CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching*

Motilal Agrawal[1], Kurt Konolige[2], and Morten Rufus Blas[3]

[1] SRI International, Menlo Park CA 94025, USA
agrawal@ai.sri.com
[2] Willow Garage, Menlo Park CA 94025, USA
konolige@willowgarage.com
[3] Elektro/DTU University, Lyngby, Denmark
mrb@elektro.dtu.dk

Abstract. We explore the suitability of different feature detectors for the task of image registration, and in particular for visual odometry, using two criteria: stability (persistence across viewpoint change) and accuracy (consistent localization across viewpoint change). In addition to the now-standard SIFT, SURF, FAST, and Harris detectors, we introduce a suite of scale-invariant center-surround detectors (CenSurE) that outperform the other detectors, yet have better computational characteristics than other scale-space detectors, and are capable of real-time implementation.

1 Introduction

Image matching is the task of establishing correspondences between two images of the same scene. This is an important problem in Computer Vision with applications in object recognition, image indexing, structure from motion and visual localization – to name a few. Many of these applications have real-time constraints and would benefit immensely from being able to match images in real time.

While the problem of image matching has been studied extensively for various applications, our interest in it has been to be able to reliably match two images in real time for camera motion estimation, especially in difficult off-road environments where there is large image motion between frames [1,2]. Vehicle dynamics and outdoor scenery can make the problem of matching images very challenging. The choice of a feature detector can have a large impact in the performance of such systems.

We have identified two criteria that affect performance.

- Stability: the persistence of features across viewpoint change
- Accuracy: the consistent localization of a feature across viewpoint change

* This material is based upon work supported by the United States Air Force under Contract No. FA8650-04-C-7136. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 102–115, 2008.

Stability is obviously useful in tracking features across frames. Accuracy of feature localization is crucial for visual odometry tasks, but keypoint operators such as SIFT typically subsample the image at higher scales, losing pixel-level precision.

Broadly speaking, we can divide feature classes into two types. *Corner detectors* such as Harris (based on the eigenvalues of the second moment matrix [3,4]) and FAST [5] (analysis of circular arcs [6]) find image points that are well localized, because the corners are relatively invariant to change of view. Both these detectors can be implemented very efficiently and have been used in structure-from-motion systems [2,7,8] because of their accuracy. However, they are not invariant to scale and therefore not very stable across scale changes, which happen constantly with a moving camera. The Harris-Laplace and the Hessian-Laplace features [9] combine scale-space techniques with the Harris approach. They use a scale-adapted Harris measure [10] or the determinant of the Hessian to select the features and the Laplacian to select the scale. Supposedly, visual odometry can benefit from scale-space features, since they can be tracked for longer periods of time, and should lead to improved motion estimates from incremental bundle adjustment of multiple frames.

While we expect scale-space features to be more stable than simple corner features, are they as accurate? The answer, at least for visual odometry, is "no". The reason is that, as typically implemented in an image pyramid, scale-space features are not well localized at higher levels in the pyramid. Obviously, features at high levels have less accuracy relative to the original image. The culprit in loss of accuracy is the image pyramid. If the larger features were computed at each pixel, instead of reducing the size of the image, accuracy could be maintained. However, computing features at all scales is computationally expensive, which is why SIFT features [11], one of the first scale-space proposals, uses the pyramid – each level incurs only 1/4 the cost of the previous one. SIFT attempts to recover some of the lost accuracy through subpixel interpolation.

Our proposal is to maintain accuracy by computing features at all scales *at every pixel* in the original image. The extrema of the Laplacian across scale have been shown to be very stable [12], so we consider this operator, or more generally, extrema of a center-surround response (CenSurE, or *Cen*ter *Sur*round *E*xtrema). We explore a class of simple center-surround filters that can be computed in time independent of their size, and show that, even when finding extrema across all scales, they are suitable for real-time tasks such as visual odometry. CenSurE filters outperform the best scale-space or corner features at this task in terms of track length and accuracy, while being much faster to compute; and they are also competitive in standard tests of repeatability for large-viewpoint changes.

While the main focus of this paper is on a novel feature detector, visual odometry (and other motion estimation tasks) can benefit from matching using a descriptor that is robust to viewpoint changes. In this paper, we develop a fast variant of the upright SURF descriptor, and show that it can be used in real-time tasks.

