Robust object detection with interleaved categorization and segmentation

Author(s):
Leibe, Bastian; Leonardis, Aleš; Schiele, Bernt

Publication Date:
2008-05

Permanent Link:

https://doi.org/10.3929/ethz-b-000012377 →

Originally published in:

International Journal of Computer Vision 77(1-3), http://doi.org/10.1007/s11263-007-0095-3 →

Int J Comput Vis (2008) 77: 259–289
DOI 10.1007/s11263-007-0095-3

Robust Object Detection with Interleaved Categorization and Segmentation

Bastian Leibe · Aleš Leonardis · Bernt Schiele

Received: 2 September 2005 / Accepted: 18 September 2007 / Published online: 17 November 2007

Abstract This paper presents a novel method for detecting and localizing objects of a visual category in cluttered real-world scenes. Our approach considers object categorization and figure-ground segmentation as two interleaved processes that closely collaborate towards a common goal. As shown in our work, the tight coupling between those two processes allows them to benefit from each other and improve the combined performance.

The core part of our approach is a highly flexible learned representation for object shape that can combine the information observed on different training examples in a probabilistic extension of the Generalized Hough Transform. The resulting approach can detect categorical objects in novel images and automatically infer a probabilistic segmentation from the recognition result. This segmentation is then in turn used to again improve recognition by allowing the system to focus its efforts on object pixels and to discard misleading influences from the background. Moreover, the information from where in the image a hypothesis draws its support is employed in an MDL based hypothesis verification stage to resolve ambiguities between overlapping hypotheses and factor out the effects of partial occlusion.

An extensive evaluation on several large data sets shows that the proposed system is applicable to a range of different object categories, including both rigid and articulated objects. In addition, its flexible representation allows it to achieve competitive object detection performance already from training sets that are between one and two orders of magnitude smaller than those used in comparable systems.

Keywords Object categorization · Object detection · Segmentation · Clustering · Hough transform · Hypothesis selection · MDL

B. Leibe (✉)
Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
e-mail: leibe@vision.ee.ethz.ch

A. Leonardis
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
e-mail: alesl@fri.uni-lj.si

B. Schiele
Department of Computer Science, TU Darmstadt, Darmstadt, Germany
e-mail: schiele@informatik.tu-darmstadt.de

1 Introduction

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

Historically, this step of *figure-ground segmentation* has long been seen as an important and even necessary precursor for object recognition (Marr 1982). In this context, segmentation is mostly defined as a data driven, that is bottom-up,

process. However, except for cases where additional cues such as motion or stereo could be used, purely bottom-up approaches have so far been unable to yield figure-ground segmentations of sufficient quality for object categorization. This is also due to the fact that the notion and definition of what constitutes an object is largely task-specific and cannot be answered in an uninformed way. The general failure to achieve task-independent segmentation, together with the success of appearance-based methods to provide recognition results without prior segmentation, has led to the separation of the two areas and the further development of recognition independent from segmentation. It has been argued, however, both in computer vision (Bajcsy et al. 1990) and in human vision (Peterson 1994; Vecera and O'Reilly 1998; Needham 2001) that recognition and segmentation are heavily intertwined processes and that top-down knowledge from object recognition can and should be used for guiding the segmentation process.

In our work, we follow this inspiration by addressing object detection and segmentation not as separate entities, but as two closely collaborating processes. In particular, we present a local-feature based approach that combines both capabilities into a common probabilistic framework. As our experiments will show, the use of top-down segmentation improves the recognition results considerably.

In order to learn the appearance variability of an object category, we first build up a codebook of local appearances that are characteristic for (a particular viewpoint of) its member objects. This is done by extracting local features around interest points and grouping them with an agglomerative clustering scheme. As this initial clustering step will be applied to large data sets, an efficient implementation is crucial. We therefore evaluate different clustering methods and describe an efficient algorithm that can be used for the codebook generation process.

Based on this codebook, we then learn an *Implicit Shape Model* (ISM) that specifies where on the object the codebook entries may occur. As the name already suggests, we do not try to define an explicit model for all possible shapes a class object may take, but instead define "allowed" shapes implicitly in terms of which local appearances are consistent with each other. The advantages of this approach are its greater flexibility and the smaller number of training examples it needs to see in order to learn possible object shapes. For example, when learning to categorize articulated objects such as cows or pedestrians, our method does not need to see every possible articulation in the training set. It can combine the information of a front leg seen on one training instance with the information of a rear leg from a different instance to recognize a test image with a novel articulation, since both leg positions are consistent with the same object hypothesis. This idea is similar in spirit to approaches that represent novel objects by a combination of class prototypes (Jones and Poggio 1996), or of familiar object views (Ullman 1998). However, the main difference of our approach is that here the combination does not occur between entire exemplar objects, but through the use of local image features, which again allows a greater flexibility.

Directly connected to the recognition procedure, we derive a probabilistic formulation for the top-down segmentation problem, which integrates learned knowledge of the recognized category with the supporting information in the image. The resulting procedure yields a pixel-wise figure-ground segmentation as a result and extension of recognition. In addition, it delivers a per-pixel confidence estimate specifying how much this segmentation can be trusted.

The automatically computed top-down segmentation is then in turn used to improve recognition. First, it allows to only aggregate evidence over the object region and discard influences from the background. Second, the information from where in the image a hypothesis draws its support makes it possible to resolve ambiguities between overlapping hypotheses. We formalize this idea in a criterion based on the Minimum Description Length (MDL) principle. The resulting procedure constitutes a novel mechanism that allows to analyze scenes containing multiple objects in a principled manner. The whole approach is formulated in a scale-invariant manner, making it applicable in real-world situations where the object scale is often unknown.

We experimentally evaluate the different components of our algorithm and quantify the robustness of the resulting approach to object detection in cluttered real-world scenes. Our results show that the proposed scheme achieves good detection results for both rigid and articulated object categories while being robust to large scale changes.

This paper is structured as follows. The next section discusses related work. After that, Sect. 3 introduces our underlying codebook representation. The following three sections then present the main steps of the ISM approach for recognition (Sect. 4), top-down segmentation (Sect. 5), and segmentation-based verification (Sect. 6). Section 7 experimentally evaluates the different stages of the system and applies it to several challenging multi-scale test sets of different object categories, including cars, motorbikes, cows, and pedestrians. A final discussion concludes our work.

2 Related Work

In the following, we give an overview of current approaches to object detection and categorization, with a focus on the structural representations they employ. In addition, we document the recent transition from recognition to top-down segmentation, which has been developing into an area of active research.

2.1 Structure Representations for Object Categorization

A large class of methods match object structure by computing a cost term for the deformation needed to transform a prototypical object model to correspond with the image. Prominent examples of this approach include *Deformable Templates* (Yuille et al. 1989; Sclaroff 1997), *Morphable Models* (Jones and Poggio 1998), or *Shape Context Matching* (Belongie et al. 2002). The main difference between them lies in the way point correspondences are found and in the choice of energy function for computing the deformation cost (e.g. Euclidean distances, strain energy, thin plate splines, etc.). Cootes et al. (1998) extend this idea and characterize objects by means and modes of variation for both shape and texture. Their *Active Appearance Models* first warp the object to a mean shape and then estimate the combined modes of variation of the concatenated shape and texture models. For matching the resulting AAMs to a test image, they learn the relationship between model parameter displacements and the induced differences in the reconstructed model image. Provided that the method is initialized with a close estimate of the object's position and size, a good overall match to the object is typically obtained in a few iterations, even for deformable objects.

Wiskott et al. (1997) propose a different structural model known as *Bunch Graph*. The original version of this approach represents object structure as a graph of hand-defined locations, at which local jets (multidimensional vectors of simple filter responses) are computed. The method learns an object model by storing, for each graph node, the set ("bunch") of all jet responses that have been observed in this location on a hand-aligned training set. During recognition, only the strongest response is taken per location, and the joint model fit is optimized by an iterative elastic graph matching technique. This approach has achieved impressive results for face identification tasks, but an application to more object classes is made difficult by the need to model a set of suitable graph locations.

In contrast to those deformable representations, most classic object detection methods either use a monolithic object representation (Rowley et al. 1998; Papageorgiou and Poggio 2000; Dalal and Triggs 2005) or look for local features in fixed configurations (Schneiderman and Kanade 2004; Viola and Jones 2004). Schneiderman and Kanade (2004) express the likelihood of object and non-object appearance using a product of localized histograms, which represent the joint statistics of subsets of wavelet coefficients and their position on the object. The detection decision is made by a likelihood-ratio classifier. Multiple detectors, each specialized to a certain orientation of the object, are used to achieve recognition over a variety of poses, including frontal and profile faces and various views of passenger cars. Their approach achieves very good detection results on standard databases, but is computationally still relatively costly. Viola and Jones (2004) instead focus on building a speed-optimized system for face detection by learning a cascade of simple classifiers based on Haar wavelets. In recent years, this class of approaches has been shown to yield fast and accurate object detection results under real-world conditions (Torralba et al. 2004). However, a drawback of these methods is that since they do not explicitly model local variations in object structure (e.g. from body parts in different articulations), they typically need a large number of training examples in order to learn the allowed changes in global appearance.

One way to model these local variations is by representing objects as an assembly of parts. Mohan et al. (2001) use a set of hand-defined appearance parts, but learn an SVM-based configuration classifier for pedestrian detection. The resulting system performs significantly better than the original full-body person detector by (Papageorgiou and Poggio 2000). In addition, its component-based architecture makes it more robust to partial occlusion. Heisele et al. (2001) use a similar approach for component-based face detection. As an extension of Mohan et al.'s approach, their method also includes an automatic learning step for finding a set of discriminative components from user-specified seed points. More recently, several other part-classifier approaches have been proposed for pedestrian detection (Ronfard et al. 2002; Mikolajczyk et al. 2004; Wu and Nevatia 2005), also based on manually specified parts.

Burl et al. (1998) learn the assembly of hand-selected (appearance) object parts by modeling their joint spatial probability distribution. Weber et al. (2000) build on the same framework, but also learn the local parts and estimate their joint distribution. Fergus et al. (2003) extend this approach to scale-invariant object parts and estimate their joint spatial and appearance distribution. The resulting *Constellation Model* has been successfully demonstrated on several object categories. In its original form, it modeled the relative part locations by a fully connected graph. However, the complexity of the combined estimation step restricted this model to a relatively small number of (only 5–6) parts. In later versions, Fergus et al. (2005) therefore replaced the fully-connected graph by a simpler star topology, which can handle a far larger number of parts using efficient inference algorithms (Felzenszwalb and Huttenlocher 2005).

Agarwal et al. (2004) keep a larger number of object parts and apply a feature-efficient classifier for learning spatial configurations between pairs of parts. However, their learning approach relies on the repeated observation of cooccurrences between the same parts in similar spatial relations, which again requires a large number of training examples. Ullman et al. (2002) represent objects by a set of fragments that were chosen to maximize the information content with respect to an object class. Candidate fragments are extracted

at different sizes and from different locations of an initial set of training images. From this set, their approach iteratively selects those fragments that add the maximal amount of information about the object class to the already selected set, thus effectively resulting in a cover of the object. In addition, the approach automatically selects, for each fragment, the optimal threshold such that it can be reliably detected. For recognition, however, only the information which model fragments were detected is encoded in a binary-valued feature vector (similar to Agarwal and Roth's), onto which a simple linear classifier is applied without any additional shape model. The main challenge for this approach is that the complexity of the fragment selection process restricts the method to very low image resolutions (e.g. 14×21 pixels), which limits its applicability in practice.

Robustness to scale changes is one of the most important properties of any recognition system that shall be applied in real-world situations. Even when the camera location is relatively fixed, objects of interest may still exhibit scale changes of at least a factor of two, simply because they occur at different distances to the camera. It is therefore necessary that the recognition mechanism itself can compensate for a certain degree of scale variation. Many current object detection methods deal with the scale problem by performing an exhaustive search over all possible object positions and scales (Papageorgiou and Poggio 2000; Schneiderman and Kanade 2004; Viola and Jones 2004; Mikolajczyk et al. 2004; Dalal and Triggs 2005; Wu and Nevatia 2005). This exhaustive search imposes severe constraints, both on the detector's computational complexity and on its discriminance, since a large number of potential false positives need to be excluded. An opposite approach is to let the search be guided by image structures that give cues about the object scale. In such a system, an initial interest point detector tries to find structures whose extent can be reliably estimated under scale changes. These structures are then combined to derive a comparatively small number of hypotheses for object locations and scales. Only those hypotheses that pass an initial plausibility test need to be examined in detail. In recent years, a range of scale-invariant interest point detectors have become available which can be used for this purpose (Lindeberg 1998; Lowe 2004; Mikolajczyk et al. 2005b; Kadir and Brady 2001; Tuytelaars and van Gool 2004; Matas et al. 2002).

In our approach, we combine several of the above ideas. Our system uses a large number of automatically selected parts, based on the output of an interest point operator, and combines them flexibly in a star topology. Robustness to scale changes is achieved by employing scale-invariant interest points and explicitly incorporating the scale dimension in the hypothesis search procedure. The whole approach is optimized for efficient learning and accurate detection from small training sets.

2.2 From Recognition to Top-Down Segmentation

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

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

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

Tu et al. (2003) have proposed a system that integrates face and text detection with region-based segmentation of the full image. However, their focus is on segmenting images into meaningful regions, not on separating objects of interest from the background.

Yu and Shi (2003) and Ferrari et al. (2004) both present parallel segmentation and recognition systems. Yu and Shi formulate the segmentation problem in a graph theoretic framework that combines patch and pixel groupings, where the final solution is found using the Normalized Cuts criterion (Shi and Malik 1997). Ferrari et al. start from a small set of initial matches and then employ an iterative image exploration process that grows the matching region by searching for additional correspondences and segmenting the object in the process. Both methods achieve good segmentation results in cluttered real-world settings. However, both systems need to know the exact objects beforehand in order to extract their most discriminant features or search for additional correspondences.

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

3 Codebook Representations

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

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

In this section, we introduce the first level of such a representation. As basis, we use an idea inspired by the work of (Burl et al. 1998; Weber et al. 2000), and (Agarwal et al. 2004). We build up a vocabulary (in the following termed a *codebook*) of local appearances that are characteristic for a certain viewpoint of an object category by sampling local features that repeatedly occur on a set of training images of this category. Features that are visually similar are grouped together in an unsupervised clustering step. The result is a compact representation of object appearance in terms of which novel images can be expressed. When pursuing such an approach, however, it is important to represent uncertainty on all levels: while matching the unknown image content to the known codebook representation; and while accumulating the evidence of multiple such matches, e.g. for inferring the presence of the object.

Codebook representations have become a popular tool for object categorization recently, and many approaches use variations of this theme (Burl et al. 1998; Weber et al. 2000; Fergus et al. 2003; Li et al. 2003; Agarwal et al. 2004; Borenstein and Ullman 2002; Ullman et al. 2002; Felzenszwalb and Huttenlocher 2005). However, there are still large differences in how the grouping step is performed, how the matching uncertainty is represented, and how the codebook is later used for recognition. In the following, we describe our codebook generation procedure and review two popular methods for achieving the grouping step, namely k-means and agglomerative clustering. As the latter usually scales poorly to large data sets, we present an efficient average-link clustering algorithm which runs at the same time and space complexity as k-means. This algorithm is not based on an approximation, but computes the exact result, thus making it possible to use agglomerative clustering also for large-scale codebook generation. After the remaining stages of our recognition method have been introduced, Sect. 7.3 will then present an experimental comparison of the two clustering methods in the context of a recognition task.

