Image Matching Based on Binarized SIFT Descriptors

Hui Huang[a], and Yan Ma[b, *]

College of Information and Electrical Engineering, Shanghai Normal University, Shanghai, 200234, China

[a]huanghui@shnu.edu.cn; [b]ma-yan@shnu.edu.cn

*The corresponding author

Keywords: Image matching; Binarization; Hamming distance; Feature description.

Abstract: The scale invariant feature transform (SIFT) algorithm is particularly effective in distinctive feature extraction. However, its matching is time consuming. The reason lies in that the Euclidean distance is used to measure the similarity of two SIFT descriptors in the SIFT matching. To improve the matching efficiency, in this paper, we present a novel image matching scheme (BI-SIFT) based on Binarized SIFT descriptors. First, 128-D SIFT descriptor is converted into 256-bit binarized SIFT (BSIFT) descriptor which retains the distinctive power of the original descriptor. Generally, the distance similarity meature between BSIFT descriptors by Hamming distance. However, it can introduce some extra false matches in the matching phase. Therefore, to avoid this problem, we also present a novel distance metric method for BSIFT descriptors. We evaluate our method on the UKBench data set. Experimental results show the superior performance of BI-SIFT method outperforms the state-of-the-art algorithms in image matching.

1. Introduction

Image matching is a crucial step in many image analysis tasks, such as image retrieval, image categorization and object recognition. There are basically two kinds of matching method, based on gray level of matching and based on feature matching respectively [1]. Promising results to image matching by utilizing local features were shown in previous works [2-6]. Like SIFT local descriptor, SIFT-based methods have been widely applied in image matching [7]. The SIFT algorithm extracts image features by searching the keypoints in the image, and then calculating the descriptors from the patch around the keypoints. The patch is first divided into 16 areas with 8 directions in each area, and each direction is given a value. Finally, 128-D descriptor is obtained. The 128-D descriptor is robust to variance in images (e.g., scale, rotation, and illumination variance) [8].

In the matching procedure, the 128-D descriptors of all keypoints in two images are extracted. The 128-D descriptors of each keypoint of the first image are compared to those of the second one. The Euclidean distance is used as the similarity measurement of two descriptors to find the nearest matching keypoint. SIFT algorithm usually generates hundreds to thousands of keypoints for each image. And correspondingly, the SIFT features could be numerous in a large image database. Moreover, the distance computation involves taking square root. Therefore, image matching in the SIFT method to large-scale image database is highly time-consuming.

To solve this problem, many approaches have been proposed in the last few years. These methods are classified into two types. The first is to reduce the computation complexity by decreasing the number [9] and dimension [10] of SIFT descriptors. Alitappeh et al. [9] proposed a method which uses the clustering technique to reduce the number of keypoints by omitting similar points. Their method decreases the time complexity in matching. However, the clustering procedure needs some time, and thus, the total processing time still has not been significantly reduced. Ke et al. [10] proposed applying PCA to reduce the size of the descriptor and thereby decreasing the feature matching time. However, this method needs an offline stage to train and estimate the covariance matrix used for PCA projection. This typically requires the system to collect and trains a large number of images. The second way to increase the matching speed is the binarized descriptor

Copyright © (2019) Francis Academic Press, UK DOI: 10.25236/isrme.2019.017

approach [11], which converts the SIFT descriptors to binarized SIFT (BSIFT) descriptors. Ni [11] first proposed a binary string approach for SIFT keypoints. Chen et al. [12] proposed to compare the absolute difference of 128-D SIFT descriptor with the threshold, and the comparison results were denoted by binary digits (0 or 1). This approach was simple and drastically decreased the matching time but required predetermining the threshold. Zhou et al. [13] compared the 128 values of a SIFT descriptor individually with two threshold values. The comparison results were denoted by three combinations, namely 11, 10, and 00, specifically, the 128 values of the SIFT descriptor were ranked in descending order, and the 32th and 64th values were exploited as the thresholds. This approach can improve matching accuracy to some extent. However, 2-bit binary number can express 2^2=4 possible states. Only 3 states have been used in [13]. Su *et al.* [14] presented a reflection invariant binary descriptor named MBR-SIFT and a fast matching algorithm that includes a coarse-to-fine two-step matching strategy in addition to two similarity measures for the MBR-SIFT descriptor. Nickfarjam et al. [15] made use of SIFT for binary image matching by taking the power of Hough Transform in line detection, which had good performance for binary images containing straight lines. Additionally, The Hamming distance between two BSIFT descriptors is calculated during the feature matching process. The distance computation between two BSIFT descriptors is hence reduced to more efficient bit-wise operations instead of square root, and therefore, the feature matching time can be greatly decreased. The Hamming distance between two BSIFT descriptors may not be consistent with the Euclidean distance between them, therefore, SIFT binarization are adopted to improve the computation efficiency in the image matching process, with sacrifice of accuracy to some extent. Moreover, the Hamming distance, which can introduce some extra false matches in the matching phase.

