Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than things you do expect) Plautus (ca 200(B.C.)

Project no: 027787

# DIRAC

## Detection and Identification of Rare Audio-visual Cues

Integrated Project
IST - Priority 2

DELIVERABLE NO: D1.2
Feature Detectors for Body Parts, Tracking, and Fast Object Detection

*Date of deliverable:31.12.2006*
*Actual submission date 26.1.2007*:

Start date of project: 01.01.2006 Duration: 60 months

*Organization name of lead contractor for this deliverable*: **Czech Technical University in Prague (CTU)**

Revision [1]

| Project co-funded by the European Commission within the Sixth Framework Program (2002-2006) | | |
|---|---|---|
| Dissemination Level | | |
| PU | Public | |
| PP | Restricted to other program participants (including the Commission Services) | |
| RE | Restricted to a group specified by the consortium (including the Commission Services) | X |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |

DIRAC
Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than
things you do expect) Plautus (ca 200(B.C.)

# D1.2 Feature Detectors for Body Parts, Tracking, and Fast Object Detection

Czech Technical University in Prague (CTU)
Eidgenoessische Technische Hochschule Zuerich (ETHZ)

***Abstract:***
This deliverable presents feature detectors and object modeling approaches for detecting objects in images. A fast local feature detector is presented with application in a multi-cue object detection. This presents a discriminative approach which classifies interest points detected in the images. A generative approach to object detection and tracking is also proposed, where a model of the object is generated from images captured by a multi-camera setup. A single camera can then be used for object tracking.

## Table of Content

# 1 Introduction

This deliverable can be divided into two main parts. The first part discusses local features for image recognition and represents a discriminative approach, that is, it selects the parts of the image which contains information based on feature detection. These detected cues are then combined together in a multi-cue object detection framework. The second part if this report presents a generative approach, where a model is generated from the data based on constraints, such as that an object can be separated from the background.

The deliverable is structured as follows. Section 2 presents a fast local feature extraction method. Evaluation of different feature classifiers is discussed in Section 3 with experimental results listed in Appendix A. Section 4 describes a multi-cue object detection and Section 5 presents a model based approach to object detection and tracking.

# 2 Fast Local-Feature Extraction from Perspective Images

ETHZ and KUL have together developed a novel scale- and rotation-invariant interest region detector and descriptor for perspective images, coined *SURF* (Speeded-Up Robust Features) (Bay et al. 2006). It is highly optimized for fast computation and approximates or even outperforms previously proposed schemes in terms of repeatability, distinctiveness, and robustness.

This is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors; and by simplifying those methods to the essential. This leads to a combination of novel detection, description, and matching steps.

Work on SURF has originally been started in a different project, but due to its success, it has been carried over to and extended in DIRAC. In particular, we have adapted SURF, which was originally designed for finding exact correspondences between the same structures in two images, to the more general task of finding corresponding object parts between different objects of the same category. This task is more difficult, since it requires to compensate for the larger intra-category variability inherent in natural object categories.

In the following, we give a short outline of the main ideas behind SURF, referring to (Bay et al. 2006, added to this deliverable) for details. Extensive experimental evaluation is presented in Appendix A, benchmarking SURF for an object detection task and comparing its performance against several other state-of-the-art feature extractors.

## 2.1 SURF Overview

The search for discrete image correspondences can be divided into three main steps. First, *interest points* are selected at distinctive locations in the image, such as corners, blobs, or T-junctions. The most valuable property of an interest point detector is its repeatability, i.e. that it reliably finds the same interest points under different viewing conditions. Next, the neighbourhood of each interest point is represented by a feature vector. This *descriptor* has to be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Finally, the descriptor vectors are *matched* between different images. The matching is often based on a distance between the vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this step takes, and a lower dimensionality is therefore desirable. However, lower-dimensional

feature vectors are in general less distinctive than their higher-dimensional counterparts. We address all three of those problems.

## 2.2 SURF Detector

When working with local features, a first issue that needs to be settled is the required level of invariance. Clearly, this depends on the expected geometric and photometric deformations, which are in turn determined by the possible changes in viewing conditions. Here, we focus on scale and image rotation invariant detectors and descriptors. These seem to offer a good compromise between feature complexity and robustness to commonly occurring deformations. In some cases, even rotation invariance can be left out, resulting in a scale-invariant only version of our descriptor, which we refer to as *upright SURF (U-SURF)*.

The SURF detector is based on the Hessian matrix (similar to the well-known *Hessian-Laplace* detector (Mikolajczyk et al. 2005a)), but uses a basic approximation. It relies on integral images to reduce the computation time for image derivatives. As a simpler alternative to Gaussian derivatives, the method evaluates 9 ´ 9 box filters, which can be computed independently of their scale by few lookups in the integral image. As verified in our experiments, the performance is comparable to the one using more exact Gaussian filters.

Our method then relies on the Hessian determinant for determining both the location and scale of feature points. Since the underlying box filters can be evaluated at any size at exactly the same speed through our use of integral images, the need to compute a Gaussian image pyramid is removed. Instead, we analyze the scale space by up-scaling the filter size rather than iteratively downscaling the image size. In order to localize interest point in the image and over scales, non-maximum suppression is applied in a 3 ´ 3 ´ 3 neighbourhood. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space.

## 2.3 SURF Descriptor

The SURF descriptor is based on similar properties as the well-known *SIFT* descriptor (Lowe 2004), with a complexity stripped down even further. It describes a distribution of Haar-wavelet responses within the interest point neighbourhood. Again, we exploit integral images for fast computation. Moreover, different versions of the descriptor with 32, 48, 64, and 128 dimensions are proposed to trade off computation and matching time for robustness.

The first computation step consists of determining a reproducible orientation based on information from a circular region around the interest point. For that purpose, we first calculate the Haar-wavelet responses in x and y direction in a circular neighbourhood around the interest point and weight them with a Gaussian centred on this point. The dominant orientation is estimated by maximizing the sum of all responses within a sliding orientation window covering an angle of 60°. Note that U-SURF skips this step.

Next, we construct a square region aligned to the selected orientation and extract the SURF descriptor from it. This region is split up regularly into 4 ´ 4 smaller sub-regions in order to preserve spatial information. For each sub-region, we compute approximated first derivatives $d_x$ and $d_y$ at a grid of 5 ´ 5 sample points and again weight them with a Gaussian centred at the interest point. Based on those results, we represent each grid sub-region by a set of 4 (for SURF-64) or 8 (for SURF-128) feature values computed from $d_x$ and $d_y$. Finally, the computed vectors for each cell are normalized to unit length and concatenated into a single feature vector.

## 2.4 Feature Matching

For fast indexing during the matching stage, we propose an improvement over the original matching scheme from (Lowe 2004). As a first and very quick matching criterion, our method considers the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest points. As those points are typically found at blob-like structures, the sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no additional computational cost, as it was already computed during the detection phase. In the matching stage, we only compare features if they come from the same contrast situation. This allows for faster matching without reducing the descriptor's performance.

## 2.5 Experimental Results

The different parameter choices for the individual SURF stages have been extensively evaluated, and the final detector and descriptor have been compared with the state-of-the-art on a set of standard test sets for the task of image matching. Results on this evaluation are presented in (Bay et al. 2006). Those results show that SURF performs at the same level or even better than previously proposed methods, while being faster by a factor of 3-5. Executables of the latest version of the SURF detector and descriptor for both Linux and Windows are made publicly available[1].

## 2.6 Connection to Acoustic Features and Speech Recognition

There are some remarkable parallels between the visual features described here and the acoustic features employed for speech processing in deliverable D1.3. Those acoustic features consist of two-dimensional matched filters modelling cortical receptive fields that are sensitive to time- and frequency-localized stimuli. Recent research results indicate that the processing of such features extending over longer temporal spans plays an important role for human speech recognition and understanding.

In comparison, the SURF detector localizes a feature in image space, and its automatic scale selection determines the spatial frequencies that are encoded by the descriptor. As each local descriptor is based only on a single image, however, the temporal dimension currently plays only a minor role. In more recent work, ETHZ and KUL have therefore started an extension of SURF for spatio-temporal feature extraction. This extension is inspired by successful speech-processing techniques, in which the temporal dimension is of prime importance. Its purpose is to find features that are well-localized in space and time and encode a longer temporal span, while applying the same speedup techniques already used in spatial SURF.

It is to be expected that the additional information available from such a descriptor will be useful for visual classification tasks in which the temporal domain becomes important, such as action classification and temporal matching of video sequences. Research on spatio-temporal SURF is underway. First experiments on inter-video feature matching and human action classification are promising, but further fine-tuning and experimental evaluation is necessary.

# 3 Evaluation of Local Features for Object Detection

Local-feature based approaches have shown considerable promise for dealing with the large degree of intra-category variation and partial occlusion inherent in real-world categorization and detection tasks. Consequently, many approaches have been developed that use local

[1] SURF features website: http://www.vision.ee.ethz.ch/~surf

features in different ways, and considerable progress has been made in the understanding of the underlying feature detectors and descriptors.

Users can now choose from a plethora of different local feature types. However, most of those have only been evaluated for exact matching tasks, such as image registration or object identification. Only few studies have so far tried to experimentally quantify the performance of different feature combinations for object detection tasks, which impose a very different set of constraints.

ETHZ therefore performed a performance evaluation of local features, extracted from perspective images, specifically for the task of fast object detection. In the context of our study, we consider a *local feature* as a combination of an *interest region detector* and a *local descriptor*.

A first version of this evaluation, comprising 3 state-of-the-art region detectors (*Harris-Laplace*, *Hessian-Laplace*, and *DoG*) and 5 different local descriptors (*SIFT*, *GLOH*, *PCA-SIFT*, *Shape Context*, *Patch*), as well as all of their 15 combinations, has been published in (Leibe et al. 2006) (appended to this deliverable). In more recent work, this evaluation has been extended to also include the *SURF* detector and *SURF-64* and *SURF-128* descriptors introduced in Section 2, raising the total number of evaluated feature combinations to 26 out of the possible 28 combinations. As this more recent part has not been published yet, we will report it in more detail in Appendix A in this deliverable.

# 4 Multi-Cue Integration for Object Detection

The results presented in Appendix A provide valuable information about the performance of a large selection of local cues and how they should be applied for fast object detection. Yet, each feature detector and descriptor can only capture part of the information contained in the image, and indeed its value for an application depends on the degree to which it can distil exactly the right kind of information for a specific purpose. As a consequence, the better a feature extractor is suited for a specific task, the more likely it is to degenerate when the task conditions deviate too far from its target scenario. In order to be both discriminative and robust, an application should therefore utilize a combination of different local cues.

For this reason, ETHZ has developed a novel method for integrating multiple local cues in the context of object detection (Leibe et al. 2006). Rather than to fuse the outputs of several distinct classifiers in a fixed setup, our approach implements a highly flexible combination scheme, in which the contributions of all individual cues are opportunistically recombined depending on their explanatory power for each new test image. The key idea behind our approach is to integrate the cues over an estimated top-down segmentation which allows to quantify how much each of them contributed to the object hypothesis. By combining those contributions on a per-pixel level, our approach ensures that each cue only contributes to object regions for which it is confident and that potential correlations between cues are effectively factored out.

Experimental results on several challenging data sets show that the proposed multi-cue integration scheme increases object detection performance significantly. This improvement is consistent over three different data sets and two categories. It is particularly prominent for the detection precision and leads to high recognition rates at the zero-false-positive level. In the following, we will shortly outline the idea behind our proposed multi-cue integration procedure. The paper (Leibe et al. 2006), attached to this report, provides more details.

## 4.1 Integration Strategy

Previous studies comparing different local cues, including the one in Section 3, have considered each cue in isolation. For multi-cue integration, it is however also important to know how different cues interact, i.e. how correlated their responses are and what new information an additional cue can contribute. This information is difficult to retrieve, as different cues are often not directly comparable, both because they typically have different dimensionalities and because they represent information in different ways.

Previous research has therefore mainly focused *classifier combination*, i.e. on the problem of fusing the outputs of several "black-box" classifiers, possibly with associated confidence ratings. This approach is valid if the classifiers are independent. In our application, however, their outputs are often correlated, and the degree of correlation may vary from image to image. Rather than just to fuse the outcomes of several classifiers, we therefore need to explore how the underlying information and the respective support in the image can be combined.

In our proposed multi-cue integration approach (Leibe et al. 2006), we present a flexible integration scheme which combines different local cues in an opportunistic manner depending on their explanatory power for the image at hand. The integration proceeds in two stages. The first stage extends the ISM recognition procedure to include multiple cues and collect their contributions in a combined Hough voting space. Its main purpose is to express the cues on a common basis, in terms of their similarities to a learned set of prototype appearances, so that their information can be pooled, and initial hypotheses can be found. This stage makes the cues comparable. However, it still ignores cue correlation. Indeed, it has no other choice, since correlation can only be measured relative to a reference hypothesis, and hypotheses are only available after the stage has been executed.

Therefore, the second step then reveals correlation by backprojecting hypotheses to the image and computing a top-down segmentation for each cue separately. This step uses an extension of the ISM top-down segmentation algorithm, described later in the appendix and in (Leibe et al. 2004, Leibe et al. 2005a), adapted to multiple cues. The thus-obtained segmentations show on a per-pixel level which image structures were responsible for a detection and how much each pixel contributed to the cue's response. The correlation between two cues can then be expressed as the overlap between their respective p(*figure*) probability maps.

This expression in terms of the pixel-wise top-segmentation is the key idea between our multi-cue integration strategy. If two cues draw their support from different image pixels, they are clearly orthogonal, and their evidence should be combined. If, on the other hand, their support is based on the same image pixels, the two cues are correlated, and their combined score should be adapted accordingly.

Once the cue correlation has been identified, the next question is how to use this information to improve recognition performance. In (Leibe et al. 2006), we present three combination criteria that relate to different strategies for this step. The canonical strategy would be to simply ignore correlation and *sum* the contributions of all cues. As the experimental results from (Leibe et al. 2006) show, this criterion breaks down when the cues are strongly correlated. The opposite philosophy would be to completely factor out correlation and only use the cue that best explains the current pixel. This translates into a per-pixel *max* operation. However, this strategy is problematic, too, if some cues cannot fully be trusted. We therefore propose a third criterion of building up a robust per-pixel *average* of only the sufficiently confident cues, which combines both ideas from above. As the evaluation in (Leibe et al. 2006) demonstrates, this criterion performs consistently best. As also shown in (Leibe et al. 2006), the resulting multi-cue integration significantly improves detection performance for different test sets and several object categories.

### 4.2 Application for Object Detection

In addition, our multi-cue detection system has been used in deliverable D3.1, where it has been successfully applied to difficult real-world object detection tasks. For this application, the multi-cue integration scheme proved very important in order to achieve robustness to degraded imaging conditions encountered there. In particular, the input image streams were often captured at relatively low resolutions (e.g. 384 ´ 248 pixels for a full street scene image) and contained strong contrast changes between brightly lit areas and dark shadows.

Any single interest region detector on its own had problems finding enough useful features under those deteriorated conditions. However, the combination of several different detectors and descriptors, robustly combined with the *average* criterion, provided enough information to allow robust object detection in this application.

## 5 Model-based Detection and Tracking of Body Parts

The discriminative approach described above has been also complemented by another approach, which uses a generative model. The model of the object to be tracked is build online from the observations in the input sequence.

### 5.1 Automatic Model Generation and Tracking

A general tracking approach for rigid bodies is described in detail in (Zimmermann et al. 2006). It is a model-based generative approach where the model can be provided beforehand but it can be also built automatically from 3D reconstruction of an object in front of the cameras. For the model construction, at least a stereo pair of cameras is required, but for the tracking, only a single camera is sufficient. **Figure 1** shows an image from a camera tracking an object with the model projected into the image, together with a reconstructed trajectory of the object in 3D and views of the 3D model reconstructed from the images.

The model is always updated when new parts consistent with the current model are found. For example, the model is built based on observation of a head but then the motion of the torso is found to be consistent with the motion of the head and the model is updated to accommodate also the torso. The model is not considered to be complete, only the fact that no new 3D points should be added is indicated.

Unlike methods based in 2D image models, a 3D model-based approach can model the variance of the tracked object caused by changing orientation. At each tracking step, the model T is projected by a function f, which describes the position and pose of the tracked object, into the current image I, see **Figure 2**.

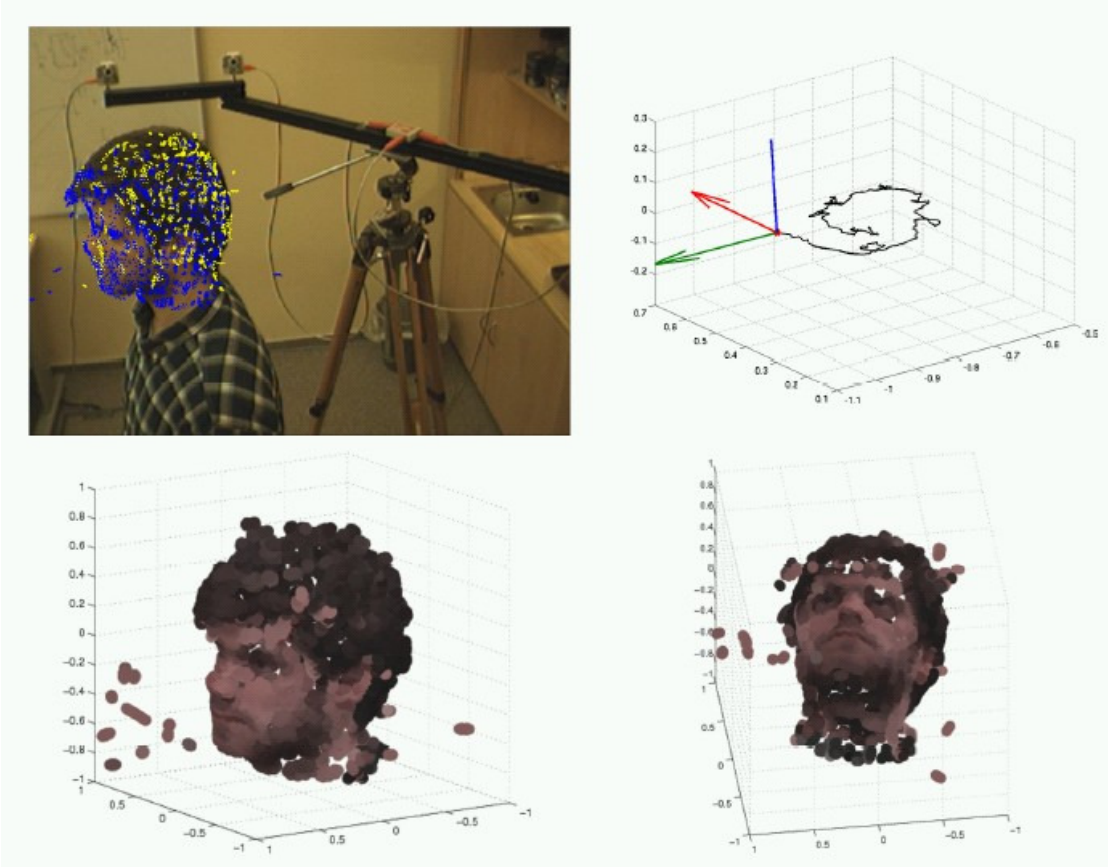


**Figure 1:** Model of a face is automatically generated from a stereo reconstruction and then tracked through a sequence. Top left: model projected into an image. Top right: tracked trajectory. Bottom row: two different views of the reconstructed model.

The 3D model does not have to be complete and dense, a fish-scales representation is used for the model. Fish-scales represent a point based model with normal and colour information, a lambertian surface is assumed. It should be noted that the method is general, not only limited to people tracking, see **Figure 3**.

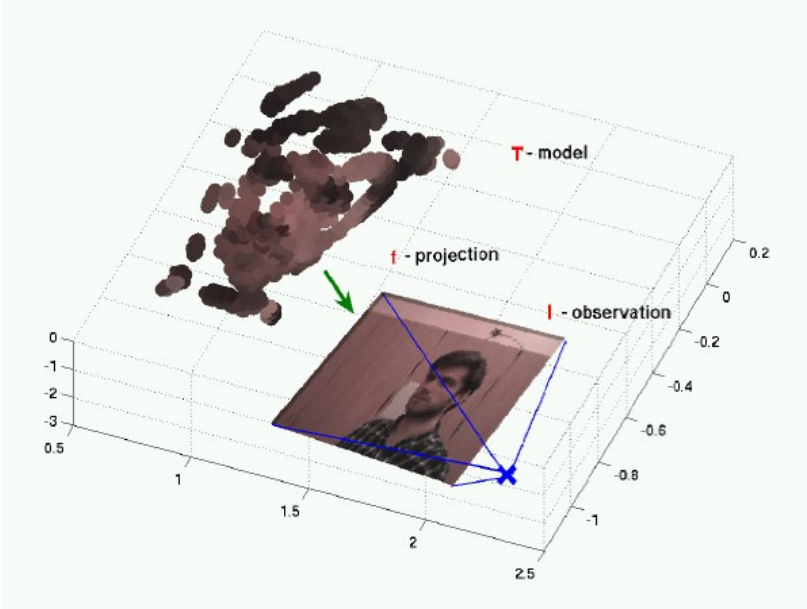



**Figure 2:** Model T is projected by a function f into an actual image I.

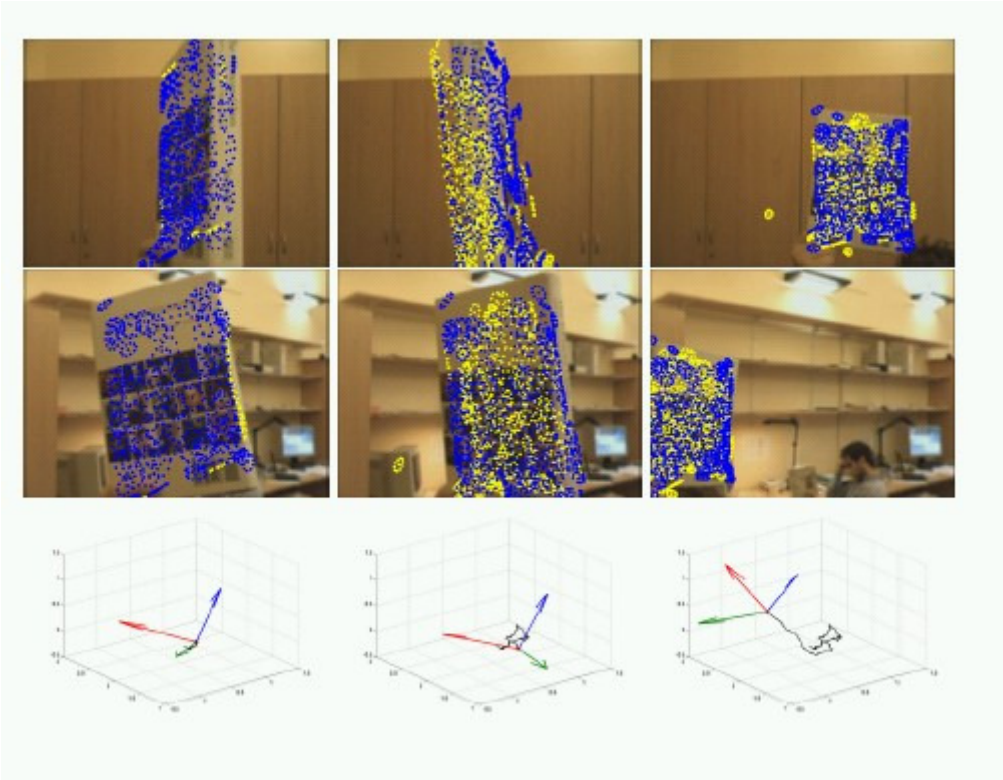


**Figure 3:** The proposed approach is not limited to human body parts.

The above method was applicable only to rigid bodies, but a human body is not rigid. There are several ways to model the human body; we will discuss two which can be applied for the DIRAC project.