3.1 Codebook Generation

We start by applying a scale-invariant interest point detector to obtain a set of informative regions for each image. By extracting features only from those regions, the amount of data to be processed is reduced, while the interest point detector's preference for certain structures assures that "similar" regions are sampled on different objects. Several different interest point detectors are available for this purpose. In this paper, we use and evaluate *Harris* (Harris and Stephens 1988), *Harris-Laplace* (Mikolajczyk et al. 2005b), *Hessian-Laplace* (Mikolajczyk et al. 2005b), and *Difference-of-Gaussian* (*DoG*) (Lowe 2004) detectors. We then represent the extracted image regions by a local descriptor. Again, several descriptor choices are available for this step. In this paper, we compare simple *Greyvalue*

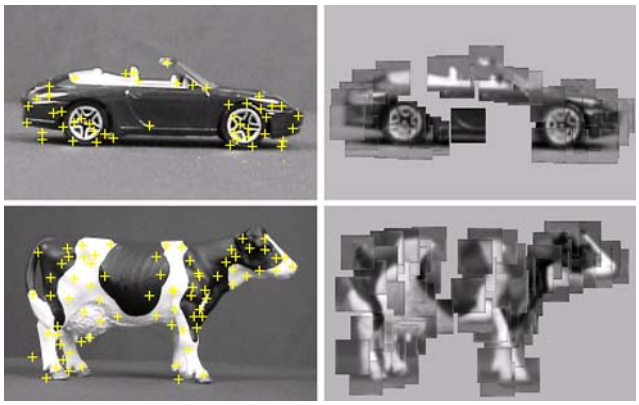


Fig. 1 Local information used in the codebook generation process: (*left*) interest points; (*right*) features extracted around the interest points (visualized by the corresponding image patches). In most of our experiments, between 50 and 200 features are extracted per object

Patches (Agarwal et al. 2004), *SIFT* (Lowe 2004), and *Local Shape Context* (Belongie et al. 2002; Mikolajczyk and Schmid 2005) descriptors. In order to develop the different stages of our approach, we will abstract from the concrete choice of region detector and descriptor and simply refer to the extracted local information by the term *feature*. Sections 7.5 and 7.6 will then systematically evaluate the different choices for the detectors and descriptors. Figure 1 shows the extracted features for two example images (in this case using *Harris* interest points). As can be seen from those examples, the sampled information provides a dense cover of the object, leaving out only uniform regions. This process is repeated for all training images, and the extracted features are collected.

Next, we group visually similar features to create a codebook of prototypical local appearances. In order to keep the representation as simple as possible, we represent all features in a cluster by their mean, the cluster center. Of course, a necessary condition for this is that the cluster center is a meaningful representative for the whole cluster. In that respect, it becomes evident that the goal of the grouping stage must not be to obtain the smallest possible number of clusters, but to ensure that the resulting clusters are visually compact and contain the same kind of structure. This is an important consideration to bear in mind when choosing the clustering method.

3.2 Clustering Methods

3.2.1 K-means Clustering

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

Many approaches employ k-means clustering because of its computational simplicity, which allows to apply it to very large data sets (Weber et al. 2000). Its time complexity is $O(Nk\ell d)$, where N is the number of data points of dimensionality d; k is the desired number of clusters; and ℓ is the number of iterations until the process converges. However, k-means clustering has several known deficiencies. Firstly, it requires the user to specify the number of clusters in advance. Secondly, there is no guarantee that the obtained clusters are visually compact. Because of the fixed value of k, some cluster centers may lie in-between several "real" clusters, so that the mean image is not representative of all grouped patches. Thirdly, k-means clustering is only efficient for small values of k; when applied to our task of finding a large number ($k \approx \frac{N}{10}$) of visually compact clusters, its asymptotic run-time becomes quadratic. Last but not least, the k-means procedure is only guaranteed to find a local optimum, so the results may be quite different from run to run.

3.2.2 Agglomerative Clustering

Other approaches therefore use agglomerative clustering schemes, which automatically determine the number of clusters by successively merging features until a cut-off threshold t on the cluster compactness is reached (Agarwal et al. 2004; Leibe and Schiele 2003). However, both the runtime and the memory requirements are often significantly higher for agglomerative methods. Especially the memory requirements impose a practical limit. The standard average-link algorithm, as found in most textbooks, requires an $O(N^2)$ similarity matrix to be stored. In practice, this means that the algorithm is only suitable for up to 15–25,000 input points on today's machines. After that, its space requirements outgrow the size of the available main memory, and the algorithm incurs detrimental page swapping costs.

Given the large amounts of data that need to be processed, an efficient implementation of the clustering algorithm is therefore not only a nice extension, but indeed crucial for its applicability. Fortunately, it turns out that for special choices of the clustering criterion and similarity measure, including the ones we are using, a more efficient algorithm is available that runs in $O(N^2 d)$ and needs only $O(N)$ space. Although the basic components of this algorithm are already more than 25 years old, it has so far been little known in the Computer Vision community. The following section will describe its derivation in more detail.

3.3 RNN Algorithm for Agglomerative Clustering

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

The algorithm is based on the construction of *reciprocal nearest neighbor* pairs (RNN pairs), that is of pairs of points a and b, such that a is b's nearest neighbor and vice versa (de Rham 1980; Benzécri 1982). It is applicable to clustering criteria that fulfill Bruynooghe's *reducibility property* (Bruynooghe 1977). This criterion demands that when two clusters c_i and c_j are agglomerated, the similarity of the merged cluster to any third cluster c_k may only decrease, compared to the state before the merging action:

$$\begin{aligned}&\operatorname{sim}(c_i, c_j) \geq \sup(\operatorname{sim}(c_i, c_k), \operatorname{sim}(c_j, c_k)) \quad \Rightarrow \\ &\sup(\operatorname{sim}(c_i, c_k), \operatorname{sim}(c_j, c_k)) \geq \operatorname{sim}(c_i \cup c_j, c_k). \end{aligned} \tag{1}$$

The reducibility property has the effect that the agglomeration of a reciprocal nearest-neighbor pair does not alter the nearest-neighbor relations of any other cluster. It is easy to see that this property is fulfilled, among others, for the *group average* criterion (regardless of the employed similarity measure) and the *centroid* criterion based on correlation (though not on Euclidean distances). Let $X = \{x^{(1)}, \ldots, x^{(N)}\}$ and $Y = \{y^{(1)}, \ldots, y^{(M)}\}$ be two clusters. Then those criteria are defined as

$$\text{group avg.:} \quad \operatorname{sim}(X, Y) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \operatorname{sim}(x^{(i)}, y^{(j)}), \tag{2}$$

$$\text{centroid:} \quad \operatorname{sim}(X, Y) = \operatorname{sim}\left(\frac{1}{N} \sum_{i=1}^{N} x^{(i)}, \frac{1}{M} \sum_{j=1}^{M} y^{(j)} \right). \tag{3}$$

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

This can be achieved by building a *nearest-neighbor chain* (Benzécri 1982). An NN-chain consists of an arbitrary point, followed by its nearest neighbor, which is again followed by its nearest neighbor from among the remaining points, and so on. It is easy to see that each NN-chain ends in an RNN pair. The strategy of the algorithm is thus to start with an arbitrary point (Algorithm 1, step (1)) and build up an NN-chain (2,3). As soon as an RNN pair is found, the corresponding clusters are merged if their similarity is above the cut-off threshold t; else the current chain is discarded (4). The reducibility property guarantees that when clusters are merged this way, the nearest-neighbor assignments stay valid for the remaining chain members, which can thus be reused for the next iteration. Whenever the current chain runs empty, a new chain is started with another random point (5). The resulting procedure is summarized in Algorithm 1.

Algorithm 1 The RNN algorithm for Average-Link clustering with nearest-neighbor chains.

```
// Start the chain L with a random point v ∈ V.
// All remaining points are kept in R.
last ← 0;  lastsim[0] ← 0
L[last] ← v ∈ V;  R ← V\v                                   (1)

while R ≠ ∅ do
  // Search for the next NN in R and retrieve its similarity sim.
  (s, sim) ← getNearestNeighbor(L[last], R)                 (2)

  if sim > lastsim[last] then
    // No RNNs → Add s to the NN chain
    last ← last + 1
    L[last] ← s;  R ← R\{s}
    lastsim[last] ← sim                                     (3)
  else
    // Found RNNs → agglomerate the last two chain links
    if lastsim[last] > t then
      s ← agglomerate(L[last], L[last − 1])
      R ← R ∪ {s}
      last ← last − 2                                       (4)
    else
      // Discard the current chain.
      last ← −1
    end if
  end if

  if last < 0 then
    // Initialize a new chain with another random point v ∈ R.
    last ← last + 1
    L[last] ← v ∈ R;  R ← R\{v}                             (5)
  end if
end while
```

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

When a new cluster is created by merging an RNN pair, its new similarity to other clusters needs to be recomputed. Applying an idea by (Day and Edelsbrunner 1984), this can be done in $O(N)$ space if the cluster similarity can be expressed in terms of centroids. In the following, we show that

this is the case for *group average* criteria based on correlation or Euclidean distances.

Let $\langle \cdots \rangle$ denote the inner product of a pair of vectors and μ_x, μ_y be the cluster means of X and Y. Then the group average clustering criterion based on correlation can be reformulated as

$$\mathrm{sim}(X, Y) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \langle x^{(i)}, y^{(j)} \rangle = \langle \mu_x, \mu_y \rangle, \tag{4}$$

which follows directly from the linearity property of the inner product.

Let μ_x, μ_y and σ_x^2, σ_y^2 be the cluster means and variances of X and Y. Then it can easily be verified that the group average clustering criterion based on Euclidean distances can be rewritten as

$$\begin{aligned} \mathrm{sim}(X, Y) &= -\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (x^{(i)} - y^{(j)})^2 \\ &= -(\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2). \end{aligned} \tag{5}$$

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

$$\mu_{new} = \frac{N\mu_x + M\mu_y}{N + M}, \tag{6}$$

$$\sigma_{new}^2 = \frac{1}{N + M} \left(N\sigma_x^2 + M\sigma_y^2 + \frac{NM}{N + M} (\mu_x - \mu_y)^2 \right). \tag{7}$$

Taken together, these steps result in an average-link clustering algorithm with $O(N^2 d)$ time and $O(N)$ space complexity. Among some other criteria, this algorithm is applicable to the group average criterion with correlation or Euclidean distances as similarity measure. As the method relies heavily on the search for nearest neighbors, its expected-time complexity can in some cases further be improved by using efficient NN-search techniques.

As a side note, we want to point out that for the cases considered in our experiments, where the number k of clusters is almost of the same order as N, average-link clustering and standard k-means have the same asymptotic time complexity. Since in our experiments between 10 and 25 iterations were necessary for k-means to converge, this number combines with the value of k to form an effective time complexity of $O(N^2 d)$.

Which clustering method is better suited for our application can only be evaluated in the context of an entire system. In Sect. 7.3, we therefore compare codebooks generated by k-means and agglomerative clustering for an object detection task. The results suggest that, although very similar detection performance can be achieved with both clustering methods, the lesser compactness of k-means clusters makes it more costly for later stages of the system to represent the matching uncertainty sufficiently well. In the following sections, we therefore use agglomerative clustering for codebook generation.

4 Object Categorization with an Implicit Shape Model

4.1 Shape Representation

As basic representation for our approach we introduce the Implicit Shape Model $ISM(C) = (\mathcal{C}, P_{\mathcal{C}})$, which consists of a class-specific alphabet $\mathcal{C}$ (the *codebook*) of local appearances that are prototypical for the object category, and of a spatial probability distribution $P_{\mathcal{C}}$ which specifies where each codebook entry may be found on the object.

We make two explicit design choices for the probability distribution $P_{\mathcal{C}}$. The first is that the distribution is defined independently for each codebook entry. This results in a star-shaped structural model, where the position of each local part is only dependent on the object center. The approach is flexible, since it allows to combine object parts during recognition that were initially observed on different training examples. In addition, it is able to learn recognition models from relatively small training sets, as our experiments will demonstrate. The second constraint is that the spatial probability distribution for each codebook entry is estimated in a non-parametric manner. This enables the method to model the true distribution in as much detail as the training data permits instead of making a possibly oversimplifying Gaussian assumption.

4.2 Learning the Shape Model

Let $\mathcal{C}$ be the learned appearance codebook, as described in the previous section. The next step is to learn the spatial probability distribution $P_{\mathcal{C}}$ (see Fig. 2 and Algorithm 2). For this, we perform a second iteration over all training images and match the codebook entries to the images. Here, we activate not only the best-matching codebook entry, but all entries whose similarity is above t, the cut-off threshold already used during agglomerative clustering. For every codebook entry, we store all positions it was activated in, relative to the object center.

By this step, we model the uncertainty in the codebook generation process. If a codebook is "perfect" in the sense that each feature can be uniquely assigned to exactly one cluster, then the result is equivalent to a nearest-neighbor matching strategy. However, it is unrealistic to expect such clean data in practical applications. We therefore keep each possible assignment, but weight it with the probability that this assignment is correct. It is easy to see that for similarity

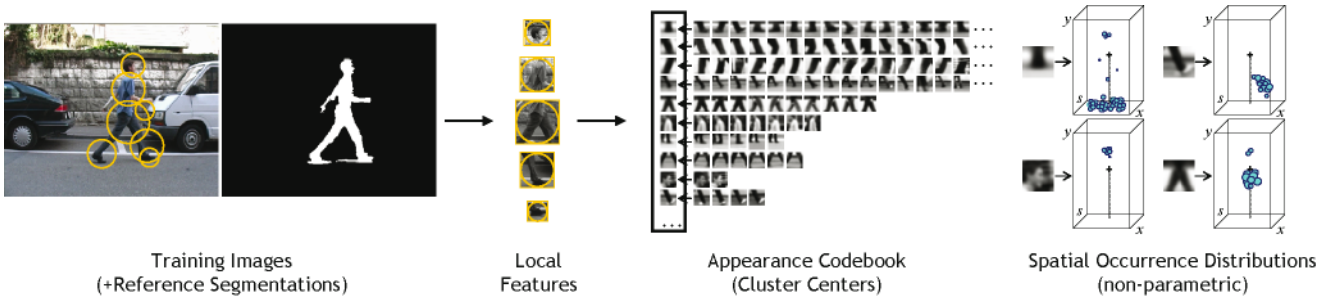


Fig. 2 The training procedure. Local features are extracted around interest points and clustered to form an appearance codebook. For each codebook entry, a spatial occurrence distribution is learned and stored in non-parametric form (as a list of occurrences)

Algorithm 2 The training procedure.

// Create an appearance codebook $\mathcal{C}$.
$\mathcal{F} \leftarrow \emptyset$ *// Initialize the set of feature vectors* $\mathcal{F}$
for all training images **do**
  Apply the interest point detector.
  for all interest regions $\ell_k = (\ell_x, \ell_y, \ell_s)$ with descriptors f_k **do**
    $\mathcal{F} \leftarrow \mathcal{F} \cup f_k$
  end for
end for
Cluster $\mathcal{F}$ with cut-off threshold t and keep cluster centers $\mathcal{C}$.

// Compute occurrences Occ.
for all codebook entries $\mathcal{C}_i$ **do**
  $Occ[i] \leftarrow \emptyset$ *// Initialize occurrences for codebook entry* $\mathcal{C}_i$
end for
for all training images **do**
  Let (c_x, c_y) be the object center at a reference scale.
  Apply the interest point detector.
  for all interest regions $\ell_k = (\ell_x, \ell_y, \ell_s)$ with descriptors f_k **do**
    for all codebook entries $\mathcal{C}_i$ **do**
      if $sim(\mathcal{C}_i, f_k) \geq t$ **then**
        // Record an occurrence of codebook entry $\mathcal{C}_i$
        $Occ[i] \leftarrow Occ[i] \cup (c_x - \ell_x, c_y - \ell_y, \ell_s)$
      end if
    end for
  end for
end for

scores smaller than t, the probability that this patch could have been assigned to the cluster during the codebook generation process is zero; therefore we do not need to consider those matches. The stored occurrence locations, on the other hand, reflect the spatial distribution of a codebook entry over the object area in a non-parametric form. Algorithm 2 summarizes the training procedure.

