RESEARCH ARTICLE OPEN ACCESS

Enhanced Approximated SURF Model For Object Recognition

S. Sangeetha*, M. Thanga Kavitha*, N. Manikandaprabu**

*(PG Scholar,Akshaya College Of Engineering And Technolgy, TN, India)
** (Lecturer, Department of ECE, Senthur Polytechnic College, TN, India)

ABSTRACT
Computer vision applications like camera calibration, 3D reconstruction, and object recognition and image registration are becoming widely popular now a day. In this paper an enhanced model for speeded up robust features (SURF) is proposed by which the object recognition process will become three times faster than common SURF model The main idea is to use efficient data structures for both, the detector and the descriptor. The detection of interest regions is considerably speed-up by using an integral image for scale space computation. The descriptor which is based on orientation histograms is accelerated by the use of an integral orientation histogram. We present an analysis of the computational costs comparing both parts of our approach to the conventional method. Extensive experiments show a speed-up by a factor of eight while the matching and repeatability performance is decreased only slightly.
Keywords **-** Feature extraction, SURF, style, styling.

I. INTRODUCTION

Extraction of features and discern information from images is one of the main aim of computer vision. Even though it can fulfill other purposes its use is well-known in near real-time applications like robot maneuvering or object tracking. Information extraction from images can resolve many issues for example; photogrammetry where geometric and geographic information are extracted from images is made by computer vision algorithms. Selecting out only the most important things of an image that can be localized repeatedly multiple images subsequently reduces the burden of data processing. However, feature extraction faces major bottlenecks for many of its implementations. For example, accurate GPS-denied visual navigation on moving vehicles requires 30 Hz frame rates on large images [2]. If the speed of feature extraction is improved it reduces the weight, size, and power demands of these systems, reducing the cost of deployment. Here comes the need implementing the most accurate extraction algorithm on readily available commercial hardware.

Since features can be viewed from different angles, distances, and illumination, it is important that a feature descriptor be relatively invariant to changes in orientation, scale, brightness, and contrast, while remaining descriptive enough to be correctly matched against a pool of thousands of candidates. We chose the Speeded-Up Robust Features (SURF) descriptor proposed by [1] and described in Section II. This produces descriptors half the size of previous algorithms, such as the Scale-Invariant Feature Transform (SIFT) [3], while retaining the same matching performance. Smaller feature vectors increase the speed of subsequent matching operations, while themselves being less expensive to compute. However, SURF cannot yet achieve interactive frame rates on a traditional CPU.

In this paper we propose a modified SIFT method for recognition purpose. Our primary motivation is to significantly speed up the SIFT computation while at the same time keep the excellent matching performance. We demonstrate that by using approximations (mainly employing integral images) both the DoG detector (see section 2) and the SIFT-descriptor (see section 3) we can speed-up the SIFT computation by at least a factor of eight compared to the binaries provided by Lowe. Extensive experimental evaluations (see section 4) show that the loss in matching performance is negligible.

II. RELATED WORK

A. 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 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-localized structures. Also, (2) approximations like the DoG can bring speed at a low cost in terms of lost accuracy.

B. 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 neighborhood. 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 localization 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 Hessian 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 neighborhood. 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 IΣ(x) at a location x = (x, y) represents the sum of all pixels in the input image I of a rectangular region formed by the point x and the origin, $I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i, j)$ With $I_{\Sigma}(x)$ calculated, it only takes four additions to calculate the sum of the intensities over any upright, rectangular area, independent of its size.

III. PROPOSED METHODOLOGY

A. Dog detector

In order to detect scale invariant key-points Lowe suggests repeatedly smoothing the input image and identifying key locations in scale space. In order to detect even very small scales Lowe extends this approach and proposes to double the input image before building the scale space. The different scale levels are produced by recursive filtering with a variable-scale Gaussian kernel. A local maxima search is finally applied to the

Difference-of-Gaussian images which can be computed of adjacent scale images, in order to detect key-points in scale space.

TABLE I. MAJOR DIFFERENCES BETWEEN ORIGINAL SURF DETECTOR AND OUR PROPOSED APPROACH.

