Technical Report

Recognizing Objects in Still Images and Video Streams

Adrian Ulges

a_ulges@informatik.uni-kl.de

IUPR Research Group
Faculty of Computer Science
Technical University Kaiserslautern

March 2006

Contents

Recognizing Objects in Still Images and Video Streams

Adrian Ulges
a_ulges@informatik.uni-kl.de
IUPR Research Group
Technical University Kaiserslautern

March 2006

Abstract

This paper addresses the problem of recognizing objects in visual media. Though the field has come a long way, this task is far from being solved for generic objects in arbitrary scenes. Nevertheless, recent developments have made object recognition more successful and flexible, with its most promising applications in multimedia indexing and retrieval.

The main purpose of this paper is to give a survey of object recognition in both still images and video. Also, a self-built prototype is described for the recognition of items presented to a camera. In experiments, a global, histogram-based method and a local, patch-based approach were compared, with the latter showing a higher robustness to scene changes.

1 Introduction

Upcoming multimedia applications allow users to deal with documents containing text, sound, images, and video at the same time and have a fundamental impact on the way we handle information. This "multimedia revolution" has been made possible by new developments in information technology, above all broad-band networks, mass storage, fast signal processors, software making the production and consumption of multimedia content foolproof, and efficient coding algorithms.

These advances pose both chances and challenges: chances because multimedia has the potential to fundamentally change how we gather information, how we are entertained, and how we organize the everyday information we get into contact with. For example, a stronger focus on visual representations makes it possible to perceive plenty of information at a glance (in contrast to text, which must be read in a sequential, ineffective manner). Also, video has found its way to the internet due to advanced streaming techniques.

Using this technology, we are also able to gather vast amounts of multimedia data ranging from private photo collections to distributed large-scale databases

(e.g., Google Image Search offers access to about 880 mio. images according to a press-release from 2004). The capacity of accessible information has grown to a sheer ocean – the user himself becomes the bottleneck, since we are not able to consume this complete ocean. We have to thoroughly pick the sips we take – which means that information must be selected and tailored to the user's needs and wishes. This leads to a more "personalized" way of structuring information, with concepts like video-on-demand becoming more popular (e.g., Apple PodCasts[1]) and allowing users to pick what they want to see in contrast to TV broadcasts.

However, the strong growth of multimedia data also poses difficulties and challenges. Multimedia databases usually hold vast amounts of information that cannot be captured at one glance. Elaborate visualization and browsing techniques can overcome this information overload to a certain level, but are often overstrained by the sheer capacity of visual information. This is particularly true for video content where the time dimension poses additional problems: the amount of information is usually much higher than for still images, and it is not straightforward to visualize a video at one glance. A survey of research in this area is given by Bashir and Khokhar [Bashir 03].

Data overload makes the access of information difficult. This is why efficient, user-friendly indexing and querying of multimedia databases must be realized. This paper focuses on the domain of visual information, for which three basic querying techniques have been explored: *query-by-example* (QBE) – the system delivers images similar to a sample –, *query-by-sketch* (QBS) – the system returns images showing features of a sketch drawn by the user –, and *query-by-keyword* (QBK) – the system returns images fitting a keyword typed in by the user –, with the latter currently being offered by commercial large-scale search engines as Yahoo and Google.

One important question in how to realize such queries is if (and how) to integrate the visual *content* of an image or video. While this is essential for QBE and QBS, state-of-the-art QBK systems sneak around it by exploiting meta-information like the document title.

One more step would be to determine the *semantic* of image and video content, which means to identify certain entities like objects, sites, and events. This is *not* essential for any of the three approaches: While QBK can even neglect content, QBE and QBS evade semantics by focusing on low-level generic features like color (e.g., see the IBM QBIC system [Faloutsos 94, Niblack 93]).

However, it is obvious that multimedia retrieval can benefit from exploiting the content of visual documents. Recognizing the entities present in images and video streams is the ultimate way to bridge the "Semantic Gap" [Bashir 03] between the raw bitstream of a video (or generic low-level features derived from it based on color, texture, or shape) on the one hand, and meaningful, compact description of objects, scenes, and events on the other.

[1] http://www.apple.com/itunes/podcasts/

Since the human visual system performs very well at recognizing entities from visual information, a simple way would be to manually label multimedia documents. Unfortunately, this is intractable due to the sheer amount of information. Instead, an automatic labeling of multimedia documents is required. Such an automatic *semantic indexing* is a multidisciplinary task: signal processing is used to obtain low-level features and clues, and methods from information retrieval provide similarity measures for indexing and querying. Computer vision is the core concept of extracting semantics, trying to infer the presence of objects from low-level clues.

This computer vision problem – recognizing the presence of entities in images and video – is the focus of this paper. Though the field has come a long way and a lot of work has been done in this area, the problem is far from being considered solved. This is due to a bunch of difficulties some of which are inherent when capturing natural scenes, while others are due to appearance variation of objects themselves.

Due to these problems, state-of-the-art visual search engines offer only weak extraction of semantics from visual content (e.g., the PhotoBook system[2] provides a face recognition module). However, object recognition is extremely interesting for practical applications due to its enormous potential for multimedia indexing and retrieval.

The purpose of this paper is to describe the state of the art in object recognition from visual information in form of still images and video. A survey of research work is given, challenges of the field are pointed out, and some experimental results are presented. The remainder of this paper is organized as follows: there is a bunch of problems related to objects in visual content, ranging from the detection of simple objects to object categories. To clarify these terms, definitions of these problems are given in Section 2. Afterwards, a survey of research work is given in two Sections (3 and 4). We start with methods for still images, which can be divided into the two general approaches of local and global methods (Section 3). Afterwards, extensions to multiple views of the same object, and particularly object recognition in video are addressed in Section 4. Some of the ideas presented in the survey part have been implemented in a system called VideoObjects, which learns and recognizes objects that are presented to a webcam. The prototype is described in Section 5.1. In quantitative experiments, the performance of a global and a local approach were compared. The results are outlined in Section 5.2.

Finally, a conclusion is given and challenges of the field are pointed out in Section 6.

[2] http://vismod.media.mit.edu/vismod/demos/photobook/

2 Problem Definitions

The problem of recognizing objects in images and video can be subdivided into several subproblems that are strongly related but differ in particular details, difficulty level, and input medium. While these specific terms are often mistaken or subsumed as "object recognition", a detailed definition of problems (and their specific difficulties) is presented here.

The listing of problems goes from "easy" to "difficult" corresponding to the way that object recognition has come throughout the last years.

2.1 2D Object Recognition

The first problem of object recognition in visual media focuses on the recognition of particular objects. Thereby, the pose of the object is strongly constrained:

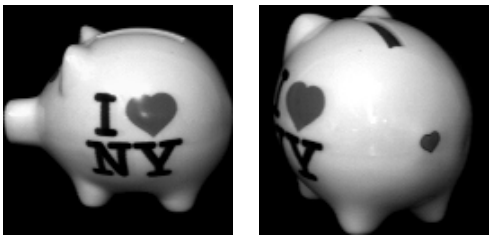
> 2D Object Recognition is the decision whether an object is present in an image. The object is specified by either a set of sample images showing the object, or by a symbolic description. The object may be shifted, scaled, or rotated in the image plane.

This definition does not include the exact *localization* of the object in the image. Also, it strongly constraints the *pose* of the object, since no in-depth rotation is allowed. Two cases in which this assumption is well-founded are a constant viewing angle (e.g., in OCR of scanned documents, where letters are parallel-projected, or for fixed cameras in industrial vision applications), or objects for which the change of appearance can be neglected and certain features remain present over the whole range of viewing angles expected (e.g., for flat objects like text). Another popular application in this area is face recognition.

Nevertheless, the problem is far from being trivial, since the appearance of the same object may differ strongly between images. This variation occurs due to several reasons:

1. features from the background – so-called "clutter"
2. partial occlusions of the object
3. while in-depth rotation is neglected so far, the object may undergo a *similarity transformation* – scale, rotation in the image plane, and translation – between several snapshots.
4. changing lighting conditions
5. changing imaging conditions like noise and blur

Variations of all these factors can influence recognition and can lead it wrong, especially if the scene changes between learning and recognition.



(a) (b)

Figure 1: sample images from a 3D object recognition database [Nene 96]. Some features vanish, while others appear between the two views.

2.2 3D Object Recognition

The assumptions on the object pose made in the last section are only true for special cases of viewing angle or object shape. To apply object recognition in a more general framework, we define 3D object recognition:

> Like in the 2D case, the problem of 3D Object Recognition is the decision whether an object is present in an image or not. Unlike in the 2D case, the image may show the object in any arbitrary pose.

This defines object recognition for arbitrary scenes, where it must be taken into account that for many objects the appearance strongly depends on the pose. Besides the difficulties presented in the last section, this is due to the following reasons:

6. self-occlusion: object features are rarely stable but can in general only be observed from a certain viewing angle. Especially when coming with a strong depth structure, objects tend to occlude their own features making recognition difficult. This can be observed in Figure 1, where some features vanish between the two views on an object, while others appear when switching between views

7. in-depth rotation causing changes in the spatial constellation of features

One way of overcoming these difficulties is to extract 3D object models. Unfortunately, such approaches usually require a controlled imaging environment or a painful calibration such that they will be neglected here. Instead, many researchers stick with *appearance-based* approaches that do recognition by image features only.

2.3 Object Recognition in Video

Another object recognition problem results from using videos instead of still images as input:

> OBJECT RECOGNITION IN VIDEO is the decision whether an object is present in a video stream. The pose of the object may vary between frames and is not constrained.

Object recognition in videos is strongly related to 3D object recognition, since a video usually provides a set of views of the object in several poses. However, there are also differences: on the one hand, video is less constrained in a way that the pose of the object is usually not known. On the other hand, additional information comes with the temporal order of frames. Usually, this induces a strong similarity between adjacent frames as well as *motion* that can be used to segment a foreground object from the background – a concept that can be helpful to reduce the influence of clutter.

2.4 Object Category Recognition

Another important problem addressed in more recent research is the recognition of *object categories* instead of individual objects.

> OBJECT CATEGORY RECOGNITION is the decision whether an instance of a certain object category is present in an image. The object category is specified by either a set of sample images showing some instances, or by a symbolic description.

Object Category Recognition leads to a wider range of applications than the detection of specific objects – for example, it allows for a more natural indexing and querying of visual content. A user will usually not ask for a specific instance of an object, but rather for general concepts, e.g."Give me all images showing the frontal view of a car".

On the other hand, object category detection is a more difficult task, since appearance variations do not only stem from changes of the capturing conditions listed in Section 2.1, but also from distances between the instances of an object class. In fact, objects within the same semantic category may look rather different, as can be observed for faces in Figure 2. However, for most object categories there are fundamental characteristics shared by nearly all instances (e.g., the "eye" feature in a face). An object category recognition system has to model these shared properties, while spurious object-dependent features (e.g., glasses or a hat) have to be filtered out as well as scene-dependent ones (e.g., background clutter).

Many approaches in this area have concentrated on specific object categories, like faces [Viola 04] or text [Chen 04]. However, we will also address more recent work providing generic ways to build models for arbitrary object categories, e.g.[Fergus 03].

Figure 2: samples of an object category (image taken from [Jesorsky 01]).

3 Recognizing Objects in Still Images

This section gives an overview of state-of-the-art methods for the problems introduced in the last section where still images are assumed as input. The methods presented are divided into *global* and *local* approaches, although this subdivision is somewhat fuzzy and some hybrid methods can be found. While global methods base recognition on one global feature vector per image and make decisions on image level, local approaches view an image as a set of local *samples* (from now on referred to as *patches*) such that recognition makes decisions on patch-level.

Similar terms have also been used by Schiele and Crowley before [Schiele 00], who compared a global and a local version of a histogram-based approach. A similar study for object recognition in video will be presented in this work after the survey (see Section 5.2).

Global methods will be introduced first (Section 3.1), followed by local ones and a discussion of both (Sections 3.2 and 3.3).

3.1 Global Approaches

The underlying concept of global object recognition methods is that the appearance of an object is described by a global feature vector, and classification is carried out on a global level. Nevertheless, the specific features and decision rules used may be manifold.

A listing of some global object detection approaches is given in the following.

Histograms Histograms can be seen as discrete, empirical approximations of probability density functions. They provide compact, global measures of image features, and have been used for object recognition in various ways.

A simple way to use them is to associate a histogram with each image and recognize object via the similarity of these histograms.

Horecki et. al. [Horecki 99] use an extended approach to localize objects in a cluttered scene. *Color histograms* are learned from object images with minor background influence and afterwards used to track the object in a *sliding window*. This gives so-called *interest maps*, peaks of which are potential object positions. The performance of several similarity measures for histograms are compared as well as color spaces used.

Schiele and Crowley [Schiele 00] use histograms of gray value image properties, more precisely *Gaussian derivatives* indicating the local gradient strength.

The fundamental benefit of histograms is that if the features are chosen properly, histograms are invariant to rotation and scale changes. Their applicability for object recognition under these transformations has been validated in experiments [Schiele 00].

PCA Nayar et. al. [Nayar 96] view images as vectors with pixel intensities as components. Unfortunately, classification using this representation suffers from the high dimensionality of these vectors. Principal Component Analysis (PCA) is used to overcome this problem: a basis of eigenvectors is chosen based on the distribution of data. Given this eigenbasis, samples can be projected to a low-dimensional subspace spanned by the first k eigenvectors such that the *variance* of the data is preserved as far as possible [Duda 00].

PCA yields a low-dimensional, compact representation for each object image in this so-called *eigenspace*, making rapid indexing and retrieval of visual information possible. The method is also well-known in face recognition as *eigenfaces* [Sirovich 87].