## 5.2 Tracking a Human Body in a Monocular Sequence

The first approach models the human body as a tree of rectangular parts connected by spring like joints, see **Figure 4**. Each body part is detected separately using colour constancy criteria on a foreground image. The most probable configuration is obtained by randomized sampling. Probabilistic model which includes appearance models of individual body parts and their mutual positions is used and the best sample is selected to minimize the Chamfer distance between the models and the features in the image, as it is illustrated in **Figure 5**. The model has to be trained separately on a training set, the poses of people in the training images determine which poses will be detected in the image and the model cannot generate new poses significantly different from those in the test set.

We have implemented this state of-the-art approach (Felzenszchwalb and Huttenlocher 2005) to gain experience and to be able to compare our techniques to the state of the art. The details are described in the master thesis (Fajt 2006).

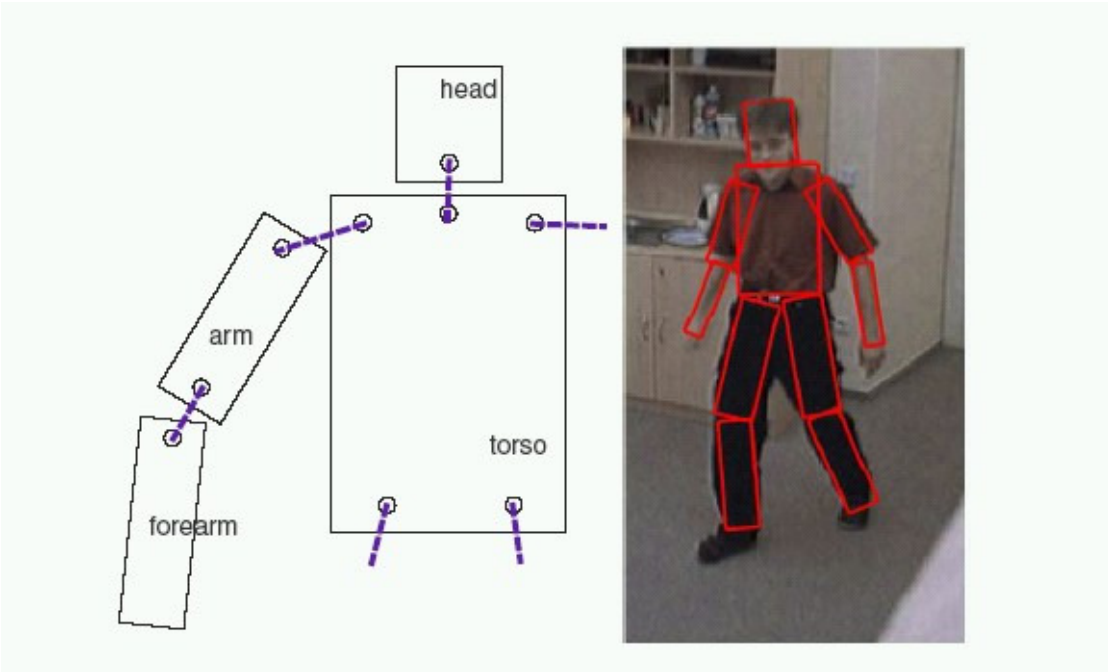



**Figure 4:** Model of a human body as rectangular part connected with strings.

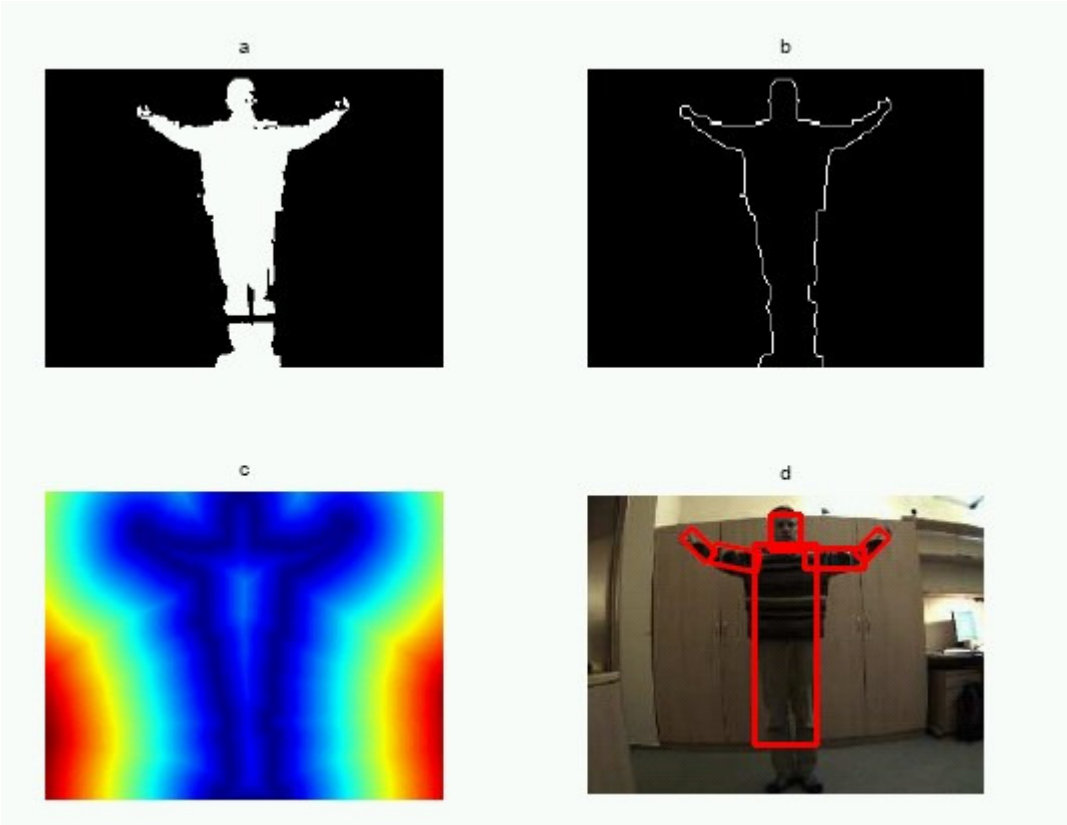



**Figure 5:** Model fitting as a minimization of Chamfer distance.

## 5.3 Bone and Mesh Based Model of a Human Body

The second approach is based on computer graphics models of human body, which consist of bones, connected by joints and covered by meshes modelling the muscles. The bone model has a structure of a tree, as it is depicted in **Figure 6** and can be quite detailed. The model can be covered with texture from real images, once it is fitted to images from a multicamera setup, see **Figure 7**.

We use this model to get ground truth data of realistic poses. There are numerous datasets with motion capture data of real humans which can be used with the bone and mesh human model to generate training and testing datasets. This method can be used to generate both training and testing datasets for the previous approach. The work was is described in detail in the master thesis (Mazany 2006).

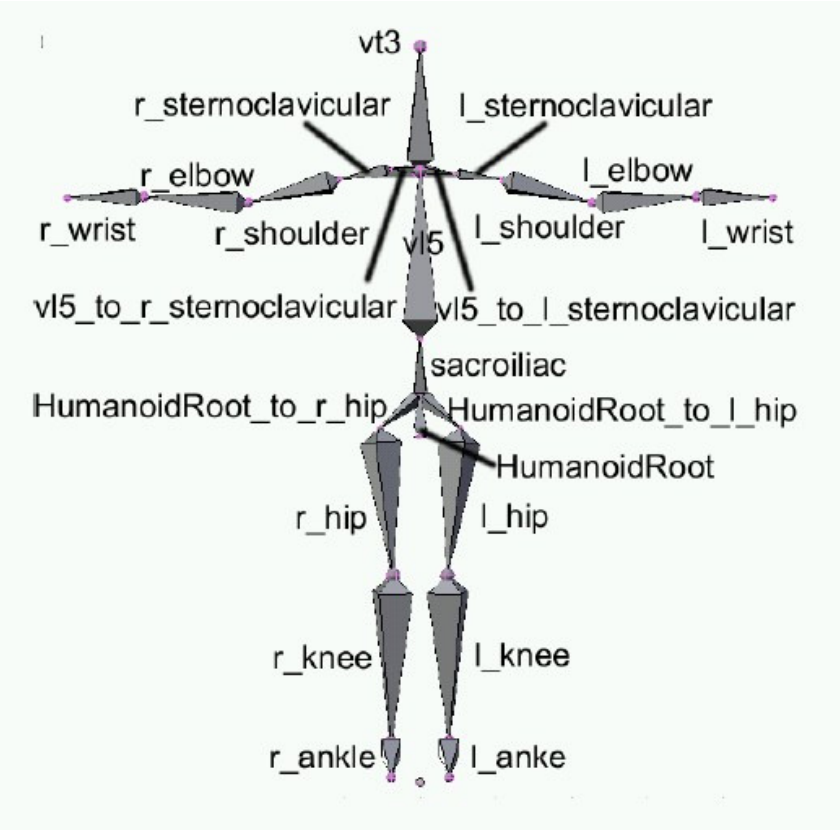



**Figure 6:** Bone-based model of a human body.

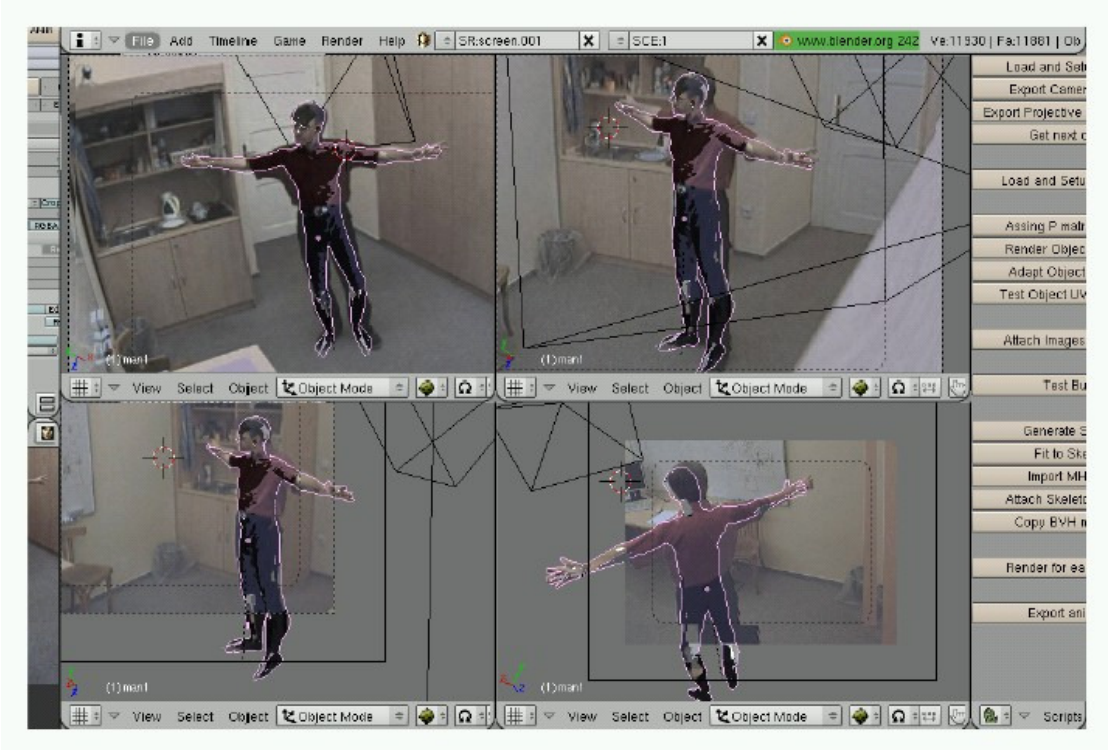



**Figure 7:** Fitting the bone-based model to images of a human in a multicamera setup.

## 6 Conclusion

We worked on object detection and tracking with focus on human body parts detection. The results are promising and several important building blocks have been implemented, such as the fast local feature detection, multi-cue object detection procedure, rigid body tracking based on a model generated automatically from the data, and human body detection using an articulated model.

## 7 Reference

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

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded-up robust features. In *Ninth European Conference on Computer Vision (ECCV'06)*, 2006. **(appended to this document)**.

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

M. Everingham and others (34 authors). The 2005 pascal visual object class challenge. In *Selected Proceedings of the 1st PASCAL Challenges Workshop*, LNAI. Springer, 2006 (in press). http://www.pascal-network.org/challenges/VOC/.

L. Fajt. Pictorial Structural Models, Learning and Recognition in Image Sequences *Diploma thesis, Department of Cybernetics, FEE, CVUT, Czech Republic. TR-CAK-2007-27*, 2006.

P. F. Felzenszchwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition, *International Journal of Computer Vision, 61(1), pp. 55-79*, 2005.

R. Fergus, A. Zisserman, and P. Perona. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pages 511–517, 2004.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, 2004.

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *submitted to International Journal of Computer Vision*, 2005.

B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *British Machine Vision Conference (BMVC'06)*, 2006. **(appended to this document)**.

B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

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

O. Mazany. Articulated 3D Human Model and its Animation for Testing and Learning the Algorithms of Multi-Camera Systems *Diploma thesis, Department of Cybernetics, FEE, CVUT, Czech Republic. TR-CAK-2007-25*, 2006.

K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Tenth International Conference on Computer Vision (ICCV'05)*, 2005.

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

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

R.Šára and R.Bajcsy. Fish-scales: Representing fuzzy manifolds. *In S. Chandran and U. Desai, editors, Proc. 6th International Conference on Computer Vision, pages 811-817, New Delhi, India, January* 1998.

K. Zimmermann, T. Svoboda, and J. Matas. Multiview 3D Tracking with an Incrementally Constructed 3D Model *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT).* 2006. **(appended to this document)**.

# A Evaluation of Local Features for Object Detection

## A.1 Recognition Approach

As a testbed for our evaluation, we use the Implicit Shape Model (ISM) approach by (Leibe et al. 2004, Leibe et al. 2005a), which combines the capabilities of object detection and top-down segmentation. This model represents an object category by a set of local appearance clusters (a *codebook*) and their spatial occurrence distributions. In the following, we will briefly review its main components. Since a basic knowledge of the ISM approach is necessary for the evaluation, we will briefly review its main components in the following.

### *A.1.1 Training*

Training proceeds in two steps. First, local features are extracted from the training images using the selected detector/descriptor combination and clustered to form the codebook. The codebook is created using Average-Link agglomerative clustering (Agarwal et al. 2004, Leibe et al. 2004). Starting with each feature as a separate cluster, the two most similar clusters $P$ and $Q$ are merged as long as the average similarity between their constituent features (and thus the cluster compactness) stays above a certain threshold $t$:

$$sim(P,Q) = \frac{1}{|P||Q|}\sum_i \sum_j sim(p_i, q_j) \geq t. \quad (1)$$

In a second run over the training data, the spatial occurrence distributions are estimated by recording for each codebook entry all matching locations on the training objects. For this, the extracted features $f_i$ are compared to all codebook entries $C_j$, and all matches are stored for which

$$sim(f_i, \mathcal{C}_j) \geq \theta. \quad (2)$$

Together with each occurrence, the approach stores a local segmentation mask, which is later used for inferring top-down segmentations.

### *A.1.2 ISM Recognition*

During recognition, local features are again extracted from the image and matched to the codebook. Each matching codebook entry then cast votes for possible object locations and scales in a probabilistic extension of the Hough transform (Leibe et al. 2004).

Formally, this is expressed as follows. Let $f$ be a local descriptor computed at location $l$. When matched to the codebook, it may activate several codebook entries $C_i$ with probabilities $p(C_i|f)$. Each matched codebook entry then votes for instances of the object category $o_n$ at different locations and scales $l=(l_x, l_y, l_s)$ according to its learned occurrence distribution $P(o_n,l|C_i,l)$. Thus, any single vote has the weight $P(o_n,l|C_i,l)p(C_i|f)$ and the feature's total contribution can be expressed by the following marginalization:

$$P(o_n, \lambda | f, \ell) = \sum_i P(o_n, \lambda | \mathcal{C}_i, \ell) p(\mathcal{C}_i | f) \quad (3)$$

The votes are collected in a continuous 3D voting space, and object hypotheses are found by Mean Shift Mode Estimation using a scale-adaptive kernel $K$ with bandwidth $b(l)$ (Leibe et al. 2005a):

$$\hat{p}(o_n, x) = \frac{1}{nb(\lambda)^d}\sum_k \sum_j p(o_n, \lambda_j | f_k, \ell_k) K\left(\frac{\lambda - \lambda_j}{b(\lambda)}\right) \quad (4)$$

### *A.1.3 Top-Down Segmentation*

Once a hypothesis $h=(o_n, l)$ has been found, its top-down segmentation can be inferred by backprojecting the supporting votes to the image and combining them with the local patch segmentation masks $p(\mathbf{p}=fig.|o_n, l, C_i, l)$ that have been stored for each recorded codebook occurrence during training. As shown in (Leibe et al. 2004), the per-pixel probabilities of each pixel containing *figure* or *ground* can then be obtained by a double marginalization, first over sampled features, then over codebook entries:

$$p(\mathbf{p}=figure|o_n,\lambda) = \sum_{\mathbf{p}\in(f,\ell)}\sum_i p(\mathbf{p}=fig.|o_n,\lambda,\mathcal{C}_i,\ell)\frac{P(o_n,\lambda|\mathcal{C}_i,\ell)p(\mathcal{C}_i|f)p(f,\ell)}{p(o_n,\lambda)} \quad (5)$$

Based on these results, the final segmentation is computed by building the likelihood ratio between *figure* and *ground* probabilities.

### *A.1.4 Segmentation-based Verification*

Finally, the ISM approach implements an MDL-based hypothesis verification stage which uses the top-down segmentations to disambiguate overlapping hypotheses. Each hypothesis $h$ is evaluated based on the *savings* \cite{Leonardis95} that can be obtained in the description of an image by explaining part of it by $h$. The savings of each hypothesis are expressed as (see (Leibe et al. 2004, Leibe et al. 2005a) for the complete derivation)

$$S_h = -\kappa_1 + (1-\kappa_2)\frac{N}{A_\sigma} + \kappa_2\frac{1}{A_\sigma}\sum_{\mathbf{p}\in Seg(h)} p(\mathbf{p}=fig.|h) \quad (6)$$

where $N$ is the number of pixels that can be explained by $h$, $A_s$ is its *expected area* at scale $s$, and $k_1$ and $k_2$ are constants. $k_2$ is a weighting factor to balance out the influence of a hypothesis's area versus its support in the image (left at a fixed value in our experiments), and $k_1$ is the parameter over which the final performance curves are plotted.

If multiple hypotheses overlap, their respective savings terms interact, since each pixel can only be assigned to a single hypothesis. Using the method from (Leibe et al. 2005a), hypothesis selection is formulated as a quadratic boolean optimization problem, which allows to find the combination of hypotheses that best explains the image (see (Leibe et al. 2005a) for details).

## A.2 Experimental Setup

### *A.2.1 Interest Region Detectors*

We compare four different scale-invariant interest region detectors. The *Harris-Laplace* and *Hessian-Laplace* detectors look for scale-adapted maxima of the Harris function and Hessian determinant, respectively (Mikolajczyk and Schmid 2005), where the locations along the scale dimension are found by the Laplacian-of-Gaussian. The *DoG* detector (Lowe 2004) finds regions at 3D scale-space extrema of the Difference-of-Gaussian. The *SURF* detector (Bay et al. 2006), finally, is highly optimized for run-time efficiency. It searches for 3D scale-space extrema of the Hessian determinant, but approximates all derivatives through Haar wavelets computed on an integral image representation.

