Speeded-Up Robust Features (SURF)

Herbert Bay [a] , Andreas Ess [a] , Tinne Tuytelaars [b] , and Luc Van Gool [a,b]

[a] *ETH Zurich, BIWI*
Sternwartstrasse 7
CH-8092 Zurich
Switzerland

[b] *K. U. Leuven, ESAT-PSI*
Kasteelpark Arenberg 10
B-3001 Leuven
Belgium

Abstract

This article presents a novel scale- and rotation-invariant detector and descriptor, coined SURF (Speeded-Up Robust Features). SURF 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 (specifically, 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 encompasses a detailed description of the detector and descriptor and then explores the effect of the most important parameters. We conclude the article with SURF's application to two challenging, yet converse goals: camera calibration as a special case of image registration, and object recognition. Our experiments underline SURF's usefulness in a broad range of topics in computer vision.

Key words: interest points, local features, feature description, camera calibration, object recognition
PACS:

1. Introduction

The task of finding point correspondences between two images of the same scene or object is part of many computer vision applications. Image registration, camera calibration, object recognition, and image retrieval are just a few.

The search for discrete image point correspondences 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. The repeatability expresses the reliability of a detector for finding the same physical 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 displacements and geometric and photometric deformations. Finally, the descriptor vectors are *matched* between different images. The matching is 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 less dimensions are desirable for fast interest point matching. However, lower dimensional feature vectors are in general less distinctive than their high-dimensional counterparts.

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

A wide variety of detectors and descriptors have already been proposed in the literature (e.g. [21,24,27,37,39,25]). Also, detailed comparisons and evaluations on benchmarking datasets have been performed [28,30,31]. Our fast detector and descriptor, called SURF (Speeded-Up Robust Features), was introduced in [4]. It is built on the insights

gained from this previous work. In our experiments on these benchmarking datasets, SURF's detector and descriptor are not only faster, but the former is also more repeatable and the latter more distinctive.

We focus on scale and in-plane 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. Note that the descriptor can be extended towards affine invariant regions using affine normalisation of the ellipse (cf. [31]), although this will have an impact on the computation time. Extending the detector, on the other hand, is less straight-forward. Concerning the photometric deformations, we assume a simple linear model with a bias (offset) and contrast change (scale factor). Neither detector nor descriptor use colour information.

In section 3, we describe the strategy applied for fast and robust interest point detection. The input image is analysed at different scales in order to guarantee invariance to scale changes. The detected interest points are provided with a rotation and scale invariant descriptor in section 4. Furthermore, a simple and efficient first-line indexing technique, based on the contrast of the interest point with its surrounding, is proposed.

In section 5, some of the available parameters and their effects are discussed, including the benefits of an upright version (not invariant to image rotation). We also investigate SURF's performance in two important application scenarios. First, we consider a special case of image registration, namely the problem of camera calibration for 3D reconstruction. Second, we will explore SURF's application to an object recognition experiment. Both applications highlight SURF's benefits in terms of speed and robustness as opposed to other strategies. The article is concluded in section 6.

2. Related Work

2.1. *Interest Point Detection*

The most widely used detector is probably the Harris corner detector [15], proposed back in 1988. It is based on the eigenvalues of the second moment matrix. However, Harris corners are not scale-invariant. Lindeberg [21] introduced the concept of automatic scale selection. 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 [26] refined this method, creating robust and scale-invariant feature detectors with high repeatability, which they coined Harris-Laplace and Hessian-Laplace. 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 [23] proposed to approximate the Laplacian of Gaussians (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 [17], which maximises the entropy within the region, and the edge-based region detector proposed by Jurie and Schmid [16]. They seem less amenable to acceleration though. Also several affine-invariant feature detectors have been proposed that can cope with wider viewpoint changes. However, these fall outside the scope of this article.

From studying the existing detectors and from published comparisons [29,30], we can conclude that Hessian-based detectors are more stable and repeatable than their Harris-based counterparts. Moreover, using the determinant of the Hessian matrix rather than its trace (the Laplacian) seems advantageous, as it fires less on elongated, ill-localised structures. We also observed that approximations like the DoG can bring speed at a low cost in terms of lost accuracy.

2.2. *Interest Point Description*

An even larger variety of feature descriptors has been proposed, like Gaussian derivatives [11], moment invariants [32], complex features [1,36], steerable filters [12], phase-based local features [6], and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. The latter, introduced by Lowe [24], have been shown to outperform the others [28]. 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 [24], 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 4×4 location bins).

Various refinements on this basic scheme have been proposed. Ke and Sukthankar [18] applied PCA on the gradient image around the detected interest point. 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 [30]; and applying PCA slows down feature computation. In the same paper [30], the authors 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 as it uses again PCA for data compression.

The SIFT descriptor still seems 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.* [37] implemented SIFT on a Field Programmable Gate Array (FPGA) and improved its speed by an order of magni-

tude. Meanwhile, Grabner *et al.* [14] also used integral images to approximate SIFT. Their detection step is based on difference-of-mean (without interpolation), their description step on integral histograms. They achieve about the same speed as we do (though the description step is constant in speed), but at the cost of reduced quality compared to SIFT. Generally, the high dimensionality of the descriptor is a drawback of SIFT at the matching step. For on-line applications relying only on a regular PC, each one of the three steps (detection, description, matching) has to be fast.

An entire body of work is available on speeding up the matching step. All of them come at the expense of getting an approximative matching. Methods include the best-bin-first proposed by Lowe [24], balltrees [35], vocabulary trees [34], locality sensitive hashing [9], or redundant bit vectors [13]. Complementary to this, we suggest the use of the Hessian matrix's trace to significantly increase the matching speed. Together with the descriptor's low dimensionality, any matching algorithm is bound to perform faster.

3. Interest Point Detection

Our approach for interest point detection uses a very basic Hessian-matrix approximation. This lends itself to the use of integral images as made popular by Viola and Jones [41], which reduces the computation time drastically. Integral images fit in the more general framework of boxlets, as proposed by Simard *et al.* [38].

3.1. *Integral Images*

In order to make the article more self-contained, we briefly discuss the concept of integral images. They allow for fast computation of box type convolution filters. The entry of an integral image $I_\Sigma(\mathbf{x})$ at a location $\mathbf{x} = (x, y)^\top$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and $\mathbf{x}$.

$$I_\Sigma(\mathbf{x}) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i, j) \qquad (1)$$

Once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any upright, rectangular area (see figure 1). Hence, the calculation time is independent of its size. This is important in our approach, as we use big filter sizes.

3.2. *Hessian Matrix Based Interest Points*

We base our detector on the Hessian matrix because of its good performance in accuracy. More precisely, we detect blob-like structures at locations where the determinant is maximum. In contrast to the Hessian-Laplace detector by Mikolajczyk and Schmid [26], we rely on the determinant of the Hessian also for the scale selection, as done by Lindeberg [21].

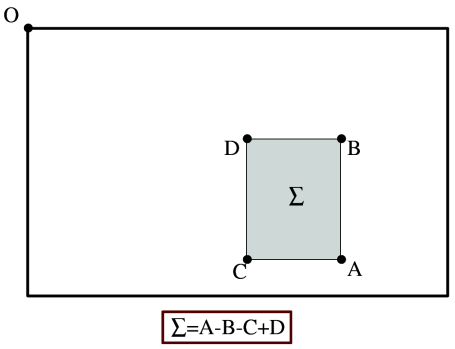

Fig. 1. Using integral images, it takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size.