4.3 Recognition Approach

Figure 3 illustrates the following recognition procedure. Given a new test image, we again apply an interest point detector and extract features around the selected locations. The extracted features are then matched to the codebook to activate codebook entries using the same mechanism as described above. From the set of all those matches, we collect consistent configurations by performing a Generalized Hough Transform (Hough 1962; Ballard 1981; Lowe 2004). Each activated entry casts votes for possible positions of the object center according to the learned spatial distribution $P_{\mathcal{C}}$. Consistent hypotheses are then searched as local maxima in the voting space. When pursuing such an approach, it is important to avoid quantization artifacts. In contrast to usual practice (e.g. Lowe 1999), we therefore do not discretize the votes, but keep their original, continuous values. Maxima in this continuous space can be accurately and efficiently found using Mean-Shift Mode Estimation (Cheng 1995; Comaniciu and Meer 2002). Once a hypothesis has been selected, all patches that contributed to it are collected (Fig. 3(bottom)), thereby visualizing what the system reacts to. As a result, we get a representation of the object including a certain border area. This representation can optionally be further refined by sampling more local features. The backprojected response will later serve as the basis for computing a category-specific segmentation, as described in Sect. 5.

4.3.1 Probabilistic Hough Voting

In the following, we cast the voting procedure into a probabilistic framework (Leibe and Schiele 2003; Leibe et al. 2004). Let f be our evidence, an extracted image feature observed at location ℓ. By matching it to the codebook, we obtain a set of valid interpretations $\mathcal{C}_i$ with probabilities $p(\mathcal{C}_i | f, \ell)$. If a codebook cluster matches, it casts votes for different object positions. That is, for every $\mathcal{C}_i$, we can obtain votes for several object categories/viewpoints o_n and positions x, according to the learned spatial distribution $p(o_n, x | \mathcal{C}_i, \ell)$. Formally, this can be expressed by the following marginalization:

$$p(o_n, x | f, \ell) = \sum_i p(o_n, x | f, \mathcal{C}_i, \ell)\, p(\mathcal{C}_i | f, \ell). \quad (8)$$

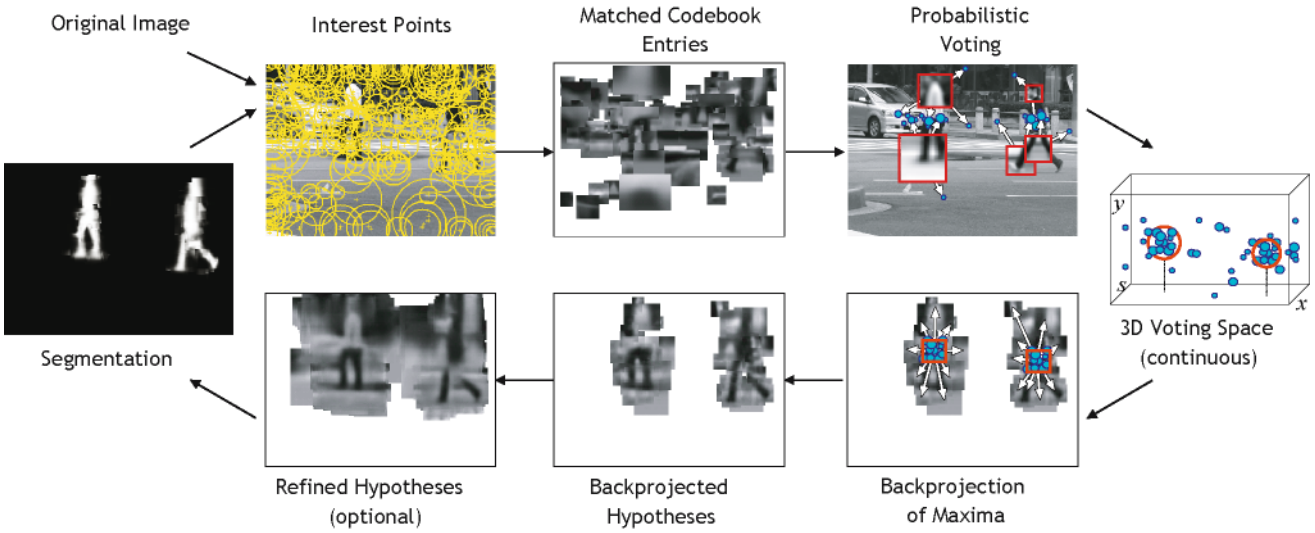


Fig. 3 The recognition procedure. Local features are extracted around interest points and compared to the codebook. Matching patches then cast probabilistic votes, which lead to object hypotheses that can optionally be later refined by sampling more features. Based on the backprojected hypotheses, we then compute a category-specific segmentation

Since we have replaced the unknown image feature by a known interpretation, the first term can be treated as independent from f. In addition, we match patches to the codebook independent of their location. The equation thus reduces to

$$p(o_n, x|f, \ell) = \sum_i p(o_n, x|\mathcal{C}_i, \ell) p(\mathcal{C}_i|f), \tag{9}$$

$$= \sum_i p(x|o_n, \mathcal{C}_i, \ell) p(o_n|\mathcal{C}_i, \ell) p(\mathcal{C}_i|f). \tag{10}$$

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

When casting votes for the object center, the object scale is treated as a third dimension in the voting space (Leibe and Schiele 2004). If an image feature found at location $(x_{img}, y_{img}, s_{img})$ matches to a codebook entry that has been observed at position $(x_{occ}, y_{occ}, s_{occ})$ on a training image, it votes for the following coordinates:

$$x_{vote} = x_{img} - x_{occ}(s_{img}/s_{occ}), \tag{11}$$

$$y_{vote} = y_{img} - y_{occ}(s_{img}/s_{occ}), \tag{12}$$

$$s_{vote} = (s_{img}/s_{occ}). \tag{13}$$

Thus, the vote distribution $p(x|o_n, \mathcal{C}_i, \ell)$ is obtained by casting a vote for each stored observation from the learned occurrence distribution $P_{\mathcal{C}}$. The ensemble of all such votes together is then used to obtain a non-parametric probability density estimate for the position of the object center.

In order to avoid a systematic bias, we require that each sampled feature have the same a-priori weight. We therefore need to normalize the vote weights such that both the $p(\mathcal{C}_i|f)$ and the $p(x|o_n, \mathcal{C}_i, \ell)$ integrate to one. In our experiments, we spread the weight $p(\mathcal{C}_i|f)$ uniformly over all valid patch interpretations (setting $p(\mathcal{C}_i|f) = \frac{1}{|\mathcal{C}^*|}$, with $|\mathcal{C}^*|$ the number of matching codebook entries), but it would also be possible to let the $p(\mathcal{C}_i|f)$ distribution reflect the relative matching scores, e.g. by using a Gibbs-like distribution $p(\mathcal{C}_i|f) = \frac{1}{Z} \exp\{-d(\mathcal{C}_i, f)^2/T\}$ with a suitable normalization constant Z. The complete voting procedure is summarized in Algorithm 3.

4.3.2 Scale-Adaptive Hypothesis Search

Next, we need to find hypotheses as maxima in the voting space. For computational efficiency, we employ a two-stage search strategy (see Fig. 4 and Algorithm 4). In a first stage, votes are collected in a binned 3D Hough accumulator array in order to quickly find promising locations. Candidate maxima from this first stage are then refined in the second stage using the original (continuous) 3D votes.

Intuitively, the score of a hypothesis $h = (o_n, x)$ can be obtained by marginalizing over all features that contribute to this hypothesis

$$p(o_n, x) = \sum_k p(o_n, x|f_k, \ell_k) p(f_k, \ell_k), \tag{14}$$

where $p(f_k, \ell_k)$ is an indicator variable specifying which features (f_k, ℓ_k) have been sampled by the interest point detector. However, in order to be robust to intra-class variation,

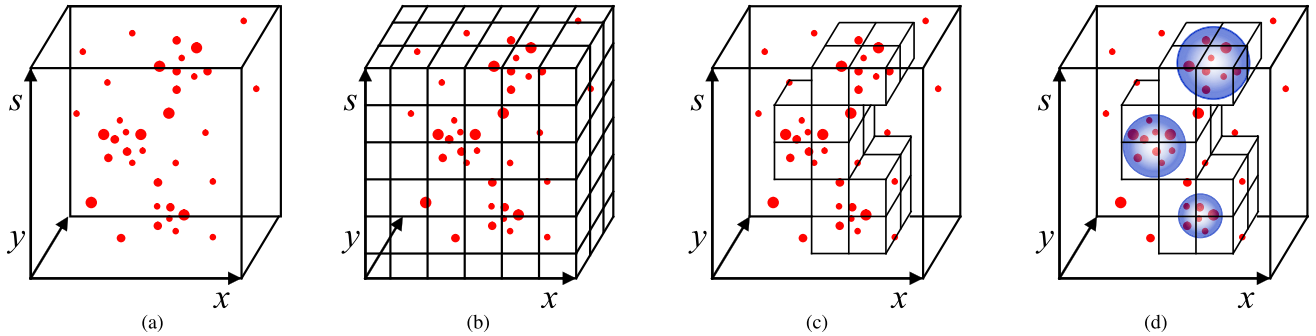


Fig. 4 Visualization of the scale-invariant voting procedure. The continuous votes (**a**) are first collected in a binned accumulator array (**b**), where candidate maxima can be quickly detected (**c**). The exact Mean-Shift search is then performed only in the regions immediately surrounding those candidate maxima (**d**)

Algorithm 3 The ISM vote generation algorithm.

// Initialize the set of probabilistic votes $\mathcal{V}$.
$\mathcal{V} \leftarrow \emptyset$
Apply the interest point detector to the test image.
for all interest regions $\ell_k = (\ell_x, \ell_y, \ell_s)$ with descriptors f_k **do**
  // Initialize the set of matches $\mathcal{M}$
  $\mathcal{M} \leftarrow \emptyset$
  // Record all matches to the codebook
  for all codebook entries $\mathcal{C}_i$ **do**
    if $sim(f_k, \mathcal{C}_i) \geq t$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup (i, \ell_x, \ell_y, \ell_s)$ *// Record a match*
    end if
  end for
  for all matching codebook entries $\mathcal{C}_i^*$ **do**
    $p(\mathcal{C}_i^* | f_k) \leftarrow \frac{1}{|\mathcal{M}|}$ *//Set the match weight*
  end for
  // Cast the votes
  for all matches $(i, \ell_x, \ell_y, \ell_s) \in \mathcal{M}$ **do**
    for all occurrences $occ \in Occ[i]$ of codebook entry $\mathcal{C}_i$ **do**
      //Set the vote location
      $x \leftarrow (\ell_x - occ_x \frac{\ell_s}{occ_s}, \ell_y - occ_y \frac{\ell_s}{occ_s}, \frac{\ell_s}{occ_s})$
      //Set the occurrence weight
      $p(o_n, x | \mathcal{C}_i, \ell) \leftarrow \frac{1}{|Occ[i]|}$
      // Cast a vote (x, w, occ, ℓ) *for position* x *with weight* w
      $w \leftarrow p(o_n, x | \mathcal{C}_i, \ell) p(\mathcal{C}_i | f_k)$
      $\mathcal{V} \leftarrow \mathcal{V} \cup (x, w, occ, \ell)$
    end for
  end for
end for

Algorithm 4 The scale-adaptive hypothesis search algorithm.

// Sample the voting space $\mathcal{V}$ *in a regular grid to obtain*
// promising starting locations.
for all grid locations x **do**
  $score(x) \leftarrow applyMSMEKernel(K, x)$
end for

// Refine the local maxima using MSME with a scale-adaptive
// kernel K. *Keep all maxima above a threshold* θ.
for all grid locations x **do**
  if x is a local maximum in a 3×3 neighborhood **then**
    // Apply the MSME search
    repeat
      $score \leftarrow 0, x_{new} \leftarrow (0, 0, 0), sum \leftarrow 0$
      for all votes $(x_k, w_k, occ_k, \ell_k)$ **do**
        if x_k is inside $K(x)$ **then**
          $score \leftarrow score + w_k K(\frac{x - x_k}{b(x)})$
          $x_{new} \leftarrow x_{new} + x_k K(\frac{x - x_k}{b(x)})$
          $sum \leftarrow sum + K(\frac{x - x_k}{b(x)})$
        end if
      end for
      $score \leftarrow \frac{1}{V_b(x)} score$
      $x \leftarrow \frac{1}{sum} x_{new}$
    until convergence
    if $score \geq \theta$ **then**
      Create hypothesis h for position x.
    end if
  end if
end for

we have to tolerate small shape deformations. We therefore formulate the search in a Mean-Shift framework with the following kernel density estimate:

$$\hat{p}(o_n, x) = \frac{1}{V_b} \sum_k \sum_j p(o_n, x_j | f_k, \ell_k) K\left(\frac{x - x_j}{b}\right) \quad (15)$$

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

(a) orig. image (b) interest regions (c) matched features (d) voting space (e) backprojected hypo.

Fig. 5 Intermediate results during the recognition process. (**a**) original image; (**b**) sampled interest regions; (**c**) extracted features that could be matched to the codebook; (**d**) probabilistic votes; (**e**) support of the strongest hypothesis. (Note that the voting process takes place in a continuous space. The votes are just discretized for visualization)

From the literature, it is also known that the performance of the Mean-Shift procedure depends critically on a good selection for the kernel bandwidth b. Various approaches have been proposed to estimate the optimal bandwidth directly from the data (e.g. Comaniciu et al. 2001; Collins 2003). In our case, however, we have an intuitive interpretation for the bandwidth as a search window for the position of the object center. As the object scale increases, the *relative errors* introduced by (11–13) cause votes to be spread over a larger area around the hypothesized object center and thus reduce their density in the voting space. As a consequence, the kernel bandwidth should also increase in order to compensate for this effect. We can thus make the bandwidth dependent on the scale coordinate and obtain the following *balloon density estimator* (Comaniciu et al. 2001):

$$\hat{p}(o_n, x) = \frac{1}{V_b(x)} \sum_k \sum_j p(o_n, x_j | f_k, \ell_k) K\left(\frac{x - x_j}{b(x)}\right). \tag{16}$$

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

4.4 Summary

We have thus formulated the multi-scale object detection problem as a probabilistic Hough Voting procedure from which hypotheses are found by a scale-adaptive Mean-Shift search. Figure 5 illustrates the different steps of the recognition procedure on a real-world example. For this example, the system was trained on 119 car images taken from the *LabelMe* database (Russell et al. 2005). When presented with the test image, the system applies a DoG interest point detector and extracts a total of 437 features (Fig. 5(b)). However, only about half of them contain relevant structure and pass the codebook matching stage (Fig. 5(c)). Those features then cast probabilistic votes, which are collected in the voting space. As a visualization of this space in Fig. 5(d) shows, only few features form a consistent configuration. The system searches for local maxima in the voting space and returns the correct detection as strongest hypothesis. By backprojecting the contributing votes, we retrieve the hypothesis's support in the image (Fig. 5(e)), which shows that the system's reaction has indeed been produced by local structures on the depicted car.

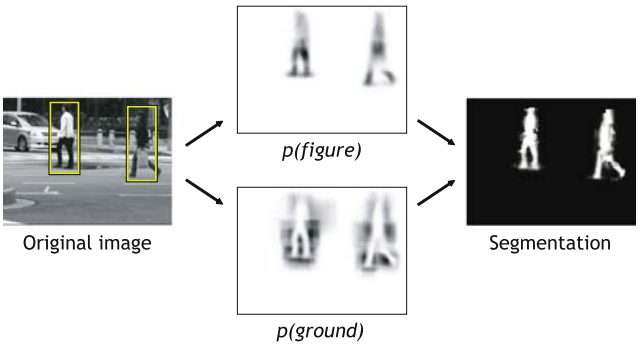


Fig. 6 Visualization of the top-down segmentation procedure. For each hypothesis h, we compute a per-pixel *figure* probability map $p(figure|h)$ and a *ground* probability map $p(ground|h)$. The final segmentation is then obtained by building the likelihood ratio between *figure* and *ground*

5 Top-Down Segmentation

The backprojected hypothesis support already provides a rough indication where the object is in the image. As the sampled patches still contain background structure, however, this is not a precise segmentation yet. On the other hand, we have expressed the a-priori unknown image content in terms of a learned codebook; thus, we know more about the semantic interpretation of the matched patches for the target object. In the following, we will show how this information can be used to infer a pixel-wise figure-ground segmentation of the object (Fig. 6).