Nayar et. al. [Nayar 96] also present an extension of the approach to 3D object recognition – while in face recognition frontal views are assumed, Nayar et. al. work with images taken from multiple perspectives of an object. Since these images are assumed to be taken from regular, known view angles, a neighborhood relationship is established over shots of similar illumination and perspective. Consequently, the representation of an object is a set of points in eigenspace linked according to this neighborhood relationship – a so-called eigenspace *manifold*. Images of unknown objects can be classified by projecting their PCA representation to the object manifolds and measuring the distance.

Splines on Gradient Fields Javed et. al. [Javed 04] developed a global method for object recognition in video. They extract the field of gradient orientations from each frame and fit a spline to it. The coefficients of this spline interpolation serve as feature vectors for classification.

Recognizing Objects by their Motion Arbel et. al. [Arbel 00] examine if objects can be characterized by their *motion structure* in video, neglecting texture and color information. When moving an object parallel to the focal plane of the camera and assuming constant velocity, the motion of a pixel is related to its distance from the camera via the laws of perspective projection. This gives a characteristic "depth map" for objects, which is used as a feature vector for classification after applying PCA.

Wavelet Coefficients A highly elaborate approach has been developed by Schneiderman [Schneiderman 00b, Schneiderman 00a]. According to the authors, the method is the first one that can handle object categories and changes of object pose at once.

This is achieved by training separate detectors for separate poses (e.g., one classifier for each of 8 possible frontal views of cars as illustrated in Figure 3).

The design of the single classifiers is then based on local *patterns* p_i, which are empirically chosen *groups* of wavelet coefficients from a wavelet transform of the image. This yields features at different scales, positions, and orientations.

Figure 3: sample images taken from a 3D object category recognition problem. Eight views of cars taken from [Schneiderman 00b], where a separate classifier is trained for each view.

The core of the probabilistic model is the class-conditional density

$$P(\,image\,|\,object\,) = \prod_i P(p_i|\,object\,), \tag{1}$$

where independence is assumed between different patterns. This does not hold for the wavelet coefficients *within* the same pattern such that the probabilities $P(p_i|object)$ are derived from a *joint histogram* for each pattern learned in training. Consequently, intra-group dependencies are respected to achieve a balance between model compactness and model accuracy. To recognize objects in cluttered scenes, object recognition is applied to a *sliding window* over the image.

According to the authors, the method handles intra-category appearance variation well. It gives impressive recognition rates for images of cars and faces. However, two major issues remain unsolved: first, the training images used seemed high-quality, which means that the object takes most of the image and the clutter has a minor influence (see Figure 3). Second, the choice of wavelet coefficients is highly ad-hoc. It is unclear whether the coefficients used are a good choice for other object categories.

The approach is strongly related to local, patch-based methods, since the "pattern" features used represent local properties, in contrast to, color histograms for example, where each bin represents pixels spread over the image. However, in contrast to manually chosen features coming with an implicit semantic, patch-based approaches view an image or video frame as a set of local generic samples derived without knowledge of the image.

3.2 Local Approaches

Global methods as introduced in the last section model the appearance of an object in terms of global features like color histograms or gradient fields. They

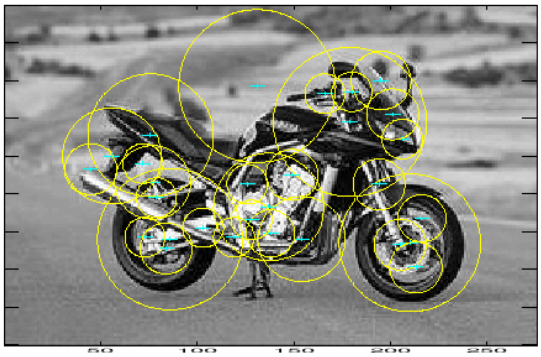


Figure 4: patch-based object recognition: an image is modeled by a set of local features derived from it (in this case, circular regions of different scales). Object recognition is based on these *patches* (image derived from [Fergus 03]).

often have problems if an object is not dominant in an image but the appearance is influenced by features from the background called "clutter". Another problem related to this is partial occlusion of the object. The influence of these phenomena on global methods is often unpredictable.

This is why recent research efforts have concentrated on *local* methods, which describe an image by a set of local features (or *patches*) as illustrated in Figure 4. In this visualization, patches at certain points of interest are highlighted. Typically, local methods extract such patches and associate each of them with a so-called *local descriptor* representing the *appearance* of the local surrounding. The presence of the object is equivalent to the presence of a certain *configuration* of local features, which refers both to the *appearance* and the *spatial constellation* of patches (e.g., the most discriminative features for the motorbike in Figure 4 might be two dark circles – one on the left, one on the right – corresponding to the tires.

This local approach – often referred to as *patch-based* or *part-based* object recognition – is usually applied without segmentation. To a certain amount – e.g., according to Fergus et. al. [Fergus 05a] the object has to "occupy a reasonable proportion of the image" –, it has the potential to overcome background influence and partial occlusion using methods of *Robust Statistics* [Huber 82].

Many methods have been published in this area, which have successfully been applied to object detection [Schmid 97], 3D object detection [Lowe 01] and object category detection [Csurka 04, Fergus 03].

Instead of listing the approaches one by one, we present a categorization that focuses on four key aspects of local object recognition. Though all methods have the patch-based view of an object in common, they differ significantly in their way of handling the following four key questions:

1. Which *interest regions* are used for the positions of patches?

2. What *local descriptors* are used to model the appearance of patches?

3. How is the *spatial constellation* of patches modeled?

4. What Pattern Recognition methods are used for *classification*?

In the following, we present answers to these key questions from the literature.

3.2.1 Interest Region Detectors

A wide variety of local feature detectors has been proposed – for example, see [Mikolajczyk 05] for another overview. One criterion for a good feature is its *repeatability*: since the fundamental purpose of local features is to represent a property of the *object* and abstract from the specific capture conditions, features should be detected reliably even in case of scene changes. Also, features should be *discriminative* in a way that they are unique and can easily be matched between images showing the same object. Note that repeatability and distinctiveness can be contrary goals – discriminative features are probably associated with large image regions, which are prone to occlusion and warping and thus less reliable.

Due to the vast amount of work on this topic, the following listing is incomplete. Its purpose is to describe some of the most popular ideas to achieve distinctiveness and repeatability including point, contour, and region features. Furthermore, the last passage discusses a quantitative evaluation of several interest region detectors.

Corners Conceptually, a corner is a point where the gray value changes in multiple directions at once. In contrast to edge points, its location can be uniquely determined by its local surrounding, so that corners make good features to track and have extensively been used in stereo vision and motion estimation. Two classical corner detectors are the Harris detector [Harris 88] and SUSAN [Smith 97].

Unfortunately, these feature detectors neglect effects of pose changes. Like all image features, corners are subject to an appearance change if the distance of the camera varies (this causes a *scale* change in the image) or due to a change of viewing angle (locally, this effect is approximated by an *affine transformation*).

This is not taken into account by basic corner detectors – they are neither scale nor affine invariant.

Scale and Affine Invariance To some extent, a lack of feature invariance can be overcome by processing the input image at different scales. An alternative way [Mikolajczyk 04] is to extend the conventional Harris detector and determine a *characteristic* scale for each keypoint (see also [Lindeberg 98]). Therefore, a range of scales is searched to maximize an information measure called the "Harris cornerness". In a second step, the local surrounding is transformed to an isotropic state based on its *second moment matrix* [Mikolajczyk 04]. Thus, the resulting interest regions are not only assigned a characteristic scale, but

also an elliptical shape that follows affine transforms of the local surrounding. This is why the resulting feature detector is called *affine invariant*.

The effect is illustrated in Figure 5: for each keypoint, a characteristic local surrounding is determined (Figure 5(a)). If this local surrounding is warped due to a pose change of the object, the feature is tracked reliably and the local surrounding is adapted (Figure 5(c)).

The resulting region detector has the capability not only to track a feature in a distorted image, but also to compensate for the distortion of the local surrounding – this boosts matching between images the better the more precisely the projected elliptical neighborhoods overlap.

(a) (b) (c)

Figure 5: due to a pose change, the local surrounding of 5(a) is distorted in 5(b). A circular neighborhood as in 5(b) is not suitable to describe the patch, since it covers a different potion of image content. However, an affine region detector compensates for this fact by warping the local surrounding 5(c) such that the neighborhoods in 5(a) and 5(c) cover the same image region (image taken from [Mikolajczyk 05]).

Salient Points Another problem with corner detectors is that the resulting features are usually not distributed regularly over the image but often focused in regions of strong contrast. A way to overcome this is the wavelet-based *salient points* method by Loupias and Sebe [Loupias 99]. The basic idea is to search the coefficients in the wavelet transform $\mathcal{W}f$ of the input image f for peaks called "salient points". For more details, see Appendix B.

Salient points have fundamentally different properties compared to corner detectors. Since strong wavelet coefficients do not demand strong derivatives in multiple directions, salient points can as well be edge points. On the one hand, these may be valuable features for object detection – e.g., Berg [Berg 04] argues that some objects may show only few corners. On the other hand, edge points make rather bad features for matching in motion tracking or stereo vision, because their position is not uniquely determined.

Salient points are inherently extracted at multiple scale levels, since they are searched for in the whole wavelet transform $\mathcal{W}f$. And there is another property that distinguishes salient points from corners: due to the hybrid nature of each

coefficient (it corresponds to both a scale and a location), salient points are more spread over all image regions and not restricted to areas with strong contrast, as for other interest point detectors. This is illustrated by examples in [Loupias 99].

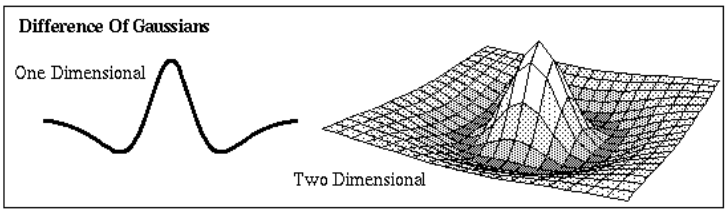


Figure 6: one- and two-dimensional examples for a difference of Gaussian functions (image taken from[4]). If convolved with a signal at the right scale, regions with strong contrast to their surrounding give extrema. Note the strong similarity to the Laplace (or "Mexican Hat") operator.

Difference-of-Gaussians (DOG) Another method is to convolve the image with a *difference of Gaussians* function and detect maxima obtaining so-called *blob* features. This approach is followed by the popular SIFT operator [Lowe 04]. It operates in the discrete *scale space* of the input image, which is obtained from convolving the image with Gaussians at several scale steps [Lindeberg 99]. At each level, the image is convolved with a difference of Gaussians (DOG), which is illustrated in Figure 6. The resulting signal in scale space is searched for maxima yielding features with an inherent scale.

The features detected are *blobs*, regions of strong contrast to their surrounding. A perfect example for this is a white circle on black background. If convolved with a DOG at the right scale, it gives a strong peak.

Rotation invariance can be achieved by further enhancements. For example, a characteristic rotation angle can be assigned to each feature by computing a dominant gradient direction in the local surrounding [Lowe 04]. Furthermore, a gradient-based local descriptor is used (see Section 3.2.2).

Sample Points A simple alternative to interest region detectors is to obtain patches simply from sampling the image regularly or randomly. In [Deselaers 05], such patches were used for object recognition and found helpful, especially for prominent large monotonous areas: in these, only few interest points will be detected, but nevertheless they provide a strong property to discriminate between objects. This can be exploited by sample points (a similar idea has been followed in [Schiele 00]).

Maximally-stable Extremal Regions Matas et. al. [Matas 02] developed another interest region detector and applied it for stereo matching. While other

[4] http://www.liden.cc/Visionary/Visionary_d.html

operators so far extracted point features, their approach operates on image *regions*. The method is based on the Watershed transform[5]: starting from a maximal threshold, the image is iteratively binarized while lowering the threshold step-wise down to the minimum. During this process, the topology of image regions changes, and bright regions are fused more and more – however, some of them may evade these fusions over a wide range of threshold values, especially if they have a strong contrast to their surrounding. These regions are picked as interest features.

Local Entropy Gilles [Gilles 98] characterizes the saliency of features by the *randomness* of a local surrounding. Using a sliding-window approach, maxima in the entropy of the local histogram are used as feature points. This approach does not take scale into account. Furthermore, the method gives many local maxima in case of complicated, strongly textured areas. Kadir and Brady [Kadir 01] argue that such areas are not *salient*, since the saliency of an observed feature – the fact that a feature is noticed "at one glance" – demands (besides geometric aspects and complexity) *rareness*. Similarly, Schiele[Schiele 00] characterizes salient points as "*unique* points that are helpful to distinguish objects".

Both problems with Gilles' approach – lack of scale and lack of true saliency – can be faced by entropy maximization over *scale space*. It is argued that features are more salient if they are stable only over a *small* range of scales instead of a wide one [Kadir 01]: e.g., consider an edge point and a typical blob feature, a white circle on black background. While the edge point will have the same entropy no matter how the surrounding is scaled, the more distinctive blob gives a strong maximum of entropy at a particular scale only. From these thoughts, a saliency measure is derived that is supposed to suppress strongly textured, but self-similar areas.

Curve Features A last feature class beneath points and regions are curves in an image, which are of special interest if the *outline* of an object is more discriminative than its texture. One example based on Canny [Canny 87] edges is described in [Fergus 05b].