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

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \qquad (2)$$

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 [19,20], but in practice they have to be discretised and cropped (figure 2 left half). This leads to a loss in repeatability under image rotations around odd multiples of $\frac{\pi}{4}$. This weakness holds for Hessian-based detectors in general. Figure 3 shows the repeatability rate of two detectors based on the Hessian matrix for pure image rotation. The repeatability attains a maximum around multiples of $\frac{\pi}{2}$. This is due to the square shape of the filter. Nevertheless, the detectors still perform well, and the slight decrease in performance does not outweigh the advantage of fast convolutions brought by the discretisation and cropping. As real filters are non-ideal in any case, and given Lowe's success with his LoG approximations, we push the approximation for the Hessian matrix even further with box filters (in the right half of figure 2). These approximate second order Gaussian derivatives and can be evaluated at a very low computational cost using integral images. The calculation time therefore is independent of the filter size. As shown in the results section and figure 3, the performance is comparable or better than with the discretised and cropped Gaussians.

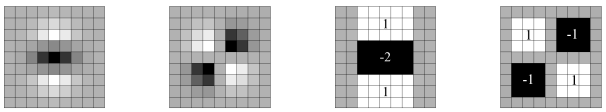

Fig. 2. Left to right: the (discretised and cropped) Gaussian second order partial derivative in y- (L_{yy}) and xy-direction (L_{xy}), respectively; our approximation for the second order Gaussian partial derivative in y- (D_{yy}) and xy-direction (D_{xy}). The grey regions are equal to zero.

The 9×9 box filters in figure 2 are approximations of a Gaussian with $\sigma = 1.2$ and represent the lowest scale (i.e. highest spatial resolution) for computing the blob response maps. We will denote them by D_{xx}, D_{yy}, and D_{xy}. The weights applied to the rectangular regions are kept simple for computational efficiency. This yields

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2. \qquad (3)$$

The relative weight w of the filter responses is used to balance the expression for the Hessian's determinant. This is needed for the energy conservation between the Gaussian kernels and the approximated Gaussian kernels,

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \simeq 0.9, \qquad (4)$$

where $|x|_F$ is the Frobenius norm. Notice that for theoretical correctness, the weighting changes depending on the scale. In practice, we keep this factor constant, as this did not have a significant impact on the results in our experiments.

Furthermore, the filter responses are normalised with respect to their size. This guarantees a constant Frobenius norm for any filter size, an important aspect for the scale space analysis as discussed in the next section.

The approximated determinant of the Hessian represents the blob response in the image at location $\mathbf{x}$. These responses are stored in a blob response map over different scales, and local maxima are detected as explained in section 3.4.

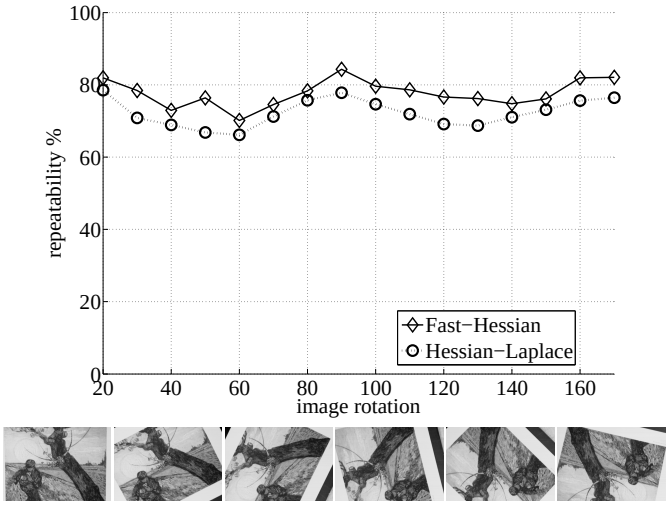

Fig. 3. Top: Repeatability score for image rotation of up to 180 degrees. Hessian-based detectors have in general a lower repeatability score for angles around uneven multiples of $\frac{\pi}{4}$. Bottom: Sample images from the Van Gogh sequence that was used. Fast-Hessian is the more accurate version of our detector (FH-15), as explained in section 3.3.

3.3. *Scale Space Representation*

Interest points need to be found at different scales, not least because the search of correspondences often requires their comparison in images where they are seen at different scales. Scale spaces are usually implemented as an image pyramid. The images are repeatedly smoothed with a Gaussian and then sub-sampled in order to achieve a higher level of the pyramid. Lowe [24] subtracts these pyramid layers in order to get the DoG (Difference of Gaussians) images where edges and blobs can be found.

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 box 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, figure 4. The output of the 9×9 filter, introduced in the previous section, is considered as the initial scale layer, to which we will refer as scale $s = 1.2$ (approximating 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.

Note that our main motivation for this type of sampling is its computational efficiency. Furthermore, as we do not have to downsample the image, there is no aliasing. On the downside, box filters preserve high-frequency components that can get lost in zoomed-out variants of the same scene, which can limit scale-invariance. This was however not noticeable in our experiments.

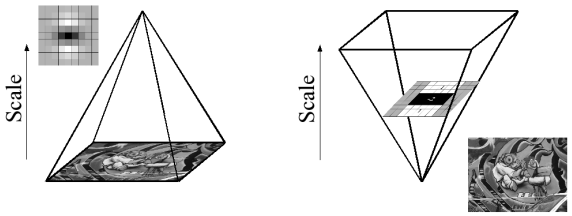

Fig. 4. Instead of iteratively reducing the image size (left), the use of integral images allows the up-scaling of the filter at constant cost (right).

The scale space is divided into octaves. An octave represents a series of filter response maps obtained by convolving the same input image with a filter of increasing size. In total, an octave encompasses a scaling factor of 2 (which implies that one needs to more than double the filter size, see below). Each octave is subdivided into a constant number of scale levels. Due to the discrete nature of integral images, the minimum scale difference between 2 subsequent scales depends on the length l_0 of the positive or negative lobes of the partial second order derivative in the direction of derivation (x or y), which is set to a third of the filter size length. For the 9×9 filter, this length l_0 is 3. For two successive levels, we must increase this size by a minimum of 2 pixels (one pixel on every side) in order to keep the size uneven and thus ensure the presence of the central pixel. This results in a total increase of the mask size by 6 pixels (see figure 5). Note that for dimensions different from

l_0 (e.g. the width of the central band for the vertical filter in figure 5), rescaling the mask introduces rounding-off errors. However, since these errors are typically much smaller than l_0, this is an acceptable approximation.

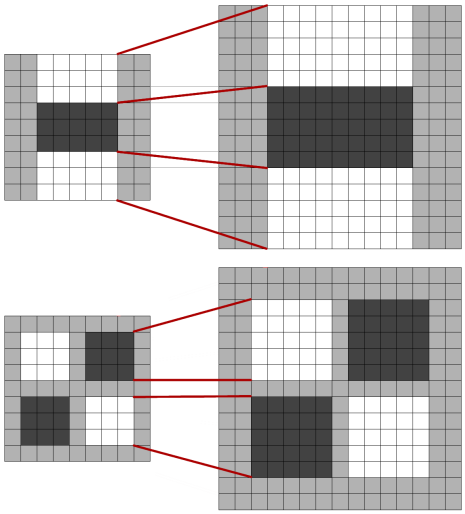

Fig. 5. Filters D_{yy} (top) and D_{xy} (bottom) for two successive scale levels (9×9 and 15×15). The length of the dark lobe can only be increased by an even number of pixels in order to guarantee the presence of a central pixel (top).

