Foundations and Trends® in
Computer Graphics and Vision
Vol. 1, No 4 (2005) 255–353

DOI: 10.1561/0600000003

Object Categorization

Axel Pinz[1]

[1] *Graz University of Technology, Austria, axel.pinz@tugraz.at*

Abstract

This article presents foundations, original research and trends in the field of object categorization by computer vision methods. The research goals in object categorization are to detect objects in images and to determine the object's categories. Categorization aims for the recognition of generic classes of objects, and thus has also been termed 'generic object recognition'. This is in contrast to the recognition of specific, individual objects. While humans are usually better in generic than in specific recognition, categorization is much harder to achieve for today's computer architectures and algorithms. Major problems are related to the concept of a 'visual category', where a successful recognition algorithm has to manage large intra-class variabilities versus sometimes marginal inter-class differences. It turns out that several techniques which are useful for specific recognition can also be adapted to categorization, but there are also a number of recent developments in learning, representation and detection that are especially tailored to categorization.

Recent results have established various categorization methods that are based on local salient structures in the images. Some of these methods use just a 'bag of keypoints' model. Others include a certain amount of geometric modeling of 2D spatial relations between parts, or 'constellations' of parts. There is now a certain maturity in these approaches and they achieve excellent recognition results on rather complex image databases. Further work focused on the description of shape and object contour for categorization is only just emerging. However, there remain a number of important open questions, which also define current and future research directions. These issues include localization abilities, required supervision, the handling of many categories, online and incremental learning, and the use of a 'visual alphabet', to name a few. These aspects are illustrated by the discussion of several current approaches, including our own patch-based system and our boundary fragment-model. The article closes with a summary and a discussion of promising future research directions.

1

Introduction

This article provides a review of existing representations, algorithms, systems and databases for visual object categorization. It describes the state of the art in this field, which has been a long standing goal, and is still a mainly unsolved problem in computer vision research. The time chosen for writing is motivated by recent success in recognition from local, salient parts, which can be considered a significant step towards object categorization.

Who are the supposed readers of this document, and what potential benefits are there for them? Students and graduate students in computer vision will get a thorough review of the state of the art in visual object categorization. Researchers in computer vision might benefit from a more complete point of view, including a number of approaches which they have not focused on within the scope of their own research. Researchers in related fields should find this article a valuable reference.

But the article goes beyond a pure review of the state of the art. It includes original research in categorization, presents a prototype system for categorization, discusses our databases and provides experimental results on object categorization and localization in still images.

1.1 Problem statement

We can define visual object categorization as the process of assigning a specific object to a certain category. This process has also been termed ‘generic object recognition’ (generic OR), and it is in contrast to ‘specific OR’, which deals with the recognition of a specific, individual object. Examples of categories in generic OR are people, children, dogs, cars, bikes or dishes, while specific OR might aim at recognizing a certain individual, like Albert Einstein, or a specific object like my car. An individual object might also be termed a specific *instance* of a more generic category. Categories can also be organized in hierarchies (child – human being – mammal), and categories might overlap – a tall glass might be used as a vase. Throughout the remainder of this article, we will use the terms *‘categorization’* for visual object categorization or generic OR, and *‘specific OR’* for the recognition of individual objects.

Looking at humans, and comparing their recognition performance with artificial systems, it turns out that humans are much better in categorization than machines, but specific OR can often be handled more efficiently, reliably or simply faster by an artificial vision system. VanRullen and Thorpe [199] point out that humans can perform ultra-rapid categorization tasks. They can decide whether a briefly flashed image belongs to a certain category in less than 150ms, and they provide experimental evidence for the two categories ‘animal’ and ‘means of transport’. On the other hand, there are numerous solutions to industrial inspection, which recognize and localize specific objects much faster and much more reliably than humans can do (see [31] for an example of such an industrial product, and [76] for the underlying theoretical foundations). A further aspect of categorization is the sheer number of visual categories. There is evidence from cognitive psychology that humans deal with about 30,000 different categories (see Biederman [21]). This would require solving currently intractable computational complexity.

This article sets out to answer the following questions: How can artificial systems perform categorization? What are the key building blocks that are required to build a categorization system? What are the main challenges? What are the bottlenecks and unsolved problems? This will also shed light on the more general question: Why is categorization simpler for humans than for machines and why is specific OR simpler for machines than for humans?

1.2 Historical development

In the following, I give a very brief sketch of some major landmarks in the history of object recognition research. This is not meant to be a complete review, but rather some useful information to pave the way for later discussion. One of the major early landmarks is certainly the work of David Marr [126], who proposed viewer-centered and object-centered representational levels (image – primal sketch – 2-1/2D sketch – 3D object model), as well as visual modules which can be used to generate these descriptions (e.g. ‘shape from X’ to produce a 2-1/2D sketch). Marr’s ideas influenced at least a decade of research, and have led to a so-called ‘reconstruction school’ which advocates that 3D reconstruction and 3D modeling of a scene (and thus of the objects in the scene) are necessary for further reasoning.

On the other hand, there is the ‘recognition school’ which favors working in the 2D domain, with 2D images, features and descriptors which are extracted from these images. Their pattern classification [43] or pattern recognition [145] approach is fundamentally different from the reconstructionist paradigm. Much of the content of this article actually is in the spirit of a ‘recognition school’ approach. We will discuss, for example, the ‘bag of keypoints’ approach, in which salient points are extracted from images, and descriptors are calculated to form feature vectors. These feature vectors can be used to learn a discriminative model from training images, and to recognize (categorize) test images. But we will also present the generative ‘constellation’ model, which employs a ‘light’ 2D geometry in terms of spatial 2D relations between key parts of the object model.

Only very recently have we seen efforts to combine discriminative and generative approaches in categorization research. This confluence of recognition and reconstruction schools has already been predicted by Aloimonos and Shulman [4] in 1989.

There are further milestones which should be mentioned. Biederman [20, 21] proposed his ‘recognition by components’ (RBC) theory. Volumetric primitives, so-called ‘geons’ can be used to recognize objects in a qualitative (and thus generic) way. While this theory is quite elegant, its implementations (see [18, 39]) lacked due to low level vision problems, so that geon-based recognition has not been applied to real-world categorization problems. Research in perceptual grouping proceeds in a similar manner [167]. Low-level 2D primitives are grouped to build object descriptions, either in a pure bottom-up (data driven) manner, or top-down, including prior knowledge (models) about the expected image content. At the other end of the spectrum of potential solutions to OR, we find the idea of purely image- or ‘appearance’-based recognition, for instance in parametric eigenspace [143]. This idea has triggered a vast number of extremely successful appearance-based approaches to specific OR.

In general, there has been more research in specific OR than in categorization in the past. Success in specific OR has influenced a number of approaches to categorization, although most of the developed algorithms for specific OR are not directly applicable to categorization. There is a paradigm of specific OR by alignment, in which spatial correspondence between groups of image features and model features is found by searching for the geometric transformation that aligns these features best. This includes affine transformations for planar objects [84] and 3D model to 2D image feature matching [121]. Another way to compare image and model features is to extract features which are invariant against certain geometric [142] or radiometric distortions [3]. Efficient indexing is needed, when a database of potentially many object models has to be matched against features extracted from a query image that contains a certain specific object. This can, for instance, be done by geometric hashing [207], a technique which is robust against partial occlusion and geometric transformations. Finally, the success in global appearance-based recognition [143] has motivated research in local appearance-based methods for specific OR [122]. At this point, we can observe that techniques for specific OR and for categorization meet.

In categorization, these various ideas have led to the development of a number of recent approaches which try to:

- model appearance more locally,
- group simple geometric primitives, and
- use learning algorithms to find common patterns that can be shared over many individuals of a category.

Within the past 5 years, we have seen a rapid development and rise in the success of object categorization in increasingly difficult, cluttered, and realistic scenes[1]. We can also observe a number of contributions from related fields as machine learning, neurosciences and cognitive psychology.

1.3 Potential applications

There are a number of obvious applications of categorization to image database annotation, image retrieval and video annotation. But potential applications of categorization go far beyond that. Reliable categorization in real-time will open up applications in surveillance, driver assistance, autonomous robots, interactive games, virtual and augmented reality and telecommunications. A more general view might include systems for ‘cognitive personal assistance’ with many potential aspects, ranging from user support in complex environments to very basic support capabilities for elderly or disabled people.

1.4 Outline of this review article

The article is structured in three major parts (Sections 2 – 4). I start with an in-depth analysis of major issues related to solving the problem of categorization mentioned in Section 2. This analysis provides at the same time an introduction to the main topics, which are then discussed in detail in Section 3 which presents the major building blocks for

[1] This may partly be related to recent European research initiatives. There has been substantial funding of basic research in ‘Cognitive Vision’ within the 5th framework program of the European Union, with an even broader perspective of ‘Cognitive Systems’ in the current, 6th framework program. There has been strong support of categorization research within these programs.

categorization systems. Finally, Section 4 presents two major aspects of our own research in categorization: a region-based approach, and categorization with a boundary-fragment-model.

You will probably recognize that the subject is quite broad and heterogeneous (ranging from the representation of scale in images, over machine learning, to 2D spatial models for categorization). Thus, there is no isolated section on the 'state of the art' and related work. I prefer, rather, to cite relevant publications throughout the whole article, which is hopefully more useful to you, the potential reader.

Finally, there is a common thread, which should provide some extra value for those who manage to read the complete article sequentially. However, many sections stand on their own and may also be consulted individually.

2

Categorization as an Issue of ...

What are the main problems to be faced (and solved) by an artificial categorization system? There are several obvious ones: *Representation, recognition,* and *learning* are the major topics which have been addressed in numerous contributions to the field (see e.g. Perona's presentation [161] or the tutorial by Fei-Fei, Fergus and Torralba at ICCV 2005 [48]). But in practice we encounter further, less obvious issues. Many systems only categorize images, but they cannot *localize* and tell us the exact position and delineation of the object in the image. Only recently did the community become fully aware of a *database* problem, because it turns out that often background (context) is learned rather than object-specific information. This leads us to the issue of *evaluation*: How to evaluate the output of a categorization system? To finish with a more practical issue, *system integration* can get quite complex when many components are required to interact smoothly. We proceed by discussing all of these seven aspects of categorization below.

2.1 ... classification

Any categorization system will have to deal with some sort of visual input like color, monochrome, and thermal images, or image sequences.

Often, the raw images are processed and features are extracted. This reduces the amount of data, while it hopefully maintains the important (salient, interesting, relevant, discriminative) visual information. Examples include detection and description of points of interest [35, 153], as well as the extraction of robust feature sets on a fixed grid basis [36]. Some systems also try to represent object parts and their spatial relationship [52, 204]. The goal is to extract and to learn as much as possible from a number of examples, with as little human supervision as needed. Issues of learning and representation will be discussed in detail later in Sections 2.2 and 2.3. For the moment let us assume that images or image sequences have been reduced to feature vectors, and that a category model has been learned from these features. The classification or recognition problem can then be formulated as follows: Given a number of learned categories, a new image/sequence should be processed and a decision should be drawn, whether a known category appears in the data or not. More formally:

Let $\mathcal{C}$ denote the set of categories c_m, and $\mathcal{I}$ the space of visual 'input events' I_n. Typically, I_n is an image, but it might also be a sequence $I_{n,t_1},\ldots,I_{n,t_k}$ of images taken at discrete instances in time, $t_1,\ldots,t_k$. Next, $\mathbf{f}_n = (f_{n,1},\ldots,f_{n,j})^T$ denotes a feature vector extracted from input event I_n. Let us for the moment assume fixed numbers of M categories, N input events, with J dimensions of a feature vector, and K describing the length of an image sequence. The task of classification can then be formulated as follows: Given a new (previously unseen) visual input event I_y, calculate the corresponding feature vector $\mathbf{f}_y$. From $\mathbf{f}_y$, infer the corresponding category c_x, or decide that visual event I_y is not related to any previously learned category $c_m,\ m = 1\ldots M$.

This is a common pattern recognition problem which is often approached in a probabilistic manner: We look for $P(c_x|I_y)$, the probability that we detect category c_x, given a certain input event I_y. One possibility to decide on c_x is to evaluate the modes of the discrete probability distribution $P(c_m|I_y),\ m = 1\ldots M$ and to choose m such that P is maximum *(Bayes decision rule)*:

$$c_x : x = \arg\max_m P(c_m|I_y). \tag{2.1}$$

Parametric techniques will try to find a model, for instance, a probabilistic model of the form $P(\mathbf{f}_y|c_x)$: Given a category c_x, how likely is a corresponding feature vector $\mathbf{f}_y$? Typically, such models will be estimated from a number of training examples.

On the other hand, there also exist nonparametric techniques, which work directly in the feature space. Such methods are needed when we have to deal with highly overlapping parametric models or with distributions in feature space that can hardly be modeled explicitly. To pick a popular example, consider k-nearest-neighbor (k-nn) estimation: Each training example I_t delivers a feature vector $\mathbf{f}_t$ that can be mapped to a point in feature space. A number of such points for each category is collected during the training phase. For recognition, a test input I_y is projected to a point in feature space. The k nearest neighbors of this point are then voting for their respective category. I_y is decided to belong to category c_x which has the highest number of votes (majority voting).

For further reading, there are several highly recommendable textbooks on pattern recognition and classification including [43][1] and [145].

But categorization imposes a number of difficult constraints and boundary conditions on established pattern recognition techniques. We have to deal with large intra-class variability, and potentially small inter-class differences. We would like to learn a category from a few examples, but at the same time we extract many features, ending up with high-dimensional but very sparsely occupied feature spaces. Learning of category models will require defining distances and finding clusters in these feature spaces. In addition, we note that variability (intra- as well as inter-class) might very much depend on the extracted features themselves. It might be wise to develop sets of category-specific features (e.g. a skin color detector for the category of human faces or certain texture measures for vegetation).

[1] See Chapter 2 of [43] for Bayes' decision theory, Chapter 3 for maximum likelihood estimation, and Chapter 4 for nonparametric techniques including k-nearest-neighbor estimation.

2.2 . . . learning

A visual categorization system should learn from a number of examples (training images or training sequences). Learning can be performed in a supervised or in an unsupervised manner. In the *supervised* case, training data is comprised of visual input events I and the desired category labels, and the parameters of a category or the boundaries between categories in a feature space have to be learned. In the *unsupervised* setting, just the visual input is presented, without providing the correct labels. Clusters in the feature space have to be found by the learning algorithm [90], and sometimes even the number of clusters is not provided (see 'the problem of validity', [43], Section 10.10). A further method, *re-inforcement learning*, provides only feedback that is positive for the correct actions (e.g. a decision for a correct category), and negative otherwise, but does not explicitly state why (does not provide the category label of the training image).

In categorization research, we also encounter the term *weak supervision*, which is related to object localization and discussed further in Section 2.4. Strong supervision provides training images, category labels *and* localization (delineation, segmentation or bounding box), whereas 'weak supervision' is defined as *not* requiring localization of the objects in the training images.

In pattern classification, the most common learning algorithms are the *maximum likelihood* and the *Bayesian* parameter estimation (see [43], Chapter 3). Both methods often produce nearly identical results, but their concepts differ.

Maximum likelihood tries to estimate fixed but unknown parameter values from the training examples, for example, the mean and covariance values of $\mathbf{f}_x$ for a specific category c_x with respect to all other categories c_m, $m \neq x$.

Bayesian estimation represents the parameters as random variables, assumes known prior distributions, and calculates posterior densities $P(c|I)$ based on the given training examples. We call $P(c_m)$ the *prior probability* that category m is being observed – for instance blue cars might be more likely to be encountered than pink ones. Furthermore, $p(I|c)$ or $p(\mathbf{f}|c)$ denotes the *category-conditional probability density* function[2]. In this context, I and $\mathbf{f}$ are now represented as random variables. If we search now for $P(c_x|I)$, we can use *Bayes formula*:

$$P(c_x|I) = \frac{p(I|c_x)P(c_x)}{p(I)}, \tag{2.2}$$

where $p(I)$ is just a scaling factor that guarantees $\sum_{m=1}^{M} P(c_m|I) = 1$:

$$p(I) = \sum_{m=1}^{M} p(I|c_m)P(c_m). \tag{2.3}$$

Bayes decision rule can be used to choose $c_x : x = \arg\max_m P(c_m|I)$ (see Equation 2.1).

Most learning algorithms require a time-consuming, intensive offline training phase and many training examples. Both may be prohibitive for visual categorization, in which the goals may be to learn from few or even one example, and to learn *online*, when a certain object is presented to the system. Another common difficulty is the re-training behavior of many learning algorithms, in which a new example requires a complete re-training of the whole system. *Incremental learning* algorithms deal with these problems and try to extend existing classifiers online when a new training example is presented.

The *Expectation-Maximization (EM)* algorithm is a popular variant of Maximum-Likelihood learning, which is well suited to visual categorization because it can handle missing information. This occurs quite often, for instance, when a part of an object is occluded or has been missed by segmentation or feature extraction processes. Generative probabilistic object models have successfully been learned with the EM algorithm by Weber et al. [204]. Extensions include the handling of varying scales [53], and the learning of new categories from single examples by integration of prior knowledge [46].

Several recent contributions explore the value of various learning techniques for categorization. Lowe uses a nearest-neighbor method [122], Agarwal and Roth [1, 2] use Winnow, and Dorko and Schmid [40] compare EM with support vector machines (SVM, see [200]). In our

[2] The exact notation would be $p_I(I|c)$ and $p_{\mathbf{f}}(\mathbf{f}|c)$ to specify that we address the density for a specific random variable.

own work [153], we learn a discriminative model and use Boosting as a learning technique very different from EM. Further principles that have been used in object recognition include PCA (principal component analysis), LDA (linear discriminant analysis), MDL (minimum description length), neural networks and genetic algorithms. Bekel et al. [16] present their compound 'VPL' approach that combines vector quantization (VQ), PCA and local linear maps (LLM). This is an attempt to achieve online adaptation for specific OR by online tuning of the LLM component.

I can recommend the following further reading on learning: Duda et al. [43] review most of the above listed techniques in the context of mostly statistical pattern classification. Hastie et al. [81] is a very complete reference to all state-of-the-art elements of statistical learning, while Vapnik's book [200] focuses strongly on support vector machines and related issues.

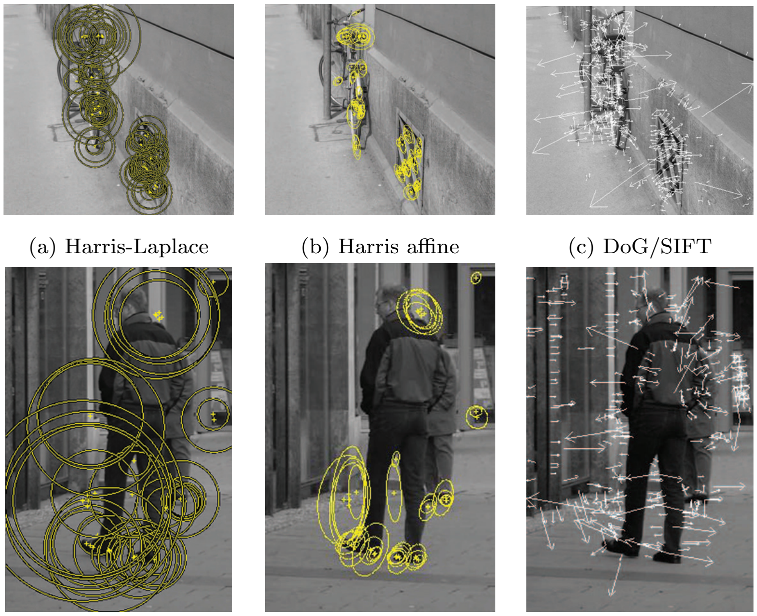
2.3 ... representation

How can visual object categories be represented? How can descriptions be extracted from a visual input event I? So far, we have discussed the pattern classification approach, where the representation is a feature vector $\mathbf{f}$, and a description of I is generated by calculating the individual features $f_1, \ldots, f_j$. This representation can cover many facets of objects in images: color, texture, homogeneous regions, discontinuities (edges, lines, corners). Even simple spatial relations may be modeled this way when features are derived from image coordinates. There are further, more complex parameters which can also be represented as features, including shape, topology and function of an object.

However, information is inevitably lost when a scene is projected to a 2D image because the observed world is 3-dimensional in space (and 4-dimensional in space and time). Thus, even the simplest geometrical features that can be measured in an image will change when certain parameters of the image capturing process are changing, for instance focal length or relative pose between camera and object. The same is true for radiometric distortions. The perceived color of a surface depends not only on its spectral albedo, but also on the color (spectral intensity distribution) of the illumination and on the spectral sensitivity of the sensor. These facts have motivated research into features that are *invariant* against all these sources of potential geometric and radiometric distortions. Especially *affine invariance* plays an important role in current categorization systems. When a surface can be assumed to be locally planar, the general case of perspective distortion is reduced to affine distortion, which is easier to model. Algorithms for extraction of affine invariant features include affine invariant moments of regions [61] and affine invariant salient points [134]. Color constancy algorithms [9, 10] have been developed to provide a certain degree of invariance to radiometric changes. Recent work discusses combined moment and color invariance [139].

A representation which is based on 'key features' typically extracts feature vectors by applying some 'interest operator' to the image. Early work on 'interest points' and saliency includes Marr's primal sketch [126] and several corner detectors [13, 62, 80, 92]. Among those, the Harris corner detector [80] is still quite popular and has recently been extended to cope with scale invariance [133] and affine invariance [12, 134, 196]. Other saliency detectors include the detection of corner orientation [30], a morphological approach to corner detection [99], saliency based on entropy [88] and the notion of 'maximally stable extremal regions (MSER)' [129]. There are surveys and comparisons for interest point detectors [171], scale and affine invariant detectors [136], and for affine region detectors [138]. Such an operator may find from several hundred up to several thousands of interest points per image, depending on the saliency threshold and on the content of the image. These operators deliver positions of corner-like or blob-like visual tokens and often include a measure of saliency (e.g. 'cornerness') and related parameters that are not sufficiently descriptive to produce a useful feature vector. Thus, additional *local descriptors* have to be extracted at the salient locations. Possible descriptors are calculated from a local support region and can for instance be moments of various order [98, 124] or intensity distributions [78]. The 'scale invariant feature transform' (SIFT) plays a special role, because this algorithm closely couples a difference of Gaussian (DoG) keypoint detector with SIFT as a local description method [122, 123]. A comparison of local descriptors is given by [135]. Figure 2.1 shows two example images (a bike and a person), with an overlay of detected key features, comparing scaled Harris, affine Harris, and DoG/SIFT detectors. We see that local saliency is well represented, including the scale of the features which is reflected by the size of the circles ellipses and arrows. For affine Harris and DoG/SIFT, there is also a 'directional' component that reflects the angle and orientation of a corner/key point (elongation and orientation of the ellipse, direction of the arrow). These examples also demonstrate that any salient patch will be detected, no matter if it is located on an object of interest or in the 'background'.