1.1 Related Work

The two scale-space detectors that are closest to our work, in technique and practicality, are SIFT [11] and SURF [13]. The main differences between approaches is summarized in the table below.

	CenSurE	**SIFT**	**SURF**
Spatial resolution at scale	full	subsampled	subsampled
Scale-space operator Approximation	Laplace (Center-surround)	Laplace (DOG)	Hessian (DOB)
Edge filter	Harris	Hessian	Hessian
Rotational invariance	approximate	yes	no

The key difference is the full spatial resolution achieved by CenSurE at every scale. Neither SIFT nor SURF computes responses at all pixels for larger scales, and consquently do not detect extrema across all scales. Instead, they consider each scale octave independently. Within an octave, they subsamples the responses, and find extrema only at the subsampled pixels. At each successive octave, the subsampling is increased, so that almost all computation is spent on the first octave. Consequently, the accuracy of features at larger scales is sacrificed, in the same way that it is for pyramid systems. While it would be possible for SIFT and SURF to forego subsampling, it would then be inefficient, with compute times growing much larger.

CenSurE also benefits from using an approximation to the Laplacian, which has been shown to be better for scale selection [12]. The center-surround approximation is fast to compute, while being insensitive to rotation (unlike the DOB Hessian approximation). Also, CenSurE uses a Harris edge filter, which gives better edge rejection than the Hessian.

Several simple center-surround filters exist in the literature. The bi-level Laplacian of Gaussian (BLoG) approximates the LoG filter using two levels. [14] describes circular BLoG filters and optimizes for the inner and outer radius to best approximate the LoG filter. The drawback is that the cost of BLoG depends on the size of the filter. Closer to our approach is that of Grabner et al. [15], who describe a difference-of-boxes (DOB) filter that approximates the SIFT detector, and is readily computed at all scales with integral images [16,17]. Contrary to the results presented in [15], we demonstrate that our DOB filters outperform SIFT in repeatability. This can be attributed to careful selection of filter sizes and using the second moment matrix instead of the Hessian to filter out responses along a line. In addition, the DOB filter is not invariant to rotation, and in this paper we propose filters that have better properties.

The rest of the paper is organized as follows. We describe our CenSurE features in detail in Section 2. We then discuss our modified upright SURF (MU-SURF) in Section 3. We compare the performance of CenSurE against several other feature detectors. Results of this comparison for image matching are presented in Section 4.1 followed by results for visual odometry in Section 4.2. Finally, Section 5 concludes this paper.

2 Center Surround Extrema (CenSurE) Features

Our approach to determining accurate large-scale features demands that we compute all features at all scales, and select the extrema across scale and location. Obviously, this strategy demands very fast computation, and we use simplified bi-level kernels as center-surround filters. The main concern is finding kernels that are rotationally invariant, yet easy to compute.

2.1 Finding Extrema

In developing affine-invariant features, Mikolajczyk and Schmid [18] report on two detectors that seem better than others in repeatability – the Harris-Laplace and Hessian-Laplace. Mikoljczyk and Schmid note that the Harris and Hessian detectors (essentially corner detectors) are good at selecting a location within a scale, but are not robust across scale. Instead, they show that the maximum of Laplacian operator across scales gives a robust characteristic scale - hence the hybrid operator, which they define as follows: first a peak in the Harris or Hessian operator is used to select a location, and then the Laplacian selects the scale at that location.

This strategy requires computing the Hessian/Harris measure at all locations and all scales, and additionally calculating the Laplacian at all scales where there are peaks in the corner detector. In our view, the Laplacian is easier to compute and to approximate than the Hessian, as was discovered by Lowe for SIFT features. So in our approach, we compute a simplified center-surround filter at all locations and all scales, and find the extrema in a local neighborhood. In a final step, these extrema are filtered by computing the Harris measure and eliminating those with a weak corner response.