Hence, we take fully into account the problems of highly time-consuming of SIFT method and Hamming distance can introduce some extra false matches in the image matching. In this paper, we propose a novel SIFT descriptor binarization approach in which the difference values of 128-D SIFT descriptor are compared with threshold and the results are expressed with 2-bit binary number with 4 states. The value for the threshold is derived by the linear relationship between the threshold and the standard deviation of 128-D SIFT descriptor. To avoid the false matches introduced by the Hamming distance, we redefine distance similarity measure between 256-bit BSIFT descriptors. Finally, we conduct image matching experiments on five representative image pairs with rotation, scale, viewport, illumination and blur variance from the UKBench data set. Compared with other state of the art algorithm, the proposed method is better both in accuracy and efficiency.

This paper is organized as follows. Section 2 describes the proposed BI-SIFT method, which includes a novel SIFT descriptor binarization approach and distance similarity measure between 256-bit BSIFT descriptors. The experimental results and analysis that evaluate the performance of the proposed method is in Section 3. Finally, Section 4 draws together some conclusions.

2. Proposed method

2.1. BSIFT Descriptor.

In this stage, the 128-D SIFT descriptor vector $(D_0, D_1,\ldots, D_{127})$ is transformed into a binary string. First, the difference value AD_i (i=1, 2,…, 128) of the two adjacent values in a descriptor (D_i and D_{i+1}) is calculated according to Eq.1.

$$AD_i = \begin{cases} D_{i+1} - D_i, & if\ i < 127 \\ D_0 - D_{127}, & otherwise \end{cases} \tag{1}$$

The method for binarizing AD_i can be classified into two categories. The first category [12] proposed to compare AD_i with the predefined threshold M. The comparison result is denoted by zero or one according Eq.2 and Eq.3, where M is the average or median value of 128-D SIFT vector.

$$AD_i = \begin{cases} |D_{i+1} - D_i|, & if\ i < 127 \\ |D_0 - D_{127}|, & otherwise \end{cases} \tag{2}$$

$$b_i = \begin{cases} 0, & if\ AD_i \le M \\ 1, & otherwise \end{cases} \tag{3}$$

The second category [13] directly compares each D_i of 128-D vector with two thresholds, M_1 and M_2. The comparison results are denoted by 11, 10, and 00, as shown in Eq.4.

$$b(i, i+128) = \begin{cases} (1,1), & if\ D_i > M_1 \\ (1,0), & if\ M_2 < D_i \le M_1 \\ (0,0), & if\ D_i \le M_2 \end{cases} \tag{4}$$

In essence, the first category converts the original 128 decimal values to a 128-bit binary value, which decreases the memory requirements and reduces the matching time. However, in this method, the absolute difference value is compared with the threshold and the sign of difference value is neglected. Correspondingly, (1, 10) and (10, 1) will be binarized as the same value and completely different values, such as (1, 10, 1) and (10, 1, 10), will be categorized into the same values, which will reduce the distinctive power of SIFT descriptors. As for the distinctive power of SIFT descriptors, the second category is better than the first one. However, the binarization result is related to the two thresholds, M_1 and M_2. Furthermore, from the perspective of information theory, a 2-bit value can represent four states, but [13] only adopted three states. Therefore, to solve the aforementioned problems, the proposed binarization method retains the sign of difference value and adopts four states for 2-bit binary value, as shown in Eq.5.

$$b_{2*i,2*i+1} = \begin{cases} 00 & if\ AD_i \le (-T) \\ 01 & if\ (-T) < AD_i < 0 \\ 10 & if\ 0 \le AD_i < T \\ 11 & if\ AD_i \ge T \end{cases} \tag{5}$$

where T is a positive. Eq.5 converts the original 128-D descriptor into a BSIFT descriptor with a 256-bit binary value, which is expressed as $B=\{b_0, b_1, \ldots, b_{255}\}$.