SURF	**Enhanced SURF Model**
Image doubling	-
-	Calculate integral image
DoG scale space	DoM scale space
Post-processing	-

To accelerate this approach we propose several approximations and changes see Table 1. The key idea of our method is to considerably reduce the costs for computing the scale space by using Difference-of-Mean (DoM) images instead of Difference-of-Gaussians (DoG). This DoM images can be computed very efficiently by using a box filter in combination with an integral image as introduced by Viola and Jones [11] (capturing the main idea of [12]). Once the integral image is computed, it allows to compute the mean within a rectangular region in constant time independent of the size of the region. This property allows fast box filtering and can be used for linear sampling of the scale axis which is realized by successively increasing the size of the filter kernel. Adjacent scale space images are subtracted and a local maxima search is applied to the Difference-of-Mean images in order to detect key-points. For a reliable detection of key-points at all scales it is important to normalize the DoM response with

$$sensitivity \cdot \left(1 - \frac{s_1^2}{s_2^2}\right)$$

where s_1, s_2 corresponds to the size of the small and larger box filter, respectively. The parameter sensitivity captures the minimal contrast of the mean gray values of the inner region (s_1) and the outer region ($s_2 - s_1$) and can be used to adjust the sensitivity of the detector. Since experiments with DoG indicate that small scales cannot be reliably matched we skip the doubling of the image size, which again provides a significant speed-up. Once the key-points have been detected we do not make any further post-processing like an accurate key-point localization because due to the use of integral images we have already pixel accuracy at each scale. But note that the accuracy of the obtained points is not as precise as with the DoG, nevertheless the detected points are good for recognition tasks but less suitable for geometric tasks like estimation of the fundamental matrix.

1) Computational costs: The box filtering approach using integral images is depicted in Algorithm 1. Once the integral image is pre-computed which takes 2 additions for each image pixel, a single box filter response can be computed, independent of its size, with 4 memory accesses, 3 additions and a single multiplication which is needed for normalizing the box region.

a) Algorithm 1 Integral image computation:

// pre-computation
for each image point do
Propagate integral image *{1 addition}*
Increase value *{1 addition}*
end for
// apply box filter with a given kernel size
for each image point do
Compute intersection *{3 addition}*
Normalize *{1 multiplication}*
end for

In Table 2 which has been adapted from [13], we compare the box filtering approach to other commonly used Gaussian filtering techniques. Simple 2-D convolution is the slowest one since the complexity

for each pixel is $O(N_2)$, where N corresponds to the filter size. Much more efficient is to make use of the separability of the Gaussian function which allows convolution by applying two passes of the 1-D function in the horizontal and vertical directions. This leads to linear costs in the kernel size N. Other methods like FFT are independent with respect to the filter kernel size but depend on the size of the input image W ×H. However, as can be seen in Figure 2(a), the computational costs are higher than for the separable Gaussian for a kernel size of 7×7 (as proposed by Lowe in [14]). A similar result holds for recursive Gaussian filters which allow convolution in constant time but are still computationally more demanding for small filter kernels.

B. SURF Descriptor

Reliable matching of key-points is performed by feature vectors generated from their local neighborhoods. Lowe suggests to use the gradient information around that the descriptor can be represented relative to this orientation, thereby achieving rotation invariance. Gradients within a circular region are used to compute an orientation histogram, and local maxima in the histogram are used as characteristic orientations.

TABLE II. COMPARISON OF VARIOUS FILTERING TECHNIQUES (CALCULATIONS PER PIXEL)

FILTER TECHNIQUE	ADDITIONS	MULTIPLICATIONS
2D-Gauss	N^2	N^2-1
Recursive Gauss	2· N-1	N+1
Separate Gauss	6	14
FFT	2 · log(W · H)	2 · log(W · H) +1
Box Filter	2+3	1

To obtain a descriptor Lowe proposes to divide the surrounding region into 4 × 4 sub-patches. From each sub-patch an orientation histogram with 8 bins is computed and concatenated to form a single feature vector. Since orientation histograms form the basic computation for the descriptor this leads to the idea to use integral histograms [15]. Integral histograms are an extension of integral images using for each histogram bin (e.g. orientation) a separate integral image. Once the integral orientation histogram is computed, histograms can be accessed in constant time independent of the size of the region. Similar to integral images integral histograms can only provide histograms of rectangular regions.

For orientation histogram computation we use un-weighted squared regions. Furthermore, for the descriptor we rotate the midpoints of each sub-patch relative to the orientation and compute the histograms of overlapping sub-patches without aligning the squared region but shifting the sub-patch histogram relative to the main orientation. The main advantage of our method is that we make use of the full resolution of the input image without additional computational costs.

1) Computational costs: The major question is how many descriptors have to be calculated in order to obtain a speed up for the integral version compared to the conventional approach. We define the costs for single histogram computation for both approaches which has been done by adapting the analysis from [15]. We assume that the gradient image has already been computed. In addition we assume computing histograms only over squared regions.

a) Algorithm 2 Conventional histogram computation:

```
//histogram computation
for each histogram do
for each gradient within window do
Find bin { 1 multiplication}
Increase bin value { 1 addition}
end for
end for
```

The conventional method for histogram computation is given in Algorithm 2. Once the gradient image is available, for each gradient in the observed region an assignment to the correct bin value must be done.