Performance Evaluation A quantitative comparison of some popular affine feature detectors can be found in [Mikolajczyk 05], where interest features from reference images were tracked in distorted image versions (blur, illumination change, 3D viewpoint change, rotation, scale). Each local feature is associated with an elliptic neighborhood which makes the feature *affine invariant* (as illustrated in Figure 5).

The quality of an affine keypoint detector is now measured using two criteria: the *repeatability* with which a keypoint is tracked in the distorted image and its *accuracy*, the percentage of overlap between matched neighborhoods. In

[5] http://cmm.ensmp.fr/~beucher/wtshed.html

[Mikolajczyk 05], these criteria were evaluated quantitatively over a set of test images[6]. The ground truth for tracking interest features between images is determined by estimating a homography between the reference image and its warped version.

Results indicate that Maximally-stable Extremal Regions [Matas 02] and the enhanced scale and affine invariant version of Harris [Mikolajczyk 04] performed well [Mikolajczyk 04].

3.2.2 Local Descriptors

Following the patch-based approach, local features are not only characterized by their position in the image but also by the local *appearance* of the surrounding patch. Like for feature detection methods, there is a wide variety of methods to extract feature vectors that describe patch appearance, so-called *local descriptors*.

The fundamental purpose is to describe *object* properties independently from the specific capture. Thus, invariance is a basic goal for local descriptors as well as for feature detectors. Ideally, the local descriptor should remain the same under varying illumination, scale, or rotation. Often, invariance is difficult to achieve and subsumed by the weaker property of *robustness*, which allows minor changes.

Another goal is a high *distinctiveness* of the descriptor. It can be difficult to satisfy both criteria at the same time: for example, the local gray value mean in a local surrounding is *invariant* to rotation, but on the other hand it has a rather weak distinctiveness.

To overcome a lack of invariance, several workarounds have been suggested: for scale invariance, local patches can be processed at multiple scale levels. For illumination, the gradient strength is used instead of the absolute gray value, or an adaptive gray value normalization is done. Rotation invariance can be achieved by using histograms or by determining a *characteristic* orientation for each patch based on local features like the dominant gradient direction in a local surrounding [Lowe 04].

An evaluation of local descriptors has been presented by Mikolajczyk and Schmid [Mikolajczyk 03], who tested the performance of local descriptors in the context of matching between two slightly modified image versions (rotation, scale, 3D viewpoint change as for stereo tasks, blur, and illumination). The local descriptors were tested for several interest region detectors. It was found that the SIFT descriptor performed best. However, these results do not necessarily hold for object category detection. First, the requirements for a local descriptor may be different since matching shall occur under a higher appearance variation. Second, setting up a ground truth for quantitative testing is painful.

In this section, we list some popular local descriptors.

[6] http://www.robots.ox.ac.uk/~vgg/research/affine/

Pixel Values A straightforward way to describe a patch feature it to use pixel values in a local surrounding. This is simple but usually yields very high-dimensional feature vectors making a later processing painful. Furthermore, image patches contain noise and high frequencies, which might have a negative influence on similarity measures for matching. For example, Lowe [Lowe 04] states that raw image content is sensitive to changes of 3D viewpoint or slight rotations. This is also found in the context of stereo matching [Ulges 04].

PCA A standard technique to reduce the dimensionality of sample vectors is Principal Component Analysis (PCA) [Duda 00]. Using a linear eigenanalysis, high-dimensional samples (e.g., vectors of local pixel values) can be projected to a lower-dimensional subspace, whose basis of eigenvectors is chosen such that the variance of samples is optimally preserved.

DCT One problem with PCA is that it demands *learning* the basis of eigenvectors from a training set. Experimental results in [Kölsch 03] show that like the PCA favors low-frequency components over details, and that comparable local descriptors can be obtained by applying the Discrete Cosine Transform (DCT) to the image and selecting a subset of low-frequency components from the DCT-transformed patch. This offers a data-independent way of dimensionality reduction.

SIFT The SIFT feature extraction method by Lowe [Lowe 04] does not only provide a keypoint detector, but also a local descriptor. Its general idea is illustrated in Figure 7: the local surrounding of a feature vector is partitioned into bins, and for each bin a histogram of gradient orientations is computed. Taking the bins of all histograms together yields a descriptor of gradient orientations in the patch. Due to a clever weighting and interpolation, the descriptor changes smoothly with variations of the local surrounding.

The model is biologically inspired – the human visual system perceives gradients in a similar manner showing robustness to small shifts on the signal.

The resulting descriptors are claimed to be scale invariant due to feature detection in scale space, rotation invariant due to estimating a characteristic orientation, and robust against illumination changes (the gradient is used instead of image intensities). They have proven a very good performance in a quantitative evaluation [Mikolajczyk 04].

Differential Invariants Other researchers have used certain local *invariants* as descriptors. These can be based on gray values, e.g.using *moments*, or on Gaussian derivatives that are obtained by differentiating the input image I convolved with a Gaussian kernel $\mathcal{N}$:

$$L_i(x) := (\frac{\partial}{\partial x_i}\mathcal{N} * I)(x).$$

The definition of higher-order derivatives is straightforward.

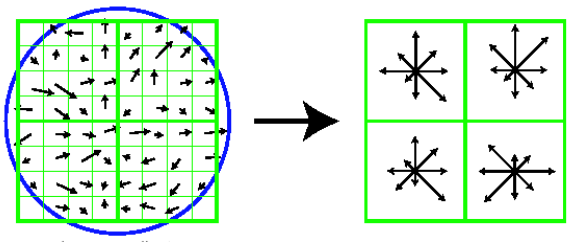


Figure 7: an example for the SIFT descriptor: a local neighborhood is divided into spatial bins, for each of which a histogram holds the local gradient distribution. Combining these histograms give the feature vector (image taken from [Lowe 04]) – in this example, there are 4 bins, each with 8 arrows corresponding to 4 gradient orientations. This yields a 4×4-dimensional feature vector.

If Gaussian derivatives are combined properly, descriptors can be obtained that are rotation invariant. Examples are the Laplacian $\sum_i L_{ii}(x)$ or the squared gradient magnitude $\sum_i L_i(x) L_i(x)$. The resulting features are called *differential* invariants.

A sequence of third-order differential invariants has been used as a local descriptor in [Schmid 97]. The choice of features in this case is rather limited, since only certain combinations of Gaussian derivatives lead to rotation invariance. Furthermore, results in [Mikolajczyk 03] indicate a rather weak distinctiveness.

Feature Discretization A common way to speedup feature matching is to apply an additional *discretization* (also referred to as *feature clustering*) as a post-processing [Keysers 06, Fergus 05a]. Typically, the feature vectors are replaced by the outcome of a clustering on training data (e.g., using k-means). In recognition, a patch p is not presented by its feature vector, but by the cluster $C(p)$ it is assigned to.

First, this reduces the amount of data to 1 dimension per patch. Second, it reduces the number of potential correspondences when it comes to matching patches from two sets (e.g., an image and an object model) by demanding that matches belong to the same cluster.

Feature discretization is also used by another group of object recognition approaches inspired by methods from the *text processing* domain (e.g., see [Sivic 03, Fergus 05a]). In this context, the clusters resulting from discretization are often referred to as a vocabulary of *visual words*, which are clusters of similar patches.

3.2.3 Using the Spatial Constellation of Patches

The local features of many real-world objects tend to appear in repetitive spatial constellations – one example is the motorbike illustrated in Figure 4, another one is a human face with eyes, nose, and mouth at fixed positions relatively to

each other.

In the last Section, it was described how to extract those local features and how to describe their appearance. Now, the next question is how to exploit spatial relationships between those features. This leads to the so-called *correspondence problem* of finding matches between an object model and an image to be recognized. Solving this problem is one of the bottlenecks in object recognition, since the space of potential correspondences is usually vast. Many strategies to face this problem and exploit spatial constellation are introduced in the following, ranging from simple heuristics to approximation techniques to statistically optimal search strategies.

No Spatial Relationships Like for local descriptors, we start with the most simple way, namely to neglect spatial relationship between patches completely. According to this approach, an object (or object category) is usually characterized only by the local descriptors of its patches. This leads to various decision strategies that will be discussed in Section 3.2.4. An alternative is to work with discretized descriptors and use their *distribution* over the clusters of the "visual vocabulary" described earlier in this section: given clusters $C_1, .., C_N$ and an image with patches $p_1, .., p_n$, this distribution is defined by

$$(\#p_i : C(p_i) = C_j)_{j=1,..,N} \tag{2}$$

Note that this sequence is a patch *histogram* holding information similar to a word vector in text processing. While the latter holds information on the frequency of words in a text document, $(\#p_i)$ provides the same information for "visual words" in an image.

Approaches neglecting spatial relationships are popular due to their simplicity [Deselaers 05, Fergus 05a, Kölsch 03] and have also shown a surprisingly good performance in practice. It seems that these approaches work well for *object* recognition but run into problems when it comes to *object category* recognition, especially for object classes where color and texture vary but shape is very discriminative (e.g., "coffee cups").

Histograms of Relative Positions Another approach using feature discretization has been proposed by Agarwal et. al. [Agarwal 04], who build a vocabulary of visual words by clustering. For recognition, a binary vector $V = (v_1, .., v_n)$ is extracted for each image where the coordinates v_j indicate the presence of a patch belonging to cluster C_j – just like in text processing, where documents can be represented as boolean vectors of word occurrences.

Furthermore, the approach is extended to take spatial relationships into accounts using *bins* of relative positions between patches. Given a reference patch, the image space is partitioned into 20 bins using 5 distance levels and 4 direction levels, and V is extended with additional coordinates v_{ij}^k indicating the presence of visual word w_j in bin k ($k = 1, .., 20$) relative to an occurrence of w_i.

Obvious problems with this approach are its very high-dimensional feature vectors in the order of several mio. dimensions (which is tried to be overcome using a winnows classifier suitable for sparse samples) and a lack of robustness against slight position changes.

Heuristics for Joint Optimization The object recognition approach of Burl et. al. [Burl 98] views the spatial relationships and appearance in one probabilistic framework. Given an image I represented by its patches $X = x_1, .., x_n$, the likelihood ratio is used for classification:

$$\Lambda = \frac{p(I|object)}{p(I|background)}$$

Λ is rewritten using a latent vector random variable X, which represents hypotheses for positions of object patches in the image:

$$\Lambda = \frac{\sum_X p(I|X, object) \cdot p(X|object)}{p(I|background)}$$

This converts to the following log-likelihood ratio, if independence of patch appearances is assumed. Furthermore, a single optimal constellation hypothesis X_0 is assumed since summing over all positions is infeasible:

$$\log \Lambda = \sum_i log \frac{p(x_i|X, object)}{p(x_i|background)} + K \cdot \log X_0$$

Optimizing $\log \Lambda$ means to jointly optimize local appearance (the first term) and spatial constellation (the second one). The factor K can be used to balance both influences.

For recognition, a part detector is run to extract the patches p_i. Then, the constellation hypothesis X_0 is chosen and the likelihood ratio can be computed. The key step is the choice of X_0 from the patches. Therefore, three heuristics are discussed:

1. for each match in the object model, find a "most similar" match in the scene based on appearance only. This automatically leads to a spatial configuration ("appearance implies shape")

2. find an optimal shape match neglecting appearance. This yields appearance correspondences. ("shape implies appearance")

3. find a set of reliable initial matches. If assuming a transformation of at most four parameters between the images (e.g., a similarity transform consisting of isotropic scale, rotation, and translation), each pair of such matches implies a global constellation for all other parts. Optimize $\log \Lambda'$ repeatedly varying the pairs of initial matches ("joint optimization")

Optimality cannot be guaranteed for any of the methods. Approaches (1) and (2) are greedy techniques. The last method uses a local optimization, which faces the problem of bad starting values.

Least Squares Lowe [Lowe 01] follows a voting approach. Between patches in input image I and in the model M, correspondences $\{(x_i^I, X_i^M)\}$ are established. In an additional postprocessing, spatial constellation is exploited by demanding these matches to give a consistent global similarity transformation T_θ. T_θ maps feature locations from I to M. The transformation parameters θ are obtained using least-squares, thus minimizing the localization error:

$$E(\theta) = \sum_i (X_i^M - T_\theta(x_i^I))^2$$

The decision whether I matches M is made by thresholding with the least squares error.

The approach has two inherent problems: First, matching and transformation estimation are separated such that errors in matching cannot be undone in parameter estimation. Second, least squares optimization is strongly influenced by such outliers.

RAST A method that overcomes both weaknesses of Lowe's approach is the RAST algorithm [Breuel 92], which solves correspondence and parameter estimation in a joint framework. Keysers [Keysers 06] follows this approach to solve the correspondence problem in object recognition. RAST does not model matches explicitly, but finds a global *mapping* T_θ between a model and an image (e.g., a 4-parameter similarity transform). The input consists of an object model O consisting of patches $o_1, .., o_m$ and an image X with patches $x_1, .., x_n$. Each patch consists of image space coordinates and an appearance vector α, e.g.$x_i = (\mu_i, \alpha_i)$. Given O and X, RAST searches the parameter space $\{\theta\}$ optimizing the *likelihood* of the observed image. Independence of patches is assumed:

$$\begin{aligned}\hat{\theta} &= \arg\max_\theta p(X|O,\theta) \\ &= \arg\max_\theta \prod_i p(x_i|O,\theta)\end{aligned}$$

For the class-conditional density $p(x_i|O,\theta)$, a truncated Gaussian $\mathcal{N}$ is chosen:

$$p(x_i|O,\theta) = \begin{cases} \mathcal{N}_{T_\theta(o_{i^*}),\sigma^2}(x_i) & \exists o_{i^*} : \mathcal{N}_{T_\theta(o_{i^*}),\sigma^2}(x_i) > T \\ T & \text{else} \end{cases}$$

This models the presence of background features: if the image patch x_i occurs due to the object, the corresponding model patch o_{i^*} will be similar to x_i and mapped near to x_i such that $T_\theta(o_{i^*}) \approx x_i$. A diagonal covariance matrix $\sigma^2 I$ is assumed, as well as a uniform distribution T for background patches. . After further simplification and taking the logarithm, we obtain

$$\hat{\theta} = \arg\max_\theta \sum_i q_{O,\theta}(x_i)$$

where $q_{O,\theta}$ gives "quality votes" for each image feature:

$$q_{O,\theta}(x_i) = \begin{cases} 1 - \frac{||x_i - T_\theta(o_{i^*})||^2}{\epsilon^2} & \exists o_{i^*} : ||x_i - T_\theta(o_{i^*})||^2 < \epsilon^2 \\ 0 & \text{else} \end{cases}$$

As mentioned above, patch vectors contain both the location and appearance: $x_i = (\mu_i, \alpha_i)$. Consequently, $||.||$ is the Euclidean distance in the joint domain of image coordinates and intensities.