### *A.2.2 Region Descriptors*

In addition, we evaluate seven different region descriptors. *SIFT* descriptors (Lowe 2004) are 3D histograms of gradient locations and orientations with 4 ´ 4 location and 8 orientation bins. The resulting descriptor has 128 dimensions. *GLOH* descriptors (Mikolajczyk and Schmid 2005) are an extension of *SIFT*. They use 17 location and 16 orientation bins organized in a log-polar grid. PCA is used to reduce the dimensionality to 128. *PCA-SIFT* (Ke and Sukthankar 2004) are vectors of image gradients in $x$ and $y$ direction sampled within the support region and reduced to 36 dimensions with PCA. *Shape Context* (*SC*) (Belongie et al.

2002, Mikolajczyk and Schmid 2005) descriptors are histograms of gradient orientations sampled at edge points in a log-polar grid with 9 location and 4 orientation bins and thus 36 dimensions.

*SURF* descriptors (Bay et al. 2006) are also based on a grid of 4 ´ 4 location bins, where each bin is represented by the sums of signed and absolute gradient values in $x$ and $y$ direction. We consider two variants, *SURF-64* and *SURF-128*, which differ in the number of dimensions used to describe each bin. For comparison and as a baseline method, we include simple 25 ´ 25 pixel *Patches* (Agarwal et al. 2004, Leibe et al. 2004), which lead to a descriptor of length 625.

This set of descriptors was explicitly chosen to sample different sources of information. *SIFT*, *GLOH*, *PCA-SIFT*, and the *SURF* variants are based on gradient information; *SC* descriptors are based on edges; and *Patches* take the full image region into account. The evaluation is performed with an own implementation of the *DoG* detector (denoted *eDoG* in the figures) and *Patch* descriptor. For all other detectors and descriptors, we used the implementations publicly available[2]. Since our test sets contain only little in-plane rotation, only the rotation-variant versions of the descriptors are used. Patches are compared using *Normalized Correlation*; all other descriptors are compared using Euclidean distances.

### *A.2.3 Training and Test data*

We first evaluate detection performance on the TUD motorbike set, which is part of the PASCAL collection (Everingham et al. 2005). This data set consists of 115 images containing a total of 125 motorbikes at different scales and with clutter and occlusion. Training is done on 153 motorbike side views from the CalTech training set (Fergus et al. 2003) which are shown in front of uniform backgrounds allowing for easy segmentation.

In an extension of this study (Leibe et al. 2006), appended to this report, we then show that the results generalize also to other scenarios by applying a subset of 9 successful detector-descriptor combinations to two more challenging data sets using the same parameter settings. The first is the VOC motorbikes `test2` set, which has been used as a localization benchmark in the 2005 PASCAL Challenge (Everingham et al. 2005). This data set consists of 202 images containing a total of 227 motorbikes at different scales and seen from different viewpoints. Only about 37% of those motorbikes are shown in side views, though, thus limiting the maximally achievable recall for our system. Finally, we apply the different cues to the pedestrian test set from (Leibe et al. 2005a). It consists of 209 images containing crowded scenes with a total of 595 pedestrians, mostly shown in side views but with significant overlap and occlusion. Training for this test is done on 216 side views of pedestrians for which a segmentation mask was available, using the same parameter settings as for the motorbike experiments.

### *A.2.4 Evaluation Criterion*

In all three cases, the task is to detect and localize the objects in the test images and determine their correct bounding boxes (using the evaluation criterion from (Leibe et al. 2005a) for the first and third test set, and the criterion from (Everingham et al. 2005) for the second test set). Throughout the evaluation, detection performance is presented in terms of *precision* and *recall*, which are defined as follows:

---

[2] Oxford interest point webpage. http://www.robots.ox.ac.uk/~vgg/research/affine/ detectors.html.
Surf features website. http://www.vision.ee.ethz.ch/~surf.

$$precision = \frac{\#CorrectDetections}{\#CorrectDetections + \#FalsePositives} \quad (7)$$

$$recall = \frac{\#CorrectDetections}{\#CorrectDetections + \#FalseNegatives} \quad (8)$$

By varying the final acceptance threshold for the hypothesis scores, we create a recall-precision curve (RPC) for the selected detector/descriptor combination.

The performance of each cue will depend on several additional parameters. In order to investigate the effects of those parameters, we create an RPC for each parameter setting and determine the cue's *equal-error-rate performance* (EER), i.e. the point on the curve at which precision and recall are equal. We then plot a curve showing how those EER values change when the parameter is varied.

### *A.2.5 Experimental Procedure*

In order to obtain an unbiased estimate of the cues' potentials, it is important to ensure that they are evaluated at their optimal setting. As a first step, we therefore try to find each cue's performance optimum. In the recognition approach, as formulated at the beginning of this appendix, there are four open parameters that need to be adjusted for each cue.

The first is related to the question how much the clustering step should compress the training features during codebook generation. With the agglomerative clustering scheme we are using, this translates to the question how compact the codebook clusters should be for optimal performance. One option is to define a *minimum similarity* after which clustering should be stopped. Another option is to fix a certain *cluster compression ratio #features/#clusters*, which should only be dependent on the selected interest region detector. Previous evaluations (Mikolajczyk et al. 2003) have favored the latter option, but it is not guaranteed that this choice is optimal. Our evaluation will investigate this issue in the next section.

The second question is how to set the activation threshold for matching features to the codebook during the first stage of the recognition procedure. This parameter determines how tolerant the matching process is and thus how many of the sampled features may contribute to a hypothesis. In this evaluation, we set the matching threshold to the same value as the final cluster similarity that stopped agglomerative clustering, so that we are left with only one parameter. The interpretation of this design choice is that we allow each test image feature to activate all codebook entries with which it *could* potentially have been merged during clustering if this feature had been observed in the training set.

The third parameter comes into play during recognition. It determines how strong an initial peak $\hat{p}(o_n, x)$ in the voting space has to be before an object hypothesis is created from it. This parameter is mainly used for speed reasons, but since it also gives a measure of the available image evidence, a high cut-off threshold reduces the potential number of false positives. In this evaluation, we set the parameter by performing an initial test run on test set 1 and fixing the cut-off threshold to the highest value that still yields 98% recall from the remaining hypotheses (or alternatively the highest reachable recall value for the given cue). Such a procedure follows the assumption that some object instances may simply be too hard to detect for any single cue.

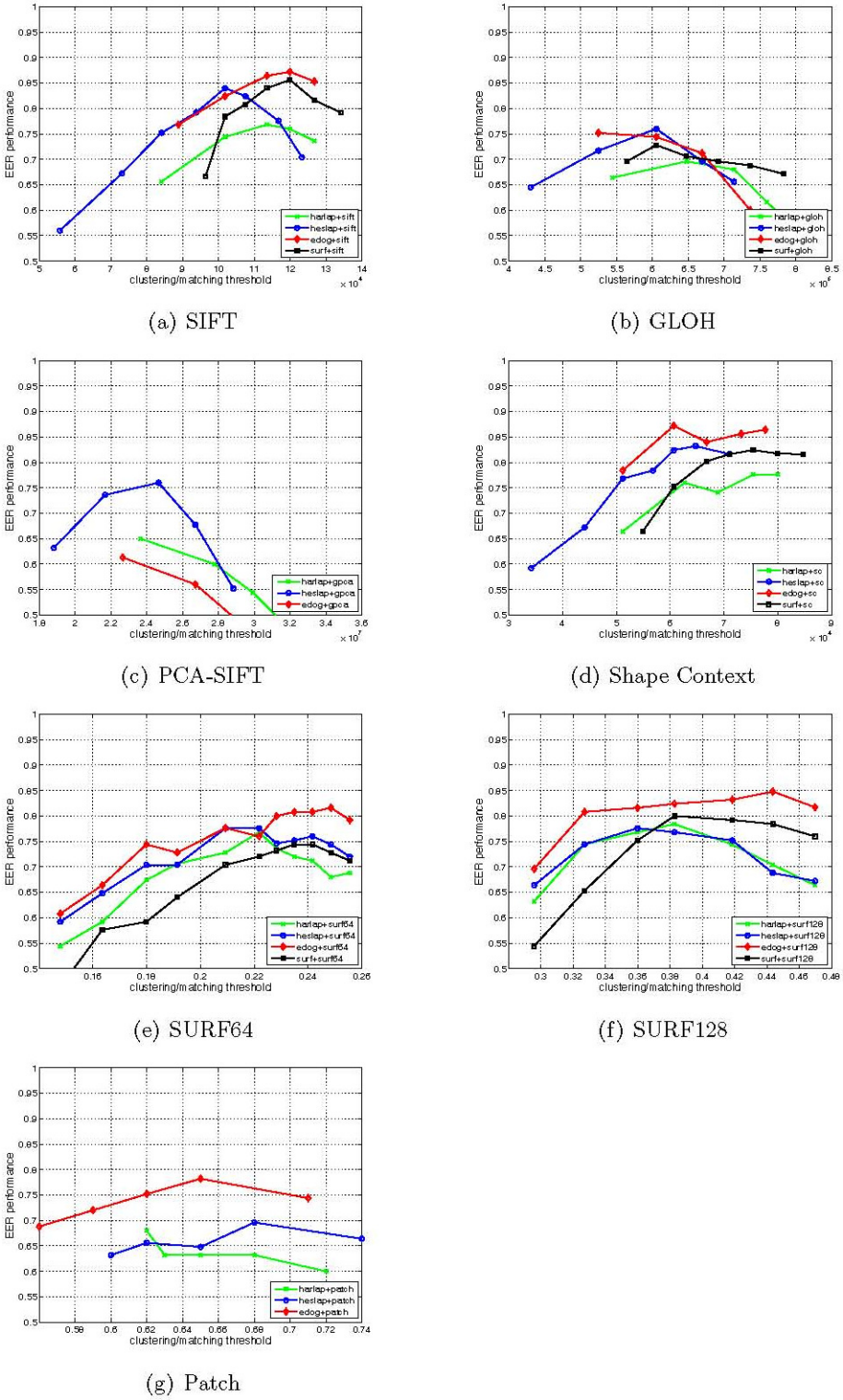
Finally, the last parameter is $k_1$ from eq. 6, which corresponds to a final acceptance threshold for the confidence in a verified hypothesis. This is the parameter over which the performance curves are plotted.

### A.3 Recognition Approach

In order to analyze the clustering/matching threshold's influence on recognition performance, we applied all detector/descriptor combinations to the TUD motorbikes set and compared their equal error rate (EER) detection performance for 5—11 different threshold settings. Figure 8 shows the results of this experiment ordered by descriptor. **Figure 10** shows the same results as above, but this time ordered by detector.

#### *A.3.1 Clustering Similarity vs. Compression Ratio*

From the plots, we can make the following observations. First, when comparing descriptors across different detectors, a distinct performance optimum can be found at a certain clustering/matching similarity for *SIFT*, *GLOH*, *PCA-SIFT*, *Shape Context*, and *SURF*. The cluster compression ratio, on the other hand, does not seem to have a consistent influence. We can therefore formulate the recommendation to use the cluster similarity as a criterion for selecting the clustering level for those descriptors.




(a) SIFT (b) GLOH

(c) PCA-SIFT (d) Shape Context

(e) SURF64 (f) SURF128

(g) Patch

**Figure 8:** Single-cue EER performances for all detector/descriptor combinations on the TUD motorbikes as a function of the clustering/matching threshold.

### *A.3.2 Effect of Different Detectors/Descriptors*

For the following experiments, we choose the best-performing clustering/matching threshold for each detector/descriptor combination and only report the cue's performance at this optimal setting. **Figure 11** and **Figure 9** show the corresponding recall-precision curves. Together, these results allow to rank the detector/descriptor combinations based on their detection performance. For the descriptors, *SIFT* and *Shape Context* perform consistently best over all four detectors. *SURF128* performs only slightly worse, followed by *SURF-64* and *GLOH*. *Patch* descriptors only perform well based on *DoG* regions, while *PCA-SIFT* can only convince in conjunction with the *Harris-Laplace* and *Hessian-Laplace* detectors.

For the detectors, *DoG* performs best in 5 of the 7 cases. Next come *Hessian-Laplace* and *SURF*, with only small differences between them. *Harris-Laplace*, finally, ranks last in all but 2 cases. In general, however, the performance differences between the detectors are much smaller than those between the descriptors.

In terms of combinations, *DoG+SIFT* and *DoG+Shape Context* obtain the best performance with 87.2\% EER, followed by *SURF+SIFT* with 85.6% EER, *DoG+SURF128* with 84.8%, *Hessian-Laplace+SIFT* with 84.0%, and *Hessian-Laplace+SC* with 83.2%. Considering the limited size of the test set and the discrete parameter sampling, the relatively small differences between those combinations should however not be overrated. All of them are viable alternatives in terms of detection performance.

A possible explanation for the relatively poor performance of *DoG* in conjunction with *GLOH* and *PCA-SIFT* could be that those two descriptors involve a PCA dimensionality reduction step, which projects the original feature vectors on a pre-defined lower-dimensional basis. Personal communication with the author of the used implementation of those descriptors[3] revealed, however, that the corresponding PCA basis functions were computed only from a collection of *Hessian-Laplace* and *Harris-Laplace* regions. As a result, the resulting descriptors are probably not well-suited for representing *DoG* regions.

[3] Oxford interest point webpage. http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html.

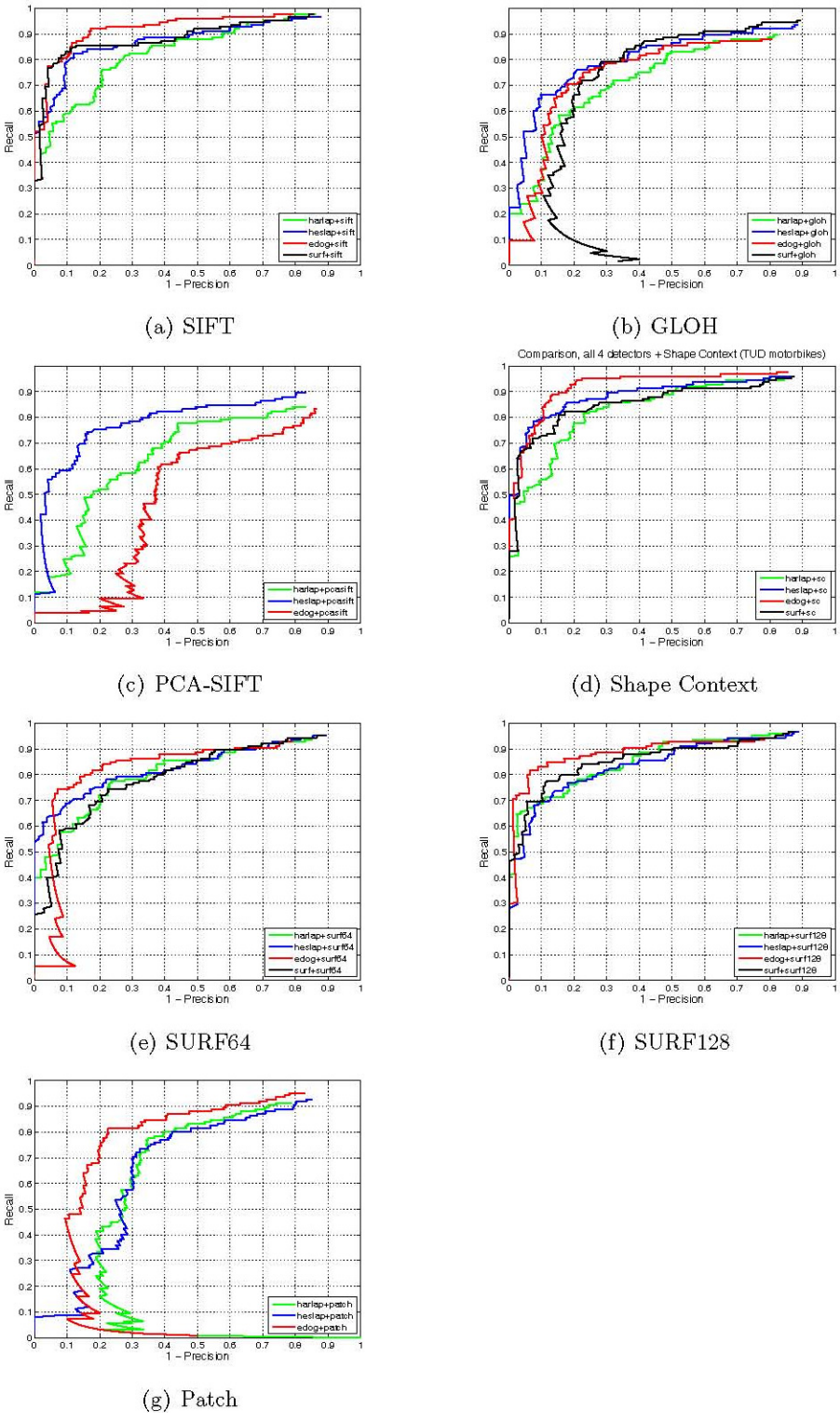



**Figure 9:** Comparison of the different detectors with all 7 descriptors (on the TUD *motorbikes, optimal parameter settings).*

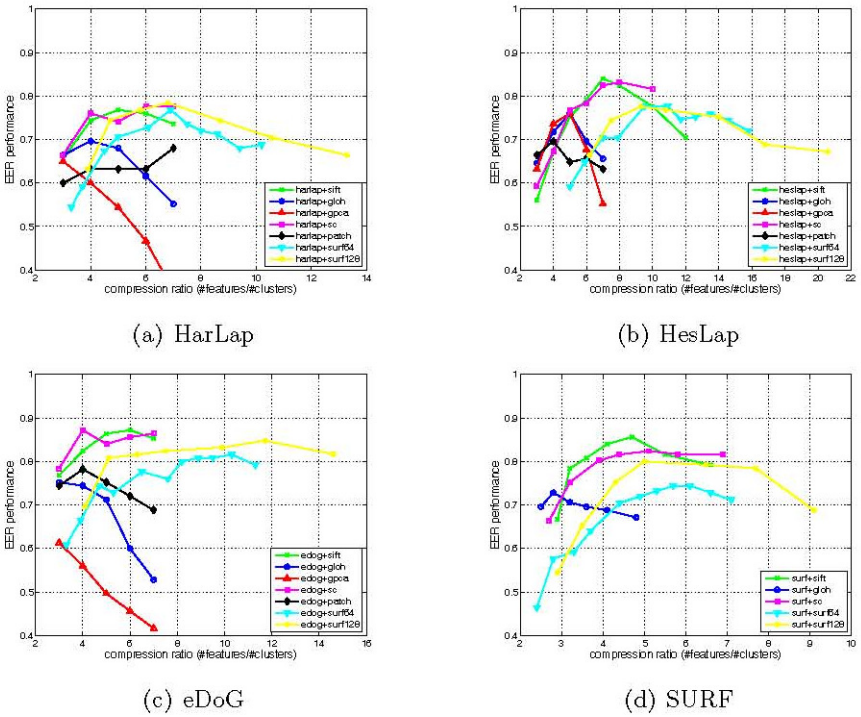



**Figure 10:** Single-cue EER performances for all detector/descriptor combinations on the *TUD motorbikes as a function of the cluster compression ratio (#features/#clusters).*

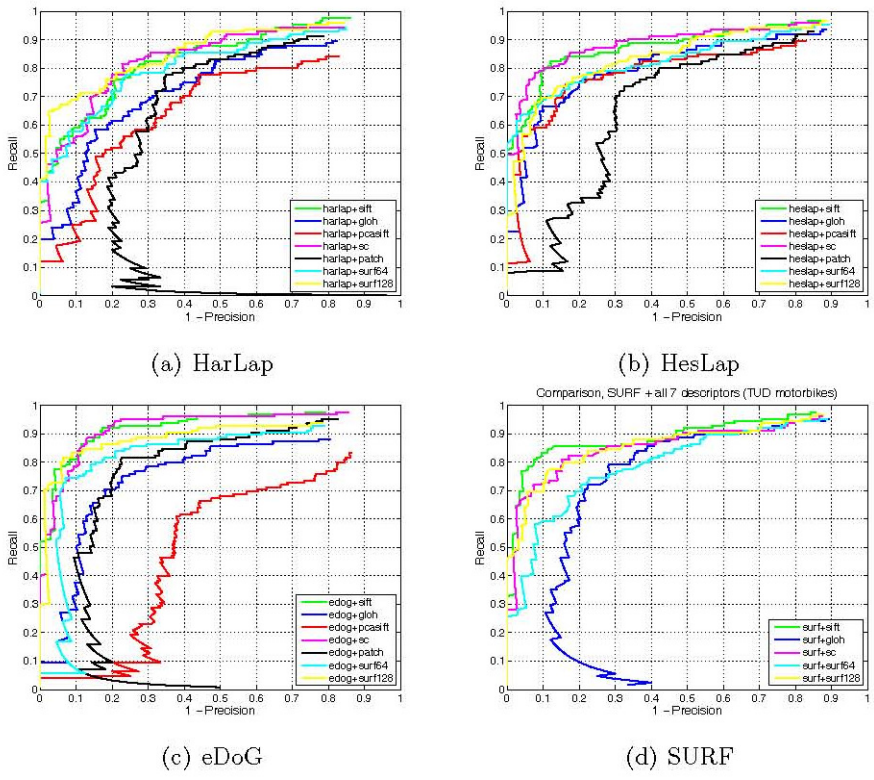



**Figure 11:** Comparison of the different descriptors for all 4 detectors (on the TUD motorbikes, optimal parameter settings).

### *A.3.3 Effect of the Number of Extracted Features*

**Table 1** shows the number of interest regions the four detectors used in our evaluation extracted on the motorbike training set. This comparison is important, since the performance of the ISM approach generally improves as a function of the number of features it can base its decisions on. It can be seen that the *Hessian-Laplace* and *DoG* detectors find roughly twice as many features as *Harris-Laplace* and *SURF*. The lower recognition performance of *Harris-Laplace* can thus partially be explained by its lower feature count.