2.2 Bi-level Filters

While Lowe approximated the Laplacian with the difference of Gaussians, we seek even simpler approximations, using center-surround filters that are bi-level, that is, they multiply the image value by either 1 or -1. Figure 1 shows a progression of bi-level filters with varying degrees of symmetry. The circular filter is the most faithful to the Laplacian, but hardest to compute. The other filters can be computed rapidly with integral images (Section 2.7), with decreasing cost from octagon to hexagon to box filter. We investigate the two endpoints: octagons for good performance, and boxes for good computation.

2.3 CenSurE Using Difference of Boxes

We replace the two circles in the circular BLoG with squares to form our CenSurE-DOB. This results in a basic center-surround Haar wavelet. Figure 1(d) shows our generic center-surround wavelet of block size n. The inner box is of size $(2n + 1) \times (2n + 1)$ and the outer box is of size $(4n + 1) \times (4n + 1)$. Convolution is done by multiplication and summing. If I_n is the inner weight

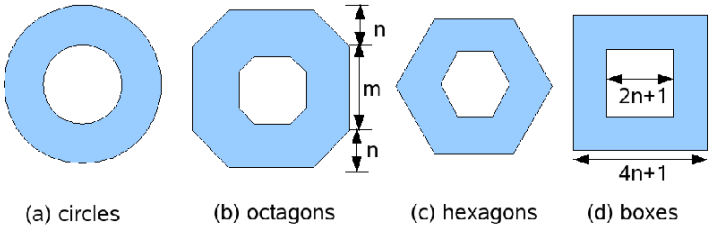

Fig. 1. Progression of Center-Surround bi-level filters. (a) circular symmetric BLoG (Bilevel LoG) filter. Successive filters (octagon, hexagon, box) have less symmetry.

and O_n is the weight in the outer box, then in order for the DC response of this filter to be zero, we must have

$$O_n(4n+1)^2 = I_n(2n+1)^2 \tag{1}$$

We must also normalize for the difference in area of each wavelet across scale.

$$I_n\,(2n+1)^2 = I_{n+1}\,(2\,(n+1)+1)^2 \tag{2}$$

We use a set of seven scales for the center-surround Haar wavelet, with block size $n = [1, 2, 3, 4, 5, 6, 7]$. Since the block sizes 1 and 7 are the boundary, the lowest scale at which a feature is detected corresponds to a block size of 2. This roughly corresponds to a LoG with a sigma of 1.885. These five scales cover $2\frac{1}{2}$ octaves, although the scales are linear. It is easy to add more filters with block sizes 8,9, and so on.

2.4 CenSurE Using Octagons

Difference of Boxes are obviously not rotationally invariant kernels. In particular, DOBs will perform poorly for 45 degrees in-plane rotation. Octagons, on the other hand are closer to circles and approximate LoG better than DOB.

In using octagons, the basic ideas of performing convolutions by inner and outer weighted additions remain the same. As in DOB, one has to find weights I_n and O_n such that the DC response is zero and all filters are normalized according to the area of the octagons.

An octagon can be represented by the height of the vertical side (m) and height of the slanted side (n) (Figure 1(b)). Table 1 shows the different octagon sizes corresponding to the seven scales. These octagons scale linearly and were experimentally chosen to correspond to the seven DOBs described in the previous section.

2.5 Non-maximal Suppression

We compute the seven filter responses at each pixel in the image. We then perform a non-maximal suppression over the scale space. Briefly, a response is

Table 1. CenSurE-OCT: inner and outer octagon sizes for various scales

scale	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$
inner (m,n)	$(3,0)$	$(3,1)$	$(3,2)$	$(5,2)$	$(5,3)$	$(5,4)$	$(5,5)$
outer (m,n)	$(5,2)$	$(5,3)$	$(7,3)$	$(9,4)$	$(9,7)$	$(13,7)$	$(15,10)$

suppressed if there is a response greater (maxima case) or a response less than (minima case) its neighbors in a local neighborhood over the location and scales. Pixels that are either maxima or minima in this neighborhood are the feature point locations. We use a 3x3x3 neighborhood for our non-maximal suppression.

The magnitude of the filter response gives an indication of the strength of the feature. The greater the strength, the more likely it is to be repeatable. Weak responses are likely to be unstable. Therefore, we can apply a threshold to filter out the weak responses.

Since all our responses are computed on the original image without subsampling, all our feature locations are localized well and we do not need to perform subpixel interpolation.