2.2. Threshold Configuration.

In the process of binarizing 128-D descriptors of SIFT features in the image, incorrect threshold (too large or too small) will reduce the distinctive power of binary strings and further affect the distance between 128-D BSIFT descriptor pairs as well as the matching results [16]. In the configuration of the threshold value, hard-threshold [12] will lead to quantization error to some extent [17], whereas soft-threshold is adaptive to the data variation and is superior to hard-threshold. In the existing soft thresholding methods, the mean or median of 128-D vector is usually exploited as the threshold [13]. Yet both of them can not reflect the diversity of data distribution in the 128-D descriptor. AD_i is compared with the threshold in Eq.5. Intuitively, considering that the goal of the threshold configuration is to reflect the diversity between AD_i as much as possible. Therefore, we exploit the standard deviation of the 128-D SIFT vector ($D_0, D_1, \ldots, D_{127}$) to configure the threshold, as shown in Eq.6 to Eq.8.

$$\overline{D} = \frac{\sum_{j=0}^{127} D_j}{128} \tag{6}$$

$$\sigma = \sqrt{\frac{\sum_{j=0}^{127}\left(D_j - \overline{D}\right)^2}{128}} \tag{7}$$

$$T = a \times \sigma + b \tag{8}$$

Where $\overline{D}$ and σ are the mean and standard deviation of the 128-D SIFT vector (D_0, D_1,…, D_{127}), respectively; and a and b are constants. Through numerous experiments, the optimal a and b values were determined to be 3.7 and 0, respectively.

Generally, the similarity between 128-D descriptors for SIFT keypoints is determined by Euclidean distance, whereas Hamming distance is used for the similarity between binarized descriptors. In order that the binarized descriptors retain the distinctive power of the original 128-D descriptor, it is required that Hamming distance of binarized descriptors is consistent with Euclidean distance of SIFT descriptors, that is, the greater the Euclidean distance of 128-D descriptors is, the greater the Hamming distance of the corresponding BSIFT descriptors is, and vice versa. To demonstrate the efficiency of the proposed thresholding approach, both Hamming distance and Euclidean distance of keypoint pairs for numerous images are tested. The proposed method, which is denoted by BI-SIFT1 in Figure 1, is compared with Chen's [12] and Zhou's [13] methods. As shown in Figure 1, when the Euclidean distance for Chen's and Zhou's methods is within the range of 32 to 40, the corresponding Hamming distance is the same value. In contrast, the relationships between Hamming distance and Euclidean distance for the proposed binarization method remain consistent, which further demonstrates the accuracy of the proposed thresholding approach.

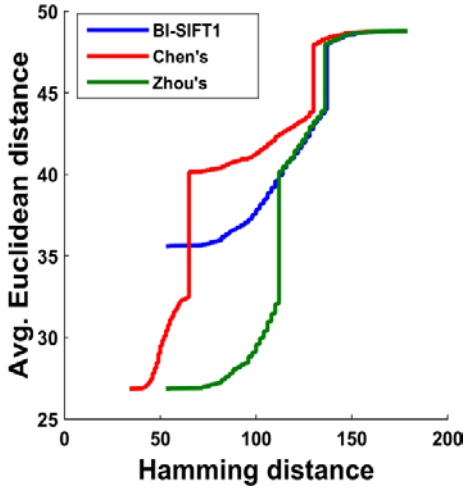

Fig 1. Euclidean Distance Versus Hamming Distance

2.3. Distance Metric of the Binarized Descriptors.

After feature extraction from the matched image and feature vector binarization, the following step is the distance metric of the BSIFT descriptors, that is, binary vectors. In the BSIFT methods, Hamming distance is exploited to the similarity between BSIFT descriptors. In order to improve the matching accuracy, we propose the improved Hamming distance metric to measure the distance according to the distribution characteristics of the binary values.

