Research Article

A New Scheme for Keypoint Detection and Description

Lian Yang^{1,2} and Zhangping Lu¹

¹College of Mechanical Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China

²Department of Mathematics, Hunan University of Humanities, Science and Technology, Loudi, Hunan 417000, China

Correspondence should be addressed to Lian Yang; ylanyang@163.com

Received 9 February 2015; Revised 17 April 2015; Accepted 4 May 2015

Academic Editor: Anders Eriksson

Copyright © 2015 L. Yang and Z. Lu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The keypoint detection and its description are two critical aspects of local keypoints matching which is vital in some computer vision and pattern recognition applications. This paper presents a new scale-invariant and rotation-invariant detector and descriptor, coined, respectively, DDoG and FBRK. At first the Hilbert curve scanning is applied to converting a two-dimensional (2D) digital image into a one-dimensional (1D) gray-level sequence. Then, based on the 1D image sequence, an approximation of DoG detector using second-order difference-of-Gaussian function is proposed. Finally, a new fast binary ratio-based keypoint descriptor is proposed. That is achieved by using the ratio-relationships of the keypoint pixel value with other pixel of values around the keypoint in scale space. Experimental results show that the proposed methods can be computed much faster and approximate or even outperform the existing methods with respect to performance.

1. Introduction

Local keypoints matching is finding corresponding points between two or more images of the same scene or object. It has demonstrated considerable success in many computer vision and pattern recognition applications such as object recognition [1], motion tracking, wide-baseline stereo [2], texture recognition, image retrieval [3, 4], robot navigation [5], video data mining, recognizing of building panorama [6], stereo correspondence, recovering camera motion, and 3D reconstruction.

The two critical aspects of local keypoint matching are detection and description of keypoints. Firstly, the detection of keypoints determines the stable keypoints that are to be matched; that is, we must localize the keypoint by using position and the corresponding scale, and at the same time we determine the appropriate neighborhood that are used in computing the descriptors. Secondly, the description of a keypoint involves building a unique descriptor for each keypoint by describing it and its neighboring regions; that is, we must create a description of the keypoint; ideally, the description has to be distinctive and invariant under various transformations due to viewpoint change, rotation, scaling, illumination change, and so forth.

Many detectors have been recently proposed in the last few years. Moravec developed in [7] a corner detector used in image matching. The detector was refined to make it more repeatable when it is over a little transformation and close to edges by Harris and Stephens [8], named Harris corner detector. Harris detector is not scale-invariant. Rosten and Drummond proposed in [9] a FAST criterion for corner detection, with AGAST [10] extending this work for improved performance. But the FAST has not an orientation operator and does not produce multiscale features, and it has large response along edges [11]. For a scale-invariant blob detector proposed by Lindeberg [12] which introduced the concept of automatic scale selection, this blob detector is defined by a maximum of the normalized Laplacian. A Harris-Laplace detector and a Hessian-Laplace detector used to detect the keypoints by using scale-adapted Harris function or Hessian matrix, respectively, are proposed by Mikolajczyk and Schmid [13, 14]. The two detectors are robust and scale-invariant. Several other keypoint detectors have been summarized by [15].

After detecting keypoints, the next step of local keypoint matching is to describe keypoints. A lot of local descriptors for describing local keypoint have been developed [16, 17]. Lowe [1] proposed a scale-invariant feature transform (SIFT) descriptor based on the gradient distribution in the detected

regions. SIFT combines a scale-invariant region detector (called DoG for short) and is invariant to image scaling and rotation and partially invariant to illumination changes and 3D camera viewpoint. Several variants and extensions to improve SIFT have been proposed. Ke and Sukthankar [18] proposed the PCA-SIFT descriptor that applies PCA to reducing the dimensionality of SIFT descriptor vector from 128 to 36. The GLOH [19] is also an extension of the SIFT descriptor which changes SIFT's location grid and uses PCA to reduce the dimension of the SIFT. GLOH is more distinctive with respect to the same dimensionality. However, GLOH is computationally more expensive. In addition to PCA-SIFT and GLOH, Bay et al. presented a scale-invariant and rotation-invariant keypoint descriptor using integral images for image convolutions, which combines a keypoint detector and a descriptor called SURF [20]. These extensions focused primarily on improving the matching performances.