The construction of the scale space starts with the 9×9 filter, which calculates the blob response of the image for the smallest scale. Then, filters with sizes 15×15, 21×21, and 27×27 are applied, by which even more than a scale change of 2 has been achieved. But this is needed, as a 3D non-maximum suppression is applied both spatially and over the neighbouring scales. Hence, the first and last Hessian response maps in the stack cannot contain such maxima themselves, as they are used for reasons of comparison only. Therefore, after interpolation, see section 3.4, the smallest possible scale is $\sigma = 1.6 = 1.2\frac{12}{9}$ corresponding to a filter size of 12×12, and the highest to $\sigma = 3.2 = 1.2\frac{24}{9}$. For more details, we refer to [2].

Similar considerations hold for the other octaves. For each new octave, the filter size increase is doubled (going from 6 to 12 to 24 to 48). At the same time, the sampling intervals for the extraction of the interest points can be doubled as well for every new octave. This reduces the computation time and the loss in accuracy is comparable to the image sub-sampling of the traditional approaches. The filter sizes for the second octave are 15, 27, 39, 51. A third octave is computed with the filter sizes 27, 51, 75, 99 and, if the original image size is still larger than the corresponding filter sizes, the scale space analysis is performed for a fourth octave, using the filter sizes 51, 99, 147, and 195. Figure 6 gives an overview of the filter sizes for the first three octaves. Note that more octaves may be analysed, but the number of detected interest points per octave decays very quickly, cf. figure 7.

The large scale changes, especially between the first filters within these octaves (from 9 to 15 is a change of 1.7),

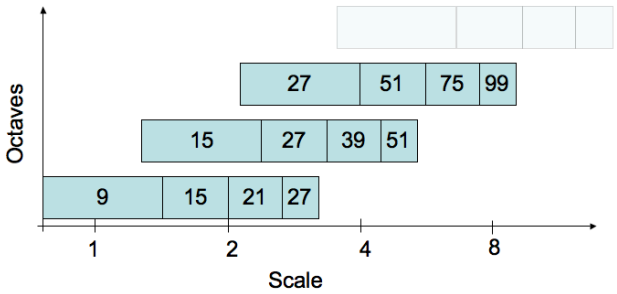

Fig. 6. Graphical representation of the filter side lengths for three different octaves. The logarithmic horizontal axis represents the scales. Note that the octaves are overlapping in order to cover all possible scales seamlessly.

renders the sampling of scales quite crude. Therefore, we have also implemented a scale space with a finer sampling of the scales. This first doubles the size of the image, using linear interpolation, and then starts the first octave by filtering with a filter of size 15. Additional filter sizes are 21, 27, 33, and 39. Then a second octave starts, again using filters which now increase their sizes by 12 pixels, after which a third and fourth octave follow. Now the scale change between the first two filters is only 1.4 (21/15). The lowest scale for the accurate version that can be detected through quadratic interpolation is $s = (1.2\frac{18}{9})/2 = 1.2$.

As the Frobenius norm remains constant for our filters at any size, they are already scale normalised, and no further weighting of the filter response is required, see [22].

3.4. *Interest Point Localisation*

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. Specifically, we use a fast variant introduced by Neubeck and Van Gool [33]. 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.* [5].

Scale space interpolation is especially important in our case, as the difference in scale between the first layers of

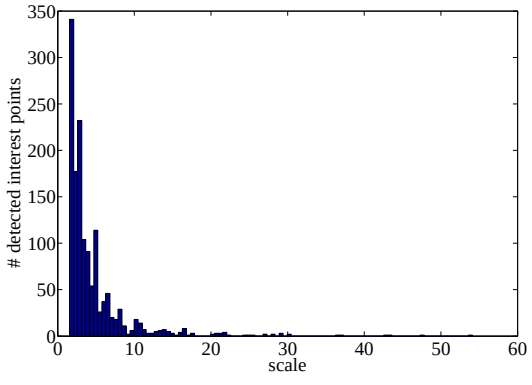

Fig. 7. Histogram of the detected scales. The number of detected interest points per octave decays quickly.

every octave is relatively large. Figure 8 shows an example of the detected interest points using our 'Fast-Hessian' detector.

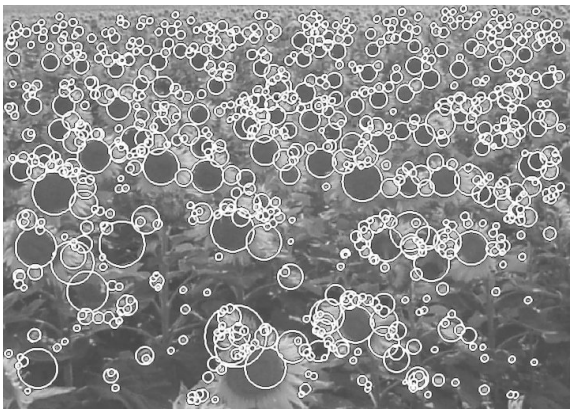

Fig. 8. Detected interest points for a Sunflower field. This kind of scenes shows the nature of the features obtained using Hessian-based detectors.

4. Interest Point Description and Matching

Our descriptor describes the distribution of the intensity content within the interest point neighbourhood, similar to the gradient information extracted by SIFT [24] and its variants. We build on the distribution of first order Haar wavelet responses in x and y direction rather than the gradient, exploit integral images for speed, and use only 64 dimensions. This reduces the time for feature computation and matching, and has proven to simultaneously increase the robustness. Furthermore, we present a new indexing step based on the sign of the Laplacian, which increases not only the robustness of the descriptor, but also the matching speed (by a factor of two in the best case). We refer to our detector-descriptor scheme as SURF – Speeded-Up Robust Features.

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. Finally, features are matched between two images. These three steps are explained in the following.

4.1. *Orientation Assignment*

In order to be invariant to image 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 within a circular neighbourhood of radius $6s$ around the interest point, with s the scale at which the interest point was detected.The sampling step is scale dependent and chosen to be s. In keeping with the rest, also the size of the wavelets are scale dependent and set to a side length of $4s$. Therefore, we can again use integral images for fast filtering. The used filters are shown in figure 9. Only six operations are needed to compute the response in x or y direction at any scale.

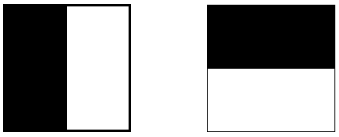

Fig. 9. Haar wavelet filters to compute the responses in x (left) and y direction (right). The dark parts have the weight -1 and the light parts $+1$.

Once the wavelet responses are calculated and weighted with a Gaussian ($\sigma = 2s$) centred at the interest point, the responses are represented as points 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 of size $\frac{\pi}{3}$, see figure 10. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector. The longest such vector over all windows defines the orientation of the interest point. The size of the sliding window is a parameter which had to be chosen carefully. Small sizes fire on single dominating gradients, large sizes tend to yield maxima in vector length that are not outspoken. Both result in a misorientation of the interest point.

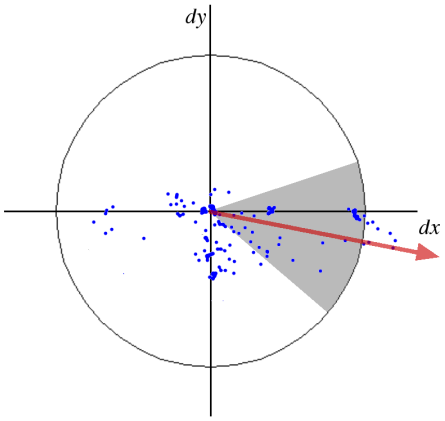

Fig. 10. Orientation assignment: A sliding orientation window of size $\frac{\pi}{3}$ detects the dominant orientation of the Gaussian weighted Haar wavelet responses at every sample point within a circular neighbourhood around the interest point.