Local features have been used very successfully in the development of current categorization systems. Categorization from local features is one of the core topics and the various aspects of this approach are discussed throughout this article. The reader may refer to research



(a) Harris-Laplace (b) Harris affine (c) DoG/SIFT

(d) Harris-Laplace (e) Harris affine (f) DoG/SIFT

Fig. 2.1 Examples of 'key features' that are detected by: The scale invariant Harris detector ('Harris-Laplace' [133]), the affine invariant Harris detector [134], and the DoG/SIFT detector/descriptor [122]. It is evident from the examples, that these detectors respond to *any* salient feature – the features are not necessarily located on the actual object of interest.

that uses local features for specific OR [57, 105, 122, 133, 139, 175] and for categorization [1, 2, 40, 46, 51, 52, 53, 54, 82, 106, 107, 110, 172, 203, 204].

Following the pattern classification paradigm up to this point leads us to what has been called ‘discriminative paradigm’ in categorization. Based on training examples, we learn to decide c_x for a given input event I_y, based on $P(c_m|I_y), m = 1 \ldots M$. A typical *discriminative model* might model cluster centers as mean feature values and statistical dependence between features as covariance matrices that shape multivariate normal distributions. Decision boundaries can then be obtained using Bayes’ decision rule which will minimize the probability of error (see again [43], Chapter 2).

In the ‘generative paradigm’, a *generative model* that captures $P(I_y|c_x)$ is searched. Generative models can be built using the same interest points as described above for discriminative approaches. These models are built per class c_m. The ‘constellation model’ is the currently most prominent representative of a generative model for categorization. This model was initially presented by Burl et al. [26] and was extended by [53, 204]. Categories are modeled as joint probability densities of parts’ appearance (local salient patches as described above) and shape. Shape is represented by the mutual position of parts. Other approaches that model the geometry of salient parts include [52] and [106]. While the constellation [204] and the k-fan model [52] represent only a few particularly relevant parts, a codebook as used in [106, 107] can contain many local patches.

Holub and Perona [82] discuss the benefits of both, discriminative and generative appoaches. Generative models provide a number of advantages. Prior knowledge can be integrated, new categories can easily be added, many categories can be represented, and correspondences between object parts can be handled. Thus, generative approaches can deal with incomplete information (e.g. missing object parts). But because each model is created specifically for each category, discriminative techniques tend to achieve higher classification accuracies when similar categories have to be distinguished. Holub and Perona [82] present a combined framework that extends the generative constellation model, but uses a discriminative model to refine categorization results for classes with similar generative models. See also the work of Bar-Hillel et al. [8], who learn a generative appearance model in a discriminative manner. The discussion of discriminative vs. generative models has only recently gained a lot of interest, but on a broader and more historic scale, similar discussions have been carried out for the past two decades. Discriminative models are strongly related to 'bottom-up', data-driven methods, while generative models are linked to 'top-down' approaches. In the mid 1980s, the use of AI, especially by expert systems was a research topic in image understanding, and bottom-up vs. top-down as well as the integration of both paradigms ('bidirectional') was then discussed [41, 130]. The formation of geons as a qualitative representation using volumetric primitives [20, 39] can be viewed as a bottom-up data-driven process (edges → boundary groups → faces → primitives) and thus, fits into the 'discriminative model' paradigm. The same is true for other perceptual grouping approaches [120, 167]. Interestingly, Lowe's SCERPO system [120] includes a top-down verification step.

Of course, there are alternatives to an object representation by local patches and their spatial constellation. Parametric eigenspace has been proposed as another generative representation which is completely image-based [143, 193]. Training images of fixed size (e.g. $n \times n$ pixels) are treated as input vectors $\mathbf{x}$ with n^2 elements and subjected to principal component analysis (PCA[3]). This leads to a system $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m_x})$, where $\mathbf{m_x}$ is the mean of all input vectors $\mathbf{x}$. The $n^2 \times n^2$ coefficients of $\mathbf{A}$ have to be chosen such that the transformation yields a diagonal covariance matrix $\mathbf{C_y}$ which holds just the n eigenvalues of $\mathbf{C_x}$ in descending order. Now, a test image $\mathbf{x}_t$ can be projected into eigenspace, yielding $\mathbf{y}_t = \mathbf{A}(\mathbf{x}_t - \mathbf{m_x})$. If we use only a subspace of those k eigenvectors that correspond to the k largest eigenvalues, we have a $k \times n$ matrix $\mathbf{A}_k$ and we obtain a lossy reconstruction $\hat{\mathbf{x}}_t = \mathbf{A}_k^T \mathbf{y}_t + \mathbf{m_x}$. This means that we obtain a certain degree of generalization by reducing the dimensions of the eigenspace, while at the same time the mean square error between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is minimized. This method has gained wide interest, and PCA has been used in numerous

[3] see for instance [74], Section 11.4 'Use of Principal Components for Description'

recognition applications over the past decade. Issues of poor robustness, e.g. against partial occlusions, have also been addressed [109]. PCA is the most prominent representative of a number of global linear subspace methods like independent component analysis (ICA [85]) and linear discriminant analysis (LDA [128]), as well as nonlinear kernel methods, for instance SVM [200]. However, these image-based methods in general require a successful object to background segmentation and a proper brightness and scale normalization, which seems unrealistic in most categorization tasks. Thus, PCA and related methods have proven most effective for specific OR, and for well-controlled tasks of object pose determination. In general, there will be too much variance in appearance and shape of categories, so that model-based approaches as discussed above seem to be superior to image-based representations. But recent developments include also local applications of such methods to image patches [151] or local descriptors [91]. In terms of the discussion of generative and discriminative methods, PCA and ICA are generative methods, while LDA and SVM are discriminative.

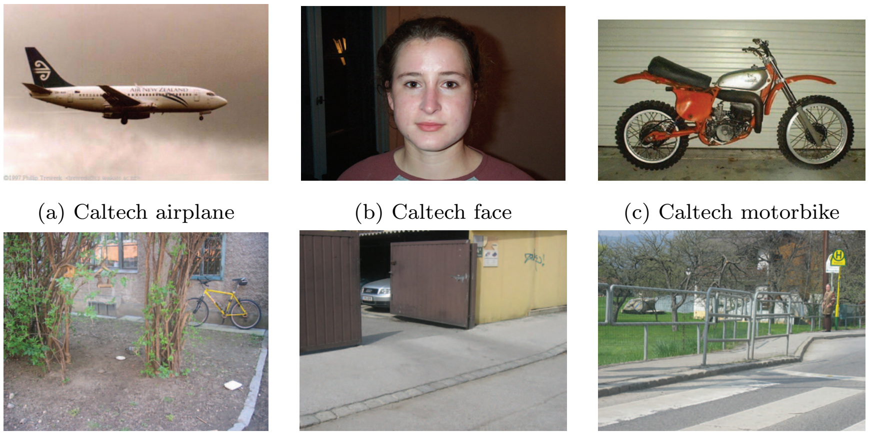
Several further schemes for object representations have been proposed. Aspect graphs[4] are a generalized, view-based method. The main idea is to combine slightly different viewing directions, in which the object looks alike, to one aspect. The object is represented by a number of aspects, a representation of these aspects and a graph that describes the possible transitions between them. Aspect graphs were used in the early 90s to recognize simple polyhedral objects [72], or objects which could be decomposed into generalized cones or geons [39]. We have proposed an extension of aspect graphs for object categorization [163] that combines aspects, CAD prototypes and a view-sphere. One advantage in using aspect graphs and related representations is that with the recognition of an object we not only know the object, but also its corresponding aspect. This aspect gives us an estimate of the viewing direction i.e. the relative pose between object and camera. In general, representations like aspect graphs require the successful segmentation of the image. Closed contours or homogeneous regions are needed to come up with contour-based (e.g. geons [39]) or region-based

[4] see e.g. [63], Chapter 20

[44] representations for categorization. Recently, this need for closed contours has been relaxed to a 'boundary-fragment-model' (BFM) [156] that represents a codebook of boundary fragments.

2.4 . . . localization

Under weak supervision, as introduced in Section 2.2, a categorization system should learn to categorize objects from examples of the form $< I, l >$, i.e. input *images* or sequences I, and corresponding labels l, but not *object locations* in the images. The images contain the desired objects, but not just the objects. Thus, the real degree of supervision in weakly supervised categorization depends very much on the complexity of the training examples. If the objects are shown prominently, without occlusions, and if the background is uniform, there is a very high degree of supervision. On the other hand, when the background is highly cluttered, and the objects are shown at smaller scales, in varying poses and partially occluded, supervision is weak. Figure 2.2 shows example images of both kinds taken from two image databases. The level of supervision is significant for the Caltech images, but supervision



(a) Caltech airplane

(b) Caltech face

(c) Caltech motorbike

(d) GRAZ02 bike

(e) GRAZ02 car

(f) GRAZ02 person

Fig. 2.2 Examples of different levels of supervision depending on object size, occlusion and background clutter. There is more supervision in the Caltech images than in the GRAZ02 images.

should be regarded as much weaker for the GRAZ02 images (see also the discussion on databases in Section 2.5 below).

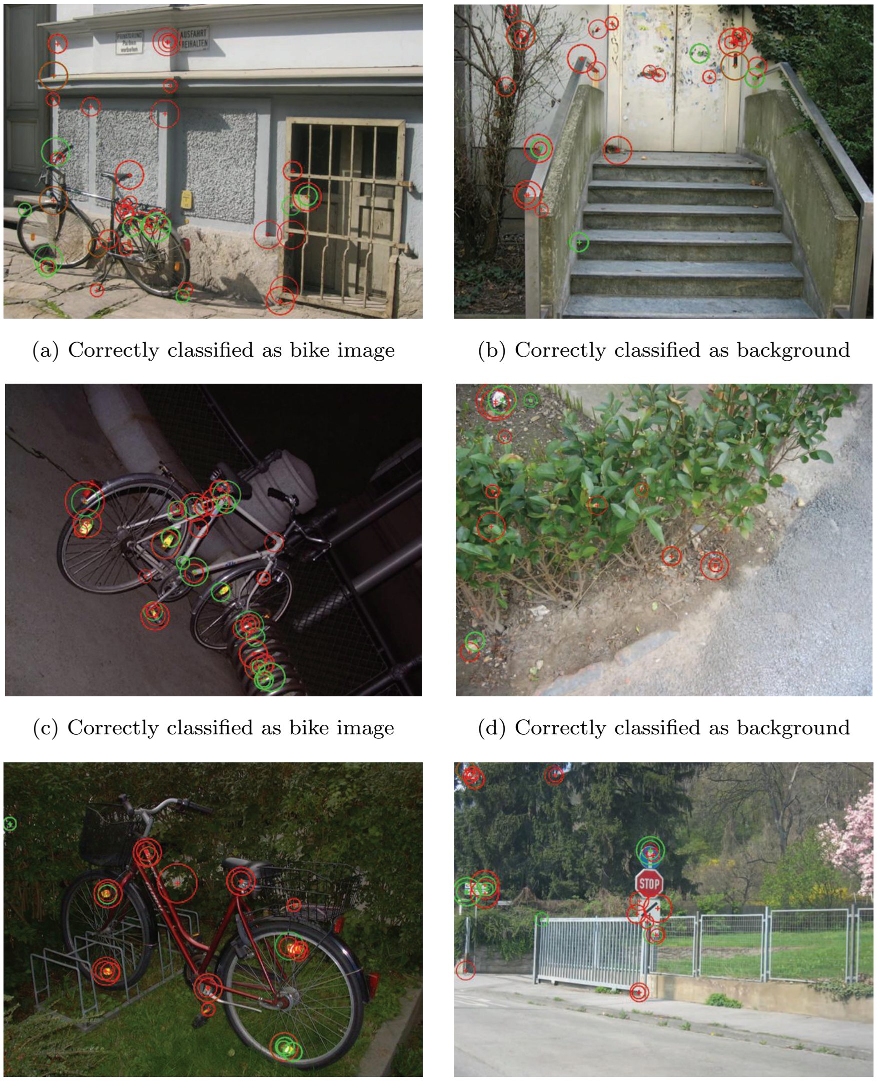
There are many contributions to weakly supervised categorization using representations that are based on local descriptors, and the training and test datasets are becoming increasingly difficult. Important questions are:

(1) Which local features are learned as category specific, discriminative features?
(2) Are the features actually located on the object or in the background?

One might even ask the question whether such systems really perform ‘object categorization’, or rather ‘image categorization’, because a high percentage of relevant patches is located in the background. An image of a car might, for instance be categorized based on a few ‘car-features’ (license plate, lights, wheels, etc.) but many visual ‘background’ clues (pavement, street signs, and signals). Figure 2.3 illustrates these aspects for the categorization of bike images vs. ‘background’ images (containing no bike).

There has been less research on localization abilities than on recognition rates. Several approaches deal with a tradeoff between categorization with low supervision and localization performance with higher supervision (e.g. [29, 54, 191]). Other approaches are very good in localization, but just for specific OR (e.g. [105, 166]). We presented a weakly supervised categorization system [153] that can handle rather complex images. To evaluate its localization abilities, we set up a new database and proposed new localization measures [154]. In this systematic evaluation of localization, we discuss requirements for datasets (see Section 2.5) and give experimental results on localization abilities. We find that many descriptors are located in the background, even when a balanced dataset is used. Furthermore, localization performance is class-dependent. Localization on our own database is, for instance, better for bikes than for cars or persons.

One can conclude that many systems which just use local features and do not account for their spatial relationship mostly categorize images rather than objects. They cannot reliably tell us *where*



(a) Correctly classified as bike image

(b) Correctly classified as background

(c) Correctly classified as bike image

(d) Correctly classified as background

(e) Incorrectly classified as background

(f) Incorrectly classified as bike image

Fig. 2.3 These images show all significant local descriptors that were detected while trying to categorize bike images. Those descriptors that are above a threshold for bike detection are shown in green, descriptors below a threshold in red. We observe a number of typical aspects of weakly supervised 'image' categorization which is based on local features and does not account for their spatial relationship: for bike images, some features are on the object, others are in the background; for background images, some 'bike' features are found; a sufficient amount of 'bike' features can result in a false positive (f).

the object is (in space and time). Thus, approaches are required that also model geometric and spatio-temporal relations in addition to local features. Such approaches have been successfully used with strong supervision (constellation [204], k-fan [52], and boundary fragment models [156], codebooks [106, 107]). There is room for challenging future research that will combine local features, spatio-temporal relations and weak supervision. First attempts towards completely unsupervised categorization have even been reported [184].

2.5 ... datasets

It is obvious that common datasets are required for the comparison and evaluation of categorization algorithms. We have already discussed several aspects of datasets for categorization. Such datasets should provide many images per category. They should cover the high intra-class variability within a category. They should contain examples of low inter-class variability between visually similar categories. *Ground truth* should be provided in terms of category labels l per input event I, but also for the localization of the object within the image. Localization ground truth could be a bitmap, a coordinate (e.g. center of gravity) and an approximate extension (e.g. radius) of the object, a contour or a bounding box. Certain approaches (e.g. [1, 2, 204]) benefit from a number of 'simple' training examples, in which the object of interest is shown prominently, with no occlusion and with homogeneous background. Weakly supervised approaches (e.g. [153]) call for more complex examples that contain the objects of interest on smaller scales, partially occluded, in unfamiliar poses, with varying illumination and with strong variations in background and clutter.

Several databases have been widely used to compare research results in specific OR, and in categorization. Popular databases for specific OR are the FERET database[5] for face recognition [162] and the COIL database[6] for simple everyday objects that are placed on a turntable and shown in many poses. A number of challenging databases for

[5] *http : //www.itl.nist.gov/iad/humanid/feret/feret_master.html*

[6] *http : //www1.cs.columbia.edu/CAVE/research/softlib/coil − 100.html*

categorization have been made available by the PASCAL project[7]. Well-known databases used for categorization experiments include:

- The ETH-80 database[8], introduced by Leibe and Schiele [108], showing 8 different categories in a controlled setting. The objects are placed on a turntable with homogeneous background, and images are taken from 41 equally spaced viewpoints of the upper viewing hemisphere.
- The Caltech database[9], used e.g. by Fergus et al. [53] showing cars (rear views), airplanes and motorbikes (side views), human faces (frontal views), and leaves. In general, the objects of interest are shown prominently and in very similar poses, with little or no background clutter.
- The UIUC 'cars side' database[10] containing side views of cars and background images [1, 2].
- The TU Darmstadt (formerly the ETH Zurich) database[11], showing side views of cows, cars and motorbikes, used by Leibe et al. [106].
- Our own databases[12] GRAZ01 (people, bicycles, counterexamples) and GRAZ02 (people, cars, bicycles, counterexamples) used in [153, 154].

Figure 2.4 shows a few example images from these image databases for categorization.

The discussion in Section 2.4 showed that recognition is often based on many local features, some of them located on the object, others in the background. If the goal is to recognize objects, independent of background and context, it would be desirable to have a high percentage of local descriptors on the object. It may, however, make sense to use context information. Context may help to estimate prior probabilities $P(c)$. It is, for instance, more likely to find cars on roads, and flowers in

[7] *http : //www.pascal − network.org/challenges/VOC/*

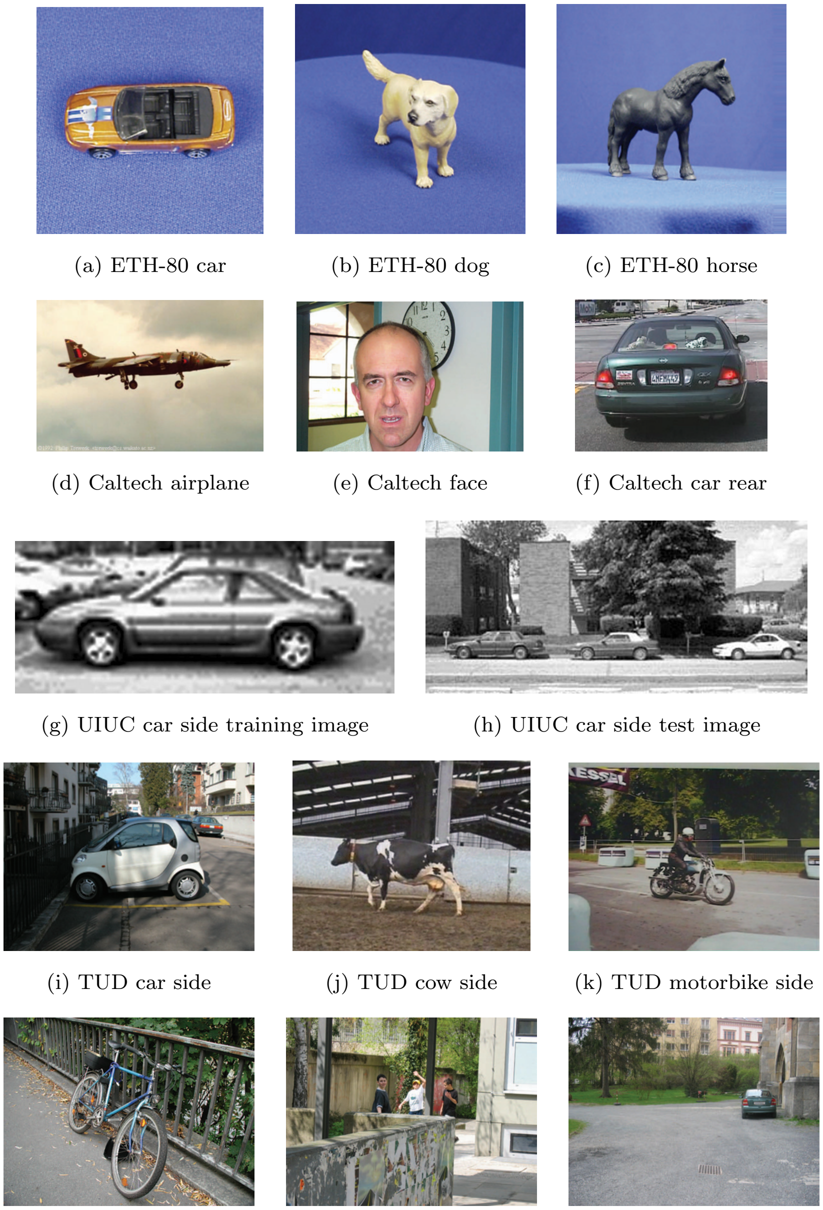
[8] *http : //www.vision.ethz.ch/projects/categorization/eth80 − db.html*

[9] *http : //www.vision.caltech.edu/html − files/archive.html* or *http : //www.robots.ox.ac.uk/ vgg/data3.html*

[10] *http : //l2r.cs.uiuc.edu/ cogcomp/Data/Car/*

[11] *http : //www.pascal − network.org/challenges/VOC/databases.html*

[12] *http : //www.emt.tugraz.at/ ∼ pinz/data*



(a) ETH-80 car (b) ETH-80 dog (c) ETH-80 horse

(d) Caltech airplane (e) Caltech face (f) Caltech car rear

(g) UIUC car side training image (h) UIUC car side test image

(i) TUD car side (j) TUD cow side (k) TUD motorbike side

(l) GRAZ01 bike (m) GRAZ02 person (n) GRAZ02 car

Fig. 2.4 Example images from the ETH-80, Caltech, UIUC, TU Darmstadt (TUD), and TU Graz (GRAZ01 and GRAZ02) databases.

a meadow, than vice versa [144]. Careful database design is required. Either certain backgrounds are strongly linked with a certain category (e.g GRAZ01), or the database needs to be balanced with respect to background, so that similar context is shared by several categories in the database (e.g. GRAZ02).

The repeatability of experiments is another very important aspect in the transparent design and use of databases. It should be made explicit which images were used during training and tests to achieve a certain categorization performance.

It will be very difficult to come up with an image database for many categories that can really cover all of the above requirements. There are algorithmic workarounds, e.g. algorithms that try to learn from very few examples [46][13]. And there are recent proposals to use the overwhelming amount of images that are available online, e.g. within Google's image search [54]. While this is a very appealing idea, the results of such a search contain many outliers, and careful training procedures are required [55].

2.6 ... evaluation

How should the performance of a categorization system be evaluated? There are several aspects that should be covered. First of all, one is interested in *recognition rates*: Presenting a number of test inputs I, what is the percentage of correctly categorized images? But there are other, more subtle evaluations. What are the rates of *false positives* – images that are classified to belong to a certain category, but do not show this kind of object – and *false negatives* – images that show a certain object, but are not recognized to belong to this category. There are several ways to evaluate this aspect of recognition performance. When a complete categorization system is available that can cope with any input I and decide on its category c_m, $m = 1 \dots M$, one can build a *confusion matrix*. This matrix represents in each of its rows j, how many examples of c_j were categorized to belong to $c_1, c_2, \dots, c_m$. Categorization is perfect when there are only entries in the main diagonal.

[13] see $http : //www.vision.caltech.edu/feifeili/101\_ObjectCategories/$ for the database of 101 objects that was used for these experiments

Significant numbers in entries other than the main diagonal point out that a certain category tends to be confused with an other one. To give an example: High values in row j, columns j and k tell us, that while category c_j is often recognized correctly, it is also hard to distinguish from category c_k. In recent categorization research, Fergus et al. [53] present a confusion matrix for the categories motorbikes, faces, airplanes and spotted cats using their own constellation model.