In order to learn this top-down segmentation, our approach requires a reference figure-ground segmentation for the training images. While this additional information might not always be available, we will demonstrate that it can be

used to improve recognition performance significantly, as our experimental results in Sect. 7 will show.

5.1 Theoretical Derivation

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

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

$$p(f, \ell|o_n, x) = \frac{p(o_n, x|f, \ell)p(f, \ell)}{p(o_n, x)}, \tag{17}$$

$$= \frac{\sum_i p(o_n, x|\mathcal{C}_i, \ell)p(\mathcal{C}_i|f)p(f, \ell)}{p(o_n, x)} \tag{18}$$

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

$$p(\mathbf{p} = figure|o_n, x) = \sum_{\mathbf{p}\in(f,\ell)} p(\mathbf{p} = figure|o_n, x, f, \ell)p(f, \ell|o_n, x) \tag{19}$$

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

$$\begin{aligned} p(\mathbf{p} = fig.|o_n, x) &= \sum_{\mathbf{p}\in(f,\ell)} \sum_i p(\mathbf{p} = fig.|o_n, x, f, \mathcal{C}_i, \ell)p(f, \mathcal{C}_i, \ell|o_n, x) \\ &= \sum_{\mathbf{p}\in(f,\ell)} \sum_i p(\mathbf{p} = fig.|o_n, x, \mathcal{C}_i, \ell) \\ &\quad \times \frac{p(o_n, x|\mathcal{C}_i, \ell)p(\mathcal{C}_i|f)p(f, \ell)}{p(o_n, x)}. \end{aligned} \tag{20}$$

This means that for every pixel, we effectively build a weighted average over all segmentations stemming from patches containing that pixel. The weights correspond to the patches' respective contributions to the object hypothesis. We further assume uniform priors for $p(f, \ell)$ and $p(o_n, x)$, so that these elements can be factored out of the equations. For the *ground* probability, the result is obtained in a similar fashion:

$$p(\mathbf{p} = ground|o_n, x) = \sum_{\mathbf{p}\in(f,\ell)} \sum_i (1 - p(\mathbf{p}{=}fig.|o_n, x, \mathcal{C}_i, \ell))p(f, \mathcal{C}_i, \ell|o_n, x). \tag{21}$$

The most important part in this formulation is the per-pixel segmentation information $p(\mathbf{p} = figure|o_n, x, \mathcal{C}_i, \ell)$, which is only dependent on the matched codebook entry, no longer on the image feature. In Borenstein and Ullman's approach (Borenstein and Ullman 2002) a fixed segmentation mask is stored for each codebook entry. Applied to our framework, this would be equivalent to using a reduced probability $p(\mathbf{p} = figure|\mathcal{C}_i, o_n)$. In our approach, however, we remain more general and keep a separate segmentation mask for every recorded *occurrence position* of each codebook entry (extracted from the training images at the location and scale of the corresponding interest region and stored as a 16×16 pixel mask). We thus take advantage of the full probability $p(\mathbf{p} = figure|o_n, x, \mathcal{C}_i, \ell)$. As a result, the same local image structure can indicate a solid area if it is in the middle of e.g. a cow's body, and a strong border if it is part of a leg. Which option is finally selected depends on the current hypothesis and its accumulated support from other patches. However, since at this point only votes are considered that support a common hypothesis, it is ensured that only consistent interpretations are used for the segmentation.

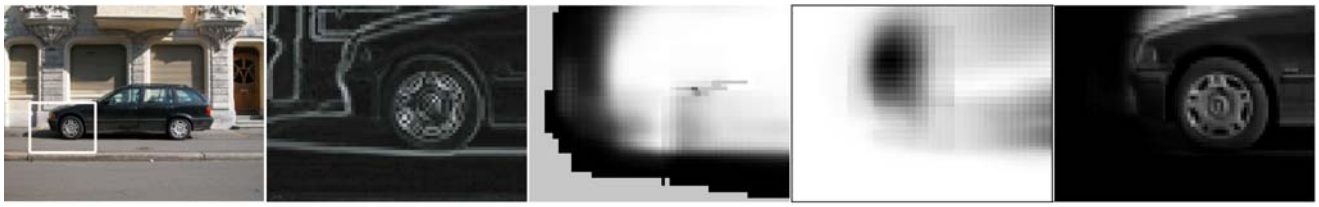
In order to obtain a segmentation of the whole image from the figure and ground probabilities, we build the likelihood ratio for every pixel:

$$L = \frac{p(\mathbf{p} = figure|o_n, x) + \epsilon}{p(\mathbf{p} = ground|o_n, x) + \epsilon}. \tag{22}$$

Figure 6 and Algorithm 5 summarize the top-down segmentation procedure. As a consequence of our non-parametric

Algorithm 5 The top-segmentation algorithm.

// Given: hypothesis h and supporting votes $\mathcal{V}_h$.
for all supporting votes $(x, w, occ, \ell) \in \mathcal{V}_h$ **do**
  Let img_{mask} be the segmentation mask corresponding to occ.
  Let $\mathtt{sz}$ be the size at which the interest region ℓ was sampled.
  Rescale img_{mask} to $\mathtt{sz}$.
  $u_0 \leftarrow (\ell_x - \frac{1}{2}\mathtt{sz})$
  $v_0 \leftarrow (\ell_y - \frac{1}{2}\mathtt{sz})$
  for all $u \in [0, \mathtt{sz} - 1]$ **do**
    for all $v \in [0, \mathtt{sz} - 1]$ **do**
      $img_{pfig}(u - u_0, v - v_0) \mathrel{+}= w \cdot img_{mask}(u, v)$
      $img_{pgnd}(u - u_0, v - v_0) \mathrel{+}= w \cdot (1 - img_{mask}(u, v))$
    end for
  end for
end for



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

Fig. 7 An example where object knowledge compensates for missing edge information

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

Fig. 8 Segmentation result of a partially occluded car. The system is able to segment out the pedestrian, because it does not contribute to the car hypothesis

representation for $P_{\mathcal{C}}$, the resulting algorithm is very simple and can be efficiently computed on the GPU (in our current implementation taking only 5–10 ms per hypothesis).

Figures 7 and 8 show two example segmentations of cars,[1] together with $p(\mathbf{p} = \mathit{figure}|o_n, x)$, the system's confidence in the segmentation result (the darker a pixel, the higher its probability of being *figure*; the lighter it is, the higher its probability of being *ground*). Those examples highlight some of the advantages a top-down segmentation can offer compared to bottom-up and gradient-based approaches. At the bottom of the car shown in Fig. 7, there is no visible border between the black car body and the dark shadow underneath. Instead, a strong shadow line extends much further to the left of the car. The proposed algorithm can compensate for that since it has learned that if a codebook entry matches in this position relative to the object center, it must contain the car's border. Since at this point only those patch interpretations are considered that are consistent with the object hypothesis, the system can infer the missing contour. Figure 8 shows another interesting case. Even though the car in the image is partially occluded by a pedestrian, the algorithm correctly finds it. Backprojecting the hypothesis yields a good segmentation of the car, without the occluded area. The system is able to segment out the pedestrian, because the corresponding region does not contribute to the car hypothesis. This capability is very hard to achieve for a system purely based on pixel-level discontinuities.

[1]For better visualization, the segmentation images in Figs. 7(c) and 8(c) show not L but $\mathit{sigmoid}(\log L)$.

6 Segmentation-Based Hypothesis Verification

6.1 Motivation

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

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

At the same time, this idea allows to compensate for a systematic bias in the initial voting scheme. The probabilistic votes are constructed on the principle that each feature has the same weight. This leads to a competitive advantage for hypotheses that contain more matched features simply

Fig. 9 (Color online) (*left*) Two examples for overlapping hypotheses (*in red*); (*middle*) $p(\mathbf{p} = figure|h)$ probabilities for the correct and (*right*) for the overlapping hypotheses. The overlapping hypothesis in the above example is almost fully explained by the two correct detections, while the one in the lower example obtains additional support from a different region in the image

because their area was more densely sampled by the interest point detector. Normalizing a hypothesis's score by the number of contributing features, on the other hand, would not produce the desired results, because the corresponding image patches can overlap and may also contain background structure. By accumulating evidence now over the *figure* pixels, the verification stage removes this overcounting bias. Using this principle, each pixel has the same potential influence, regardless of how many sampled patches it is contained in.

Finally, this strategy makes it possible to resolve ambiguities from overlapping hypotheses in a principled manner. When applying the recognition procedure to real-world test images, a large number of the initial false positives are due to secondary hypotheses which overlap part of the object (see Fig. 9). This is a common problem in object detection that is particularly prominent in scenes containing multiple objects. Generating such secondary hypotheses is a desired property of a recognition algorithm, since it allows the method to cope with partial occlusions. However, if enough support is present in the image, the secondary detections should be suppressed in favor of other hypotheses that better explain the image. Usually, this problem is solved by introducing a bounding box criterion and rejecting weaker hypotheses based on their overlap. However, such an approach may lead to missed detections, as the second example in Fig. 9 shows. Here the overlapping hypothesis really corresponds to a second car, which would be rejected by the simple bounding box criterion.

Again, using the top-down segmentation our system can improve on this and exactly quantify how much support the overlapping region contains for each hypothesis. In particular, this permits us to detect secondary hypotheses, which draw all their support from areas that are already better explained by other hypotheses, and distinguish them from true overlapping objects. In the following, we derive a criterion based on the principle of Minimal Description Length (MDL), which combines all of those motivations.

6.2 MDL Formulation

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

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

$$S_h = K_0 S_{area} - K_1 S_{model} - K_2 S_{error}. \tag{23}$$

In this formulation, S_{area} corresponds to the number N of pixels that can be explained by h; S_{error} denotes the cost for describing the error made by this explanation; and S_{model} describes the model complexity. Since objects at different scales take up different portions of the image, we make the model cost dependent on the *expected area* A_s an object occupies at a certain scale.[2] As an estimate for the error cost we collect, over all pixels that belong to the segmentation of h, the negative *figure* log-likelihoods:

$$\begin{aligned} S_{error} &= -\log \prod_{\mathbf{p}\in Seg(h)} p(\mathbf{p} = fig.|h)) \\ &= -\sum_{\mathbf{p}\in Seg(h)} \log p(\mathbf{p} = fig.|h) \\ &= \sum_{\mathbf{p}\in Seg(h)} \sum_{n=1}^{\infty} \frac{1}{n}(1 - p(\mathbf{p} = fig.|h))^n \\ &\approx \sum_{\mathbf{p}\in Seg(h)} (1 - p(\mathbf{p} = fig.|h)). \end{aligned} \tag{24}$$

Here we use a first-order approximation for the logarithms, which we found to be more stable with respect to outliers and unequal sampling, since it avoids the logarithm's singularity around zero. In effect, the resulting error term can be understood as a sum over all pixels allocated to a hypothesis h of the probabilities that this allocation was incorrectly made.

[2] When dealing with only one object category, the true area A_s can be replaced by the simpler term s^2, since the expected area grows quadratically with the object scale and the constant K_1 can be set to incorporate the proportionality factor. However, when multiple categories or different views of the same object category are searched for, the model cost needs to reflect their relative size differences.

The constants K_0, K_1, and K_2 are related to the average cost of specifying the segmented object area, the model, and the error, respectively. They can be determined on a purely information-theoretical basis (in terms of bits), or they can be adjusted in order to express the preference for a particular type of description. In practice, we only need to consider the relative savings between different combinations of hypotheses. Thus, we can divide (23) by K_0 and, after some simplification steps, we obtain

$$\begin{aligned} S_h &= -\frac{K_1}{K_0} + \left(1 - \frac{K_2}{K_0}\right)\frac{N}{A_s} \\ &\quad + \frac{K_2}{K_0}\frac{1}{A_s}\sum_{\mathbf{p}\in Seg(h)} p(\mathbf{p} = fig.|h) \\ &= -\kappa_1 + (1-\kappa_2)\frac{N}{A_s} \\ &\quad + \kappa_2\frac{1}{A_s}\sum_{\mathbf{p}\in Seg(h)} p(\mathbf{p} = fig.|h), \qquad (25) \\ &= -\kappa_1 + \frac{1}{A_s}\sum_{\mathbf{p}\in Seg(h)} ((1-\kappa_2) + \kappa_2 p(\mathbf{p} = fig.|h)). \qquad (26) \end{aligned}$$

This leaves us with two parameters: $\kappa_2 = \frac{K_2}{K_0}$, which encodes the relative importance that is assigned to the support of a hypothesis, as opposed to the area it explains; and $\kappa_1 = \frac{K_1}{K_0}$, which specifies the total weight a hypothesis must accumulate in order to provide any savings. Essentially, (26) formulates the merit of a hypothesis as the sum over its pixel assignment likelihoods, together with a regularization term κ_2 to compensate for unequal sampling and a counterweight κ_1. In our experiments, we leave κ_2 at a fixed setting and plot the performance curves over the value of κ_1.

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

$$S_{h_1 \cup h_2} = S_{h_1} + S_{h_2} - S_{area}(h_1 \cap h_2) + S_{error}(h_1 \cap h_2). \qquad (27)$$

Both the overlapping area and the error can be computed from the segmentations obtained in Sect. 5. $S_{area}(h_1 \cap h_2)$ is just the area of overlap between the two segmentations. Let h_1 be the higher-scoring hypothesis of the two in terms of the optimization function. Under the assumption that h_1 opaquely occludes h_2, we can adjust for the error term $S_{error}(h_1 \cap h_2)$ by setting $p(\mathbf{p} = figure|h_2) = 0$ wherever $p(\mathbf{p} = figure|h_1) > p(\mathbf{p} = ground|h_1)$, that is for all pixels that belong to the segmentation of h_1.

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

$$S(\hat{m}) = \max_m m^T Q m = m^T \begin{bmatrix} q_{11} & \cdots & q_{1M} \\ \vdots & \ddots & \vdots \\ q_{M1} & \cdots & q_{MM} \end{bmatrix} m. \qquad (28)$$

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

$$\begin{aligned} q_{ii} = S_{h_i} &= -\kappa_1 + (1-\kappa_2)\frac{N}{A_s} \\ &\quad + \frac{\kappa_2}{A_s}\sum_{\mathbf{p}\in Seg(h_i)} p(\mathbf{p} = fig.|h_i) \qquad (29) \end{aligned}$$

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

$$q_{ij} = \frac{1}{2A_{s^*}}\left(-(1-\kappa_2)|O_{ij}| - \kappa_2 \sum_{\mathbf{p}\in O_{ij}} p(\mathbf{p} = figure|h^*)\right) \qquad (30)$$

where h^* denotes the weaker of the two hypotheses h_i and h_j and $O_{ij} = Seg(h_i) \cap Seg(h_j)$ is the area of overlap between their segmentations. As the number of possible combinations grows exponentially with increasing problem size, it may become intractable to search for the globally optimal solution. In practice, however, we found that only a relatively small number of hypotheses interact in most cases, so that it is usually sufficient to just compute a greedy approximation. Algorithm 6 summarizes the verification procedure.

7 Experimental Evaluation

7.1 Test Datasets and Experimental Protocol

In order to evaluate our method's performance and compare it to state-of-the-art approaches, we apply our system to several different test sets of increasing difficulty.

UIUC Cars(side) The *UIUC single-scale test set* consists of 170 images containing 200 side views of cars of approximately the same size. The *UIUC multi-scale test set* consists of 108 images containing 139 car side views at different scales. Both sets include instances of partially occluded cars, cars that have low contrast with the background, and images with highly textured backgrounds. For all experiments on these datasets, we train our detector on an own training set

Algorithm 6 The MDL verification algorithm.

Input: hypotheses $\mathcal{H} = \{h_i\}$ and corresponding segmentations $\left\{(img^{(i)}_{pfig}, img^{(i)}_{pgnd})\right\}$.
Output: indicator vector m of selected hypotheses.

