

Robust Features Matching Using Scale-invariant Center Surround Filter

Zeng-You Sun, Yu-Shuai Duan, Dong-Na Yang

School of Information Engineering
Northeast Electric Power University
Jilin 132012, China

Xiang Fang*

College of Science
Northeast Electric Power University
Jilin 132012, China

*Corresponding author: fangxiang_cs@163.com

Received October, 2017; revised March, 2018

ABSTRACT. *Image matching is the base of many computer vision problems, such as object recognition, image or motion tracking. Most of current methods rely on costly descriptors for detection and matching. This paper presents a method for extracting distinctive invariant features from images, coined SCFD (Scale-invariant Center surround Filter Detection). We demonstrate through experiments how SCFD can be used to perform reliable and high-precision matching between different views of an object or scene, yet can be computed much faster.*

Keywords: Feature point matching, Scale-invariance, Feature point detection, Symmetric matching, ORB.

1. **Introduction.** Image matching is the task of establishing correspondence between images of the same scene. This is a fundamental aspect of many problems in computer vision applications, such as target detection [1], image index [2], visual positioning [3], and visual navigation [4], etc. However, most of applications are constrained by real-time and stability (persistence across viewpoint change). Particularly, in Vehicle dynamics and outdoor scenery can make the problem of matching images very challenging. Such as visual odometry system, images are collected from different time, thus perspectives of images are also different. At the same time, affected by illumination and noise in environment, the edge profile of images will have a large difference, even if images are extremely vague and noise disturbance is large. As a result, it has the important significance to design a feature point with rapid and stable extraction, enhance image matching accuracy and anti-disturbance ability to capacity.

Broadly speaking, we can divide image matching classes into two types. The matching algorithm based on grey information conducts matching through two-dimensional space sliding form. The operation process is simple and matching has the high precision. However, the algorithm has the large operand and it is relatively sensitive to noise. The matching algorithm based on features which include corner features, line features and edge features. The extraction process of image feature points is less affected by noise. At the

same time, it has the strong anti-disturbance ability to grey changes, image deformation and shield.

A wide variety of feature detection methods have been proposed, such as Moravec [5] and Harris [6], their feature point detection process is only conducted on the single scale. And they are easy to be affected by noise. Lowe [7] proposed SIFT (Scale-Invariant Feature Transform) algorithm and used Gaussian function to construct scale space to maintain invariance on image scaling, rotation and affine transformation. However, due to the application of 128-dimension operation operator and large calculated amount, it is not suitable to be applied in image matching with the real-time requirements. Sukthankar [8] used the Principal Component Analysis to replace the histogram in SIFT algorithm, so as to achieve the goal of reduce dimensions for SIFT descriptor, however, it affects distinctive of features while increases the formative time of descriptor.

We strike a balance between the real-time and accuracy by taking a simple center-surround filters approach. Following are the major stages of computation used to generate the set of image features:

1. Scale-space point detection: The first stage is compute all features at all scales, and select the extrema across scale and location.
2. Subpixel interpolation: The detector is used to obtain higher precision feature point positioning by subpixel interpolation.
3. Keypoint localization: At each candidate location, the keypoints are selected according to their stability measurements.
4. Keypoint descriptor: A simple and efficient descriptor base on ORB is proposed.

To validate SCFD, we compare the performance of SCFD against several other feature detectors.

2. Related Work. Harris corners was introduced by Harris and Stephen [6], it is based on eigenvalues of second moment matrix. FAST is probably the most widely used detector [9, 10], proposed back in 2006. It is based on the gray difference between the pixel and its neighborhood. While these feature detectors are usually called corner detector, they are not scale-invariant, so it does not provide a good basis for matching images of different sizes. The concept of automatic scale selection was proposed by Lindeberg [11], who studied the problem of identifying an appropriate and consistent scale for feature detection. Rattarangsi proposed the multi-scale method based on Gaussian scale space [12]. However due to large number of scales used in feature detection process, resulted in a very large amount of computation. Low proposed a Difference of Guassians (DOG) filter to approximate the Laplacian of Guassians (LOG) [7]. Similarly, Bay and Ess used Hessian approximation, which drastically reduce the number of operations for simple box convolution. Neither of them computes responses at all pixels for larger scales, and consequently do not detect extrema across all scales. Instead, they subsampled the responses at each scale octave independently, and find extrema only at the subsampled pixels, yield poor accuracy.