In recent years, several fast descriptors have been proposed. The BRIEF [21] is a keypoint descriptor that is designed for superfast description and matching and uses simple binary tests between pixels in a smoothed image patch. However, it is very sensitive to in-plane rotation. Rublee et al. in [11] proposed the ORB descriptor whose binary descriptor is rotation-invariant and robust to noise. At the same time, Leutenegger et al. in [22] presented a binary descriptor invariant to scale and rotation called BRISK. Recently Alahi et al. in [23] proposed a keypoint descriptor inspired by the human visual system and more precisely the retina, coined FREAK.

Inspired by the above presented detectors and descriptors, this paper proposed a new scheme for keypoint detection and description. The main contribution of this paper lies in two aspects. On the one hand, we apply the Hilbert curve scanning to converting a two-dimensional (2D) digital image into a one-dimensional (1D) gray-level sequence. Based on the 1D image sequence, we propose an approximation of DoG detector by using second-order difference-of-Gaussian function, coined DDoG detector. On the other hand, a new fast binary ratio-based keypoint descriptor is proposed. The primary motivation of descriptor is looking for some significant pixels around a keypoint. This is achieved by using the ratio-relationships of the keypoint pixel value with other pixel values around the keypoint in scale space. The proposed keypoint descriptor also uses a binary bit-string and has lower computational complexity than the existing descriptor.

The remainder of this paper is organized as follows. In Section 2 we firstly introduce the Hilbert curve, and then we construct the 1D image pyramid based on Hilbert curve scanning. Section 3 proposes a new detector called DDoG based on 1D image pyramid by zeroes detecting. Section 4 presents a new descriptor based on binary bit-string. Section 5 shows the experiments and results. Finally, the paper is concluded in Section 6.

2. 1D Image Pyramid Based on Hilbert Curve

2.1. Hilbert Curve Scanning. In our proposed keypoint detector, the first important step is to convert a 2D digital image into a 1D gray-level sequence by Hilbert curve scanning.

Hilbert curve is one of the space-filling curves published by G. Peano that visits every point in a square grid with a size of $2 * 2$, $4 * 4$, $8 * 8$, $16 * 16$, or any other power of 2 (showed in Figure 1) and has one-to-one mapping between an n -dimensional space and a one-dimensional space. Because the curve can keep the relevancy of neighboring points in the original space as far as possible, that is, fairly well preserving locality [24], it has been widely used in computer science especially in image processing. The detailed process of constructing a Hilbert curve scan is omitted which can be seen in [25]. Using the Hilbert curve scanning, a 2D digital image can be converted into a 1D gray-level sequence.

2.2. 1D Image Pyramid. Then, we can construct the image pyramid based on the 1D gray-level sequence of the image. Therefore, scale space of the 1D sequence is defined as a function of one-variable, $L(x, \sigma)$, that is produced from the convolution of one-variable Gaussian function, $G(x, \sigma)$, with 1D sequence of an image, $I(x)$:

$$L(x, \sigma) = G(x, \sigma) * I(x), \quad (1)$$

where $*$ is the convolution operation in x and y and

$$G(x, \sigma) = \frac{1}{\pi\sigma^2} e^{-x^2/\sigma^2}. \quad (2)$$

Then the difference-of-Gaussian function convolved with the 1D image sequence, $D(x, \sigma)$, can be computed as follows:

$$\begin{aligned} D(x, \sigma) &= (G(x, k\sigma) - G(x, \sigma)) * I(x) \\ &= L(x, k\sigma) - L(x, \sigma), \end{aligned} \quad (3)$$

where k is the factor of two nearby scales which is the same as standard SIFT.

