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 4×4 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_Σ 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 σ 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 9×9 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

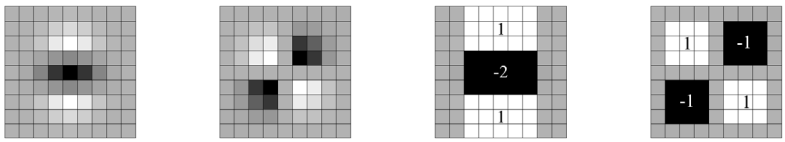

Fig. 1. Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and our approximations thereof using box filters. The grey regions are equal to zero.

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 9×9 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 9×9, 15×15, 21×21, 27×27, 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 27×27 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

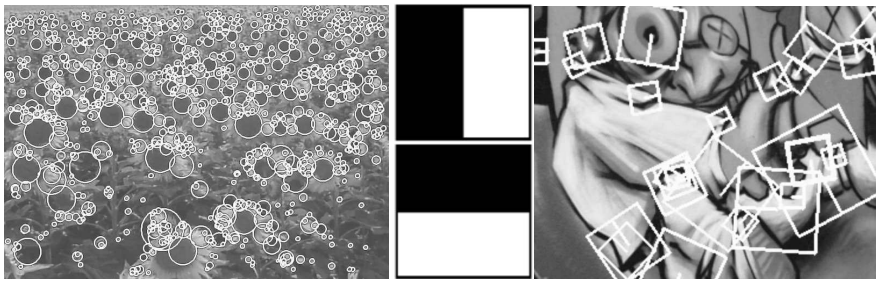

Fig. 2. Left: Detected interest points for a Sunflower field. This kind of scenes shows clearly the nature of the features from Hessian-based detectors. Middle: Haar wavelet types used for SURF. Right: Detail of the Graffiti scene showing the size of the descriptor window at different scales.

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 4×4 square sub-regions. This keeps important spatial information in. For each sub-region, we compute a few simple features at 5×5 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 4×4 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

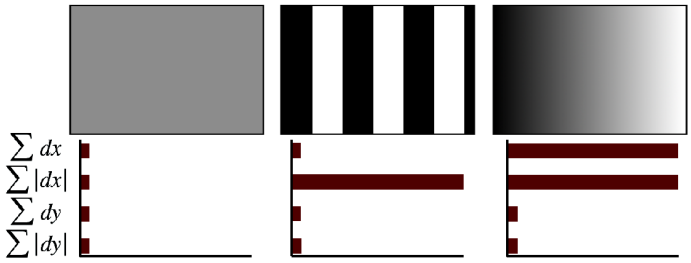

Fig. 3. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity 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 4×4 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 3×3 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).

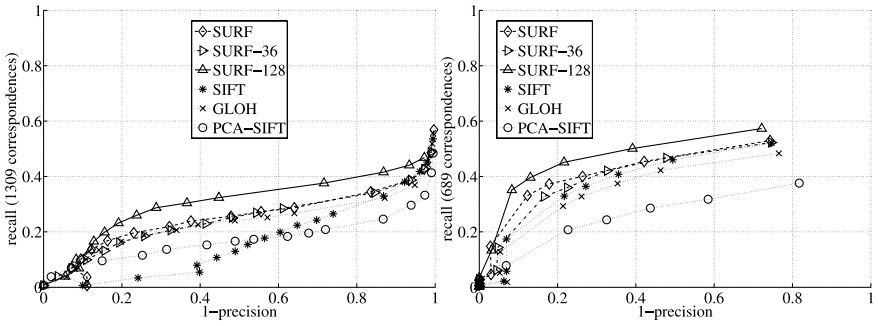

Fig. 4. The *recall* vs. *(1-precision)* graph for different binning methods and two different matching strategies tested on the 'Graffiti' sequence (image 1 and 3) with a view change of 30 degrees, compared to the current descriptors. The interest points are computed with our 'Fast Hessian' detector. 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 4×4 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, 800×640)

	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 (320×240) 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.