2.6 Line Suppression

Features that lie along an edge or line are poorly localized along it and therefore are not very stable. Such poorly defined peaks will have large principal curvatures along the line but a small one in the perpendicular direction and therefore can be filtered out using the ratio of principal curvatures. We use the second moment matrix of the response function at the particular scale to filter out these responses.

$$H = \begin{bmatrix} \sum L_x^2 & \sum L_x L_y \\ \sum L_x L_y & \sum L_y^2 \end{bmatrix} \tag{3}$$

L_x and L_y are the derivatives of the response function L along x and y. The summation is over a window that is linearly dependent on the scale of the particular feature point: the higher the scale, the larger the window size. Note that this is the scale-adapted Harris measure [18,10] and is different from the Hessian matrix used by SIFT [11,15] to filter out line responses. Once the Harris measure is computed, its trace and determinant can be used to compute the ratio of principal curvatures. We use a threshold of 10 for this ratio and a 9×9 window at the smallest scale of block size 2.

The Harris measure is more expensive to compute than the Hessian matrix used by SIFT. However, this measure needs to be computed for only a small number of feature points that are scale-space maxima and whose response is above a threshold and hence does not present a computational bottleneck. In our experience it does a better job than Hessian at suppressing line responses.

2.7 Filter Computation

The key to CenSurE is to be able to compute the bi-level filters efficiently at all sizes. The box filter can be done using integral images [16,17]. An integral image

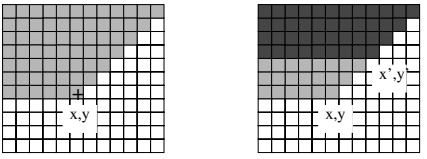

Fig. 2. Using slanted integral images to construct trapezoidal areas. Left is a slanted integral image, where the pixel x, y is the sum of the shaded areas; α is 1. Right is a half-trapezoid, from subtracting two slanted integral image pixels.

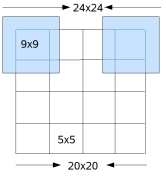

Fig. 3. Regions and subregions for MU-SURF descriptor. Each subregion (in blue) is 9x9 with an overlap of 2 pixels at each boundary. All sizes are relative to the scale of the feature s.

I is an intermediate representation for the image and contains the sum of gray scale pixel values of image N with height y and width x, i.e.,

$$I(x, y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} N(x', y') \tag{4}$$

The integral image is computed recursively, requiring only one scan over the image. Once the integral image is computed, it it takes only four additions to calculate the sum of the intensities over any upright, rectangular area, independent of its size.

Modified versions of integral images can be exploited to compute the other polygonal filters. The idea here is that any trapezoidal area can be computed in constant time using a combination of two different *slanted* integral images, where the sum at a pixel represents an angled area sum. The degree of slant is controlled by a parameter α:

$$I_\alpha(x, y) = \sum_{y'=0}^{y} \sum_{x'=0}^{x+\alpha(y-y')} N(x', y'). \tag{5}$$

When $\alpha = 0$, this is just the standard rectangular integral image. For $\alpha < 0$, the summed area slants to the left; for $\alpha > 0$, it slants to the right (Figure 2, left). Slanted integral images can be computed in the same time as rectangular ones, using incremental techniques.

Adding two areas together with the same slant determines one end of a trapezoid with parallel horizontal sides (Figure 2, right); the other end is done similarly, using a different slant. Each trapezoid requires three additions, just as in the rectangular case. Finally, the polygonal filters can be decomposed into 1 (box), 2 (hexagon), and 3 (octagon) trapezoids, which is the relative cost of computing these filters.

3 Modified Upright SURF (MU-SURF) Descriptor

Previously, we have demonstrated accurate visual odometry using ZNCC for feature matching [1] (using a 11×11 region). However, it is relatively sensitive to in-plane rotations (roll), larger changes in perspective, and inaccuracies in keypoint localization. The problems related to rolls and perspective changes become more significant as the region size increases. We have therefore decided to switch to an upright SURF type descriptor [13].

The SURF descriptor builds on from the SIFT descriptor by encoding local gradient information. It uses integral images to compute Haar wavelet responses, which are then summed in different ways in 4×4 subregions of the region to create a descriptor vector of length 64.