Compared to the other interest region detectors, *SURF* however performs very well, especially considering the lower number of regions it delivers (c.f. Figure 8 (a,b,d)). It reaches the same performance level as *Hessian-Laplace* and performs only slightly worse than *DoG*. For this evaluation, we ran all detectors at their default settings, which were typically optimized for exact matching tasks. It can thus be expected that the performance of the SURF detector can still be improved by adaptations that increase the number of interest regions it returns.

**Table 1:** Number of features extracted on the motorbike training set for the different interest region detectors.

| Detector | *Harris-Laplace* | *Hessian-Laplace* | *DoG* | *SURF* |
|---|---|---|---|---|
| # features | 12,547 | 23,379 | 19,221 | 10,128 |

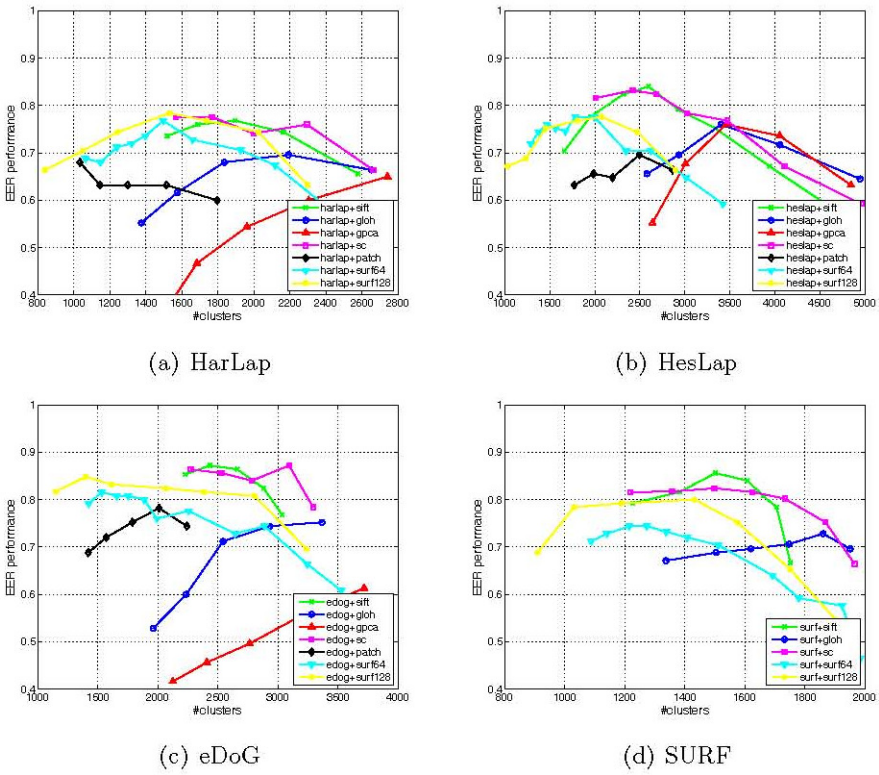



**Figure 12:** Single-cue EER performances for all detector/descriptor combinations on the TUD motorbikes as a function of the number of clusters.

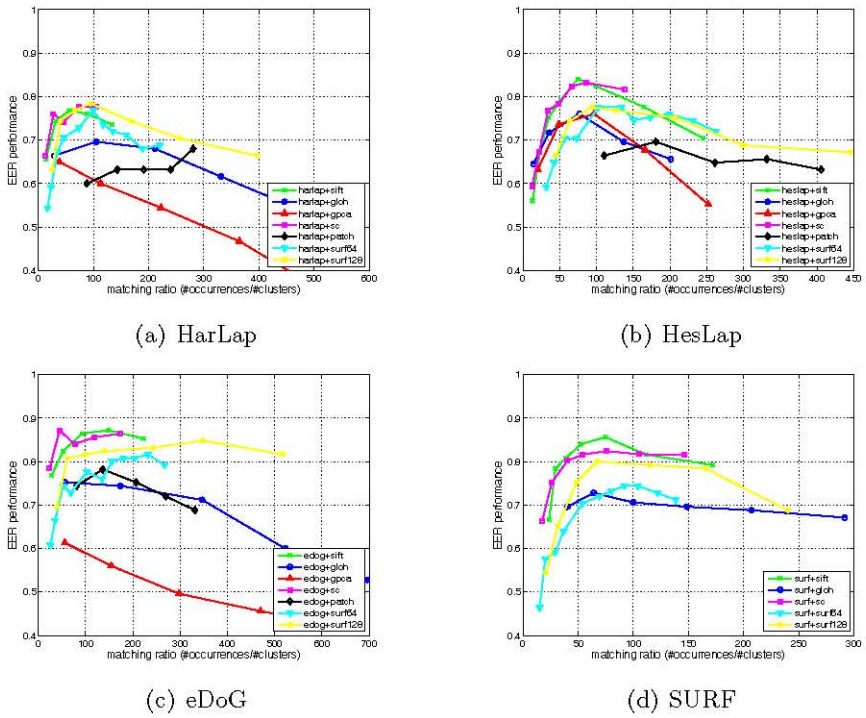
### *A.3.4 Run-Time Considerations*

*SURF* was developed with the goal of fast feature extraction. In this respect, it is clearly superior to all 3 other region detectors. However, in the context of an entire recognition task, total system performance also depends on the other stages of the pipeline.

In the ISM approach, the main other factors influencing recognition run-time are the *number of codebook clusters* the extracted features are matched to and the number of *stored occurrence locations per cluster*. The former determines the effort required for feature matching; the latter is responsible for the complexity of the Hough voting stage. The exact influence of both stages depends on implementation details, but generally speaking, the lower both numbers are, the faster recognition will be.

**Figure 12** and **Figure 13** display the recognition performance as a function of the *codebook size* (*#clusters*) and the *matching ratio* (*#occurrences/#clusters*), respectively. It can be seen that both the *SURF-64* and *SURF-128* descriptors reach their performance optima already at a relatively small codebook size, resulting in low matching costs. Except for their conjunction with the *DoG* detector, both descriptors also require a relatively low matching ratio.

However, in terms of absolute recognition performance, *SURF-128* is still outperformed by *SIFT* and *SC* for three of the four detectors. It reaches its best absolute performance in combination with *DoG* (with 84.8% EER compared to 87.2% for *DoG+SIFT/SC*), but this performance is only achieved at a relatively high matching ratio, indicating a high cost for the voting stage. Together, these results suggest that there is still some potential for improving the *SURF* descriptors for their application to object detection.




(a) HarLap (b) HesLap

(c) eDoG (d) SURF

**Figure 13:** Single-cue EER performances for all detector/descriptor combinations on the TUD motorbikes as a function of the matching ratio (#occurrences/#clusters).

## A.4 Discussion

It is important to also keep in mind the limits of this evaluation. The results analyzed so far have been obtained on a single test set. Small differences between the different detection performances should therefore not be overrated. In order to compare the effects of different parameters, we had to reduce the full performance curves to a single key figure, in our case the performance at the EER. This will naturally introduce additional noise in the evaluation. In (Leibe et al. 2006), the evaluation is therefore extended to two other data sets for a subset of 3 ´ 3=9 detector/descriptor combinations, showing that the results generalize also to other scenarios.

Nevertheless, we can draw several interesting conclusions from the results of our experiments.

- When comparing descriptors across different detectors, we could find a clear performance optimum at a certain clustering/matching similarity for *SIFT*, *GLOH*, *Shape Context*, and *SURF-64/SURF-128*. The cluster compression ratio, on the other hand, did not seem to have a consistent influence. We can therefore formulate the recommendation to use the cluster similarity as a criterion for selecting the clustering level for those descriptors. As our evaluation was based on commonly available feature implementations, our results should be readily transferrable to other researchers.
- For the categories we tested, *SIFT/Shape Context* features seemed to be the best local descriptors, closely followed by *SURF-128*. This result was consistent across all four detectors. Those three descriptors can therefore be seen as almost equivalent, and it is to be expected that their relative ordering will depend mainly on the characteristics of the object category.
- In terms of the interest region detectors, *DoG* performed best in most cases, followed by *Hessian-Laplace* and *SURF*. The good performance of *SURF* is however worth noting, as it is based on a far smaller number of interest regions. It can thus be expected that this result can further be improved by adapting the *SURF* detector to return more regions.

# B Appended Publications

See next page

# SURF: Speeded Up Robust Features

Herbert Bay[1], Tinne Tuytelaars[2], and Luc Van Gool[12]

[1] ETH Zurich
{bay, vangool}@vision.ee.ethz.ch
[2] Katholieke Universiteit Leuven
{Tinne.Tuytelaars, Luc.Vangool}@esat.kuleuven.be

**Abstract.** In this paper, we present a novel scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Features). It approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster.
This is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors (*in casu*, using a Hessian matrix-based measure for the detector, and a distribution-based descriptor); and by simplifying these methods to the essential. This leads to a combination of novel detection, description, and matching steps. The paper presents experimental results on a standard evaluation set, as well as on imagery obtained in the context of a real-life object recognition application. Both show SURF's strong performance.

## 1 Introduction

The task of finding correspondences between two images of the same scene or object is part of many computer vision applications. Camera calibration, 3D reconstruction, image registration, and object recognition are just a few. The search for discrete image correspondences – the goal of this work – can be divided into three main steps. First, 'interest points' are selected at distinctive locations in the image, such as corners, blobs, and T-junctions. The most valuable property of an interest point *detector* is its repeatability, i.e. whether it reliably finds the same interest points under different viewing conditions. Next, the neighbourhood of every interest point is represented by a feature vector. This *descriptor* has to be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Finally, the descriptor vectors are *matched* between different images. The matching is often based on a distance between the vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this takes, and a lower number of dimensions is therefore desirable.

It has been our goal to develop both a detector and descriptor, which in comparison to the state-of-the-art are faster to compute, while not sacrificing performance. In order to succeed, one has to strike a balance between the above requirements, like reducing the descriptor's dimension and complexity, while keeping it sufficiently distinctive.

A wide variety of detectors and descriptors have already been proposed in the literature (e.g. [1–6]). Also, detailed comparisons and evaluations on benchmarking datasets have been performed [7–9]. While constructing our fast detector and descriptor, we built on the insights gained from this previous work in order to get a feel for what are the aspects contributing to performance. In our experiments on benchmark image sets as well as on a real object recognition application, the resulting detector and descriptor are not only faster, but also more distinctive and equally repeatable.

When working with local features, a first issue that needs to be settled is the required level of invariance. Clearly, this depends on the expected geometric and photometric deformations, which in turn are determined by the possible changes in viewing conditions. Here, we focus on scale and image rotation invariant detectors and descriptors. These seem to offer a good compromise between feature complexity and robustness to commonly occurring deformations. Skew, anisotropic scaling, and perspective effects are assumed to be second-order effects, that are covered to some degree by the overall robustness of the descriptor. As also claimed by Lowe [2], the additional complexity of full affine-invariant features often has a negative impact on their robustness and does not pay off, unless really large viewpoint changes are to be expected. In some cases, even rotation invariance can be left out, resulting in a scale-invariant only version of our descriptor, which we refer to as 'upright SURF' (U-SURF). Indeed, in quite a few applications, like mobile robot navigation or visual tourist guiding, the camera often only rotates about the vertical axis. The benefit of avoiding the overkill of rotation invariance in such cases is not only increased speed, but also increased discriminative power. Concerning the photometric deformations, we assume a simple linear model with a scale factor and offset. Notice that our detector and descriptor don't use colour.

The paper is organised as follows. Section 2 describes related work, on which our results are founded. Section 3 describes the interest point detection scheme. In section 4, the new descriptor is presented. Finally, section 5 shows the experimental results and section 6 concludes the paper.

## 2 Related Work

*Interest Point Detectors* The most widely used detector probably is the Harris corner detector [10], proposed back in 1988, based on the eigenvalues of the second-moment matrix. However, Harris corners are not scale-invariant. Lindeberg introduced the concept of automatic scale selection [1]. This allows to detect interest points in an image, each with their own characteristic scale. He experimented with both the determinant of the Hessian matrix as well as the Laplacian (which corresponds to the trace of the Hessian matrix) to detect blob-like structures. Mikolajczyk and Schmid refined this method, creating robust and scale-invariant feature detectors with high repeatability, which they coined Harris-Laplace and Hessian-Laplace [11]. They used a (scale-adapted) Harris measure or the determinant of the Hessian matrix to select the location, and the

Laplacian to select the scale. Focusing on speed, Lowe [12] approximated the Laplacian of Gaussian (LoG) by a Difference of Gaussians (DoG) filter.

Several other scale-invariant interest point detectors have been proposed. Examples are the salient region detector proposed by Kadir and Brady [13], which maximises the entropy within the region, and the edge-based region detector proposed by Jurie *et al.* [14]. They seem less amenable to acceleration though. Also, several affine-invariant feature detectors have been proposed that can cope with longer viewpoint changes. However, these fall outside the scope of this paper.

By studying the existing detectors and from published comparisons [15, 8], we can conclude that (1) Hessian-based detectors are more stable and repeatable than their Harris-based counterparts. Using the determinant of the Hessian matrix rather than its trace (the Laplacian) seems advantageous, as it fires less on elongated, ill-localised structures. Also, (2) approximations like the DoG can bring speed at a low cost in terms of lost accuracy.

*Feature Descriptors* An even larger variety of feature descriptors has been proposed, like Gaussian derivatives [16], moment invariants [17], complex features [18, 19], steerable filters [20], phase-based local features [21], and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. The latter, introduced by Lowe [2], have been shown to outperform the others [7]. This can be explained by the fact that they capture a substantial amount of information about the spatial intensity patterns, while at the same time being robust to small deformations or localisation errors. The descriptor in [2], called SIFT for short, computes a histogram of local oriented gradients around the interest point and stores the bins in a 128-dimensional vector (8 orientation bins for each of the $4 \times 4$ location bins).

Various refinements on this basic scheme have been proposed. Ke and Sukthankar [4] applied PCA on the gradient image. This PCA-SIFT yields a 36-dimensional descriptor which is fast for matching, but proved to be less distinctive than SIFT in a second comparative study by Mikolajczyk *et al.* [8] and slower feature computation reduces the effect of fast matching. In the same paper [8], the authors have proposed a variant of SIFT, called GLOH, which proved to be even more distinctive with the same number of dimensions. However, GLOH is computationally more expensive.

The SIFT descriptor still seems to be the most appealing descriptor for practical uses, and hence also the most widely used nowadays. It is distinctive *and* relatively fast, which is crucial for on-line applications. Recently, Se *et al.* [22] implemented SIFT on a Field Programmable Gate Array (FPGA) and improved its speed by an order of magnitude. However, the high dimensionality of the descriptor is a drawback of SIFT at the matching step. For on-line applications on a regular PC, each one of the three steps (detection, description, matching) should be faster still. Lowe proposed a best-bin-first alternative [2] in order to speed up the matching step, but this results in lower accuracy.

*Our approach* In this paper, we propose a novel detector-descriptor scheme, coined SURF (Speeded-Up Robust Features). The detector is based on the Hes-

sian matrix [11, 1], but uses a very basic approximation, just as DoG [2] is a very basic Laplacian-based detector. It relies on integral images to reduce the computation time and we therefore call it the 'Fast-Hessian' detector. The descriptor, on the other hand, describes a distribution of Haar-wavelet responses within the interest point neighbourhood. Again, we exploit integral images for speed. Moreover, only 64 dimensions are used, reducing the time for feature computation and matching, and increasing simultaneously the robustness. We also present a new indexing step based on the sign of the Laplacian, which increases not only the matching speed, but also the robustness of the descriptor.

In order to make the paper more self-contained, we succinctly discuss the concept of integral images, as defined by [23]. They allow for the fast implementation of box type convolution filters. The entry of an integral image $I_\Sigma(\mathbf{x})$ at a location $\mathbf{x} = (x, y)$ represents the sum of all pixels in the input image $I$ of a rectangular region formed by the point $\mathbf{x}$ and the origin, $I_\Sigma(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j)$. With $I_\Sigma$ calculated, it only takes four additions to calculate the sum of the intensities over any upright, rectangular area, independent of its size.

## 3 Fast-Hessian Detector

We base our detector on the Hessian matrix because of its good performance in computation time and accuracy. However, rather than using a different measure for selecting the location and the scale (as was done in the Hessian-Laplace detector [11]), we rely on the determinant of the Hessian for both. Given a point $\mathbf{x} = (x, y)$ in an image $I$, the Hessian matrix $\mathcal{H}(\mathbf{x}, \sigma)$ in $\mathbf{x}$ at scale $\sigma$ is defined as follows

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \tag{1}$$

where $L_{xx}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image $I$ in point $\mathbf{x}$, and similarly for $L_{xy}(\mathbf{x}, \sigma)$ and $L_{yy}(\mathbf{x}, \sigma)$.

Gaussians are optimal for scale-space analysis, as shown in [24]. In practice, however, the Gaussian needs to be discretised and cropped (Fig. 1 left half), and even with Gaussian filters aliasing still occurs as soon as the resulting images are sub-sampled. Also, the property that no new structures can appear while going to lower resolutions may have been proven in the 1D case, but is known to not apply in the relevant 2D case [25]. Hence, the importance of the Gaussian seems to have been somewhat overrated in this regard, and here we test a simpler alternative. As Gaussian filters are non-ideal in any case, and given Lowe's success with LoG approximations, we push the approximation even further with box filters (Fig. 1 right half). These approximate second order Gaussian derivatives, and can be evaluated very fast using integral images, independently of size. As shown in the results section, the performance is comparable to the one using the discretised and cropped Gaussians.

The $9 \times 9$ box filters in Fig. 1 are approximations for Gaussian second order derivatives with $\sigma = 1.2$ and represent our lowest scale (i.e. highest spatial resolution). We denote our approximations by $D_{xx}$, $D_{yy}$, and $D_{xy}$. The weights

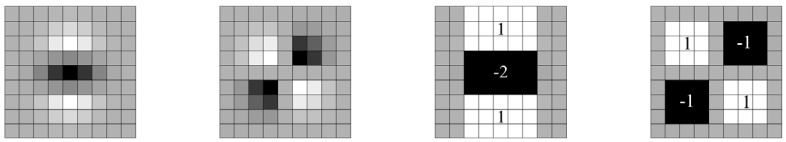



**Fig. 1.** Left to right: the (discretised and cropped) Gaussian second order partial derivatives in $y$-direction and $xy$-direction, and our approximations thereof using box filters. The grey regions are equal to zero.

applied to the rectangular regions are kept simple for computational efficiency, but we need to further balance the relative weights in the expression for the Hessian's determinant with $\frac{|L_{xy}(1.2)|_F|D_{xx}(9)|_F}{|L_{xx}(1.2)|_F|D_{xy}(9)|_F} = 0.912... \simeq 0.9$, where $|x|_F$ is the Frobenius norm. This yields

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2. \tag{2}$$

Furthermore, the filter responses are normalised with respect to the mask size. This guarantees a constant Frobenius norm for any filter size.

Scale spaces are usually implemented as image pyramids. The images are repeatedly smoothed with a Gaussian and subsequently sub-sampled in order to achieve a higher level of the pyramid. Due to the use of box filters and integral images, we do not have to iteratively apply the same filter to the output of a previously filtered layer, but instead can apply such filters of any size at exactly the same speed directly on the original image, and even in parallel (although the latter is not exploited here). Therefore, the scale space is analysed by up-scaling the filter size rather than iteratively reducing the image size. The output of the above $9 \times 9$ filter is considered as the initial scale layer, to which we will refer as scale $s = 1.2$ (corresponding to Gaussian derivatives with $\sigma = 1.2$). The following layers are obtained by filtering the image with gradually bigger masks, taking into account the discrete nature of integral images and the specific structure of our filters. Specifically, this results in filters of size $9 \times 9$, $15 \times 15$, $21 \times 21$, $27 \times 27$, etc. At larger scales, the step between consecutive filter sizes should also scale accordingly. Hence, for each new octave, the filter size increase is doubled (going from 6 to 12 to 24). Simultaneously, the sampling intervals for the extraction of the interest points can be doubled as well.

As the ratios of our filter layout remain constant after scaling, the approximated Gaussian derivatives scale accordingly. Thus, for example, our $27 \times 27$ filter corresponds to $\sigma = 3 \times 1.2 = 3.6 = s$. Furthermore, as the Frobenius norm remains constant for our filters, they are already scale normalised [26].

In order to localise interest points in the image and over scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood is applied. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Brown *et al.* [27]. Scale space interpolation is especially important in our case, as the difference in scale between

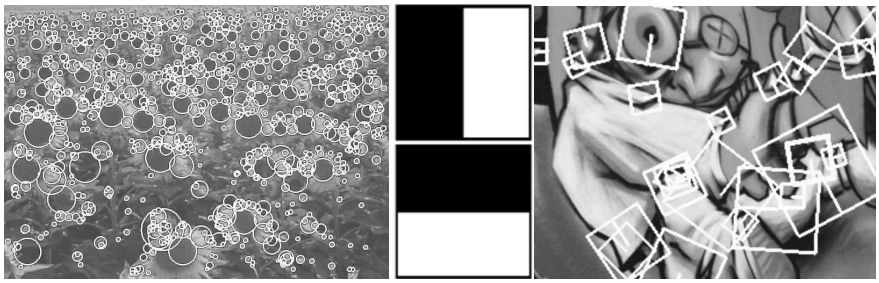


**Fig. 2.** Left: Detected interest points for a Sunflower field. This kind of scenes shows clearly the nature of the features from Hessian-based detectors. Middle: Haar wavelet types used for SURF. Right: Detail of the Graffiti scene showing the size of the descriptor window at different scales.

the first layers of every octave is relatively large. Fig. 2 (left) shows an example of the detected interest points using our 'Fast-Hessian' detector.

## 4 SURF Descriptor

The good performance of SIFT compared to other descriptors [8] is remarkable. Its mixing of crudely localised information and the distribution of gradient related features seems to yield good distinctive power while fending off the effects of localisation errors in terms of scale or space. Using relative strengths and orientations of gradients reduces the effect of photometric changes.

