a classification problem of limited complexity the performance differences of the three approaches are also reduced. Nevertheless, this evaluation does not only show that the full compositional model outperforms the bag representation. It also underlines that even the simpler compositional representation from (8.10) without any spatial information performs significantly better than the bag model. In conclusion this investigation has corroborated the hypothesis that compositions play a crucial part in building powerful computer vision systems.
Recognition, Segmentation, and Tracking of Multiple Objects in Video

Combined tracking, segmentation, and recognition of objects in videos is one of the long standing challenges of computer vision. When approaching real world scenarios with large intra-category variations, with minimal supervision during training, and with real-time constraints during prediction, this problem becomes particularly difficult. By establishing a compositional representation, the complexity of object models can be reduced significantly and learning such models from limited training data becomes feasible. However, a structured representation might entail disadvantages during recognition, especially given the high computational demands of video. This chapter presents a compositional approach to video analysis that performs near real-time and demonstrates how the key concept of compositionality can actually be exploited for both, rendering learning tractable and making recognition computationally feasible.

The compositional video analysis system unites category-level, multi-class object recognition, segmentation, and tracking in the same probabilistic graphical model. Moreover, this Bayesian network combines compositions together with object shape. Learning object models requires only a category label for the most prominent object in a complete video sequence, thereby even tolerating distracting clutter and other objects in the background. Category specific compositions of local features are automatically learned so that irrelevant image regions can be identified and discarded without supervision. As a result tedious hand-segmentations, object localizations,
or initializations of a tracker become superfluous. There has been only very little work on category-level segmentation and recognition in video. Therefore, a video categorization database has been established in this project for evaluation purposes (cf. [OB07a]). It consists of four object categories (bicycle, car, pedestrian, and streetcar). Videos have been recorded in their natural outdoor environment with a hand-held camera and show significant scale variation, large intra-category variability, camera panning, and background clutter.

As before in the setting of still images, the goal is to learn category-specific compositions of generic parts which, in comparison to the whole object, show small intra-category variations. The robustness of compositions to image changes can be exploited for tracking and grouping them over consecutive video frames. Combining the compositional representation of several frames does then enhance object segmentation and recognition. To be able to simultaneously recognize multiple objects in a video, a model selection strategy is incorporated that automatically estimates the correct model complexity based on a stability analysis.

9.1 Crucial Components of Video Retrieval Systems

Category-level recognition, segmentation, and tracking of objects in videos is related to a number of subtasks.

Optical Flow: First, motion information can be exploited by selecting relevant features for tracking (e.g. [ST94]) and establishing correspondences between frames, e.g. using the method of Lucas and Kanade [LK81]. The pixel-level correspondences yield an estimate of optical flow, which refers to the motion of individual pixels. However, we cannot expect to obtain correct estimates for optical flow at all individual pixels in an image due to the so called aperture problem [Hor86]: The motion of image contours such as discontinuity edges is ambiguous because the motion component parallel to the contour cannot be estimated from the visual input. Only at corners this ambiguity is resolved, whereas homogeneous areas do not provide any flow information. Therefore, Lucas and Kanade have proposed an approach that tracks extended regions and it can been extended to multiple scales as shown in [Bou]. This approach exploits that extended areas of an image show edges of multiple orientations so that at least a sparse set of such regions can be tracked reliably from one frame into the next. In contrast to this, Horn and Schunck [HS81] introduce
9.1. Crucial Components of Video Retrieval Systems

a global smoothness constraint to deal with the aperture problem on a pixel level. The constraint penalizes large gradients in optical flow. To get started the algorithm establishes ambiguous flow information by computing image derivatives in space and time. Using the global smoothness assumption, these discontinuities are then propagated over the image by means of the Gauss-Seidel method (e.g. [BBC+94]) for solving linear equations. As a consequence, missing flow information is filled in and the initial flow estimates are disambiguated.

**Representation Schemes:** Second, an object representation has to be found as discussed in Section 2.2. Typical descriptors that are used are SIFT features [Low04], flow histograms [DTS06], and template-based appearance patches (e.g. [AAR04, LS04]). Combining local features in an object model can then proceed along several lines. A simple approach is to compute descriptors on a regular grid and concatenate all cells to obtain a joint model [DTS06]. More complex representations of the spatial object structure are constellation models [FPZ03], hough voting strategies [LS04], many-to-many feature correspondences [DSK+06], image parsing graphs [TCYZ05], feature correspondences with minimal overall shape deformation [BBM05], and compositional models [OB06]. The problem of most representation schemes that have been developed for still image recognition is that they are computationally too costly to be applied to video. Therefore, Viola and Jones [VJ01] have employed a cascade of classifiers to build a real-time recognition system for faces. The idea is that many object hypotheses can be rejected reliably even when only few observations are made and simple classifiers are used. Early stages in the cascade do therefore limit the amount of possible hypotheses significantly before later stages decide between the few remaining difficult cases.

Besides faces, another object class that many vision systems have been specifically developed for are pedestrians, e.g. [VJS03, GGM04]. Multi-class, category-level recognition as pursued in this thesis does, however, pose additional challenges because the system cannot be tailored specifically towards a single object category.

**Object Tracking:** Optical flow is only an indication for object motion. However, object motion cannot be inferred directly from the motion of individual pixels. A counter example is for instance a shadow that a moving object casts on another still object. In this case optical flow is perceived on the still object due to the moving shadow although the object itself is not moving. Consequently, an object tracking algorithm, which estimates the motion of complete objects, needs to integrate over the motion of all object pixels. Thereby, corrupted or otherwise misleading flow
information has to be explained away or must at least be suppressed so that object instances in different frames can be brought into correspondence. A purely flow-based tracking algorithm has been studied in [BC06]. Moreover, Sivic et al. have combined the tracking algorithm with a query-by-example approach to recognition that searches for regions which are similar to a user selected one. Another method for tracking user specified regions is the approach by Comaniciu et al. [CRM03] which is based on mean-shift clustering and also the ensemble approach of [Avi05]. Finally, Goldberger and Greenspan [GG06] propose a method for using segmentations of previous video frames to obtain a segmentation for the next.

9.2 Outline of the Video Composition System

The following briefly sketches the architecture of the video composition system for object recognition, segmentation, and tracking (depicted in Figure 9.1) before giving a detailed account of the individual building blocks in later sections. A novel video sequence is analyzed sequentially in a frame-by-frame manner, while the underlying statistical model is propagating information over consecutive frames. For
9.3. Estimation of Optical Flow

Each new frame, optical flow is estimated at interest points using the Lucas-Kanade method [LK81]. These points and their motion pattern constitute the atomic parts of the composition system. Since interest points and their optical flow cannot be computed reliably, tracking individual points through a whole image sequence becomes error-prone. As a remedy for this problem compositions of atomic parts are established. These are represented by probability distributions over their constituent parts. Thereby invariance with respect to individual missing parts is achieved and compositions can be tracked reliably through a video by considering the optical flow distribution of all their constituents. The correspondence between compositions of consecutive frames is exploited to group composition over time. Non-discriminative compositions are then discarded by means of a Bayesian relevance detection approach.