In order to quickly find the keypoints and reduce the computation, our method using DDoG detector instead of DoG will be discussed much more in detail in next section.

3. DDoG Detector Based on 1D Image Pyramid

3.1. DDoG Detector. To simplify the structure of pyramid can effectively reduce the computation of constructing pyramid. According to the theory of differential and integral calculus, the local extremum point of a function corresponds to the zero of its first derivative. So we have

$$D(x, k\sigma) - D(x, \sigma) = DD(x, \sigma), \quad (4)$$

where $DD(x, \sigma)$ represents the second-order difference-of-Gaussian function and the value of k is the same as the standard SIFT. From (4), we can get

$$\frac{\partial D}{\partial \sigma} \approx \frac{D(x, k\sigma) - D(x, \sigma)}{k\sigma - \sigma} = \frac{DD(x, k\sigma)}{k\sigma - \sigma} \quad (5)$$

FIGURE 1: First 6 steps toward building the Hilbert curve in two dimensions.

and therefore

$$DD(x, k\sigma) = D(x, k\sigma) - D(x, \sigma) \approx \frac{\partial D}{\partial \sigma}(k\sigma - \sigma) \quad (6)$$

and because of $k\sigma - \sigma \neq 0$, $DD(x, \sigma) = 0 \Leftrightarrow \partial D / \partial \sigma \approx 0$.

This shows that the local maximum or minimum value obtained by DoG is actually the zero of the first derivative of the function. Because the zero detection is much easier than the extreme point detection, we can use the zero detection of first derivative function of the DoG to replace the local extreme point detection of the DoG.

Therefore, this paper proposes the DDoG detection algorithm. The first step of our method is to obtain the

second-order difference-of-Gaussian (DDoG) function by constructing DDoG pyramid. The second step is to find the local extremum by using zeroes detection of DDoG function, which determines the keypoint and its scale. The third step is to make accurate localization of keypoint using curve by fitting for the DDoG function.

3.2. Constructing DDoG Pyramid. The DDoG pyramid is based on DoG pyramid and the process of its construction is exemplified in Figure 2. Making a subtraction of the same octave in two adjacent layers in the DoG pyramid forms a layer of DDoG, whose scale is the same as the lower layer of the two adjacent layers in the DoG pyramid. For example,

FIGURE 2: Construction process of DDoG pyramid.

the first layer of the first octave of DDoG pyramid is obtained by making a subtraction between the second layer $D(x, k^2\sigma)$ of the first octave of DoG pyramid and the first layer $D(x, k\sigma)$ of the first octave of DoG pyramid, and its scale is $k\sigma$. Therefore, the number of octaves in the DDoG pyramid is the same as in the DoG pyramid, and the number of layers of each octave in the DDoG has one less than in the DoG pyramid.

3.3. Zeroes Detection. In order to check whether the pixels whose absolute value of the second-order difference-of-Gaussian function is close to zero at every layer of DDoG pyramid, the algorithm has to set a proper threshold T ; the absolute value of second-order difference-of-Gaussian function of each pixel in DDoG pyramid is compared to T . If the value is less than or equal to T , then the pixel is considered a keypoint, and its location and scale (x_i, σ_i) are recorded.

Obviously, the threshold selection is very important. The larger the threshold, the more number keypoints extracted. However, some false keypoints may be produced which lead to the increased false match rate and the higher computation cost; the smaller the threshold, the less number keypoints extracted, which is not enough to reflect the distribution of the keypoints and leads to the lower matching reliability.

In order to study the relationship between the threshold T and the performance of the detector, we can determine the best choices by extensive experiments under the matching task. The results of the experiment are shown in Figures 3 and 4. These figures are based on a matching task by using 100 images from the Oxford Buildings dataset (the dataset is available at <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>). We can see that the proposed algorithm can realize best results when $T \in [350 \ 450]$; we set $T = 400$ in this paper.