Note that for many applications, rotation invariance is not necessary. Experiments of using the upright version of SURF (U-SURF, for short) for object detection can be found in [3,4]. U-SURF is faster to compute and can increase distinctivity, while maintaining a robustness to rotation of about $+/-15°$.

4.2. *Descriptor based on Sum of Haar Wavelet Responses*

For the extraction of the descriptor, the first step consists of constructing a square region centred around the interest

point and oriented along the orientation selected in the previous section. The size of this window is $20s$. Examples of such square regions are illustrated in figure 11.

Fig. 11. Detail of the Graffiti scene showing the size of the oriented descriptor window at different scales.

The region is split up regularly into smaller 4×4 square sub-regions. This preserves important spatial information. For each sub-region, we compute Haar wavelet responses 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 $2s$), see figure 9 again. "Horizontal" and "vertical" here is defined in relation to the selected interest point orientation (see figure 12). [1] To increase the robustness towards geometric deformations and localisation errors, the responses d_x and d_y are first weighted with a Gaussian ($\sigma = 3.3s$) centred at the interest point.

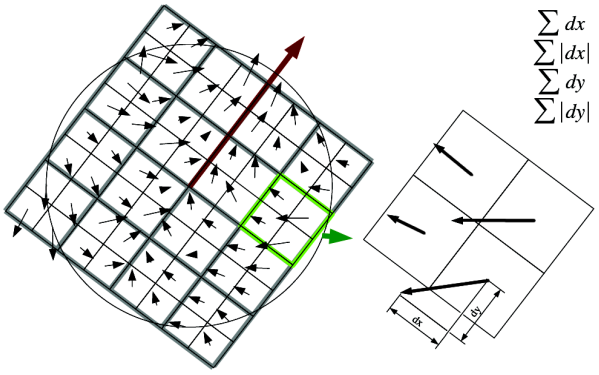

Fig. 12. To build the descriptor, an oriented quadratic grid with 4×4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed. The 2×2 sub-divisions of each square correspond to the actual fields of the descriptor. These are the sums dx, $|dx|$, dy, and $|dy|$, computed relatively to the orientation of the grid (right).

Then, the wavelet responses d_x and d_y are summed up over each sub-region and form a first set of entries in 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 $\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Concatenating this for all 4×4 sub-regions, this results in a descriptor vector 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.

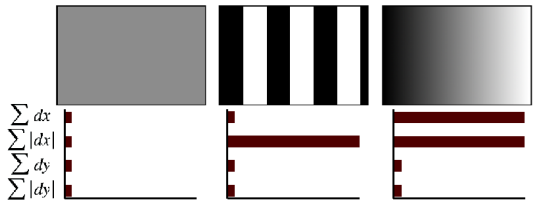

Fig. 13. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

Figure 13 shows the properties of the descriptor for three distinctively different image intensity patterns within a sub-region. One can imagine combinations of such local intensity patterns, resulting in a distinctive descriptor.

SURF is, up to some point, similar in concept as SIFT, in that they both focus on the spatial distribution of gradient information. Nevertheless, SURF outperforms SIFT in practically all cases, as shown in Section 5. We believe this is due to the fact that SURF integrates the gradient information within a subpatch, whereas SIFT depends on the orientations of the individual gradients. This makes SURF less sensitive to noise, as illustrated in the example of Figure 14.

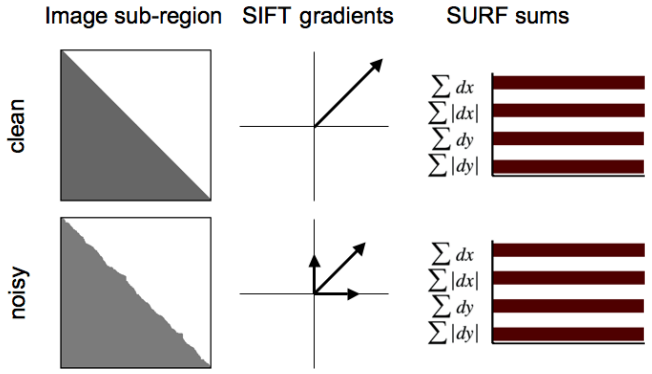

Fig. 14. Due to the global integration of SURF's descriptor, it stays more robust to various image perturbations than the more locally operating SIFT descriptor.

In order to arrive at these SURF descriptors, we experimented with fewer and more wavelet features, second order derivatives, higher-order wavelets, PCA, median values, average values, etc. From a thorough evaluation, the

[1] For efficiency reasons, the Haar wavelets are calculated in the unrotated image and the responses are then interpolated, instead of actually rotating the image.

proposed sets turned 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, see also section 5. 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 sub-regions (SURF-36) performs slightly worse, but allows for very fast matching and is still acceptable in comparison to other descriptors in the literature.

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.

4.3. *Fast Indexing for Matching*

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, see figure 15. Hence, this minimal information allows for faster matching, without reducing the descriptor's performance. Note that this is also of advantage for more advanced indexing methods. E.g. for k-d trees, this extra information defines a meaningful hyperplane for splitting the data, as opposed to randomly choosing an element or using feature statistics.

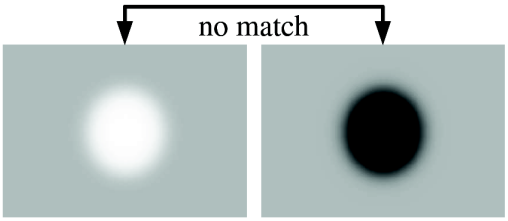

Fig. 15. If the contrast between two interest points is different (dark on light background vs. light on dark background), the candidate is not considered a valuable match.

5. Results

The following presents both simulated as well as real-world results. First, we evaluate the effect of some parameter settings and show the overall performance of our detector and descriptor based on a standard evaluation set.

detector	threshold	nb of points	comp. time (ms)
FH-15	60000	1813	160
FH-9	50000	1411	70
Hessian-Laplace	1000	1979	700
Harris-Laplace	2500	1664	2100
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)

Then, we describe two possible applications. For a detailed comparative study with other detectors and descriptors, we refer to [4]. SURF has already been tested in a few real-world applications. For object detection, its performance has been illustrated in [3]. Cattin *et al.* [7] use SURF for mosaicing human retina images—a task that no other detector/descriptor scheme was able to cope with. Taking this application to image registration a bit further, we focus in this article on the more difficult problem of camera calibration and 3D reconstruction, also in wide-baseline cases. SURF manages to calibrate the cameras even in challenging cases reliably and accurately. Lastly, we investigate the application of SURF to the task of object recognition.

5.1. *Experimental Evaluation and Parameter Settings*

We tested our detector using the image sequences and testing software provided by Mikolajczyk [2] . The evaluation criterion is the *repeatability score*.

The test sequences comprise images of real textured and structured scenes. There are different types of geometric and photometric transformations, like changing viewpoints, zoom and rotation, image blur, lighting changes and JPEG compression.

In all experiments reported in this paper, the timings were measured on a standard PC Pentium IV, running at 3 GHz.

5.1.1. *SURF Detector*

We tested two versions of our *Fast-Hessian* detector, depending on the initial Gaussian derivative filter size. *FH-9* stands for our Fast Hessian detector with the initial filter size 9×9, and *FH-15* is the 15×15 filter on the double input image size version. Apart from this, for all the experiments shown in this section, the same thresholds and parameters were used.