As pointed out by David Lowe [11], "it is important to avoid all boundary effects in which the descriptor abruptly changes as a sample shifts smoothly from being within one histogram to another or from one orientation to another." The SURF descriptor [13] weighs the Haar wavelet responses using a Gaussian centered at the interest point. This single weighting scheme gave poor results and we were unable to recreate the SURF descriptor results without accounting for these boundary effects.

To account for these boundary conditions, each boundary in our descriptor has a padding of $2s$, thereby increasing our region size from $20s$ to $24s$, s being the scale of the feature. The Haar wavelet responses in the horizontal (d_x) and vertical (d_y) directions are computed for each 24×24 point in the region with filter size $2s$ by first creating a summed image, where each pixel is the sum of a region of size s. The Haar wavelet output results in four fixed-size d_x,d_y, $|d_x|$,$|d_y|$ images that have the dimensions 24×24 pixels irrespective of the scale.

Each d_x,d_y, $|d_x|$,$|d_y|$ image is then split into 4×4 square overlapping subregions of size 9×9 pixels with an overlap of 2 pixels with each of the neighbors. Figure fig:descriptor shows these regions and subregions. For each subregion the values are then weighted with a precomputed Gaussian ($\sigma_1 = 2.5$) centered on the subregion center and summed into the usual SURF descriptor vector for each subregion: $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Each subregion vector is then weighted using another Gaussian ($\sigma_2 = 1.5$) defined on a mask of size 4×4 and centered on the feature point. Like the original SURF descriptor, this vector is then normalized.

The overlap allows each subregion to work on a larger area so samples that get shifted around are more likely to still leave a signature in the correct subregion vectors. Likewise, the subregion Gaussian weighting means that samples near borders that get shifted out of a subregion have less impact on the subregion descriptor vector.

From an implementation point of view the dynamic range of the vector was small enough that the end results could be scaled into C++ shorts. This allows for very fast matching using compiler vectorization.

CenSurE features themselves are signed based on their being dark or bright blobs. This is similar to SURF and can also be used to speed up the matching by only matching bright features to bright features and so forth.

We have compared the performance of MU-SURF with U-SURF for matching and found them to be similar. As will be pointed out in Section 4.3, our implementation of MU-SURF is significantly faster than U-SURF. It is unclear to us as to why MU-SURF is so much faster. We are currently looking into this.

4 Experimental Results

We compare CenSurE-DOB and CenSurE-OCT to Harris, FAST, SIFT, and SURF feature detectors for both image matching and visual odometry. Results for image matching are presented in Section 4.1 and VO in Section 4.2.

4.1 Image Matching

For image matching, we have used the framework of [12] to evaluate repeatability scores for each detector on the graffiti and boat sequences[1]. We have used the default parameters for each of these detectors. In addition, since each of these detectors has a single value that represents the strength of the feature, we have chosen a strength threshold such that each of these detectors results in the same number of features in the common overlapping regions. Figure 4 (a) & (b) shows a plot of the detector repeatability and number of correspondences for each detector using 800 features and an overlap threshold of 40% for the graffiti sequence. For Harris and FAST, the scale of all detected points was assumed to be the same and set at 2.0.

Both versions of CenSurE are better than SIFT or SURF, although for large viewpoint changes, the differences become only marginal. As can be expected, CenSurE-OCT does better than CenSurE-DOB.

The *boat* sequence is more challenging because of large changes in rotation and zoom. Figure 4 (c) & (d) shows the detector performance for this sequence for 800 features. On this challenging sequence, CenSurE performs slightly worse than either SIFT or SURF, especially for the larger zooms. This can be attributed to CenSurE's non-logarithmic scale sampling. Furthermore, CenSurE filters cover only $2\frac{1}{2}$ octaves and therefore has less degree of scale-invariance for large scale changes.

To evaluate the matching performance, we used our MU-SURF descriptor for each of those detectors and matched each detected point in one image to the one with the lowest error using Euclidean distance. A correspondence was deemed as matched if the true match was within a search radius r of its estimated correspondence. Note that this is a different criterion than considering overlap error and we have chosen this because this same criterion is used in visual odometry to perform image registration. Figure 5 shows the percentage of correct matches as a function of search radius when the number of features is fixed to 800.