3.4. Accurate Localization of Keypoint. The extreme point detected by previous steps is not the accurate extreme point but the extreme point of discrete space since the number of octaves and layers is limited; in other words, although the scale space is discrete rather than continuous, we can take advantage of interpolating to the known points of discrete

FIGURE 3: Average repeatability for different threshold.

space to obtain the accurate extreme point of continuous space, that is, subpixel interpolation proposed by Brown and Lowe [26]. In this paper, we can use curve fitting for the DDoG function; the fitting function is 2D quadratic function:

$$DD(x) = DD + \frac{\partial DD^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 DD}{\partial X^2} X, \quad (7)$$

where $X = (x, \sigma)^T$ is offset from keypoint, the accurate location of keypoint. $\hat{X}$ is acquired by finding the derivative of the 2D function with respect to x ; then setting it to zero, we can get

$$\hat{X} = -\frac{\partial^2 DD^{-1}}{\partial X^2} \frac{\partial DD}{\partial X}. \quad (8)$$

By substituting (8) into (7), giving

$$DD(\hat{X}) = DD + \frac{1}{2} \frac{\partial DD^T}{\partial X} \hat{X}. \quad (9)$$

FIGURE 4: Average numbers of keypoints for different threshold.

Notably, $X = (x, \sigma)^T$ represents the offset from the interpolating central point; when $\hat{X}$ is larger than 0.5 in x or σ , then it means that the interpolating central point has shifted to its neighboring points, so the keypoint is changed; then we can perform the interpolation instead of its neighboring point and ensure convergence of interpolation. In addition to this, $DD(x)$ represents the contrast of keypoints. In order to improve the stability of keypoints, we can remove unstable extreme point with low contrast. In this paper, the extremum point with the contrast less than $0.04/s$ is discarded, where s is the number of layers per octave.

It should be pointed out that the origin SIFT algorithm uses curve fitting for the DoG function and the fitting function is 3D quadratic function.

Compared with the original DoG algorithm, DDOG algorithm has a greater advantage that it can simplify the Gaussian pyramid and improve the speed of computation. Extrema detection of the DoG is carried out in a $3 \text{ pix} \times 3 \text{ pix} \times 3 \text{ pix}$ of 3D space; the determination of every extreme point needs to use the local information of the three layers of DoG image in pyramid; and the zero detection only needs to be performed in a linear space; each zero indeed only relates to a layer image of DDOG pyramid.

According to Lowe's theory, the performance of the algorithm is the best when the effective layers of DoG extremum detection, that is, which can detect the keypoints, are three layers and then each octave contains five layers in the DoG pyramid, so each octave contains six layers corresponding to LoG pyramid. But zero detection involves only one layer of image in the DDOG pyramid, so every layer is effective layer in this pyramid. In order to guarantee that there are three effective layers, every octave of DDOG pyramid should contain three layers; then every octave of the corresponding DoG contains four layers while there are five layers in LoG pyramid. By comparison it is known that when using DDOG constructs Gaussian pyramid, every octave can compute less one layer of Gaussian filter and it is possible to reduce Gaussian filtering computation of four-layer image for constructing four-octave Gaussian pyramid. Because of constructing

the Gaussian pyramid that accounted for more than 70% of total time in the DoG detection algorithm, the proposed algorithm can greatly improve the speed and reduce the time cost. Moreover, the computational complexity of DDOG algorithm for zeroes detection in the linear space is lower than the origin algorithm in 3D space, which further reduces the computational time and improves the algorithm in real time.

Therefore, the computation complexity of our method at the theory aspect is lower than that of the original one. Because the DDOG detector remains the essence of DoG detector, the matching performance should keep unchanged.

4. Fast Binary Ratio-Based Keypoint Descriptor

This paper proposed a novel method to compute the descriptor. The primary motivation is looking for some significant pixels around a keypoint, and the ratios of the keypoint pixel value with other pixel values around the keypoint will be kept invariant by means of certain data conversion in scale space. Inspired by [11, 21–23], we also propose to use binary strings as a keypoint descriptor, which we call FBRK (fast binary ratio-based keypoint).