In many cases it will be difficult to obtain a complete evaluation in terms of a confusion matrix, because individual classifiers have been trained for each category, using examples showing objects which belong to this category, and counterexamples that do not contain any object of the category in question. The individual classifiers are then trained to distinguish between exactly two categories: object c_1 and counterclass c_2. Then, the following four cases can occur during the process of categorizing an input I to belong either to c_1 or c_2: hit (true positive), false positive, miss (false rejection) and correct rejection. Rates are required to normalize these quantities:

$$\begin{aligned} p(\text{true positive}) &\sim \text{positive detection rate} \\ &= \frac{\text{number of true positives}}{\text{total number of positives in the dataset}} \end{aligned} \tag{2.4}$$

$$\begin{aligned} p(\text{false positive}) &\sim \text{false detection rate} \\ &= \frac{\text{number of false positives}}{\text{total number of negatives in the dataset}} \end{aligned} \tag{2.5}$$

The *receiver-operator-characteristic (ROC)* is a common way to model the *discriminability* of c_1 and c_2 (see e.g. [43], Section 2.8.3) for a given classifier. An 'ROC curve' plots positive detection rates against false detection rates and is obtained by varying a parameter that changes the relation between true and false positives. Many recent publications use ROC curves (e.g. [26]), the area under the ROC curve (e.g. [204]) and ROC equal error rates (e.g. [53]) to present their categorization results. ROC equal error rate is defined as the point on the ROC curve, where $p(\text{true positive}) = 1 - p(\text{false positive})$.

While the receiver operator characteristic is a good tool for describing discriminability in recognition tasks, *recall precision curves (RPC)* are better suited to characterize *localization*, especially for object detection systems that use local patches. This situation is well described in [1, 2]. Here the task is no longer to categorize an image, but to decide, whether a certain local patch belongs to an object or not. Usually, there will be many salient patches found in an image, and often the majority of the patches will be located in the background. Recall and precision are defined as follows:

$$\text{recall} = \text{positive detection rate} \tag{2.6}$$

$$\begin{aligned}&\text{precision}\\&\quad = \frac{\text{number of correct positives}}{\text{number of correct positives} + \text{number of false positives}}\end{aligned} \tag{2.7}$$

The RPC curve plots recall against (1 − precision):

$$\begin{aligned}&1 - \text{precision}\\&\quad = \frac{\text{number of false positives}}{\text{number of correct positives} + \text{number of false positives}}\end{aligned} \tag{2.8}$$

Thus, RPC can be used to evaluate performance with respect to localization of individual patches. Agarwal and Roth [1, 2] also discuss the difficulties of evaluating and comparing object localization results. They propose a method of manually collecting ground truth (rectangular 100 × 40 windows around a car viewed from the side) and they compare the results of their car detector with this ground truth (in terms of positional displacement and overlapping area thresholds). We adopted a similar approach in [154], but a general method for the comparison of localization abilities of a certain categorization system depends on many aspects and is hard to define. This includes the question of ground truth representation (bit map/bounding box/contour), as well as category-specific issues (should the background that can be seen through the wheel of a bicycle be counted to belong to the object or not?).

Further methods have been proposed to quantify localization performance. Dalal and Triggs [36] detect humans and use *detection error tradeoff (DET)* curves, where they plot miss rates (1 − recall) versus

FPPW. FPPW stands for 'false positives per window', which is a well-suited measure for methods which use a sliding detection window. Leibe et al. [108] also work on pedestrians and apply three evaluation criteria: relative distance, aerial coverage and overlap of bounding boxes.

2.7 ... system integration

Having discussed these numerous issues of categorization in Sections 2.1–2.6, it is quite obvious, that building a categorization system will require many different components: Image or video acquisition and databases, image processing, feature detection and extraction, learning, pattern recognition and classification. Of course it has been a substantial effort in software engineering to make all these components work together to be able to conduct significant categorization experiments. But this section on issues of system integration should point out other, less obvious aspects.

There is a strong argument in favor of *embodiment* of cognitive systems. This starts with Brooks' 'building brains for bodies' paradigm [25] and has also been advocated within more recent research in the European 'Cognitive Vision' and 'Cognitive Systems' program of the 5^{th} and 6^{th} framework program [75, 202]. When it is possible to close the cycle of perception and cognition by action, actions can be used to verify the cognitive abilities of a system. Recent research programs aim, for example, at the embodiment of cognitive vision in autonomous robotic platforms[14]. This goal of embodiment is very ambitious and will definitely not be reached within the next few years of research, although there are several interesting research platforms, even of humanoid robots[15], that suggest humanlike behavior. Current cognitive performance is *very* limited. As an example, the European cognitive vision project 'Actipret'[16] resulted in an integrated system that understood scenes in which human operators were loading a CD into a CD player.

The role of *context* has already been mentioned above. In terms of system integration, spatial and temporal context should influence the

[14] e.g. the European project CoSy, see $http : //www.cognitivesystems.org/$

[15] e.g. Honda's Asimo $http : //world.honda.com/ASIMO/$

[16] $http : //actipret.infa.tuwien.ac.at/$

behavior of an integrated cognitive vision system. Which sequence of actions has led to the current situation? How has a certain strategy been used to solve a certain problem in the past? There is a trade-off between short- and long-term visual memory. Which information is required at which level of abstraction? At which point in time can certain levels of visual detail be forgotten? Humans learn from examples. What is the role of visual *episodes*? Is there a kind of episodic memory? These and similar questions have been researched under the term 'Visual Active Memory Processes' within the European Cognitive Vision Project 'Vampire'[17], and a prototype of visual active memory that can hold information about visual events and episodes at different levels of abstraction has been built [77].

As a final remark, analysis of visual structure is related to analysis of *motion*. There is experimental evidence for joint activations of primary visual areas (V1,V2,V3,V4) and other areas that are responsible for disparity, speed, direction of motion (MT), e.g. in functional MRI of monkeys [119]. There are action-related representations, and motion analysis plays an important role. Experimental evidence has also been found in the recognition of 3D objects, indicating that three-dimensional shape representation couples 3D structure from motion with stationary visual cues [176]. This leads us to the idea that an artificial categorization system might benefit from integration of image-related (2D) information with 3D spatial and 4D spatio-temporal relations.

[17] $http://www.vampire-project.org/$

3

Building Blocks for Categorization

This section discusses a number of topics in detail. While these topics are quite different and only loosely coupled, they have to be considered the most important building blocks for visual categorization systems. Each of the sections may be consulted individually, and not all components are required to build a working categorization system.

3.1 Scale in space and time

The ability to deal with spatial scale will undoubtedly be required by any visual categorization system. In addition, temporal scale can be important, when categorization is performed not only for still images, but also for image sequences. As a first step, we have to relate the size of the pixels of the digital image to spatial extents in the scene. While this is rather easy for approximately planar scenes observed from a perpendicular viewing direction (e.g. in remote sensing or in microscopy images), this task is much harder for three-dimensional scenes and general perspective projection. The notion of scale in vision has been researched from several directions and in detail. It is related to the concept of spatial resolution of an image and temporal resolution of a sequence. Both are well grounded in digital signal processing and the sampling theorem[1] tells us the smallest spatial/temporal detail that can be reliably resolved with respect to the image plane/frame rate of the camera.

The image itself represents the maximum spatial resolution that can be achieved for a given constellation of camera and scene. Based on this maximum resolution, several kinds of *multiscale* or *multiresolution* representations of an image have been proposed.

3.1.1 Image Pyramids

Using common language, an *image pyramid* [164] can be seen as a hierarchical pictorial representation (a stack of image layers) of a certain visual content at different spatial resolutions. The architecture of a *regular* image pyramid can be specified by three properties. The *reduction factor* denotes the change in image size between two successive layers of the pyramid. The *reduction window* specifies how many children in a layer contribute to the calculation of one parent pixel in the next layer. The *reduction function* specifies the algorithm that should be applied to the reduction window. This can be the average, a convolution with a Gaussian, a minimum, maximum or median operator, to give a few examples. Using this terminology, the architecture of the regular pyramid can be specified by 'reduction window/reduction factor', for instance a '$2 \times 2/4$' pyramid, and the construction algorithm for the pyramid is defined by the reduction function. Many applications of regular image pyramids were proposed in the 1980s. For instance, Gaussian and Laplacian (a bank of bandpass filters) pyramids have been widely used in image encoding, event detection and object tracking [6, 27].

Are image pyramids a well-suited representation for categorization? At fist glance, the answer might be yes. Since the *scale* of an object – its size in terms of image pixels – will depend on focal length, resolution of the imaging sensor and distance between camera and object, a pyramid could be used to select the appropriate layer that shows the

[1] The maximum frequency f_0 that can be reliably detected requires more than the double temporal sampling Δt of the according signal [178]. The same applies to the smallest spatial detail Δx and the according maximum spatial frequency μ_0.

object at a specific, preferred, 'dominant' or 'canonical' scale. However, a closer examination reveals, that a representation in a pyramid as in the earlier description (e.g. $2 \times 2/4$, averaging) is not invariant with respect to rotation and translation. Especially the shift-variance has been critically examined (see Bister et al. [22]). Overlapping reduction windows can be used as a remedy against shift-variance (e.g. in a $4 \times 4/4$ architecture), so that every child pixel contributes to the information of more than one parent in the next layer. In a reduction window, it might also be possible to compute features which are invariant with respect to rotation, and even to affine distortion (see Section 3.2).

Still, the representation in any of the pyramid layers is pixels, not objects. But so far, we have considered only regular pyramids. Pioneered by the work of Meer [131] and Montanvert et al. [140], the idea of *irregular pyramids* has been researched. The original goal was to represent homogeneous regions as one node in a planar graph structure. The nodes would correspond with an irregular tessellation of the pixels of the original image. Several layers of planar graphs could be connected to form an irregular pyramid. While this approach was originally targeted at new segmentation algorithms, an irregular pyramid could also be viewed as a proper representation for categorization, in which one node might represent an object and the neighboring nodes the background/context of the object. Details about the object could be found in the children of this node (traversing down through the irregular pyramid to obtain more level of detail for an object).

3.1.2 Scale Space

In image pyramids, there are discrete layers which correspond to specific, discrete spatial resolutions or spatial scales. The notion of a continuous *scale* parameter t has been approached in a general manner in *'scale space theory in computer vision'*. The idea of scale in signals and scale-space filtering of one-dimensional signals dates back to Witkins work in 1983 [206] and was further investigated by Koenderink [93], and by Yuille and Poggio [209]. The scale space is represented as a family L of signals which are derived from the original signal f at scale $\sigma = 0$.

We can define a scale-space family $L : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ for N-dimensional signals $f : \mathbb{R}^N \rightarrow \mathbb{R}$:

$$L(\cdot;\sigma) = g(\cdot;\sigma) * f(\cdot), \tag{3.1}$$

where $\cdot$ denotes an N-dimensional vector $\mathbf{x} = (x_1, \ldots, x_N)^T$, and the N-dimensional Gaussian kernel $g : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is defined as

$$g(\mathbf{x};\sigma) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp - \frac{x_1^2 + \ldots + x_N^2}{2\sigma^2}. \tag{3.2}$$

Babaud et al. [7] proved that the Gaussian kernel is unique for scale-space filtering. It is the only kernel which guarantees a 'monotonic' smoothing of the original signal f. This means that the number of extrema of f is monotonically increasing/decreasing when moving to lower/higher scales. Furthermore, they show, that when moving from higher to lower scales σ, new extrema may emerge, while existing extrema will be preserved (although their position $\mathbf{x}$ can vary with changing σ).

For spatial scale in images, we face a 2-dimensional scale space with $N = 2$. Of course, we have to drop the idea of a continuous 2-dimensional signal f and of a continuous scale σ when it comes to discrete digital images, and when tractable computations of scale-space call for discrete scales. But the notion of $L(\cdot;\sigma)$ allows to select and to compute any desired scale σ, which is a major difference to the previously discussed pyramid structure which represents only a very limited number of fixed scales. In [111], Lindeberg developed the *discrete scale-space representation* which uses a discrete analogue of Gaussian for smoothing and preserves all the convenient properties of continuous Gaussian scale-space. For further reading, scale space theory has been discussed in detail by Lindeberg [112], by Florack et al. [60], and in [189].

Let us now focus on aspects of scale-space theory that are relevant for categorization. In general, objects will appear in images at varying scales. It would be very desirable to be able to calculate a *'characteristic scale'* for any kind of image event, covering edges, corners, blobs, and objects. At this scale, the required amount of smoothing should cancel superfluous level of detail but preserve the significant visual information about the object (its shape, color, texture, connectivity and topology). If it were possible to extract any object at its characteristic scale, and if the spatial resolution of the image could be adapted to this scale (e.g. by subsampling as for pyramids), we would obtain a *scale normalization* for the pixel matrix representing the object[2]. In general, it will be necessary to detect and to track salient image events over several scales to establish relations between salient structures of different scales. This concept has been researched under the term *deep structure* of images by Koenderink [93] and by Lindeberg [112], and it has led to the idea of extracting image features using *scale selection*. Since this feature detection with automated scale selection [115] turns out to be a crucial technique used in current categorization systems, we shall discuss it here in some detail, closely following [115].

Blobs are probably the most obvious features to be usefully represented in Gaussian scale-space. A 'blob' is defined as a compact, homogeneous region in an image. With increasing smoothing, smaller blobs will disappear or merge with neighboring structures to represent a larger blob. Imagine for example the individual keys of a phone or of a pocket calculator (individual, small blobs at lower scales), that might merge into one larger blob representing the keypad at a higher scale.

Other features like edges, lines and corners are related to spatial derivatives. In general, we denote the *scale-space derivative* by

$$L_{x^\alpha}(\cdot;\sigma) = \frac{\partial^\alpha}{\partial x_1^{\alpha_1}\ldots\partial x_N^{\alpha_N}} L(\cdot;\sigma) = (\frac{\partial^\alpha}{\partial x_1^{\alpha_1}\ldots\partial x_N^{\alpha_N}} g(\cdot;\sigma)) * f(\cdot), \quad (3.3)$$

where $\alpha = \sum_{i=1}^{N} \alpha_i$. We can simplify this notation for 2 dimensions (x,y) of an image, and give some examples: L_x and L_y denote the gradients of L in x and y direction, and L_{xx} denotes the second order partial derivative of L in x direction. Blobs can also be detected by scale-space derivatives because a blob will essentially degenerate into a

[2] The characteristic scale might vary for several objects in one image, and so might the scale/subsampling of individual image portions containing these objects at their respective characteristic scales. This again is a major difference to regular image pyramids that contain layers of images at certain, but fixed, scales and subsampling factors.

point-like structure at high scale, so that it can be detected as a local extremum of the 2-dimensional Laplacian:

$$\nabla^2 L = L_{xx} + L_{yy}. \tag{3.4}$$

Smoothing with Gaussian kernels guarantees that with increasing scale no new extrema will be generated, and that existing extrema will not be amplified. From this property follows directly, that the amplitude of scale-space derivatives must decrease with increasing scale. The goal of automatic scale-selection, however, is to find a specific scale σ_0 for a certain feature that characterizes this feature best. How can a significant scale for a certain feature be detected? The solution is to introduce *normalized coordinates* $\xi = x/\sigma$. It can be shown, that for these normalized coordinates, the scale-space derivatives will assume local maxima. According to Lindeberg [115]:

> "*Principle for automatic scale selection:* In the absence of other evidence, assume that a scale level, at which some (possibly nonlinear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in data."

Following this principle, automatic scale selection is possible and leads to a certain location (x_0, y_0) and scale σ_0 that characterizes a specific, salient image event. Lindeberg has discussed this and similar scale selection principles for blobs [115, 118], interest points and corners [115], and edges and ridges [114, 115]. His experimental results are quite impressive and strongly suggest that feature detection by scale selection in scale-space is a powerful tool for many cognitive aspects in computer vision[3], including attention, object representation, recognition and categorization. As a clear consequence, the detection and analysis of scale and affine invariant interest points has become a research topic in categorization ([12, 88, 123, 129, 133, 134, 136, 171, 196], see Section 3.2 for a detailed discussion).

[3] Lindeberg [115] also points out neurophysiological evidence for receptive fields in retina and visual cortex of mammals whose response can be modeled by Gaussian derivatives [208].

When we consider categorization from image sequences instead of still images, it is a quite natural idea to try to extend spatial scale-space towards a *spatio-temporal scale-space* representation. This idea has been pioneered by Koederink [94] and has drawn recent attention with significant contributions by Lindeberg [113, 116], and with the development of spatio-temporal interest points [100, 101, 103, 117]. A straightforward extension of the above notation of $L(\cdot;\sigma)$, following [101] models a spatio-temporal image sequence[4] as a function $f(x,y,t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ and constructs its linear scale-space representation $L(x,y,t;\sigma,\tau) : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2_+ \rightarrow \mathbb{R}$ by convolution with an anisotropic Gaussian kernel with spatial variance σ^2 (spatial scale parameter σ) and temporal variance τ^2 (temporal scale parameter τ):

$$L(x,y,t;\sigma,\tau) = g(x,y,t;\sigma,\tau) * f(x,y,t), \tag{3.5}$$

where the spatio-temporal Gaussian kernel is defined as [101]

$$g(x,y,t;\sigma,\tau) = \frac{1}{\sqrt{(2\pi)^3\sigma^4\tau^2}} \exp(-\frac{x^2+y^2}{2\sigma^2} - \frac{t^2}{2\tau^2}). \tag{3.6}$$

While this extension seems straightforward, it assumes symmetry not only in the spatial, but also in the temporal domain, which is not true for online processing of image sequences (the future can not be accessed to smooth symmetrically in temporal dimension). But the assumption will hold for reasonably long video sequences that are processed offline. It has been found that significant scales and events can be detected in spatio-temporal data in a very similar manner as for spatial scale-space. As an example, Laptev and Lindeberg showed that space-time interest points do characterize significant events in image sequences and can be efficiently used to model human motion patterns [100, 101] and to recognize human actions [102].

3.1.3 Discussion

The previous subsections have stressed the importance of dealing with scale in space and time to represent salient structures for categorization. Such structures might be whole objects (if they can be perceived as

[4] with spatial parameters (x,y), and temporal parameter t.

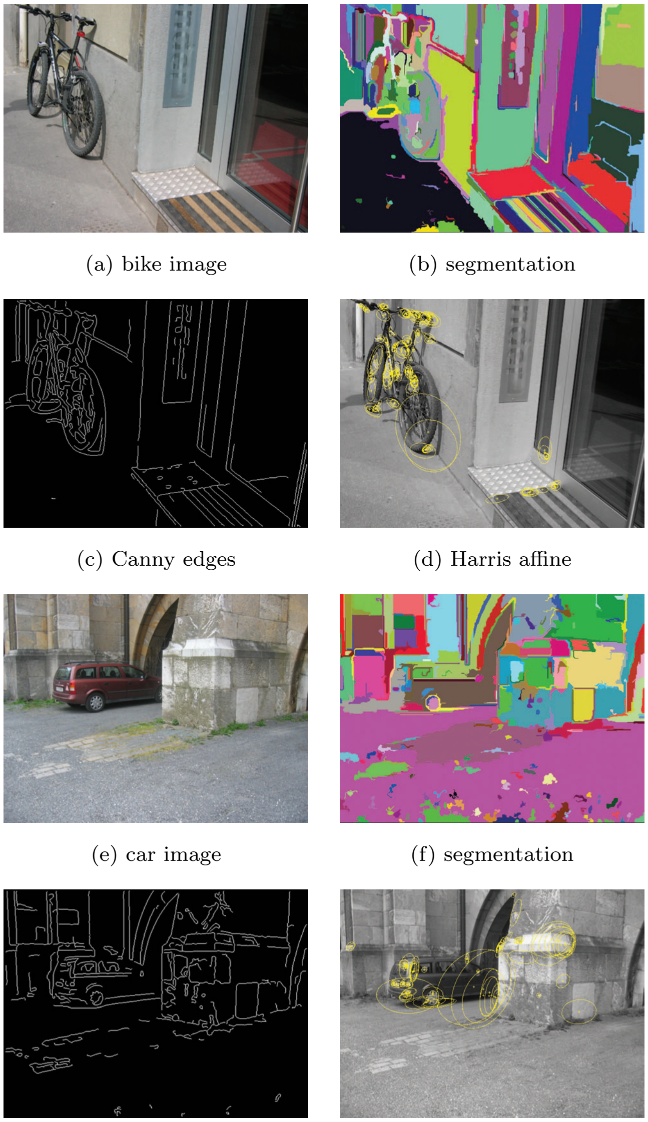
homogeneous blobs with sufficient contrast to the background), object parts, or ‘interest points’ which describe a very local but salient feature at its significant scale. There are of course further ways to look at scale in spatial and temporal signals, Fourier analysis probably being the most obvious one. Let us just state that in comparison to pyramids and scale space, the Fourier spectrum can reveal *which* significant frequencies occur in a signal, but it will not tell us *where* they are located in the image, which is a drawback when objects should not only be categorized but also localized. The same is true for banks of Gabor filters, which also operate on the Fourier spectrum. In comparison, the Wavelet transform offers localization similar to pyramids, but it also lacks continuous scales.

3.2 Saliency, key-features, points and regions of interest

In Section 2.3, we already briefly discussed the need for detectors of salient points (‘interest operators’). Learning and classification algorithms require rich descriptions of such points and their surrounding regions in terms of feature vectors. The following two subsections provide an overview of existing methods for *detection* and *description* of saliency, key-features, points and regions of interest. While many detection and description methods have been used for decades, object categorization has shed a new light on these algorithms, and there has been a significant number of new contributions and of review articles in the past few years. For summaries, comparison and evaluation we refer the reader to [171] (evaluation of interest point detectors), [138] (comparison of affine region detectors), and [137] (performance evaluation of local descriptors).

3.2.1 Detectors

Detectors can help to reduce the amount of data to be processed, focusing attention on salient image events as salient points, lines/edges, and regions of homogeneity. Figure 3.1 shows two examples of segment, edge and point detection for a bike and a car image. We see that quite different aspects may be emphasized by the three methods.



(a) bike image

(b) segmentation

(c) Canny edges

(d) Harris affine

(e) car image

(f) segmentation

(g) Canny edges

(h) Harris affine

Fig. 3.1 Saliency can be extracted from images by segmentation of regions, edge/line detection, and by salient point (corner or blob) detectors. For these illustrations, we used the graph based segmentation method by Felzenszwalb and Huttenlocher [50], Canny edges, and the Harris affine corner detector by Mikolajczyk and Schmid [134].

Region detectors: In general, some kind of homogeneity criterion, or 'segmentation cue' is required to find homogeneous regions by *segmentation* algorithms. Basic segmentation algorithms that can be found in any textbook on image processing will typically range from simple thresholding to region growing and split-and-merge algorithms. More advanced, 'high level segmentation' techniques have been developed, including mean-shift [32], normalized cuts (n-cut [179]), and level-set segmentation [158]. Of course, one can switch between segmented regions and their closed boundaries, as is done, for instance, with the boundaries of the level set function to obtain geodesic contours [160]. For example, the results of normalized cuts segmentation have been used as tokens for the categorization approach by Duygulu et al. [44].