The detector is compared to the Difference of Gaussians (DoG) detector by Lowe [24], and the Harris- and Hessian-Laplace detectors proposed by Mikolajczyk [29]. The number of interest points found is on average very similar for all detectors (see table 1 for an example). The thresholds were adapted according to the number of interest points found with the DoG detector.

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

The *FH-9* detector is more than five times faster than DoG and ten times faster than Hessian-Laplace. The *FH-15* detector is more than three times faster than DoG and more than four times faster than Hessian-Laplace (see also table 1). At the same time, the repeatability scores for our detectors are comparable or even better than for the competitors.

The repeatability scores for the Graffiti sequence (figure 16 top) are comparable for all detectors. The repeatability score of the *FH-15* detector for the Wall sequence (figure 16 bottom) outperforms 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 invariant to image rotation and scale. Hence, these deformations have to be accounted for by the overall robustness of the features. In the Boat sequence (figure 17 top), the *FH-15* detector shows again a better performance than the others. The *FH-9* and *FH-15* detectors are outperforming the others in the Bikes sequence (figure 17 bottom). The superiority and accuracy of our detector is further underlined in the sections 5.2 and 5.3.

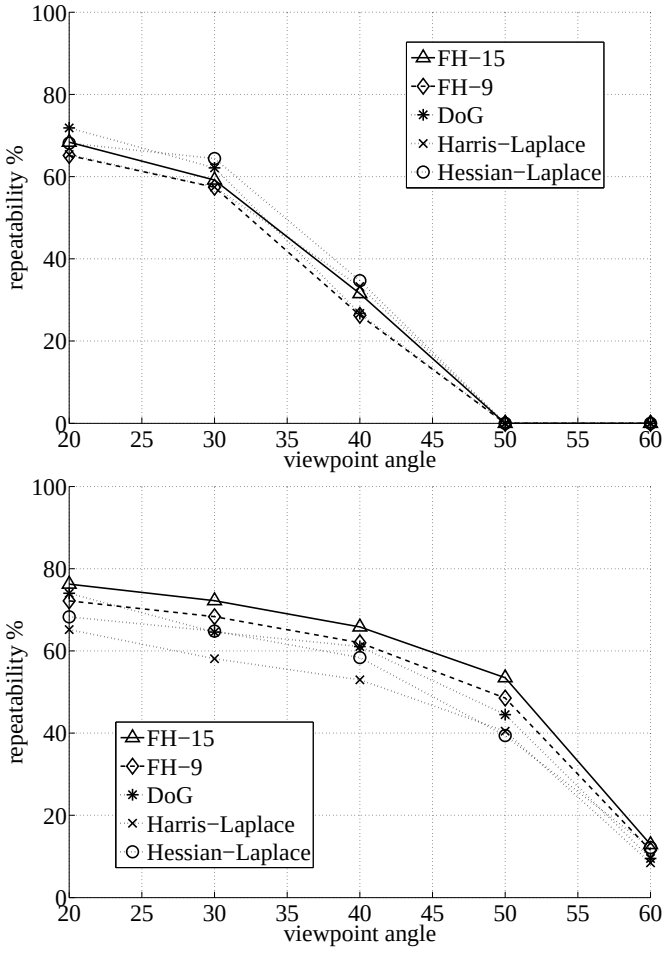

Fig. 16. Repeatability score for the Graffiti (top) and Wall (bottom) sequence (Viewpoint Change).

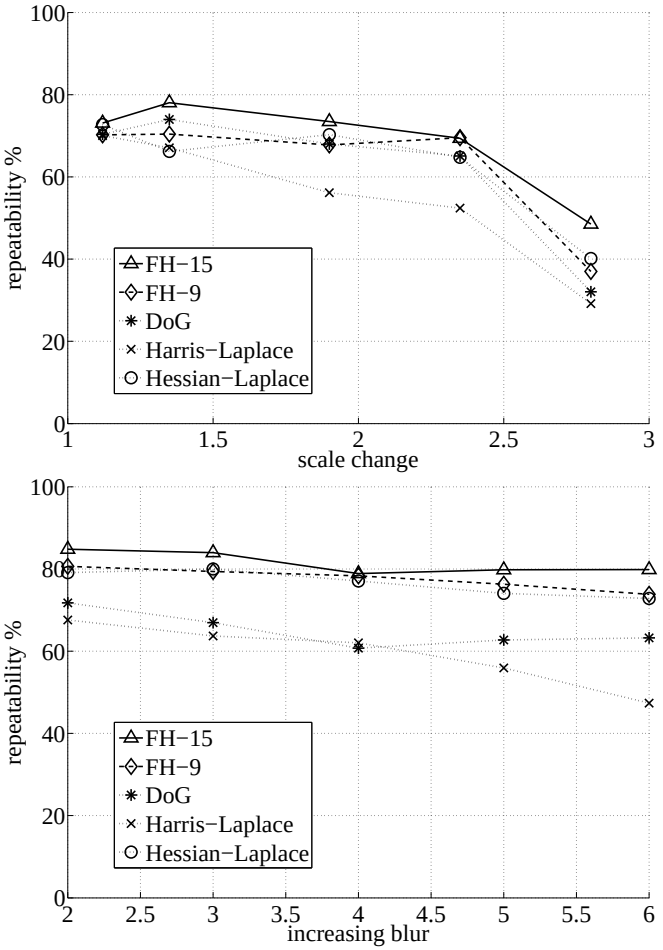

Fig. 17. Repeatability score for the Boat (top) and Bikes (bottom) sequence (Scale Change, Image Blur).

5.1.2. *SURF Descriptor*

Here, we focus on two options offered by the SURF descriptor and their effect on recall/precision.

Firstly, the number of divisions of the square grid in figure 12, and hence the descriptor size, has a major impact on the matching speed. Secondly, we consider the extended descriptor as described above. Figure 18 plots recall and precision against the side length of the square grid, both for the standard as well as the extended descriptor. Only the number of divisions is varied, not the actual size of the parent square. SURF-36 refers to a grid of 3×3, SURF-72 to its extended counterpart. Likewise, SURF-100 refers to 5×5 and SURF-144 to 6×6, with SURF-200 and SURF-288 their extended versions. To get averaged numbers over multiple images (we chose one pair from each set of test images), the ratio-matching scheme [24] is used.

Clearly, a square of size 4×4 performs best both with respect to recall and precision in both cases. Still, 3×3 is a viable alternative as well, especially when matching speed is of importance. From further analysis, we discovered that the extended descriptor loses with respect to recall, but exhibits better precision. Overall, the effect of the extended version is minimal.