In practice, location and appearance are dealt with separately. The appearance can be discretized and matches are accepted only within the same cluster. The distance measure $||.||$ is replaced with the Euclidean distance for image coordinates only.

Optimizing the likelihood is done by searching the parameter space $\{\theta\}$ in a branch-and-bound manner, guaranteeing a global optimum. Note that while other methods like in Burl et. al. [Burl 98] try to establish correspondences between patches explicitly, the RAST algorithm yields matches *implicitly* once θ is known. Classification of the image I might be done using the overall quality, which is an equivalent to the log-likelihood.

The Constellation-Based Model A highly elaborate approach in the area object recognition is the *constellation-based model*, which views the appearance of patches and their spatial constellation (the overall *shape* of an object) in a joint probabilistic framework.

The model is inspired by the work of Burl et al.[Burl 96] as it is based on maximizing appearance and shape terms in a joint likelihood. Several variations can be found in the literature [Weber 00, Fergus 03, Fergus 05b]. While earlier work is focused on object recognition and skips automatic learning, later approaches even include learning object category models in an unsupervised manner from weakly labeled images [Fergus 03]. Neglecting some variations between the concrete approaches, this section tries to present the essence of the model.

THE MODEL: The central idea is that a set of *characteristic patches* M forms the model for the object or object category to be recognized (e.g., the eyes and the nose are characteristic for human faces). Given a new image, patches P are extracted that make potential candidates for these object patches. A so-called hypothesis $h : P \to M \cup \{background\}$ maps image features to model patches. The space of possible hypotheses may become vast: $|h| \in O(|M|^{|P|})$, and distinguishing "good" hypotheses from "bad" ones is a key problem.

For each hypothesis, a class-conditional density $p(P, h \mid \theta)$ is derived using a complex model including distributions for the appearance, scale, and spatial constellation of patches. The concrete realization of these distributions can depend on the approach – Fergus et al.[Fergus 03] choose Gaussian distributions, while Weber et al.[Weber 00] discretize the appearance of patches by a preceding feature clustering.

The parameters θ of these distributions describe the appearance of an object, including distributions for the appearance of characteristic patches as well as for their spatial constellations. θ is derived from a set of training images.

RECOGNITION: In recognition, we theoretically have to marginalize over all hypotheses to obtain the likelihood $p(P|\theta) = \sum_h p(P, h|\theta)$. The basic question is how to handle the amount of hypotheses. If summing over all hypotheses in a brute force manner, the number of model features must be kept very low (e.g., $|M| = 5$). Another approach is to only sum over a subset of *promising* hypotheses, or to even search for an optimal hypothesis in a greedy manner [Burl 96] by picking pairs of features and inferring the remaining patches. Other work by Fergus et al.[Fergus 05b] replaces the joint density for the spatial constellation of patches with a star-based model including only pairwise dependencies with a so-called "landmark" patch. In this case, sums over the hypothesis space can be evaluated in $O(|M|^2 \cdot |P|)$ instead of $O(|M|^{|P|})$. However, the selection of a stable landmark is an open problem.

LEARNING: An open question left is how to determine the parameters θ – to *learn* object category models – from sets of unsegmented images labeled with object names. While this has not been addressed by earlier approaches, recent work like [Fergus 03] faces the problem, especially two key difficulties:

- *part selection*: Select the most discriminative patches that represent the "essence" of an object category best. Ignore clutter patches.
- estimate the associated model parameters θ that determine where model patches are to be expected in the image and what appearance they have.

Both problems can be addressed at once by data fitting using an Expectation Maximization (EM) algorithm (for a short introduction, see Appendix A). This generic scheme is an iterative technique to find a maximum-likelihood solution for parameters θ in the presence of missing data. EM is extremely suitable in our case, since we have missing data in form of occluded object patches (the ones for which $h(p) = background$). EM also has the capability to "select" the best model parts in a maximum likelihood sense and ignore clutter.

DISCUSSION: With the complexity of the constellation-based model comes a high flexibility. Since both appearance and shape are modeled via parameters in one joint framework, the model is capable of a wide range of object categories, for some of which the appearance might be more restricted (e.g., for "spotted cats" the texture is more discriminative while the shape may vary), while others are characterized more by a restricted shape (e.g.tea cups).

One surprising fact is that only very few (in the order of 5) patches have been used in the corresponding object models so far. One reason for this is that learning is very time-consuming. Another one is that more model patches – thus, more parameters – might lead to overfitting for the small training sets used.

Although only very few patches per object category are used so far, error rates seem competitive.

3.2.4 Classification

The last key question to be answered is how to make a good decision whether an object is present in the image based on *local* features.

This topic is also strongly related to the way *appearance* and *shape* are modeled. Many approaches that take shape into account – e.g., the constellation-based model [Fergus 03] – use probabilistic frameworks. In these cases, the decision rule follows directly from the model using Bayesian methods. Consequently, we will not discuss these approaches further but focus on approaches that neglect spatial relationships. An overview and comparison of some methods is given in Deselaers et al.[Deselaers 05].

Patch Histogram Similarity One simple way is to associate an image with its patch histogram as introduced in equation (2). A new object image can then be classified using histograms of a set of training images.

Naive Bayes On a local level, the patches X in an input image are viewed as *samples*. Usually independence is assumed, which leads to a *Naive Bayes* approach:

$$P(X|O) = \prod_i P(x_i|O)$$

A feature discretization is used to cluster patches into classes such that the class-conditional probabilities $P(x_i|O)$ can be obtained as relative frequencies from histograms.

AdaBoost Another possibility [Opelt 04] is to view the presence of a patch in an image as a binary *weak classifier* and combine these using *AdaBoost* [Freund 96]. Given a set of training images with object labels viewed as sets of patches $X = \{X_i = \{x_{i1}, .., x_{in_i}\}\}$, the method precomputes a distance between each feature-image pair:

$$d(x_{ij}, X) = \min_{x \in X} |x_{ij} - x|$$

Iteratively, the feature x_{ij} is picked that minimizes classification error when thresholding over all images, and AdaBoost reweights all images such that "difficult" ones are given more attention. The resulting classifier is a linear combination of the weak "patch classifiers" obtained in each iteration. The method has also successfully been applied before to specific object categories like faces [Viola 04] and text [Chen 04]. Its recognition rates for object detection on standard databases seem competitive.

Direct Voting Direct Voting is another appearance-only approach viewing an image as a set of local features $x_1, .., x_n$. From a set of training images, patches $X_1, .., X_N$ associated with object labels $L(X_j)$ are extracted. The decision rule is the following: for each feature x_i, the nearest neighbor in the training set $N(x_i) := \arg\min_{X_j} |X_j - x_i|$ gives a vote for its object label $L(X_j)$:

$$vote_L(x_i) = \begin{cases} 1 & L = L(N(x_i)) \\ 0 & \text{else} \end{cases} \tag{3}$$

Object classification is then based on the majority of votes:

$$\hat{L} := \arg\max_L \sum_{x_i} vote_L(x_i) \tag{4}$$

The method has a statistical foundation as is shown in Appendix C. Furthermore, it has proven to perform well in practice [Kölsch 03, Lowe 01]. This is why we follow a similar approach for object recognition in video (see Section 5).

3.3 Discussion of Global and Local Approaches

A fundamental criterion to judge object recognition approaches is their *robustness* with respect to appearance variations. The reasons and characteristics of such variations may be manifold (e.g., see Sections 2.1 and 2.2), including lighting changes, geometric transformations, and background influence.

A look at global methods reveals that they may cope well with some of these changes – e.g., color histograms are invariant to geometric transformations like scale, rotation, and shift.

On the other hand, the majority of global approaches are strongly affected by background influence in form of clutter and occluded object features, which often has an unpredictable influence on global feature vectors and makes such approaches less flexible – usually, they demand objects images free of occlusion and with minor background influence [Arbel 00, Nayar 96, Horecki 99].

One way to overcome the problem of background clutter is to *segment* the image and extract features only from its object region. For still images, a huge quantity of segmentation approaches exists ranging from low-level methods like the watershed transform or region growing to elaborate ones like Markov random fields [Geman 84] or normalized cuts [Shi 00]. Still, dividing an image into the right regions is a painful task when it comes to unknown objects like in generic object recognition. This is aggravated by the fact that segmentation is in general subjective [Martin 01].

The situation may be somewhat easier when it comes to objects in video content. If the object moves in a different way than the background, this knowledge can be exploited using *motion segmentation*. Black and Anandan [Black 96] have pointed out the difficulties in this field and presented robust methods. For an even more comprising overview of the field, see [Smolic 01].

Nevertheless, most global methods suffer from a lack of robustness. This has also been observed by Schiele and Crowley [Schiele 00], who studied a global approach based on so-called *receptive field histograms* (RFHs). Like color histograms, RFHs measure the distribution of a pixel property over an image, just that they replace color by Gaussian derivatives. First, a global method is presented that matches RFHs of images using several similarity measures for histograms. This approach is then compared to a method that views an image as a set of local features $x_1, .., x_n$ (in [Schiele 00], these were Gaussian derivatives at positions obtained from sampling over a regular grid). An object O is chosen by a Bayesian decision assuming equal priors and feature independence such that the posterior rewrites as:

$$P(O|x_1, .., x_n) = \frac{\prod_i p(x_i|O)}{\sum_k \prod_i p(x_i|O_k)} \tag{5}$$

Such an approach is considered as *local*, since an image is viewed as a set of local samples. This differentiation is somewhat artificial as is revealed by taking a closer look at the posterior. When sampling in steps of 1 (at every pixel position), we can rewrite the decision $\hat{O}$ based on Equation (5) as

$$\begin{aligned} \hat{O} &= \arg\max_{O_k} P(O_k|x_1, .., x_n) \\ &= \arg\max_{O_k} \prod_i p(x_i|O_k) \\ &= \arg\max_{O_k} \prod_x H_{O_k}(x)^{H_I(x)} \\ &= \arg\max_{O_k} \sum_x H_I(x) \log H_{O_k}(x) \end{aligned} \tag{6}$$

where x runs over all possible feature values, H_{O_k} is the model RFH corresponding to object O_k, and H_I is the RFH corresponding to the image.

What equation (6) states is that when sampling densely our "local" decision criterion just rewrites as a new similarity measure for global histograms, rendering the difference between the global and the local approach useless in this special case.

Nevertheless, we keep the differentiation in general since it is based on the way of viewing images globally or as sets of local samples. Another reason for this is that local approaches have – in contrast to global ones – proven the capability to deal with partial occlusion and clutter in a natural way: only a small fraction of features gives erroneous, unpredictable "votes", and the global decision is robust against a certain fraction of background clutter. This has been validated in experimental results by [Schiele 00], and similar observations have been made in experiments described in Section 5.2. Although the global approach showed robustness to some degree, it was not able to compete with the local probabilistic method in the presence of occlusion [Schiele 00] and background changes (Section 5.2).

4 From Still Images to Video

The object recognition methods introduced in the last two sections all deal with still images. One straightforward way of transferring the concepts to video streams is to view video as a set of multiple views showing an object at various poses. When observing multiple views of the same object in such a manner, appearance changes are a problem, especially when features disappear due to self-occlusion. This poses problems e.g.for the *constellation-based model*: usually, there is just no single unique constellation of patches that is sufficient to characterize multiple views of a 3D object. The problem turns into a 3D object recognition problem as introduced in Section 2.2. Furthermore, there comes additional information with video (besides audio tracks or closed captions) in form of the temporal relation between frames, and in form of the fact that consecutive frames should be similar assuming smooth pose changes.

Consequently, this section starts with approaches from 3D object recognition and afterwards introduces some video-specific concepts.

Inherent Treatment of Multiple Views Some classification concepts from the 2D case can be transfered naturally to multiple views, e.g.*direct voting* as introduced in Section 3.2.4: since all features of an image are stored in a database and recognition is done by Nearest Neighbor classification on patch basis, a video is simply represented by all patches from all frames.

The same expansion can be made for most global methods, where matching is done by NN classification on frame basis.

Training View-Specific Classifiers An alternative way is to train a set of view-specific classifiers for view classes of an object, as has been done in [Schneiderman 00b]. For example, all frontal views of cars are partitioned into 8 subsets (see Figure 3). Note that this approach as it is introduced is not generic – whenever a new object category shall be introduced, new view classes – so-called *aspects* – must be determined manually.