We also benefits from using an approximation to the Laplacian. In this paper, we seek even simpler approximations, using bi-level center-surround filters which is fast to compute and insensitive to rotation. Various center-surround filters have been proposed. Pei and Horng used the bi-level Laplacian-of-Guassian (BLoG) filter to approximate the LoG filter [13]. Describes circular BLoG fileter and optimizes for the inner and outer radius to best approximate the LoG filter, while the cost of BLoG depends on the size of the filter. Closer to our approach is that of Crabner [14], who proposed a difference-of-boxes (DOB) filter that approximates the SIFT descriptor, which is compute at all scales

with integral images [15, 16]. We demonstrate that our DOB filters out perform SITF in repeatability. This can attributed to selection and subpixel interpolation.

The rest of the paper is organized as follows. We describe the strategy applied for fast and robust feature point detection in section 3. We then discuss our modified upright ORB in section 4. To validate SCFD, we perform experiments that test the properties of SCFD and several other feature detectors, and performance in image-matching applications in section 5. The article is concluded in section 6.

3. Center Surround Filter Detection (SCFD). Our approach for feature point detection uses simplified bi-level kernels as center-surround filters which is easy to compute. The first stage of feature point detection is construct the scale space and calculate center surround Haar wavelet response value of each pixel point in original images [17, 18]. In addition, integral image speed-up operation process is applied [12]. Then, non-maximum suppression method is applied to detect extreme values. At last, Harris and sub-pixel interpolation are applied to acquire the more stable feature points.

3.1. Bi-level filters. Low approximated the Laplacian with the DoG, in this paper we uses the simpler bi-level filter of center surround approximate Gauss-Laplace operator [12], so as to achieve the goal of simplifying calculation. The general center surround wavelet of block size n is shown in Figure 1. The inner box is $(2n + 1) \times (2n + 1)$ and the outer box is of size $(4n + 1) \times (4n + 1)$. If I_n is the inner weight and O_n is the weight of the outer box, then in order for the DC response of this filter to be zero, we must have:

$$O_n (4n + 1)^2 = I_n (2n + 1)^2 \quad (1)$$

The scale space is conducted the normalization processing:

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

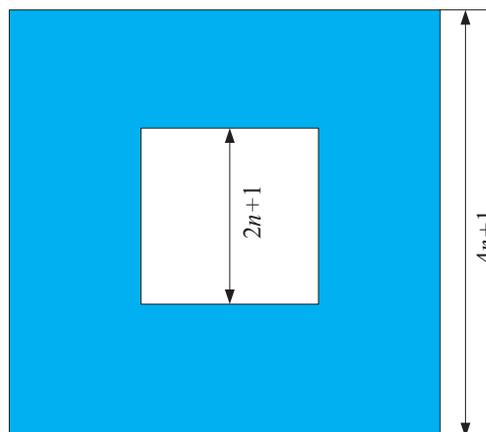


FIGURE 1. Center-Surround bi-level filters

We used a set of five scales for the center-surround Harr wavelet, with block size $n = [1, 2, 3, 4, 5]$. While the block size 1 and 5 are the boundary, the lowest scale at which a feature is detected corresponds to a block size of 2.

3.2. Construction of Scale Space. We use of center-surround filters and integral images, therefore, we do not have to iteratively apply the same on 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.

The scale space is divided into octaves. Each octave is composed of five layers. In order to improve extraction precision of feature points, the incremental center surround filter and original images in each group are used as convolution to get a series of response figures, differing from SIFT of which the next group conducts down-sampling on the previous group. We use subpixel interpolation to locate the feature points accurately, and further enhance the stability and matching accuracy of feature points. We compute the five filter response at each pixel in the image. The filter with the core size of 3×3 is considered as the initial layer of the scale space. For two successive levels, we must increase this size by a minimum of 2 pixels in order to keep the size uneven and thus ensure the presence of the central pixel. This results in a total increase of the filter size by 2 pixels (see figure 2).