Subsequently, salient objects are detected and segregated from each other by establishing multiple segmentations. Finding the most appropriate segmentation is then mastered by a model selection strategy that analyzes the stability of the proposed segmentations over the preceding frames. The model with highest stability is selected and combined with models from previous video frames to segment the current frame. Recognition of objects in the individual segments is then based on an extension of the compositional shape model from Section 4.5.4 which couples all compositions belonging to the same segment in a Bayesian network. In conclusion, tracking object constituents, segmenting objects from another, and recognizing the object category are all captured in the same statistical framework, namely the graphical model illustrated in Figure 9.4. In this model, object representations of consecutive frames are linked together by a Markov backbone that connects segmentations of subsequent frames. Learning the underlying structured object models is carried out without requiring hand-segmentations or localization of objects in training videos. Only a category label of the most prominent object in a sequence is required, regardless of other objects in the scene.

9.3 Estimation of Optical Flow

Due to its favorable processing speed we will use a multi-scale version [Bou] of the tracking algorithm proposed by Lucas and Kanade [LK81]. Let us now briefly review this approach. Given a sequence of gray-scale images \( \hat{I}^t, t = 0, 1, 2, \ldots \), the goal of the tracking algorithm is to estimate the optical flow \( d^t_x \in \mathbb{R}^2 \) at image locations \( x \) in frame \( t \). The flow vector field \( d^t \) brings successive frames into correspondence so
that \( \hat{I}^{t+1}(x) = \hat{I}(x + d^t_x) \). More precisely, we seek \( d^t \) that shifts the current frame so that it can be perfectly registered against the next frame. Therefore, a registration error \( E(d^t, x) \) is computed for a local region \( R(x) \) around \( x \),

\[
E(d^t, x) = \sum_{x' \in R(x)} \left( \hat{I}^{t+1}(x') - \hat{I}(x' + d^t_x) \right)^2.
\]  

(9.1)

Then a flow field is sought that minimizes this error for all locations \( x \) where reliable flow estimates are available. The overall registration error is therefore given by weighting each location with its reliability \( \varphi(x) \),

\[
\tilde{E}(d^t) = \sum_x \varphi(x) \sum_{x' \in R(x)} \left( \hat{I}^{t+1}(x') - \hat{I}(x' + d^t_x) \right)^2.
\]  

(9.2)

Due to the aperture problem only a subset of all \( x \) will provide reliable information. Moreover, the underlying assumption is that optical flow is approximately constant in small regions \( R \) and that the regions do not change otherwise between frames. Wherever this assumption is violated (e.g. illumination changes or complex motion patterns with local singularities) additional errors are introduced. Now a trade-off becomes obvious. On the one hand the integration window \( R \) should be small to allow for accurate flow estimates and to avoid an averaging over discontinuities in the flow field (for instance at object boundaries). On the other hand a large region is required to be able to estimate large motion vectors reliably. Because of this trade-off it is favorable to follow a multi-scale approach that starts by roughly estimating large flows on a coarse scale and iteratively updates these estimates on finer scales later on. In [Bou] an pyramidal implementation of the Lucas-Kanade algorithm is described that efficiently computes flow on a pyramid of rescaled versions of images.

The Lucas-Kanade tracking algorithm aims at minimizing \( E(d^t, x) \) for each location \( x \). Now a linear Taylor expansion is used to approximate how the image is changed by optical flow,

\[
\hat{I}(x' + d^t_x) \approx \hat{I}(x') + \left( \frac{\partial \hat{I}(x')}{\partial x}, \frac{\partial \hat{I}(x')}{\partial y} \right) \cdot d^t_x
\]  

(9.3)

\[
= \hat{I}(x') + \left( \nabla \hat{I}(x') \right)^\top d^t_x.
\]  

(9.4)

Substitution of this Taylor approximation into (9.1) yields

\[
E(d^t, x) \approx \sum_{x' \in R(x)} \left( \hat{I}^{t+1}(x') - \hat{I}(x') - \left( \nabla \hat{I}(x') \right)^\top d^t_x \right)^2.
\]  

(9.5)
The minimizer of (9.5) can be found by differentiating $E$ at $x$ w.r.t. the unknown flow $d^t_x$ and setting the result to zero,

$$0 = \frac{\partial E(d^t, x)}{\partial d^t_x} = -2 \sum_{x^t \in R(x)} \left( \hat{I}^{t+1}(x^t) - \hat{I}^t(x^t) - \left( \nabla \hat{I}^t(x^t) \right)^\top d^t_x \right) \left( \nabla \hat{I}^t(x^t) \right)^\top .$$

Since $\frac{\partial \hat{I}^t(x^t)}{\partial t} \approx \hat{I}^{t+1}(x^t) - \hat{I}^t(x^t)$ above equation can be written as

$$0 = \sum_{x^t \in R(x)} \left( \nabla \hat{I}^t(x^t) \right)^\top d^t_x \left( \nabla \hat{I}^t(x^t) \right)^\top = \sum_{x^t \in R(x)} \frac{\partial \hat{I}^t(x^t)}{\partial t} \left( \nabla \hat{I}^t(x^t) \right)^\top$$

$$\Leftrightarrow (d^t_x)^\top = \left[ \sum_{x^t \in R(x)} \left( \nabla \hat{I}^t(x^t) \right) \left( \nabla \hat{I}^t(x^t) \right)^\top \right]^{-1} \times \sum_{x^t \in R(x)} \frac{\partial \hat{I}^t(x^t)}{\partial t} \left( \nabla \hat{I}^t(x^t) \right)^\top$$

$$\Leftrightarrow d^t_x = \left[ \sum_{x^t \in R(x)} \left( \frac{\partial^2 \hat{I}^t(x^t)}{\partial x^2} \frac{\partial \hat{I}^t(x^t)}{\partial x^t} \frac{\partial \hat{I}^t(x^t)}{\partial y^t} \frac{\partial^2 \hat{I}^t(x^t)}{\partial y^2} \right) \right]^{-1} \times \sum_{x^t \in R(x)} \frac{\partial \hat{I}^t(x^t)}{\partial t} \frac{\partial \hat{I}^t(x^t)}{\partial x^t} \frac{\partial \hat{I}^t(x^t)}{\partial y^t} .$$

Therefore, the optical flow is determined by a linear equation system with the solution $d^t_x = A^{-1}b$. Obviously the solution can only be computed if the $2 \times 2$ matrix $A$ is well-conditioned and above the image noise level [ST94]. Shi and Tomasi have therefore suggested to compute the flow only at those locations $x$ for which both eigenvalues $\lambda_1$ and $\lambda_2$ of $A$ are above a predefined threshold $\lambda$,

$$\min(\lambda_1, \lambda_2) > \lambda .$$

At these interest points the optical flow can be computed by iteratively minimizing (9.11) using the Newton-Raphson algorithm [LK81].