Consequently the conventional method strongly depends on the number of gradients contributing to the histogram which leads to the complexity O(N2) for a squared region where N corresponds to the window size. In addition the computational costs for a squared region is

$$k \,.\, N^2 .\, (c_{add} + c_{mult})$$

where k corresponds to the number of histograms, c_{add} represent costs for an addition and c_{mult} are the costs for a multiplication.

a) Algorithm 3 Integral histogram computation:

```
//pre-computation
for each gradient do
for each bin do
Propagate integral histogram { 1 addition}
end for
Find bin { 1 multiplication}
Increase bin value { 1 addition}
end for
//histogram computation
foreach histogram do
for each bin do
Compute intersection { 3 additions}
end for
end for
```

Considering the integral histogram computation illustrated in Algorithm 3, we see that equivalent to integral images some pre-computations have to be done. Once the integral orientation histogram has been computed, orientation histograms can accessed in $k \cdot b \cdot 3 \cdot c_{add}$, where b corresponds to the number of bins (in our case 16 bins are used). Similar to integral images rectangular regions can be accessed. The costs for histogram computation does not depend on the number of gradients within a region. Consequently the total costs including the computation of the integral orientation histogram can be written as

$$W \,.\, H. \,(b \,.\, c_{add} + c_{add} + c_{mult}) + k \,.\, b \,.\, 3 \,.\, c_{add}$$

where $W \times H$ represents the input image size.

Figure 2(b) compares standard histogram and integral histogram computation, where we have used relative costs for additions and multiplications from [15] (addition:1 - multiplication:4). Other parameters of the cost functions, such as the histogram patch size, have been experimentally determined. As we can see in Figure 2(b), initially the costs for the integral histogram are much higher however once the integral image is computed the costs increase very slowly. In contrast the costs of the conventional method increase linearly with the number of computed descriptors. Integral orientation histograms are profitable especially when calculated over large regions. This is especially suited for our approach because we always compute the descriptors on the original resolution. Consequently, we take advantage of using the whole information of the input image.

IV. EXPERIMENTAL RESULTS

We compare our novel approach to Lowe's method with respect to performance and speed. For matching performance we run two types of experiments to explore the effects of the approximations made in our approach. First, both methods are examined with respect to rotation, scale and perspective invariance on a data-set of 15 commonly used images. Secondly we compare the runtime of our approach to Lowe's publicly available binaries 1.

A. Artificial Transformations

For all artificial transformations we used the same criterions for determining repeatability of the detector and the matching score of the descriptor. The repeatability is obtained through a simple location criterion while

for the matching score a key-point match and the corresponding nearest descriptor match is required. In Fig. 12, each line between two images indicates a pair of corresponding feature points.

Fig 1.Features extracted using Approximated SURF from an image pair matching point for normal case. Test images at (Top) the best case, (Bottom) the normal case

Due to the box filter approximation the rotation is the worst case scenario for the detector. Even for the descriptor the worst case because no rotational sampling is done. Therefore we artificially rotate each image from 0°to 90°of our data-set with steps of 15°. In Figure 3 we see that both, the detector and the descriptor of the approximated SURF implementation behave worst at a rotation of 45°. However, at the same time the performance is not much worse to SIFT. The strong performance decrease of SURF can be explained by the fact that the small scale key-points are lost because of the smoothing effect after the bilinear transformation.

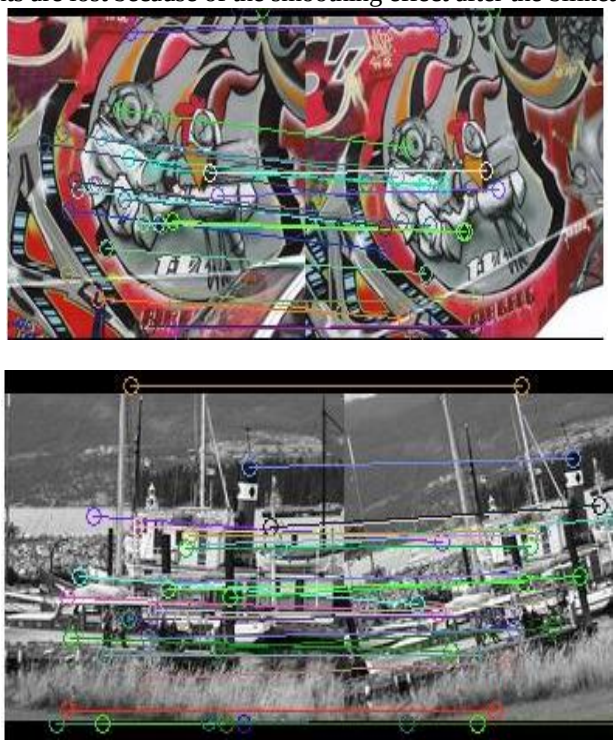

Fig 2.Matching point for Worst cases.