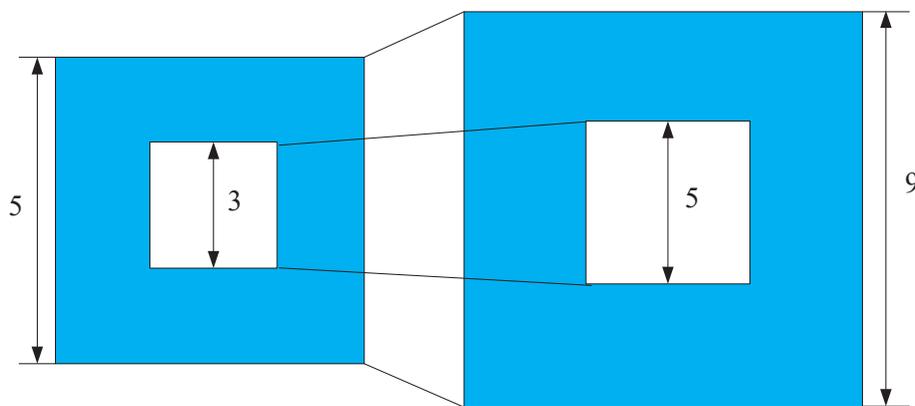


FIGURE 2. Filter inner and outer boxes sizes for two successive scale levels (3×3 and 5×5). The length of the block size can only be increased by an even number of pixels in order to guarantee the presence of a central pixel.

The construction of the scale space starts with the 3×3 , which calculates the filter response of the image for the smallest scale. Then filter size and step are gradually increased. For example, the filter size in the first group is 3, 5, 7 and 9, the core of the filter in each layer of the second group is successively increased by 4, then the core size is 5×5 , 9×9 , 13×13 , and 17×17 . In order to confirm the extreme point in 3D neighborhood, more two layers are required. In other words, the initial layer and top layer only can be used for comparison, while can't contain extreme point. Similar considerations hold for the other octaves. For each new octave, the filter size increase in double (going from 2 to 4 to 8 to 16). The filter sizes for the second octave are 5×5 , 9×9 , 13×13 , and 17×17 . A third octave is computed with the filter sizes 9×9 , 17×17 , 25×25 , 33×33 . And the fourth octave using filter sizes 17×17 , 33×33 , 49×49 , 55×55 . Figure 2 gives an overview of the filter size of the filter for the first three octaves.

3.3. Feature point detection. We compute the five filter response at each pixel in the image. Then a non-maximum suppression in a $3 \times 3 \times 3$ neighborhood is applied. The filter's response amplitude gives the indication of feature strength. The stronger response is, the better repetition of feature points will be. Feature points that lie along an edge or line are poorly localized along it and therefore are not very stable. Therefore, we can apply Hessian matrix used by SIFT to filter out line responses [11, 15].

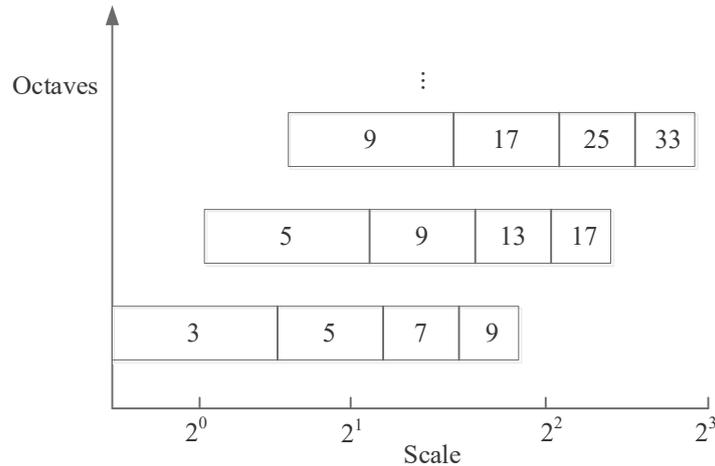


FIGURE 3. The filter sizes for three different octaves. The logarithmic horizontal axis represents the scales.