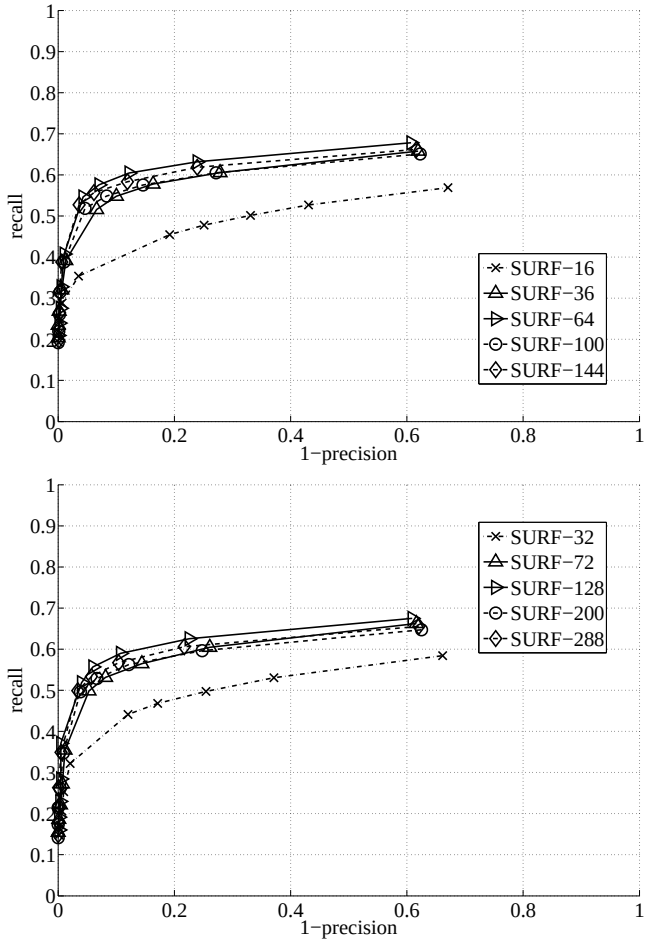

Fig. 18. Recall-precision for nearest neighbor ratio matching for varying side length of square grid. A maximum is attained for a square of 4×4. Figures averaged over 8 image pairs of Mikolajczyk's database. Top: standard descriptor, bottom: extended descriptor.

Extensive comparisons with other descriptors can be found in [4]. Here, we only show a comparison with two other prominent description schemes (SIFT [24] and GLOH [30]), again averaged over the test sequences (Figure 19). SURF-64 turns out to perform best.

Another major advantage of SURF is its low computation time: detection and description of 1529 interest points takes about 610 ms, the upright version U-SURF uses a mere 400 ms. (first Graffiti image; Pentium 4, 3 GHz)

5.2. *Application to 3D*

In this section, we evaluate the accuracy of our Fast-Hessian detector for the application of camera self-calibration and 3D reconstruction. The first evaluation compares different state-of-the-art interest point detectors for the two-view case. A known scene is used to provide some quantitative results. The second evaluation considers the N-view case for camera self-calibration and dense 3D reconstruction from multiple images, some taken under wide-baseline conditions.

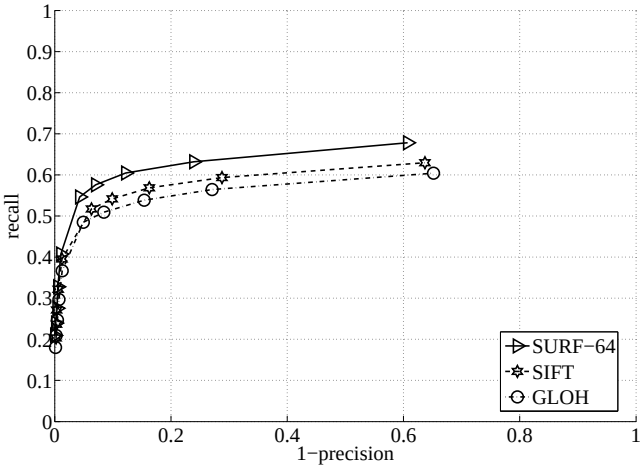

Fig. 19. Recall-precision for nearest neighbor ratio matching for different description schemes, evaluated on SURF keypoints. Figures averaged over 8 image pairs of Mikolajczyk's database.

5.2.1. *2-view Case*

In order to evaluate the performance of different interest point detection schemes for camera calibration and 3D reconstruction, we created a controlled environment. A good scene for such an evaluation are two highly textured planes forming a right angle (measured 88.6° in our case), see figure 20. The images are of size 800×600. Principal point and aspect ratio are known. As the number of correct matches is an important factor for the accuracy, we adjusted the interest point detectors' parameters so that after matching, we are left with 800 *correct matches* (matches not belonging to the angle are filtered). The SURF-128 descriptor was used for the matching step. The location of the two planes was evaluated using RANSAC, followed by orthogonal regression. The evaluation criteria are the angle between the two planes, as well as the mean distance and the variance of the reconstructed 3D points to their respective planes for different interest point detectors.

Table 2 shows these quantitative results for our two versions of the Fast-Hessian detector (FH-9 and FH-15), the DoG features of SIFT [24], and the Hessian- and Harris-Laplace detectors proposed by Mikolajczyk and Schmid [29]. The FH-15 detector clearly outperforms its competitors.

Figure 21 shows the orthogonal projection of the Fast-Hessian (FH-15) features for the reconstructed angle. Interestingly, the theoretically better founded approaches like the Harris- and Hessian-Laplace detectors perform worse than the approximations (DoG and the SURF features).

5.2.2. *N-view Case*

The SURF detection and description algorithms have been integrated with the Epoch 3D Webservice of the VISICS research group at the K.U. Leuven [3]. This webservice allows users to upload sequences of still images

[3] `http://homes.esat.kuleuven.be/~visit3d/webservice/html`

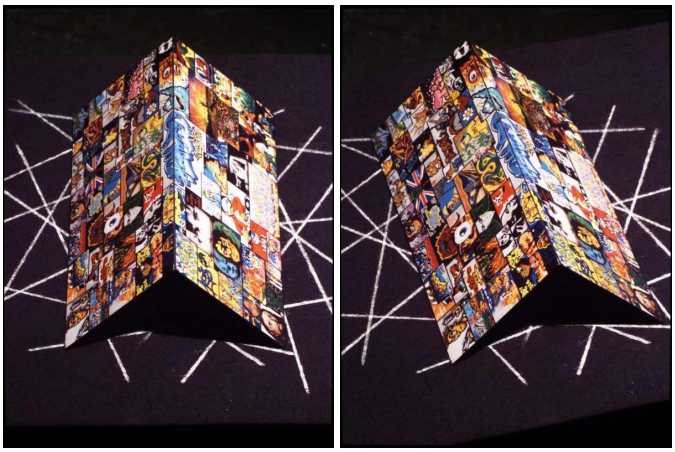

Fig. 20. Input images for the quantitative detector evaluation. This represents a good scene choice for the comparison of different types of interest point detectors, as its components are simple geometric elements.

Fig. 21. Orthogonal projection of the reconstructed angle shown in figure 20.

to a server. There, the calibration of the cameras and dense depth maps are computed automatically using these images only [40]. During the camera calibration stage, features need to be extracted and matched between the images. The use of SURF features improved the results of this step for many uploaded image sets, especially when the images were taken further apart. The previous procedure using Harris corners and normalised cross correlation of image windows has problems matching such wide-baseline images. Furthermore, the DoG detector combined with SIFT description failed on some image sequences, where SURF succeeded to calibrate all the cameras accurately.

For the example in figure 22, the traditional approach managed to calibrate only 6 from a total of 13 cameras. Using SURF however, all 13 cameras could be calibrated. The vase is easily recognisable even in the sparse 3D model.

Figure 23 shows a typical wide-baseline problem: three

detector	angle	mean dist	std-dev
FH-15:	88.5°	1.14 px	1.23 px
FH-9:	88.4°	1.64 px	1.78 px
DoG:	88.9°	1.95 px	2.14 px
Harris-Laplace:	88.3°	2.13 px	2.33 px
Hessian-Laplace:	91.1°	2.85 px	3.13 px

Table 2
Comparison of different interest point detectors for the application of camera calibration and 3D reconstruction. The true angle is 88.6°.