// Build up the matrix $Q = \{q_{ij}\}$
for all hypotheses $h_i \in \mathcal{H}$ **do**
  $sum \leftarrow 0, N \leftarrow 0$
  Let A_i be the expected area of h_i at its detected scale.
  // Set the diagonal elements
  for all pixels $\mathbf{p} \in img$ **do**
    if $img^{(i)}_{pfig}(\mathbf{p}) > img^{(i)}_{pgnd}(\mathbf{p})$ **then**
      $sum \leftarrow sum + img^{(i)}_{pfig}(\mathbf{p})$
      $N \leftarrow N + 1$
    end if
  end for
  $q_{ii} \leftarrow -\kappa_1 + (1-\kappa_2)\frac{N}{A_i} + \kappa_2 \frac{1}{A_i} sum$
  // Set the interaction terms
  for all hypotheses $h_j \in \mathcal{H}, j \neq i$ **do**
    $sum \leftarrow 0, N \leftarrow 0$
    Let $k \in \{i, j\}$ be the index of the weaker hypothesis.
    for all pixels $\mathbf{p} \in img$ **do**
      if $img^{(i)}_{pfig}(\mathbf{p}) > img^{(i)}_{pgnd}(\mathbf{p}) \wedge$
        $img^{(j)}_{pfig}(\mathbf{p}) > img^{(j)}_{pgnd}(\mathbf{p})$ **then**
        $sum \leftarrow sum + img^{(k)}_{pfig}(\mathbf{p})$
        $N \leftarrow N + 1$
      end if
    end for
    $q_{ij} \leftarrow \frac{1}{2}(-(1-\kappa_2)\frac{N}{A_k} - \kappa_2 \frac{1}{A_k} sum)$
  end for
end for

// Greedy search for the best combination of hypotheses
$m \leftarrow (0, 0, \ldots, 0)$, finished $\leftarrow$ false
repeat
  for all unselected hypotheses h_i **do**
    $\tilde{m} \leftarrow m, \tilde{m}(i) \leftarrow 1$
    $S_i \leftarrow \tilde{m}^T Q \tilde{m} - m^T Q m$ *// Savings when h_i is selected*
  end for
  $k \leftarrow \arg\max_i (S_i)$
  if $S_k > 0$ **then**
    $m(k) \leftarrow 1$
  else
    finished $\leftarrow$ true
  end if
until finished

of only 50 hand-segmented images[3] (mirrored to represent both car directions) that were originally prepared for a different experiment. Thus, our detector remains more general and is not tuned to the specific test conditions. Since the original UIUC sets were captured at a far lower resolution than our training images, we additionally rescaled all test images by a constant factor prior to recognition (Note that this step does not increase the images' information content).

All experiments on these sets are performed using the evaluation scheme and detection tolerances from (Agarwal et al. 2004) based on bounding box overlap: a hypothesis with center coordinates (x, y, s) is compared with an annotation rectangle of size $(width, height)$ and center coordinates (x^*, y^*, s^*) and accepted if

$$\frac{|x - x^*|^2}{(0.25 width)^2} + \frac{|y - y^*|^2}{(0.25 height)^2} + \frac{|s/s^* - 1|^2}{(0.25)^2} \leq 1. \qquad (31)$$

In addition, only one hypothesis per object is accepted as correct detection; any additional hypothesis on the same object is counted as false positive.

CalTech Cars(rear) In addition to side views, we also test on rear views of cars using the 526 car and 1,370 non-car images of the CalTech `cars-brad` data set. This data set contains road scenes with significant scale variation, taken from the inside of a moving vehicle. The challenge here is to reliably detect other cars driving in front of the camera vehicle while restricting the number of false positives on background structures. For those experiments, our system is trained on the 126 (manually segmented) images of the CalTech `cars-markus` data set.

In order to evaluate detection accuracy with possibly changing bounding box aspect ratios, we adopt a slightly changed evaluation criterion for this and all following experiments (Leibe et al. 2005). We still check whether the detected bounding box center is close enough to the annotated center using the first two terms of (31), but we additionally demand that the mutual overlap between the hypothesis and annotation bounding boxes is at least 50%. Again, at most one hypothesis per object is counted as correct detection.

TUD Motorbikes Next, we evaluate our system on the TUD Motorbikes set, which is part of the PASCAL collection (Everingham 2006). This test set consists of 115 images containing 125 motorbike side views at different scales and with clutter and occlusion. For training, we use 153 motorbike side views from the CalTech database which are shown in front of uniform background allowing for easy segmentation (a subset of the 400 images (Fergus et al. 2003) used for training).

VOC'05 Motorbikes In order to show that our results also generalize to other scenarios, we apply our system to the VOC motorbike `test2` set, which has been used as a localization benchmark in the 2005 PASCAL Challenge (Everingham 2006). This data set consists of 202 images containing a total of 227 motorbikes at different scales and seen

[3] All training sets used in our experiments, as well as executables of the recognition system, are made available on the following webpage: http://www.vision.ee.ethz.ch/bleibe/ism/.

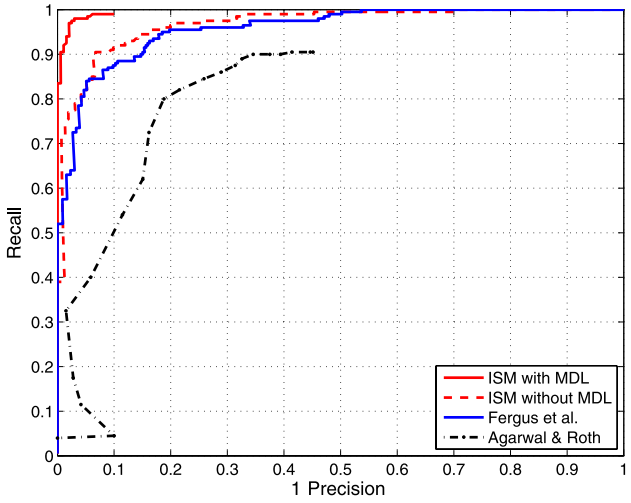
from different viewpoints. For this experiment, we use the same training set of 153 motorbike side views as above, but since only 39% of the test cases are shown in side views, the maximally achievable recall for our system is limited.

Leeds Cows The above datasets contain only relatively rigid objects. In order to also quantify our method's robustness to changing articulations, we next evaluate it on a database of video sequences of walking cows originally used for detecting lameness in livestock (Magee and Boyle 2002). Each sequence shows one or more cows walking from right to left in front of different, static backgrounds. For training, we took out all sequences corresponding to three backgrounds and extracted 113 randomly chosen frames, for which we manually created a reference segmentation. We then tested on 14 different video sequences showing a total of 18 unseen cows in front of novel backgrounds and with varying lighting conditions. Some test sequences contain severe interlacing and MPEG-compression artifacts and significant noise. Altogether, the test suite consists of a total of 2217 frames, in which 1682 instances of cows are visible by at least 50%. This provides us with a significant number of test cases to quantify both our method's ability to deal with different articulations and its robustness to (boundary) occlusion.

TUD Pedestrians Last but not least, we evaluate our method on the TUD pedestrian set. This highly challenging test set consists of 206 images containing crowded street scenes in an Asian metropolis with a total of 595 annotated pedestrians, most of them in side views (Leibe et al. 2005). The reason why we only speak of "annotated" pedestrians here is that in the depicted crowded scenes, it is often not obvious where to draw the line and decide whether a pedestrian should be counted or not. People occur in every state of occlusion, from fully visible to just half a leg protruding behind some other person. We therefore decided to annotate only those cases where a human could clearly detect the pedestrian without having to resort to reasoning. As a consequence, all pedestrians were annotated where at least some part of the torso was visible. For this experiment, our detector was trained on use 210 training images of pedestrian side views, recorded in Switzerland with a static camera, for which a motion segmentation was computed with a Grimson-Stauffer background model (Stauffer and Grimson 1999).

7.2 Object Detection Performance

In order to demonstrate the different stages of our system, we first apply it to the *UIUC single-scale cars* dataset. Since this dataset contains only very limited scale variation, we use Harris interest points and simple 25×25 patch features compared by *normalized correlation*. Figure 10 shows



Method	Agarwal (2004)	Garg (2002)	Fergus (2003)	ISM, no MDL	ISM + MDL	Mutch (2006)
EER	~79%	~88%	88.5%	91.0%	97.5%	99.9%

Fig. 10 Comparison of our results on the UIUC single-scale car database with others reported in the literature

a recall-precision curve (RPC) of our method's performance before and after the MDL verification stage. As can be seen from the figure, the initial voting stage succeeds to generalize from the small 50-image training set and achieves already good detection results with an Equal Error Rate (EER) performance of 91% (corresponding to 182 out of 200 correct detections with 18 false positives). When the MDL criterion is applied as a verification stage, the results are significantly improved, and the EER performance increases from 91% to 97.5%. Without the verification stage, our algorithm could reach this recall rate only at the price of a reduced precision of only 74.1%. This means that for the same recall rate, the verification stage manages to reject 64 additional false positives while keeping all correct detections. In addition, the results become far more stable over a wider parameter range than before.

The same figure and the adjacent table also show a comparison of our method's performance with other results reported in the literature. With an EER performance of 97.5%, our method presents a significant improvement over previous results. In very recent work, Mutch and Lowe (2006) reported even better performance with 99.94% EER using densely sampled features and a biologically motivated multi-level representation. This indicates that there may still be some potential for improvement in the feature extraction stage. In the following sections, we will therefore examine different choices for the feature detector and descriptor.

Some example detections in difficult settings and the corresponding top-down segmentations can be seen in Fig. 11. Those results show that our method still works in the presence of occlusion, low contrast, and cluttered backgrounds.

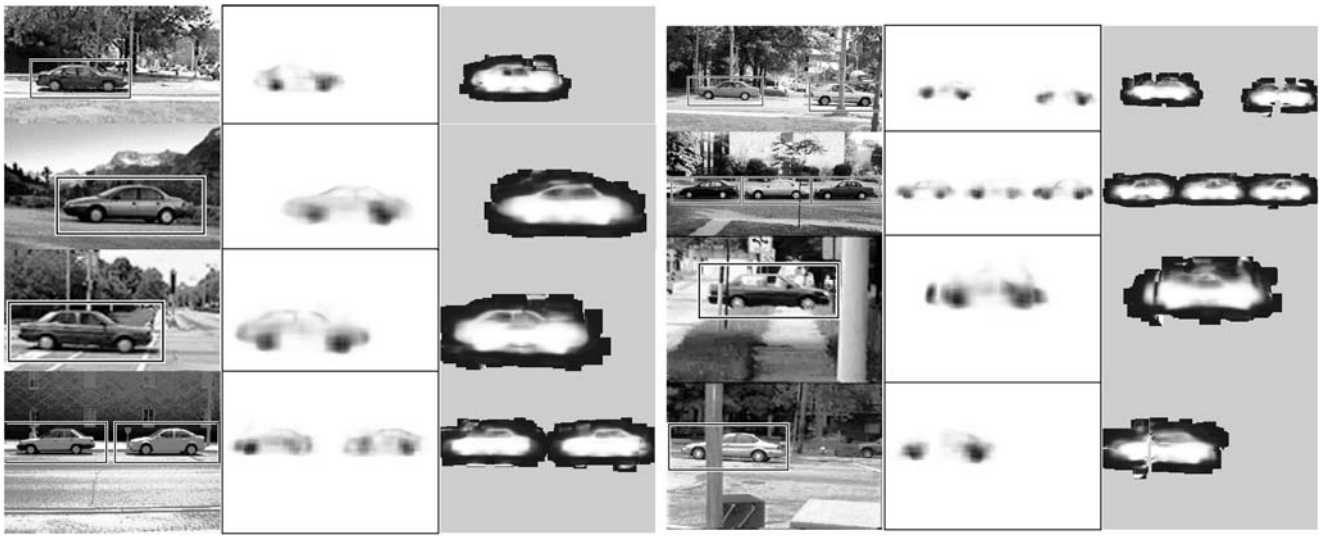


Fig. 11 Example object detections, *figure* probabilities, and segmentations automatically generated by our method

Fig. 12 All error cases (missing detections and false positives) our algorithm returned on the UIUC single-scale car test set

At the EER point, our method correctly finds 195 of the 200 test cases with only 5 false positives. All of those error cases are displayed in Fig. 12. The main reasons for missing detections are combinations of several factors, such as low contrast, occlusion, and image plane rotation, that push the object hypothesis below the acceptance threshold. The false positives are due to richly textured backgrounds on which a large number of spurious object parts are found.

In addition to the recognition results, our method automatically generates object segmentations from the test images. Even though the quality of the original images is rather low, the segmentations are reliable and can serve as a basis for later processing stages, e.g. to further improve the recognition results using global methods. In particular, the examples show that the system can not only detect cars despite partial occlusion, but it is often even able to segment out the occluding structure.[4]

7.3 Experimental Comparison of Clustering Algorithms

Next, we evaluate the different clustering methods by applying them to the same data set and comparing the suitability of the resulting codebooks for recognition. The evaluation is based on two criteria. One is the recognition performance the codebook allows. The other is its representational quality, as measured by the number of occurrences that need to be stored, which determines the effective cost of the recognition process.

Starting from the 50-image training set, a total of 6,413 patches are extracted with the Harris interest point operator. Average-link clustering with *normalized correlation* as similarity measure and a cut-off threshold of $t = 0.7$ produces 2,104 visually compact clusters. However, 1,241 of these clusters contain only one patch, which means that they do not correspond to any repeating structure. We therefore discard those clusters and keep only the remaining 863 prototypes. In comparison, k-means clustering is executed with different values for k ranging from 100 to 2,000. In addition to the original codebooks, we also try the codebook reduction step and measure the performance when single-patch clusters are removed.

Figure 13 shows the results of this experiment. In the left diagram, the recognition performance is plotted as a function of the codebook size. The codebook obtained by k-means reaches approximately the same performance as the

[4] In the presented examples, our method is also able to segment out the car windows, since those were labeled *ground* in the training data.

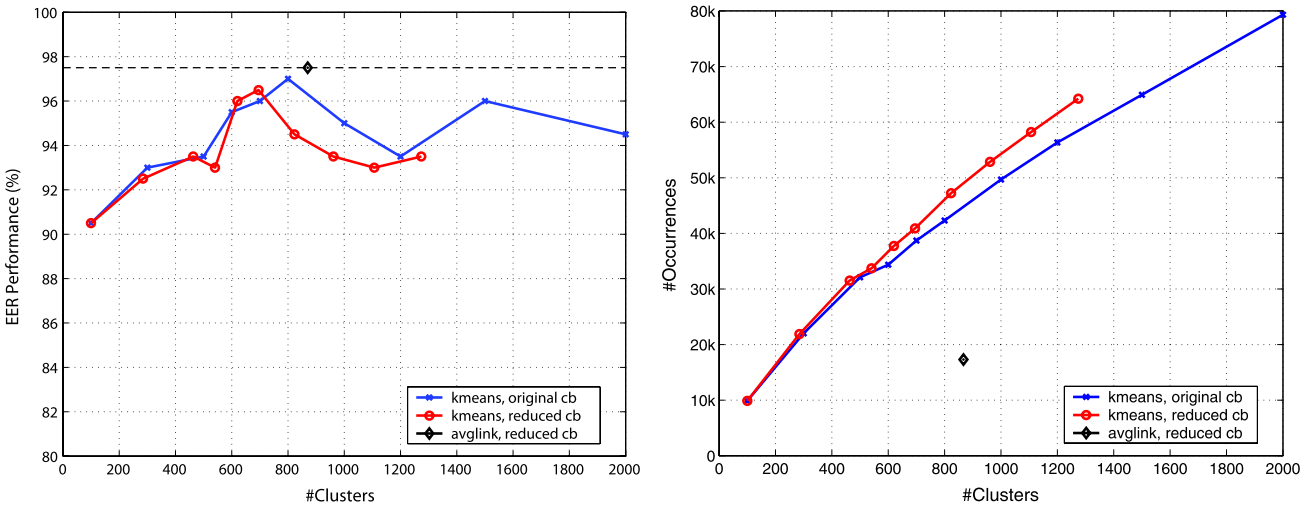


Fig. 13 Comparison of codebooks built with k-means and average-link clustering. (*left*): Recognition performance on the UIUC car database. (*right*): Number of stored activations needed to represent the matching uncertainty. As can be seen from the plots, the k-means codebook achieves nearly the same performance as the one built by average-link clustering, but it requires more than twice as many activations to be stored

one obtained by average-link clustering when k is set to a similar number of clusters. This is the case for both the original and the reduced k-means codebook (without single-patch clusters). However, as can be seen from the right diagram, the number of occurrences for this codebook size is more than twice as high for k-means as for average-link clustering. All in all, the k-means codebook with $k = 800$ clusters generates 42,310 occurrences from the initial 6,413 training patches, while the more specific average-link codebook can represent the full appearance distribution with only 17,281 occurrences.