4.1. Orientation. The histogram of gradient computations for original SIFT algorithm is very time consuming. This paper proposed a novel method to make our descriptor have rotation invariance.

First, according to the original Hilbert curve, it is to convert the 1D scale space into 2D scale space defined as $L(x, y)$. In fact, the $L(x, y)$ is Gaussian smoothed 2D image. Then we can select a 3×3 patch centered on every keypoint in their scale space and construct a matrix S and a matrix V with the same size of 3×3 ; each value of matrix S and matrix V is represented by S_{ij} and V_{ij} ($i, j = 1, 2, 3$), and the S_{ij} is computed by (10) as follows:

$$S_{ij} = \frac{L(i, j) - l_{\min}}{l_{\max} - l_{\min}} \quad (i, j = 1, 2, 3), \quad (10)$$

where $L(i, j)$ is value of pixel $p(i, j)$ in the patch and $l_{\min}$ and $l_{\max}$ are the minima and maxima in the patch. Then, we can compute matrix V as follows:

$$V_{ij} = \begin{cases} 1 & \text{if the keypoint is maxima and } S_{ij} < RT_{\min} \\ 1 & \text{if the keypoint is minima and } S_{ij} > RT_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (i, j = 1, 2, 3), \quad (11)$$

in which the $RT_{\min}$ and $RT_{\max}$ are two thresholds.

Accordingly, we can obtain binarization of all the pixels around this keypoint. These binary values in clockwise order form a state vector. Obviously there are 2^8 kinds of state. But according to certain rules, these states can be divided into

FIGURE 5: Two examples of state diagram.

FIGURE 6: The two state vectors corresponding to the two state diagrams as shown in Figure 5.

FIGURE 7: The bitwise circular moving results for two state vectors as shown in Figure 6.

36 categories by bitwise circular moving. For example, Figure 5 shows two state diagrams.

The two state vectors corresponding to the upper two state diagrams can be described as shown in Figure 6.

The two state vectors can be bitwise circular moved into the vector as shown in Figure 7.

The vector is the smallest number in results of bitwise circular moving. Thus the two state vectors correspond to the same state which means that the keypoint has eight directions.

Then we can rotate the image according to the length of bitwise circular moving to ensure rotation-invariant for our descriptor proposed in the next step.

4.2. Feature Vector. After rotating the image in scale space, we can select a bigger $n * n$ patch, such as $n = 7, 9, 11, 13$. First, we construct a matrix D and a matrix B with the same size of patch; each value of matrix D and matrix B is represented by D_{ij} and B_{ij} ($i, j = 1, 2, \dots, n$), and the D_{ij} is computed by (12) as follows:

$$D_{ij} = \frac{1}{8} \sum_{k=1}^8 |L(i, j) - l_k| \quad (i, j = 1, 2, \dots, n), \quad (12)$$

where $L(i, j)$ is value of pixel $p(i, j)$ in the patch and l_k ($k = 1, 2, \dots, 8$) is one of eight neighboring pixel values. Then, we can compute ratio r by (13)

$$r = \begin{cases} \frac{D_{ij}}{d_k} & \text{if } D_{ij} < d_k \\ \frac{d_k}{D_{ij}} & \text{otherwise} \end{cases} \quad (i, j = 1, 2, \dots, n), \quad (13)$$

1	1	1	1	1	1	1
1	3	3	3	3	3	1
1	3	5	5	5	3	1
1	3	5		5	3	1
1	3	5	5	5	3	1
1	3	3	3	3	3	1
1	1	1	1	1	1	1

FIGURE 8: An example of assigning weight to each bit in matrix B ($n = 7$).

where d_k is pixel value corresponding to keypoint in matrix D .