Aspect Graphs The approach introduced in the last section does not include an automatic grouping of similar views: the car views illustrated in Figure 3 have been selected manually. Automatic approaches to do this have to respect the structure of the system of views, which is an individual property of an object. More formally, the problem is to automatically generate an *aspect graph* of the given object as illustrated in Figure 8. Images of an object (in this case represented by the object silhouettes) correspond to positions on the unit sphere. Some of these images hold the same features and are very similar – they are clustered to so-called *aspects*. Edges between adjacent aspects correspond to so-called "accidental" views were the appearance of the object changes distinctly. Cluster representatives for some aspects are displayed in Figure 8 – appearance changes between the samples can be observed, e.g.the presence or absence of the kangaroo tail.

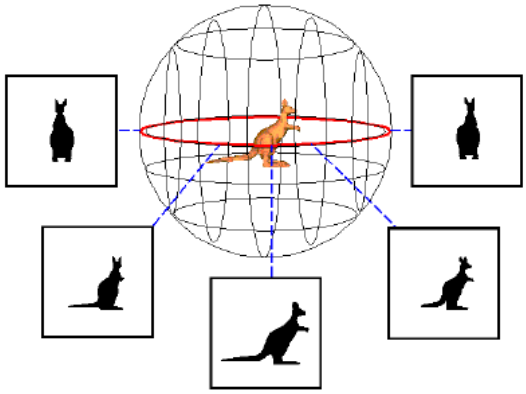


Figure 8: an aspect graph (image taken from [Cyr 04]).

A method to automatically construct an aspect graph has been presented by Cyr and Kimia [Cyr 04]. The input is a set of views $V = \{V_i\}$ of the object, which are assumed to be taken from the same distance and sample the whole hemisphere in regular steps. Due to this controlled setup, a neighbor relation over V is given for views taken from an adjacent angle.

The construction of an aspect graph is now viewed as a problem of *segmenting* V into clusters of adjacent and similar views ("aspects"). For each aspect, a representative is stored and used for NN-based recognition.

One key problem for clustering is to measure the similarity of images, for which Cyr and Kimia propose two metrics based on the object silhouette. Another one is the clustering algorithm itself – therefore, a greedy region growing scheme is proposed. Starting from singleton clusters of isolated images, similar clusters are iteratively joined. To avoid undersegmentation with large, heterogenous clusters, two clusters only combined to a new aspect $A \subset V$ if A satisfies the following constraint. Assume that r_A is the representative of aspect A:

$$\forall V_j \in A, V_k \in (V \setminus A): \quad d(r_A, V_j) < d(r_A, V_k)$$

View Clustering Like Cyr's approach [Cyr 04], Lowe [Lowe 01] clusters several views on an object. Nevertheless, the methods differ in some significant aspects: first, the approach is based on *local* SIFT features and not on global similarity measures. Second, images are not assumed to be taken in a constrained setup such that no neighborhood relationship between adjacent views is given. This makes the approach suitable for general, unconstrained video. Third, the *aspect graph* does not contain edges on aspect level. Instead, links on patch-level are introduced between similar patches from different aspects.

Thus, the object model is just a set of aspects, where an aspect is a set of

similar views of the same object, and a view is just a set of patches. Learning is equivalent to storing all patches and their links.

Both classification and learning are done using an enhanced direct voting: for each new image, patches x_i^I are extracted and matched to the database of model patches x_j^M using NN. Each such match votes for one – or via the links between similar patches for several – aspects. From these votes, a posterior $P(A|x_i^I)$ is derived for each aspect A. By thresholding $P(a|x_i^I)$ it is decided whether an object is present in the image.

Though no quantitative evaluation is presented, the approach looks promising for a test image, where it was able to detect multiple objects in a cluttered scene. Another benefit is that no constrained capture environment is needed (solely, the object is required to take on a sufficient fraction of the image space). This is why a related approach was followed in the prototype system outlined in Section 5.1.

Using the Temporal Order of Frames Javed et al.[Javed 04] exploit the temporal structure of video in a Markov framework. Given video frames $M = (m_1, .., m_T)$, each one is assigned to one aspect (determining both object and pose). This yields hypotheses in form of sequences of aspects $A = (a_1, .., a_T)$. Using a Markov property we obtain

$$\begin{aligned} P(A|M) &= \left(\prod_{t=2}^{T} P(a_t, a_{t-1}|m_t) \right) \cdot P(a_1|m_1) \\ &\propto \left(\prod_{t=2}^{T} P(m_t|a_t, a_{t-1}) P(a_t|a_{t-1}) P(a_{t-1}) \right) \cdot P(a_1|m_1) \\ &\approx \left(\prod_{t=2}^{T} P(m_t|a_t) P(a_t|a_{t-1}) P(a_{t-1}) \right) \cdot P(a_1|m_1) \end{aligned} \tag{7}$$

where the likelihood $P(m_t|a_t)$ is learned in a feature space of gradient orientations. The pose transition probability $P(a_t|a_{t-1})$ is modeled as a distribution over the 3D unit sphere (the angle corresponding to aspects is assumed as known due to a controlled setup) favoring slow transitions over abrupt ones. Unfortunately, the experimental results presented are not very convincing. Only four objects are presented without clutter or occlusion.

Note that the Markov approach outlined in equation (7) relies on knowledge from previous frames by boosting the probability of an aspect respecting the previous one. An alternative to modeling a transition probability explicitly is given by *Bayesian Chaining* [Arbel 00]. Given frames $M_t = (m_1, .., m_t)$ and object hypotheses $O_1, .., O_n$, the posterior is estimated by

$$P(O|m_t) \propto P(O) P(m_t|O)$$

While the likelihood $P(m_t|O)$ is modeled in a standard way as a Gaussian in feature space, the information from the previous frame is integrated by replacing

the *prior* $P(O)$ with the posterior from the previous frame:

$$P(O) \approx P(O|M_{t-1})$$

Video Google A completely different approach to object recognition in video has been followed by Sivic and Zissserman [Sivic 03], who focused on rapid indexing and retrieval in video databases. Therefore, parallels to the world of text retrieval are drawn (this is why the approach is called *VideoGoogle*): a text document containing words corresponds to a video frame with *visual words* in it. These visual words are just local features: they are obtained by detecting interest regions and clustering them based on their local descriptors. Furthermore, the frame structure of video is exploited in two ways:

1. features are tracked and only stable features are kept
2. local descriptors are smoothed by averaging over subsequent frames

This way, a finite *vocabulary* of patch classes is obtained, which allows to transfer text retrieval methods in a straightforward way: like a document, a video frame can be represented by a document vector (using boolean entries, the frequency, or the *inverse document frequency*), and well-known similarity measures like the cosine can be used for frames. This approach has been used to discover certain *scenes* in video streams.

A second scenario is to find frames containing a certain object, which is a more difficult problem due to clutter. Therefore, features between the object and each frame are matched, and two additional constraints are imposed:

1. *Spatial Consistency*: instead of deriving an affine transform mapping features from the object into the frame, a simple and fast voting scheme is used: the 15 nearest neighbors in the object are determined and give a boosting extra vote if they are neighbors in the frame, too.
2. *Stop Words*: in text documents, very frequent words are not discriminative and are thus discarded in a preprocessing step. The same phenomenon can be observed for *visual* words, which are discarded for object retrieval, too.

5 A Practical Approach to Object Recognition in Video

The object recognition survey given in the last two sections made a differentiation between *global* and *local* approaches. It is an interesting question to compare the performance of these classes in a practical environment. Therefore, we built a system called VideoObjects for the learning and recognition of objects in video streams. Using this infrastructure, we evaluated the performance of one global and one local approach in quantitative experiments. To our knowledge, the only previous work presenting such a direct comparison is [Schiele 00].

Also, our goal was to build some initial infrastructure for following in-depth research on object recognition. In the following, we first introduce the VideoObjects prototype in detail in Section 5.1. Afterwards, our experiments are described and results are discussed (Section 5.2).

5.1 The VideoObjects System

The object recognition system built is called VideoObjects. Its purpose is to learn objects that are moved manually in front of a fixed video camera, and later on recognize them when presented again. The only user interaction required is to present the object to the camera, and – during learning – entering the name of the presented object.

The system setup is described in this Section, including a detailed outline of the single components and the underlying pattern recognition techniques.

The system setup is illustrated in Figure 9. A firewire webcam[7] with a 320×240 resolution observes a scene, which is assumed static except for an object presented to the camera. The resulting motion is detected, and a video of the moving object is stored. Furthermore, the VideoObjects thread is triggered, which segments the object from the background using *motion segmentation* and extracts features from the object area in each video frame.

To learn the appearance of objects and afterwards use this information to recognize unknown items, we use a semi-supervised approach: if a new object is learned (blue path in Figure 9), the only manual labeling of data required is that the user types in the object name. The name and the object features extracted from the video are stored in the *Object Base*, where all system knowledge is represented. If an unknown object is recognized (red path in Figure 9), features are extracted from the video and classification is done by matching with the object base. In the following, the single components of the system are depicted in more detail.

[7] UniBrain Fire-I webcam

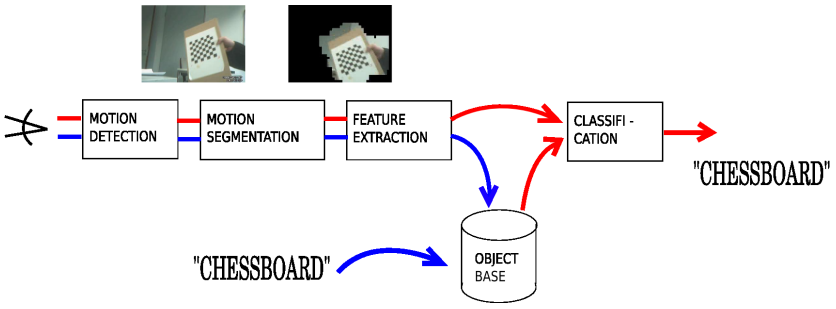


Figure 9: the setup of the VIDEOOBJECTS system: if an object is moved in front of the camera, it segmented from the background and features are extracted. When learning an object (blue), the features are stored in the object base together with the object name typed in by the user. If an unknown object is recognized (red), the extracted features are matched with the object base.

5.1.1 Motion Detection

To detect the presence of an object and trigger object recognition, the Linux software tool MOTION[8] is used. The tool is based on a rather simple technique: the difference between a new image and the *reference frame* (a weighted sample of previous frames) is computed and thresholded to decide whether motion is present. This method proved absolutely sufficient for our purposes.

5.1.2 Motion Segmentation

It has been outlined in previous sections that *clutter* poses a problem for object recognition and can have a disturbing influence on classification. Especially if the object covers a relatively small potion of the image, object recognition can benefit from discarding background features.

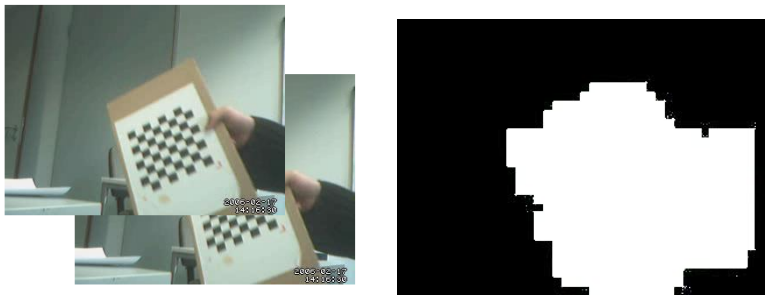
Therefore, we extract a binary object mask from each frame where `ON` pixels belong to the object and `OFF` pixels to the background. An example is given in Figure 10(c). Extracting this mask for an unknown object in front of arbitrary background is intractable as long as we view the frames as isolated still images.

However, a simple way to achieve segmentation is to make use of the temporal structure of video. We just exploit the fact that the object pixels *move* in a different way than the background pixels (which we assume static). This approach is called *motion segmentation*. For an illustration of the general concept, see Figure 10.

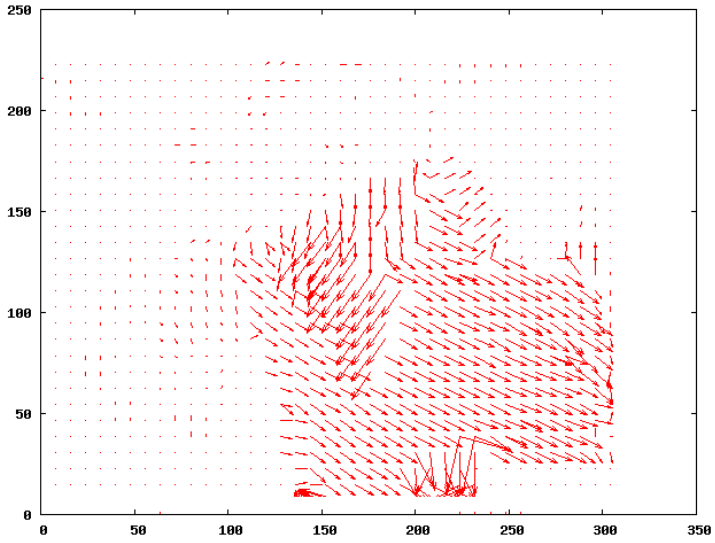
Our motion segmentation procedure consists of three steps: first, motion is estimated. Afterwards, the resulting motion field is segmented yielding the object mask, which is finally refined in a post-processing step:

1. MOTION ESTIMATION: computing motion between adjacent frames usu-

[8] http://www.lavrsen.dk/twiki/bin/view/Motion/WebHome



(a) (c)



(b)

Figure 10: the concept of motion segmentation: for each pair of consecutive frames (10(a)), blocks of pixels are tracked using block matching. This gives a sparse *field* of motion vectors (10(b)), which is afterwards use to segment the object from the background, obtaining an *object mask* (10(c)).

ally means to *track* pixels, which is a time-consuming task including a region search for each vector in the motion field.