9.4 “Atoms” of the Compositional Representation

In each video frame optical flow is estimated at reliable Shi-Tomasi interest points (cf. (9.12)) using the Lucas-Kanade tracking algorithm presented in Section 9.3. This registration of points in consecutive frames yields an estimate of the optical flow $d^t_i$ at interest point $i$ in frame $t$, i.e. the displacement vector, see Figure 9.2 b). The interest points constitute the atomic parts of the composition system.
9.4.1 Codebook-Based Representation of Atomic Parts

Compositions are represented by a distribution over a codebook of atomic parts to support varying numbers of constituents. Let $e^t_i$ denote a feature vector that represents an atomic part $i$ in frame $t$. By performing a $k$-means clustering on all feature vectors $e^t_i$ detected in the training data a common codebook of atomic parts for all object categories is obtained (see Section 3.3). To robustify the representation each feature is again described by the Gibbs distribution (3.7) over the codebook: Let $d_{\nu}(e^t_i)$ denote the squared Euclidean distance of a measured feature $e^t_i$ to a centroid $a_\nu$. The local descriptor is then represented by the following distribution of its cluster assignment random variable $F_i$,

$$P(F_i = \nu | e^t_i) := \frac{\exp(-d_{\nu}(e^t_i))}{\sum_{\nu} \exp(-d_{\nu}(e^t_i))}.$$  \hspace{1cm} (9.13)

9.4.2 Descriptors for Local Parts

To establish a representation of the atomic parts, two different types of local features are proposed. The first type simply represents the optical flow at an interest point, whereas the second is based on localized feature histograms Section 3.2.2 of a small surrounding region. Tracking of compositions is based on the optical flow of their constituents: For each interest point $i$ in frame $t$ we use its optical flow $d^t_i$, giving a 2-dimensional feature vector $e^t_i = d^t_i$. For the second local descriptor, quadratic patches with a side length of 20 pixels are extracted at each interest point. Each of these patches is then described by the localized feature histograms from (3.5). Therefore the patches are divided up into four equally sized subpatches with locations fixed relative to the patch center. In each of these subwindows marginal histograms over
edge orientation and edge strength are computed (allocating four bins to each of them). Furthermore, an eight bin color histogram over all subpatches is extracted. All these histograms are combined in a common feature vector $e_t$.

For both types of features a separate codebook is established (optical flow features are quantized with a 10-dimensional codebook, the localized feature histograms are represented by a 60-dimensional codebook). Tracking of compositions and object segmentation is then based on the optical flow alone. Only the final inference of the object category based on the compositions in the detected foreground uses the second descriptor type.

### 9.5 Establishing Dynamic Compositions

In the first frame of a video ($t = 0$), a random subset of all detected interest points is selected. Each of these points is then grouped with the atomic parts in its local neighborhood (radius of 25 pixel) yielding compositions of atomic parts. A composition in frame $t$ is then represented as a mixture over the distributions (9.13) of its parts (cf. (4.2)). Let $\Gamma_j = \{e_1^t, \ldots, e_m^t\}$ denote the grouping of parts represented by features $e_1^t, \ldots, e_m^t$. The composition is then represented by the multivariate random variable $G_j$ which is a bag-of-parts, i.e. its value $g_j^t$ is a multivariate probability distribution over the $k$-dimensional codebook of atomic parts

$$g_j^t \propto \sum_{i=1}^{m} P(F_i = 1|e_i^t), \ldots, P(F_i = k|e_i^t)$$

(9.14)

Finally, each of the $k$ dimensions is independently standardized to zero mean and unit variance across the whole training set, giving $z$-scores. This mixture model has the favorable property of robustness with respect to variations in the individual parts. The two types of features yield two different representations of the $j$-th composition: $g_j^t$ is the representation based on localized feature histograms, whereas $\tilde{g}_j^t$ builds on optical flow.

Compositions are tracked throughout a video based on the average flow estimated at their constituent parts. Given the position $x_j^t$ of a composition in frame $t$ and the optical flow vectors of its parts $d_i^t$, its predicted position in the next frame is

$$x_j^{t+1} := x_j^t + \frac{1}{m} \sum_{i=1}^{m} d_i^t.$$ 

(9.15)

Compositions are dynamic, i.e. they move with objects and assignments of parts to
compositions are updated as interest points are recomputed: In frame $t+1$ all parts $e_{t+1}^i$ in the local neighborhood of $g_{t+1}^j$ are assigned to this composition.

### 9.6 Grouping Compositions over Time

In contrast to the preceding spatial grouping the following presents a grouping of compositions over time. This grouping aims at establishing compositions that are robust with respect to measurement errors in individual frames (e.g. incorrect flow estimates). A temporal grouping of the $j$-th composition over consecutive frames yields the composition $h_j^t$ represented by the distribution

$$h_j^t \propto \begin{cases} \eta g_j^t + (1 - \eta)h_{j-1}^t, & \text{if } t > 1, \\ g_j^t, & \text{else.} \end{cases}$$

(9.16)

The optical flow representation of compositions is computed according to the same recursion formula, i.e. $\tilde{h}_j^t \propto \eta \tilde{g}_j^t + (1 - \eta)\tilde{h}_j^{t-1}$, and the mixing factor is chosen to be $\eta = 1/2$. Thus, the influence of previous $h_j^{t-\Delta t}$ and $\tilde{h}_j^{t-\Delta t}$ on a current temporal composition decays exponentially with the lag $\Delta t$. The corresponding transition probability of the graphical model in Figure 9.4 is defined as

$$p(H_j^t = \xi | g_j^t, h_j^{t-1}) := 1_{\{\xi = \eta g_j^t + (1 - \eta)h_j^{t-1}\}}.$$  

(9.17)

In other words, this posterior is one iff the representation $h_j^t$ of a composition is according to (9.16).

Figure 9.2 c) shows the centers $x_j^t$ of compositions $\tilde{h}_j^t$ which have been established for the video frame shown in Figure 9.2 a). Compositions are represented as
Figure 9.4: Graphical model that unites category-level object recognition, segmentation, and tracking in the same statistical framework. Shaded nodes denote evidence. The graph shows the dependencies between the three processes for frame $t$ as well as the connection with the preceding frame. The involved random variables are the following: compositions described by localized feature histograms, $G_j$, and by optical flow, $\tilde{G}_j^t$; temporal groupings of compositions: $H_j^t$ and $\tilde{H}_j^t$; location of $j$-th composition: $X_j^t$; assignment of compositions to segments: $Q_j^t$; combining multiple segmentations over consecutive frames: $\hat{Q}_j^t$; segment priors: $\gamma^{t,\nu}$; segment prototypes: $\theta^{t,\nu}$; classification of object in segment $\nu$: $C^{t,\nu}$; localization of the object: $X^{t,\nu}$.