The proposed SURF descriptor is based on similar properties, with a complexity stripped down even further. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then, we construct a square region aligned to the selected orientation, and extract the SURF descriptor from it. These two steps are now explained in turn. Furthermore, we also propose an upright version of our descriptor (U-SURF) that is not invariant to image rotation and therefore faster to compute and better suited for applications where the camera remains more or less horizontal.

### 4.1 Orientation Assignment

In order to be invariant to rotation, we identify a reproducible orientation for the interest points. For that purpose, we first calculate the Haar-wavelet responses in $x$ and $y$ direction, shown in Fig. 2, and this in a circular neighbourhood of radius $6s$ around the interest point, with $s$ the scale at which the interest point was detected. Also the sampling step is scale dependent and chosen to be $s$. In keeping with the rest, also the wavelet responses are computed at that current scale $s$. Accordingly, at high scales the size of the wavelets is big. Therefore, we use again integral images for fast filtering. Only six operations are needed to

compute the response in $x$ or $y$ direction at any scale. The side length of the wavelets is $4s$.

Once the wavelet responses are calculated and weighted with a Gaussian ($\sigma = 2.5s$) centered at the interest point, the responses are represented as vectors in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of $\frac{\pi}{3}$. The horizontal and vertical responses within the window are summed. The two summed responses then yield a new vector. The longest such vector lends its orientation to the interest point. The size of the sliding window is a parameter, which has been chosen experimentally. Small sizes fire on single dominating wavelet responses, large sizes yield maxima in vector length that are not outspoken. Both result in an unstable orientation of the interest region. Note the U-SURF skips this step.

### 4.2 Descriptor Components

For the extraction of the descriptor, the first step consists of constructing a square region centered around the interest point, and oriented along the orientation selected in the previous section. For the upright version, this transformation is not necessary. The size of this window is $20s$. Examples of such square regions are illustrated in Fig. 2.

The region is split up regularly into smaller $4 \times 4$ square sub-regions. This keeps important spatial information in. For each sub-region, we compute a few simple features at $5 \times 5$ regularly spaced sample points. For reasons of simplicity, we call $d_x$ the Haar wavelet response in horizontal direction and $d_y$ the Haar wavelet response in vertical direction (filter size $2s$). "Horizontal" and "vertical" here is defined in relation to the selected interest point orientation. To increase the robustness towards geometric deformations and localisation errors, the responses $d_x$ and $d_y$ are first weighted with a Gaussian ($\sigma = 3.3s$) centered at the interest point.

Then, the wavelet responses $d_x$ and $d_y$ are summed up over each subregion and form a first set of entries to the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, $|d_x|$ and $|d_y|$. Hence, each sub-region has a four-dimensional descriptor vector $\mathbf{v}$ for its underlying intensity structure $\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. This results in a descriptor vector for all $4 \times 4$ sub-regions of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

Fig. 3 shows the properties of the descriptor for three distinctively different image intensity patterns within a subregion. One can imagine combinations of such local intensity patterns, resulting in a distinctive descriptor.

In order to arrive at these SURF descriptors, we experimented with fewer and more wavelet features, using $d_x^2$ and $d_y^2$, higher-order wavelets, PCA, median values, average values, etc. From a thorough evaluation, the proposed sets turned

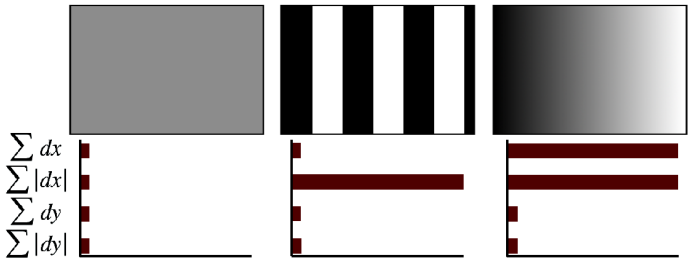



**Fig. 3.** The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in $x$ direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in $x$ direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

out to perform best. We then varied the number of sample points and sub-regions. The $4 \times 4$ sub-region division solution provided the best results. Considering finer subdivisions appeared to be less robust and would increase matching times too much. On the other hand, the short descriptor with $3 \times 3$ subregions (SURF-36) performs worse, but allows for very fast matching and is still quite acceptable in comparison to other descriptors in the literature. Fig. 4 shows only a few of these comparison results (SURF-128 will be explained shortly).

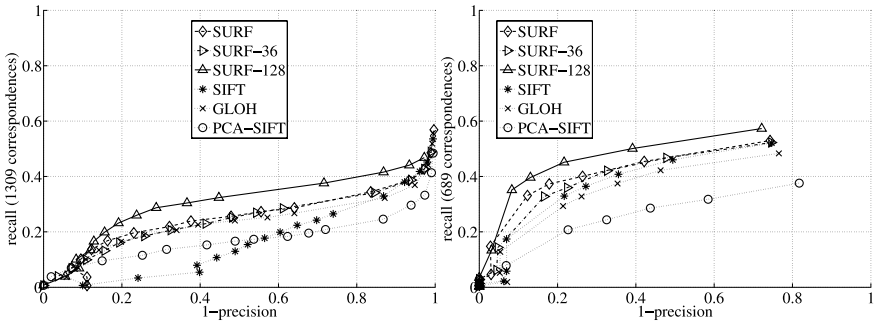



**Fig. 4.** The *recall* vs. *(1-precision)* graph for different binning methods and two different matching strategies tested on the 'Graffiti' sequence (image 1 and 3) with a view change of 30 degrees, compared to the current descriptors. The interest points are computed with our 'Fast Hessian' detector. Note that the interest points are not affine invariant. The results are therefore not comparable to the ones in [8]. SURF-128 corresponds to the extended descriptor. Left: Similarity-threshold-based matching strategy. Right: Nearest-neighbour-ratio matching strategy (See section 5).

We also tested an alternative version of the SURF descriptor that adds a couple of similar features (SURF-128). It again uses the same sums as before, but now splits these values up further. The sums of $d_x$ and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of $d_y$ and $|d_y|$ are split up according to the sign of $d_x$, thereby doubling the number of features. The descriptor is more distinctive and not much slower to compute, but slower to match due to its higher dimensionality.

In Figure 4, the parameter choices are compared for the standard 'Graffiti' scene, which is the most challenging of all the scenes in the evaluation set of Mikolajczyk [8], as it contains out-of-plane rotation, in-plane rotation as well as brightness changes. The extended descriptor for $4 \times 4$ subregions (SURF-128) comes out to perform best. Also, SURF performs well and is faster to handle. Both outperform the existing state-of-the-art.

For fast indexing during the matching stage, the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest point is included. Typically, the interest points are found at blob-type structures. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no extra computational cost, as it was already computed during the detection phase. In the matching stage, we only compare features if they have the same type of contrast. Hence, this minimal information allows for faster matching and gives a slight increase in performance.

## 5 Experimental Results

First, we present results on a standard evaluation set, fot both the detector and the descriptor. Next, we discuss results obtained in a real-life object recognition application. All detectors and descriptors in the comparison are based on the original implementations of authors.

*Standard Evaluation* We tested our detector and descriptor using the image sequences and testing software provided by Mikolajczyk [3]. These are images of real textured and structured scenes. Due to space limitations, we cannot show the results on all sequences. For the detector comparison, we selected the two viewpoint changes (Graffiti and Wall), one zoom and rotation (Boat) and lighting changes (Leuven) (see Fig. 6, discussed below). The descriptor evaluations are shown for all sequences except the Bark sequence (see Fig. 4 and 7).

For the detectors, we use the repeatability score, as described in [9]. This indicates how many of the detected interest points are found in both images, relative to the lowest total number of interest points found (where only the part of the image that is visible in both images is taken into account).

The detector is compared to the difference of Gaussian (DoG) detector by Lowe [2], and the Harris- and Hessian-Laplace detectors proposed by Mikolajczyk [15]. The number of interest points found is on average very similar for all

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

detectors. This holds for all images, including those from the database used in the object recognition experiment, see Table 1 for an example. As can be seen our 'Fast-Hessian' detector is more than 3 times faster that DoG and 5 times faster than Hessian-Laplace. At the same time, the repeatability for our detector is comparable (Graffiti, Leuven, Boats) or even better (Wall) than for the competitors. Note that the sequences Graffiti and Wall contain out-of-plane rotation, resulting in affine deformations, while the detectors in the comparison are only rotation- and scale invariant. Hence, these deformations have to be tackled by the overall robustness of the features.

The descriptors are evaluated using recall-(1-precision) graphs, as in [4] and [8]. For each evaluation, we used the first and the fourth image of the sequence, except for the Graffiti (image 1 and 3) and the Wall scene (image 1 and 5), corresponding to a viewpoint change of 30 and 50 degrees, respectively. In figures 4 and 7, we compared our SURF descriptor to GLOH, SIFT and PCA-SIFT, based on interest points detected with our 'Fast-Hessian' detector. SURF outperformed the other descriptors for almost all the comparisons. In Fig. 4, we compared the results using two different matching techniques, one based on the similarity threshold and one based on the nearest neighbour ratio (see [8] for a discussion on these techniques). This has an effect on the ranking of the descriptors, yet SURF performed best in both cases. Due to space limitations, only results on similarity threshold based matching are shown in Fig. 7, as this technique is better suited to represent the distribution of the descriptor in its feature space [8] and it is in more general use.

The SURF descriptor outperforms the other descriptors in a systematic and significant way, with sometimes more than 10% improvement in recall for the same level of precision. At the same time, it is fast to compute (see Table 2). The accurate version (SURF-128), presented in section 4, showed slightly better results than the regular SURF, but is slower to match and therefore less interesting for speed-dependent applications.

Note that throughout the paper, including the object recognition experiment, we always use the same set of parameters and thresholds (see table 1). The timings were evaluated on a standard Linux PC (Pentium IV, 3GHz).

*Object Recognition* We also tested the new features on a practical application, aimed at recognising objects of art in a museum. The database consists of 216 images of 22 objects. The images of the test set (116 images) were taken un-

| detector | threshold | nb of points | comp. time (msec) |
|---|---|---|---|
| Fast-Hessian | 600 | 1418 | 120 |
| Hessian-Laplace | 1000 | 1979 | 650 |
| Harris-Laplace | 2500 | 1664 | 1800 |
| DoG | default | 1520 | 400 |

**Table 1.** Thresholds, number of detected points and calculation time for the detectors in our comparison. (First image of Graffiti scene, $800 \times 640$)

| | U-SURF | SURF | SURF-128 | SIFT |
|---|---|---|---|---|
| time (ms): | 255 | 354 | 391 | 1036 |

**Table 2.** Computation times for the joint detector - descriptor implementations, tested on the first image of the Graffiti sequence. The thresholds are adapted in order to detect the same number of interest points for all methods. These relative speeds are also representative for other images.

der various conditions, including extreme lighting changes, objects in reflecting glass cabinets, viewpoint changes, zoom, different camera qualities, etc. Moreover, the images are small ($320 \times 240$) and therefore more challenging for object recognition, as many details get lost.

In order to recognise the objects from the database, we proceed as follows. The images in the test set are compared to all images in the reference set by matching their respective interest points. The object shown on the reference image with the highest number of matches with respect to the test image is chosen as the recognised object.

The matching is carried out as follows. An interest point in the test image is compared to an interest point in the reference image by calculating the Euclidean distance between their descriptor vectors. A matching pair is detected, if its distance is closer than 0.7 times the distance of the second nearest neighbour. This is the nearest neighbour ratio matching strategy [18, 2, 7]. Obviously, additional geometric constraints reduce the impact of false positive matches, yet this can be done on top of any matcher. For comparing reasons, this does not make sense, as these may hide shortcomings of the basic schemes. The average recognition rates reflect the results of our performance evaluation. The leader is SURF-128 with 85.7% recognition rate, followed by U-SURF (83.8%) and SURF (82.6%). The other descriptors achieve 78.3% (GLOH), 78.1% (SIFT) and 72.3% (PCA-SIFT).

**Fig. 5.** An example image from the reference set (left) and the test set (right). Note the difference in viewpoint and colours.

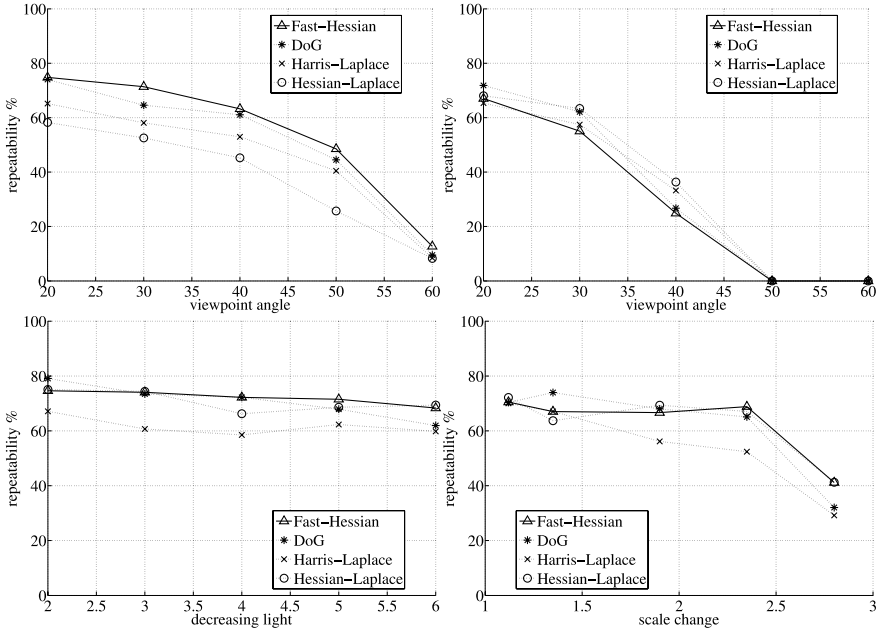



**Fig. 6.** Repeatability score for image sequences, from left to right and top to bottom, Wall and Graffiti (Viewpoint Change), Leuven (Lighting Change) and Boat (Zoom and Rotation).

## 6 Conclusion

We have presented a fast and performant interest point detection-description scheme which outperforms the current state-of-the art, both in speed and accuracy. The descriptor is easily extendable for the description of affine invariant regions. Future work will aim at optimising the code for additional speed up. A binary of the latest version is available on the internet[4].

**Acknowledgements:** The authors gratefully acknowledge the support from Swiss SNF NCCR project IM2, Toyota-TME and the Flemish Fund for Scientific Research.

## References

1. Lindeberg, T.: Feature detection with automatic scale selection. IJCV **30(2)** (1998) 79 – 116
2. Lowe, D.: Distinctive image features from scale-invariant keypoints, cascade filtering approach. IJCV **60** (2004) 91 – 110

[4] http://www.vision.ee.ethz.ch/~surf/

3. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: ECCV. (2002) 128 – 142
4. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR (2). (2004) 506 – 513
5. Tuytelaars, T., Van Gool, L.: Wide baseline stereo based on local, affinely invariant regions. In: BMVC. (2000) 412 – 422
6. Matas, J., Chum, O., M., U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. (2002) 384 – 393
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: CVPR. Volume 2. (2003) 257 – 263
8. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27** (2005) 1615–1630
9. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV **65** (2005) 43–72
10. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference. (1988) 147 – 151
11. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: ICCV. Volume 1. (2001) 525 – 531
12. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV. (1999)
13. Kadir, T., Brady, M.: Scale, saliency and image description. IJCV **45(2)** (2001) 83 – 105
14. Jurie, F., Schmid, C.: Scale-invariant shape features for recognition of object categories. In: CVPR. Volume II. (2004) 90 – 96
15. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV **60** (2004) 63 – 86
16. Florack, L.M.J., Haar Romeny, B.M.t., Koenderink, J.J., Viergever, M.A.: General intensity transformations and differential invariants. JMIV **4** (1994) 171–187
17. Mindru, F., Tuytelaars, T., Van Gool, L., Moons, T.: Moment invariants for recognition under changing viewpoint and illumination. CVIU **94** (2004) 3–27
18. Baumberg, A.: Reliable feature matching across widely separated views. In: CVPR. (2000) 774 – 781
19. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In: ECCV. Volume 1. (2002) 414 – 431
20. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. PAMI **13** (1991) 891 – 906
21. Carneiro, G., Jepson, A.: Multi-scale phase-based local features. In: CVPR (1). (2003) 736 – 743
22. Se, S., Ng, H., Jasiobedzki, P., Moyung, T.: Vision based modeling and localization for planetary exploration rovers. Proceedings of International Astronautical Congress (2004)
23. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (1). (2001) 511 – 518
24. Koenderink, J.: The structure of images. Biological Cybernetics **50** (1984) 363 – 370
25. Lindeberg, T.: Discrete Scale-Space Theory and the Scale-Space Primal Sketch, PhD, KTH Stockholm,. KTH (1991)
26. Lindeberg, T., Bretzner, L.: Real-time scale selection in hybrid multi-scale representations. In: Scale-Space. (2003) 148–163
27. Brown, M., Lowe, D.: Invariant features from interest point groups. In: BMVC. (2002)

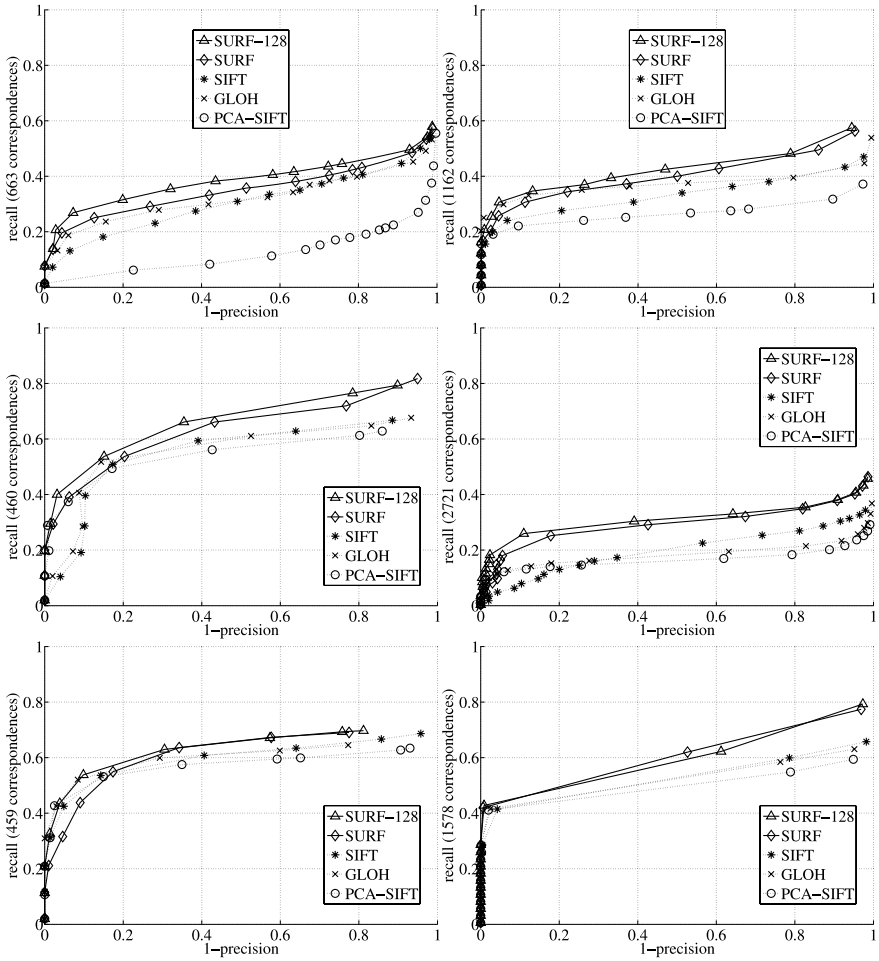



**Fig. 7.** Recall, 1-Precision graphs for, from left to right and top to bottom, Viewpoint change of 50 (Wall) degrees, scale factor 2 (Boat), image blur (Bikes and Trees), brightness change (Leuven) and JPEG compression (Ubc).

# Segmentation Based Multi-Cue Integration for Object Detection

Bastian Leibe
ETH Zurich
Zurich, Switzerland

Krystian Mikolajczyk
University of Surrey
Guildford, UK

Bernt Schiele
TU Darmstadt
Darmstadt, Germany

**Abstract**

This paper proposes a novel method for integrating multiple local cues, i.e. local region detectors as well as descriptors, in the context of object detection. Rather than to fuse the outputs of several distinct classifiers in a fixed setup, our approach implements a highly flexible combination scheme, where the contributions of all individual cues are flexibly recombined depending on their explanatory power for each new test image. The key idea behind our approach is to integrate the cues over an estimated top-down segmentation, which allows to quantify how much each of them contributed to the object hypothesis. By combining those contributions on a per-pixel level, our approach ensures that each cue only contributes to object regions for which it is confident and that potential correlations between cues are effectively factored out. Experimental results on several benchmark data sets show that the proposed multi-cue combination scheme significantly increases detection performance compared to any of its constituent cues alone. Moreover, it provides an interesting evaluation tool to analyze the complementarity of local feature detectors and descriptors.

## 1 Introduction

Local feature based approaches have shown considerable promise for dealing with the large degree of intra-category variation and partial occlusion inherent in real-world categorization and detection tasks. Consequently, many approaches have been developed that use local features in different ways [1, 6, 4, 10, 12], and considerable progress has been made in the design and understanding of the underlying feature detectors and descriptors [12, 14]. Yet, each feature descriptor and detector can only capture part of the information contained in the image, and indeed its value for an application depends on the degree to which it can distill exactly the right kind of information for a specific purpose. As a consequence, the better a descriptor or detector is suited to a specific task, the more likely it is to degenerate when the task conditions deviate too far from its target scenario. In order to be both discriminative and robust, an application should therefore utilize a combination of different local cues.