3.4. Accurate Feature point localization. We found an feature point candidate by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location and scale which allows points to be rejected that are sensitive to noise or are poorly localized along edge. We use the same method as Brown to fit the 3D quadratic function to a local sampling point to determine the maximum interpolated position, and the experiments show that this provides a substantial improvement in matching and stability. We used the Taylor expansion of the scale-space function, $D(x, y, \delta)$, shifted so that the origin is at the sample point:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (3)$$

Where D and its derivatives are evaluated at the sample point and $X = (x, y, \delta)^T$ is the offset from this point. The location of the extremum, X , is determined by taking the derivative of this function with respect to x and setting it to zero.

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X} \quad (4)$$

The function value at the extremum, $D(\hat{X})$, can be obtained by substituting equation (4) into (3), giving:

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X} \quad (5)$$

If the offset \hat{X} is larger than 0.5 in any dimension (x, y , or δ), then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset \hat{x} is added to the location of its sample point to get the interpolated estimate for the location of the extremum. In addition, $|D(X)|$ is too small to be susceptible to noise interference and becomes unstable, so it will be smaller than an empirical value (0.03, Used in the Lowe paper). At the same time, in this process to obtain the exact location of the feature points and scales.

4. Symmetric Matching rBRIEF (SM-rBRIEF) descriptor. In section 3, we have introduced a method for extracting invariant feature point from images that can be used to perform reliable matching between different views of an object or scene. In this section, we first introduce an ORB descriptor, Rublee has shown its performance and efficiency

relative to other popular features [19]. However, it is not addressed is scale invariance and relatively to rotations and inaccuracies in feature point matching. Due to the existence of error match, bilateral matching is used to improve the match accuracy by eliminating the false match.

$$\tau(p; x, y) := \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \quad (6)$$

Where, $p(x)$ is the intensity of P at point x . The feature is defined as a vector of n binary tests:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (7)$$

Where n can take 128, 256 and other values, different values will affect the speed, recognition rate. We also choose a vector length $n = 256$.

However, the feature points described by BRIEF do not have the rotation invariance, and the stability of the matching is improved by using Intensity Centroid method [18]. Define the $2 \times n$ matrix:

$$S = \begin{pmatrix} x_1 & x_2 & \dots & x_{2n} \\ y_1 & y_2 & \dots & y_{2n} \end{pmatrix} \quad (8)$$

Where (x_i, y_i) represents a test point pair. For each feature point, the corrected matrix $S_\theta = R_\theta S$ can be obtained by using the rotation matrix R_θ constructed by its main direction θ . This gives a description of the rotation invariance.

$$g_n(p, \theta) = f_n(p) | (x_i, y_i) \in s_\theta \quad (9)$$

Since the random point pairs that generate the Brief descriptor do not necessarily have low correlation and high variance properties, the variance and correlation over steered BRIEF may be decreased. To recover from the loss of variance in steered BRIEF, and to reduce correlation among the binary tests, we use a learning method for choosing a good subset of binary tests.

4.1. Symmetric matching. The generated descriptor is a binary code string, it is simple to use the Hamming distance to match the feature points, and the matching can be achieved by setting the threshold. However, is likely to cause matching errors and instability under the outdoor environment noise. Therefore, we proposed a symmetric matching method, which has the advantages of high matching precision, high reliability and strong robustness. Assuming that image I_1 and image I_2 , for each new image, we perform the following process.

1. Distinctive features are extracted from the left image I_1 and right image I_2 . European distance methods are used to find the corresponding point in the right image.
2. Take one of the point i in the feature point set in the image I_1 , and then search for the two points with the shortest European distance in the feature point set in the target image I_2 . By comparing the distance of the closest neighbor to that of the second-closest neighbor, if the ratio is less than a given threshold, it is shown that the feature point i and the feature point j are a pair of matching points.
3. Based on the above-mentioned first matching result, search the closest point of the image I_2 feature point set has been matched in the image I_1 feature point set.
4. If the point i in I_1 and point j in I_2 are a pair of matching points, and the point j in I_2 and point i in I_1 is also a pair of matching points, then the matching point is considered to be the correct match.

Figure 4 shows the value of this measure for real image data. The probabilities for correct and incorrect matches are shown in terms of the ratio of closest to second-closest neighbors of each interest point.