Probability distributions over the part codebook from Section 9.4. The index of the codebook vector that received most probability mass is encoded in the brightness of the circle at $x_j^t$. The favorable saliency of the object region in this representation is valuable for a later segmentation of the object from background or other objects. In Figure 9.3 strong camera panning yields fairly similar optical flow for object and background. Compositions, however, can compensate for this difficulty and lay the basis for a satisfying segmentation.
Figure 9.5: Multiple segmentation hypotheses established for two frames. a) and d) show a 2 cluster solution. b) and e) 3 clusters. c) and f) 4 clusters. The segmentation with 3 clusters features the highest stability over the two frames and is, therefore, chosen by model selection.

9.7 Segmentation Hypotheses for Multiple Objects

In the following, several hypotheses for the locations and shapes of objects that are present in a video frame are to be derived from the compositions. As the number of objects that are present in a scene is a priori unknown, we have to address a difficult model selection problem. Therefore, segmentations with varying numbers of segments are established before pursuing a model selection strategy that retrieves only the most reliable segmentation.

A segmentation partitions compositions $\tilde{h}_j$ in the optical flow feature space into $K$ segments using histogram clustering (e.g. [PHB99]): Compositions defined by (9.16) are represented as multivariate distributions over the $k$-dimensional part codebook, $\tilde{h}_j = (\tilde{h}_{j,1}, \ldots, \tilde{h}_{j,k}) \in [0, 1]^k$ with $\sum_{l=1}^{k} \tilde{h}_{j,l} = 1$. Clustering then aims at representing $\tilde{h}_j$ by a mixture of $K$ clusters $\theta^{i,1}, \ldots, \theta^{i,K} \in [0, 1]^k$. In this mixture model the contribution of individual prototypes is weighted by mixture weights or class priors $\gamma^{i,1}, \ldots, \gamma^{i,K} \in [0, 1]$ for which $\sum_{\nu=1}^{K} \gamma^{i,\nu} = 1$ holds. The mixture model takes the
9.7. Segmentation Hypotheses for Multiple Objects

\[ p(\tilde{h}_t^j|\theta_t^{1}, \ldots, \theta_t^{K}, \gamma_t^{1}, \ldots, \gamma_t^{K}) = \sum_{\nu=1}^{K} \gamma_t^{\nu} p(\tilde{h}_t^j|\theta_t^{\nu}) \]  \hspace{1cm} (9.18)

Subsequently, the individual mixture components are approximated by multinomial distributions, i.e. for large \( N \in \mathbb{N} \) the distribution of \((\tilde{h}_{t,j}^l \cdot N)\) is multinomial with parameter \( \theta_t^{\nu} \). Transforming the definition of the multinomial distribution yields (here \( \lfloor \cdot \rfloor : \mathbb{R} \to \mathbb{Z} \) is the rounding function)

\[ p(\tilde{h}_t^j|\theta_t^{\nu}) = \frac{N!}{\prod_l \lfloor \tilde{h}_t^{j,l} \rfloor ! \prod_l (\theta_t^{\nu,l}) \tilde{h}_t^{j,l}} \times \exp \left\{ - \sum_l \tilde{h}_t^{j,l} \log (\theta_t^{\nu,l}) \right\} \]  \hspace{1cm} (9.19)

Here \( D_{KL}(\cdot||\cdot) \) denotes the Kullback Leibler distance between compositions and cluster prototypes, and the prefactors are for normalization purposes.

The clusters \( \theta_t^{\nu} \) and the assignment \( Q_t^j \in \{1, \ldots, K\} \) of compositions to clusters, i.e.

\[ P(Q_t^j = \nu) := \text{Prob}\{j\text{-th composition assigned to cluster } \nu\} \]  \hspace{1cm} (9.23)

are computed by iterating an expectation-maximization algorithm [MK97]. In the expectation-step, assignment probabilities of compositions to segments are computed conditioned on the current estimate of clusters,

\[ P(Q_t^j = \nu) := \frac{\sum_j P(Q_t^j = \nu)}{\sum_j P(Q_t^j = \nu)} \]  \hspace{1cm} (9.24)

In the maximization-step, class priors \( \gamma_t^{\nu} \) and cluster prototypes \( \theta_t^{\nu} \) are updated conditioned on the assignment probabilities

\[ \gamma_t^{\nu} := \frac{\sum_j P(Q_t^j = \nu)}{\sum_j P(Q_t^j = \nu)} \]  \hspace{1cm} (9.25)

After convergence of the EM-algorithm, the cluster assignment probabilities \( P(Q_t^j = \nu) \) of compositions represent the segmentation of a video frame into \( K \) segments. Since background is surrounding objects, the segment that covers most of the frame border is labeled as background, \( \nu = \text{BG} \). Figure 9.5 shows segmentations with 2, 3, and 4 segments for two video frames. Interest points in the different segments are displayed in distinct color (black is used for the background segment).
9.8 Model Selection for Identifying Reliable Segmentation Hypotheses

As there is no prior information regarding the number of objects that are present in a scene a model selection strategy has to be pursued to estimate the number of object segments. Therefore, segmentations $Q_j^K(t)$ for different numbers $K$ of segments are established in each frame (the following uses $K = 2, \ldots, 5$). Bipartite matching [CLRS01] is performed to make the current segmentation comparable with the one of the previous frame, i.e. labels are permuted so that they fit best to the preceding segment labeling. Thereafter multiple segmentations of consecutive video frames are combined into a single, more robust one $\hat{Q}_j^K(t)$, with

$$P(\hat{Q}_j^K(t) = \nu | Q_j^K(t), \hat{Q}_j^{t-1}(K)) \propto \begin{cases} \eta P(Q_j^K(t) = \nu) + (1 - \eta) P(\hat{Q}_j^{t-1}(K) = \nu | Q_j^{t-1}(K), \hat{Q}_j^{t-2}(K)), & \text{if } t > 1, \\ P(Q_j^K(t) = \nu), & \text{else}. \end{cases}$$ (9.26)

This dependency between segmentations of consecutive frames constitutes the Markov backbone that is represented at the bottom of the graphical model in Figure 9.4. It propagates segmentation hypotheses from previous frames into the current one.

An inappropriate model complexity is likely to yield unstable segmentations that change even when the input data varies only slightly. By observing the fluctuations of segmentations over multiple frames we can estimate their stability (cf. [RL04]) and select the most appropriate model complexity (see Figure 9.5 for an illustration).