Several recent studies have evaluated the suitability of various local features in the context of object identification [14] and categorization tasks [13]. However, those studies have only considered each cue in isolation. For multi-cue integration, it is also important to know how the different cues interact, i.e. how correlated their responses are and what new information an additional cue can contribute. However, this information is difficult to retrieve, as different cues are often not directly comparable, both because they typically have different dimensionalities and because they represent information in different ways.

Previous research has therefore mainly focused on *classifier combination*, i.e. on the problem of fusing the outputs of several "black-box" classifiers, possibly with associated confidence ratings [20, 9, 7, 15]. This approach is valid if the classifiers are independent. In our application, however, their outputs are often correlated, and the degree of correlation

may vary from image to image. Rather than just to fuse the outcomes of several classifiers, we therefore need to explore how the underlying information and the respective support in the image can be combined.

In this paper, we present a flexible integration scheme which combines different local cues in an opportunistic manner depending on their explanatory power for the image at hand. The integration proceeds in two steps. First, the sampled features are represented in terms of their similarity to a set of prototypes, an *appearance codebook*, which has been learned for each cue separately. Together with their learned spatial distributions, those codebook prototypes convert the activations from matching features into a probability distribution for possible object locations and scales. This makes the cues comparable. However, their individual responses might still be correlated. Therefore, the second step backprojects the extracted object hypotheses to the image in order to determine for each cue separately which image pixels were responsible for a detection and how much each pixel contributed to the cue's response. By comparing the overlap in their supporting area, our approach can determine the complementarity between two cues and integrate their contributions more robustly.

This paper makes the following three contributions. Firstly, it develops a robust multi-cue integration approach that can be applied regardless of whether the cues are correlated or not. The proposed scheme is directly interpretable and opens up interesting venues for analyzing the complementarity of local cues. Secondly, it presents an extensive evaluation of state-of-the-art region detectors and descriptors in the context of multi-cue integration. The obtained results allow us to rank the cues based on their individual performances and to formulate clear usage guidelines for their combination. Last but not least, experimental results on several challenging data sets show that the proposed multi-cue integration scheme increases object detection performance significantly. The improvement is particularly prominent for the detection precision and leads to high recognition rates at the zero-false-positive level. The paper is structured as follows. The next section discusses related work. Section 2 then reviews the basic recognition approach. Extending this approach, we derive our proposed multi-cue integration scheme in Section 3. Section 4 describes our experimental setup, and Section 5 finally presents the results of our evaluation.

**Related Work.** Many authors have stressed the need for integrating multiple global or local cues in order to increase robustness of recognition [18, 11, 7]. In practice, multi-cue systems for object recognition have often been implemented by combining classifiers [20, 9, 7] or by using cue confidences in a voting scheme [3, 15]. However, these approaches are often static in that they use a fixed confidence rating per cue, e.g. based on previously observed performance. As such, they cannot readily adapt to novel settings when a cue's performance characteristics degrade due to changed environmental conditions. It has therefore been argued that cue weights should be adapted dynamically [17]. For tracking scenarios, cue integration techniques have been proposed which combine cues probabilistically based on their estimated likelihood [19]. However, in the context of single-frame object detection, no such mechanism has been known. In this paper, we propose such a mechanism based on the top-down segmentation approach by [10].

# 2 Recognition Approach

Our multi-cue recognition approach closely builds upon the Implicit Shape Model (ISM) formalism by [10, 11], which combines object detection and top-down segmentation capabilities. This model represents an object category by a set of local appearance clusters (a *codebook*) and their spatial occurrence distributions. Since a basic knowledge of this approach is necessary to understand our method, we will briefly review its main components.

**Training.** For training, local features are extracted from the training images and clustered to form the codebook [1, 10]. In a second run over the training data, the spatial occurrence distributions are estimated by recording for each codebook entry all matching locations on the training objects. Together with each occurrence, the approach stores a local segmentation mask, which is later used for inferring top-down segmentations.

**ISM Recognition.** During recognition, local features are extracted from the image and matched to the codebook. Each matching codebook entry then casts votes for possible object locations and scales in a probabilistic extension of the Hough transform [10]. For each hypothesis, the approach then computes a top-down segmentation and finally selects the subset of hypotheses that best explain the image content under the constraint that each pixel can be assigned to at most one hypothesis.

# 3 Multi-Cue Integration

We now present our novel approach for integrating multiple local cues. In the context of this paper, we understand this as a combination of different local descriptors, but also of different region detectors, since their preference for certain image structures influences the characteristics of the sampled information. As already mentioned before, the question how to combine local cues has no obvious answer, since they are typically not directly comparable.

We therefore proceed in two stages. The first stage extends the recognition procedure to include multiple cues. Its main purpose is to express the cues on a common basis, so that their information can be pooled and initial object hypotheses can be found. This stage still ignores cue correlation. Indeed, it has no other choice, since correlation can only be measured relative to a reference hypothesis, and hypotheses are only available after the stage has been executed. However, the second stage then reveals the correlation by backprojecting hypotheses to the image and computing a top-down segmentation for each cue. This step extends the ISM segmentation algorithm to deal with multiple cues. The obtained segmentations show on a per-pixel level which image structures were responsible for a cue's response. The correlation between two cues can then be expressed as the overlap of their respective $p(\textit{figure})$ probability maps. Once the cue correlation has been identified, the next question is how to use this information to improve recognition performance. In the last part of this section, we present three combination criteria that relate to different strategies for this step.

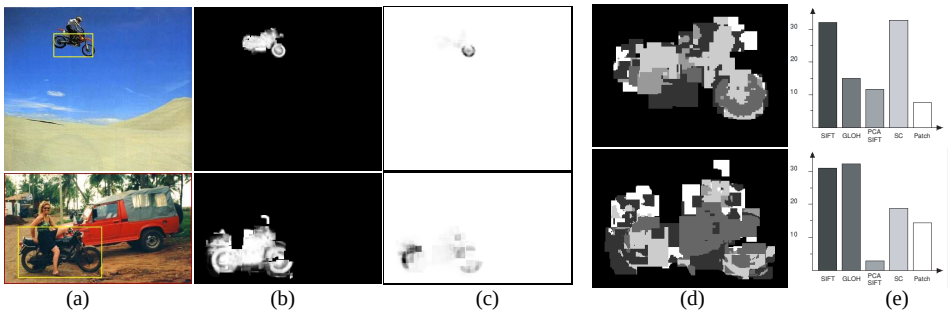
**Initial Recognition Stage.** The key to integrating multiple local cues is to express them on a common basis. We create such a basis by representing sampled features through their similarity to stored prototypes. We therefore extend the recognition approach by keeping a separate codebook $\mathcal{C}^q$ for every cue $q$. Let $\mathbf{e}$ be a local descriptor computed at location $\ell$. When matched to the codebook, it may activate several codebook entries $\mathcal{C}_i^q$ with probabilities $p(\mathcal{C}_i^q|\mathbf{e})$. Each matched codebook entry then votes for instances of the object category $o_n$ at different locations and scales $\lambda = (\lambda_x, \lambda_y, \lambda_\sigma)$ according to its learned occurrence distribution $P(o_n, \lambda|\mathcal{C}_i^q, \ell, q)$. A feature's contribution to an object hypothesis can thus be expressed as

$$p(o_n, \lambda|\mathbf{e}, \ell, q) = \sum_i P(o_n, \lambda|\mathcal{C}_i^q, \ell, q) p(\mathcal{C}_i^q|\mathbf{e}). \quad (1)$$

The contributions from all cues are pooled in a shared 3-dimensional voting space, from which maxima are extracted by Mean Shift Mode Estimation using a scale-adaptive kernel $K$ [11], marginalizing over the cues $q_m$

$$\hat{p}(o_n, \lambda) = \frac{1}{nb(\lambda)^3} \sum_m \sum_k \sum_j p(o_n, \lambda_j|\mathbf{e}_k, \ell_k, q_m) K\left(\frac{\lambda - \lambda_j}{b(\lambda)}\right) p(\mathbf{e}_k, \ell_k|q_m) p(q_m), \quad (2)$$

where $b(\lambda)$ is the scale-adaptive kernel bandwidth; $p(\mathbf{e}_k, \ell_k|q_m)$ is an indicator variable specifying which image patches and locations have been sampled for $q_m$; and $p(q_m)$ is a prior




(a) (b) (c) (d) (e)

**Figure 1:** *Visualization of the multi-cue integration stages: (a) initial detection, (b) top-down segmentation, (c) $p(figure)$ maps obtained by average combination, (d) closeup view of the* argmax *visualization (cf. eq.(9)) , (e) histogram of relative cue contributions.*

determining how much this cue can be trusted. This prior can be set to reflect previously observed performance. In order to avoid any bias, however, we leave it at a uniform setting.

**Multi-Cue Segmentation.** Once a hypothesis $h = (o_n, \lambda)$ has been found, its top-down segmentation can be inferred by backprojecting the supporting votes to the image and combining them with the local patch segmentation masks $p(\mathbf{p} = fig. | o_n, \lambda, c_i^q, \ell)$ that have been stored for each recorded codebook occurrence during training. As shown in [10], the per-pixel probabilities of each pixel containing *figure* or *ground* can then be obtained by a double marginalization, first over sampled features, then over codebook entries. We adapt this formulation here to compute a separate segmentation for each cue

$$p(\mathbf{p} = fig. | o_n, \lambda, q) = \sum_{\mathbf{p} \in (\mathbf{e}, \ell)} \sum_i p(\mathbf{p} = fig. | o_n, \lambda, \mathbf{e}, c_i^q, \ell, q) p(\mathbf{e}, c_i^q, \ell, q | o_n, \lambda) \quad (3)$$

$$= \sum_{\mathbf{p} \in (\mathbf{e}, \ell)} \sum_i p(\mathbf{p} = fig. | o_n, \lambda, c_i^q, \ell) \frac{p(o_n, \lambda | c_i^q, \ell, q) p(c_i^q | \mathbf{e}) p(\mathbf{e}, \ell)}{p(o_n, \lambda)} \quad (4)$$

Based on these results, the final segmentation is computed by building the likelihood ratio between *figure* and *ground* probabilities.

**Segmentation-Based Cue Combination.** Now we can proceed to combining the contributions of different cues on the pixel level. For this, we adopt the idea of formulating hypothesis selection as a Quadratic Boolean Optimization Problem in an MDL framework [11]. Each hypothesis is evaluated in terms of the *savings* that can be obtained in the description of an image by explaining part of it by $h$. The savings of each hypothesis are expressed as

$$S_h = -\kappa_1 + (1 - \kappa_2) \frac{N}{A_\sigma} + \kappa_2 \frac{1}{A_\sigma} \sum_{\mathbf{p} \in Seg(h)} f(\mathbf{p}, h, Q) \quad (5)$$

where $N$ is the number of pixels that can be explained by $h$, $A_\sigma$ is its *expected area* at scale $\sigma$, $\kappa_2$ is a weighting factor to balance out the influence of a hypothesis's area versus its support in the image (left at a fixed value in our experiments), and $\kappa_1$ is the parameter over which the final performance curves are plotted. If multiple hypotheses overlap, their respective savings terms interact, since each pixel can only be assigned to a single hypothesis.

Depending on the definition of $f$, we can achieve different effects. The canonical way of combining the different cues would be to simply ignore possible correlations and marginalize over the cues $q_m$. This can be expressed by the following *sum* criterion:

$$f_{\text{sum}}(\mathbf{p}, h, Q) = \sum_m p(\mathbf{p} = figure | h, q_m) p(q_m). \quad (6)$$

However, this marginalization has the problem that it may reinforce local misclassifications if the cues are correlated. An opposite strategy is to completely remove correlation by only trusting the strongest cue. This leads to the *max* criterion:

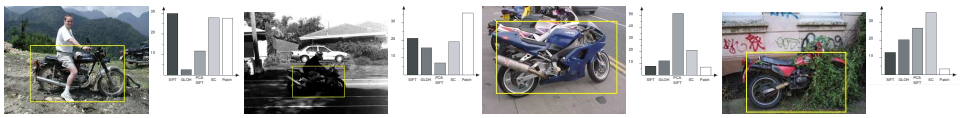


**Figure 2:** *Some detections and the corresponding relative cue contributions.*

$$f_{\max}(\mathbf{p},h,Q) = \max_m p(\mathbf{p} = \mathit{figure}|h,q_m)p(q_m). \tag{7}$$

However, this criterion is also problematic, since it relies on the assumption that all cues are well-behaved. If one or more cues respond too strongly to background structures, the whole system may become biased and additional false positives may be generated. For this reason, we also propose a third criterion, which is a combination of the two extremes. It builds the per-pixel *average* over all cues that are sufficiently confident, i.e. where $p(\mathbf{p} = \mathit{figure}|h,q_m) \gg p(\mathbf{p} = \mathit{ground}|h,q_m)$.

$$f_{\text{avg}}(\mathbf{p},h,Q) = \operatorname*{avg}_m p(\mathbf{p} = \mathit{figure}|h,q_m)p(q_m) \tag{8}$$

These criteria implement a highly flexible combination strategy. Instead of weighting each cue just by a fixed prior, they can decide for each image pixel anew which cues to consider, where the decision is made based on the cues' own confidence estimates. At the same time, eqs. (6) and (8) avoid putting all trust into a single cue that might bias the results negatively. Figure 1 summarizes the final cue combination procedure. The system first generates a set of hypotheses (Fig. 1(a)) by pooling the information from all cues. For each hypothesis, it then computes a top-down segmentation per cue (Fig. 1(b)), whereupon the verification criterion from eq. (3) is executed in order to fuse the individual cues' $p(\mathit{figure})$ probability maps (Fig. 1(c)) into a common system response.

**Discussion and Analysis.** It is important to emphasize the difference of the proposed cue integration scheme to the far simpler approach of running several region detectors in parallel and pooling their features in a common codebook (as used e.g. in [4]). If only a single kind of region descriptors is used, such an approach would be similar to our integration using the *sum* criterion. However, as soon as several different region descriptors shall be employed, a combination into a common codebook is no longer possible, since the different descriptors are not comparable. Our proposed approach, on the other hand, readily scales to this case and allows to combine the different cue contributions on a flexible per-pixel basis, which is something no other current approach can achieve.

The proposed cue integration scheme was motivated by the potential of different local cues to complement each other by interpreting the image information in different ways. In order to visualize that this can positively affect recognition performance, we introduce the following *argmax* criterion as an analysis tool.

$$f_{\text{argmax}}(\mathbf{p},h,Q) = \operatorname*{argmax}_m p(\mathbf{p} = \mathit{figure}|h,q_m)p(q_m) \tag{9}$$

This criterion selects for each hypothesis pixel the index of the most confident cue. Fig. 1(d) shows the resulting maps for the two example images, where each shade of gray corresponds to one of the five descriptors *SIFT*, *GLOH*, *PCA-SIFT*, *Shape Context*, and *Patch* (c.f. Sec. 4). These images are readily interpretable. For instance, it becomes evident that in the top example, the outer rim of the front wheel is best captured by *Shape Context* descriptors, while the wheel's hub is better represented by *GLOH*. In the bottom example, on the other hand, changed contrast to the background has modified the image content sufficiently, such that similar structures on the rear wheel are better captured by *SIFT*.

We can further quantify the relative importance of each cue to a particular hypothesis $h$ by building up a histogram of their individual contributions. Fig. 1(e) shows the corresponding

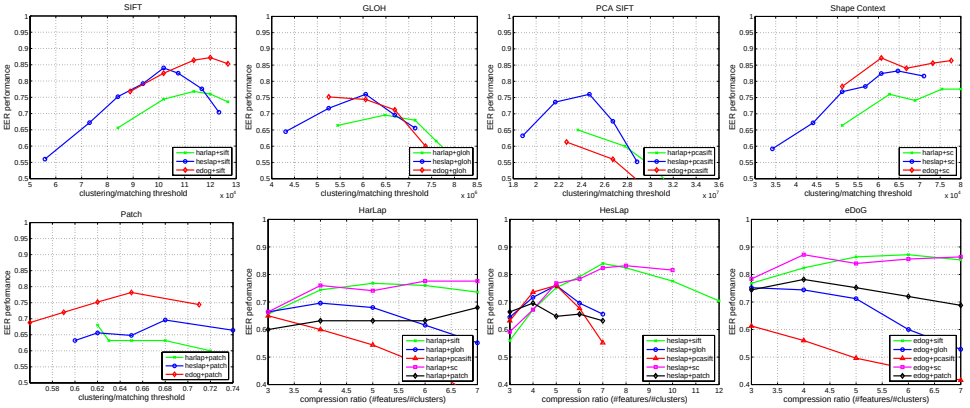



**Figure 3:** *Single-cue EER performances for all detector/descriptor combinations on the TUD motorbikes. The plots show the performance gradation when the clustering/matching threshold is varied. In all following experiments, we use only the best-performing parameter setting for each cue.*

cue importance histograms. As can be seen, the relative importance of the cues changes also quantitatively. Some more examples for different test images are shown in Fig. 2, further corroborating this observation.

# 4 Experimental Setup

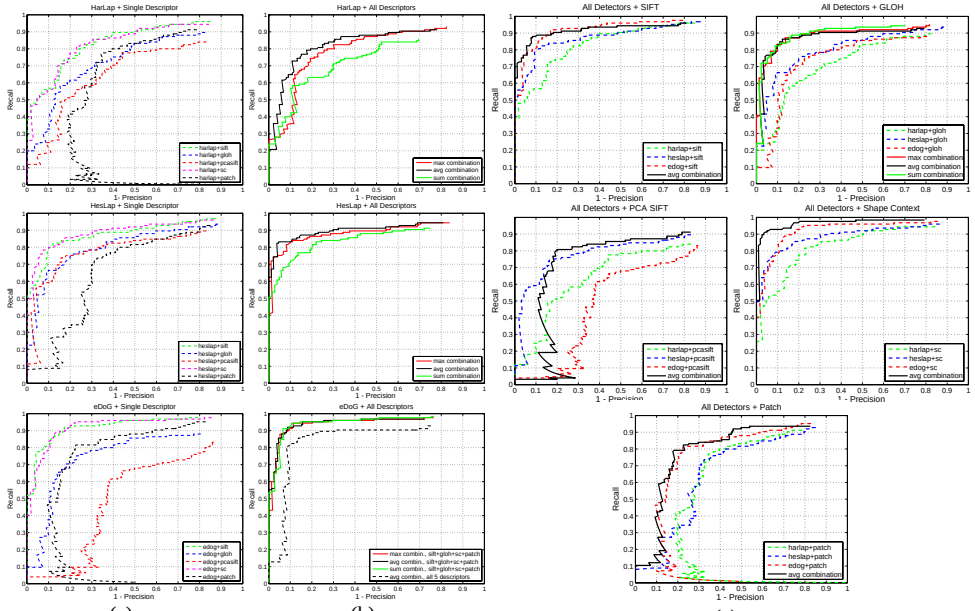
In the rest of the paper, we evaluate our proposed multi-cue integration method on real-world detection tasks. We first describe the selection of cues we build upon and the test data sets.

**Interest Region Detectors.** We compare three different scale-invariant interest region detectors. The *Harris-Laplace* and *Hessian-Laplace* detectors look for scale-adapted maxima of the Harris function and Hessian determinant, respectively [14], where the locations along the scale dimension are found by the Laplacian-of-Gaussian. The *DoG* detector [12] finds regions at 3D scale-space extrema of the Difference-of-Gaussian.

**Region Descriptors.** In addition, we evaluate five different region descriptors. *SIFT* descriptors [12] are 3D histograms of gradient locations and orientations with $4 \times 4$ location and 8 orientation bins. The resulting descriptor has 128 dimensions. *GLOH* descriptors [14] are an extension of *SIFT*. They use 17 location and 16 orientation bins organized in a log-polar grid. PCA is used to reduce the dimensionality to 128. *PCA-SIFT* [8] are vectors of image gradients in *x* and *y* direction sampled within the support region and reduced to 36 dimensions with PCA. *Shape Context* (*SC*) [2, 14] descriptors are histograms of gradient orientations sampled at edge points in a log-polar grid with 9 location and 4 orientation bins and thus 36 dimensions. For comparison, we include $25 \times 25$ pixel *Patches* [1, 10], which lead to a descriptor of length 625. This set of descriptors was explicitly chosen to sample different sources of information. *SIFT*, *GLOH*, and *PCA-SIFT* are based on gradient information; *SC* descriptors are based on edges; and *Patches* take the full image region into account.

The evaluation is performed with an own implementation of the *DoG* detector (denoted *eDoG* in the figures) and *Patch* descriptor. For all other detectors and descriptors, we used the implementations publicly available at [16]. Patches were compared using *Normalized Correlation*; all other descriptors were compared using Euclidean distances.

**Training and Test Data.** We first evaluate the different stages of our approach on the TUD motorbike set, which is part of the PASCAL collection [5]. This data set consists of 115 im-




(a) (b) (c)

**Figure 4:** *Cue combination performances on the TUD motorbikes: (a) single-cue performance; (b) performance of the different combination strategies using all 5 descriptors with the same detector; (c) cue combination performance when the same descriptors are applied to different detectors.*

ages containing a total of 125 motorbikes at different scales and with clutter and occlusion. Training is done on 153 motorbike side views from the CalTech training set [6] which are shown in front of uniform backgrounds allowing for easy segmentation. We then show that the results generalize also to other scenarios by applying the approach to two more challenging data sets using the same parameter settings. The first is the VOC motorbikes `test2` set, which has been used as a localization benchmark in the 2005 PASCAL Challenge [5]. This data set consists of 202 images containing a total of 227 motorbikes at different scales and seen from different viewpoints. Only about 37% of those motorbikes are shown in side views, though, thus limiting the maximally achievable recall for our system. Finally, we apply our method to the pedestrian test set from [11]. It consists of 209 images containing crowded scenes with a total of 595 pedestrians, mostly shown in side views but with significant overlap and occlusion. Training for this test is done on 216 side views of pedestrians for which a segmentation mask was available, using the same parameter settings as for the motorbike experiments. In all three cases, the task is to detect and localize the objects in the test images and determine their correct bounding boxes (using the evaluation criterion from [11] for the first and third test set, and the criterion from [5] for the second test set).