We follow a *sparse* motion estimation using "Enhanced Predictive Zonal Search" [Tourapis 02] (EPZS). The technique is based on *block matching*, which is a popular method in the video domain: the reference frame is divided into rectangular blocks, and for each block the best fit in the new frame is found by minimizing the sum of squared difference (SSD) between blocks using a discrete Gradient Descent Technique. This gives a field of block motion vectors. In a second sweep, these motion vectors are *smoothed*: again, gradient descent is used, but this time the initial shift is chosen depending on the neighbor blocks' motion.

The result is a sparse *motion field* as can be observed in Figure 10(b). In the two reference frames, the background remains static (the motion vectors are near 0), while for the blocks belonging to the object non-zero motion vectors were found.

The use of motion segmentation has a practical consequence for the usability of the system: only as long as the object moves, it is detected and reasonable features are extracted. In contrast, frames in which the object is held still may yield erroneous features due to failures of motion segmentation.

2. CLASSIFICATION: to construct an object mask, each block B is assigned to the object (or background, respectively) depending on its motion vector $v(B)$. Afterwards, the mask is constructed by setting all pixels of a block to `ON` (`OFF`, respectively). Our approach for this is to threshold with the length of the motion vector v:

$$B = \texttt{ON} \quad \leftrightarrow \quad |v(B)|^2 > T$$

This can be motivated using statistical motion *models* for background and object: the background is assumed static except for isotropic noise. Consequently, the distribution of background vectors has a strong peak at 0 and is isotropic. We model this moving a two-dimensional Gaussian $\mathcal{N}_{0,\sigma^2}$.

Unfortunately, we do not know that much about the object motion. Thus, a uniform distribution over a sufficiently large range R is assumed.

Imposing uniform priors, the Bayesian classification decision rewrites as:

$$\begin{aligned} B = \texttt{ON} \quad &\leftrightarrow \quad \mathcal{N}_{0,\sigma^2}(v(B)) < \frac{1}{|R|} \\ &\leftrightarrow \quad |v(B)|^2 > \underbrace{-2\sigma^2 \ln \frac{\sqrt{4\pi}\sigma^2}{|R|}}_{=:T} \end{aligned} \tag{8}$$

In practice, the threshold T is chosen empirically based on the framerate, the camera resolution, the expected distance of the object from the camera, and its expected velocity.

3. POSTPROCESSING: Our motion estimation procedure is error-prone due to several reasons, including effects of varying illumination, EPZS gradient descent being caught in local minima, or inadequate motion models. In practice, the object mask usually contains some outlier blocks. This is why we refine it using a morphological dilatation and a connected component analysis rejecting all components except for the largest one.

Figure 11: a typical motion segmentation result. The foreground does not only contain the object (as would be perfect for recognition) but also the operator's arm – which is correct, because it moves – and parts the background resulting from motion segmentation errors.

A typical motion segmentation result is visualized in Figure 11. The image is obtained from matting the input frame with the object mask obtained from motion segmentation. It can be seen that the foreground also includes the operator's arm due to its motion, as well as some parts of the background. This poses problems for the following object recognition.

Also, the procedure gives rather coarse object masks on block basis. Since it might be interesting to obtain a pixel grain object masks, we also did some tests for a publicly available implementation of a pixel-based approach [Black 96]. However, EPZS was found to perform better and significantly faster due to its coarseness and fast assembler routine from the video compression codec XviD[9] used to compute the block SSD. To improve the object mask further, we think of another postprocessing step refining the block results on pixel level.

5.1.3 Feature Extraction

Feature vectors are extracted from the object regions in the segmented video. Our approach does not take into account temporal relationships so far – features are extracted separately for each frame. Furthermore, feature extraction and

[9] www.xvid.org

classification are kept independent. This makes our approach suitable for any type of feature described in Section 3.

Since one fundamental purpose of the VIDEOOBJECTS prototype is to compare the performance of patch-based object recognition and global methods, two kinds of features were implemented so far. Both are extracted only from object region that is determined by the object mask.

1. GLOBAL FEATURES - COLOR HISTOGRAMS: As a representative for global approaches and as a baseline method, *color histograms* were used (see Section 3.1). For each frame, the color histogram of the object region is computed after motion segmentation. The RGB model is used with 5 bins per axis, yielding a 125-dimensional feature vector for each video frame.

2. LOCAL FEATURES - PATCHES: For the patch-based approach, interest points in the object region are extracted using the *salient points* detector [Loupias 99] (see 3.2.1). For each keypoint, a local patch of 16×16 pixels is extracted and transferred to the YUV color space. The Discrete Cosine Transform (DCT) (see Section 3.2.2) is applied, and the resulting low-frequency coefficients are used as a local descriptor (35 ones for the intensity Y, and 20 for both chroma components U and V). The lowest component – the average value – is left out for illumination invariance purposes. A set of up to 500 feature vectors is obtained per frame.

Tests on the recognition performance and robustness of both approaches will be described described in Section 5.2.

5.1.4 Classification

Feature extraction yields a set of d-dimensional features $\{x_1, .., x_n\}$ for each video with n and d depending on the specific feature extraction method. To recognize an unknown object, these features have to be matched with the object base $\{Y_1, .., Y_N\}$ with entries $Y_j = (X_j, L(X_j))$ consisting of features X_j and object labels $L(X_j)$.

For the classification of a new video, *direct voting* is followed as described in Section 3.2.4. For each feature x_i, the nearest neighbor X_j in the object base is picked and gives a vote for the object $L(X_j)$. The object with the majority of votes is chosen.

IMPLEMENTATION DETAILS: direct voting is a time-consuming task for large object bases (e.g., in some of the experiments conducted, up to $10,000$ patch features are extracted from each video). The heart of the method is a Nearest Neighbor query in feature space, which can be done efficiently using space-partitioning data structures. For the VIDEOOBJECTS system, a free kd-tree implementation[10] for fast NN queries has been chosen [Paredes 01].

As a distance measure, the Euclidean distance in feature space was used for both histograms and local descriptors.

[10] available from Javier Cano, ITI, University of Valencia

(a) (b)

(c) (d)

Figure 12: sample images for the `galaxy` (12(a), 12(c)) and the `flips` object (12(b), 12(d)). 12(a) and 12(b) were taken in the `OFFICE`, 12(c) and 12(d) in the `LAB`.

5.2 Experiments

We conducted some quantitative experiments with the VideoObjects prototype as outlined in the last section. This was done following two goals: first, to validate the general performance of the VideoObject setup in practice. Second, to compare the feature types *color histograms* and *patches*: the first one serves as a baseline method making use of global appearance. On the other hand, patch-based object recognition is currently an active area of research and has proven a high robustness against noise and clutter. We want to study the performance and robustness of the two approaches in video object recognition.

Our results show that both methods perform well for constant capturing conditions, while the local patch features show a higher robustness when it comes to generalization to new scenes. The following sections describe the system setup, the data used, and experimental results.

5.2.1 Setup

The system setup is the same as presented in Section 5.1 and illustrated in Figure 9. In front of a static background, objects are presented manually to a firewire webcam, which takes videos of 320×240 pixels at a frame-rate of $25/s$. Each video contains about 50 frames.

The objects are presented to the camera by hand. Though this leads to slight variations in object pose, a "front side" is chosen for each object.

5.2.2 Datasets

In the following experiments, the performance of the VideoObjects system is evaluated in a natural, unconstrained environment. Especially, we want to study the effects of scene changes including illumination and clutter. This is why we avoid standard databases like COIL[11], where object images are usually given in high quality and with hardly any background influence (for an example, see Figure 1). To the authors knowledge, no standard video dataset for our specific setup exists – this is why a self-made dataset was preferred over artificial standard databases.

We took videos of 16 everyday objects. The resulting frames show typical problems when working in an unconstrained environment like motion blur, illumination variations, and clutter. The set of objects includes very simple ones like a chessboard posing very discriminative features, as well as a water bottle showing specular highlights and transparency, and also quite similar objects like two red shirts.

Videos were taken at two different locations on different days:

1. `OFFICE`: in this scene, videos were taken with light from the right. The background is a weakly textured white wall showing only few strong edges.

2. `LAB`: this scene is characterized by strong daylight. The objects in the resulting videos are generally brighter, with some of them showing specular highlights. The background is strongly textured.

Examples for both cases are illustrated in Figure 12. The most obvious difference is the background change, but some more effects can be observed: objects in the `office` images 12(a) and 12(b) are a bit darker, while specular highlights can be observed in the `LAB` (see Figures 12(c) and 12(d)). Furthermore, motion blur (12(c)) can be observed as well as rotation and scale change between 12(b) and 12(d).

For each object, 6 videos were taken at each location and divided into 3 training videos for direct voting and 3 for testing. Each video shows only one object. This gave sets `OFFICE_TRAIN`, `OFFICE_TEST`, `LAB_TRAIN`, `LAB_TEST` of 48 videos each. From these videos, we extracted both types of features (obtaining one feature vector per frame for color histograms and up to 500 per frame for patches) and tested them using direct voting.

[11] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

location	color histograms	patches
OFFICE	0	0
LAB	2	0

Table 1: numbers of errors for Experiment 1 (testing at the same location as training). Each test set consisted of 48 videos.

5.2.3 Experiment 1: Same Scene

In a first experiment the general performance of the approach is validated for the simpler case of recognizing an object in the same scene in which it was learned. For example, OFFICE_TRAIN is used for training and OFFICE_TEST for recognition. The error rates are presented in Table 5.2.3. They represent *global* performance, namely the percentage of test videos that have been classified incorrectly.

The low error rates validate the general performance of both methods for this problem. Though the histogram method confused two objects of similar color distribution in the LAB scene, it offers a fast and simple alternative: when using motion segmentation with EPZS, only 1.7 frames per second could be processed for patches (where interest points have to be extracted and DCT descriptors computed), while color histograms allowed for a frame-rate of 7.6.

5.2.4 Experiment 2: Different Scene

training location	test location	color histograms	patches
LAB	OFFICE	39	12
OFFICE	LAB	29	14

Table 2: numbers of errors for Experiment 2 (scene changes between training and testing). Each test set consisted of 48 videos.

In a second experiment, we want to study the influence of scene changes on recognition performance. Especially, we want to evaluate the robustness of both methods against the influence of clutter. Therefore, we train the VideoObjects system in one location and test the recognition result in the other one. Table 5.2.4 illustrates the numbers of errors for this setup.

The results show strongly increased error rates of at least 25 %. Also, a clear difference between both methods can be observed, since histograms seem to be much more sensitive to the scene change.

To explain these results, we take a closer look at motion segmentation. As has been illustrated in Figure 11, motion segmentation does usually not yield a

perfect object mask, but the matted frames often contain spurious background parts. Especially for frames with the object held still, it is obvious that motion is not a sufficient criterion to discriminate between object and background.

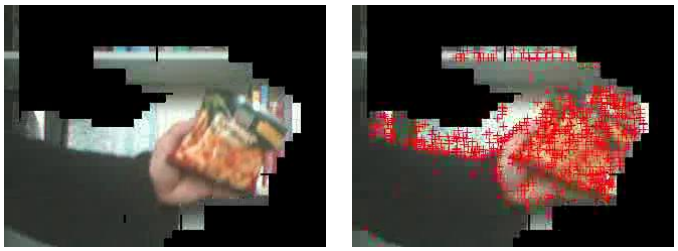
Figure 13 illustrates the effects of an error-prone motion segmentation on recognition. An example for a frame in the `LAB` is given in Figure 13(a), with spurious parts of the background assigned to the object region. Figure 13(b) shows keypoints extracted from this matted frame (red). It can be seen that some keypoints have been extracted from the arm of the operator, as well as from the textured background.

Obviously, object recognition suffers from these suboptimal segmentation results: histograms show erroneous peaks at background colors, and background patches give noisy votes due to background texture.

Obviously, this becomes a serious problem when the scene changes and a different background texture influences the recognition process. Furthermore, there are other difficulties that may cause problems:

- illumination changes, as is illustrated in Figure 12
- differences in presentation: e.g., in the `LAB` objects are presented from the left while from the right in the `OFFICE`. Furthermore, the distance of the object from the camera may vary.

As the results presented in Table 5.2.4 indicate, histograms seem to be much more sensitive to these influences. One possible explanation is that for global methods such as histograms, the whole feature changes in a way that is hard to predict. On the other hand, the patch-based approach still reliably produces a fraction of true object votes besides noisy votes for background patches.



(a) (b)

Figure 13: a typical result of motion segmentation and its effects: some spurious parts of the background are classified as object parts. It can be seen from Figure 13(b) that some patches (red) are extracted from these background areas.

A closer look at the classification results for the single objects shows some obvious error reasons like the transparency of the `bottle` object, which was misclassified in the new environment just because its appearance changes strongly.

Another interesting fact is that the correctness of classification is correlated with the number of patches: for each object, we count the patches that have been derived in training. Obviously, this number is strongly related to the number of patches *on* the object: for large objects with many patches, this number is high (e.g., the `chessboard` gives the most features with 40.791). On the other hand, we measure the correctness of classification by the percentage of patches that vote for the *correct* object in the test videos, ranging from 6% to more than 40%. It could be observed that slightly more than 10% of correct votes were sufficient for a correct classification.

Between these two features, a low but present correlation of 0.572 is found. Thus, the correctness of classification increases with the number of object features. An explanation for this is that more object features overlay the "noisy" votes from background patches.