The stability $\zeta(t)(K)$ of a $K$ cluster segmentation is measured by the entropies $H$ of the segment assignments

$$\zeta(t)(K) := \sum_j H(\hat{Q}_j^K(t)) = -\sum_j \sum_{\nu=1}^K P(\hat{Q}_j^K(t) = \nu) \log P(\hat{Q}_j^K(t) = \nu).$$ (9.27)

The optimal number of segments is determined by selecting the $K^*$ that minimizes this stability measure and we use the abbreviation

$$P(\hat{q}_j^K) := P(\hat{Q}_j^K(K^*) = \hat{q}_j^K).$$ (9.29)

Let $A_t^K$ denote the set of all compositions assigned to the $\nu$-th segment,

$$A_t^K := \{ j : \nu = \arg\max_{\nu'} P(\hat{Q}_j^K(K) = \nu') \}.$$ (9.30)
The location \( x^t(\nu) \) of the center of the \( \nu \)-th segment is estimated as the center of mass of all compositions \( j \in \mathcal{A}_t^\nu \) assigned to this segment,

\[
x^t,\nu := \frac{1}{|\mathcal{A}_t^\nu|} \sum_{j \in \mathcal{A}_t^\nu} x_j^t.
\] (9.31)

### 9.9 Learning Relevant Compositions

Not all compositions that belong to an object segment are actually characteristic for its category. The distractors which are mainly due to the weak labeling of the training videos (only the class label of the most prominent object in a whole sequence is given) should be discarded during learning. During recognition, relevance selection retrieves only the most characteristic compositions and thereby speeds up recognition. In the following, we build on the Bayesian approach to automatic learning of relevant compositions for categories that is introduced in Section 7.4.

To learn relevant compositions for category \( c \) we first need a set of irrelevant compositions as a set of negative samples. This set is acquired by randomly sampling compositions from training videos of all other categories. From a Bayesian standpoint, a composition \( h_j^t \) is relevant for representing objects of some category \( c \in \mathcal{L} \) (\( \mathcal{L} \) is the set of category labels), if it has a high likelihood \( p(h_j^t|\chi_c) \). The indicator function \( \chi_c \) is defined by \( \chi_c = 1 \) iff \( h_j^t \) is from a frame showing an object of category \( c \) (cf. (7.2)). Using Bayes' theorem, the likelihood can be factorized

\[
p(h_j^t|\chi_c) \propto P(\chi_c|h_j^t) p(h_j^t).
\] (9.32)

Here \( P(\chi_c) \) has been discarded, since all categories should be a priori equally likely. Incorporating the object center \( x^t,\nu \) from (9.31) and the position \( x_j^t \) of the composition yields an improved estimate of compositional relevance that is according to (7.6),

\[
p(h_j^t|\chi_c, x_j^t, x^t,\nu) = p(h_j^t|\chi_c, S_j^t,\nu = x^t,\nu - x_j^t) \propto P(\chi_c|h_j^t, S_j^t,\nu = x^t,\nu - x_j^t) p(h_j^t|S_j^t,\nu = x^t,\nu - x_j^t).
\] (9.33)

Here the relative position of a composition with respect to the object center is represented by the shift \( S_j^t,\nu = x^t,\nu - x_j^t \). Therefore, we exploit the fact that compositions are only depending on relative shifts, not on absolute object locations. In Equation (9.32) and (9.34), the relevance of compositions factorizes into two distributions. The first captures how discriminative \( h_j^t \) is, whereas the second indicates how reliably it can be detected. To circumvent density estimation of \( p(h_j^t|S_j^t,\nu) \) and to avoid overfitting when learning compositions the learning algorithm presented in Algorithm 7.2 is again employed.
9.10 Object Recognition Using the Compositional Shape Model

In every frame of a novel test video, the object in each segment has to be recognized. Therefore, the individual compositions that have been established must be combined to obtain a concerted object hypothesis. In a first step irrelevant clutter compositions are to be filtered out keeping only compositions that are actually relevant for recognition. This step is discussed in Section 7.6.1 and it is based on the estimate of compositional relevance, \( p(h_{jt} | \chi_c, x_{jt}, x^{t,\nu}) \), that has been learned during training.

All remaining relevant compositions \( h_{jt}, j \in A^\nu_t \) that are assigned to a segment \( \nu \) are coupled in the graphical model shown in Figure 9.4. This statistical model builds on the compositional shape model from Section 4.5.4. The category \( c_{t,\nu} \in \mathcal{L} \) of the object in segment \( \nu \) can then be inferred from its posterior distribution (given the collected evidence \( E \))

\[
P(c_{t,\nu} | E) \propto p \left( \{h_{jt}, x_{jt}, \hat{q}_{jt}\}_{j \in A^\nu_t}, \{h_{jt}^\nu, x_{jt}^\nu, \hat{q}_{jt}^\nu\}_{j \notin A^\nu_t} | c_{t,\nu}, x_{t,\nu} \right) P(c_{t,\nu} | x_{t,\nu}) \tag{9.35}
\]

by applying Bayes' formula. Now the denominator can be omitted because it is independent of \( c_{t,\nu} \). Furthermore, the category of an object should be independent of its absolute position in a frame and there should be no bias on any category, i.e. all classes are a priori equally likely. Therefore, \( P(c_{t,\nu} | x_{t,\nu}) \) can be discarded as well, which yields

\[
P(c_{t,\nu} | E) \propto p \left( \{h_{jt}^\nu, x_{jt}^\nu, \hat{q}_{jt}^\nu\}_{j \notin A^\nu_t}, \{h_{jt}, x_{jt}, \hat{q}_{jt}\}_{j \in A^\nu_t} | c_{t,\nu}, x_{t,\nu} \right) . \tag{9.36}
\]

Since the category of segment \( \nu \) determines only compositions that have been assigned to this segment (i.e. \( j \in A^\nu_t \)), all other compositions are independent of \( c_{t,\nu} \) and can be skipped. Moreover, an assignment to segment \( \nu \) implies \( \hat{q}_{jt}^\nu = \nu \). Therefore \( \hat{q}_{jt}^\nu \) can be dropped as well for \( j \in A^\nu_t \) and we obtain

\[
P(c_{t,\nu} | E) \propto p \left( \{h_{jt}^\nu, x_{jt}^\nu\}_{j \notin A^\nu_t}, \{h_{jt}, x_{jt}, \hat{q}_{jt}\}_{j \in A^\nu_t} | c_{t,\nu}, x_{t,\nu} \right) . \tag{9.37}
\]