We can thus draw the following conclusions. First, the experiment confirms that the proposed uncertainty modeling stage can indeed compensate for a less specific codebook. As can be seen from Fig. 13(left), the recognition performance degrades gracefully for both smaller and larger values of k. This result has important consequences for the scalability of our approach, since it indicates that the method can be applied even to cases where no optimal codebook is available. Second, the experiment indicates that the visually more compact clusters produced by average-link clustering may be better suited to our problem than the partition obtained by k-means and may lead to tighter spatial occurrence distributions with fewer entries that need to be stored. Ideally, this result would have to be verified by more extensive experiments over several test runs and multiple datasets. However, together with the additional advantage that the compactness parameter of agglomerative clustering is only dependent on the selected feature descriptor, whereas the k of k-means has to be adjusted anew for every new training set, the experiment already provides a strong argument for agglomerative clustering.

7.4 Effect of the Training Set Size

Next, we explore the effect of the training set size on detection performance. Up to now, all detectors in this section have been trained on the original 50 car images. We now compare their performance when only a subset of those images is considered. In addition to the single-scale *Harris* detector, we also apply a scale-invariant *DoG* detector (Lowe 2004). Figure 14 shows the resulting performance for different training set sizes from 5 to 50 images. As can be seen from the plot, both the *Harris* and the *DoG* codebook reach 90% EER performance already with 20 training examples. When more training images are added, the *Harris* codebook further improves to the known rate of 97.5%. In contrast, the performance of the *DoG* detector reaches a saturation point and increases only to 91% for the full training set. Here the advantage of seeing more training images is offset by the increased variance in patch appearance caused by the additional scale dimension.

Apart from this evaluation, the figure also compares the performance for the original codebooks with the reduced codebooks that are obtained when all single-patch clusters are discarded. It can be observed that the two versions show some differences for the initial voting stage, which however level out when the MDL verification stage is applied. Considering that the original codebooks typically contain more than twice as many clusters as the reduced versions, the reduction step can thus be safely advised in order to increase run-time performance.

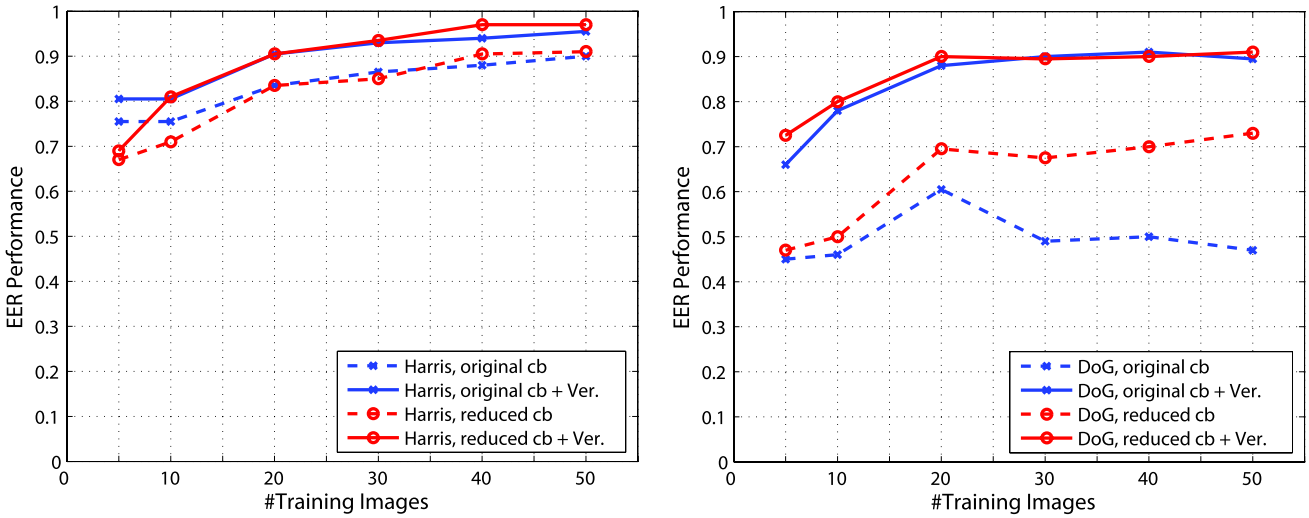


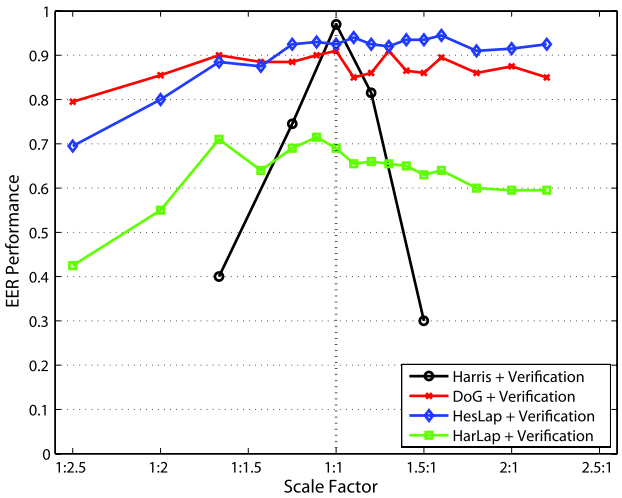
Fig. 14 EER performance on the UIUC database for varying training set sizes: (*left*) for the Harris detector; (*right*) for the DoG detector. The plots show the performance for the original codebooks and for the reduced codebooks when all single-patch clusters are discarded. As can be seen from the plots, both detectors achieve good performance already with 20 training examples. Moreover, the experiment shows that the codebook reduction step does not lead to a decrease in performance

7.5 Comparison of Different Interest Point Detectors

In the next experiment, we evaluate the effect of the interest point detector on recognition performance. In previous studies, interest point detectors have mainly been evaluated in terms of their repeatability. Consequently, significant effort has been spent on making the detectors discriminant enough that they find exactly the same structures again under different viewing conditions. However, in our case, the task is to recognize and localize previously unseen objects of a given category. This means that we cannot assume to find exactly the same structures again; instead the system needs to generalize and find structures that are similar enough to known object parts while still allowing enough flexibility to cope with variations. Also, because of the large intra-class variability, more potential matching candidates are needed to compensate for inevitable mismatches. Last but not least, the interest points should provide a sufficient cover of the object, so that it can be recognized even if some important parts are occluded. Altogether, this imposes a rather different set of constraints on the interest point detectors, so that their usefulness for our application can only be determined by an experimental comparison.

In the following experiment, we evaluate three different types of scale-invariant interest point operators: the *Harris-Laplace* and *Hessian-Laplace* detectors (Mikolajczyk et al. 2005b) and the *DoG* (Difference of Gaussian) detector (Lowe 2004). All three operators have been shown to yield high repeatability (Mikolajczyk et al. 2005b), but they differ in the type of structures they respond to. The *Harris-Laplace* and *Hessian-Laplace* detectors look for scale-adapted maxima of the Harris function and Hessian determinant, respectively, where the locations along the scale dimension are found by the Laplacian-of-Gaussian (Mikolajczyk et al. 2005b). The *DoG* detector (Lowe 2004) finds regions at 3D scale-space extrema of the Difference-of-Gaussian.

In a first step, we analyze the different detectors' robustness to scale changes. In particular, we are interested in the limit to the detectors' performance when the scale of the test images is altered by a large (but known) factor and the fraction of familiar image structures is thus decreased. In the following experiment, the UIUC single-scale car database images are rescaled to different sizes and the performance is measured as a function of the scaling factor relative to the size of the training examples. Figure 15(left) shows the EER performances that can be achieved for scale changes between factor 0.4 (corresponding to a scale reduction of 1:2.5) and factor 2.2. When the training and test images are approximately of the same size, the single-scale *Harris* codebook is highly discriminant and provides the good performance described in the previous sections. However, the evaluation shows that it is only robust to scale changes up to about 20%, after which its performance quickly drops. As a result of its scale selection step, the *Harris-Laplace* codebook performs more stably over a larger range of scales. However, with 69% at the EER, its absolute performance is far below that of the single-scale version. The main reason for this poor performance is that the *Harris-Laplace* detector returns a smaller absolute number of interest points on the object, so that a sufficient cover is not always guaranteed. Although previous studies have shown that Harris-Laplace points are more discriminant individually (Dorko and Schmid 2003), their smaller number is a strong disadvantage. The *Hessian-Laplace* and *DoG* detectors, on the



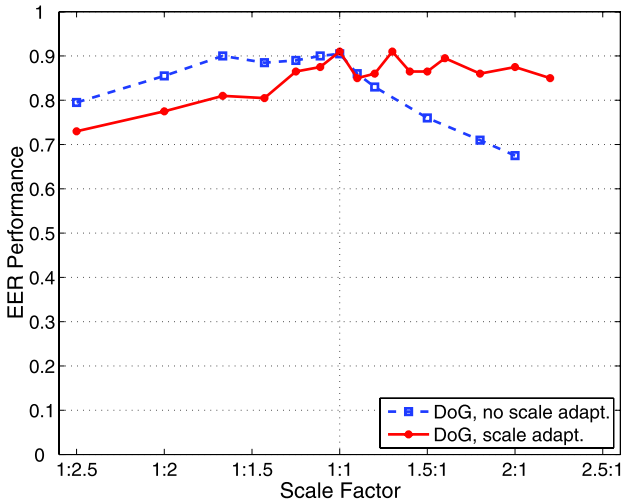


Fig. 15 (*left*): EER performance over scale changes relative to the size of the training examples. While optimal for the single-scale case, the Harris codebook is only robust to small scale changes. The DoG and Hessian-Laplace codebooks, on the other hand, maintain high performance over a large range of scales. (*right*): A comparison of the performances with and without the scale-adaption mechanism. As can be seen from the plot, the adapted search window size is necessary for scales greater than 1.0, but reduces performance for smaller scales, since a certain minimum search window size needs to be maintained

other hand, both find enough points on the objects and are discriminant enough to allow reliable matches to the codebook. They start with 92.5% and 91%, respectively, for test images at the same scale and can compensate for both enlargements and size reductions of more than a factor of two. If only one type of points shall be used, they are thus better suited for use in our framework. Figure 15(right) also shows that the system's performance quickly degrades without the scale adaptation step from Sect. 4.3.2, confirming that this step is indeed important.

7.6 Comparison of Different Local Descriptors

In the previous experiments, we have only considered simple image patches as basic features of our recognition system. While this has been a straightforward choice, it is not necessarily optimal. However, our approach is not restricted to patches, but can in principle be operated with any type of local feature. Recently, a vast array of different local descriptors have become available, and several studies have investigated their suitability for object categorization (Mikolajczyk et al. 2005a; Seemann et al. 2005). In the following experiment, we evaluate two of those region descriptors. *SIFT* descriptors (Lowe 2004) are 3D histograms of gradient locations and orientations with 4×4 location and 8 orientation bins, resulting in 128-dimensional feature vectors. Local *Shape Context* (Belongie et al. 2002) descriptors are histograms of gradient orientations sampled at edge points in a log-polar grid. Here we use them in the implementation of (Mikolajczyk and Schmid 2005) with 9 location and 4 orientation bins and thus 36 dimensions. For comparison, we include our previous choice of 25×25 pixel *Patches* (Agarwal et al. 2004; Leibe et al. 2004), which lead to a descriptor of length 625. The evaluation is performed with an own implementation of the *DoG* detector and *Patch* descriptor. For all other detectors and descriptors, we use the implementations publicly available at the Oxford Interest Point Webpage (http://www.robots.ox.ac.uk/~vgg/research/affine). Patches are compared using *Normalized Correlation*; all other descriptors are compared using Euclidean distances.

One open parameter has to be adjusted for each cue, namely the question how much the clustering step should compress the training features during codebook generation. When using agglomerative clustering, this translates to the question how to set the cut-off and matching threshold t for optimal performance. Clearly, this parameter depends on the choice of descriptor. In order to find good values for this parameter and analyze its influence on recognition performance, we applied all 9 detector/descriptor combinations to the TUD motorbikes set and compared their EER detection performance for 5–8 different threshold settings. Figure 16 shows the results of this experiment, separated per descriptor. We can make two observations. First, when comparing descriptors across different detectors, a clear performance optimum can be found at certain similarity values for all three descriptors. Those threshold settings can thus serve as default values whenever the descriptors are used in future experiments. Second, the results allow to rank the detector/descriptor combinations based on their performance. For the descriptors, *SIFT* and *Shape Context* perform consistently best over all three detectors. For the detectors, *Hessian-Laplace* and *DoG* perform best in all but one case.

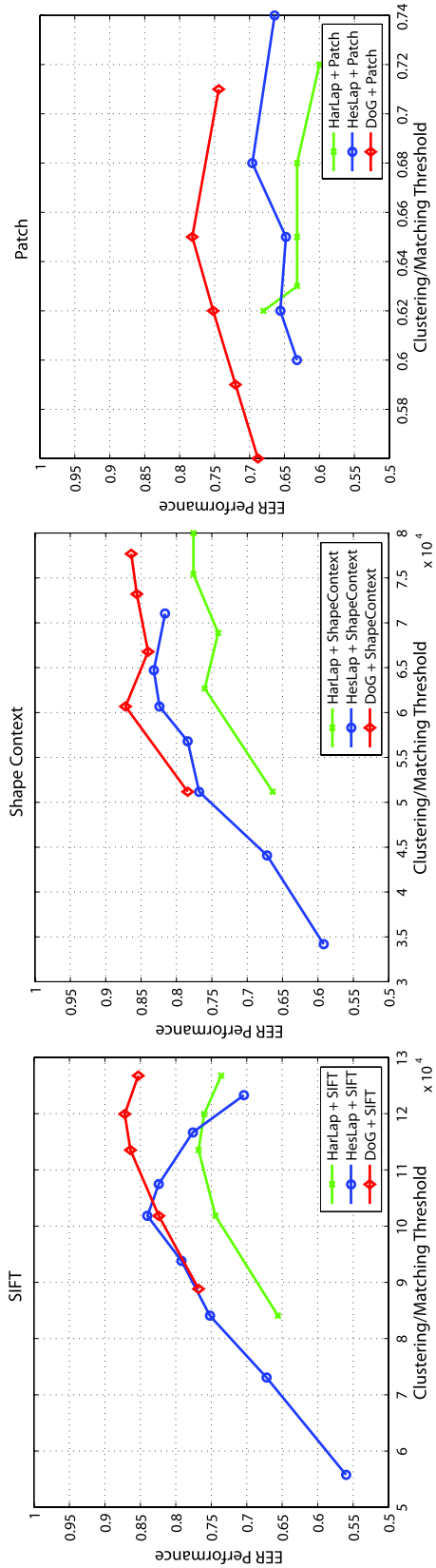


Fig. 16 EER performances for all 9 detector/descriptor combinations on the TUD motorbikes. The plots show the performance gradation when the clustering/matching threshold is varied

In terms of combinations, *DoG+SIFT* and *DoG+Shape Context* obtain the best performance with 87% EER.

7.7 Results for Other Test Sets

In the following, we evaluate the system on more difficult scenes, containing multiple objects at different scales. In addition to cars, we apply our algorithm to three additional categories: motorbikes, cows, and pedestrians.