# 5 Results

**Single-Cue Performance.** In order to obtain an unbiased estimate of the cues' potentials, it is important to ensure that they are evaluated at their optimal setting. As a first step, we therefore evaluate each cue separately and try to find its performance optimum.

In our formulation of the approach, there is one open parameter that has to be adjusted for each cue, namely the question how much the clustering step should compress the training

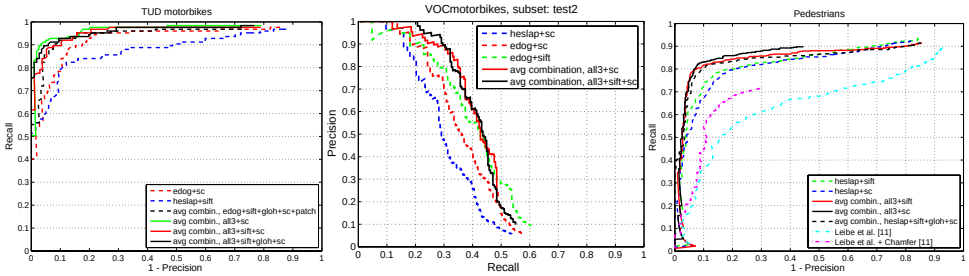



**Figure 5:** *Performance comparison on the TUD motorbikes (left), the more difficult VOC motorbike* `test2` *set (middle), and the pedestrian test set (right). The middle plot is rotated* $90^\circ$ *to make it consistent with the ones in [5]. Please note that while our detector is exclusively trained on side views, only 39% of the motorbikes in the VOC set are shown in side views, thus limiting the maximally achievable recall.*

features during codebook generation. When using agglomerative clustering, this translates to the question how compact the codebook clusters should be for optimal performance. One option is to define a *minimum similarity* after which clustering should be stopped. Another option is to fix a certain *cluster compression ratio* (*#features*/*#clusters*). Previous evaluations [13] have favored the latter option, but it is not guaranteed that this choice is optimal.

In order to analyze the clustering/matching threshold's influence on recognition performance, we applied all 15 detector/descriptor combinations to the TUD motorbikes set and compared their equal error rate (EER) detection performance for 5–7 different threshold settings. Figure 3 shows the results of this experiment, both separated per descriptor and per detector. We can make two observations. First, when comparing descriptors across different detectors, a clear performance optimum can be found at a certain similarity for *SIFT*, *GLOH*, *PCA-SIFT*, and *SC*. The cluster compression ratio, on the other hand, does not seem to have a consistent influence. We can therefore formulate the recommendation to use the cluster similarity as a criterion for selecting the clustering level for those descriptors. Second, the results allow to rank the detector/descriptor combinations based on their single-cue performance. For the descriptors, *SIFT* and *SC* perform consistently best over all three detectors. For the detectors, *Hessian-Laplace* and *DoG* perform best in all but one case. In terms of combinations, *DoG+SIFT* and *DoG+SC* obtain the best performance with 87% EER.

**Combining Different Descriptors.** Next, we examine cue combination in a maximally correlated setting. For this, we apply all five region descriptors to the output of the same interest point detector and compare the performance of the three proposed combination strategies. The results of this experiment can be seen in Fig. 4(a,b). For *Harris-Laplace* and *Hessian-Laplace*, there is a significant difference between the three performance curves, with *sum* combination performing worst, then *max* combination, and *average* combination performing best. This confirms our expectations from Section 3. Compared to the best single-cue performance with *SIFT* or *SC* descriptors, *average* combination achieves a small performance increase from 77.6% to 80.0% (*Harris-Laplace*) and from 82.4% to 85.6% EER (*Hessian-Laplace*), respectively. For *DoG*, a significant performance increase from 87.2% to 91.2% EER can be shown if all descriptors except *PCA-SIFT* are combined. Including *PCA-SIFT* degrades overall performance to 85.6%, suggesting that those descriptors are not as informative as the others, perhaps because of their projection onto a general-purpose PCA basis.

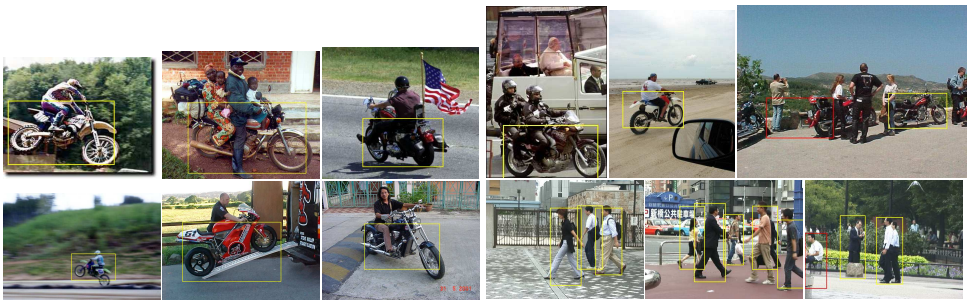


**Figure 6:** *Example multi-cue detections of our approach on difficult images from the VOC motorbikes and the pedestrian set (at the EER).*

**Combining Different Detectors.** The opposite experiment is to apply the same descriptors to three different region detectors and compare the combined performances. This is shown in Fig. 4(c). As there are only small differences between the performance of the three combination strategies, we just display the curve for *average* combination in order to reduce clutter. The most remarkable observation from this experiment is the improvement of over 10% EER obtained by the *GLOH* descriptors from 76.0% to 86.4%. Apparently, this descriptor benefits most from additional samples in the image. In contrast, *SIFT* shows only a small improvement to 88.8% EER. The best absolute performance is achieved by the *SC* combination with 92.8% EER. The *PCA-SIFT* and *Patch* descriptors, finally, do not profit from the evaluated combination.

**Full Multi-Cue Combination.** Finally, we present results combining multiple detectors and multiple descriptors at the same time. Fig. 5(left) compares the performance of *SIFT+SC* and *SIFT+GLOH+SC* with all three detectors. Although those combinations do not increase EER performance any more, further improvement can be observed in terms of precision. In particular, recall at the zero-false-positive level is increased from 50% (only *SC*) over 62% (*SIFT+SC*) to 75% (all three descriptors). This is an important result, since high precision is a prerequisite for many real-world applications.

In order to ensure that the results generalize also to different settings, we apply our multi-cue approach to the more challenging VOC motorbikes set using the same parameter settings as for the first experiments. Fig. 5(middle) shows the results of this experiment. As can be seen from the plot, the combination of multiple cues again improves performance and increases the detection precision considerably. As a comparison with [5] shows, it is the best result reported for this data set so far. The best combination of *SIFT+SC* achieves 21% recall with zero false positives and scales up to 30% recall at 90% precision. Considering that the test set contains only about 39% side views, this is an excellent result. Fig. 6 visualizes the range of motorbike appearances that are still reliably detected by our approach. Although the system has only been trained on a single viewpoint, the increased robustness from multi-cue integration makes it possible to compensate for a certain level of out-of-plane rotation.

Last but not least, we apply our multi-cue approach to the pedestrian test set from [11] using the same clustering/matching thresholds as for the motorbikes. The results are shown in Fig. 5(right). Again, the combination of multiple cues increases performance significantly from 80% EER for the best single cues to 84.7% for *SC* with all three detectors and to 82.6% with *HesLap* with *SIFT+GLOH+SC*. In comparison, we show the results from [11], which are clearly outperformed by our multi-cue system.

# 6 Discussion & Conclusion

In conclusion, we have proposed a robust and flexible multi-cue integration scheme that operates even when the cues are highly correlated. It has been shown to improve performance consistently on three different data sets and for two different categories. The improvement is particularly visible in terms of recognition precision and, for the motorbike test sets, high recall values at the zero-false-positive level. Compared to a canonical cue combination strategy of simply adding the weighted cue responses, our proposed approach can react more flexibly to varying cue performance and adapt itself automatically. This advantage could also be verified quantitatively in cases where the cues were strongly correlated.

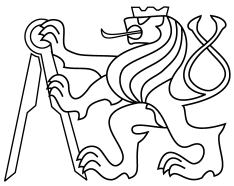
In order to further evaluate its performance we have conducted an extensive study, comparing 3 state-of-the-art interest region detectors and 5 different descriptors in the context of multi-cue integration. The results of this evaluation allow to rank the cues both based on their individual performance and their suitability for integration. In addition, we can draw several interesting conclusions. When set to the right clustering level, *SIFT* and *SC* features performed consistently better than all other descriptors in this evaluation. In addition, feature combinations with either *SC* descriptors and several different region detectors or *DoG*/*Hessian*-*Laplace* regions with several different descriptors achieved the highest overal performance level. These two extremes thus provide an axis along which the set of cues can be varied depending on implementation tradeoffs (i.e. either sampling more points or using the sampled information more efficiently).

**Acknowledgments.** This work has been funded, in part, by the EU projects COSY (IST-2002-004250) and DIRAC (IST-2005-27787).

# References

[1] S. Agarwal, A. Atwan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, 2004.
[2] S. Belongie, J. Malik, and J. Puchiza. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, April 2002.
[3] C. Brautigam, J.-O. Eklund, and H. Christensen. A model-free approach for integrating multiple cues. In *ECCV'98*, 1998.
[4] G. Dorko and C. Schmid. Selection of scale invariant parts for object class recognition. In *ICCV'03*, 2003.
[5] M. Everingham et al. (34 authors). The 2005 pascal visual object class challenge. In *Selected Proceedings of the 1st PASCAL Challenges Workshop*, LNAI. Springer, to appear. `http://www.pascal-network.org/challenges/VOC/`.
[6] R. Fergus, A. Zisserman, and P. Perona. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*, 2003.
[7] A. Garg, S. Agarwal, and T. Huang. Fusion of global and local information for object detection. In *ICPR'02*, 2002.
[8] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR'04*, 2004.
[9] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *PAMI*, 20(3):226–239, 1998.
[10] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Stat. Learn. in Comp. Vis.*, 2004.
[11] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.
[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
[13] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *ICCV'05*, 2005.
[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):31–37, 2005.
[15] M.E. Nilsback and B. Caputo. Cue integration through discriminative accumulation. In *CVPR'04*, 2004.
[16] Oxford interest point webpage. `http://www.robots.ox.ac.uk/~vgg/research/affine/`.
[17] Z. Sun. Adaptation for multiple cue integration. In *CVPR'03*, 2003.
[18] J. Triesch and C. Eckes. Object recognition with multiple feature types. In *ICANN'98*, 1998.
[19] J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. In *Neural Computation*, pages 2049–2074, 2001.
[20] K. Woods, W.P. Kegelmeyer Jr., and K. Bowyer. Combination of multiple classifiers using local accuracy estimation. *PAMI*, 19(4):405–410, 1997.

CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

REPRINT

# Multiview 3D Tracking with an Incrementally Constructed 3D Model

Karel Zimmermann, Tomáš Svoboda and Jiří Matas

zimmerk@cmp.felk.cvut.cz

Karel Zimmermann, Tomáš Svoboda and Jiří Matas, *Multiview 3D Tracking with an Incrementally Constructed 3D Model*,3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill, USA, 2006

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/zimmerk/zimmerk-3dpvt06.pdf

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

# Multiview 3D Tracking with an Incrementally Constructed 3D Model

Karel Zimmermann[1], Tomáš Svoboda[1,2] and Jiří Matas[1]

[1]: Center for Machine Perception
Czech Technical University
Prague, Czech Republic

[2]: Center for Applied Cybernetics
Czech Technical University
Prague, Czech Republic

## Abstract

*We propose a multiview tracking method for rigid objects. Assuming that a part of the object is visible in at least two cameras, a partial 3D model is reconstructed in terms of a collection of small 3D planar patches of arbitrary topology. The 3D representation, recovered fully automatically, allows to formulate tracking as gradient minimization in pose (translation, rotation) space. As the object moves, the 3D model is incrementally updated. A virtuous circle emerges: tracking enables composition of the partial 3D model; the 3D model facilitates and robustifies the multiview tracking.*

*We demonstrate experimentally that the interleaved track-and-reconstruct approach successfully tracks a 360 degrees turn-around and a wide range of motions. Monocular tracking is also possible after the model is constructed. Using more cameras, however, significantly increases stability in critical poses and moves. We demonstrate how to exploit the 3D model to increases stability in the presence of uneven and/or changing illumination.*

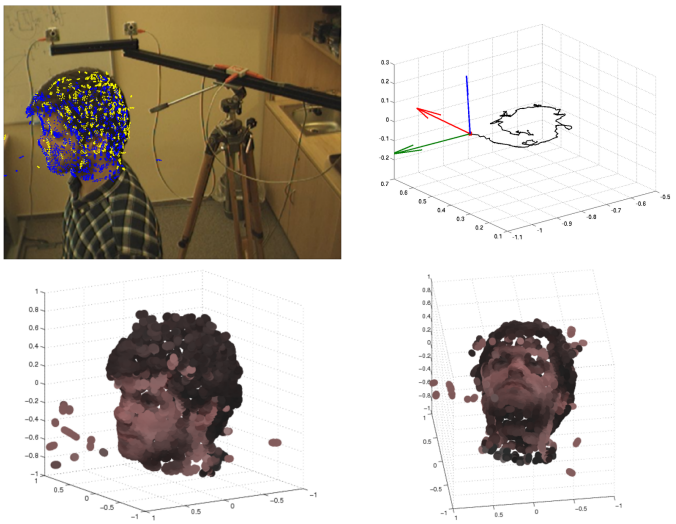


**Figure 1. Interleaved model contruction and tracking: Camera image with reprojected model, trajectory of the head and two different views of the automatically constructed model.**

## 1 Introduction

Existing multiview approaches mostly represent objects as blobs. Blob representation assumes that the appearance of an object does not significantly change when the object rotates. Global object position is sought and the methods do not attempt to recover the *orientation* of the object [3, 9].

Most *model-based tracking* methods use 3D models prepared offline. An overview of such methods was recently published by Lepetit et al. [7]. Vacchetti et al. [16] propose a tracker based on matching with keyframes. The method demonstrates impressive results on out-of-plane rotation data. Still, it cannot track complete turn of the object and needs offline manual selection of keyframes which are essential for its stability. Muñoz et al. [10] suggest a method that track even deformable objects. Their model is composed of small textured planar patches, a set of shape bases, and a set of texture bases. The tracking procedure needs a reference image and optimizes over local shape deformations, colour/texture changes and overall motion. Results on real data show successful tracking only of small variations in object pose and negligible local deformations.

Several approaches build elaborated 3D models from multiple views. The methods rely heavily on carefully constructed and expensive setup and require special scene arrangement since they are based on scene/object segmentation [1, 5, 8, 17]. Würmlin et al. [17] propose dynamic 3D point samples for streaming 3D video. This point based representation somehow resembles our model. However, the method does not track object and needs many cameras and very precise pixel-wise motion segmentation.

We propose a combined method that tracks objects in 3D and constructs a point based appearance model simultaneously. The primary interest is object tracking and detection. The model is rather simple, a set of 3D points associated with 3D orientation and albedo. Despite its simplicity, the model is rich enough for recovering orientation of the object. The tracking can follow a complete 360 degree turn of object. Rothganger et al. [12] also compose a 3D model from small planar patches. The patches are reconstructed from multiview correspondences. Objects are photographed an object from several viewpoints, corresponding image patches are found by affine covariant feature matching. Finally, patches are reconstructed in 3D. In fact, it would be possible to use this model in our tracking. Any complete off-line built model [11] could be used, too.

Cobzas and Jagersand [2] propose a monocular, registration-based, 3D camera tracking of the planar 3D patches. The 3D planar patches are estimated from tracks. Although the formulation of the tracking resembles our method, there are several differences. The patch based model is initialized at the beginning of the sequence (in about 100 frames) by using a standard 2D patch based tracker. Then the algorithm switches to tracking and refine the model using 3D model-based tracking. Cobzas et al. estimate camera pose, assuming a rigid scene. Unlike our method which models illumination changes, Cobzas et al. assume constant illumination and intensity of observed points. Our method builds the model from the very beginning of the sequence. Tracked objects change their position and orientation w.r.t. to light sources. In this case, constant pixel intensities cannot be assumed even for Lambertian surfaces and our method reflects this.

## 2 3D tracking

An object $O$ is modelled as a triplet $(X, \alpha, N)$ where $X$ is a set of 3D points, $\alpha : X \rightarrow \mathcal{R}$ assigns albedo and $N : X \rightarrow \mathcal{S}^2$ a normal to each point $\mathbf{x} \in X$, where $\mathcal{S}^2$ is a sphere. During tracking, intensity $T(\mathbf{x})$ of point $\mathbf{x}$ in a given frame is predicted from its albedo $\alpha(\mathbf{x})$ and an estimated illumination as detailed in section 3.

Assuming rigidity, the motion of points $\mathbf{x} \in X$ between two time instances $t_1$ and $t_2$ is

$$\mathbf{x}^{t_2} = \mathtt{R}\mathbf{x}^{t_1} + \mathbf{d},$$

where $\mathtt{R}$ represents rotation and $\mathbf{d}$ translation. When the rotation is small [4] (e.g. between two consecutive video frames), the motion equation simplifies to

$$\mathbf{x}^{t} = (\mathtt{I} + \mathtt{D})\mathbf{x}^{t-1} + \mathbf{d}, \tag{1}$$

where the rotation matrix $\mathtt{R}$ is replaced by an antisymmetric matrix $\mathtt{D}$ and an identity matrix $\mathtt{I}$. Matrix $\mathtt{D}$ is defined by

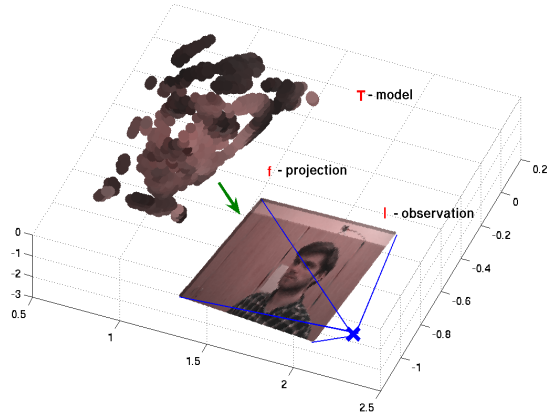



**Figure 2. Model (template) $T$ is projected by projection function $f$ and compared to the current observation $I$.**

three parameters $\mathbf{u} = [D_1, D_2, D_3]^T$;

$$\mathtt{D} = \begin{bmatrix} 0 & D_3 & -D_2 \\ -D_3 & 0 & D_1 \\ D_2 & -D_1 & 0 \end{bmatrix}.$$

Tracking in 3D is defined as the process of finding motion parameters $\mathtt{D}, \mathbf{d}$ minimizing the following image dissimilarity

$$\sum_{\mathbf{x} \in X} \left[ T\left(\mathbf{x}^{t-1}\right) - I\Big(f\left(\mathbf{x}^{t}\right)\Big) \right]^2, \tag{2}$$

where $I : \mathcal{R}^2 \rightarrow \mathcal{R}$ assigns intensity to each pixel, $T : X \rightarrow \mathcal{R}$ assigns intensity to each 3D point. The projection function $f : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ maps 3D points to image coordinates and depends on internal and external parameters of the camera, see Appendix A for details.

Substituting from equation (1) for $\mathbf{x}^t$ in the dissimilarity function (2) and simplifying notation by setting $\mathbf{x}^{t-1} = \mathbf{x}$, a cost function in six unknowns is obtained

$$J(\mathbf{u}, \mathbf{d}) = \sum \left[ T\left(\mathbf{x}\right) - I\Big(f\left(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d}\right)\Big) \right]^2, \tag{3}$$

where the sum is over all $\mathbf{x} \in X$ as in (2); starting from (3) the summation range is omitted for brevity. We seek motion parameters $\mathbf{u}$ and $\mathbf{d}$ that minimize dissimilarity $J(\mathbf{u}, \mathbf{d})$. At the minimum, the partial derivatives with respect to all variables must be zero:

$$\frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{d}} = \mathbf{0}, \quad \frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}} = \mathbf{0},$$

which yields the following two vector equations

$$\sum \Big[ T(\mathbf{x}) - I\big(f(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d})\big) \Big] \frac{\partial I\big(f(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d})\big)}{\partial \mathbf{d}} = \mathbf{0}, \tag{4}$$

$$\sum \Big[T(\mathbf{x}) - I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)\Big] \frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{u}} = \mathbf{0}, \quad (5)$$

There is no closed-form solution for $(\mathbf{u}, \mathbf{d})$. We therefore apply Newton-Raphson minimization, approximating $I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)$ by its first-order Taylor expansion

$$I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big) \approx I\big(f(\mathbf{x})\big) + \mathbf{g}^T(\mathtt{D}\mathbf{x}+\mathbf{d}), \quad (6)$$

where

$$\mathbf{g}^T = I'^T\big(f(\mathbf{x})\big) f'(\mathbf{x}); \quad (7)$$

$I' : \mathcal{R}^2 \to \mathcal{R}^2$ is the gradient of image $I$ and $f' : \mathcal{R}^3 \to \mathcal{R}^{2\times 3}$ is the Jacobian of the projection function $f$.