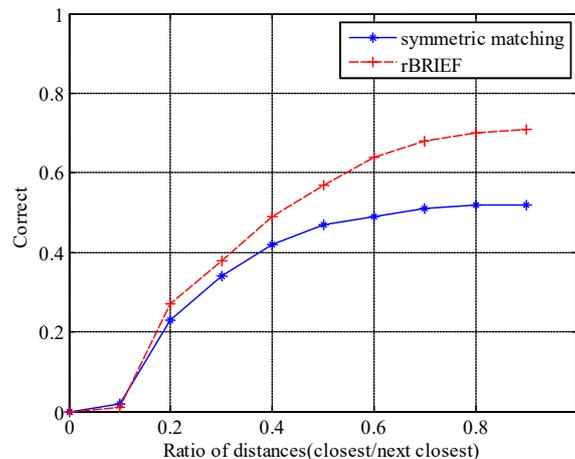


FIGURE 4. The percentage that a match is correct can be determined by taking the symmetric matching and Hamming distance. Using a database of 6425 feature points.

Figure 4 shows the percentage of correct for the two matching strategies. In each test image (synthetic rotation or real-world viewpoint change), the symmetric matching shows again a better performance than the Hamming distance. Symmetric matching selects only the best match below the threshold and rejects all others; therefore, there are less false matches and the precision is high. For our object recognition implementation, we reject all matches in which the distance ratio is greater than 0.8. This figure was generated by matching images following random scale and orientation change, a depth rotation of 15 degrees, and addition of 2% image noise, against a database of 6425 feature points.

5. Experimental Results. We evaluate our detector, which we call SCFD, using the image sequences and testing software provided by Mikolajczyk [20]. 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, lighting changes. In all experiments reported in this paper, the timings were measured on a standard PC Intel Core i5, running at 2.5 GHz.

5.1. SCFD Detector. We tested the SCFD to FAST, Harris, SIFT, and SURF feature detectors for image matching, using two datasets: images with synthetic in-plane rotation (boat sequence) and an out-door images captured from different viewpoints (wall sequence). We have used the default parameters for each of these detectors and chosen a strength threshold such that each of these detectors results in the same number of features in the common overlapping regions.

The repeatability scores for the walls sequence (figure 5(a)) are comparable for all detectors. The SCFD detector shows again a better performance than SIFT or SURF, although for large viewpoint changes, the differences become only minimal.

The boat sequence (figure 5(b)) is more challenging because of large changes in rotation and zoom. On this challenging sequence, the SCFD detector shows again a better performance than the SURF and FAST. While, it performs slightly worse than SIFT,