Edge/line detectors: Edges can be detected as locations in the images with a significant gradient in one direction. More formally, we can search for local extrema of the image gradient ∇I, or for zero crossings of the second derivative ΔI. Lines can be seen as ridges or valleys in the greyvalue image, and represent already local extrema (with respect to a gradient profile direction p perpendicular to the line). They can be detected as zero crossings of the first derivative $\partial I/\partial p$. Numerous approaches to edge detection have been proposed, including Sobel, Canny, and LoG/DoG edge detection. In most cases, edge detectors will not provide closed contours around an object of interest. Thus, contour-based categorization algorithms will have to deal with contour or boundary fragments (see e.g. [156, 181]).

Salient point detectors: Salient points can be defined as locations in the images with significant change in *more than one* direction. In [171], Schmid et al. distinguish between contour based methods (maximum curvature, junctions, crossings), intensity based methods (e.g. significant gradients in two directions), and parametric model based methods (e.g. analytic junction models, deformable templates). Salient points might be sharp corners where two straight edges meet, isolated pixels in a homogeneous background, but also smoother corners (a circular arc connecting two straight edges), circular discs, or blobs, depending on the 'dominant scale' of the salient structure.

Subsequently, we present a variety of saliency detectors and briefly discuss their properties. Where derivatives of the input image are required, directional derivatives in x and y direction are denoted by I_x, and I_y, respectively. I_x^2 denotes the matrix product $I_x I_x$, whereas second derivatives are denoted by I_{xx}, I_{xy}, and I_{yy}. Figure 3.2 at the end of Section 3.2.1 compares the results of nine of the subsequently mentioned saliency detectors applied to a car image.

3.2.1.1 Using first derivatives

Autocorrelation: Several corner detectors have been proposed that are related to autocorrelation of the 2D image signal. $\mathbf{M}$ denotes the autocorrelation matrix (sometimes also called second moment matrix $\boldsymbol{\mu}$):

$$\mathbf{M} = \boldsymbol{\mu} = \begin{pmatrix} \sum_{(x_w,y_w)\in W} I_x^2 & \sum_{(x_w,y_w)\in W} I_x I_y \\ \sum_{(x_w,y_w)\in W} I_x I_y & \sum_{(x_w,y_w)\in W} I_y^2 \end{pmatrix} \tag{3.7}$$

If the rank of $\mathbf{M}$ is two (i.e. both eigenvalues are large), then there are gradients in more than one direction within the local neighborhood W. For edges (significant gradient in one direction), the rank of $\mathbf{M}$ will be one, and zero for homogeneous regions. Several variants of corner detectors have been proposed that are based on these observations. The KLT tracker[5] tracks points of interest which are detected as locations with two large eigenvalues of $\mathbf{M}$ [180]. Förstner has presented several variants of an interest point detector that uses $\mathbf{M}$ (see [62], and also Chapter 16.4 of [79]).

Harris corners: Harris and Stephens [80] have published the probably most popular variant of a corner detector based on the second moment matrix. They enhance the approach described above in several ways. Instead of using the sum $\sum_{(x_w,y_w)\in W}$, they convolve the local derivatives I_x and I_y with a Gaussian G. They introduce two scales, the *integration scale* σ_I, and the *derivation scale* σ_D, and they obtain a 2×2 matrix $\mathbf{M}_{Harris}$ for each point $\mathbf{x}$ in the image I (again, this

[5] Kanade-Lucas-Tomasi, see http://www.ces.clemson.edu/~stb/klt/

matrix captures the gradient distribution in a local neighborhood of $\mathbf{x}$, where the size of the neighborhood depends on σ_I and σ_D, see also [138]):

$$\begin{aligned}\mathbf{M}_{Harris}(\mathbf{x},\sigma_I,\sigma_D) &= \boldsymbol{\mu}(\mathbf{x},\sigma_I,\sigma_D)\\ &= \sigma_D^2 G(\sigma_I) * \begin{pmatrix} I_x^2(\mathbf{x},\sigma_D) & I_xI_y(\mathbf{x},\sigma_D)\\ I_xI_y(\mathbf{x},\sigma_D) & I_y^2(\mathbf{x},\sigma_D)\end{pmatrix}\end{aligned} \tag{3.8}$$

Harris and Stevens define a measure of cornerness c_{Harris} that does not require computing the eigenvalues of $\mathbf{M}$:

$$c_{Harris} = \det\mathbf{M} - \alpha\text{trace}^2\mathbf{M} \tag{3.9}$$

A corner is detected at $\mathbf{x}$, if C_{Harris} is above a threshold t_{Harris}.

3.2.1.2 Using second derivatives

Corner detectors based on the Hessian determinant: In general, these detectors are related to curvature and are invariant to rotation. The first detector of this kind was published by Beaudet [13]. Corners are defined as local maxima of the determinant of the Hessian matrix $\mathbf{H}$.

$$\det\mathbf{H} = \det\begin{pmatrix} I_{xx} & I_{xy}\\ I_{xy} & I_{yy}\end{pmatrix} = I_{xx}I_{yy} - I_{xy}^2 \tag{3.10}$$

Variants of this detector include the detectors of Kitchen and Rosenfeld [92] and of Dreschler and Nagel [42].

DoG/LoG: We refer to Section 3.1.2, and Eq. 3.4, where we already discussed the use of a Laplacian to detect edges, corners, and points (blobs) in a smoothed image. The first detector of this kind was proposed by Marr and Hildreth [127] as an edge detector. The image is first smoothed with a Gaussian to reduce noise, and edges are located at zero-crossings of the Laplacian:

$$L = \Delta(G * I) = (\Delta G) * I = 0. \tag{3.11}$$

This ΔG kernel, the 'Laplacian of Gaussian', or LoG operator can be closely approximated by the 'Difference of Gaussian', or DoG operator, by taking the difference of two images that are smoothed with two Gaussians G_1 and G_2, with σ_1 and σ_2. The DoG operator searches for zero-crossings of

$$D = G_1 * I - G_2 * I = 0. \tag{3.12}$$

Yuille and Poggio analyzed a scale space of zero-crossings. They discussed the change in position and the vanishing of zero-crossings with increasing scales, and they showed that Gaussian smoothing guarantees that no new zero-crossings can emerge [209].

Lowe's keypoints: For a greyvalue profile perpendicular to an edge, it is easy to see that the DoG operator will have a zero-crossing at the edge, with a local extremum (one maximum and one minimum) at each side of the edge. For a blob at a corresponding scale, the DoG will deliver one local maximum (or minimum, depending on foreground/background greyvalues). This has led to Lowe's idea of a keypoint detector that searches for local maxima and minima of D [122, 123]. To suppress corners with low contrast, Lowe applies a threshold, and to distinguish between edges and corners, he also assesses local curvature, based on the Hessian matrix of D.

$$\mathbf{H}_D = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \tag{3.13}$$

A keypoint is detected when

$$\frac{\text{trace}^2 \mathbf{H}_D}{\det \mathbf{H}_D} < th. \tag{3.14}$$

3.2.1.3 Saliency without derivatives

A number of saliency detectors has been proposed which work directly on the images without using derivatives at all. In the following we briefly describe four detectors of this kind. The detectors by Kadir and Brady [88] and by Matas et al. [129] have been used in object recognition approaches. While the morphological detector [99] and the SUSAN detector [186] deliver quite impressive results, they have not been used in recognition, probably because they are not straightforward enough to extend to handling varying scales.

Morphological corner detector: This detector is based on mathematical morphology and was proposed by Laganière [99]. The detector uses four different structuring elements and applies them by performing an assymetrical closing. It is rather invariant against arbitrary corner rotations and insensitive to small image structures.

SUSAN corners: Smith and Brady [186] presented an approach to low level image processing that can be used for edge and corner detection and for structure preserving noise reduction. They use local circular windows and observe the area within the mask that is sufficiently similar in brightness to the center pixel of the mask. In a sliding window approach, this area will attain local minima at edges and corners. Their algorithm is very robust when degraded by noise, can deal with low contrast, and is computationally efficient (about 10 times faster than the Harris corner detector).

Kadir/Brady saliency: This detector builds on an idea of Gilles [73], who observed grayvalue histograms for image patches $R_{\mathbf{x}}$ around an image location $\mathbf{x}$, and found Shannon entropy to be a good measure of local saliency. The histograms can be interpreted as probability distributions (probability $P_{R_{\mathbf{x}}}(d_i)$ that a pixel in the patch has a certain grayvalue d_i), and the local entropy

$$H_{R_{\mathbf{x}}} = -\sum_i P_{R_{\mathbf{x}}}(d_i) \log P_{R_{\mathbf{x}}}(d_i) \tag{3.15}$$

will be higher for patches of high signal complexity (for salient regions with flat distributions). Kadir and Brady [88] extend this idea towards an automatic selection of the optimal *scale*, where the entropy is a maximum with respect to the selected diameter of a circular patch $R_{\mathbf{x}}$. This detector has been successfully used by Fergus et al. [53] in their seminal categorization work.

MSER – maximally stable extremal regions: MSERs can be computed by successively thresholding an image. For the illustration of the method, let us assume images with grayvalues ranging from 0 to 255. Applying successive thresholds $t = 1 \ldots 256$ will lead to 256 binary images – homogeneous white for $t = 1$ and homogeneous black for $t = 256$, with a number of connected regions emerging, merging, and disappearing for the thresholds in between. These regions are already 'extremal regions' because they represent regions R in the original image, where all pixels belonging to R are brighter than all neighboring pixels of R. To obtain 'maximally stable' extremal regions, specific threshold values are selected with respect to *stability*. Regions that don't change their shape for a wide range of thresholds are regarded as salient, and the most appropriate threshold is selected for such regions. This is a brief qualitative description of the algorithm. For a detailed mathematical formulation see [129]. Further explanations can also be found in [138].

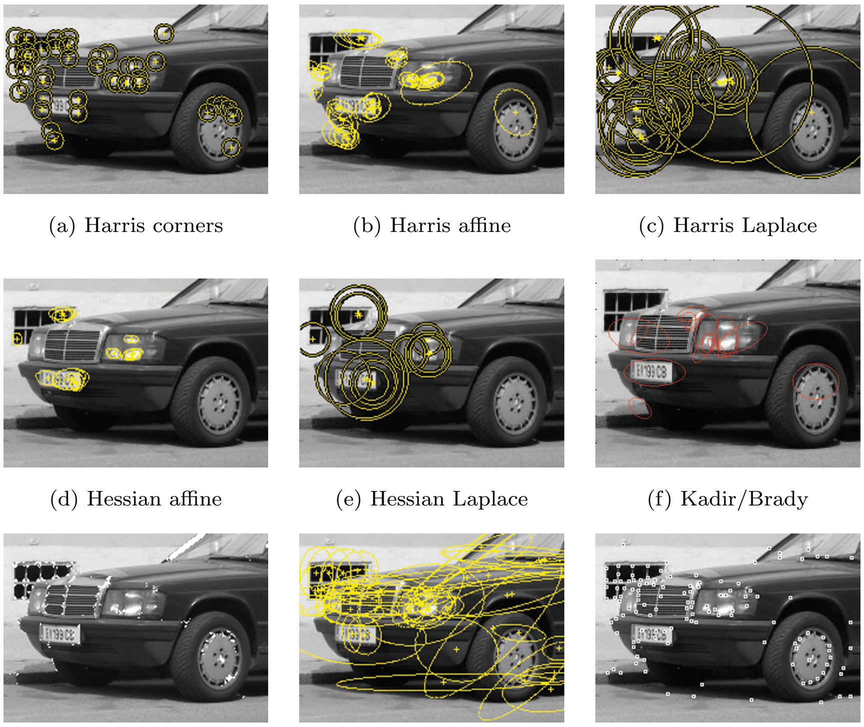
3.2.1.4 Affine covariant detectors

We have seen already in Section 2.3, that *affine invariance* plays an important role in categorization from local features. Especially when categorization is based on a 'bag of keypoints' model (see Section 3.3.1), it is very interesting to detect salient points under significantly varying scales and for various poses of the object. When salient points are located *on* the surface of the object, and when the region of support of a point is sufficiently small, perspective distortion can be modeled locally by an affine transformation (treating each keypoint separately with its own, specific affine transformation). When we can assume that the pose does not vary significantly, it is sufficient to observe local features at varying spatial scales. These ideas have been very common over the past years, which has led to a number of extensions of existing interest point detectors. In general, the goal is to detect a *characteristic scale* for each interest point and to recover an affine deformation that fits the local image data best. Common variants include scale [133] and affine 'invariant' [134, 169] Harris corners, and extensions of the Hessian [134] and of the Kadir/Brady detector [89] towards affine invariance. The original DoG/LoG and the Kadir/Brady detector can already handle changes in scale, and MSERs can cope with affine distortions.

3.2.1.5 Performance of detectors

There are performance evaluations for interest point detectors [136], and for scale and affine invariant detectors [171]. The most recent and complete summary can be found in [138], where the authors also discuss the term ‘affine invariant’ and conclude that these detectors should, in fact, be termed *affine covariant*, because the corresponding regions change covariantly with the transformation.

The survey [138] also proposes comparative measures (repeatability, overlap) and compares the various detectors on several datasets with varying changes in viewpoint, scale, illumination and blur. The conclusion is that there is not a single detector that performs best, but in many cases the MSER detector obtains high scores, followed by the Hessian-Affine. But there are further considerations, e.g. the required number of points, where Hessian-Affine and Harris-Affine deliver more points than the other detectors. Figure 3.2 provides a qualitative comparison of the saliency detectors that were discussed in Section 3.2.1.



(a) Harris corners

(b) Harris affine

(c) Harris Laplace

(d) Hessian affine

(e) Hessian Laplace

(f) Kadir/Brady

(g) morphological

(h) MSER

(i) SUSAN corners

Fig. 3.2 This figure gives a qualitative comparison of nine popular saliency detectors. All detectors are described in Section 3.2.1 and are applied to the same image showing the front of a car.

3.2.2 Descriptors

When salient regions or points with their supporting regions have been detected, the obvious way to proceed is to try to come up with a representation of such salient regions in terms of descriptive features. Features are required for a feasible correspondence analysis (a matching of two sets of feature points that is purely based on their respective image coordinates is highly ambiguous, and of prohibitive complexity). Typically, a descriptor of a salient region will be comprised of a number of different features, often represented as a *feature vector*. Many such features have been proposed in the pattern recognition literature, some of them related to pixel intensities and color, some to information extracted from boundaries, others using transformations to other coordinate spaces (most popular: features extracted from the Fourier spectrum). In this section, we focus on descriptors for local salient regions, which are useful for categorization. A number of such descriptors have been proposed and were used for categorization and for specific OR approaches, and the performance of such descriptors has been evaluated (see [132] for categorization, and also [137], but more in terms of recall for specific recognition than for categorization).

There are several possibilities to obtain descriptors which are invariant or at least robust in response to certain distortions. For example, affine invariant description can be obtained by using affine covariant detectors (see Section 3.2.1.4) to detect a salient region and to normalize it to a canonical patch so that any kind of descriptor can be used. Another possibility is to use descriptors which themselves are invariant against distortions and can be used on arbitrary patches which are extracted by any saliency detector. Thus, normalization and/or invariance can occur in both of the processes of detection and/or description. The resulting combination of detector and descriptor can be invariant to scaling, rotation, affine deformation and change in illumination, depending on the amount of geometric and radiometric distortion that should be compensated in the application.

Our own work [155] showed that in categorization, the performance of certain combinations of descriptors and detectors may be category-specific. This is also supported by Zhang et al. [210]. In general, good descriptors for categorization may differ from good descriptors for other applications like specific recognition, image retrieval, or wide baseline matching. They should exhibit sufficient descriptive power, but at the same time not over-emphasize specificity related to a specific individual. There certainly remains research to be done into the utility of certain descriptors for categorization. The remainder of this section presents a number of popular descriptors that have been successfully applied in categorization systems.

Grayvalues: The simplest way to describe a patch is by its raw pixel values (grayvalue or color). Often this leads to very high-dimensional yet redundant feature vectors, so that subsampled grayvalues are used instead. Two descriptors can be compared by normalized cross-correlation. This descriptor is not robust against any (radiometric or geometric) distortions.

Obdržálek and Matas define 'local affine frames' (LAFs [147]) for MSER regions and use grayvalue profiles on this affine invariant representation for specific OR under strong viewpoint variation.

Moments: Moments have been used as descriptors for a long time (see e.g. [74], p.672 ff). When we restrict ourselves to greyvalue images, the moments m_{pq} for a patch P are calculated from the grayvalues $I(x,y)$ as follows:

$$m_{pq} = \sum_{x \in P} \sum_{y \in P} x^p y^q I(x,y) \tag{3.16}$$

Here the m_{pq} are general moments of patch P of $(p+q)^{th}$ order. Many extensions and generalizations have been proposed towards *moment invariants*.

Moment invariants: Let $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$, then central moments μ_{pq} of order $(p+q)$ are defined as:

$$\mu_{pq} = \sum_{x \in P} \sum_{y \in P} (x - \bar{x})^p (y - \bar{y})^q I(x,y) \tag{3.17}$$

Central moments are invariant with respect to translation.

Hu [83] has derived a set of seven *invariant moments* $\Phi_1 \dots \Phi_7$, which was extended by Maitra [124]. These Φ_i are based on the notation of *normalized central moments* $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$, with $\gamma = 1 + (p+q)/2$. For example:

$$\Phi_1 = \eta_{20} + \eta_{02} \tag{3.18}$$

$$\Phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{3.19}$$

$\Phi_1 \dots \Phi_7$ are invariant to translation, rotation, and scale (see [74], p.674). Based on this idea, Flusser [61] defines a number of *affine invariant moments* $I_1 \dots I_4$.

These concepts have been extended to geometric/photometric invariants by van Gool et al. [98]. Their invariants include shape and intensity moments up to the order of two for grayvalue images. These moment invariants have been used as descriptors for registration and recognition [194], image retrieval [195], categorization [153, 155], and for wide baseline matching [197]. Their work was further generalized to moment invariants for color images [139].

Filters: There is a large number of possible features that can be calculated based on local filter operations, e.g. by convolution with a kernel (high-pass, low-pass, directional derivatives). Local geometry has been extracted by Koenderink and van Doorn using banks of oriented filters that are claimed to model receptive fields of the human visual system [95]. These "local jets" were used as descriptors for image retrieval by Schmid and Mohr [170]. Other examples include differential invariants [183], Gabor filter banks [45] and steerable filters [64]. Obdržálek and Matas [148] generate discrete cosine transform (DCT) representations for normalized local patches and use these descriptors for image retrieval.

SIFTs: SIFT descriptors were originally proposed by Lowe [122] in tight coupling with the DoG/LoG keypoint detectors (see Section 3.2.1.2). However, the descriptors can be used as well on any normalized patch. In summary, scale-space extrema are detected, and DoG/LoG keypoints are localized. Next, local gradient directions are calculated to obtain one or more orientations for each keypoint location, and image data is transformed with respect to location, scale, and orientation. This leads to a normalized local window, which is invariant to translation, scaling, and rotation of a keypoint. The term "SIFT" is an abbreviation for "scale invariant feature transform".

SIFT descriptors are calculated for local patches (called 'sample arrays' by Lowe), typically of size 8 × 8 or 16 × 16 pixels. A patch is subdivided into 4 × 4 sample regions, where a weighted histogram of gradient directions is computed per sample region. Lowe uses a histogram binning of 8 different gradient directions (covering 360°). For a 16 × 16 patch, this leads to 16 sample regions, and a SIFT descriptor vector of dimension 128 (16 regions × 8 orientations). For the most up-to-date and complete presentation of SIFT descriptors see [123]. The SIFT representation has been further improved by applying principal component analysis to the gradient patches, resulting in PCA-SIFT descriptors [91].

3.2.3 The role of detectors/descriptors for categorization

Descriptors are required as the basic building blocks of most of the representations described in Section 3.3 below. Only global appearance based representations (e.g. parametric eigenspace) work directly on the grayvalues of the pixels of an input image, but are more suited for specific OR than for categorization. The role of saliency detectors may be questioned, however. While the past years have seen a tremendous success of detectors, which focus the extraction of descriptors on 'salient' regions, this may be due to the use of too simple training data[6]. When training images show category examples at prominent scale, well-structured and with high contrast to rather homogeneous background, one can expect to detect many salient points on the objects. With increasingly complex training data, we can observe a paradigm shift. People start to calculate descriptors at every pixel in the image, or at a rather dense grid. Deselaers et al. [38] pioneered this idea, using a grid in conjunction with salient point detectors. They found that salient points

[6] See also the discussion on the degree of supervision in Section 2.4.

work well in the foreground (on the objects of the Caltech database), but the grid-based descriptors deliver useful information about homogeneous regions and in the background. Now, there are many results on more complex data, relying just on a grid, e.g. Bosch et al. [23] who calculate overlapping descriptors at every 3rd or 7th pixel, or Winn et al. [205] who convolve the images with a filter bank to obtain a dense set of local descriptors. Nowak et al. [146] compare randomly sampled points to interest points and find that interest points work well for a small number of points per image. In general, however, the performance improves with the number of points, and is best for many, randomly sampled points.

These results show that both, sparse descriptors (sampled at 'salient' points) and dense descriptors (sampled randomly or at a dense grid) have their advantages and disadvantages. Performance characteristics are closely related to training and test datasets. For prominent objects and little background clutter, saliency detectors are recommended as an efficient means to reduce the amount of data to be processed. Grid based methods may be preferable when information about homogeneous regions is essential (sky, grass, water, foliage) and when the objects appear at smaller scales in cluttered images.

3.3 Object models

To the expert in categorization, this section may give rise to controversial discussion. What is the best presentation? There are definitely several, probably equally justified ways. One could for instance try to structure the section as follows: appearance-based; keypoint-based; contour/shape-based; graph-based (including qualitative spatial models); 3D reconstruction-based. Which models should be considered most important? The field has seen considerable success for bags of keypoints and for constellations of local salient features over the past few years. But it might well be the case that the limits of these approaches have already been reached. This is suggested by a number of recent contour- and shape-based models.