7.7.1 UIUC Multi-Scale Cars

First, we present results on the UIUC multi-scale database (Agarwal et al. 2004). Figure 17(a) shows the results of this experiment. The black line corresponds to the performance reported by (Agarwal et al. 2004), with an EER of about 45%. In contrast, our approach based on *DoG* interest points and *Patch* features achieves an EER performance of 85.6%, corresponding to 119 out of 132 correct detections with 20 false positives. Using *Hessian-Laplace* interest points and local *Shape Context* features, this result is again significantly improved to an EER performance of 95%. This number also compares favorably to the performance reported by Mutch and Lowe (2006), who obtained 90.6% EER with their method.

7.7.2 Motorbikes

Next, we show our system's results on motorbikes. Figure 17(b) summarizes the results from Sect. 7.6 on the TUD motorbike set. The best performance is achieved by the combinations of *DoG* and *SIFT*/*Shape Context*, both with an EER score of 87%. Figure 18 shows example detections on difficult images from this test set that demonstrate the appearance variability spanned by the motorbike category and the segmentation quality that can be achieved by our approach. As these results show, our method manages to adapt to different appearances and deliver accurate segmentations, even in scenes with difficult backgrounds and partial occlusion. Due to the larger appearance variability of motorbikes, however, it is in general not possible anymore to segment out the occluding structure (as was the case for the car category in the previous experiments).

In order to ensure that the results generalize also to different settings, we apply our approach to the more challenging VOC motorbikes set using the same parameter settings as for the previous experiment. Figure 17(c) shows the results of this experiment. Since only about 39% of the test cases are visible in the side views our detector was trained on, the EER performance is not as meaningful; instead, we compare recall in the high-precision range. From this, it can be seen that the best recognition performance is achieved by the combinations of *DoG+SIFT* and *DoG+Shape Context*,

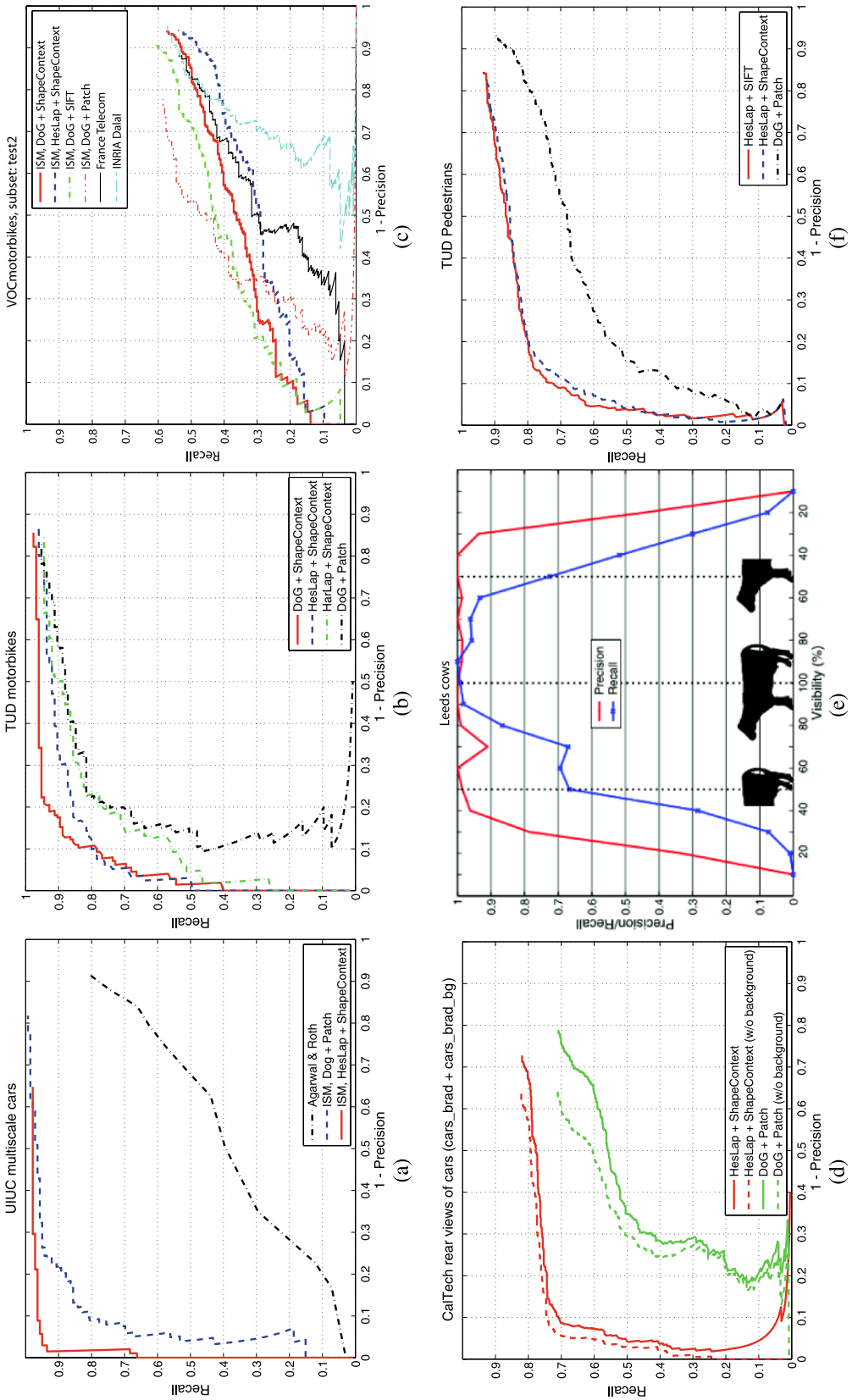


Fig. 17 Performance comparison on several data sets: (**a**) the UIUC multi-scale cars; (**b**) the TUD motorbikes; (**c**) the more difficult VOC motorbike `test2` set (note that this plot is rotated 90° compared to the version shown in (Everingham 2006)); (**d**) the CalTech cars-rear set; (**e**) precision/recall curves for the cow sequences when $x\%$ of the cow's length is visible; and (**f**) results on the TUD Pedestrian set showing crowded scenes. Please note in plot (**c**) that while our detector is exclusively trained on side views, only 39% of the motorbikes in the VOC set are shown in side views, which limits the maximally achievable recall

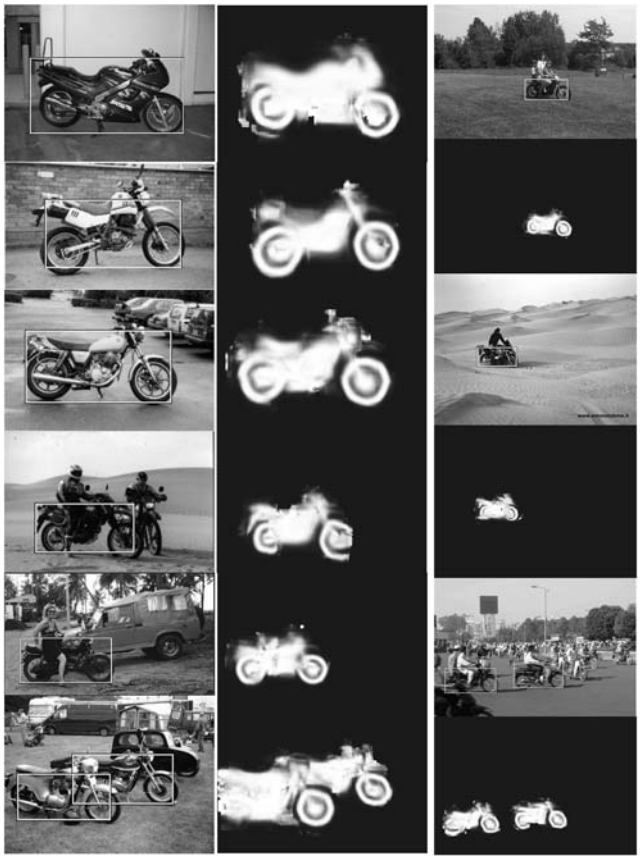


Fig. 18 Examples for the variety of motorbike shapes and appearances that are still reliably detected and segmented

which both find half the available side views at a precision of 90%. Considering the difficulty of the test set, this is still a very good result. The best recall is achieved by the feature combination *DoG+Patch* with an EER performance of 48%. For comparison, we also show the performance curves for two other approaches from the 2005 PASCAL Challenge (Everingham 2006): the one from Garcia and Delakis (2004) and the one from Dalal and Triggs (2005). (For fairness it must be said, however, that only our *DoG+Patch* version was entered into the original competition and trained on the slightly smaller training set provided there; the other feature combinations were produced afterwards).

7.7.3 Rear Views of Cars

Next, we apply our system to the detection of rear views of cars, using the `cars-brad` set of CalTech database. Figure 17(d) displays the detection results for two feature types: *DoG+Patch* and *Hessian-Laplace+Shape Context*. The dashed curves show the detection performance on the 526 positive images; the solid lines show the performance taking also the 1370 background images into account. Since many of the annotated cars are strongly occluded, only about 82% recall can be reached. However, the results show that

Table 1 Performance comparison on a *present/absent* classification task for two of the CalTech categories with other methods from the literature, according to the evaluation scheme in (Fergus et al. 2003)

Data Set	Motorbikes	Cars Rear
Weber (2000)	88.0%	–
Opelt (2004)	92.2%	–
Thureson (2004)	93.2%	–
Fergus (2003)	93.3%	90.3%
J. Zhang (2007)	98.5%	98.3%
Deselaers (2005)	–	98.9%
W. Zhang (2005)	99.0%	–
ISM (*Patch*)	94.0%	93.9%
ISM (*SC*)	97.4%	96.7%

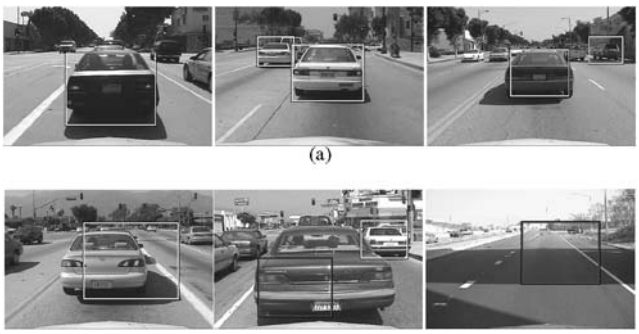


Fig. 19 Examples for (**a**) correct detections of rear views of cars on the CalTech data set and for some typical problem cases: (**b**) alignment of the detection bounding box on the car's shadow; (**c**) incorrect scale estimates; (**d**) spurious detections caused by similar image structures

our approach yields very accurate detections. Looking at the 90% precision level, our approach achieves 74.4% recall on the positive set, which only reduces to 70.9% recall when the 1370 negative images are added (corresponding to 0.103 and 0.027 false positives per image, respectively).

Since no other localization results on this data set have been reported in the literature so far, we also evaluate our method on an *object present/absent* classification task, according to the evaluation scheme in (Fergus et al. 2003). In order to decide whether or not a test image contains a rear view of a car, we apply our scale-invariant detector with a scale search range of [0.3,1.5] and classify an image with the label *present* if at least one detection can be found. Table 1 shows the results of this experiment. With *DoG+Patch* features, our approach achieves an EER classification performance of 93.9%. Using *Hessian-Laplace+Shape Context*, this result improves to 96.7%. Both results compare favorably with (Fergus et al. 2003). Similar results can be achieved for the CalTech motorbikes, as also shown in the same table. As a comparison with other more recent approaches shows, however, discriminative methods using

densely sampled features and SVM classifiers seem to be generally more suited to this classification task.

Figure 19(a) presents some examples of correct detections on the test set. As can be observed, the approach is able to find a large variety of car appearances at different scales in the images. Some typical problem cases are shown in the bottom part of the figure. As the car's shadow proves to be an important feature for detection, a displaced shadow sometimes leads to a misaligned bounding box (Fig. 19(b)). Another cause for incorrect detections are self-similarities in the car structure at different scales that sometimes lead to a wrong scale estimate (Fig. 19(c)). Finally, some spurious detections are found on regions with similar image structure (Fig. 19(d)).

7.7.4 Cows

Up to now, we have only considered static objects in our experiments. Even though environmental conditions can vary greatly, cars and motorbikes are still rather restricted in their possible shapes. This changes when we consider articulated objects, such as walking animals. In order to fully demonstrate our method's capabilities, we therefore apply it to the *Leeds cows* set. The 2217 frames from this test suite provide us with a significant number of test cases to quantify both our method's ability to deal with different articulations and its robustness to (boundary) occlusion. Using video sequences for testing also allows to avoid any bias caused by selecting only certain frames. However, since we are still interested in a single-frame recognition scenario, we apply our algorithm to each frame separately. That is, no temporal continuity information is used for recognition, which one would obviously add for a tracking scenario.

We apply our method to this test set using exactly the same detector settings as before to obtain equal error rate for the single-scale car experiments. Using *Harris* interest points and *Patch* descriptors, our detector correctly finds 1535 out of the 1682 cows, corresponding to a recall of 91.2%. With only 30 false positives over all 2217 frames, the overall precision is at 98.0%. Figure 17(e) shows the precision and recall values as a function of the visible object area. As can be seen from this plot, the method has no difficulties in recognizing cows that are fully visible (99.1% recall at 99.5% precision). Moreover, it can cope with significant partial occlusion. When only 60% of the object is visible, recall only drops to 79.8%. Even when half the object is occluded, the recognition rate is still at 69.0%. In some rare cases, even a very small object portion of about 20–30% is already enough for recognition (such as in the leftmost image in Fig. 21). Precision constantly stays at a high level. False positives mainly occur when only one pair of legs is fully visible and the system generates a competing hypothesis interpreting the front legs as rear legs, or vice versa. Usually, such secondary hypotheses are filtered out by the MDL stage, but if the correct hypothesis does not have enough support in the image due to partial visibility, the secondary hypothesis sometimes wins.

Fig. 20 Example detections and automatically generated segmentations from one cow sequence. (*middle row*) segmentations obtained from the initial hypotheses; (*bottom row*) segmentations from refined hypotheses (with additional features sampled in a uniform grid)

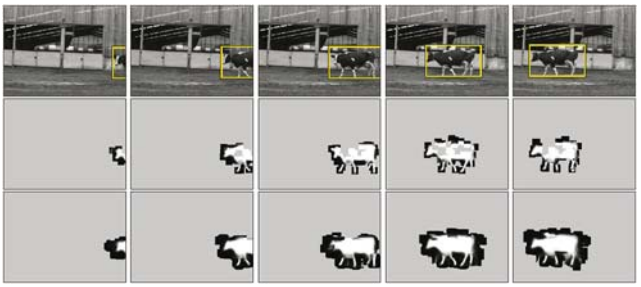


Fig. 21 Example detections and automatically generated segmentations from another cow sequence. Note in particular the leftmost image, where the cow is correctly recognized and segmented despite a large degree of occlusion

Figures 20 and 21 show example detection and segmentation results for two of the sequences used in this evaluation. As can be seen from these images, the system not only manages to recognize unseen-before cows with novel texture patterns, but it also provides good segmentations for them. Again, we want to emphasize that no tracking information is used to generate these results. On the contrary, the capability to generate object segmentations from single frames could make our method a valuable supplement to many current tracking algorithms, allowing to (re)initialize them through shape cues that are orthogonal to those gained from motion estimates.