Furthermore, we can compute matrix B as follows:

$$B_{ij} = \begin{cases} 1 & \text{if } r \leq RT \\ 0 & \text{otherwise} \end{cases} \quad (i, j = 1, 2, \dots, n). \quad (14)$$

Accordingly, we obtain binarization bit-string of all the pixels around the keypoint except for the keypoint itself. And the binary bit in matrix B ranked in accordance with certain rules, such as from top to bottom, left to right, forms a feature vector as a descriptor.

In order to improve the performance of scale-invariant, we can assign a weight which is used as an integer for each pixel tin the patch according to the distance from keypoint. The shorter the distance is, the greater the weight is. The specific method is that binarized bits are copied according to their weight and inserted into their original position in the bit-string. The weight determines the length of copied binary bit. For example, if a bit in matrix B is 1 and its weight is 5, the bit is copied 5 times, that is, 11111, and the copied bits are inserted into the original position. This paper considers the following rule for setting weight: if a pixel is adjacent to keypoint, its weight is 5; if the distance of a pixel from keypoint is 2 pixels, its weight is 3; the others are 1. For example, if $n = 7$, the weight of each bit in matrix B is shown in Figure 8.

The similarity between descriptor vectors is clearly measured by the Hamming distance between the corresponding binary vectors.

In this paper, we set $RT_{\max} = 0.8$, $RT_{\min} = 0.2$, $RT = 0.2$, $n = 11$ and will show in experiment section that these parameters lead to good performance, speed, and storage efficiency. The length of descriptor vector is $11 * 11 - 1 + 8 * (5 - 1) + 16 * (3 - 1) = 184$ bit.

Our method uses the ratio and threshold to obtain the binary-string both the orientation assignment stage and the keypoint description stage. But we have done some data conversion operations at the beginning to obtain the binary-string. For orientation assignment, nine pixel values have been mapped to $[0, 1]$ by using interval-valued conversion, so

TABLE 1: Average calculation time for the detectors over 10 independent runs (Graffiti scene, six images).

Detector	DoG	SURF	Harris-Laplace	DDoG
Average time (ms)	986	407	2139	382

that proposed descriptor can greatly reduce the effects of illumination change. The proposed method uses the difference of pixels value between a keypoint and its eight neighboring pixels when computing descriptor, which can effectively reduce the effects of illumination change. Using threshold can find the significant pixels around the keypoint. Assigning appropriate weights to these significant pixels can ensure that the proposed descriptor has more robustness to scale changes. The following experimental results shown in next section will demonstrate the above.

5. Experimental Results

We compare our detector and descriptor to others with respect to performance and speed. The original implementations are used in the comparison for these detectors and these descriptors. The datasets used in the experiments are the INRIA dataset [19] which contains eight groups of images with five geometric and photometric transformations for different scene types including viewpoint change, zoom + rotation, image blur, illumination change, and JPEG compression. The INRIA datasets are available at <http://www.robots.ox.ac.uk/~vgg/research/affine/>.

For the detector comparison, we selected these sequences: *Graffiti* (viewpoint changes), *Boat* and *Bark* (zoom and rotation), and *Leuven* (lighting changes). And we use a criterion similar to the one proposed in [21], that is, repeatability score. The repeatability score indicates the average number of the detected keypoints in both images.

The detector is compared to the DoG detector, the SURF detector and the Harris-Laplace detector proposed by Mikolajczyk and Schmid [19]. The default thresholds for all detectors are used. In addition to this, for all the experiments reported in this paper, the same parameters are used, overlap error is fixed to 40%, and normalized size is fixed to 30 pixels as an example. In Table 1 the computational times for keypoints detection of *Graffiti* sequences are listed. The repeatability scores of these detectors are shown in Figure 9. As it can be seen in Table 1 our DDoG detector is more than 2 times faster than DoG and 5 times faster than Harris-Laplace or even better than SURF detector. At the same time, the repeatability for our detector is comparable (*Boats*, *Bark*, and *Leuven*) or even better (*Graffiti*) than for the others. Specifically, Figures 9(a) and 9(c) show our approach slightly better than the competitors for structure scene and slightly weaker than the competitors for texture scene, while the repeatability scores for the Boat sequence and Leuven sequence are comparable for all detectors.