Compositions are conditionally independent, conditioned on the object model parameters \( c_{t,\nu} \) and \( x_{t,\nu} \). Therefore, the likelihood factorizes and we can apply Bayes' formula again to obtain

\[
P(c_{t,\nu} | E) \propto \prod_{j \in A^\nu_t} \frac{P(c_{t,\nu} | x_{t,\nu}, h_{jt}^\nu, x_{jt}^\nu) \cdot p(h_{jt}^\nu, x_{jt}^\nu | x_{t,\nu})}{P(c_{t,\nu} | x_{t,\nu})} . \tag{9.38}
\]
The factor $p(h_j^t, x_j^t|x_t^{\nu})$ does not depend on the object category and can be omitted. Moreover, the category of an object should be independent of its absolute position in a frame and there should be no bias on any category. Therefore, $P(c_t^{\nu}|x_t^{\nu})$ can again be left out and we obtain

$$
P(c_t^{\nu}|E) \propto \prod_{j \in A_t^\nu} P(c_t^{\nu}|h_j^t, S_j^{t,\nu} = x_t^{\nu} - x_j^t)$$

(9.39)

$$
= \exp \left[ \sum_{j \in A_t^\nu} \ln P(c_t^{\nu}|h_j^t, S_j^{t,\nu} = x_t^{\nu} - x_j^t) \right].
$$

(9.40)

Here the relative position of a composition with respect to the object center is represented by the shift $s_j^{t,\nu} = x_t^{\nu} - x_j^t$. NKDA (see Section 4.3.2) is used to estimate the distribution in (9.40). During recognition, an object can be recognized efficiently by applying the classifier to all compositions $h_j^t$ and computing (9.40).

### 9.11 Experiments

In still image categorization the compositional approach has been shown to yield competitive performance to state of the art approaches on standard benchmark databases (see [OB06, OB07b]). Comparable benchmark datasets are, however, still needed for weakly supervised, multi-category object recognition and segmentation with moving camera. Therefore, a database for category-level object recognition in video has been assembled for this project. It consists of 24 videos per category (categories car, streetcar, pedestrian, and bicycle). As can be seen from the examples in the figures, videos feature large intra-category variation (cf. Figure 9.6c and 9.7i), occlusion (e.g. Figure 9.6i), significant scale and viewpoint variation (e.g. Figure 9.7a, i), camera panning (cf. Figure 9.3), and background clutter. In the subsequent experiments 10-fold cross-validation is performed. For each cross-validation step a random sample of 16 videos per category is drawn for training and the remainder is used for testing. Learning proceeds then on a randomly selected subset of 15 frames per video, while testing is performed on each frame. To avoid a bias towards categories with many test frames the standard evaluation procedure in image categorization is followed: the retrieval rates are computed separately for each category before averaging these scores over all categories (cf. Section 4.6.1).
9.11.1 Evaluating the Different Building Blocks of the Composition System

In the following, a series of experiments is conducted to evaluate the system design decisions: To estimate how much performance we gain from the individual components, all key parts are disabled in a first experiment. Adding individual components in subsequent experiments and also comparing to an approach like [DTS06] yields a performance gain that underlines the importance of each individual part of the composition system. In this first series of experiments only the most prominent object in a frame is to be detected.

Baseline Performance of a Bag-of-Features Approach

The presented approach establishes an intermediate compositional representation that is based on groupings of elementary parts and the spatial structure of objects. To evaluate the gain of compositionality, this hidden representation layer is neglected in a first experiment. A frame is then simply represented by a bag-of-parts representation \( b' \) over all localized feature histograms \( e_i \) that have been extracted at interest points in the image (see Section 9.4)

\[
\mathbf{b'} \propto \sum_i \left( P(F_i = 1|e_i), \ldots, P(F_i = k|e_i) \right) ^\top.
\] (9.41)

The object in a frame is then labeled with the category of highest posterior \( P(c|\mathbf{b'}) \). The posterior is again learned from the training data using NKDA. This approach yields a retrieval rate of 53.1 ± 5.5%.

Compositional Segmentation and Recognition w/o Compositions

This experiment is designed to measure the benefit of combining segmentation with recognition. Therefore, compositions are inferred as described in Section 9.5 and 9.6. The prominent object in a video frame is then segmented from background clutter by establishing a 2-class segmentation as described in Section 9.7. Since only a single segmentation hypothesis is established no model selection is required. All compositions that are not assigned to the background, \( \nu \neq BG \), are then taken into account to recognize the most prominent object. Therefore, these compositions are combined using a bag of compositions descriptor \( \mathbf{\tilde{b}'} \propto \sum_{j \in A'_{\nu \neq BG}} \mathbf{h}_{ji} \). Frames are then categorized without the compositional shape model by simply selecting the category with highest posterior \( P(c'|\mathbf{\tilde{b}'}) \). Combining foreground segmentation with bag-of-features recognition improves the retrieval rate to 64.5 ± 5.5%.
9.11. Experiments

Compositional Segmentation and a Fixed Grid of Compositions like [DTS06]

Now the model that is used for recognition is to be extended by following the approach of Dalal et al. [DTS06]. Therefore, we divide the bounding box of the foreground segment into a regular grid of four subsegments. Each of these segments is then represented by a bag $\vec{b}_i$ of all the compositions which it covers. The four individual bag descriptors are then concatenated to obtain a joint representation of the segment. Thereby spatial information is incorporated. Recognition is then based on maximizing the posterior $P(c|\vec{b}_1, \ldots, \vec{b}_4)$. This model improves the performance to $68.5 \pm 4.3\%$.

Segmentation, Relevance Detection, and Compositional Shape Model for Recognition

In contrast to the previous experiment we now use the compositional shape model of Section 9.10 to recognize the foreground object. Moreover, relevance detection as presented in Section 9.9 is incorporated to retain only the category specific compositions. The complete system performs at $74.3 \pm 4.3\%$ and the category confusion table is presented in Table 9.1.

<table>
<thead>
<tr>
<th>True classes $\rightarrow$</th>
<th>bicycle</th>
<th>car</th>
<th>pedestrian</th>
<th>streetcar</th>
</tr>
</thead>
<tbody>
<tr>
<td>bicycle</td>
<td>73.5</td>
<td>5.1</td>
<td>13.3</td>
<td>4.7</td>
</tr>
<tr>
<td>car</td>
<td>10.2</td>
<td>83.1</td>
<td>13.2</td>
<td>15.2</td>
</tr>
<tr>
<td>pedestrian</td>
<td>13.1</td>
<td>4.0</td>
<td>65.0</td>
<td>4.6</td>
</tr>
<tr>
<td>streetcar</td>
<td>3.3</td>
<td>7.6</td>
<td>8.5</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Table 9.1: Confusion table (percentages) for the complete system.

In another setting, videos are categorized as a whole and not the individual frames. For this task the category hypothesis is chosen that is most consistent with all video frames. Here the compositional model performs at $87.4 \pm 4.8\%$. Performance increases significantly since information from an ensemble of frames can be used simultaneously.