Differentiating the linear approximation (6) leads to

$$\frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{d}} \approx \mathbf{g}, \quad (8)$$

$$\frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{u}} \approx \frac{\partial \mathbf{g}^T \mathtt{D}\mathbf{x}}{\partial \mathbf{u}}. \quad (9)$$

Applying the approximations (8), (9), equations (4), (5) are simplified to

$$\sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big) - \mathbf{g}^T\mathtt{D}\mathbf{x} - \mathbf{g}^T\mathbf{d}\big]\mathbf{g} = \mathbf{0}, \quad (10)$$

$$\sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big) - \mathbf{g}^T\mathtt{D}\mathbf{x} - \mathbf{g}^T\mathbf{d}\big]\frac{\partial \mathbf{g}^T\mathtt{D}\mathbf{x}}{\partial \mathbf{u}} = \mathbf{0}, \quad (11)$$

Simple algebraic manipulations confirms that the following two identities hold

$$\mathbf{g}^T\mathtt{D}\mathbf{x} = (\mathbf{g}\times\mathbf{x})^T\mathbf{u},$$

$$\frac{\partial \mathbf{g}^T\mathtt{D}\mathbf{x}}{\partial \mathbf{u}} = (\mathbf{g}\times\mathbf{x}),$$

where $\times$ is the cross product. Equations (11) and (10) can be compactly represented as a system of six linear equations $\mathtt{A}$.

$$\mathtt{A}\begin{bmatrix}\mathbf{u}\\ \mathbf{d}\end{bmatrix} = \mathbf{b}, \quad (12)$$

where

$$\mathtt{A} = \sum \begin{bmatrix} (\mathbf{g}\times\mathbf{x})(\mathbf{g}\times\mathbf{x})^T & (\mathbf{g}\times\mathbf{x})\mathbf{g}^T \\ \mathbf{g}(\mathbf{g}\times\mathbf{x})^T & \mathbf{g}\mathbf{g}^T \end{bmatrix}, \quad (13)$$

$$\mathbf{b} = \sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big)\big] \begin{bmatrix} (\mathbf{g}\times\mathbf{x}) \\ \mathbf{g} \end{bmatrix}. \quad (14)$$

Assuming regular $\mathtt{A}$, the solution approximately minimizing equation $J(\mathbf{u}, \mathbf{d})$ is

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{d}\end{bmatrix} = \mathtt{A}^{-1}\mathbf{b}. \quad (15)$$

The $6 \times 6$ matrix $\mathtt{A}$ consists of four $3 \times 3$ sub-matrices and is block-wise symmetric. Unknown motion parameters $\mathbf{d}$, $\mathbf{u}$ are both $3 \times 1$ column vectors and $\mathbf{b}$ is a $6 \times 1$ column vector.

At least six points are required for $\mathrm{rank}(\mathtt{A}) = 6$. In practice, many more points are visible. If the object is weakly textured back-projected image derivatives $\mathbf{g}$ may get close to zero and matrix $\mathtt{A}$ becomes nearly singular. Texture properties needed for reliable tracking of the object are discussed in [14]. Unlike [14], we optimize over the whole object not just over a small patch.

Newton-Raphson iterations are carried out until convergence or a maximum number of steps $N$. Experiments showed the process converged usually in $8 - 10$ iterations. Convergence may require more iterations when the motion is fast, so $N$ was set to 20.

The tracking method was derived for an intensity image and single camera. Extension to RGB tracking is straightforward. The single sum in solution (13,14) is replaced by summations over all visible points, cameras and all RGB channels.

## 3 Compensation of Illumination

Intensity recorded during model acquisition depends, besides the object shape and reflectance, on light sources. We treat the intensity as albedo. As the object moves, the set of light sources visible from a point and their photometric angles change. When modeling these effects we assume:

- cast shadows can be ignored,
- the light sources are distant,
- no specular reflectance.

Under these assumptions, intensities of all points with identical normals will be scaled by a common matrix (for grayscale images only scalar is considered). We adopted a simple method for estimation of the matrix, which performed well in experiments. The method clusters the points $X$ into $n$ groups $G_1, \ldots, G_n$ according to their normals and compensates the illumination of $i$-th cluster in each optimization step (15) by a color correction matrix

$$\mathtt{E}_i^* = \arg\min_{\mathtt{E}_i} \sum_{\mathbf{x}\in G_i} \|\mathtt{E}_i I(f(\mathbf{x})) - T(\mathbf{x})\|_2^2. \quad (16)$$

Let us denote

$$F(\mathtt{E}_i) = \sum_{\mathbf{x}\in G_i} \|\mathtt{E}_i I(f(\mathbf{x})) - T(\mathbf{x})\|_2^2 =$$

$$\sum_{\mathbf{x}\in G_i} I^T(f(\mathbf{x}))\mathtt{E}_i^T\mathtt{E}_i I(f(\mathbf{x})) - 2T^T(\mathbf{x})\mathtt{E}_i I(f(\mathbf{x})) + T^T(\mathbf{x})T(\mathbf{x}),$$

then minimization yields the following matrix equation

$$\frac{\partial F(\mathtt{E}_i)}{\partial \mathtt{E}_i} = \sum_{\mathbf{x}\in G_i} -2T(\mathbf{x})I^T(f(\mathbf{x}))+2\mathtt{E}_i^* I(f(\mathbf{x}))I^T(f(\mathbf{x})) = \mathtt{0} \tag{17}$$

and the least square solution is

$$\mathtt{E}_i^* = \left[\sum_{\mathbf{x}\in G_i} I(f(\mathbf{x}))I^T(f(\mathbf{x}))\right]^{-1} \sum_{\mathbf{x}\in G_i} T(\mathbf{x})I^T(f(\mathbf{x})). \tag{18}$$

## 4 Tracking-Modeling Algorithm

A minimal configuration able to build the model must include at least one stereo pair. For tracking, a single camera is sufficient.

If no model is available from a previous tracking-modeling session, the processing starts with a stereo-based reconstruction [6] of the visible part of the object. Albedo of each point is determined from the average of intensities at its projections onto images used for 3D reconstruction. The reconstructed points are clustered and replaced by points on fish-scales [13]. Fish-scales are small oriented planar patches obtained by local clustering of the cloud of points. Small clusters of points are replaced by ellipses with half-axes corresponding to the two main eigenvectors of their covariance matrix. The third eigenvector defines the surface normal. Note that, computation of fish-scale representation is much simpler then a complete surface triangulation. Still the fish-scales are experimentally shown to be sufficient representation for 3D tracking. Knowledge of surface orientation at each points allows:

- Efficient visibility calculations for convex objects.
- Compensation of illumination effects.

Once the partial model is known, it can be used for pose estimation. If observed motion in the image indicates that a part of the image moves consistently with points currently in the model, stereo is invoked again and newly reconstructed patches are merged into the model. The complete algorithm is summarized in Figure 3.

Note, that the system never knows when the model is completed, because another consistently moving rigid part of the object can appear later. The system only detects that no reconstruction is currently needed.

1. Capture images
2. If needed, invoke **stereo reconstruction** and merge it to the model.
3. **Estimate the pose** of the object by iterating least square solution (15).
4. **Update matrices** $\mathtt{E}_1, \ldots, \mathtt{E}_n$ and for all $i$ and each $\mathbf{x} \in G_i$ recompute object intensity $T(\mathbf{x}) \leftarrow \mathtt{E}_i T(\mathbf{x})$. **goto 1**.

**Figure 3. Tracking-Modeling Algorithm**

## 5 Experiments

The sequences were captured in an office. We used four firewire cameras with resolution of $640 \times 480$ pixels connected to Linux operated computers. The acquisition was TCP/IP synchronized and the setup was calibrated. The total cost of the setup (without computers) is less than 500 dollars and calibration is easy since a free software for automatic (self)calibration exists [15].

Two different sequences were used. In the *human* sequence, a person makes a variety of motions. The individual walks around, shakes and tilts his head. The camera setup consists of two narrow-baseline cameras for stereo reconstruction and two other cameras spanning approximately a half-circle.

The *book* sequence poses slightly different challenges. The book is a relatively thin object and in some poses the dominant planes (front and back cover) are invisible. The camera setup consists of three cameras located near each other. Two of them are used for stereo, all of them are used for tracking. The model of the book is incrementally constructed from a stereo pair and tracked in all cameras.

Objects are tracked successfully in both sequences and their shapes are correctly reconstructed. We performed experiments to assess the accuracy and robustness of multiview and monocular tracking. Section 5.1 shows that the accuracy of multiview tracking is sufficient for incremental model construction without additional alignment. Section 5.2 compares monocular and polynocular tracking. We show that monocular tracking often estimates poses which are incorrect but look correct in the tracking camera. Robustness is tested in section 5.3 on the book sequence where the tracking survives even in frames where dominant planes are absent. Experiments showing illumination compensation are described in section 5.4. Tracking speed is considered in section 5.5. Experiments in sections 5.4,5.5 are conducted with illumination compensation.

In Figures 3-5, projections of visible points are depicted in blue and invisible in yellow. Readers are encouraged to zoom-in the Figures in the electronic version of the document and watch the accompanying video sequences.

### 5.1 Interleaved Tracking and Model Construction

The first experiment demonstrates the interleaved operation of tracking and model construction. The process starts with a partial reconstruction in the first frame, see the leftmost column of Figure 4. The tracker is initialized using this partial model. As the human is turning Fig.4(b), the model, is augmented by adding further partial reconstructions Figs. 4(c,d). Once the 360 turn is finished, the model is complete and further reconstruction are not required.

The 3D model is only a side product of the tracking. Its visual appearance cannot match models created with specialized stereo algorithms or visual-hull based algorithms.

### 5.2 Monocular Model-Based 3D Tracking

In the case of monocular tracking, a 3D model and its initial position are considered to be known in advance (e.g. we use the model from previous experiment). The head was successfully tracked over 630 frames, despite the fact that both 3D translation and out-of-plane rotation were present in the sequence. Tracking results are shown in Figure 5. In images from the tracking camera, the projected model poses seem correct. However, since only a single camera was used, the recovered 3D position is inaccurate, see row 2 in Figure 5. Naturally, the more cameras are used for the optimization, the more accurate 3D pose becomes. Results from the same sequence with the object tracked by all cameras are depicted in the last row of Figure 5.

### 5.3 Robustness against Critical Poses

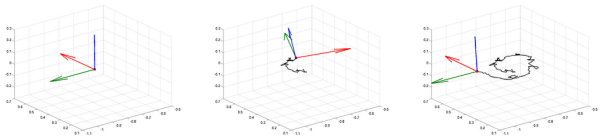
A thin object like the book used in the experiment may easily appear in poses which are inherently challenging for a tracking algorithm. If only the back is visible, the tracking may get unstable. Even during multiview tracking it may happen that most of the object is visible only in a single camera. We call such poses critical.

In a critical pose, the book has to be tracked virtually from the single view. The position of the model does not correspond to the projection in the cameras where only a small fraction of the book is observable. After the object leaves critical pose, the model converges to the true position, see Figure 6.

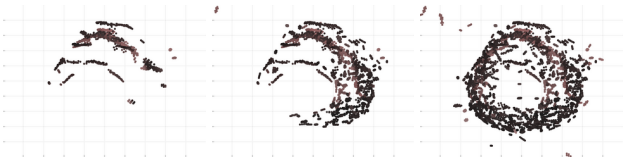
### 5.4 Compensation of Illuminance Effects

The model points are clustered in 14 equally distributed clusters according to their normals. Each cluster is associated with illuminance constant $E_i$ which changes during the tracking to best fit the observed data.

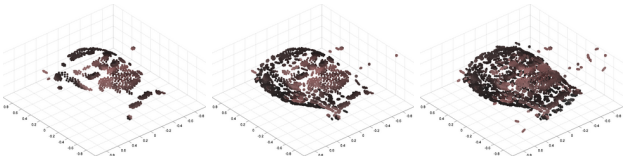
a) Multiview tracking; blue are visible, yellow invisible (occluded) projections



b) Corresponding poses and path recorded



c) Incremental construction of the model, as seen from top



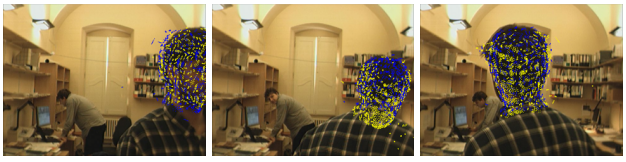
d) Incremental construction of the model, an random view

**Figure 4. Incremental model construction from partial 3D reconstructions and registered by 3D tracking. Rows 1-3: Different views with projected model. Row 4: Position and orientation in 3D space. Rows 5-6: incrementally constructed 3D model. Columns correspond to frames 1, 100 and 310.**

Figure 8-left shows a view with a projected model. Gray levels of particular fish-scales correspond to the values of illuminance constants. Higher values corresponds to the recently illuminated points and vice versa. One can see that in this case light sources were located on the left side of the object which corresponds to the reality.

*Tracking camera*, in monocular tracking, this is the only one used for optimization. Results of monocular tracking projected

*Monocular tracking* results as projected to a camera which approximately orthogonal to the tracking one.



*Polynocular tracking*. The same camera as above. Note the essentially more consistent 3D pose.

**Figure 5. Comparison of monocular (rows 1-2) and polynocular (row 3) tracking. Monocular: Row 1: view from the tracking camera, Row 2: observing camera (shows that, accuracy in orthogonal direction is low). Polynocular: Row 3: The same camera with the projected model from multiview tracking.**

The office has several light sources placed on opposite walls and oriented to the irregularly arched ceiling. Corresponding changes of the illuminance constant $E_6$ during 360 turn are shown at Figure 8-right. Two significant changes during the turn corresponding the light sources are clearly visible. The function of illumination changes is not smooth because during the turn, fish-scales visibility in particular cameras changes and in different times different sets of fish-scales are used for the compensation of illumination effects. Another reason is local inaccuracy of tracking caused by image discretization. Tracking trajectories as well as illumination changes could be smoothed using a motion model, but in our experiments only the output of the optimization is used.

### 5.5 Speed Evaluation

The speed of the algorithm shown in Figure 3 was tested on the sequence introduced in the first two experiments (i.e. 4 cameras, RGB images). Slightly-optimized implementation in Matlab runs cca 1.8 s/frame on an AMD-64b linux running machine. We experimentally show that the tracking of the same sequences in graysacle is successful as well as in RGB. Since one of the most important property of the tracking is the framerate, we increase it 3 many times by considering only grayscale model/sequence.

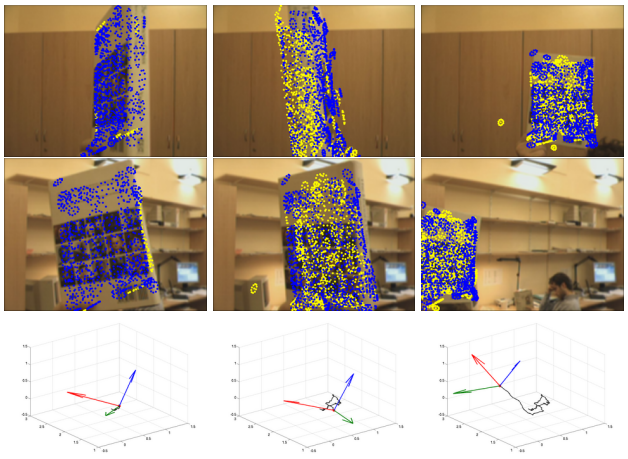


**Figure 6. Book tracking: Rows 1-2: different cameras with projected model, row 3: shows position and orientation in 3D space, columns correspond to frames 55, 205 and 265. The second column shows the book in a critical position where dominant plane is visible only in one camera.**

Tracking of grayscale sequence takes approximately 800 ms/frame. Typically, multiple cameras are connected to different computers. Hence, all the contributions to the $\mathtt{A}, \mathbf{b}$ from equation (13,14) can be computed independently on the particular computers. Using such a system, a frame rate of 5 frames per second can be achieved with the current Matlab implementation.

## 6 Conclusions

We proposed a fully automatic approach of multiview/monocular 3D object tracking interleaved with incremental model construction. Neither model nor initialization are needed to be known in advance. We formulated tracking as a gradient based method minimizing dissimilarity of the observe image and projected 3D point intensities. We showed that the fish-scale 3D model [13] is accurate enough to support stable 3D tracking.

We experimentally demonstrated that the proposed interleaved approach, successfully tracks a complete 360 turn and a wide range of motion without a need for pre-prepared

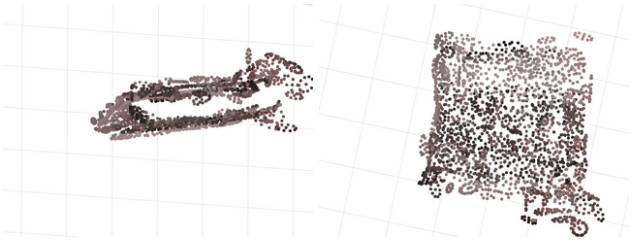


**Figure 7. Book Model: Different views of the book model. Small non-planarity in one corner is the reconstructed hand.**

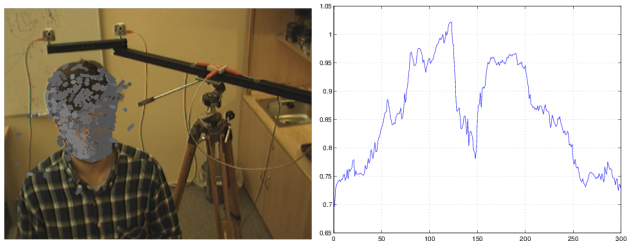


**Figure 8. Left: The image with projected model. Colors correspond to the computed illuminance $\mathtt{E}_i$ of each particular cluster. Right: Values of $\mathtt{E}_6$ during the the 360 turn.**

3D model. A 3D model is delivered as a side product. We demonstrated the robustness of our method on a sequence with a thin object where the dominant plane was often tracked only from one view.

We showed that monocular tracking is possible if the model is available. The model projection to the tracking camera often looks correct, projections to other cameras reveals 3D inaccuracies. Still, monocular tracking can provide results acceptable for some applications. Using more cameras significantly increases stability and accuracy in critical poses and moves. Exact 3D pose may be necessary in many application ranging from virtual reality, human computer interfaces to visual surveillance.

## Acknowledgement

Karel Zimmermann was supported by The Czech Academy of Sciences under project 1ET101210407. Tomáš Svoboda was supported by The Czech Ministry of Education under project 1M0567. Jiří Matas was supported by The European Commission under project IST-004176. Partial support of EU project Dirac FP6-IST-027787 and The STINT under project Dur IG2003-2 062 is also acknowledged.

## Appendix A

A 3D point $\mathbf{x}$ is projected to 2D image (pixel) coordinates $\mathbf{p}$ as

$$\begin{bmatrix} \lambda\mathbf{p} \\ \lambda \end{bmatrix} = \mathtt{P} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix},$$

where $\mathtt{P}$ is $3 \times 4$ camera matrix [4] and $\lambda \in \mathcal{R}$. Let the camera matrix be parameterized as

$$\mathtt{P} = \begin{bmatrix} \mathbf{m}_1^T & t_1 \\ \mathbf{m}_2^T & t_2 \\ \mathbf{m}_3^T & t_3 \end{bmatrix} \tag{19}$$

the function $f : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ projecting 3D point to the camera coordinates is

$$f(\mathbf{x}) = \begin{bmatrix} \frac{\mathbf{m}_1^T\mathbf{x}+t_1}{\mathbf{m}_3^T\mathbf{x}+t_3} \\ \frac{\mathbf{m}_2^T\mathbf{x}+t_2}{\mathbf{m}_3^T\mathbf{x}+t_3} \end{bmatrix}. \tag{20}$$

Differentiating $f$ with respect to $\mathbf{x}$ we obtain $f' : \mathcal{R}^3 \rightarrow \mathcal{R}^{2\times 3}$ Jacobian matrix function, which consists of elements

$$f'_{pq} = \frac{m_{pq}(\mathbf{m}_3^T\mathbf{x} + t_3) - m_{3q}(\mathbf{m}_1^T\mathbf{x} + t_p)}{(\mathbf{m}_3^T\mathbf{x} + t_3)^2} \tag{21}$$

where $m_{pq}, p = 1 \ldots 2, q = 1 \ldots 3$ is $q$-th elements of $\mathbf{m}_p^T$.

## References

[1] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Transaction on Computer Graphics*, 22(3), July 2003.

[2] D. Cobzas and M. Jagersand. 3D SSD tracking from uncalibrated video. In *Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA), in conjunction with ECCV 2004*, 2004.

[3] F. Fleuret, R. Lengagne, and P. Fua. Fixed point probability field for complex occlusion handling. In *IEEE International Conference on Computer Vision*, 2005.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] T. Kanade, P. Narayanan, and P. W. Rander. Virtualized reality: Concepts and early results. In *IEEE Workshop on the Representation of Visual Scenes*, pages 69–76, June 1995.

[6] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In R. Harvey and J. A. Bangham, editors, *BMVC 2003: Proceedings of the 14th British Machine Vision Conference*, volume 1, pages 339–348, London, UK, September 2003. British Machine Vision Association.

[7] V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[8] B. C. R. R. M. L. Matusik, W. and S. Gortler. Image–based visual hulls. In *Proceedings of ACM SIGGRAPH*, 2000.

[9] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *The seventh European Conference on Computer Vision, ECCV2002*, number 2350 in LNCS, pages 18–36. Springer, May 2002.

[10] E. Muñoz, J. Buenaposada, and L. Baumela. Efficient model-based 3D tracking of deformable objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 877–882, China, October 2005.

[11] D. Nistér. Automatic passive recovery of 3d from images and video. In *Second International Symposium on 3D Data Processing, Visualization and TRansmission (3DPVT04)*, 2004. Invited paper.

[12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 272–277, 2003.

[13] R. Šára and R. Bajcsy. Fish-scales: Representing fuzzy manifolds. In S. Chandran and U. Desai, editors, *Proc. 6th International Conference on Computer Vision*, pages 811–817, New Delhi, India, January 1998. IEEE Computer Society, Narosa Publishing House.

[14] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[15] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[16] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004.

[17] S. Würmlin, E. Lamboray, and M. Gross. 3D video fragments: Dynamic point samples for real-time free-viewpoint video. *Computers and Graphics*, 28(1):3–14, 2004.