5.2.5 Further Improvements

The experimental results show the general applicability of the current VideoObjects prototype, particularly when remaining in the same scene for recognition. Starting from this infrastructure, we plan to participate the TRECVID video retrieval evaluation in 2006[12].

However, a variety of improvements for the current setup can be thought of. A long-term goal is the enhancement for *object category* detection. In the video domain, this problem has only been addressed by very few approaches so far [Schneiderman 00b]. Other 3D object recognition methods like [Lowe 01] cannot generalize to new class instances, while most object category recognizers like [Fergus 03] are not suitable for multiple perspectives or rotations.

Another problem is that the object is assumed to be in a constant distance from the camera so far. To allow for more flexibility, the sensitivity to depth changes should be studied, and scale invariant feature points (see Section 3.2.1) should be used. Though salient points generally come with a scale, this information has not been used yet.

Another topic related to the flexibility of the system is *3D object recognition*. In the experiments presented in Section 5.2, a front side has been chosen for each object and only slight pose changes take place. Specific aspects of *3D object recognition* like view clustering have not been addressed explicitly in these experiments. Though we expect the direct voting strategy of the VideoObjects setup to cope with multiple perspectives well, it might be interesting to study the problem in future work.

Generally, other feature extraction methods like corners could be tested, because they allow for a more stable tracking of features. Other topics are the visualization of patch votes, and some work on enhancing background segmentation and examining its influence.

[12] http://www-nlpir.nist.gov/projects/trecvid/

6 Challenges

The vast amount of work on object recognition indicates its potential as well as its difficulty. The main reasons for the latter are appearance variations. These may concern the object itself, like deformations of non-rigid objects or differences between instances of an object category. Changes of the surrounding scene including background clutter, illumination, pose, and occlusion pose fundamental problems as well.

Nevertheless, object recognition has made fundamental progress during the last years concerning its robustness and flexibility. For still images, special emphasis has recently been put on local, patch-based methods, which proved successful for object category recognition as well as 3D object recognition. However, there is no approach to the knowledge of the author that poses a generic solution for both problems at the same time.

For video, the problem of object recognition is viewed in the context of *semantic modeling* of content [Bashir 03], which includes the recognition of objects as well as other concepts like events and sites. While lots of publications can be found on rather technical low-level tasks like shot boundaries, keyframe extraction, and similarity matching of frames, only little work has been done on using actual object queries (e.g., [Chang 98]). However, the potential has been recognized and respected – e.g., video compression standards as MPEG-4[13] envision to compress a video in layers of objects.

Work on semantic modeling of video content that allows query-by-keyword is rare. One framework has been presented by Naphade and Huang [Naphade 00], where so-called *multijects* representing objects, sites, and events in video are learned in a semi-automatic manner. On a higher level, a factor-graph framework is used to represent interdependencies between multijects, such as the presence of "beach" boosting the presence of "water".

One major challenge of the field has not been addressed yet in this survey, namely the problem of *training data* acquisition. Obviously, the amount of human interaction must be kept reasonably low if semantic indexing of multimedia documents shall be applied widely in practice. On the other hand, all approaches introduced in this survey use machine learning techniques and thus extract the representation of an object from a *training set* usually consisting of hundreds or thousands of images (or video shots, respectively).

The acquisition of such training sets should be done with the least amount of user interaction possible, but should also provide the necessary information to build discriminative object models. Three levels of user interaction can be identified:

1. SUPERVISED: Some approaches in 3D object recognition like [Javed 04, Nayar 96] learn the appearance of an object in a completely controlled setup, where images are taken in a scene free of occlusions and sometimes from predefined perspectives. Also, the background is monotonous.

[13] http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm

This corresponds to a *segmentation* of the image and makes it possible to completely ignore the background. In this case, we speak of a supervised setup.

2. SEMI-SUPERVISED: Here, a training set of unsegmented images is provided. Such data is called *weakly labeled*.

   Some state-of-the-art approaches modeling objects and object categories in such a weakly supervised manner have been introduced in this survey [Fergus 03, Deselaers 05, Weber 00, Keysers 06], and standard databases for object category detection exist[14,15].

   However, even the amount of user interaction to build such datasets can be unacceptable for large-scale generic object recognition. For the video domain, the required annotations can be provided to some extent by closed captions and speech extraction. Still images, however, do usually not come with such information. In these cases, other ideas inspired by *semi-supervised learning* [Zhu 06] have been presented to reduce the amount of labeled data needed.

   Rosenberg et al.[Rosenberg 05] trained the appearance of human eyes from few (in the order of 40) labeled training samples L with eyes landmarked and a large set of unlabeled images S. A bootstrapping approach was followed by alternately training a classifier from L and using the classification output to label the most confident samples and shift them from S to L.

   In the video domain, Yan and Naphade [Yan 05] presented another approach called *semi-supervised cross-feature learning*. The method is related to *co-training* [Zhu 06], where the features of each sample are divided into two different sets called *views*, and classifiers are trained on each set separately. Iteratively, the most confident samples of one classifier are added to the training set of the other classifier. This requires *view sufficiency*: the features in each view must be sufficient to train a "good" classifier. Yan and Naphade improve conventional co-training by linearly combining the classifiers they obtain from the iterations of their algorithm.

   Another approach has been presented by Fei-Fei et al.[Fei-Fei 03], who tried to learn object models from very few sample images only. The method generally adapts the constellation-based Model (see Section 3.2). However, the parameters θ describing the location and appearance of the object class features are handled in a different way. While previous approaches [Fergus 03] learn a fixed instance of model parameters θ in a maximum likelihood manner, Fei-Fei et al.estimate a *distribution* for θ following Bayesian parameter estimation [Duda 00]. Object categories are

[14] http://www.pascal-network.org/challenges/VOC/databases.html
[15] http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

learned incrementally: if a new object O is to be trained, priors are imposed on the expected distribution of θ_O that represent *where* and *which* important features have been found for previous objects and should thus be used for O. According to the authors, this makes it possible to learn object categories from very few samples.

3. UNSUPERVISED: Unsupervised learning methods use unlabeled training sets such that no user information is required at all. Experiments in this area have been inspired by *probabilistic Latent Semantic Analysis* (pLSA), a method from the text processing domain [Hofmann 99, Hofmann 01]. In conventional pLSA, the input consists of a set of documents D each containing words of a vocabulary W with a certain frequency. The correlations between word occurrences is used to detect latent *topics* T in documents.

   Given a word $w \in W$, a document $d \in D$, and topics $t \in T$ the following model is used to determine the probability of a topic occurring in a document $P(t|d)$.

$$\begin{aligned} P(w|d) &= \sum_{t \in T} P(w,t|d) \\ &\approx \sum_{t \in T} P(w|t) \cdot P(t|d) \end{aligned} \tag{9}$$

   Sivic et al.[Sivic 05] transfered this model to object recognition such that documents turn into images, words into "visual words" (discretized local features), and topics into the objects to be recognized. In experiments, they used training set of up to 9 object categories.

   Fergus et al.[Fergus 05a] extended pLSA to take the spatial constellation of features into account. They also pictured a new training scenario in which the training set is obtained from a conventional large-scale image search engine (in their case, Google Image Search). For example, if a model for "airplane" shall be learned, the training set is obtained from typing "airplane" in Google Image Search.

   An impression of the result is given in Figure 14. Of course, training with such an image set poses additional problems due to erroneous images returned by the search engine. To identify a subset of images actually showing airplanes, the pLSA model (9) is used fitting "background" topics and an "airplane" topic.

   This leads to a new, completely unsupervised learning of object models, with large-scale search engines delivering a candidate set that is afterwards refined using content-based methods.

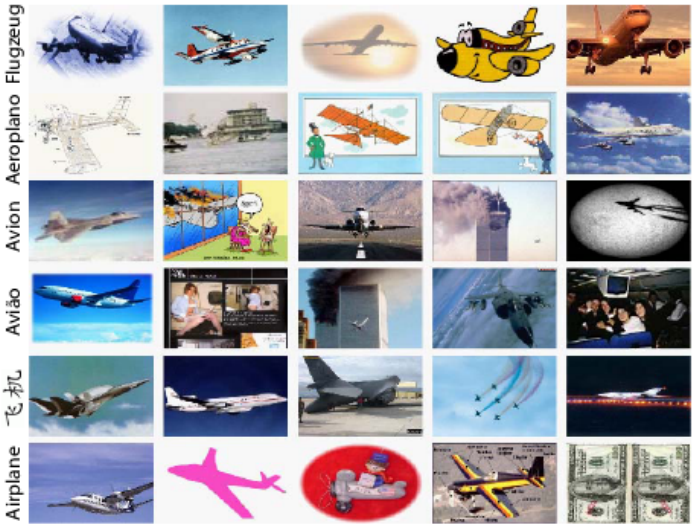


Figure 14: some "airplane" images returned by Google Image Search (image taken from [Fergus 05a]).

A Expectation Maximization

Expectation Maximization estimates a set of distribution parameters θ that maximize the likelihood for a set of observed data vectors $X = \{X_1, .., X_N\}$ (e.g., for fitting a Gaussian mixture model (GMM)). This is often aggravated by the fact that a certain fraction of the data is missing: each X_i consists of known – or *observed* – coordinates X_i^o and missing ones X_i^m.

The optimal way of determining θ would be to maximize the log-likelihood while marginalizing over the missing data:

$$L(X|\theta) = \sum_i \log \int_{X_i^m} p(X_i|\theta) \quad dX_i^m$$

Unfortunately, there is usually no closed-form solution for $\arg\max_\theta L(X|\theta)$.

This motivates the use of iterative *Expectation Maximization* schemes. Starting with an initial θ_0, the parameters are iteratively refined by maximizing

$$\begin{aligned} \mathcal{Q}(\tilde{\theta}|\theta) &= \sum_i E[\log p(X|\tilde{\theta})] \\ &= \sum_i \int_{X_i^m} p(X_i^m|\theta) \cdot \log p(X|\tilde{\theta}) \quad dX_i^m. \end{aligned}$$

While $\theta = \theta_k$ is fixed, $\mathcal{Q}$ is maximized with respect to $\tilde{\theta}$ obtaining a solution

$$\theta_{k+1} := \arg\max_{\tilde{\theta}} Q(\tilde{\theta}|\theta),$$

An iteration is made up of two steps:

1. *Estimation*: to later on estimate $\mathcal{Q}$, the posterior probabilities $p(X_i^m|\theta)$ are determined
2. *Maximization*: $\mathcal{Q}$ is maximized with respect to $\tilde{\theta}$

The convergence in a local maximum is guaranteed [Dempster 77].

B Salient Points Feature Detection - Details

While the features given by many other feature detectors tend to focus in regions of strong contrast, Loupias and Sebe [Loupias 99] present a wavelet-based method that delivers features that are more regularly distributed.

The basic idea is to search the coefficients in the wavelet transform $\mathcal{W}f$ of the input image f for peaks called "salient points". These coefficients $\mathcal{W}_{2^{-k}}f(x)$ are extracted at different scales by convolving the signal (or image, respectively) with a wavelet functions of a certain frequency 2^{-k}. A strong coefficient thus indicates the presence of an edge or corner at a given scale.

For each $\mathcal{W}_{2^{-k}}f(x)$, a *support region* $S(\mathcal{W}_{2^{-k}}f(x))$ in the original image is defined as the pixels that contribute to the coefficient. In contrast to the Fourier Transform, this support is finite for the Wavelet transform. The higher the frequency, the smaller this region. For example, for $k = 0$ it consists of only one pixel, while low-frequency coefficients are computed over large image regions. Furthermore, support regions for coefficients at different scales overlap due to the recursive nature of the wavelet transform.

This leads to the definition of *children* for each wavelet coefficient as coefficients at the next scale step with the same support:

$$C(\mathcal{W}_{2^{-k}}f(x)) = \{ \quad \mathcal{W}_{2^{-k+1}}f(x') \quad | \quad S(\mathcal{W}_{2^{-k+1}}f(x')) \subset S(\mathcal{W}_{2^{-k}}f(x)) \quad \}$$

The extraction of salient points uses this definition: it scans through *all* coefficients in $\mathcal{W}f$ and chooses the child with the highest coefficient value. From its support region, the image pixel with the strongest gradient is picked as a *salient point*. Afterwards, the resulting points are filtered by thresholding with a saliency measure.

For a further discussion of the method, see 3.2.1.

C Direct Voting – Statistical Motivation

Direct Voting as a decision strategy based on local patches has been introduced in Section 3.2.4. An image is viewed as a set of local features $x_1, .., x_n$. From

a set of training images, patches $X_1, .., X_N$ associated with object labels $L(X_j)$ are extracted. The decision strategy is to collect votes by Nearest Neighbor on patch level and classify based on the majority of votes. The spatial constellation of patches is neglected.

A statistical motivation for direct voting consists of two steps. First, it is shown that $vote_L(x)$ is an approximation of the posterior $P(L|x)$. This is true for equal class priors and if $p(x|L)$ is modeled by a kernel density estimate with a Gaussian kernel $\mathcal{N}$. Let

$$p_\alpha(x|L) = \frac{1}{|C_L|} \sum_{X_j \in C_L} \mathcal{N}_{X_j,\alpha^2}(x),$$

where C_L is the set of samples X_j with label L. Then for the corresponding posterior

$$p_\alpha(L|x) = \frac{p_\alpha(x|L)}{\sum_l p_\alpha(x|l)}$$

the following convergence can be shown for $\alpha \to 0$ [Kölsch 03]:

$$\begin{aligned} \lim_{\alpha \to 0} p_\alpha(L|x) &= \frac{\frac{1}{|C_L|}\delta(L(N(x)), L)}{\frac{1}{|C_{L(N(x))}|}} \\ &= \delta(L(N(x)), L) \\ &= vote_L(x) \end{aligned} \tag{10}$$

The intuitive interpretation is that the more local the influence of samples is – the smaller α – the less influence examples distant from x have on $p_\alpha(L|x)$. For the border case $\alpha \to 0$, this reduces to zero influence for all but the nearest neighbor $N(x)$. A detailed proof has been presented in [Kölsch 03].