Note that in all experiments mentioned in this paper, the computation times have all been measured on an Intel Core i5-3230M standard PC with 2.6 GHz.

For the descriptors, we compared our method FBRK with popular descriptors, that is, SIFT, SURF, BRISK, ORB, and FREAK, which are based on keypoints detected using their respective detector except for FREAK that uses SURF detector. We use the *recognition rate* in image pairs proposed in [11] to quantify these descriptors performance, for both images of a pair and for a given number of corresponding keypoints between them; it quantifies how often the correct match can be established by using these methods for description. Moreover, we selected the *Graffiti*, *Wall*, *Bark*, *Trees*, *Leuven*, and *Ubc* for the descriptor comparison. Since the NNDR (nearest neighbor distance ratio) matching strategy is more robust and its precision is higher than the NN (nearest neighbor based) matching and the threshold-based matching [19], though the performance of FBRK is comparable for the three strategies, only results on NNDR matching are shown in Figure 10 in this paper. The ratio is fixed to 0.7 for these experiments.

In order to evaluate the performance for image with viewpoint change, we selected the *Graffiti* and *Wall* sequences. For *Graffiti* sequence (as shown in Figure 10(a)), the trend in recognition rate is clearly downward for each descriptor, but the SIFT is the best performing descriptor and our method is either as good as other descriptors or slightly better. For *Wall* sequence (as shown in Figure 10(b)), the performances of six descriptors are similar at the first three comparisons, but for the last two comparisons (i.e., 1v5 and 1v6) the SIFT and FBRK are better than other methods. In order to evaluate the performance for image rotation, we used the *Boat* sequences with scale and rotation change, *Boat* sequence is challenging for six descriptors and SIFT outperforms the other descriptors, but our proposed method is comparable or better than other competitors except SIFT (as shown in Figure 10(c)). In order to evaluate the performance for blur images by selecting the *Trees* sequence, our method is the best performing descriptor, followed by ORB method (as shown in Figure 10(d)). This is possibly because our method provides more distinctive information than all the other five descriptors due to applying scale space and assigning weight in descriptor vector. In order to evaluate the performance for images with light change, we selected the *Leuven* sequence. For *Leuven* sequence (as shown in Figure 10(e)), FBRK outperforms the other five approaches. It happens possibly because there is no any rotation change in the sequence and our method provides more robustness than all the other five descriptors for light change. At the same time, we also evaluate the performance for JPEG compression images by using *Ubc* sequence. For *Ubc* sequence (as shown in Figure 10(f)), the leader is our proposed method, followed by ORB and SIFT. The reason is that there is no rotation change in the sequence which is similar to the *Leuven* sequence, while the image compression is similar to scale change especially for regions around keypoint.

The experiment results show that our proposed method has leading performance under image blur, light, and JPEG compression and is comparable to other competitors for viewpoint and scale changes, while for rotation changes our method is slightly weaker than SIFT descriptor and better than other descriptors.

FIGURE 9: Repeatability scores for image sequences, from (a) to (d), *Graffiti*, *Boat*, *Bark*, and *Leuven*.

At the same time, Table 2 compares the average running time between the six methods. The measure in the table is computed between the first image of *Graffiti* as a reference image and the other images in *Graffiti*. The first row shows the time needed to calculate the descriptor and the second row shows the matching time. FBRK is much faster than other descriptors and even faster than ORB both for description and for matching. Notably, the result in the table is the time per keypoint in millisecond.

6. Conclusions

This paper presented a new scheme for keypoint detection and description. The proposed DDoG detector achieves a considerable speed-up and approximates the DoG detector by using second-order difference-of-Gaussian function based on 1D image sequence. The proposed keypoint descriptor has leading performance under these conditions of image

TABLE 2: This table shows the average running time for six methods over 10 independent runs, in milliseconds. The measure in the table is computed between the first image of *Graffiti* as a reference image and the other images in *Graffiti*. The result in the table is the time per keypoint.