Grouping atomic parts of limited reliability into compositions that can be tracked reliably has condensed information and improved the robustness of object representations. The underlying statistical inference problem can be solved efficiently. As a result the compositional model segments, tracks, and recognizes objects in videos of full PAL resolution ($768 \times 576$ pixel) at the order of 1 fps on an ordinary desktop
computer. Finally, increasing the codebook size from 60 dimensions in the previous experiments to 200 dimensions yields a performance gain of approximately 2% for all of the previous experiments. Larger codebooks, however, did not improve performance any further.

9.11.2 Multi-Object Recognition

In the following experiment, multiple objects that appear simultaneously in a video have to be recognized. Therefore, the model selection strategy of Section 9.8 is applied to find the correct number of objects. A frame is then correctly classified if all the present objects have been found. Missing an object or detecting an object that is not present counts as an error. Given this significantly harder task our full compositional model classifies $71.0 \pm 3.7\%$ of all frames correctly.

9.11.3 Analyzing the Contributions of Individual Compositions

In the following, we analyze the contribution of individual compositions to recognizing objects. Therefore, the category posterior of compositions

$$P(c^{t,v}|x^{t,v}, h^t_j, x^t_j, \hat{q}^t_j)|c^{t,v} = \text{True Category}$$

(9.42)

is evaluated for each composition. In Figure 9.6 and 9.7 the category posterior is then encoded i) in the darkness of a circle around the composition center and ii) in the opaqueness of the underlying image region, i.e. alpha blending is used for visualization. Moreover, Figure 9.6 shows the propagation of an object segment over several frames.
Figure 9.6: Visualizing the contributions of compositions $h_j^t$ to object recognition. Dark circles correspond to compositions with high posterior from (9.42). The gap between a) and c) and between e) and g) is both times 60 frames. i) and k) have a gap of 10 frames. Class labels are placed at the location of the segment center $x$ (c: car, p: pedestrian).
Figure 9.7: Visualization of compositions. See Figure 9.6 for details.
Discussion

This thesis has investigated the design of composition systems for visual object recognition in real world scenarios involving large numbers of object categories with high intra-class variations. The underlying statistical object model captures the compositional structure of object categories. It is learned without any user supervision regarding the compositional nature of objects and without requiring hand segmentations or other localization information during training. The underlying learning algorithm has been derived following a Bayesian approach. It selects relevant compositions out of a set of composition candidates that are established using perceptual organization. Object models are founded on a codebook of atomic parts that is shared by all categories. The descriptors that are underlying these compositional atoms have been specifically designed to have low dimensionality and to exhibit robustness with respect to intra-category variations. They have been shown to outperform a standard descriptor as feature for compositionality. The semantic gap between these low level features and the final high level object representation needed for recognition is bridged by establishing intermediate compositions. All compositions are represented as probability distributions over their constituents. Moreover, a Bayesian network has been presented that couples all compositions, object shape, and scene context. It has then been shown how concerted object hypotheses can be calculated in a computationally efficient manner by means of statistical inference. Given the generative object models, evidence can also be propagated in the direction that is inverse to that of recognition. Consequently, object representations can be sampled conditioned on the object parameters. This process renders it possible to fill in missing information and it provides a detailed analysis of the underlying object models.
Moreover, compositionality has been demonstrated to be not only valuable for rendering learning statistically tractable but also for turning recognition into a computationally feasible problem. Therefore, the approach has been extended to category-level recognition, segmentation, and tracking of multiple objects in videos in near real-time. A stability-based model selection lays the basis for multi-object detection and classification. As a result, neither training nor recognition require any manual interaction. Only the category label of the most prominent object in a complete video sequence is required during training. Moreover, there are no simplifying constraints imposed on videos. Experiments have confirmed that the method is capable of recognizing multiple objects in videos gathered with a hand-held camera that records scenes with heavy clutter and camera motion.

**Interesting Outcomes**

A major goal of this thesis has been to devise algorithms that automatically discover structure in our visual world without requiring user supervision. In particular, this means that structured object models are learned without demanding any information regarding the compositional nature of objects and without needing any localization information such as hand segmentations. At the outset of this thesis it was unclear if and how such complex structures could be learned without supervision. Typically this problem is avoided either by requiring intensive user supervision [FH05] (e.g. manual pinpointing of part configurations), or by limiting the structure to few parts (e.g. [FPZ03]), or even by restricting the representation to unstructured models such as bag-of-features [CDF+04]. Other research that has studied complex compositional models such as the work of Jin and Geman [JG06] has explicitly excluded the question of learning such structures and identified this problem as a future challenge. Therefore, it is remarkable that this line of work could actually show how complex structured object models can be learned without requiring tedious supervision. Key to this success is the compositional approach which divides the complex learning task into tractable subproblems and interlinks them. As a result the developed vision system is able to learn what relevant compositions are and it can even segment the object from the clutter in a scene. These results demonstrate that hidden structure is automatically discovered without manual supervision. Besides employing the compositional idea, these capabilities are also heavily relying on statistical modeling together with current machine learning techniques and they underline the potential of this combination.

Another interesting outcome is the observation that it is actually possible to learn
and represent object models without having to design highly specific descriptors or keypoint detectors that are category specific. In contrast to such category specific descriptors the compositional representation is rather based on a small codebook of low dimensional atomic parts that are shared by all categories. This small codebook is not at all class specific. Nevertheless, the higher levels of the compositional representation are capable of capturing all relevant details of object categories. The information that is lacking in the low level representation is compensated by relations on higher levels of the compositional hierarchy. These relations can be learned for new categories without having to adapt the low-level representation. Moreover, the compositional approach is not dependent on the presence of few, unreliable, highly specific features that act as fingerprints of an object. Rather it automatically combines numerous unspecific features with characteristic relations between them to obtain robust object representations. This finding demonstrates that it is actually possible to represent object models based on generic, low-level descriptors. Human cognition and especially language features related abilities. A small set of atomic parts such as phonemes or letters suffices to represent arbitrary words and sentences in spoken or written language. New concepts can then be represented by establishing characteristic relations between atomic parts. Hence words and sentences are formed that describe the new concepts without having to expand the set of atomic parts.