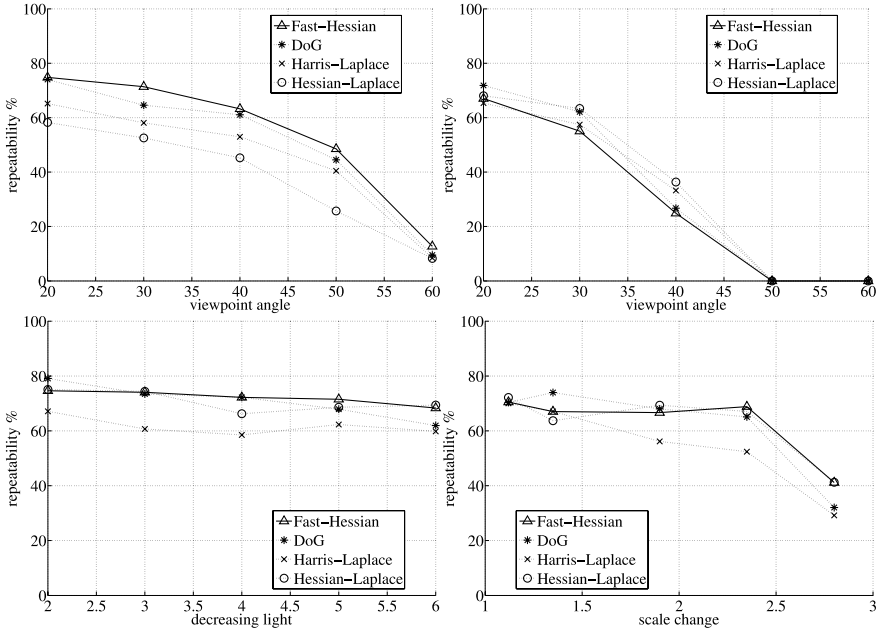

Fig. 6. Repeatability score for image sequences, from left to right and top to bottom, Wall and Graffiti (Viewpoint Change), Leuven (Lighting Change) and Boat (Zoom and Rotation).

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)

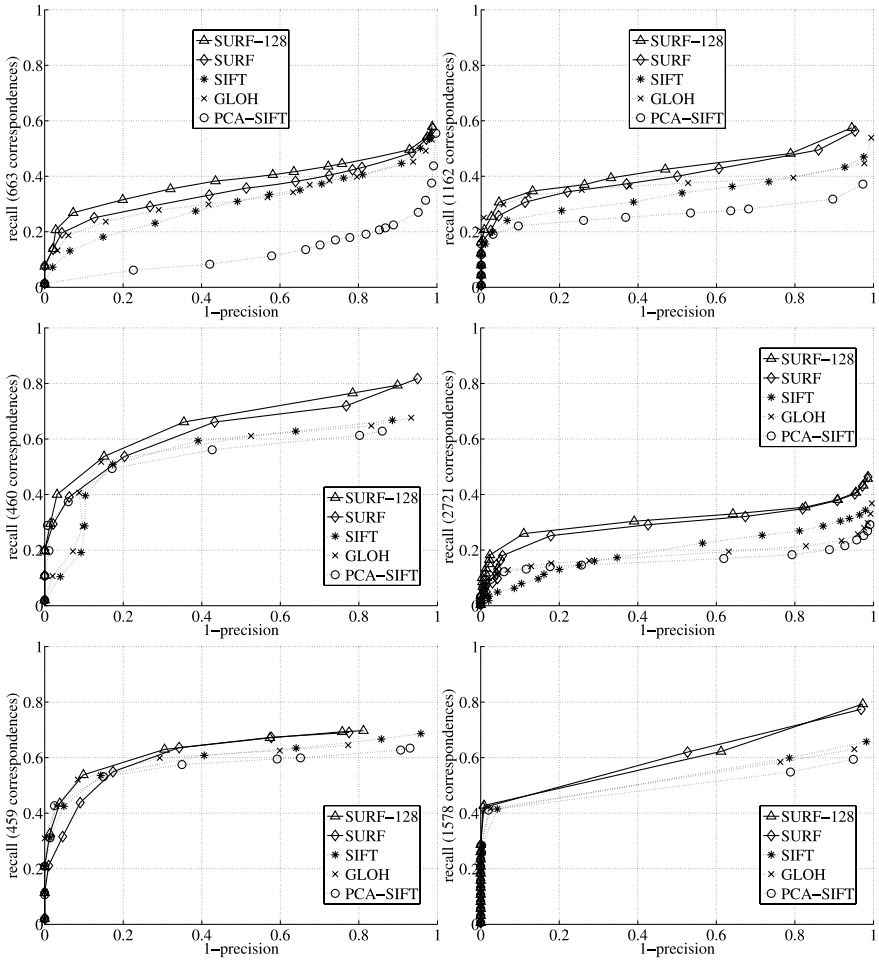

Fig. 7. Recall, 1-Precision graphs for, from left to right and top to bottom, Viewpoint change of 50 (Wall) degrees, scale factor 2 (Boat), image blur (Bikes and Trees), brightness change (Leuven) and JPEG compression (Ubc).