[1] Available from `http://www.robots.ox.ac.uk/~vgg/research/affine/`

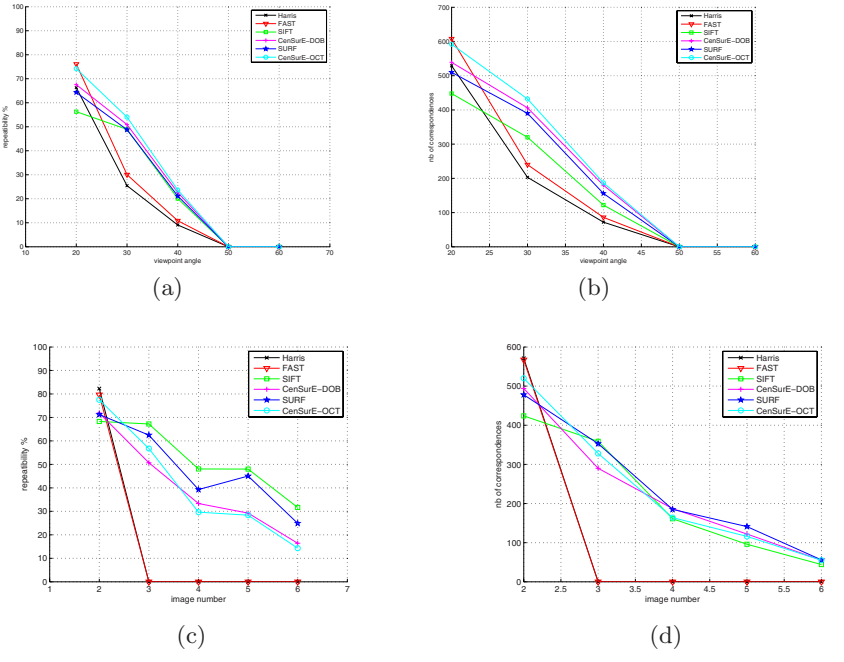

Fig. 4. Repeatability and number of correspondences for different detectors for the graffiti and boat sequences. The number of features is the same for each detector. (a) & (b) graffiti sequence. (c) & (d) boat sequence.

4.2 Visual Odometry

We evaluate the performance of CenSurE for performing visual odometry in challenging off-road environments. Because there can be large image motion between frames, including in-plane rotations, the tracking task is difficult: essentially, features must be re-detected at each frame. As usual, we compare our method against Harris, FAST, SIFT, and SURF features. Note that this is a test of the *detectors*; the same MU-SURF descriptor was used for each feature.

The Visual Odometry (VO) system derives from recent research by the authors and others on high-precision VO [1,2] using a pair of stereo cameras. For each new frame, we perform the following process.

1. Distinctive features are extracted from each new frame in the left image. Standard stereo methods are used to find the corresponding point in the right image.
2. Left-image features are matched to the features extracted in the previous frame using our descriptor. We use a large area, usually around 1/5 of the image, to search for matching features.
3. From these uncertain matches, we recover a consensus pose estimate using a RANSAC method [19]. Several thousand relative pose hypotheses are generated by randomly selecting three matched non-collinear features, and then scored using pixel reprojection errors.

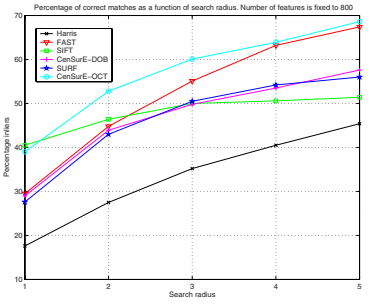

Fig. 5. Percentage of correct matches as a function of search radius

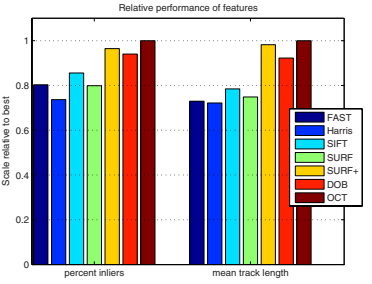

Fig. 6. Basic performance of operators in the VO dataset

4. If the motion estimate is small and the percentage of inliers is large enough, we discard the frame, since composing such small motions increases error. A kept frame is called a *key frame*. The larger the distance between key frames, the better the estimate will be.
5. The pose estimate is refined further in a sparse bundle adjustment (SBA) framework [20,21].