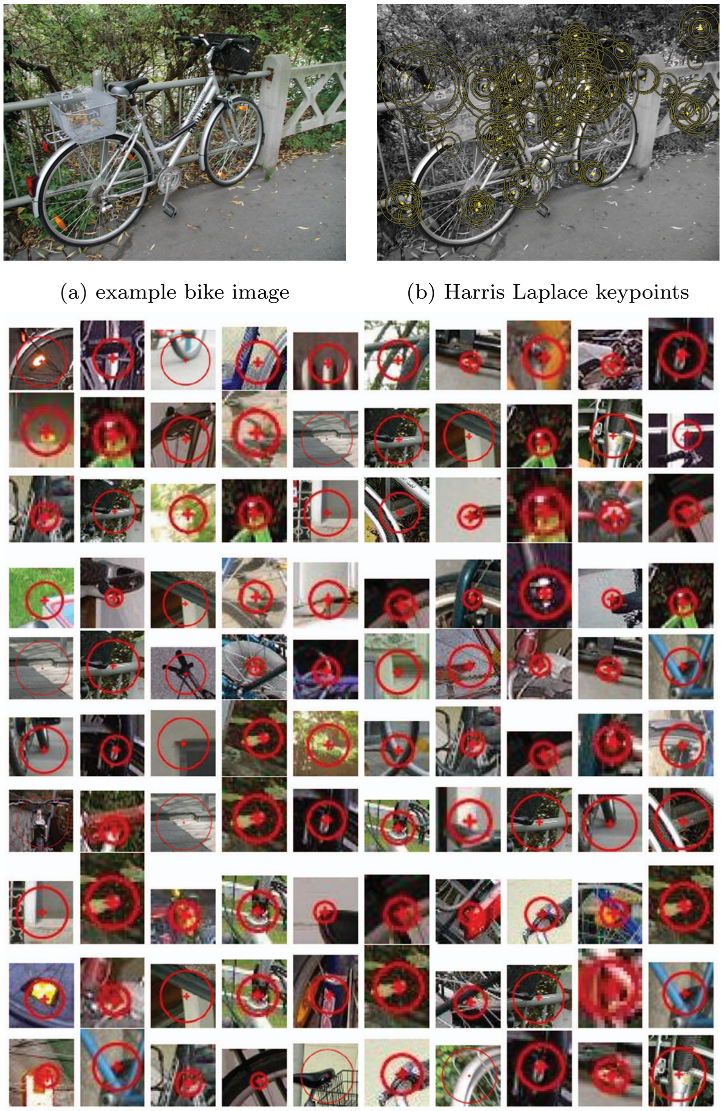
With this in mind, I have decided to constrain the models which are presented below to 2D. But I include a very brief discussion of potential benefits that can be expected from 3D reconstructionist approaches at the end of this section.

3.3.1 Bags of keypoints

The simplest possible object model is to use no model at all. This approach has also been termed ‘model-free’ or ‘geometry-free’. The basic idea is to extract salient points (keypoints) from images, and to represent an image as a set of such keypoints including some descriptor. The basic techniques which are required to extract keypoints and descriptors for this approach are presented in the previous Section 3.2.

In the first step, a set of keypoints and their descriptors (feature vectors) can be extracted from all images that are presented to a ‘bags of keypoints’ categorization system. Next, classifiers have to be found that can discriminate between the various categories. A number of successful categorization systems of this type have been presented over the past years. These systems include the one by Csurka et al. [35] (who branded the term ‘bag of keypoints’), our own system ([153, 155]), recent work by Sivic et al. [184] and several others ([8, 40, 87, 177, 191, 203]). Figure 3.3 shows an example from our own work [155], in which a bag of 100 keypoints is learned to represent the bike images from the GRAZ02 database.

The various approaches differ mainly in the types of keypoints and their descriptors, in the learning algorithms that are used to obtain classifiers (popular methods are Boosting, SVM, and EM), and in the amount of supervision that is required. Boosting in particular, has gained wide interest as a well suited learning approach, because Boosting selects a collection of very diverse features (weak classifiers) that are combined to form a diverse final (strong) classifier. What do we mean by ‘amount of supervision’? The ‘bags of keypoints’ approach requires a number of training images per category to learn a classifier for each category. Typically, there will be category specific images, and negative (counter-) examples, i.e. images that do not contain objects of the category in question. Aspects of supervision in the learning stage include the number of training images that are required to learn a classifier and the way in which objects are presented (i.e. How prominent are the objects



(a) example bike image

(b) Harris Laplace keypoints

(c) 'bag of 100 keypoints' model for bikes learned from the GRAZ02 database

Fig. 3.3 A 'bag of keypoints' of 100 salient points for bike images has been learned from the GRAZ02 image database. This figure shows an example bike image, all keypoints that were detected in this image (some of these keypoints were learned to belong to the 'bag of keypoint' bike model), and the bag of 100 keypoints. The red circles denote the exact location and scale of each keypoint.

shown? Is a bounding box or some other hint on object localization provided? How homogeneous/cluttered is the background?).

There is a further important issue which should be explicitly discussed. This approach will usually lead to a huge number of individual keypoints that have to be processed. One can expect from hundreds to several thousands of keypoints per image, depending on the image content and also on the selectivity of a keypoint detector, which is typically tuned, for instance, by a threshold. Accepting only a low number of keypoints by setting high detection thresholds may lead to problems when category-specific keypoints are neglected. Permitting very high numbers of keypoints, on the other hand, will significantly decrease the efficiency of learning a representative 'bag of keypoints'. An obvious solution to this problem is to allow a sufficient number of keypoints, but to *cluster* them in the feature space. Most of the 'bag of keypoint' models mentioned above require such a clustering step. For each keypoint, its corresponding description vector is extracted, and clusters are formed by vector quantization (VQ) based on a distance measure between description vectors (e.g. using the Mahalanobis distance). Typical clustering algorithms which have been used for this purpose include k-means clustering and agglomerative clustering. Still, the number of cluster centers needs to be rather high (e.g. $k = 1000$ in [35]) to capture high intra-class variability. Cluster centers which are obtained by such a procedure have also been termed 'visual words' [185], and can be used for image representations that go beyond a pure 'bag of keypoints', for instance, when histograms of visual words are used to evaluate the frequency and co-occurance of visual words in images.

Once a 'bag of keypoints' has been learned for each category, it can be applied to categorize new, previously unseen test images. State of the art systems achieve very high recognition rates for rather complex images, and with a fair amount of required supervision. When affine covariant detectors and/or affine invariant descriptors are used, this model can cope with even huge variations in object pose, without modeling the aspect of the object explicitly. When the task is to categorize an *image*, this approach is probably state of the art, and it will be successfully applicable for categorization, annotation and retrieval of images, as long as the number of different categories is kept reasonably low. One can envision categorization systems that might be capable of handling several hundred categories, including some hierarchy of classifiers to keep the complexity tractable.

However, there is one major drawback inherent to this approach: When the training images are not restricted to a pure presentation of the objects themselves (cropped out from the background, or in front of a prepared, homogeneous background), object *localization* will generally be poor. In [154], we have investigated this drawback based on experiments using our own categorization system [153], and also devised potential extensions to improve its object localization capabilities. The reason is that a classifier will typically contain features on the objects of interest, as well as features that are learned from keypoints that belong to background clutter. As long as the goal is image categorization, this will not matter much. It can even be an advantage to also learn some contextual information that describes the preferred surrounding, context or background for a certain category. Other models will be required when good object localization is required as well, especially when multiple instances of several different categories may occur in one test image. Such models should be able to represent compactness, spatial relationship between parts/contours/segments, etc. Some models of this kind are discussed below.

3.3.2 Contours, boundary fragments

Object contours can provide a very powerful cue about object identity, and a human observer can also generalize very well using contours, silhouettes, line- or edge drawings for visual categorization. There are a number of machine vision techniques that can be successfully used for contour-based detection and recognition, including, for instance, many variants of the Hough transform [86], geometric hashing (which was originally developed for contours, but can be used for constellations of any type of features, see [207]), and perceptual grouping [120, 167]. However, these techniques have mostly been used for the description of shape of specific objects, or for general bottom-up grouping and recognition processes. There are also many efforts to describe complete objects based on their contour, but they require a clean object-background segmentation, which can not be assumed in a realistic categorization application.

In general, boundaries provide information about object *shape*, which requires proper description in terms of shape features. An example of successful generalization for handwritten digit recognition is ‘shape context’ [17]. In the same paper, Belongie et al. [17] demonstrated the use of shape context to the recognition of specific objects (external and internal contours of COIL-20 objects), and to categorization of silhouettes (from the MPEG-7 silhouette database [104]). Ghosh and Petkov applied shape context also to robust recognition from broken contours of the same silhouette database [70]. At present, there is no ‘bags of boundary fragments’ categorization approach similar to the ‘bags of keypoints’ described above. But we see a number of promising categorization results when geometric relations between boundary fragments are modeled. These approaches are summarized in Section 3.3.4.

3.3.3 Segments

Image segments can be used in a similar way as was done for keypoints. Any existing segmentation algorithm can be applied to obtain a segmentation of an input image I into segments S_i. A description by segments is less local than for keypoints and their support regions, but many of the descriptors from Section 3.2.2 can be used. For instance, affine invariant moments can be expected to yield good results for planar segments.

Duygulu et al. [44] have presented an interesting approach to categorize segments. Images are segmented using the normalized cuts algorithm [179], and segments are described by a vector of 33 features (color, texture, size, moments, etc.). Segment categorization is modeled as a process of *machine translation* (from segments to words), and learned by EM based on a database of annotated images.

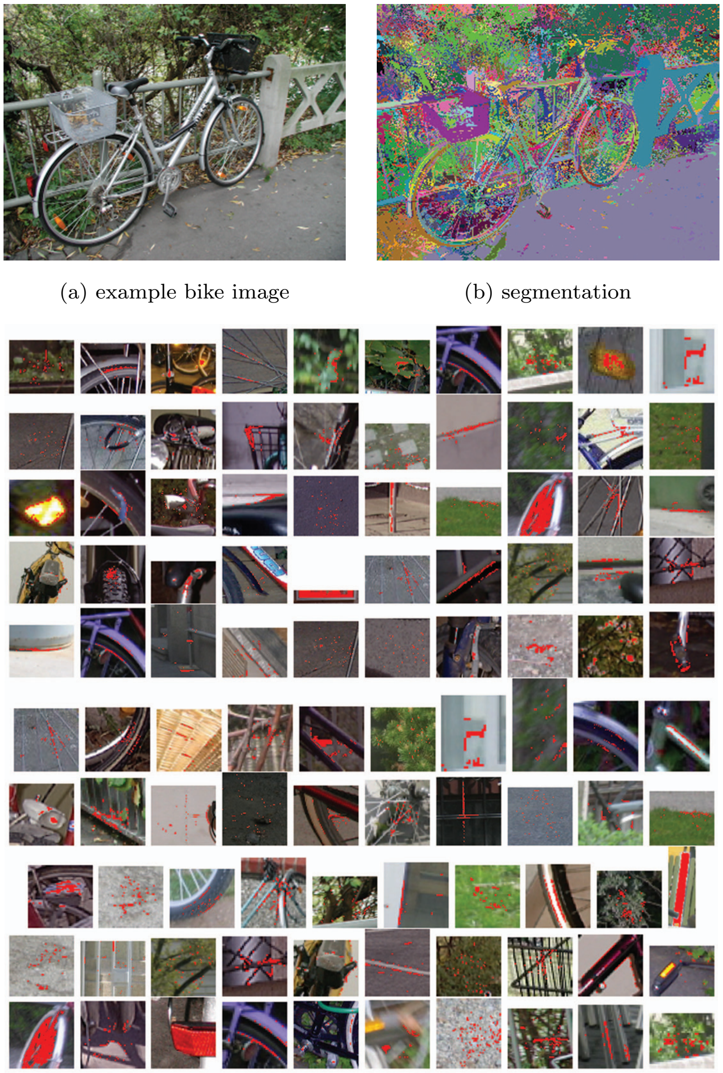
In our own categorization system [153, 155], we have implemented a ‘bags of segments’ approach together with ‘bags of keypoints’. We, thus, can compare the categorization performances for various combinations of detectors and descriptors and we find that the performance depends on the categories as well as on the databases used for training and tests. There are categories that are better modeled by keypoints, and others that are better modeled by segments. In general, we found that 'bags of segments' tend to model more context than keypoints, i.e. there is a higher percentage of segments located in the background than keypoints. For direct comparison with Figure 3.4, we show the same bike image, and a bag of 100 segments model for bikes, obtained from the GRAZ02 database, in Figure 3.4.

3.3.4 Constellations, codebooks

When we observe the merits and drawbacks of a pure 'bags of keypoints' approach, it seems quite obvious to try to extend this idea to a model that captures local saliency, but also represents spatial relationship between parts. This is now a very popular approach, and much of the recent success in categorization is due to such models of 'constellations of parts'.

Research on part-based representation can be traced down to Fischler and Elschlager's pictorial structure model [59]. They propose a model of parts that are spatially, but flexibly related to each other. Such a model can be deformed to a certain degree (one can imagine the various parts being connected by springs). This idea has been picked up by several researchers. For recent work in this direction, including a review of related work, see Felzenszwalb and Huttenlocher's article on 'pictorial structures for object recognition' [51], where such models are built for the recognition of faces and human bodies. Hand-labeled images are required to train the model.

The 'constellation' model has gained a lot of interest and probably is the most popular part-based model today. It gradually developed through a number of publications ([26, 53, 110, 204]). Objects are modeled as random constellations of parts, explicitly representing the mutual spatial relationship between parts (in a spring-like manner). Parts are detected as local salient points. In [53], the saliency detector by Kadir and Brady [88] is used. The support regions are normalized to a patch of fixed size which describes the appearance of the part. A complete probabilistic formulation of the recognition problem is obtained,



(a) example bike image

(b) segmentation

(c) ‘bag of 100 segments’ model for bikes learned from the GRAZ02 database

Fig. 3.4 A ‘bag of segments’ of 100 salient regions for bike images has been learned from the GRAZ02 image database. This figure shows an example bike image, all segments that were detected in this image, and the bag of 100 segments. Note that we use our ‘similarity measure’ segmentation algorithm described in [69]. This algorithm tolerates regions (shown in red in (c)) which are not connected, but may consist of closely neighbored parts.

based on position **X**, scale **S**, and appearance **A** of salient parts (in the most recent contributions by this group [54, 56], **D** is used as a more general acronym for any kind of local descriptor instead of **A**, but in the subsequent discussion, we stick to the notation from [53], which is *the* reference for the constellation model).

A generative model is learned from training images. Typically, N (about 30) salient parts are selected per image, and a model consists of P (about 6) parts and a set of parameters θ. Recognition of a model in a test image is assumed, when a Bayesian decision R is above the threshold.

$$R = \frac{p(Obj|\mathbf{X},\mathbf{S},\mathbf{A})}{p(noObj|\mathbf{X},\mathbf{S},\mathbf{A})} \approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{Obj})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{noObj})} \tag{3.20}$$

A factorization of

$$\begin{aligned} p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) &= \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\theta) \\ &= \sum_{\mathbf{h}\in H} p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)p(\mathbf{S}|\mathbf{h},\theta)p(\mathbf{h}|\theta) \end{aligned} \tag{3.21}$$

is used, and models of appearance $p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)$, shape $p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)$, and scale $p(\mathbf{S},\mathbf{h}|\theta)$ can be learned. The vector **h** describes a 'model hypothesis' of P parts that are drawn from the total number of N available parts. H is the set of all valid hypotheses. Since $|H| = N^P$, these constellation models are limited in the affordable number P of parts of a model, and they have to be quite restrictive in the application of the saliency detector to keep N tractable. Figure 3.5 from [53] shows a typical motorbike model with six parts and a number of correct recognitions on images from the Caltech database.

In summary, the main limitations of the constellation approach as published in [53] lie in the relatively low number of parts (learning models with many parts is of prohibitive complexity), and in the amount of required supervision. The objects have to be shown very prominently so that a sufficient number of salient parts, that are located on the objects, can be shared in many training images. When these conditions hold, categorization results are excellent for various categories of the Caltech database (motorbikes, faces, airplanes, cars(side)), as reported

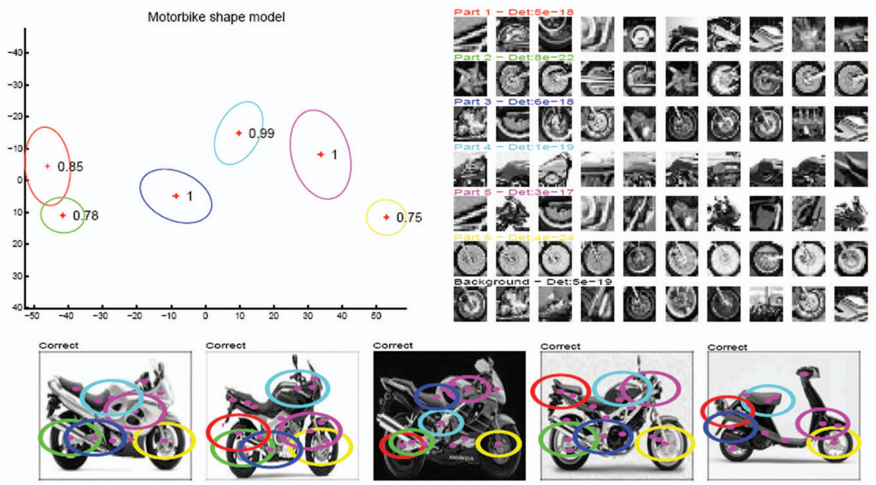


Fig. 3.5 Example of a typical motorbike constellation model with six parts (©2003 IEEE, from Fergus et al. [53], "Object class recognition by unsupervised, scale-invariant learning", Proc. CVPR, 2003, p. II-270, reprinted with permission from the IEEE Computer Society). The top left shows the model with the ellipses representing the variances of each of the six parts. The top right shows ten patches for each of the six parts and for the background. The bottom row shows five correctly recognized motorbikes from the Caltech image database.

in [53]. Research on the constellation model has led to a number of further significant results from the group around P. Perona. This includes one-shot learning [46], incremental learning of many (101) categories [47], a common reference frame instead of a landmark [141], and the combination discriminative learning with a generative model [82].

Shape deserves further discussion at this point. In [53], shape is represented by a joint Gaussian density of the locations $\mathbf{X}$ of a hypothesis $\mathbf{h}$. This *joint* density is certainly a good model because all parts are treated equally. In terms of a graph structure, the model corresponds to a fully connected graph (equivalently, all parts can be considered as fully connected by springs), and thus models most flexibly the spatial constellation of its parts. In terms of its complexity of $O(N^P)$, other models would be preferable. This has been discussed in [56], where a star-shaped model (HSM – 'heterogeneous star-shaped model') is derived. However, such a model requires the selection of a *landmark* part, that forms the root of the star (which can also be considered a tree of depth one). This landmark is difficult to choose in the learning process. But the complexity of the star model is just $O(N^2P)$, so that models with more parts, and the extraction of more salient points become feasible. A similar approach towards a hierarchical model has been presented by Bouchard and Triggs [24]. Here, the object is modeled by a tree of depth 2 (object, parts, local feature classes), and good results for many categories have been reported.

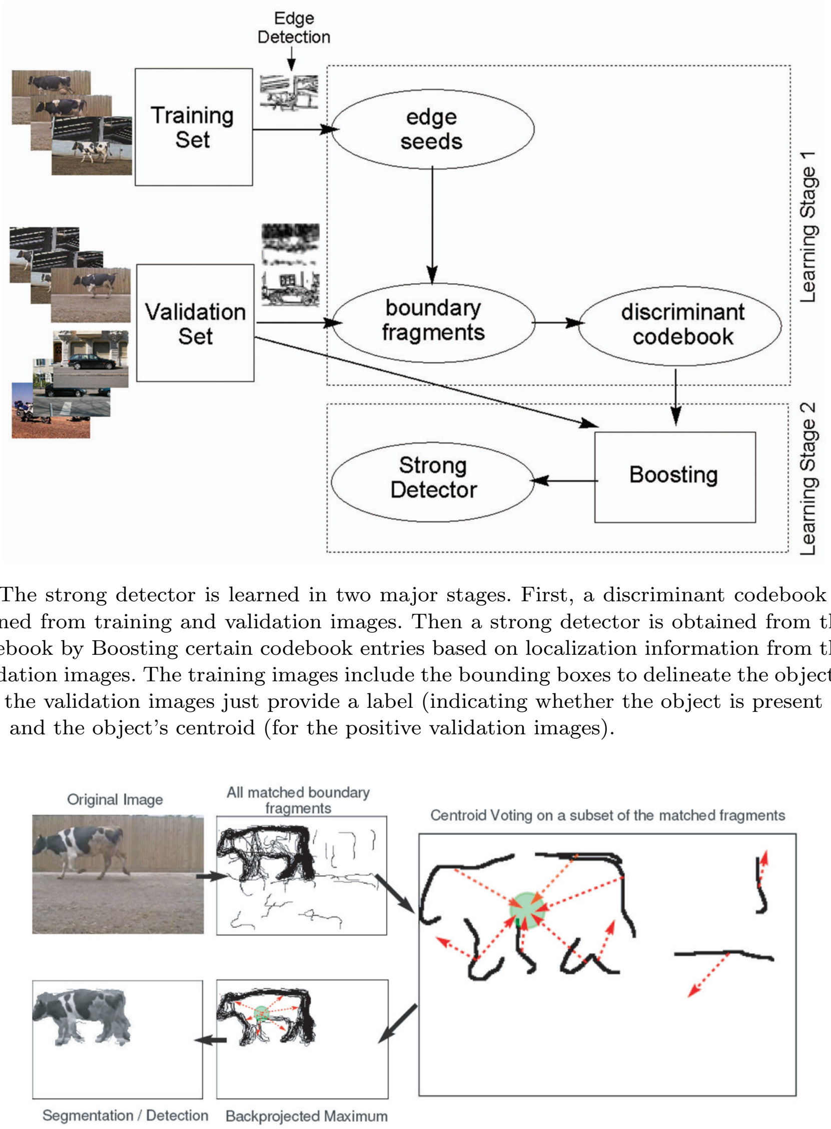
The required degree of spatial structure of a model has also been discussed by Crandall et al. [33]. They presented their $k-$fan model, where essentially k models the number of landmarks that are fully connected to each other, while each landmark may be connected to further parts in a star-shaped manner. The parameter k can range from 0 (modeling no spatial dependencies, similar to a 'bag of keypoints'), 1 (star-shaped model like HSM), to $P - 1$ (fully connected graph, e.g. joint Gaussian model). Crandall et al. show that a rather limited amount of geometric structure is sufficient to successfully categorize images from the Caltech database. Essentially, a 1-fan is sufficient for the categorization of the images, with minor improvements for 2-fans. However, localization abilities for parts may be considerably improved by adding more structure (i.e. increasing k). In an extension of their own work, Crandall and Huttenlocher [34] show that k-fan models can also be learned under weak supervision (just providing image labels) and on complex databases (GRAZ01).

The previously discussed methods try to learn and to model shape *explicitly* in terms of constellations, star-, tree- or k-fan graphical models. In practice, all these explicit models exhibit advantages (representation of salient parts of an object) and disadvantages (limited number of parts, computational complexity of learning the model). Alternatively, Leibe et al. [106] proposed a *codebook of local appearance* that is used together with an *implicit shape model* as it collects a potentially large number of salient patches and maps the location of the patches relative to the object center by a probabilistic voting scheme. This implicit shape model has gained much interest. It was significantly enhanced to deal with varying object scale [107], and applied to the task of pedestrian detection [108, 174].