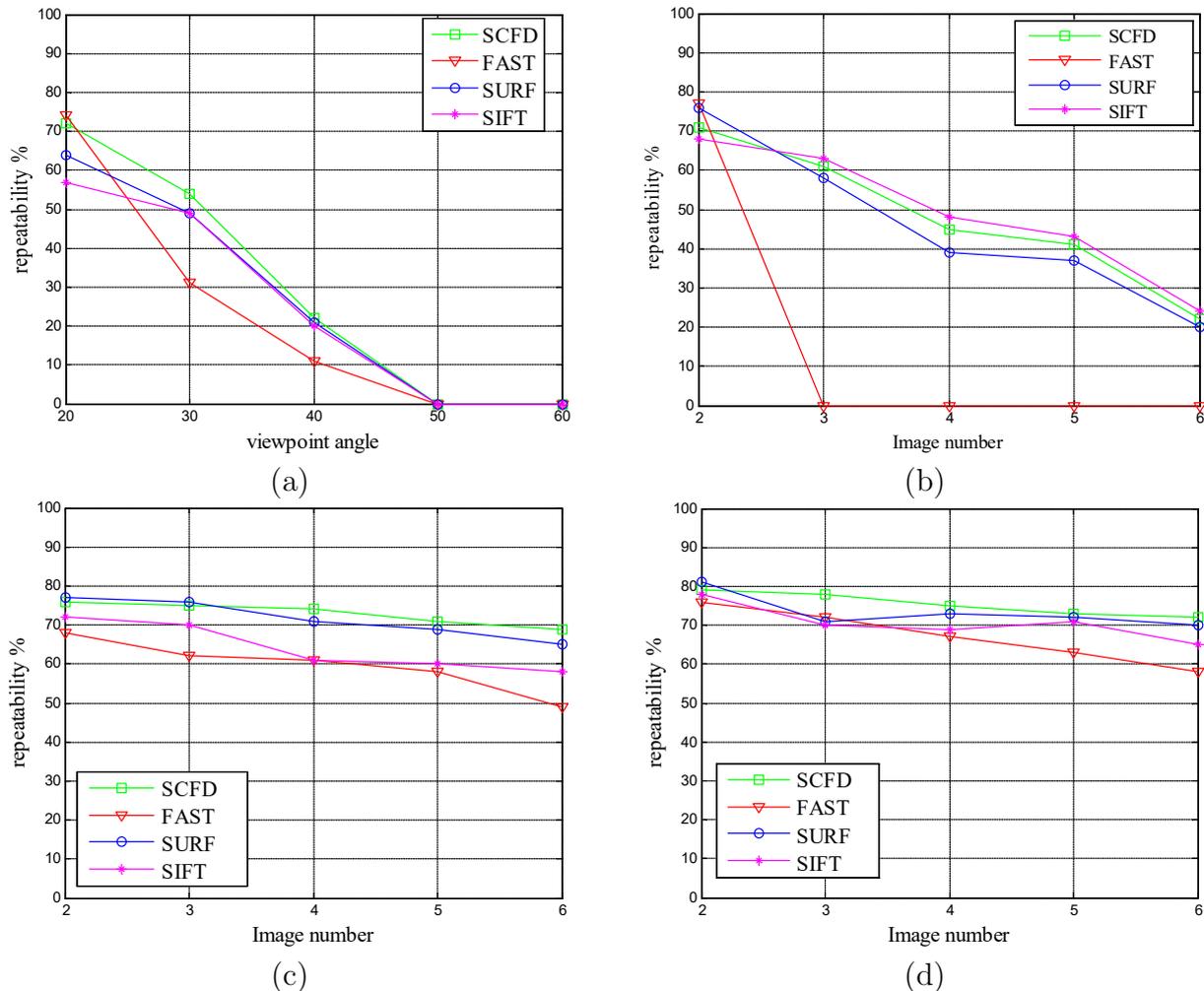


FIGURE 5. Repeatability score for different sequences for all detectors. (a) wall sequence (b) boat sequence. (c) bikes sequence (d) leuven sequence.

especially for the larger zooms. This can be attributed to SCFD scale sampling, SCFD's filters cover only $2\frac{1}{2}$ octaves and therefore has less degree of scale-invariance for large scale changes.

The repeatability score of the SCFD detector for the bikes and leuven sequence (figure 5(c) & (d)) outperforms the competitors. The experimental show results for image blur and illumination changes, SCFD detector has the favorable matching stability by comparing with other three algorithms, because the sub-pixel interpolation is used to confirm the accurate position of feature points, therefore, the feature points have a strong robustness.

5.2. Benchmarks. One emphasis for SCFD is the efficiency of detection and description on standard CPUs. Timing results for our SCFD detector and SM-rBRIEF descriptor implementations was executed in a single thread running on an Intel i5 2.5 GHz processor. We computed each detector and descriptor separately on five scales of the image, with a scaling factor of $\sqrt{2}$. The SCFD system breaks down into the following times per typical image of size 640×480 .

Comparing to SIFT, SURF and ORB on the same data, for averaged over 24 640×480 images from the Mikolajczyk dataset, we get the following times: Our SCFD is also more than Eighteen times faster than SURF. It is clear that feature detection using SCFD

TABLE 1. Time in milliseconds for SCFD feature detectors

	SCFD	SM-rBRIEF
Time(ms)	153	45

TABLE 2. Time in milliseconds for different feature detectors

Detector	SCFD	SIFT	SURF	ORB
Time(ms)	198	10544	3582	39

features and matching using SM-rBRIEF descriptors can be easily achieved in applications where the real-time requirements are relatively high.