The dataset for this experiment consists of 19K frames taken over the course of a 3 km autonomous, rough-terrain run. The images have resolution 512x384, and were taken at a 10 Hz rate; the mean motion between frames was about 0.1m. The dataset also contains RTK GPS readings synchronized with the frames, so ground truth to within about 10 cm is available for gauging accuracy.

We ran each of the operators under the same conditions and parameters for visual odometry, and compared the results. Since the performance of an operator is strongly dependent on the number of features found, we set a threshold of 400 features per image, and considered the highest-ranking 400 features for each operator. We also tried hard to choose the best parameters for each operator. For example, for SURF we used doubled images and a subsampling factor of 1, since this gave the best performance (labeled "SURF+" in the figures).

The first set of statistics shows the raw performance of the detector on two of the most important performance measures for VO: the average percentage of inliers to the motion estimate, and the mean track length for a feature (Figure 6). In general, the scale-space operators performed much better than the simple corner detectors. CenSurE-OCT did the best, beating out SURF by a small margin. CenSurE-DOB is also a good performer, but suffers from lack of radial symmetry. Surprisingly, SIFT did not do very well, barely beating Harris corners.

Note that the performance of the scale-space operators is sensitive to the sampling density. For standard SURF settings (no doubled image, subsampling of 2) the performance is worse than the corner operators. Only when sampling densely for 2 octaves, by using doubled images and setting subsampling to 1, does performance approach that of CenSurE-OCT. Of course, this mode is much more expensive to compute for SURF (see Section 4.3).

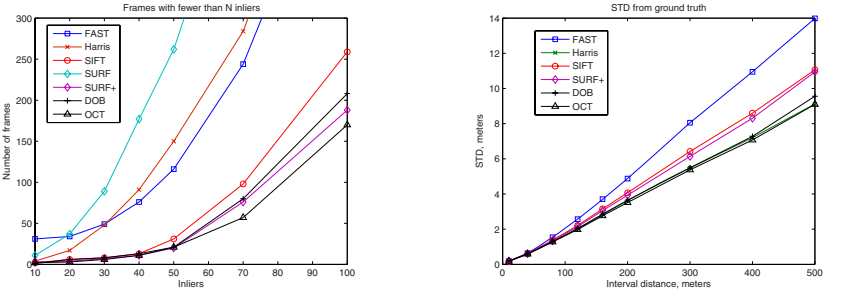

Fig. 7. Accuracy statistics. Left: number of frames with inliers less than a certain amount, out of 19K frames. For example, FAST and Harris both have around 50 frames with fewer than 30 inliers. Right: standard deviation from ground truth, over trajectories of varying length.

The question to ask is: do these performance results translate into actual gains in accuracy of the VO trajectory? We look at two measures of accuracy, the number of frames with low inlier counts, and the deviation of the VO trajectory from ground truth (Figure 7). The graph at the left of the figure can be used to show how many frames are not matched, given a threshold for inliers. For example, we typically use 30 inliers as a cutoff: any frames with fewer matches are considered to have bad motion estimates. With this cutoff, SIFT, SURF+, OCT, and DOB all have less than 10 missed frames, while Harris and FAST have around 50. To show the influence of low-resolution localization, standard SURF does very poorly here, as we expect from the previous performance graph.

Finally, we looked at the deviation of the VO estimates from ground truth, for different trajectory lengths. At every 10 key frames along the VO trajectory, we compared a trajectory of length N against the corresponding ground truth, to give a dense sampling (about 1000 for each trajectory length). The standard deviation is a measure of the goodness of the VO trajectory. Here, OCT, DOB and Harris were all about equivalent, and gave the best estimates. Although Harris does not do well in getting large numbers of inliers for difficult motions, it is very well localized, and so gives good motion estimates. SIFT and SURF+ give equivalent results, and are penalized by their localization error.