In most of the previous work, the models were built from local salient features (as described in Section 3.2.1), mostly from interest points and local descriptors extracted from the interest point's support region. Some recent models use also local edge information ([33, 34, 174]). Building on what we explained on the use of boundaries to model shape in Section 3.3.2, there are now initial research efforts into the direction of boundary models for categorization. Such a model must be able to cope with cluttered scenes, broken contours, many contours in the background, and also 'internal contours' which originate from edges or texture on the object of interest. Jurie and Schmid present scale-invariant shape features that can model local support for circles and arcs [87]. Crandall et al. [33, 34] build their explicit model from local edge support, using oriented edge appearance templates. Seemann et al. [174] extend the implicit model of Leibe et al. [106, 107] with a local chamfer descriptor on edge structure which is derived from a Canny edge detector. Bernstein and Amit [19] build a model from oriented edges. Kumar et al. [96] present an explicit shape model that extends the model of [51] and combines the outline and the enclosed texture of parts. This idea, in combination with the implicit shape model from [106], has led to new implicit models of shape based on boundary-fragments (termed 'contour fragments' by Shotton et al. [181], or 'boundary-fragment-model', BFM in our own work [156]). While these implicit models are learned from training (and sometimes also validation) images, Ferrari et al. [58] propose a method that learns the boundary from one single hand-drawn prototype and can deal with cluttered images and broken contours for detection. Figure 3.6 gives an overview of our BFM approach which learns a strong detector from a codebook of boundary fragments. While it may be too early to judge the relevance of these recent contributions to contour fragment based categorization, I definitely think that this is an important direction of research that should be further explored in the future.

3.3.5 Modeling objects in more than two dimensions

Some of the models in the previous section do already go beyond pure 2D object models. Aspect graphs model the spatial relationship between camera and object to a certain extent. From a more general point of view, it might be useful to model three-dimensional objects in



(a) The strong detector is learned in two major stages. First, a discriminant codebook is learned from training and validation images. Then a strong detector is obtained from the codebook by Boosting certain codebook entries based on localization information from the validation images. The training images include the bounding boxes to delineate the objects, and the validation images just provide a label (indicating whether the object is present or not) and the object's centroid (for the positive validation images).

(b) Detection of an object in a test image (©2006 Springer, from Opelt et al. [156], "A Boundary-fragment-model for object detection", Proc. ECCV, 2006, p. II-576, with kind permission from Springer Science and Business Media). Those boundary fragments that vote consistently for an object centroid define the segmentation matte of the object.

Fig. 3.6 Overview of our 'boundary-fragment-model' (BFM) approach to detect and segment objects (see [152, 156]).

three dimensions. Also a transition from 'viewer-centered' to 'object centered' coordinates might be of interest. Other possibilities to increase the number of dimensions include 2D space and time to model typical motion patterns (see e.g. [150] for dances of honey-bees), 3D space and time to model structure and motion for recognition [173]. This kind of modeling may be considered a reconstructionists approach to the recognition problem, and is certainly not very popular nowadays. From my own perspectives and research interest, I believe that knowledge and control of camera-to-object pose can open up a new perspective to categorization, and should probably become a future direction of research.

3.4 Learning and recognition

We have seen a number of introductory comments, mostly on probabilistic approaches, in Sections 2.1 and 2.2. This section briefly presents those methods which have been successfully applied to categorization. In general, we have to search for learning techniques that can cope with high intra-class variability. This means that we cannot rely on complete representations in our training data. Thus, successful learning algorithms for categorization must cope with incomplete, inhomogeneous, and partly missing training data. Since learning and recognition (training and test) are quite interwoven, these two aspects are discussed jointly for the various approaches below.

3.4.1 Expectation maximization EM

Expectation maximization (EM) is a well established method which can be used to produce maximum likelihood estimates in cases of missing or hidden features. Therefore, it is well suited to learn probabilistic models for categorization from examples. A general discussion of EM, as it can be found for instance in [43] (Section 3.2 for maximum likelihood and Section 3.9 for EM), presents the learning problem as a problem of estimating a vector of model parameters θ.

Adhering to the notation introduced in Section 2, feature vectors $\mathbf{f}_y$ are extracted from images I_y. We want to learn generative models that represent $P(\mathbf{f}_y|c_x)$ and are built per class c_m. We assume that the category-conditional densities $p(\mathbf{f}|c_x)$ have a known parametric form so that they can be modeled by a set of category-specific model parameters θ_x. Thus, $p(\mathbf{f}|c_x)$ depends on θ_x and we can make this explicit by writing $p(\mathbf{f}|c_x, \theta_x)$. However, we do constrain our further discussion to the learning of *one* specific model for one specific category c which simplifies the problem of learning $p(\mathbf{f}|\theta)$. The optimal parameter values $\hat{\theta}$ can be learned from a set of training examples $\mathcal{D} = \{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$. In maximum likelihood estimation, $\hat{\theta}$ is found as

$$\hat{\theta} = \arg\max_{\theta} l(\theta), \tag{3.22}$$

where the log-likelihood

$$l(\theta) = \log p(\mathcal{D}|\theta) = \sum_{k=1}^{n} \log p(\mathbf{f}_k|\theta) \tag{3.23}$$

is maximized with respect to θ. Representing the set of parameters θ as a vector of p parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^T$, extrema of $l(\theta)$ can be found by setting $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\, l = \mathbf{0}$, with $\boldsymbol{\nabla}_{\boldsymbol{\theta}} = (\partial/\partial\theta_1, \ldots, \partial/\partial\theta_p)^T$.

In the standard formulation of expectation maximization, this maximum-likelihood formulation is extended to cases where the training data $\mathcal{D}$ is incomplete. This can be formulated as feature vectors $\mathbf{f}_y$ that contain 'known' features $\mathbf{f}_{yk}$ and 'unknown' ones $\mathbf{f}_{yu}$, so that the training data can be split into sets of known and unknown features, with $\mathcal{D} = \mathcal{D}_k \cup \mathcal{D}_u$. Now EM finds an approximation of $\hat{\theta}$ by iterating a sequence of expectation (E) and maximization (M) steps using a function $q(\theta; \theta^i)$ which models the goodness of parameter fit.

$$q(\theta; \theta^i) = E(\log p(\mathcal{D}_k, \mathcal{D}_u; \theta)|\mathcal{D}_k; \theta^i) \tag{3.24}$$

This notation assumes that i enumerates the iteration, and θ^i is fixed (q is a function of θ only). E is the expectation for the log-likelihood, including the unknown features $\mathcal{D}_u$ marginalized with respect to the current best distribution that is modeled by the current parameter estimation θ^i. The EM algorithm starts at $i = 0$ with initial parameter values θ^0, and continues to expect (E-step, calculate q), and to maximize (M-step, select those θ^{i+1} that result from $\arg\max_\theta q(\theta; \theta^i)$). The process terminates when the gain in quality of fit is below a given threshold $q(\theta^{i+1}; \theta^i) - q(\theta^i; \theta^{i-1}) < th$.

Now how can this formulation of 'known' and 'unknown' features be translated into representations used in categorization? We will illustrate this using the constellation model ([53], see also Section 3.3.4). We refer to Equation 3.21, and its models of appearance, shape and scale that are learned from training images I_y. Local salient patches are extracted, forming a feature vector $\mathbf{f}_y$ for each training image. While it is known, that the object of interest (an exemplar of a certain category, e.g. a motorbike) is present in the training image, there is no supervision in terms of *parts*. This means, that there are 'good' features $\mathcal{D}_k$ that would correspond to parts (e.g. the front wheel) and 'bad' features. Furthermore, the mapping from $\mathbf{f}_y$ to $\mathcal{D}_k$ which would select the relevant parts of the model is not given. The solution is, to introduce hidden ('unknown') features that would map good features to part labels, and bad features to background. The values for both, part features and mappings are represented as the model parameters θ and are learned using EM.

Once that generative models have been learned for each class c_m, recognition of a previously unseen image I_x is straightforward. The feature vector $\mathbf{f}_x$ is extracted, and $p(\mathbf{f}_x|c_i), i = 1 \ldots m$ are calculated. If some c_i are above threshold, the class with maximum $p(\mathbf{f}_x|c_i)$ is assigned, otherwise I_x is regarded to contain no known category[7].

Similar situations as explained above for the constellation model occur quite often in categorization. Examples of successful categorization systems that use EM as their underlying learning technique include [44, 168, 188, 204].

3.4.2 Boosting

Boosting was introduced by Freund in 1995 [65]. Boosting is a very powerful learning algorithm that combines a number of 'weak' classifiers to form a final 'strong' classifier which can handle very diverse individual 'clues'. Only those weak classifiers that contribute to the strong classifier actually have to be found, so that the learning phase can be kept computationally feasible, if the training data is efficiently organized.

[7] In [53], Fergus et al. discuss the distinction between a single category c, and background bg, and demand that the ratio $R = \frac{p(\mathbf{f}_x|c)}{p(\mathbf{f}|bg)}$ is above a threshold.

The most popular extension of Boosting is probably AdaBoost [66]. This 'adaptive Boosting' algorithm uses weighted training samples. At each of its iterations, it finds a weak classifier that performs just better than guessing[8]. After each iteration, the weights of the training samples are adapted such that the performance of the previously selected weak classifier would decrease. This technique supports the selection of very diverse weak classifiers.

To translate these very general remarks to the domain of categorization, I will give a brief example (adapted from [155]) that demonstrates how AdaBoost can be used to learn a discriminative model from training data. We want to learn a strong classifier which can discriminate between 'positive' images that show objects of a category and 'negative' images that do not contain such objects. m labelled images $(I_k, l_k), k = 1 \ldots m$ are provided as training data, with $l_i = +1$ for a 'positive' example I_i, and $l_j = -1$ for a negative example I_j. The learning algorithm is expected to deliver a function $H : I_{test} \mapsto l_{test}$. This prediction of a label l_{test} for the test image I_{test} constitutes the discriminative recognition of I_{test} using a strong classifier H.

AdaBoost finds a number of weak classifiers $h_t, t = 1 \ldots T$ in the following way. A weight w_k is associated with each training image. Initially, $w_k = 1, k = 1 \ldots m$. A weak hypothesis h_t just has to fulfill the requirement

$$cc = \sum_{k=1}^{m} w_k|_{h_t(I_k)=l_k} > \sum_{k=1}^{m} w_k|_{h_t(I_k)\neq l_k} = mc, \tag{3.25}$$

with cc denoting the number of correctly classified, and mc the number of misclassified images. This means that h_t classifies a majority of the training images correctly.

After each round $t = 1 \ldots T$, the weights w_k are adapted by a factor β_t, such that in round $t + 1$ less emphasis will be put on images that are correctly classified in round t. In fact, AdaBoost sets the weights

[8] When we assume a situation, in which a weak classifier has to be selected based on training data that contain examples and counterexamples for one category, 'better than guessing' means that just more than 50% of the training examples need to be correctly classified. This is really a very weak requirement.

such, that h_t is *not* discriminative with respect to the *new* weights in round $t+1$.

After T iterations, T weak hypotheses have been selected, and the strong classifier H can be built from these h_t by

$$H(I) = \begin{cases} +1 & \text{if } \sum_{t=1}^{T} h_t(I) \ln \beta_t \geq 0, \\ -1 & \text{else.} \end{cases} \tag{3.26}$$

The implementation of this learning technique for categorization faces two problems. First, weak hypotheses h_t with $cc \gg mc$ (see Eq. 3.25) should be preferred. The selection of a well-suited weak hypothesis h_t in round t of the algorithm is difficult, because it requires the proper representation of features ('description vectors') and the calculation of distances ('similarity' of features). Second, the exhaustive search of all distances for all features is computationally expensive. Exhaustive search for the best weak hypothesis h_t would require testing all description vectors that have been extracted from all training images, with respect to the current weights w_k.

In summary, Boosting is a valuable learning technique which is well suited to learning discriminative models for categorization. For the interested reader, Hastie et al. ([81], Chapter 10) provide a sound general description of Boosting. Starting with the seminal face detection work by Viola and Jones [203] in 2001, Boosting has only recently (2003–2006) gained wide interest as a powerful learning method in object categorization (see e.g. [8, 153, 191, 192]). In its standard implementation, AdaBoost is computationally quite expensive, and current research on learning for categorization is thus focused on various improvements. Major issues are the reduction of computational complexity, the joint learning of many categories, and the online, incremental learning of new categories/from new training images. Algorithmic advances include GentleBoost [67], LPBoost [37], and joint Boosting [191].

3.4.3 Decision trees

In their seminal work, Beis and Lowe [15] discuss the drawbacks of existing hashing techniques for the recognition of 3D objects. They claim that there exist no invariants for general 3D point sets, and introduce their kd-tree algorithm for the efficient recognition of 3D shape. In general, kd-trees were introduced by Friedman et al. [68] as an efficient data structure to decide nearest neighbor problems in a k-dimensional space. The basic idea of Beis and Lowe is to learn such a data structure from training images, which can be used for the recognition of a test image by extracting a feature vector and finding a set of nearest neighbors in a binary decision tree. While [15] apply kd-trees to the recognition of specific objects from 3D models, Lowe [123] extended this approach to the recognition of specific objects from clusters of SIFT descriptors. However, the dimension k of his SIFT descriptors is 128, and kd-trees are known to provide no speedup compared to exhaustive nearest neighbor search for high-dimensional spaces. Thus, they resort to using the 'best-bin-first' (BBF [14]) algorithm, which delivers an approximate solution.

There is an excellent recent source on the use of decision trees for specific OR by Obdržálek and Matas [149]. They present a novel approach called LAF-tree (local affine frame tree), where they use their maximally stable extremal regions (MSERs, see Section 3.2.1.3) as salient features. In comparison to Lowe [123], they overcome the limitations of fixed size feature vectors by interleaving the process of recognition and extraction. They report close to real-time recognition rates for hundreds of specific objects.

It should be expected from this success in specific recognition from local features, that decision trees might also be well applicable to categorization. Indeed, Marée et al. [125] present a method which is based on ensembles of randomized decision trees and report good results for image categorization. Further contributions on decision trees for categorization can be expected in the future.

3.4.4 Discussion of categorization specific topics

Although we have already discussed a number of categorization related problems throughout this paper, there are several topics that deserve specific discussion with respect to learning and recognition. When appropriate, I give references to previous work, but in general I think that these topics still constitute promising future research directions.

We wish to learn models that can cope with large intra-class variability. Boosting has been presented as a learning method that extracts very diverse weak hypotheses. Category specific combinations of detectors and descriptors can be learned. Shotton et al. [182] combine clues about appearance, shape, and context in a Boosting framework which is based on textons that jointly model shape and texture.

We are still far from learning and recognizing Biederman's 30,000 different categories [21]. In many cases, individual discriminative classifiers or individual generative models are learned *per category*. There are a few approaches towards multiclass recognition, including the joint Boosting of common features that can be shared between classes [191], the incremental learning of generative models for 101 categories from few examples [47], and *cross-generalization* by adapting features that proved useful for previously learned categories to new classes [11].

Both, generative and discriminative models have their specific merits. A discriminative approach may best learn category-specific information in a bottom-up, data driven manner, such that discriminative systems yield superior recognition rates. On the other hand, generative models are learned to provide top-down reasoning and these approaches are often better in object localization. These aspects have recently been discussed by a number of authors (e.g. [166, 192]), and led to the conclusion that neither approach alone can suffice for large scale object recognition [198]. Currently, there are a number of emerging approaches that combine discriminative and generative approaches to categorization [8, 82, 97, 165].

Finally, generative models and shape- (contour-) based approaches generally provide better localization of the objects. This is highly relevant not only for object localization and segmentation, but also in cases, when objects from more than one category occur in one image. These observations might favor approaches that learn salient features *on* the objects of interest. On the other hand, *context* can play an important role as an additional clue that might, for instance, influence the prior probability for a certain category to appear in an image [144]. In this sense, it may be useful to first categorize the scene [49] and only then categorize the objects in the scene.

Object localization should also be discussed from the point of view of implementation. There are detectors of instances of categories in images which simply perform category recognition (and not localization) in a sliding window. The face detector of Viola and Jones [203] is a typical example of this kind. Other approaches, including the detector of Leibe et al. [106] and our own BFM [156], can be considered implicit shape models, which also provide a voting for potential object centroids. These object models can perform the direct localization of category instances, e.g. by detecting local maxima in a Hough voting space. Both approaches, the sliding window and the direct localization, have their specific advantages and disadvantages. A sliding window might use very category-specific features, while the direct localization approach will require less specific features to contribute to a sufficiently significant local maximum. On the other hand, direct localization is computationally efficient and more appealing in terms of a category model that can be detected at a larger scale than a sliding window, which allows the efficient implementation of scale-invariant detection [107].

4

A Prototype System for Categorization

In this section, I give a summary of our own original research work[1]. This summary is presented from my current perspective and accumulated experience within a number of research projects and research collaborations.

Over the past years, we have researched categorization from still images, starting with a weakly supervised approach following the 'bags of keypoints' paradigm. The resulting region-based categorization system is discussed in Section 4.1. While this system has shown excellent image categorization capabilities, the localization and detection of objects in the images can be considerably improved by modeling object shape. This was done in our 'boundary-fragment-model' approach which is described in Section 4.2. Finally, we are currently exploring a number of further directions, including the combination of contour and patches, the online incremental learning of many categories, as well as the formation of spatio-temporal representations and

[1] The content of this section is related to the following key publications: [69, 153, 154, 155, 156, 173]. The following people have contributed to this research: Peter Auer, Michael Fussenegger, Andreas Opelt, Andrew Zisserman. I gratefully acknowledge the support by the following projects: Austrian Science Fund national research network 'Cognitive Vision' S9103-N04, and the European projects LAVA, ECVision, and Pascal network of excellence.

3D category models from structure and motion analysis. These future trends are briefly discussed in Section 4.3.

4.1 Region-based image categorization

Figure 4.1 gives an overview of our region-based image categorization system. This approach to categorization has been termed *weakly supervised*, because just a number of training images, together with their category labels is provided to train the system. No information is required about the object localization (e.g. object centroid) and the extent of the objects in the images (e.g. bounding box). There may still be a considerable amount of supervision in training such a system when the training images show the objects very prominently. This is, for instance, the case for many of the images in the Caltech database. Furthermore, our approach is *geometry-free*, because it learns a 'bag of keypoints' model for each category. A final, strong hypothesis is learned as a collection of weak hypotheses using a slightly modified variant of

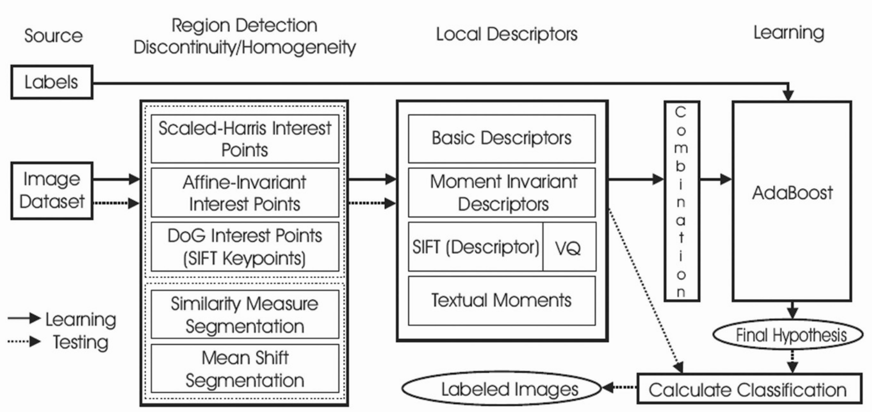


Fig. 4.1 Overview of our region-based categorization framework (©2006 IEEE, from Opelt et al. [155], "Generic object recognition with Boosting", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28(3). p. 418, reprinted with permission from the IEEE Computer Society). 'Bags of keypoints' are learned from training images using a variant of AdaBoost. A number of detectors of discontinuities and homogeneous regions can be used and the resulting patches can be described with a variety of local descriptors. The performance of individual detector/descriptor combinations, as well as the combination of several detectors and descriptors can be experimentally evaluated.

AdaBoost. Weak hypotheses are drawn from a potentially very diverse combination of local salient features and descriptors. The framework is quite general and it allows the combination of various detectors of discontinuities and of homogeneous patches with a number of description methods. The most complete description of this system is given by Opelt et al. [155] and by Opelt [152].

Figure 4.2 illustrates an example, in which a patch-based strong classifier is learned for the category bikes. As described in Section 3.4.2, Boosting selects weak hypotheses which are 'typical' for bike images (highlighted in green in Fig. 4.2), and do not occur in the counterexamples. As a further example, we refer to Fig. 3.3, which shows all 100 weak hypotheses that were learned for a specific constellation of our system: The keypoints were detected with the affine invariant Harris detector, the local patches were described by moment invariants, and the GRAZ02 bike and counterexample images were used to train the strong classifier. A different constellation of the system was used to produce Fig. 3.4: The regions were detected by similarity measure segmentation [69] and described by intensity moments, again on GRAZ02 bikes and counterexamples.

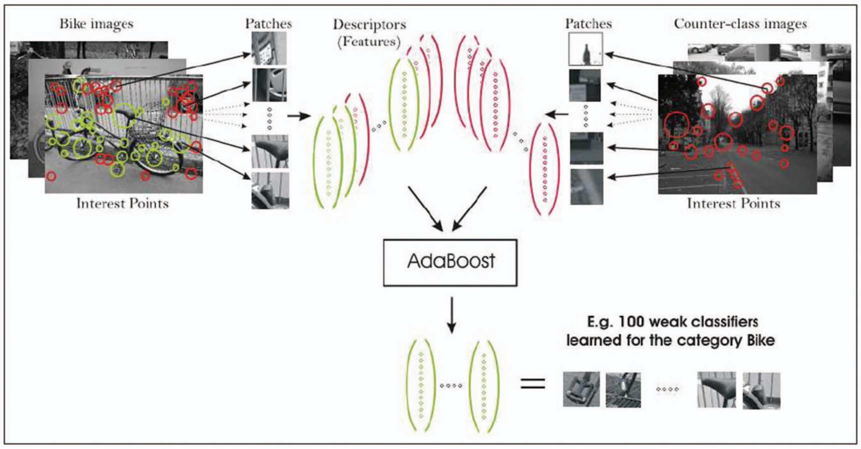


Fig. 4.2 Learning a strong classifier for the category bike. The classifier is learned from examples and counterexamples. The description consists of local patches that are extracted around interest points. The final strong classifier is a collection of 100 weak hypotheses that model the category bike.

We obtained several interesting results in our various experiments with this system. Our categorization results were excellent on available databases, and in comparison with the state of the art in 2004. This led to the need for further, more complex image databases to evaluate all aspects of our approach. We set up our GRAZ01 and GRAZ02 databases, as explained in Section 2.5. Tables 4.1 and 4.2 give quantitative results on these two datasets. To illustrate the performance of our system, Fig. 4.3 shows correctly categorized images from these datasets, while Fig. 4.4 shows some of the errors. Several of our results are quite obvious, while others require further investigation and explanation. These aspects are discussed below.