6. Conclusions. In this paper, we have presented a scale and rotation invariant feature point detector and descriptor, then demonstrated its performance and efficiency relative to other popular features. The speed gain is due to the use of center-surround filters over multiple scales, which are an approximation to the scale-space Laplacian of Gaussian and can be computed in real time using integral images. The high repeatability and stable in changes of viewpoint and illumination is achieved by subpixel interpolation and construction of multi-scale space.

Our descriptor, based on the rBRIEF descriptor, we have modified it so as to handle the real-world image matching better, and it is also stable. Experiments for real image matching at different types of geometric and photometric transformations, like changing viewpoints, zoom and rotation, lighting changes, highlighted SCFD's potential in a wide range of computer vision applications.

REFERENCES

- [1] J. Gance, J. P. Malet, T. Dewez, et al, Target Detection and Tracking of Moving Objects for Characterizing Landslide Displacements from Time-lapse Terrestrial Optical Images. *Engineering Geology*, vol.172, no.5, pp.26–40, 2014.
- [2] X. Tan, B. Triggs, Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, vol.19, no.6, pp.1635, 2010.
- [3] D. Xu, L. Han, M. Tan, et al. Ceiling-Based Visual Positioning for an Indoor Mobile Robot With Monocular Vision. *IEEE Transactions on Industrial Electronics*, vol.56, no.5, pp.1617–1628, 2009.
- [4] L. Meier, P. Tanskanen, L. Heng, et al. PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight Using Onboard Computer Vision, *Autonomous Robots*, vol.33, no.1, pp.21–39, 2012.
- [5] H. P. Moravec, Obstacle Avoidance and Navigation in The Real World by A Seeing Robot Rover. Stanford University, 1980.
- [6] C. Harris A Combined Corner and Edge Detector. *Proc Alvey Vision Conf*, vol.1988, no.3, pp. 147–151, 1988.
- [7] D. G. Lowe, Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, vol.60, no.2, pp.91–110, 2004.
- [8] Y. Ke, R. Saththar, PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*. vol.2, no.2, pp.506–513, 2004.
- [9] Rosten E, Drummond T. Machine Learning for High-Speed Corner Detection. *European Conference on Computer Vision*, vol.2006, no.3951, pp.430–443, 2006.
- [10] E. Rosten, R. Porter, T. Drummond, Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.32, no.1, pp.105–119, 2009.
- [11] T. Lindeberg, Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, vol.30, no.2, pp.79–116, 1998.
- [12] A. Rattarangsi, R. T. Chin, Scale-Based Detection of Corners of Planar Curves. *Pattern Analysis & Machine Intelligence IEEE Transactions on*, vol.14, no.4, pp.430–449, 1992.

- [13] S. C. Pei, J. H. Horng, Design of FIR bilevel Laplacian-of-Gaussian filter. *Signal Processing*, vol.82, no.4, pp.677–691, 2002.
- [14] M. Grabner, H. Grabner, H. Bischof, Fast Approximated SIFT. *Lecture Notes in Computer Science*, vol.26, no.3851, pp.918–927, 2006.
- [15] A. P. Witkin, Scale-space Filtering. *Readings in Computer Vision*, vol.42, no.3, pp.329–332, 1987.
- [16] Witkin A. Scale-space filtering: A New Approach to Multi-scale Description. *IEEE International Conference on ICASSP*, vol.9, no.3, pp.150–153, 2003.
- [17] R. Lienhart, An Extended Set of Haar-like Feature for Rapid Object Detection. *Proc of Icip*, vol.1, no.1, pp.900–903, 2002.
- [18] M. Calonder, V. Lepetit, C. Strecha, et al. BRIEF: Binary Robust Independent Elementary Features. *Computer Vision-Eccv 2010*, vol.2010, no.6314, pp.778–792, 2010.
- [19] E. Rublee, V. Rabaud, K. Konolige, et al. ORB: An efficient alternative to SIFT or SURF. *IEEE International Conference on Computer Vision*, vol.58, no.11, pp.2564–2571, 2011.
- [20] K. Mikolajczyk, C. Schmid, An Affine Invariant Interest Point Detector. *International Journal of Computer Vision*, vol.1, no.10, pp.128–142, 2002.