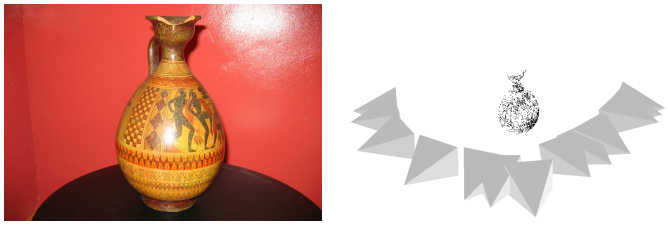

Fig. 22. 3D reconstruction with KU-Leuven's 3D webservice. Left: One of the 13 input images for the camera calibration. Right: Position of the reconstructed cameras and sparse 3D model of the vase.

images, taken from different, widely separated view points. It is a challenging example, as three images represent the absolute minimum number of images needed for an accurate dense 3D reconstruction. The obtained 3D model can be seen in figure 23 (bottom). In general, the quality of the camera calibration can best be appreciated on the basis of the quality of such resulting dense models. These experiments confirm the use of the SURF detector/descriptor pair for applications in image registration, camera calibration, and 3D reconstruction, where the accuracy of the correspondences is vital.

5.3. *Application to Object Recognition*

Bay *et al.* [3] already demonstrated the usefulness of SURF in a simple object detection task. To further illustrate the quality of the descriptor in such a scenario, we present some further experiments. Basis for this was a publicly available implementation of two bag-of-words classifiers [10]. [4] Given an image, the task is to identify whether an object occurs in the image or not. For our comparison, we considered the naive Bayes classifier, which works directly on the bag-of-words representation, as suggested by Dance *et al.* [8]. This simple classifier was chosen as more complicated methods like pLSA might wash out the actual effect of the descriptor. Similar to [10], we executed our tests on 400 images each from the Caltech background and airplanes set [5]. 50% of the images are used for training, the other 50% for testing. To minimise the influence of the partitioning, the same random permutation of training and test sets was chosen for all descriptors. While this is a rather simple test set for object recognition in general, it definitely serves the purpose of comparing the performance of the actual descriptors.

The framework already provides interest points, chosen randomly along Canny edges to create a very dense sampling. These are then fed to the various descriptors. Additionally, we also consider the use of SURF keypoints, generated with a very low threshold, to ensure good coverage.

Figure 24 shows the obtained ROC curves for SURF-128, SIFT and GLOH. Note that for the calculation of SURF, the sign of the Laplacian was removed from the descriptor. For both types of interest points, SURF-128 outperforms

[4] http://people.csail.mit.edu/fergus/iccv2005/bagwords.html
[5] http://www.vision.caltech.edu/html-files/archive.html

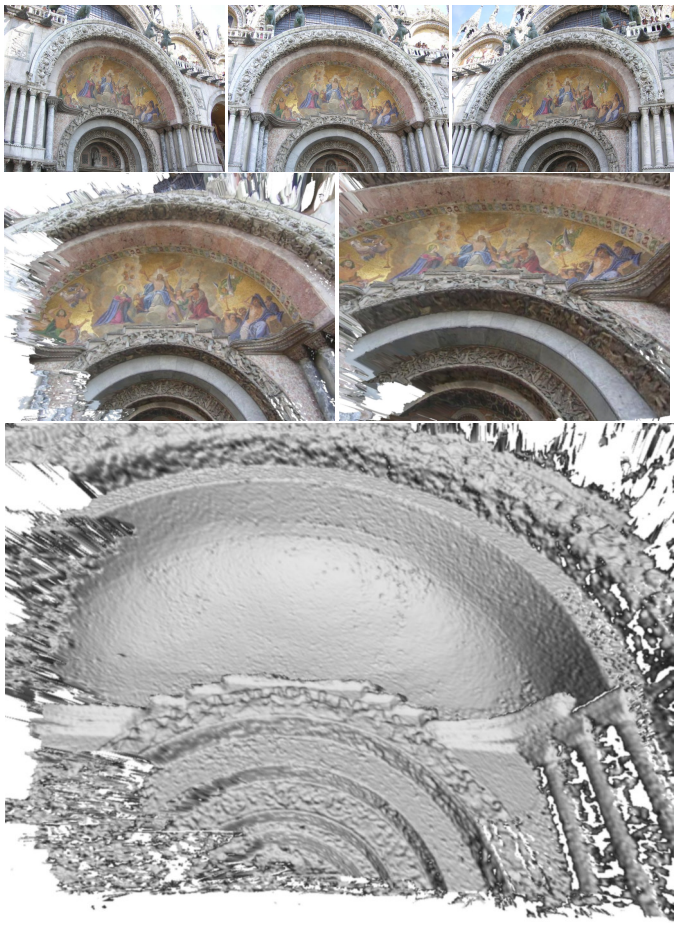

Fig. 23. 3D reconstruction with KU-Leuven's 3D webservice. Top row: The 3 input images of a detail of the San Marco Cathedral in Venice. Middle row: Samples of the textured dense reconstruction. Bottom row: un-textured dense reconstruction. The quality of the dense 3D model directly reflects the quality of the camera calibration. The images were taken by Maurizio Forte, CNR-ITABC, Rome).

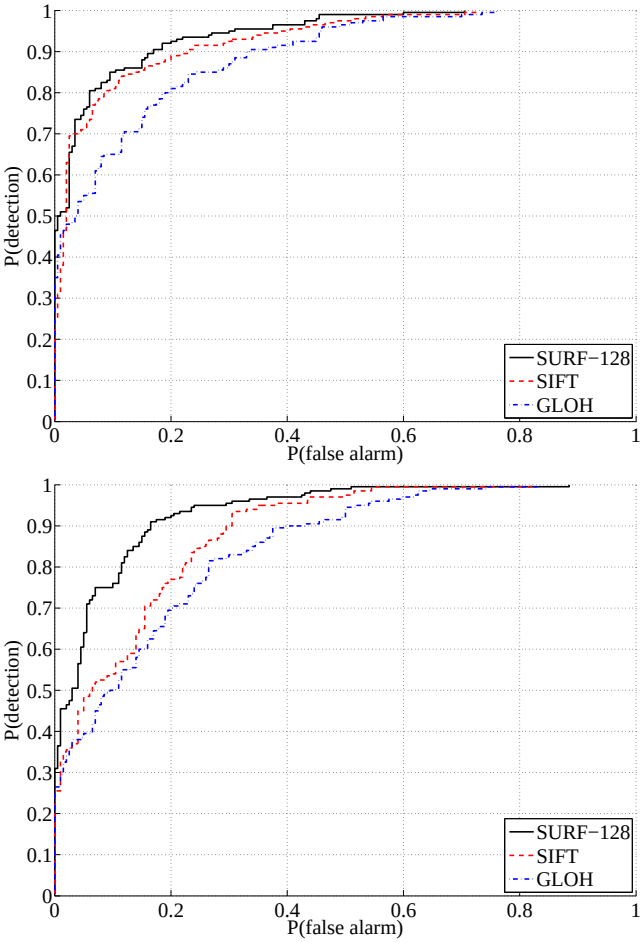

Fig. 24. Comparison of different descriptor strategies for a naive Bayes classifier working on a bag-of-words representation. Top: descriptors evaluated on random edge pixels. Bottom: on SURF keypoints.

its competitors on the majority of the curve significantly. Figure 25 investigates the effect of the index size and the extended descriptor of SURF. As can be seen, the upright counterparts for both SURF-128 and SURF-64 perform best. This makes sense, as basically all the images in the database were taken in an upright position. The other alternatives perform only slightly worse, but are comparable. Even SURF-36 exhibits similar discriminative power, and provides a speed-up for the various parts of the recognition system due to its small descriptor size.

The same tests were also carried out for the Caltech motorcycles (side) and faces dataset, yielding similar results.