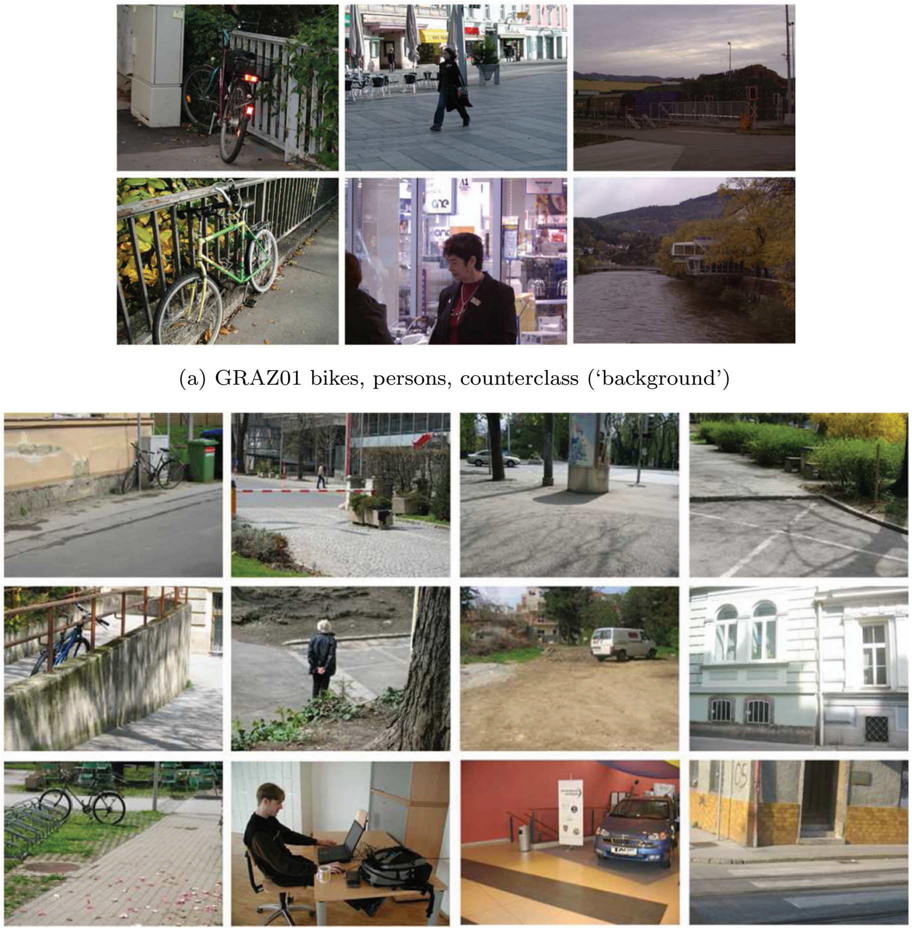
Recognition rates are category-specific, but also the best individual detector/descriptor as well as the best combinations of various detectors and descriptors depend on the category. This may be expected, because features that model a bike may obviously differ from good features for modeling people or cars. It also stands to reason that some categories are harder to recognize than others, but there are many potential explanations for this fact. Recognition performance might

Table 4.1 Comparison of ROC-equal error rates on the GRAZ01 dataset, achieved with three specific combinations: Affine invariant interest point detection with description by Moment Invariants, DoG keypoint detection combined with SIFT as description method, and Similarity-Measure-Segmentation (SM, [69]) described by intensity distributions.

Dataset	Moment Invariants	SIFTs	SM
Bikes	26.5	22.0	16.5
Persons	37.0	23.5	43.5

Table 4.2 ROC-equal-error rates of various specific combinations of region extractions and description methods on the three categories of the GRAZ02 dataset. The first and the second column are obtained with the affine invariant interest point detection and Moment Invariants or basic intensity moments as local descriptor. The third column was achieved using DoG keypoint detection and SIFTs as description method. The last column shows the results of experiments performed using Similarity-Measure-Segmentation (SM) and description via intensity distributions.

Dataset	Moment Invariants	Basic Moments	SIFTs	SM
Bikes	27.5	23.5	23.6	26.0
Persons	19.0	22.8	30.0	25.9
Cars	33.0	29.8	31.1	43.5

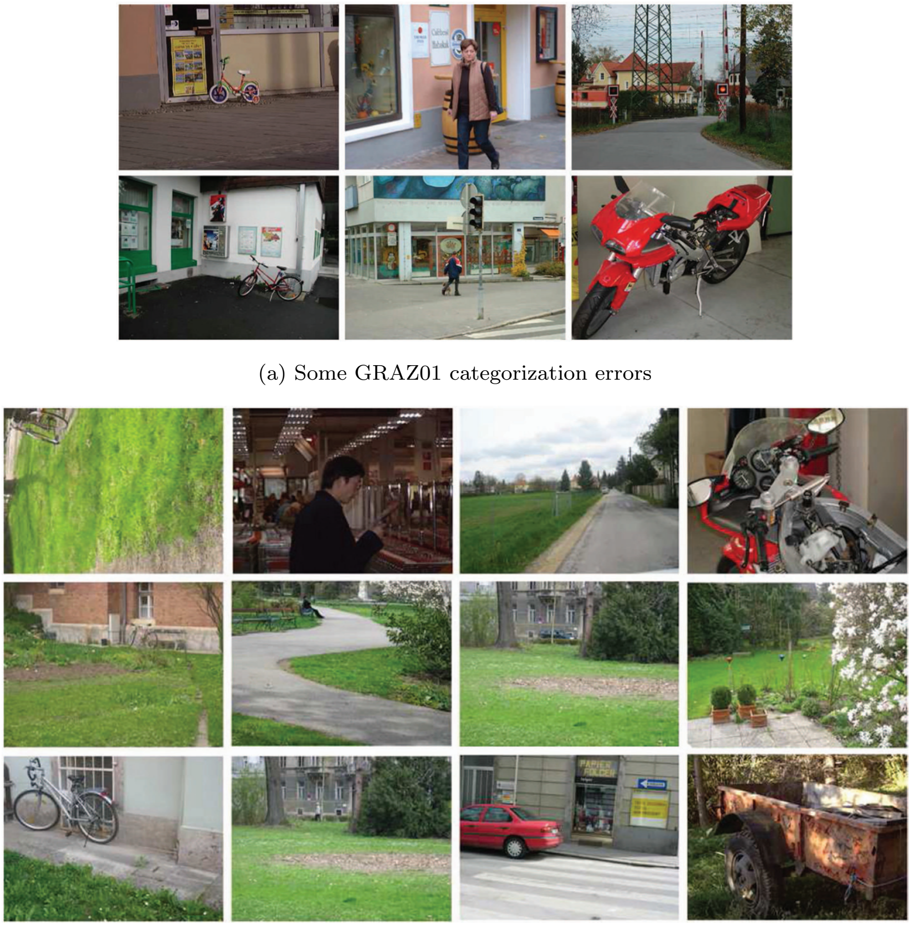


(a) GRAZ01 bikes, persons, counterclass (‘background’)

(b) GRAZ02 bikes, persons, cars, counterclass (‘background’)

Fig. 4.3 Examples of images from our GRAZ01 and GRAZ02 databases that were categorized correctly by our region-based system.

depend on the training (e.g. ‘better’ training images for certain categories or counterexamples that favor a certain category), on a different amount of intra-class and inter-class variation, but also on the category itself. A region-based approach will be better suited for solid objects with sufficient texture on the surface than for partly transparent or even fragmented objects. For instance, there is a lot of background visible within the outline of a bike. When this background is often



(a) Some GRAZ01 categorization errors

(b) Some GRAZ02 categorization errors

Fig. 4.4 These images from our GRAZ01 and GRAZ02 databases were categorized incorrectly. All instances of objects (bikes, persons, cars) were incorrectly categorized as countercless ('background'). The counterclass images were categorized as bikes.

pavement, a facade or foliage in the training images, the performance of the bike detector will strongly depend on the balance of the image database (does pavement/facade/foliage also occur in the counterexamples and/or in images of other categories of interest?).

Localization performance of such a geometry-free approach is in general poor, as has to be expected. The algorithm learns a bag of

Table 4.3 This table shows the percentage of the weak hypotheses that are *not* located on the object. Here we used the same combinations of detectors / descriptors as in Table 4.2 on the GRAZ02 dataset.

Dataset	Moment Invariants	Basic Moments	SIFTs	SM
Bikes	21	30	39	55
Persons	23	45	54	74
Cars	56	63	52	84

keypoints, but these keypoints may be located anywhere in the images. Table 4.3 shows that the percentage of patches in the background varies between 21 and 84 percent, depending on the category, and also on the detector/descriptor combination. There are several potential remedies for this poor localization capabilities, but in general they will require an increased amount of supervision (see e.g. [154]), and also benefit from modeling of geometric relations (e.g. the constellation model of Fergus et al. [53]).

4.2 Detection and localization with a Boundary-Fragment-Model (BFM)

Our second approach to categorization builds on shape information which is extracted from edges that belong to the (external or 'internal') contour of the object. We do not require complete boundaries, but use a codebook of 'boundary-fragments' that vote for potential object centroid locations. An overview of the learning and detection components of this BFM system has already been presented in Fig. 3.6 as an example for a shape-based codebook approach in Section 3.3.4. In the first stage of learning, a codebook of boundary fragments together with their centroid votes is learned from training images that show the objects of interest. This learning step requires also a bounding box around the objects in the training images and an additional set of validation images. This validation set contains examples, including the object's centroid position, and counterexamples. The learning is thus more supervised than the region-based approach described in Section 4.1. In the second stage of learning, a strong detector is learned from weak detectors that are selected from the codebook of boundary fragments. Since one boundary-fragment is often not sufficiently discriminative (its shape may be too general, or it may match with background clutter of similar shape), a weak detector consists of k different boundary fragments, where all fragments are required to vote for a common centroid position of the object (see Fig. 3.6). We use $k = 2$ or 3 boundary fragments to build such weak detectors (the complexity of the approach increases dramatically with increasing values of k). The formation of these weak hypotheses, and the Boosting of a final strong classifier also requires the set of validation images. Again, there is a bit more supervision needed than for our region-based approach.

Figure 4.5 depicts a number of detection results obtained with a multiclass extension of this BFM approach (see [157]) for various categories from the Caltech and GRAZ image databases. With this model, we obtain excellent localization of the objects of interest, and also very good recognition rates. Tables 4.4 and 4.5 show that the boundary

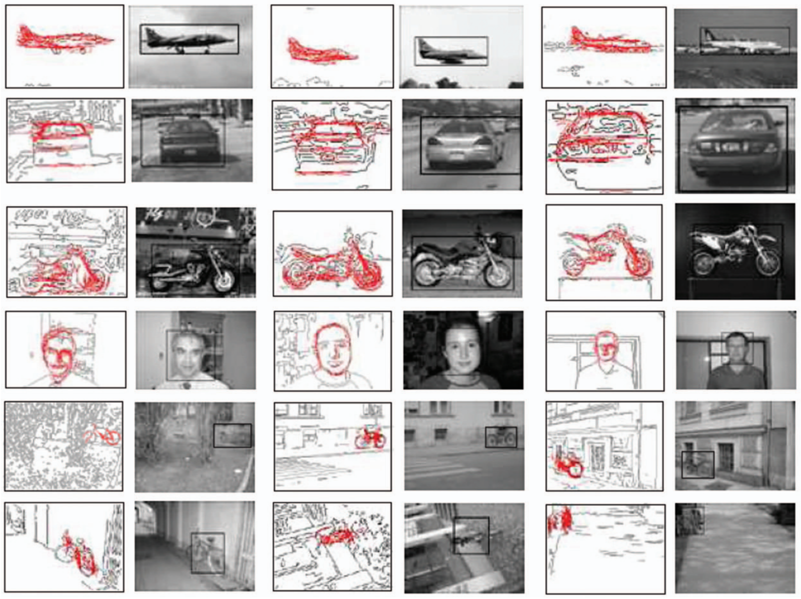


Fig. 4.5 This figure shows a number of detection results obtained with our BFM approach. For each edge image (left), the extracted Canny edges are shown in grey, and those boundary fragments that led to a detection are shown in red. Each original greyvalue image (right) is shown together with the detection result, a bounding box around the object. Examples include airplanes, cars rear, motorbikes, and faces from the Caltech database, and GRAZ bikes.

Table 4.4 Comparison of the BFM detector to other published results on the Caltech dataset, part a: We compare the actual object *detection* error (RPC-equal-error) of our BFM (BFM-D) with the results of Leibe et al. [106].

Category	BFM-D	[106]
Cars-rear (C)	**2.25**	6.1
Airplanes (A)	**7.4**	–
Motorbikes (M)	**4.4**	6.0
Faces (F)	**3.6**	–

Table 4.5 Comparison of the BFM detector to other published results on the Caltech dataset, part b: We compare the *categorization* errors (ROC-equal-error rates) of our BFM (BFM-C) with a number of other results. The Caltech categories are the same as in Table 4.4: Cars-rear (C), Airplanes (A), Motorbikes (M), and Faces (F).

	BFM-C	[53]	[153]	[184]	[5]	[8]	[56]	[190]	[210]	[28]
C	**0.05**	9.7	8.9	21.4	3.1	2.3	0.7	9.8	–	2.2
A	**2.6**	7.0	11.1	3.4	4.5	10.3	4.7	17.1	5.6	–
M	**3.2**	6.7	7.8	15.4	5.0	6.7	6.2	6.8	5.0	–
F	**1.9**	3.6	6.5	5.3	10.5	7.9	17.0	16.9	0.3	7.6

alone is sufficiently discriminative to outperform most state of the art patch-based categorization approaches. A detailed description of the BFM is given by Opelt et al. [156, 157] and by Opelt [152].

What is the price of this increase in performance compared to a ‘bag of keypoints’? First of all, more supervision is required in the training phase. Bounding boxes have to be provided for the training images and object centroids for the validation images. Furthermore, edge-based approaches need a higher spatial resolution than patch-based ones. For instance, the spatial resolution of the UIUC cars side data is too low to achieve a high performance. There are further limitations to scale, which are more stringent than for patches. We succeed in scaling a boundary fragment approximately in a range of 0.5 to 2 of its original size while maintaining the performance of the system. For scales < 0.5 or > 2, the performance drops rapidly.

There is one major advantage of a patch-based approach: When the patch is located on the object and can be assumed to be locally planar, affine covariant detectors and affine invariant descriptors will yield good results over a wide variety of object-to-camera poses. In contrast, object shape (and thus object contour) will vary significantly with changes of object pose. This means that patch-based models (especially bags of keypoints) can be built per category, but contour-based models have to be built per significant aspect, so that one category might require many different contour-based models. In practice, it turns out that for the external contour, the BFM models the visual rim of the object. The general shape of the visual rim may be approximately the same, even over a wide range of poses, so that the necessary number of different BFMs per category can be reduced to a handful (< 10 for reasonably complex categories, e.g. horses or cows).

4.3 Discussion

With our two quite complementary approaches (patch-based, Section 4.1 and boundary-based, Section 4.2), we have developed a solid basis to perform further categorization research in a number of relevant directions. Below, I discuss briefly several aspects that have already been tackled, as well as potential future trends.

Combination of contour and patches: Such a combination constitutes the obvious next step, given the complementary nature of the two approaches. Potential goals include the further improvement of recognition rates, improved localization (when adding contour to the patch-based approach), and reduced numbers of false positives (when adding patches to the contour-based approach). However, there are several quite different possibilities to perform a useful fusion. One possibility is to treat patches in a similar way as boundary fragments. Patches might vote for the object centroid, and a combination of several patches might form a star-shaped weak hypothesis. This idea has been tried out in Opelt's thesis [152], and has led to a certain improvement of recognition rates compared to the BFM-based approach. The approach is quite elegant, because it requires just the fusion of the two Hough voting spaces (one for boundary fragments, the other one for patches). But the complexity of this model is prohibitive, when trying to model $k > 3$ patches per weak hypothesis, which would probably be required given the lower indexing power of patches as compared to boundary fragments. Another possibility is to use a BFM to constrain the selected patches to be located within the segmentation matte obtained from the BFM. This could either result in a patch-based system that could distinguish between patches in the background and patches on the object of interest, or in a contour-based system, that uses the patches as additional source of information. Related recent work on the use of complementary features includes [174, 210].

Online incremental learning of many categories: Previous approaches are limited in a number of ways. Learning is complex and slow. Experiments are performed only for a few categories (3 to 20, with exceptions of up to 101 categories [47]). Adding a new category requires the re-training of the whole system. Similarities between categories might be represented by similar features, but usually are learned on a per-category basis.

A potential remedy is to *share* common features between classes, as proposed by Torralba et al. [191]. A similar idea is *cross-generalization* [11], in which features that proved useful for a previously learned class, are selected to learn a novel class. In our own recent contribution to this topic [157], we introduce the idea of a visual shape alphabet that can be learned incrementally. AdaBoost is adapted to learn BFMs jointly, for many categories. New categories can be added incrementally, and category similarities can be predicted from the resulting alphabet. We achieve excellent recognition rates, and a sublinearly growing number of alphabet entries with respect to the number of trained categories. However, learning is still a time-consuming task, and a future possibility might be to improve the system using online Boosting [159].

Beyond still image categorization: This is currently one of my favorite topics. At the same time it is the most speculative one in this article. Up to this point, we have mostly discussed image or object categorization from still images. Only in the introduction (Section 1.2) did we briefly mention the various tradeoffs between a ‘recognition school’ (as advocated in the remainder of this article) and a ‘reconstruction school’ that aims at building 3D models and scene descriptions. I think that time is now ripe to use the large momentum that has been gained in still image based categorization, and to potentially improve existing approaches by adding some flavor of 3D reconstruction. There are already a number of efforts going on in this direction, which are discussed in a broader sense below (Section 5, confluence of recognition and reconstruction). A straightforward idea is to combine existing (e.g. patch-based) image categorization with structure and motion analysis. Such a combination may lead, for instance, to a 3D point cloud of salient points that is extracted from a video with a relevant amount of change in object-to-camera pose. Another possibility is to model spatio-temporal relationships (e.g. typical object trajectories or motion patterns).

We have reported first results towards the use of 3D point clouds for categorization in [173], but there remains plenty of work to be done. In the lucky event, that a separation between object and background is possible, and when there is sufficient texture on the object's surface, a dense point cloud can be generated for a *specific* object. But when we assume that during a (very expensive) training phase, there are a number of point clouds generated for each category, how should we generalize towards a 3D category model? How do we robustly find an object centered coordinate system so that these point clouds can be directly compared and potentially be fused? Are point clouds a well-suited representation for categories at all? These are just a few of the open questions related to this idea, which will hopefully be researched more in depth in the near future.

5

Final Remarks

Perhaps the rapid development of research in categorization is also well documented by the sheer number of relevant publications. When I agreed to write this review article (at ECCV in 2004), the speed and amount of progress was actually hard to foresee. Now that I have come close to finishing this work, I find that roughly 30% of the references in this article refer to work that was published only after ECCV 2004. This implies, of course, that the content of this article, in some parts even the line of argument, had to change significantly from the originally planned outline. However, my hopes are that I could present the major foundations of categorization, and that some of the future trends which I discussed here will actually come true in the near future.

So, what are useful 'final' remarks on a field that is currently in such a rapid development? Appearance- and patch-based approaches to categorization have already reached a certain maturity. This is not yet the case for aspects of geometry and shape (although there seems to be some consensus about the use of 2D 'constellations' of significant parts). In terms of comparison with human categorization capabilities, I do not believe that any of the approaches presented here can really cope with *many* (thousands of) categories. But on the other hand, I am quite confident that many of these techniques and algorithms will find their way into useful components of ‘smart’ products in the near future.

What are the most likely directions of future research in categorization? I have already discussed a number of specific points that are quite related to our own research in Section 4.3. In a broader sense, I hope to see some of the following topics emerge in the near future:

Confluence of recognition and reconstruction: It might be a good time for recognition and reconstruction schools to meet. There has been recent success in online structure and motion analysis which might provide additional clues about object shape. A 2D constellation of parts will always be restricted to certain aspects of an object. Reconstructing 3D spatial relations between parts, it would be possible to build 3D constellation models. Furthermore, when the pose of the object is known, it would also be possible to reason about currently visible parts and about self-occlusion. Reconstructionist approaches might also help to extract useful information from videos. This includes object and camera trajectories, as well as higher-level information that can be generated from the 4D spatio-temporal representation of a scene (e.g. typical motion patterns, interactions between objects, occlusions).

Fusion of sensory modes: Humans often do not only rely on vision to categorize correctly. There are other important clues like weight, temperature or sound. A combination of complementary sensory modalities might be a very powerful instrument to solve otherwise ill-posed problems. For example, small visual inter-class differences might be disambiguated by adding sound as a second source of information: a cat does not bark; in comparison to a motorbike, a bicycle has no engine, etc.

Reasoning about object function: Now that some close to real time functionality is available, one can watch a scene and extract useful information about the function of an object. Both a glass and a vase may be filled with water, but the recognition of a certain action, like drinking, will provide essential clues. This ‘functional object recognition’ has roots in the works of Gibson [71], who branded the term ‘affordance properties’ of an object, and there has also been pioneering work by Stark and Bowyer [187] in this direction. With the emerging possibilities of online learning and reconstruction mentioned above, it will be possible to integrate perceptual reasoning about object function into a categorization system [201].

Embodiments: There are many interesting future embodiments of a cognitive vision system with categorization capabilities. This includes not only active (pan-tilt-zoom) cameras, mobile robots and autonomous vehicles, but also smart mobile devices that are controlled by a human user. The major advantage of such platforms is their ability to move. They can change their position, pose, and several other parameters (including e.g. zoom and aperture) to gather further information in the sense of the active vision paradigm. This might lead to future *active categorization systems*, which actively collect the necessary amount of information to successfully perform a certain categorization task.

Acknowledgements

Andreas Opelt helped with preparing the figures and tables, and he was very helpful with proofreading of the manuscript. Numerous discussions with Andrew Zisserman about Andreas Opelt's research have influenced the way I think about categorization. I have learned a lot about machine learning from discussions with Peter Auer.

I gratefully acknowledge the support by the following projects: Austrian Science Fund national research network 'Cognitive Vision' S9103-N04, and the European projects LAVA, ECVision, Pascal network of excellence, and euCognition.

References

[1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2005.

[2] S. Agarwal and D. Roth, "Learning a sparse representation for object recognition," in *Proc. ECCV*, pp. 113–130, 2002.

[3] R. O. Alferez and Y. F. Wang, "Geometric and illumination invariants for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 505–536, 1999.

[4] Y. Aloimonos and D. Shulman, *Integration of visual modules: An extension of the Marr paradigm*, Academic Press, 1989.

[5] J. Amores, N. Sebe, and P. Radeva, "Fast spatial pattern discovery integrating Boosting with constellations of contextual descriptions," in *Proc. CVPR*, pp. 769–774, 2005.

[6] C. H. Anderson, P. J. Burt, and G. S. van der Wal, "Change detection and tracking using pyramid transform techniques," in *Proc. SPIE Intelligent Robots and Computer Vision*, pp. 72–78, 1985.

[7] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 26–33, 1986.

[8] A. Bar-Hillel, T. Hertz, and D. Weinshall, "Object class recognition by boosting a part-based model," in *Proc. CVPR*, 2005.

[9] K. Barnard, V. Cardei, and B. Funt, "A comparison of color constancy algorithms. Part one: Methodology and experiments with synthesized data," *IEEE Transactions in Image Processing*, vol. 11, no. 9, pp. 972–984, 2002.

[10] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of color constancy algorithms. Part two. Experiments with image data," *IEEE Transactions in Image Processing*, vol. 11, no. 9, pp. 985–996, 2002.

[11] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *Proc. CVPR*, 2005.

[12] A. Baumberg, "Reliable feature matching across widely separated views," in *Proc. CVPR*, pp. 774–781, 2000.