7.7.5 Pedestrians

Finally, we apply our approach to pedestrian detection in crowded street scenes using the challenging *TUD pedestrian* set. Figure 17(f) shows the results of this experiment. Using *DoG+Patches*, our approach achieves an EER performance of 64% (corresponding to 363 correct detections with 204 false positives) (Leibe et al. 2005). Applying *Hessian-Laplace* and *SIFT/Shape Context* features, this performance

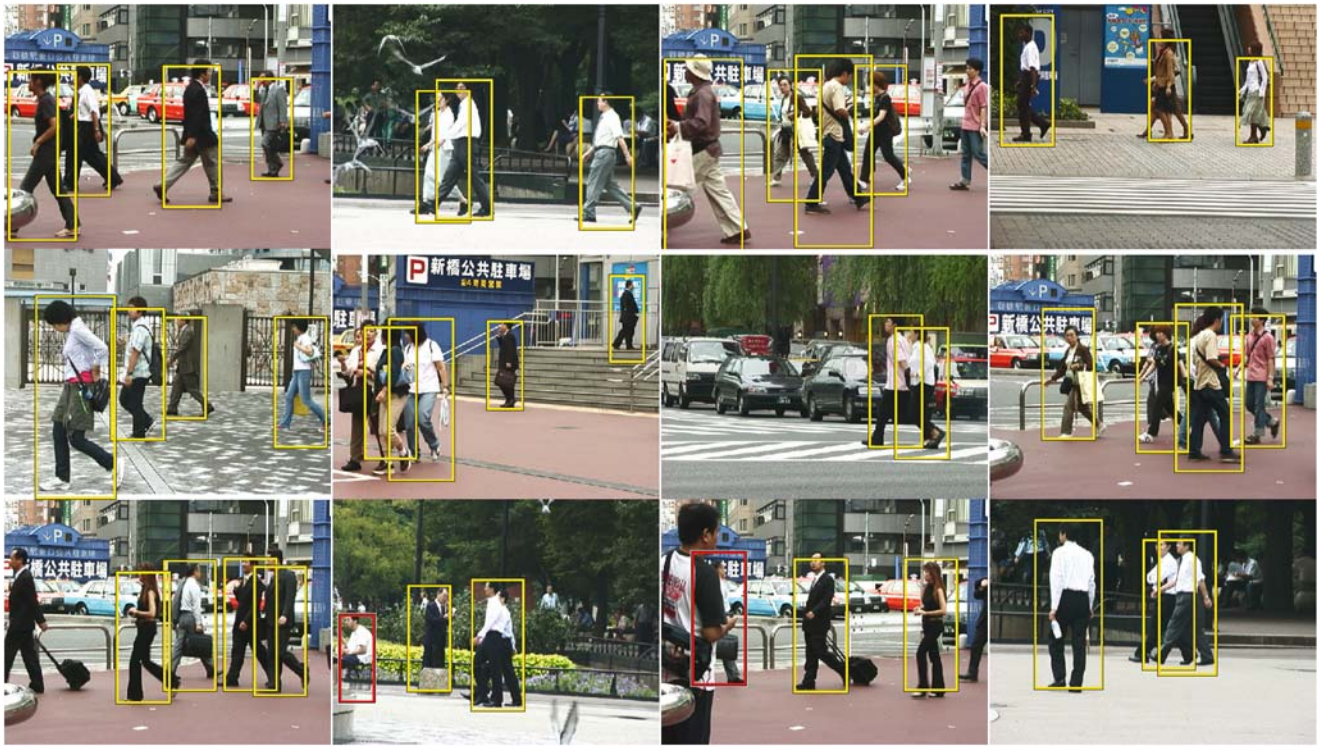


Fig. 22 (Color online) Example detections of our approach on difficult crowded scenes from the TUD Pedestrian test set (at the EER). Correct detections are shown in *yellow*, false positives in *red*

is again improved to 80% EER (476 correct detections with 119 false positives).

As already pointed out before, these quantitative results should be regarded with special consideration. Many pedestrians in the test set are severely occluded, and it is often difficult to decide whether a pedestrian should be annotated or not. As a consequence, our detector still occasionally responded to pedestrians that were not annotated. On the other hand, a significant number of the annotated pedestrians are so severely occluded that it would be unrealistic to expect any current algorithm to reach 100% recognition rate with only a small number of false positives. In order to give a better impression of our method's performance, Fig. 22 therefore shows obtained detection results on example images from the test set (at the EER). As can be seen from those examples, the proposed method can reliably detect and localize pedestrians in crowded scenes and with severe overlaps.

7.7.6 Results on Other Datasets

To conclude, we present some example results on images from the *LabelMe* database (Russell et al. 2005) to demonstrate that our system can also be applied when dealing with very large images, where a large number of potential false positives need to be rejected. Those results are however only intended to give a visual impression of our method's performance, not as a systematic evaluation (which the *LabelMe* dataset also wouldn't permit due to its dynamically changing nature). Figure 23 shows example detections on several such images, processed at their original resolution of 2048×1536 pixels, and combining both a car and a pedestrian detector. As can be seen from those results, the system yields accurate detections even under those conditions while keeping only a small number of false positives.

8 Discussion and Conclusion

In this paper, we have proposed a method for learning the appearance and spatial structure of a visual object category in order to recognize novel objects of that category, localize them in cluttered real-world scenes, and automatically segment them from the background. We have provided efficient algorithms for each of those step and evaluated the resulting recognition performance on several large data sets. Our results show that the method scales well to different object categories and achieves good object detection and segmentation performance in difficult real-world scenes.

A main contribution of our work is the integration of object category detection and figure-ground segmentation into a common probabilistic framework. As shown in our experiments, the tight coupling between those two processes allows both to benefit from each other and improve their individual performances. Thus, the initial recognition phase not

Fig. 23 Example detections on difficult test images from the MIT LabelMe data set (Russell et al. 2005). All images were processed at their original resolution of 2048 × 1536 pixels. The results confirm our approach's ability to yield accurate detections in such complex scenes with only very few false positives, as the enlargements in the *bottom rows* show

only initializes the top-down segmentation process with a possible object location, but it also provides an uncertainty estimate of local measurements and of their influence on the object hypothesis. In return, the resulting probabilistic segmentation permits the recognition stage to focus its effort on object pixels and discard misleading influences from the background. Altogether, the two processes collaborate in an iterative evidence aggregation scheme which tries to

make maximal use of the information extracted from the image.

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

The core part of our approach is the Implicit Shape Model defined in Sect. 4.3. This implicit representation is flexible enough that it can combine the information of local object parts observed on different training examples and interpolate between the corresponding objects. As a result, our approach can learn object models from few examples and achieve competitive object detection performance already with training sets that are between one and two orders of magnitude smaller than those used in comparable approaches. Taking a broader view, this implicit model can be seen as a further generalization of the Hough Transform to work with uncertain data. In our approach, we have used this capability to represent the uncertainty from intra-class variation, but it would also be possible to use it with different sources of uncertainty, e.g. for the identification of known objects under lighting variations.

The run time of the resulting approach mainly depends on three factors: model complexity (the number of codebook entries and occurrences), image size, and the selected search scale range. Using our current implementation, the single-scale car detector based on Harris points takes between 2–3 s for a typical 320×240 test image. Typical runtimes of the pedestrian detector (without our more recent GPU-based top-down segmentation) range between 4–7 s for the same image size, including feature extraction, detection, top-down segmentation, and MDL verification. We expect that both run-times can still be considerably improved by a more efficient implementation.

The system can still be extended in several ways. For very large scale changes such as the ones encountered in the last experiment, it can be advantageous to work on several rescaled versions of the image, simply because of computational efficiency. Other possible extensions include the integration of multiple cues and the combination of several detectors for multi-category discrimination. Finally, many real-world applications require that objects be recognized from multiple viewpoints. While this can in principle be achieved by simply stacking several single-view detectors, such an approach would not take advantage of the possibility to share features (Torralba et al. 2004). Extending the ISM approach towards this goal will be a topic of future work.

Acknowledgements This work has been funded, in part, by the EU projects CogVis (IST-2000-29375) and CoSy (IST-2002-004250) and the Swiss Federal Office for Education and Science (BBW 00.0617). We thank Shivani Agarwal for the UIUC car data, Rob Fergus for the CalTech categories, Derek Magee for the cow sequences, Toyota Motor Corporation Europe for the pedestrian data, and Krystian Mikolajczyk for his interest point detector and descriptor implementations.

References

Agarwal, S., Atwan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1475–1490.

Bajcsy, R., Solina, F., & Gupta, A. (1990). Segmentation versus object representation—are they separable? In *Analysis and interpretation of range images* (pp. 207–223). New York: Springer.

Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, *13*(2), 111–122.

Belongie, S., Malik, J., & Puchiza, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(4), 509–522.

Benzécri, J. P. (1982). Construction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques. *Cahiers de l'Analyse des Données*, *7*(2), 209–218.

Borenstein, E., & Ullman, S. (2002). Class-specific, top-down segmentation. In *Lecture notes in computer science: Vol. 2353. ECCV'02* (pp. 109–122). Berlin: Springer.

Borenstein, E., Sharon, E., & Ullman, S. (2004). Combining top-down and bottom-up segmentations. In *Workshop on perceptual organization in computer vision*, Washington, DC, June 2004.

Bruynooghe, M. (1977). Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. *Statistique et Analyse des Données*, *3*, 24–42.

Burl, M. C., Weber, M., & Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV'98*.

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

Collins, R. (2003). Mean-shift blob tracking through scale space. In *CVPR'03*.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Comaniciu, D., Ramesh, V., & Meer, P. (2001). The variable bandwidth mean shift and data-driven scale selection. In *ICCV'01*.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. In *ECCV'98*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR'05*.

Day, W. H. E., & Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, *1*, 7–24.

de Rham, C. (1980). La classification hiérarchique ascendante selon la méthode des voisins réciproques. *Cahiers de l'Analyse des Données*, *5*(2), 135–144.

Deselaers, T., Keysers, D., & Ney, H. (2005). Improving a discriminative approach to object recognition using image patches. In *DAGM'05*.

Dorko, G., & Schmid, C. (2003). Selection of scale invariant parts for object class recognition. In *ICCV'03*.

Everingham, M., et al.(2006). The 2005 PASCAL visual object class challenge. In J. Quinonero-Candela, I. Dagan, B. Magnini,

& F. d'Alche-Buc (Eds.), *Lecture notes in artificial intelligence: Vol. 3944. Machine learning challenges. Evaluating predictive uncertainity, visual object classification, and recognising textual entailment*. Berlin: Springer. http://www.pascal-network.org/challenges/VOC/.

Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1).

Fergus, R., Perona, P., & Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *CVPR'05*.

Fergus, R., Zisserman, A., & Perona, P. (2003). Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*.

Ferrari, V., Tuytelaars, T., & van Gool, L. (2004). Simultaneous recognition and segmentation by image exploration. In *ECCV'04*.

Garcia, C., & Delakis, M. (2004). Convolutional face finder: a neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1408–1423.

Garg, A., Agarwal, S., & Huang, T. (2002). Fusion of global and local information for object detection. In *ICPR'02*.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference* (pp. 147–151).

Heisele, B., Serre, T., Pontil, M., & Poggio, T. (2001). Component-based face detection. In *CVPR'01* (pp. 657–662).

Hough, P. V. C. (1962). *Method and means for recognizing complex patterns*. U.S. Patent 3069654.

Jones, M., & Poggio, T. (1996). *Model-based matching by linear combinations of prototypes*. MIT AI Memo 1583, MIT.

Jones, M. J., & Poggio, T. (1998). Multidimensional morphable models: a framework for representing and matching object classes. *International Journal of Computer Vision*, *29*(2), 107–131.

Kadir, T., & Brady, M. (2001). Scale, saliency, and image description. *International Journal of Computer Vision*, *45*(2), 83–105.

Leibe, B., & Schiele, B. (2003). Interleaved object categorization and segmentation. In *BMVC'03* (pp. 759–768), Norwich, UK, September 2003.

Leibe, B., & Schiele, B. (2004). Scale invariant object categorization using a scale-adaptive mean-shift search. In *Lecture notes in computer science: Vol. 3175. DAGM'04* (pp. 145–153). Berlin: Springer.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 workshop on statistical learning in computer vision*.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *CVPR'05*.

Leonardis, A., Gupta, A., & Bajcsy, R. (1995). Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, *14*, 253–277.

Li, F.-F., Fergus, R., & Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV'03*.

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

Lowe, D. G. (1999). Object recognition from local scale invariant features. In *ICCV'99*.

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

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Magee, D., & Boyle, R. (2002). Detecting lameness using 're-sampling condensation' and 'multi-stream cyclic hidden Markov models'. *Image and Vision Computing*, *20*(8), 581–594.

Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, *43*(1), 7–27.

Marr, D. (1982). *Vision*. San Francisco: Freeman.

Matas, J., Chum, O., Martin, U., & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *BMVC'02* (pp. 384–393).

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10).

Mikolajczyk, C., Schmid, C., & Zisserman, A. (2004). Human detection based on a probabilistic assembly of robust part detectors. In *Lecture notes in computer science: Vol. 3021. ECCV'04* (pp. 69–82). Berlin: Springer.

Mikolajczyk, K., Leibe, B., & Schiele, B. (2005a). Local features for object class recognition. In *ICCV'05*.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2005b). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1/2), 43–72.

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

Mutch, J., & Lowe, D. (2006). Multiclass object recognition with sparse, localized features. In *CVPR'06*.

Needham, A. (2001). Object recognition and object segregation in 4.5-month-old infants. *Journal of Experimental Child Psychology*, *78*(3), 3–24.

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In *ECCV'04*.

Papageorgiou, C., & Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision*, *38*(1), 15–33.

Peterson, M. A. (1994). Object recognition processes can and do operate before figure-ground organization. *Current Directions in Psychological Science*, *3*, 105–111.

Ronfard, R., Schmid, C., & Triggs, B. (2002). Learning to parse pictures of people. In *ECCV'02* (pp. 700–714).

Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), 23–38.

Russell, B., Torralba, A., & Freeman, W. T. (2005). The MIT LabelMe database. http://people.csail.mit.edu/brussell/research/LabelMe.

Schmid, C., & Mohr, R. (1996). Combining greyvalue invariants with local constraints for object recognition. In *CVPR'96*.

Schneiderman, H., & Kanade, T. (2004). Object detection using the statistics of parts. *International Journal of Computer Vision*, *56*(3), 151–177.

Sclaroff, S. (1997). Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, *30*(4).

Seemann, E., Leibe, B., Mikolajczyk, K., & Schiele, B. (2005). An evaluation of local shape-based features for pedestrian detection. In *BMVC'05*, Oxford, UK.

Sharon, E., Brandt, A., & Basri, R. (2000). Fast multiscale image segmentation. In *CVPR'00* (pp. 70–77).

Shi, J., & Malik, J. (1997). Normalized cuts and image segmentation. In *CVPR'97* (pp. 731–737).

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for realtime tracking. In *CVPR'99*.

Thureson, J., & Carlsson, S. (2004). Appearance based qualitative image description for object class recognition. In *ECCV'04*.

Torralba, A., Murphy, K., & Freeman, W. (2004). Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR'04*.

Tu, Z., Chen, X., Yuille, A. L., & Zhu, S.-C. (2003). Image parsing: Unifying segmentation, detection, and recognition. In *ICCV'03*.

Tuytelaars, T., & van Gool, L. (2004). Matching widely separated views based on affinely invariant neighbourhoods. *International Journal of Computer Vision*, *59*(1), 61–85.

Ullman, S. (1998). Three-dimensional object recognition based on the combination of views. *Cognition*, *67*(1), 21–44.

Ullman, S., Vidal-Naquet, M., & Sali, E. (2002). Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, *5*(7), 682–687.

Vecera, S. P., & O'Reilly, R. C. (1998). Figure-ground organization and object recognition processes: an interactive account. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(2), 441–462.

Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154.

Weber, M. (2000). *Unsupervised learning of models for object recognition*. PhD thesis, California Institute of Technology, Pasadena, CA.

Weber, M., Welling, M., & Perona, P. (2000). Towards automatic discovery of object categories. In *CVPR'00*.

Wiskott, L., Fellous, J. M., Krueger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775–779.

Wu, B., & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *ICCV'05*.

Yu, S. X., & Shi, J. (2003). Object-specific figure-ground segregation. In *CVPR'03*.

Yuille, A. L., Cohen, D. S., & Hallinan, P. W. (1989). Feature extraction from faces using deformable templates. In *CVPR'89*.

Zhang, W., Yu, B., Zelinsky, G. J., & Samaras, D. (2005). Object class recognition using multiple layer boosting with heterogeneous features. In *CVPR'05*.

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, *73*(2), 213–238.