Overall, CenSurE-OCT gives the best results in terms of accurate motion estimates, and misses very few frames. Harris does very well in accuracy of motion, but misses a large number of frames. SURF+ is a reasonable performer in terms of missed frames, but is not as accurate as the CenSurE or Harris features.

4.3 Timing Results

Timing results for our CenSurE and MU-SURF implementations on an Intel Pentium-M 2 GHz machine for a 512×384 image are presented in Table 2. For comparison, SURF timings based on the original author's implementations[2] (on the same computational platform and on the same images) are also included.

[2] Available from `http://www.vision.ee.ethz.ch/~surf/download.html`

Table 2. Time in milliseconds for different feature detectors and descriptors

detector							descriptor	
SURF+	SURF-1	SIFT	SURF	OCT	DOB	Harris	U-SURF	MU-SURF
3408	292	304	75	23	17	10	308	16

SURF has default parameters (no doubled image, subsampling of 2), whereas SURF-1 has subsampling set to 1, and SURF+ is SURF-1 with a doubled image. For the descriptor, both U-SURF and MU-SURF are given the same features (about 1000 in number).

For VO the best performance is with SURF+. In this case, CenSurE-OCT yields more than a hundred-fold improvement in timing. Our MU-SURF is also more than twenty times faster than U-SURF. It is clear that feature detection using CenSurE features and matching using MU-SURF descriptors can be easily accomplished in real time.

5 Conclusion

We have presented two variants of center-surround feature detectors (CenSurE) that outperform other state-of-the-art feature detectors for image registration in general and visual odometry in particular. CenSurE features are computed at the extrema of the center-surround filters over multiple scales, using the original image resolution for each scale. They are an approximation to the scale-space Laplacian of Gaussian and can be computed in real time using integral images. Not only are CenSurE features efficient, but they are distinctive, stable and repeatable in changes of viewpoint. For visual odometry, CenSurE features result in longer track lengths, fewer frames where images fail to match, and better motion estimates.

We have also presented a modified version of the upright SURF descriptor (MU-SURF). Although the basic idea is same as the original SURF descriptor, we have modified it so as to handle the boundaries better, and it is also faster. It has been our experience that MU-SURF is well suited for visual odometry and performs much better than normalized cross-correlation without much computational overhead.

CenSurE is in constant use on our outdoor robots for localization; our goal is to ultimately be able to do visual SLAM in real time. Toward this end, we are exploiting CenSurE features to recognize landmarks and previously visited places in order to perform loop closure.

References

1. Konolige, K., Agrawal, M., Solà, J.: Large scale visual odometry for rough terrain. In: Proc. International Symposium on Robotics Research (November 2007)
2. Agrawal, M., Konolige, K.: Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In: ICPR (August 2006)

3. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–151 (1988)
4. Shi, J., Tomasi, C.: Good features to track. In: Proc. Computer Vision and Pattern Recognition (CVPR) (1994)
5. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision, vol. 1 (2006)
6. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking 2, 1508–1515 (2005)
7. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Real time localization and 3rd reconstruction. In: CVPR, vol. 1, pp. 363–370 (June 2006)
8. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (June 2004)
9. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350. Springer, Heidelberg (2002)
10. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30(2) (1998)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
12. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV, 43–72 (2005)
13. Herbert Bay, T.T., Gool, L.V.: Surf: Speeded up robust features. In: European Conference on Computer Vision (May 2006)
14. Pei, S.C., Horng, J.H.: Design of FIR bilevel Laplacian-of-Gaussian filter. Signal Processing 82, 677–691 (2002)
15. Grabner, M., Grabner, H., Bischof, H.: Fast approximated SIFT. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 918–927. Springer, Heidelberg (2006)
16. Viola, P., Jones, M.: Robust real-time face detection. In: ICCV 2001 (2001)
17. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: IEEE Conference on Image Processing (ICIP) (2002)
18. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: International Conference on Computer Vision (ICCV) (2001)
19. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun. ACM 24, 381–395 (1981)
20. Engels, C., Stewénius, H., Nister, D.: Bundle adjustment rules. Photogrammetric Computer Vision (September 2006)
21. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Vision Algorithms: Theory and Practice. LNCS, pp. 298–375. Springer, Heidelberg (2000)