[13] P. R. Beaudet, "Rotational invariant image operators," in *Proc. ICPR*, pp. 579–583, 1978.

[14] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proc. CVPR*, pp. 1000–1006, 1997.

[15] J. S. Beis and D. G. Lowe, "Indexing without invariants in 3D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1000–1015, 1999.

[16] H. Bekel, I. Bax, G. Heidemann, and H. Ritter, "Adaptive computer vision: Online learning for object recognition," in *Proc. DAGM 2004*, (C. E. R. et al., ed.), pp. 447–454, Springer, 2004.

[17] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[18] R. Bergevin and M. D. Levine, "Generic object recognition: Building and matching coarse descriptions from line drawings," *PAMI*, vol. 15, no. 1, pp. 19–36, 1993.

[19] E. J. Bernstein and Y. Amit, "Part-based statistical models for object classification and detection," in *Proc. CVPR*, 2005.

[20] I. Biederman, "Human image understanding: Recent research and a theory," *CVGIP*, vol. 32, pp. 29–73, 1985.

[21] I. Biederman, "Visual object recognition," in *Chapter 4 of: An invitation to cognitive science, vol. 2, Visual Cognition*, (S. F. Kosslyn and D. N. Osherson, eds.), pp. 121–165, MIT press, 1995.

[22] M. Bister, J. Cornelis, and A. Rosenfeld, "A critical view of pyramid segmentation algorithms," *Pattern Recognition Letters*, vol. 11, no. 9, pp. 605–617, 1990.

[23] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proc. ECCV*, pp. 517–530, 2006.

[24] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *Proc. CVPR*, 2005.

[25] R. A. Brooks and L. A. Stein, "Building brains for bodies," *Autonomous Robots*, vol. 1, pp. 7–25, 1994.

[26] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," in *Proc. ECCV*, pp. 628–641, 1998.

[27] P. J. Burt, "The Laplacian pyramid as a compact image code," *IEEE Trans. on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[28] B. Caputo, C. Wallraven, and M. E. Nilsback, “Object categorization via local kernels,” in *Proc. ICPR*, pp. 132–135, 2004.

[29] P. Carbonetto, G. Dorko, and C. Schmid, “Bayesian learning for weakly supervised object classification,” Tech. Rep., INRIA Rhone-Alpes, Grenoble, France, August 2004.

[30] F. Chabat, G. Z. Yang, and D. M. Hansell, “A corner orientation detector,” *Image and Vision Computing*, vol. 17, pp. 761–769, 1999.

[31] Cognex Corporation, “http://www.cognex.com/products/visiontools/patmax .asp,” page visited April 26, 2005.

[32] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

[33] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, “Spatial priors for part-based recognition using statistical models,” in *Proc. CVPR*, 2005.

[34] D. Crandall and D. Huttenlocher, “Weakly supervised learning of part-based spatial models for visual recognition,” in *Proc. ECCV*, pp. 16–29, 2006.

[35] G. Csurka, C. Bray, C. Dance, and L. Fan, “Visual categorization with bags of keypoints,” in *ECCV Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.

[36] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. CVPR*, 2005.

[37] A. Demiriz, K. Bennett, and J. Shawe-Taylor, “Linear programming boosting via column generation,” *Machine Learning*, vol. 46, no. 1-3, pp. 225–254, 2002.

[38] T. Deselaers, D. Keysers, and H. Ney, “Discriminative training for object recognition using image patches,” in *Proc. CVPR*, pp. 157–162, 2005.

[39] S. Dickinson, A. Pentland, and A. Rosenfeld, “3-D Shape recovery using distributed aspect matching,” *PAMI*, vol. 14, no. 2, pp. 174–198, 1992.

[40] G. Dorko and C. Schmid, “Selection of scale-invariant parts for object class recognition,” in *In Proc. International Conference on Computer Vision*, 2003.

[41] B. Draper, R. Collins, J. Brolio, A. Hanson, and E. Riseman, “The schema system,” *Int. J. Computer Vision*, vol. 2, pp. 209–250, 1989.

[42] L. Dreschler and H. H. Nagel, “Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene,” *Computer Graphics and Image Processing*, vol. 20, pp. 199–228, 1982.

[43] R. O. Duda, P. E. Hart, and D. C. Stork, *Pattern classification*, Wiley, 2001.

[44] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, “Object recognition as machine translation; Learning a lexicon for a fixed image vocabulary,” in *Proc. ECCV*, pp. 97–112, 2002.

[45] I. R. Fasel, M. S. Bartlett, and J. R. Movellan, “A comparison of Gabor filter methods for automatic detection of facial landmarks,” in *Proc. 5th Int. Conf. Automatic Face and Gesture Recognition*, 2002.

[46] L. Fei-Fei, R. Fergus, and P. Perona, “A Bayesian approach to unsupervised one-shot learning of object categories,” in *In Proc. International Conference on Computer Vision*, pp. 1134–1141, 2003.

[47] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” in *Proc. CVPR*, 2004.

[48] L. Fei-fei, R. Fergus, and A. Torralba, “Recognizing and learning object categories,” http://people.csail.mit.edu/torralba/iccv2005, Tutorial presented at ICCV 2005 pages visited Feb. 7, 2006.

[49] L. Fei-Fei and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” in *Proc. CVPR*, 2005.

[50] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[51] P. Felzenszwalb and D. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[52] P. Felzenszwalb and D. Huttenlocher, “Spatial priors for part-based recognition using statistical models,” in *Proc. CVPR*, 2005.

[53] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *In Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, 2003.

[54] R. Fergus, P. Perona, and A. Zisserman, “A visual category filter for Google images,” in *Proc. ECCV*, pp. 242–256, 2004.

[55] R. Fergus, P. Perona, and A. Zisserman, “Learning object categories from Google’s image search,” in *Proc. ICCV*, 2005.

[56] R. Fergus, P. Perona, and A. Zisserman, “A sparse object category model for efficient learning and exhaustive recognition,” in *Proc. CVPR*, 2005.

[57] V. Ferrari, T. Tuytelaars, and L. Van Gool, “Simultaneous object recognition and segmentation by image exploration,” in *Proc. ECCV*, pp. 40–54, 2004.

[58] V. Ferrari, T. Tuytelaars, and L. Van Gool, “Object detection by contour segment networks,” in *Proc. ECCV*, pp. 14–28, 2006.

[59] M. A. Fischler and R. A. Elschlager, “The representation and matching of pictorial structures,” *IEEE Trans. Computers*, vol. 22, no. 1, pp. 67–92, 1973.

[60] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, “Linear scale space,” *Journal of Mathematical Imaging and Vision*, vol. 4, pp. 325–351, 1994.

[61] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, 1993.

[62] W. Förstner and E. Gülch, “A fast operator for detection and precise location of distinct points, corners and centres of circular features,” in *Intercommission conference on Fast Processing of Photogrammetric Data*, pp. 281–305, 1987.

[63] D. Forsyth and J. Ponce, *Computer vision, a modern approach*, Prentice Hall, 2003.

[64] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[65] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.

[66] Y. Freund and R. Shapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[67] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," 1998.

[68] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Software*, vol. 3, pp. 209–226, 1977.

[69] M. Fussenegger, A. Opelt, A. Pinz, and P. Auer, "Object recognition using segmentation for feature detection," in *Proc. ICPR'04*, pp. 41–44, 2004.

[70] A. Ghosh and N. Petkov, "Robustness of shape descriptors to incomplete contour representations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1793–1804, 2005.

[71] J. Gibson, *The ecological approach to visual perception*, Lawrence Erlbaum, 1979.

[72] Z. Gigus and J. Malik, "Computing the aspect graph for line drawings of polyhedral objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 2, pp. 113–122, 1990.

[73] S. Gilles, *Robust description and matching of images*, Ph.D. thesis, University of Oxford, 1998.

[74] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Prentice Hall, 2nd ed., 2002.

[75] G. Granlund, "Organization of architectures for cognitive vision systems," in *Proceedings of Workshop on Cognitive Vision*, (Schloss Dagstuhl, Germany), October 2003.

[76] W. E. L. Grimson, *Object recognition by computer: The role of geometric constraints*, MIT Press, 1990.

[77] M. Hanheide, C. Bauckhage, and G. Sagerer, "Memory consistency validation in a cognitive vision system," in *Proc. ICPR*, pp. 459–462, 2004.

[78] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[79] R. M. Haralick and L. G. Shapiro, *Computer and robot vision*, Vol. II, Addison-Wesley, 1993.

[80] C. Harris and M. Stevens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, pp. 147–151, 1988.

[81] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, 2001.

[82] A. Holub and P. Perona, "A discriminative framework for modelling object classes," in *In Proc. Computer Vision and Pattern Recognition, CVPR*, 2005.

[83] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory*, vol. IT-8, pp. 179–187, 1962.

[84] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Computer Vision*, vol. 5, no. 2, pp. 195–212, 1990.

[85] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, Wiley, 2001.

[86] J. Illingworth and J. Kittler, “A survey of the Hough transform,” *CVGIP*, vol. 44, pp. 87–116, 1988.

[87] F. Jurie and C. Schmid, “Scale-invariant shape features for recognition of object categories,” in *Proc. CVPR*, 2004.

[88] T. Kadir and M. Brady, “Scale, saliency and image description,” *Int. J. Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[89] T. Kadir, A. Zisserman, and M. Brady, “An affine invariant salient region detector,” in *ECCV (1)*, pp. 228–241, 2004.

[90] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, Wiley, 1990.

[91] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image Descriptors,” in *Proc. CVPR*, pp. 506–513, 2004.

[92] L. Kitchen and A. Rosenfeld, “Gray-level corner detection,” *Pattern Recognition Letters*, vol. 1, pp. 95–102, 1982.

[93] J. J. Koenderink, “The structure of images,” *Biol. Cybern.*, vol. 50, pp. 363–370, 1984.

[94] J. J. Koenderink, “Scale-time,” *Biol. Cybern.*, vol. 58, pp. 159–162, 1988.

[95] J. J. Koenderink and A. J. van Doorn, “Representation of local geometry in the visual system,” *Biol. Cybern.*, vol. 55, pp. 367–375, 1987.

[96] M. P. Kumar, P. H. S. Torr, and A. Zisserman, “Extending pictorial structures for object recognition,” in *Proc. BMVC*, 2004.

[97] M. P. Kumar, P. H. S. Torr, and A. Zisserman, “Obj cut,” in *Proc. CVPR*, 2005.

[98] l. Van Gool, T. Moons, and D. Ungureanu, “Affine/photometric invariants for planar intensity patterns,” in *Proc. ECCV*, pp. 642–651, 1996.

[99] R. Laganiére, “Morphological corner detection,” in *Proc. 6th ICCV*, pp. 280–285, 1999.

[100] I. Laptev, “On space-time interest points,” *Int. J. Computer Vision*, vol. 64, no. 2/3, pp. 107–123, 2005.

[101] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proc. ICCV*, 2003.

[102] I. Laptev and T. Lindeberg, “Local descriptors for spatio-temporal recognition,” in *Proc. ECCV Workshop on Spatial coherernce for visual motion analysis*, 2004.

[103] I. Laptev and T. Lindeberg, “Velocity adaptation of space-time interest points,” in *Proc. ICPR*, 2004.

[104] L. J. Latecki, R. Lakämper, and U. Eckhardt, “Shape descriptors for non-rigid shapes with single closed contour,” in *Proc. CVPR*, pp. 424–429, 2000.

[105] S. Lazebnik, C. Schmid, and J. Ponce, “Semi-local affine parts for object recognition,” in *Proc. BMVC*, 2004.

[106] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model,” in *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[107] B. Leibe and B. Schiele, “Scale-invariant object categorization using a scale-adaptive mean-shift search,” in *Proc. DAGM Pattern Recognition Symposium*, 2004.

[108] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. CVPR*, 2005.

[109] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding: CVIU*, vol. 78, no. 1, pp. 99–118, 2000.

[110] T. K. Leung, M. C. Burl, and P. Perona, "Probabilistic affine invariants for recognition," in *Proc. CVPR*, pp. 678–684, 1998.

[111] T. Lindeberg, "Scale-space for discrete signals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 234–254, 1990.

[112] T. Lindeberg, *Scale space theory in computer vision*, Kluwer, 1994.

[113] T. Lindeberg, "Linear spatio-temporal scale-space," in *Proc Scale-Space'97: Scale-Space Theories in Computer Vision*, pp. 113–127, Springer, 1997.

[114] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.

[115] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 77–116, 1998.

[116] T. Lindeberg, "Time-recursive velocity-adapted spatio-temporal scale-space filters," in *Proc. ECCV*, pp. 52–67, Springer, 2002.

[117] T. Lindeberg, A. Akbarzadeh, and I. Laptev, "Galilean-diagonalized spactio-temporal interest points," in *Proc. ICPR*, 2004.

[118] T. Lindeberg and J.-O. Eklundh, "Scale-space primal sketch: Construction and experiments," *Image and Vision Computing*, vol. 10, no. 1, pp. 3–18, 1992.

[119] N. K. Logothetis and D. L. Sheinberg, "Visual object recognition," *Annu. Rev. Neurosci.*, vol. 19, pp. 577–621, 1996.

[120] D. G. Lowe, *Perceptual organization and visual cognition*, Kluwer, 1985.

[121] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[122] D. G. Lowe, "Object recognition from local scale-invariant features," in *In Proc. International Conference on Computer Vision*, pp. 1150–1157, 1999.

[123] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, no. 2, pp. 91–110, 2004.

[124] S. Maitra, "Moment invariants," *Proceedings of the IEEE*, vol. 67, no. 4, pp. 697–699, 1979.

[125] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Proc. CVPR*, 2005.

[126] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information.*, W. H. Freeman, 1982.

[127] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society London B*, vol. 207, pp. 187–217, 1980.

[128] A. M. Martinez and A. C. Kak, "PCA versus LDA," *PAMI*, vol. 23, no. 2, pp. 228–233, 2001.

[129] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. 13th BMVC*, pp. 384–393, 2002.

[130] T. Matsuyama and V. Hwang, “SIGMA: A framework for image understanding – integration of bottom-up and top-down analyses,” in *Proc. 9th IJCAI*, pp. 908–915, 1985.

[131] P. Meer, “Stochastic image pyramids,” *Computer Graphics and Image Processing*, vol. 45, pp. 269–294, 1989.

[132] K. Mikolajczyk, B. Leibe, and B. Schiele, “Local features for object class recognition,” in *Proc. ICCV*, pp. 1792–1799, 2005.

[133] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points,” in *Proc. ICCV*, pp. 525–531, 2001.

[134] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector,” in *ECCV (1)*, pp. 128–142, 2002.

[135] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” in *Proc. CVPR*, pp. 257–263, 2003.

[136] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *Int. J. Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[137] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[138] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[139] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, “Moment invariants for recognition under changing viewpoint and illumination,” *Computer Vision and Image Understanding*, vol. 94, no. 1–3, pp. 3–27, 2004.

[140] A. Montanvert, P. Meer, and A. Rosenfeld, “Hierarchical image analysis using irregular tessellations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 307–316, 1991.

[141] P. Moreels and P. Perona, “Common-frame model for object recognition,” in *Proc. NIPS*, 2004.

[142] J. Mundy and A. Zisserman, eds., *Geometric invariance in computer vision*, MIT Press, 1992.

[143] H. Murase and S. K. Nayar, “Visual learning and recognition of 3-d objects from appearance,” *Int.J. Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.

[144] K. Murphy, A. Torralba, and W. T. Freeman, “Using the forest to see the trees: A graphical model relating features, objects, and scenes,” in *Proc. NIPS*, 2003.

[145] M. Nadler and E. P. Smith, *Pattern recognition engineering*, Wiley, 1993.

[146] E. Nowak, F. Jurie, and B. Triggs, “Sampling strategies for bag-of-features image classification,” in *Proc. ECCV*, pp. 490–503, 2006.

[147] S. Obdržálek and J. Matas, “Object recognition using local affine frames on distinghuished regions,” in *Proc. 13th BMVC*, pp. 113–122, 2002.

[148] S. Obdržálek and J. Matas, “Image retrieval using local compact DCT-based representation,” in *Proc. 25th DAGM*, pp. 490–497, 2003.

[149] S. Obdržálek and J. Matas, “Sub-linear indexing for large scale object recognition,” in *Proc. BMVC*, pp. 1–10, 2005.

[150] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, “Learning and inference in parametric switching linear dynamic systems,” in *Proc. ICCV*, 2005.

[151] K. Ohba and K. Ikeuchi, "Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 1043–1047, 1997.

[152] A. Opelt, *Generic object recognition*, Ph.D. thesis, Graz University of Technology, 2006.

[153] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak hypotheses and boosting for generic object detection and recognition," in *ECCV'04*, (T. Pajdla and J. Matas, eds.), pp. 71–84, Springer, 2004.

[154] A. Opelt and A. Pinz, "Object localization with boosting and weak supervision for generic object recognition," in *Proc. SCIA*, 2005.

[155] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with Boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 416–431, 2006.

[156] A. Opelt, A. Pinz, and A. Zisserman, "A boundary-fragment-model for object detection," in *Proc. ECCV*, pp. 575–588, 2006.

[157] A. Opelt, A. Pinz, and A. Zisserman, "Incremental learning of object detectors using a visual shape alphabet," in *Proc. CVPR*, 2006. Best paper prize – runner up.

[158] S. Osher and N. Paragios, eds., *Geometric level set methods in imageing vision and graphics*, Springer, 2003.

[159] N. C. Oza and S. Russell, "Online bagging and boosting," in *Proc. Workshop on Artificial Intelligence and Statistics*, 2001.

[160] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.

[161] P. Perona, "A concise taxonomy of visual recognition," http://vasc.ri.cmu.edu/~hebert/04workshop/presentations/Perona-Sicily-Oct04.pdf, presented at the International Object Recognition Workshop, Sicily, October 1004, http://www.pascal-network.org/Workshops/IOR04/Programme/, pages visited May 10, 2005.

[162] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The FERET evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[163] A. Pinz and J.-P. Andreu, "Qualitative spatial reasoning to infer the camera position in generic object recognition," in *Proceedings ICPR'98*, pp. 770–773, 1998.

[164] A. Rosenfeld, *Multiresolution image processing and analysis*, Springer, 1984.

[165] P. M. Roth, H. Grabner, D. Skocaj, H. Bischof, and A. Leonardis, "Online conservative learning for person detection," in *Proc. VS-PETS Workshop at ICCV*, 2005.

[166] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?," in *Proc. CVPR*, 2004.

[167] S. Sarkar and K. Bowyer, *Computing perceptual organization in computer vision*, World Scientific, 1994.

[168] F. Scalzo and J. H. Piater, "Statistical learning of visual feature hierarchies," in *Proc. CVPR*, 2005.

[169] F. Schaffalitzky and A. Zisserman, "Multi-view matching of unordered image sets, or 'How do I organize my holiday snaps?'," in *ECCV (1)*, pp. 414–431, 2002.

[170] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 19, no. 5, pp. 530–535, 1997.

[171] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.

[172] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *Int. J. Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[173] G. Schweighofer, A. Opelt, and A. Pinz, "Improved object categorization by unsupervised object localization," in *Proc. Int. Workshop on Learning for Adaptable Visual Systems LAVS'04*, (St. Catherine's College, Cambridge), 2004.

[174] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele, "An evaluation of local shape-based features for pedestrian detection," in *Proc. BMVC*, 2005.

[175] A. Selinger and R. C. Nelson, "Improving appearance-based object recognition in cluttered background," in *Proc. ICPR*, pp. 1–8, 2000.

[176] M. E. Sereno, T. Trinath, M. Augath, and N. K. Logothetis, "Three-dimensional shape representation in monkey cortex," *Neuron*, vol. 36, pp. 635–652, 2002.

[177] T. Serre, L. Wolf, and T. Poggio, "A new biologically motivated framework for robust object recognition," in *In Proc. Computer Vision and Pattern Recognition, CVPR*, 2005.

[178] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, pp. 10–21, 1949.

[179] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[180] J. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR*, pp. 593–600, 1994.

[181] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *Proc. ICCV*, 2005.

[182] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint appearance, shape and context modeling for multi-class object recognitoin and segmentation," in *Proc. 9th ECCV*, pp. 1–15, 2006.

[183] A. Siebert, "Retrieval of gamma corrected images," *Pattern Recognition Letters*, vol. 22, no. 2, pp. 249–256, 2001.

[184] J. Sivic, B. C. Russell, A. A. Elfros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proc. ICCV*, 2005.

[185] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, pp. 1470–1477, Oct. 2003.

[186] S. M. Smith and J. M. Brady, "SUSAN – a new approach to low level image processing," *Int. J. Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[187] L. Stark and K. Bowyer, "Generic recognition through qualitative reasoning about 3-d shape and object function," in *Proc. CVPR*, pp. 251–256, 1991.

[188] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *Proc. ICCV*, 2005.

[189] B. M. ter Haar Romeny, ed., *Geometry-driven diffusion in computer vision*, Kluwer, 1994.

[190] J. Thureson and S. Carlsson, "Appearance based qualitative image description for object class recognition," in *Proc. ECCV*, pp. 518–529, 2004.

[191] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *Proc. CVPR*, 2004.

[192] Z. Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in *Proc. ICCV*, 2005.

[193] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[194] T. Tuytelaars, *Local, invariant features for registration and recognition*, Ph.D. thesis, K.U. Leuven, 2000.

[195] T. Tuytelaars and L. Van Gool, "Content-based image retrieval based on local affinely invariant regions," in *Proc. ICVS*, pp. 493–500, 1999.

[196] T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proc. 11th BMVC*, pp. 412–425, 2000.

[197] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Computer Vision*, vol. 59, no. 1, pp. 61–85, 2004.

[198] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *Proc. CVPR*, 2005.

[199] R. VanRullen and S. J. Thorpe, "Is it a bird? Is it a plane? Ultra-rapid visual categorisation of natural and artificial objects," *Perception*, vol. 30, pp. 655–668, 2001.

[200] V. N. Vapnik, *The nature of statistical learning theory*, Springer, 1999.

[201] M. M. Veloso, P. E. Rybski, and F. von Hundelshausen, "FOCUS: A generalized method for object discovery for robots that observe and interact with humans," in *Proc. HRI'06*, ACM, 2006.

[202] D. Vernon, "Cognitive vision – the development of a discipline," http://europa.eu.int/information_society/istevent/2004/cf/document.cfm?doc_id=568, page visited August 1st, 2006.

[203] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[204] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *ECCV (1)*, pp. 18–32, 2000.

[205] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. ICCV*, 2005.

[206] A. Witkin, "Scale-space filtering," in *Proc. 8th IJCAI*, pp. 1019–1022, 1983.

[207] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Computational Science & Engineering*, vol. 4, no. 4, pp. 10–21, 1997.

[208] R. A. Young, "The Gaussian derivative model for spatial vision: I. retinal mechanisms," *Spatial Vision*, vol. 2, pp. 273–293, 1987.

[209] A. L. Yuille and T. A. Poggio, “Scaling theorems for zero-crosssings,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 15–25, 1986.

[210] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras, “Object class recognition using multiple layer boosting with heterogeneous features,” in *Proc. CVPR*, pp. 323–330, 2005.