In conclusion, SURF proves to be work great for classification tasks, performing better than the competition on the test sets, while still being faster to compute. These positive results indicate that SURF should be very well suited for tasks in object detection, object recognition or image retrieval.

6. Conclusion and Outlook

We presented a fast and performant scale and rotation-invariant interest point detector and descriptor. The important speed gain is due to the use of integral images, which drastically reduce the number of operations for simple box convolutions, independent of the chosen scale. The results showed that the performance of our Hessian approximation is comparable and sometimes even better than the state-of-the-art interest point detectors. The high repeatability is advantageous for camera self-calibration, where an accurate interest point detection has a direct impact on the accuracy of the camera self-calibration and therefore on the quality of the resulting 3D model.

The most important improvement, however, is the speed of the detector. Even without any dedicated optimisations, an almost real-time computation without loss in performance is possible, which represents an important advantage for many on-line computer vision applications.

Our descriptor, based on sums of Haar wavelet components, outperforms the state-of-the-art methods. It seems

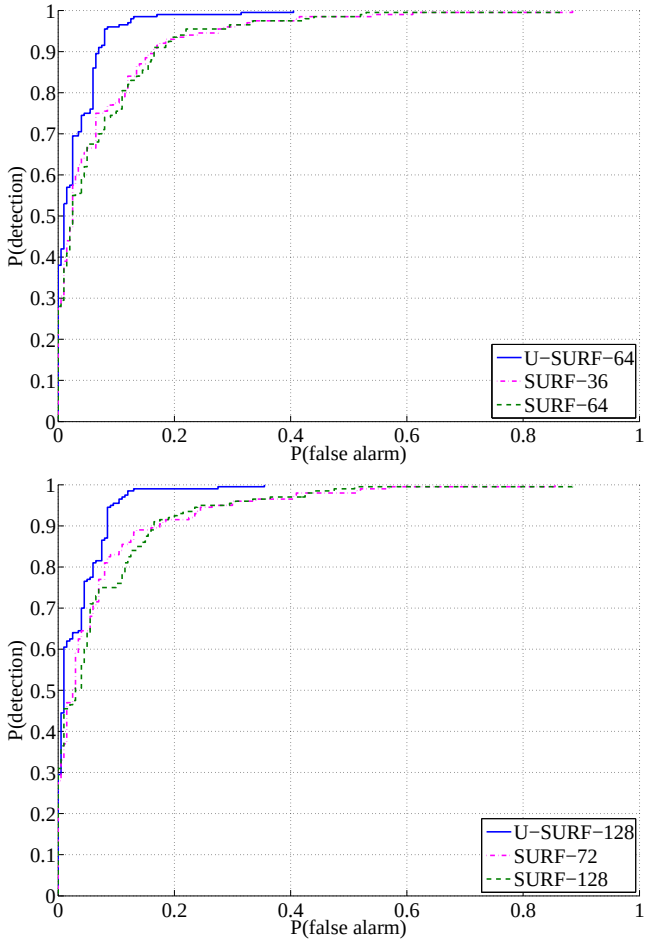

Fig. 25. Comparison of different options for the SURF descriptor for a naive Bayes classifier working on a bag-of-words representation. The descriptor was evaluated on SURF keypoints. Top: standard, bottom: extended descriptor.

that the description of the nature of the underlying image-intensity pattern is more distinctive than histogram based approaches. The simplicity and again the use of integral images make our descriptor competitive in terms of speed. Moreover, the Laplacian-based indexing strategy makes the matching step faster without any loss in terms of performance.

Experiments for camera calibration and object recognition highlighted SURF's potential in a wide range of computer vision applications. In the former, the accuracy of the interest points and the distinctiveness of the descriptor showed to be major factors for obtaining a more accurate 3D reconstruction, or even getting any 3D reconstruction at all in difficult cases. In the latter, the descriptor generalises well enough to outperform its competitors in a simple object recognition task as well.

The latest version of SURF is available for public download. [6]

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

Acknowledgments

The authors gratefully acknowledge the support from Toyota Motor Europe and Toyota Motor Corporation, the Swiss SNF NCCR project IM2, and the Flemish Fund for Scientific Research. Geert Willems for identifying a bug in the interpolation used in the descriptor. Stefan Saur for porting SURF to Windows. Maarten Vergauwen for providing us with the 3D reconstructions. Dr. Bastian Leibe for numerous inspiring discussions. The DoG detector was kindly provided by David Lowe.

References

[1] A. Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 774 – 781, 2000.

[2] H. Bay. *From Wide-baseline Point and Line Correspondences to 3D*. PhD thesis, ETH Zurich, 2006.

[3] H. Bay, B. Fasel, and L. van Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *Proceedings of the first international workshop on mobile vision*, May 2006.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[5] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC*, 2002.

[6] G. Carneiro and A.D. Jepson. Multi-scale phase-based local features. In *CVPR (1)*, pages 736 – 743, 2003.

[7] P. C. Cattin, H. Bay, L. Van Gool, and G. Székely. Retina mosaicing using local features. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, October 2006. in press.

[8] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision, Prague*, 2004.

[9] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, pages 253–262, 2004.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR (2)*, pages 264–271, 2003.

[11] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *JMIV*, 4(2):171–187, 1994.

[12] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891 – 906, 1991.

[13] J. Goldstein, J. C. Platt, and C. J. C. Burges. Redundant bit vectors for quickly searching high-dimensional regions. In *Deterministic and Statistical Methods in Machine Learning*, pages 137–158, 2004.

[14] M. Grabner, H. Grabner, and H. Bischof. Fast approximated sift. In *ACCV (1)*, pages 918–927, 2006.

[15] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147 – 151, 1988.

[16] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, volume II, pages 90 – 96, 2004.

[17] T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 45(2):83 – 105, 2001.

[18] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506 – 513, 2004.

[19] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363 – 370, 1984.

[20] T. Lindeberg. Scale-space for discrete signals. *PAMI*, 12(3):234–254, 1990.
[21] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79 – 116, 1998.
[22] T. Lindeberg and L. Bretzner. Real-time scale selection in hybrid multi-scale representations. In *Scale-Space*, pages 148–163, 2003.
[23] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
[24] D. Lowe. Distinctive image features from scale-invariant keypoints, cascade filtering approach. *IJCV*, 60(2):91 – 110, January 2004.
[25] J. Matas, O. Chum, Urban M., and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384 – 393, 2002.
[26] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, volume 1, pages 525 – 531, 2001.
[27] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, pages 128 – 142, 2002.
[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, volume 2, pages 257 – 263, June 2003.
[29] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63 – 86, 2004.
[30] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.
[31] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005.
[32] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *CVIU*, 94(1-3):3–27, 2004.
[33] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *ICPR*, 2006.
[34] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR (2)*, pages 2161–2168, 2006.
[35] S.M. Omohundro. Five balltree construction algorithms. Technical report, ICSI Technical Report TR-89-063, 1989.
[36] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *ECCV*, volume 1, pages 414 – 431, 2002.
[37] S. Se, H.K. Ng, P. Jasiobedzki, and T.J. Moyung. Vision based modeling and localization for planetary exploration rovers. *Proceedings of International Astronautical Congress*, 2004.
[38] P. Simard, L. Bottou, P. Haffner, and Y. LeCun. Boxlets: A fast convolution algorithm for signal processing and neural networks. In *NIPS*, 1998.
[39] T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In *BMVC*, pages 412 – 422, 2000.
[40] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Special Issue of Machine Vision and Applications*, to appear.
[41] P.A. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511 – 518, 2001.