Second, scale invariance is tested. As a reference image we used a downscaled image (0.8) in order to have scale changes in both directions. Figure 4(a) shows that our approach which passes on detecting key-points with small scales performs slightly better than SURF.

Finally, we examined the repeatability of the detector and the matching of the descriptor by generating different projective transformations of the image. Again the results in Figure 4(b) show good performance for the approximated SURF implementation.

B. Speed

We have a non-optimized C++ implementation of the approximated SURF which has been compared to the SURF binaries provided by Lowe.

IMAGE SIZE	SURF	Approximated SURF
800 x 640	4.05 s	0.515 s
400 x 320	1.45 s	0.710 s
200 x 160	0.55 s	0.044 s

In Table 3 the processing times for feature detection of different image sizes are listed. This experiment was done on a Pentium 4 with 3.2 GHz. Results show that approximated SURF provides a speed-up of a factor 8 with this non optimized implementation where the major benefit is obtained in the detection process. Optimizing the implementation we expect to achieve at least a factor 12 to 16.

V. CONCLUSION

In this paper we have presented a novel approximation of the SURF model that achieves a considerable speed-up of the original method (at least a factor of eight using our non -optimized C++ implementation) while at the same time achieving comparable matching performance. We have carefully analyzed the speed-up gain theoretically and have performed extensive experimental evaluations.

This new fast SURF variant opens several venues of further research which we are currently investigating. Once we have calculated the integral images the costs for the descriptor calculation is negligible. Therefore, we can perform a local neighbor search around a key-point for more discriminative /reliable descriptors. This should further increase the matching performance. Having such a fast method, tracking using SURF becomes feasible. This should result in highly robust trackers. Another idea that is currently investigated is to use SURF in an Adaboost framework. This has already been proposed by Zhang et al. [16], but having a fast SURF will considerably speed-up the training process.

REFERENCES

[1] D.G. Lowe, D.G, “Distinctive image features from scale-invariant keypoints” IJCV 60 (2004) 91–110
[2] K. Mikolajczyk, C. Schmid, “Indexing based on scale invariant interest points,” In:Proc. ICCV, Vancouver, Canada (2001) 525–531
[3] K. Mikolajczyk, C. Schmid, “Scale & affine invariant interest point detectors,” In:Proc. CVPR. Number 1 (2004) 63–86
[4] J. Matas, O. Chum, M. Urban, T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” In: Proc. BMVC. (2002)

[5] S. Se, D. Lowe, J. Little, " Local and global localization for mobile robots using visual landmarks", In: Proc. IROS. Volume 2. (2001) 414–420
[6] Y. Ke, R. Sukthankar, L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system", In: Proc. Int. Conf. on Multimedia. (2004) 869–876.
[7] T. Tuytelaars, L.V Gool, " Content-based image retrieval based on local affinely invariant regions," In: Proc. Int. Conf. on Visual Information and Information Systems. (1999) 493–500.
[8] M. Brown, D. Lowe, " Recognising panoramas," In: Proc. ICCV. (2003) 1218–1225.
[9] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors", IEEE Trans. PAMI 27 (2005) 1615–1630
[10] Y. Ke, R. Sukthankar, " PCA-SIFT: A more distinctive representation for local image descriptors", In: Proc. CVPR. Volume 2. (2004) 506–513.
[11] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features", In: Proc. CVPR. Volume I. (2001) 511–518.
[12] P. Simard, L. Bottou, P. Haffner, Y. LeCun, "Boxlets: A fast convolution algorithm for signal processing and neural networks", In: Proc. NIPS. (1998) 571–577
[13] J. Geusebroek, A. Smeulders, J. van de Weijer,"Fast anisotropic Gauss filtering. In: Proc. ECCV. (2002) 99–112
[14] D. Lowe, "Object recognition from local scale-invariant features", In: Proc. ICCV.Volume 2. (1999) 1150–1157
[15] F. Porikli, "Integral histogram", A fast way to extract histograms in cartesian spaces. In: Proc. CVPR. Volume 1. (2005) 829–836
[16] W. Zhang, B. Yu, G. Zelinsky, D. Samaras, " Object class recognition using multiple layer boosting with heterogeneous features", In: Proc. CVPR. Volume 2. (2005) 323–330.