Time per keypoint	SIFT	SURF	BRISK	ORB	FREAK	FBRK
Representing time	2.071	0.815	0.416	0.071	0.072	0.058
Matching time	0.258	0.298	0.060	0.020	0.054	0.019

blur, light, and JPEG compression and is comparable to other competitors for viewpoint and scale changes, while for rotation changes our method is slightly weaker than SIFT descriptor and better than other descriptors. We have also

FIGURE 10: Recognition rates on (a) *Graffiti*, (b) *Wall*, (c) *Boat*, (d) *Trees*, (e) *Leuven*, and (f) *Ubc*.

demonstrated that our descriptor tends to be faster than other state-of-the-art ones.

In future work, one of the issues that we have not adequately addressed is that performance of rotation-invariance is nevertheless unsatisfactory. This is possibly because each keypoint has been assigned only eight directions. We will improve the performance of rotation-invariance for our

keypoint descriptors so that it can compete with state-of-the-art ones in a wider set of situations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Hunan Provincial Natural Science Foundation of China (no. 13JJ6081), Scientific Research Fund of Hunan Provincial Education Department (no. 14C0598), and the Higher Education Innovative Foundation for Doctoral Candidate of Jiangsu Province, China (no. CXZZ13.0658).

References

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference (BMVC '02)*, Cardiff, UK, September 2002.
- [3] V. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proceedings of the 12th ACM International Conference on Multimedia*, pp. 869–876, October 2004.
- [4] T. Tuytelaars and L. V. Gool, "Content-based image retrieval based on local affinity invariant regions," in *Proceedings of the 3rd International Conference on Visual Information and Information Systems (VISUAL '99)*, pp. 493–500, Amsterdam, The Netherlands, June 1999.
- [5] S. Se, D. Lowe, and J. Little, "Local and global localization for mobile robots using visual landmarks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 414–420, IEEE, Maui, Hawaii, USA, November 2001.
- [6] M. Brown and D. G. Lowe, "Recognising panoramas," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 1218–1225, October 2003.
- [7] H. Moravec, "Rover visual obstacle avoidance," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 785–790, 1981.
- [8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 23.1–23.6, Manchester, UK, 1988.
- [9] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 430–443, Springer, Berlin, Germany, 2006.
- [10] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Computer Vision—ECCV 2010: Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Part II*, vol. 6312 of *Lecture Notes in Computer Science*, pp. 183–196, Springer, Berlin, Germany, 2010.
- [11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2564–2571, November 2011.
- [12] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [13] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision*, pp. 525–531, July 2001.
- [14] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceedings of the European Conference on Computer Vision*, pp. 128–142, May 2002.
- [15] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1–2, pp. 43–72, 2005.
- [16] A. E. Johnson and M. Hebert, "Recognizing objects by matching oriented points," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 684–689, June 1997.
- [17] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision—ECCV '94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6 1994 Proceedings, Volume II*, vol. 801 of *Lecture Notes in Computer Science*, pp. 151–158, Springer, Berlin, Germany, 1994.
- [18] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II-506–II-513, June–July 2004.
- [19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [20] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Proceedings of the European Conference on Computer Vision*, pp. 778–792, Crete, Greece, September 2010.
- [22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, vol. 65, pp. 2548–2555, Barcelona, Spain, November 2011.
- [23] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: fast retina keypoint," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 510–517, IEEE, Providence, RI, USA, June 2012.
- [24] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124–141, 2001.
- [25] A. R. Butz, "Alternative algorithm for Hilbert's space filling curve," *IEEE Transactions on Computers*, vol. 20, no. 4, pp. 424–426, 1971.
- [26] M. Brown and D. G. Lowe, "Invariant features from interest point groups," in *Proceedings of the British Machine Vision Conference*, pp. 656–665, 2002.