We have seen that compositional representations can be used to decompose complex object models into simpler submodels. Thereby, learning of structured object models is turned into a feasible learning task. However, compositional representations might entail disadvantages during recognition, especially under the high computational demands posed by videos. For that reason a compositional approach to video analysis has been investigated. It performs near real-time and demonstrates how the key concept of compositionality can actually be exploited for both, rendering learning tractable and making recognition computationally feasible. Thus recognition performance is not compromised by choosing compositional models for simplifying learning. Additionally it has been shown that tracking, segmentation, and recognition can all be closely coupled in a single statistical model so that all three individual processes benefit.
Bibliography


<table>
<thead>
<tr>
<th>Reference</th>
<th>Title and Authors</th>
</tr>
</thead>
</table>
Index

appearance patches .............. 17
artificial intelligence .......... 3
atomic parts .................. 35, 47
attention control ............... 16

bag-of-features model .......... 25
bag-of-words model ............ 25
Bayesian networks .............. 44
belief .......................... 45, 56
belief propagation ............. 56, 61
binding problem .............. 19

categorization ................. 3
category
   functional .................. 7
   hierarchy .................. 106
   visual ..................... 7
classifier
   discriminative ............ 40
   generative ................ 40
   nearest neighbor ........ 9
clique ........................ 43
   maximal ................... 43
color constancy ............... 10
composition rules ............. 36
compositional shape model .... 83, 85
compositionality ............ 2, 6, 34, 35
computer vision ............... 2
confusion matrix ............. 152
conjugate gradient ........... 77
content-based image retrieval .. 24
contingency table ............. 152
correspondence problem ....... 20
derivative of Gaussian ........ 51
descriptor .................... 16
   local ........................ 16
   difference of Gaussian ...... 17
difference of Gaussian ......... 17

top-down ..................... 95
evidence ..................... 45
exemplar detection ............ 7
expectation-maximization .... 26
factor graph .................. 43
feature ....................... 16
   local ........................ 17
Fisher's linear discriminant analysis 24

Gabor filter ................... 19
generative model ............. 19
geometric blur ................. 4
geons ........................... 24, 35
Gestalt laws ................... 36
graph theory .................. 42
graphical models .............. 42
grouping
   bottom-up .................. 95
top-down ..................... 95
Harris corner detector ....... 17
<table>
<thead>
<tr>
<th>Hypothesize and test</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hysteresis thresholding</td>
<td>52</td>
</tr>
<tr>
<td>Image processing</td>
<td>3</td>
</tr>
<tr>
<td>Image understanding</td>
<td>3</td>
</tr>
<tr>
<td>Interest point detection</td>
<td>16</td>
</tr>
<tr>
<td>Kadir and Brady detector</td>
<td>17</td>
</tr>
<tr>
<td>Kernel function</td>
<td>72</td>
</tr>
<tr>
<td>Trick</td>
<td>72</td>
</tr>
<tr>
<td>Latent Dirichlet allocation</td>
<td>26</td>
</tr>
<tr>
<td>Localized feature histograms</td>
<td>53</td>
</tr>
<tr>
<td>Machine learning</td>
<td>4</td>
</tr>
<tr>
<td>Markov random field</td>
<td>43</td>
</tr>
<tr>
<td>Message passing</td>
<td>45</td>
</tr>
<tr>
<td>Model</td>
<td>2</td>
</tr>
<tr>
<td>Model-based vision</td>
<td>20</td>
</tr>
<tr>
<td>Moment invariants</td>
<td>20</td>
</tr>
<tr>
<td>Multi-class classification</td>
<td>78</td>
</tr>
<tr>
<td>Newton, Isaac</td>
<td>2</td>
</tr>
<tr>
<td>Non-accidental principle</td>
<td>36</td>
</tr>
<tr>
<td>Non-maximum suppression</td>
<td>52</td>
</tr>
<tr>
<td>Nonlinear kernel discriminant analysis</td>
<td>77</td>
</tr>
<tr>
<td>Optical flow</td>
<td>160</td>
</tr>
<tr>
<td>Pairwise coupling</td>
<td>78</td>
</tr>
<tr>
<td>Part-based models</td>
<td>21, 27</td>
</tr>
<tr>
<td>Parts and structure model</td>
<td>28</td>
</tr>
<tr>
<td>Pattern recognition</td>
<td>4</td>
</tr>
<tr>
<td>Perceptual grouping</td>
<td>23</td>
</tr>
<tr>
<td>Perceptual organization</td>
<td>36</td>
</tr>
<tr>
<td>Plate notation</td>
<td>57</td>
</tr>
<tr>
<td>Plutarch</td>
<td>1</td>
</tr>
<tr>
<td>Pragnanz</td>
<td>36</td>
</tr>
<tr>
<td>Precision recall curve</td>
<td>155</td>
</tr>
<tr>
<td>Primitives</td>
<td>35</td>
</tr>
<tr>
<td>Principle component analysis</td>
<td>20, 24</td>
</tr>
<tr>
<td>Probabilistic latent semantic analysis</td>
<td>26</td>
</tr>
<tr>
<td>Receiver operator characteristic</td>
<td>152</td>
</tr>
<tr>
<td>Recognition</td>
<td>2, 3</td>
</tr>
<tr>
<td>Representation</td>
<td>2, 3</td>
</tr>
<tr>
<td>Retrieval rate</td>
<td>88</td>
</tr>
<tr>
<td>Semantic gap</td>
<td>33</td>
</tr>
<tr>
<td>Shape</td>
<td>19, 21</td>
</tr>
<tr>
<td>Shape context</td>
<td>18</td>
</tr>
<tr>
<td>SIFT features</td>
<td>18</td>
</tr>
<tr>
<td>Signal-to-noise ratio</td>
<td>51</td>
</tr>
<tr>
<td>Sliding windows</td>
<td>20</td>
</tr>
<tr>
<td>Structure</td>
<td>1</td>
</tr>
<tr>
<td>Sum-product algorithm</td>
<td>56</td>
</tr>
<tr>
<td>Support vector machine</td>
<td>73</td>
</tr>
<tr>
<td>Template matching</td>
<td>20</td>
</tr>
<tr>
<td>Deformable</td>
<td>21</td>
</tr>
<tr>
<td>Thermodynamics, second law of</td>
<td>1</td>
</tr>
<tr>
<td>Tracking</td>
<td>161</td>
</tr>
<tr>
<td>View-based approach</td>
<td>20</td>
</tr>
</tbody>
</table>
Curriculum Vitae

Name: Björn Ommer
Citizenship: German
Date of birth: October 28, 1979
Place of birth: Cologne, Germany

1986 – 1990 Primary school in Pulheim, Germany.
1998 Graduated with ’Allgemeine Hochschulreife (Abitur)’.

1998 – 2003 Diploma studies in computer science with minor in physics at the Rheinische Friedrich-Wilhelms Universität Bonn, Germany.
2000 – 2002 Teaching and research assistant,
Institute of computer science, University of Bonn.
2003 Graduated with diploma in computer science (Diplom Informatiker).

2003 Ph.D. studies in computer science and teaching and research assistant in the Computer Vision and Pattern Recognition Group (Inst. of computer science), Rheinische Friedrich-Wilhelms Universität Bonn, Germany.
Since Sept. 2003 Continuation of Ph.D. studies in computer science and teaching and research assistant in the Machine Learning Group (Inst. of computational science), ETH Zurich, Switzerland.