The second question is why the *sum* of votes is a good choice for the combination of features (or how it is related to the Bayesian decision maximizing $P(L|x_1, .., x_n)$).

One motivation can be obtained by marginalizing over the patches. Let $\mathcal{X}$ be the input image with patches $\{x_i\}$ extracted:

$$\begin{aligned} P(L|\mathcal{X}) &= \sum_{x_i} P(L, x_i|\mathcal{X}) \\ &= \sum_{x_i} P(L|x_i, \mathcal{X}) \cdot P(x_i|\mathcal{X}) \\ &\approx \frac{1}{n} \sum_{x_i} P(L|x_i) \end{aligned}$$

References

[Agarwal 04] S. Agarwal, A. Awan & D. Roth. *Learning to Detect Objects in Images via a Sparse, Part-Based Representation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pages 1475–1490, November 2004.

[Arbel 00] T. Arbel, F.P. Ferrie & M. Mitran. *Recognizing Objects From Curvilinear Motion.* In Proceedings of the British Machine Vision Conference, pages 765–774, Bristol, UK, 2000.

[Bashir 03] F. Bashir & A. Khokhar. *Video Content Modeling: An Overview.* Technical Report, Dept. of CS/ECE, UIC, 2003.

[Berg 04] A. Berg, T.L. Berg & J. Malik. *Shape Matching and Object Recognition using Low Distortion Correspondences.* Technical Report, EECS Department, University of California, Berkeley, 2004.

[Black 96] M.J. Black & P. Anandan. *The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields.* Computer Vision and Image Understanding, vol. 63, no. 1, pages 75–104, 1996.

[Breuel 92] T.M. Breuel. *Fast Recognition using Adaptive Subdivisions of Rransformation Space.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 445–451, 1992.

[Burl 96] M.C. Burl & P. Perona. *Recognition of Planar Object Classes.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 223–230, Washington, DC, USA, 1996.

[Burl 98] M.C. Burl, M. Weber & P. Perona. *A Probabilistic Approach to Object Recognition using Local Photometry and Global Geometry.* Lecture Notes in Computer Science, vol. 1407, pages 628–641, 1998.

[Canny 87] J. Canny. *A Computational Approach to Edge Detection.* In Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, pages 184–203. Kaufmann, Los Altos, CA., 1987.

[Chang 98] S. Chang, W. Chen, H.J. Meng, H. Sundaram & D. Zhong. *A Fully Automated Content Based Video Search Engine Supporting Spatio-Temporal Queries.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 5, pages 602–615, 1998.

[Chen 04] X. Chen & A.L. Yuille. *Detecting and Reading Text in Natural Scenes.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 366–373, 2004.

[Csurka 04] G. Csurka, C. Bray, C. Dance & L. Fan. *Visual Categorization with Bags of Keypoints.* In Proceedings of the 8th European Conference on Computer Vision, Prague, May 2004.

[Cyr 04] C.M. Cyr & B.B. Kimia. *A Similarity-Based Aspect-Graph Approach to 3D Object Recognition.* International Journal of Computer Vision, vol. 57, no. 1, pages 5–22, 2004.

[Dempster 77] A.P. Dempster, N.M. Laird & D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM algorithm.* Journal of the Royal Statistical Society B, no. 39, pages 1–38, 1977.

[Deselaers 05] T. Deselaers, D. Keysers & H. Ney. *Discriminative Training for Object Recognition Using Image Patches.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 157–162, Washington, DC, USA, 2005.

[Duda 00] Richard O. Duda, Peter E. Hart & David G. Stork. Pattern Classification (2nd Edition). Wiley-Interscience, 2000.

[Faloutsos 94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic & W. Equitz. *Efficient and Effective Querying by Image Content.* Journal of Intelligent Information Systems, vol. 3, no. 3/4, pages 231–262, 1994.

[Fei-Fei 03] L. Fei-Fei, R. Fergus & P. Perona. *A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories.* In Ninth IEEE International Conference on Computer Vision, volume 2, pages 1134–1141, 2003.

[Fergus 03] R. Fergus, P. Perona & A. Zisserman. *Object Class Recognition by Unsupervised Scale-Invariant Learning.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 264–271, 2003.

[Fergus 05a] R. Fergus, L. Fei-Fei, P. Perona & A. Zisserman. *Learning Object Categories from Google's Image Search.* In Proceedings of the 10th International Conference on Computer Vision, pages 1816–1823, Beijing, China, 2005.

[Fergus 05b] R. Fergus, P. Perona & A. Zisserman. *A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 380–387, 2005.

[Freund 96] Y. Freund & R.E. Schapire. *Experiments with a New Boosting Algorithm.* In Proceedings of the International Conference on Machine Learning, pages 148–156, 1996.

[Geman 84] S. Geman & D. Geman. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 6, pages 721–741, Nov. 1984.

[Gilles 98] S. Gilles. *Robust Description and Matching of Images.* PhD thesis, University of Oxford, 1998.

[Harris 88] C. Harris & M. Stephens. *A Combined Corner and Edge Detector.* In Proceedings of the 4th ALVEY vision conference, pages 147–151, September 1988.

[Hofmann 99] T. Hofmann. *Probabilistic Latent Semantic Analysis.* In Proceedings of Uncertainty in Artificial Intelligence, pages 289–296, Stockholm, Sweden, 1999.

[Hofmann 01] T. Hofmann. *Unsupervised Learning by Probabilistic Latent Semantic Analysis.* Machine Learning, vol. 42, no. 1-2, pages 177–196, 2001.

[Horecki 99] K. Horecki, D. Paulus & K. Wojciechowski. *Object Localization Using Color Histograms* . In 5. Workshop Farbbildverarbeitung , Schriftenreihe des Zentrums für Bild- und Signalverarbeitung e.V. Ilmenau , pages 59–66, 1999.

[Huber 82] P. Huber. Robust Statistics. John Wiley, 1982.

[Javed 04] O. Javed, M. Shah & D. Comaniciu. *A Probabilistic Framework for Object Recognition in Video.* In Proceedings of the International Conference on Image Processing, pages 2713–2716, 2004.

[Jesorsky 01] O. Jesorsky, K.J. Kirchberg & R.W. Frischholz. *Robust Face Detection Using the Hausdorff Distance.* In Audio- and Video-Based Person Authentication - AVBPA 2001, volume 2091 of *Lecture Notes in Computer Science*, pages 90–95, Halmstad, Sweden, 2001.

[Kadir 01] T. Kadir & M. Brady. *Saliency, Scale and Image Description.* International Journal of Computer Vision, vol. 45, no. 2, pages 83–105, 2001.

[Keysers 06] D. Keysers, T. Deselaers & T.M. Breuel. *Optimal Geometric Matching for Patch-Based Object Detection.* In International Conference on Pattern Recognition, Hong Kong, China, August 2006. Submitted for review.

[Kölsch 03] T. Kölsch. Local Features for Image Classification. Master's thesis, Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), 2003.

[Lindeberg 98] T. Lindeberg. *Feature Detection with Automatic Scale Selection.* International Journal of Computer Vision, vol. 30, no. 2, pages 79–116, 1998.

[Lindeberg 99] T. Lindeberg. *Principles for Automatic Scale Selection.* In Handbook of Computer Vision and Applications, volume 2, pages 239–274. Academic Press, 1999.

[Loupias 99] E. Loupias & N. Sebe. *Wavelet-Based Salient Points for Image Retrieval.* Technical Report, Laboratoire Reconnaissance de Formes et Vision, 1999.

[Lowe 01] D.G. Lowe. *Local Feature View Clustering for 3D Object Recognition.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 682–688, Kauai, Hawaii, December 2001.

[Lowe 04] D.G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004.

[Martin 01] D. Martin, C. Fowlkes, D. Tal & J. Malik. *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.* In Proceedings of the 8th International Conference on Computer Vision, volume 2, pages 416–423, July 2001.

[Matas 02] J. Matas, O. Chum, M. Urban & T. Pajdla. *Robust Wide Baseline Stereo from Maximally Stable Extremal Regions.* In Proceedings of the British Machine Vision Conference, pages 414–431, 2002.

[Mikolajczyk 03] K. Mikolajczyk & C. Schmid. *A Performance Evaluation of Local Descriptors.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 257–263, 2003.

[Mikolajczyk 04] K. Mikolajczyk & C. Schmid. *Scale and Affine Invariant Interest Point Detectors.* International Journal of Computer Vision, vol. 60, no. 1, pages 63–86, 2004.

[Mikolajczyk 05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir & L. Van Gool. *A Comparison of Affine Region Detectors.* International Journal of Computer Vision, vol. 65, no. 1-2, pages 43–72, 2005.

[Naphade 00] M.R. Naphade & T.S. Huang. *A Probabilistic Framework for Semantic Indexing and Retrieval in Video.* In Proceedings of the IEEE International Conference on Multimedia and Expo (I), pages 475–478, 2000.

[Nayar 96] S.K. Nayar, H. Murase & S.A. Nene. *Parametric Appearance Representation.* In Early Visual Learning, pages 131–160. Oxford University Press, February 1996.

[Nene 96] S.A. Nene, S.K. Nayar & H. Murase. *Columbia Object Image Library (COIL-20).* Technical Report, Columbia University, February 1996.

[Niblack 93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E.H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos & G. Taubin. *The QBIC Project: Querying Images by Content, Using Color, Texture, and Shape.* In Proceedings of the SPIE International Symposium on Storage and Retrieval for Image and Video Databases, pages 173–187, 1993.

[Opelt 04] A. Opelt, M. Fussenegger, A. Pinz & P. Auer. *Weak Hypotheses and Boosting for Generic Object Detection and Recognition.* In Proceedings of the 8th European Conference on Computer Vision, volume 2, pages 71–84, 2004.

[Paredes 01] R. Paredes & A. Perez-Cortes. *Local Representations and a Direct Voting Scheme for Face Recognition.* In Workshop on Pattern Recognition in Information Systems, Setubal, Portugal, July 2001.

[Rosenberg 05] C. Rosenberg, M. Hebert & H. Schneiderman. *Semi-Supervised Self-Training of Object Detection Models.* In Seventh IEEE Workshop on Applications of Computer Vision, pages 29–36, January 2005.

[Schiele 00] B. Schiele & J.L. Crowley. *Recognition without Correspondence using Multidimensional Receptive Field Histograms.* International Journal of Computer Vision, vol. 36, no. 1, pages 31–50, 2000.

[Schmid 97] C. Schmid & R. Mohr. *Local Grayvalue Invariants for Image Retrieval.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 5, pages 530–535, 1997.

[Schneiderman 00a] H. Schneiderman. *A Statistical Approach to 3D Object Detection Applied to Faces and Cars.* PhD thesis, Carnegie Mellon University, Pittsburgh, 2000.

[Schneiderman 00b] H. Schneiderman & T. Kanade. *A Statistical Model for 3D Object Detection Applied to Faces and Cars.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1746–1759, June 2000.

[Shi 00] J. Shi & J. Malik. *Normalized Cuts and Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 22, no. 8, pages 888–905, August 2000.

[Sirovich 87] L. Sirovich & M. Kirby. *Low-Dimensional Procedure for the Characterization of Human Faces.* Journal of Optical Society of America, vol. 4, pages 519–524, 1987.

[Sivic 03] J. Sivic & A. Zisserman. *Video Google: A Text Retrieval Approach to Object Matching in Videos.* In Proceedings of the IEEE International Conference on Computer Vision, pages 1470–1477, Washington, DC, USA, October 2003.

[Sivic 05] Josef Sivic, Bryan Russell, Alexei Efros, Andrew Zisserman & William Freeman. *Discovering object categories in image collections.* In ICCV 2005, Beijing, China, 2005.

[Smith 97] S.M. Smith & J.M. Brady. *SUSAN - A New Approach to Low Level Image Processing.* International Journal of Computer Vision, vol. 23, no. 1, pages 45–78, 1997.

[Smolic 01] A. Smolic. *Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung parametrischer 2-D Modelle, Schätzverfahren und Anwendungen.* PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), 2001.

[Tourapis 02] A.M. Tourapis. *Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation.* In Proceedings of the SPIE Conference von Visual Communications and Image Processing, pages 1069–1079, January 2002.

[Ulges 04] A. Ulges. *Stereo Book - Document Capture using Stereo Vision.* Technical Report, Technical University Kaiserslautern, 2004.

[Viola 04] P. Viola & M.J. Jones. *Robust Real-Time Face Detection.* International Journal of Computer Vision, vol. 57, no. 2, pages 137–154, 2004.

[Weber 00] M. Weber, M. Welling & P. Perona. *Unsupervised Learning of Models for Recognition.* In Proceedings of the 6th European Conference on Computer Vision-Part I, pages 18–32, London, UK, 2000.

[Yan 05] R. Yan & M.R. Naphade. *Semi-Supervised Cross Feature Learning for Semantic Concept Detection in Videos.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages I: 657–663, 2005.

[Zhu 06] X. Zhu. *Semi-Supervised Learning Literature Survey.* Technical Report, Computer Sciences, University of Wisconsin-Madison, 2006.