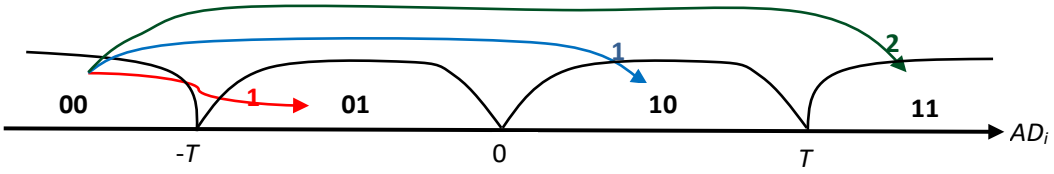

Fig 2. The Encoding of AD_i Versus Hamming Distance

According to Eq.5, AD_i is encoded as a set of binary values 00, 01, 10, and 11 according the relationship between AD_i and threshold T. In general, the distance between one of the binary values and the rest binary values should be the same. However, if we directly use Hamming distance, the result is not satisfactory. As shown in Figure 2, AD_i axis is divided into four parts (e.g., 00, 01, 10, and 11) by $-T$, 0, and T. Suppose that binary value 00 is selected as reference. The Hamming distance between 00 and 01, 10 is 1, whereas the Hamming distance between 00 and 11 is 2, which will lead to inaccurate matching. To avoid this problem, in the process of calculating the distance between BSIFT descriptors, we add 1 to counter P only when the Hamming distance between BSIFT descriptors is 0, which means both BSIFT descriptors are the same. Furthermore, because AD_i is encoded with 2-bit binary values, BSIFT descriptor should be equally divided by $2^n(0\leq n\leq 7)$ in the calculation of the distance between 256-bit BSIFT descriptor pairs. In this paper, we set n=2, that is, 256-bit BSIFT descriptor are equally divided into 64 parts. In other words, each part contains 4-bit. The main reason is that each 4-bit binary values in 256-bit BSIFT descriptor reflect the relationship between D_i in the original 128-D SIFT descriptor and its previous value D_{i-1} or the following value D_{i+1}. When two SIFT descriptors are similar to each other, the size relationships between the corresponding D_i and D_{i-1} (or between D_i and D_{i+1}) in the two descriptors should be highly correlated. In order to reflect this correlation, each 256-bit BSIFT string ($B=\{b_0,b_1,\cdots,b_{255}\}$) is equally divided into 64 parts, and each part contains 4 bits, that is, $B=\{bg_0,bg_1,\cdots,bg_{63}\}$, where $bg_i=\{b_{i*4},b_{i*4+1},b_{i*4+2},b_{i*4+3}\}$. Suppose we need to calculate the distance between two BSIFT strings $B^1=\{bg_0^1,bg_1^1,\cdots,bg_{63}^1\}$ and $B^2=\{bg_0^2,bg_1^2,\cdots,bg_{63}^2\}$. Each pair of corresponding part (e.g., bg_i^1 and bg_i^2) is compared. If the Hamming distance between the two parts is 0, P is increased by 1. Otherwise, P remains the same. Then the counter P is normalized by 64. A large value of P indicates that two binary strings B^1 and B^2 are more similar and vice versa. Generally, it is required that when two binary strings B^1 and B^2 are more similar, the value of P is smaller, and vice versa. We here exploit arccos(P) to meet this requirement. As known to all, arccos(P) decreases monotonously in the interval [0,1], hence the aforementioned requirement is satisfied. Algorithm 1 provides the pseudocode for summarizing the aforementioned method of distance metric. In algorithm 1, the output P is the distance between B^1 and B^2.

Algorithm 1: Distance Metric

Input: the two binary strings to be matched, $B^1=\{b_0^1,b_1^1,\cdots,b_{255}^1\}$ and $B^2=\{b_0^2,b_1^2,\cdots,b_{255}^2\}$, and the counter P=0

Output: Distance P between B^1 and B^2

For i=0 to 63

$bg_i^1=\{b_{i*4}^1,b_{i*4+1}^1,b_{i*4+2}^1,b_{i*4+3}^1\}$

$bg_i^2=\{b_{i*4}^2,b_{i*4+1}^2,b_{i*4+2}^2,b_{i*4+3}^2\}$

If Hamming(bg_i^1,bg_i^2)=0

P=P+1

End For

P=arccos(P/64)

Given the image pair to be matched, each keypoint in one given image needs to compare with all the keypoints in another given image by calculating distance P as described in Algorithm 1. A common issue when working with keypoint-based feature matching is to establish a threshold for distinguish true matches from false matches[18]. To suppress matches that could be falsely matched, following Lowe's suggestion [7], we only accept the matches if the ratio between the distances to the nearest and the secondary nearest points is less than a predefined value distratio, as shown in Eq.9.

$$match\ or\ not?\begin{cases}match & if\ P_1<P_2*distratio\\ not\ match & otherwise\end{cases} \tag{9}$$

where P_1 and P_2 represent the smallest distance and secondary smallest distance, respectively. Additionally, the predefined value distratio should be selected according to realistic scenarios.

3. Experiment

We evaluated the proposed approach on the public dataset, the UKBench dataset [19], which contains 10,200 images from 2,550 object/scene groups. Each group consists of four images taken from different views or in different imaging conditions.

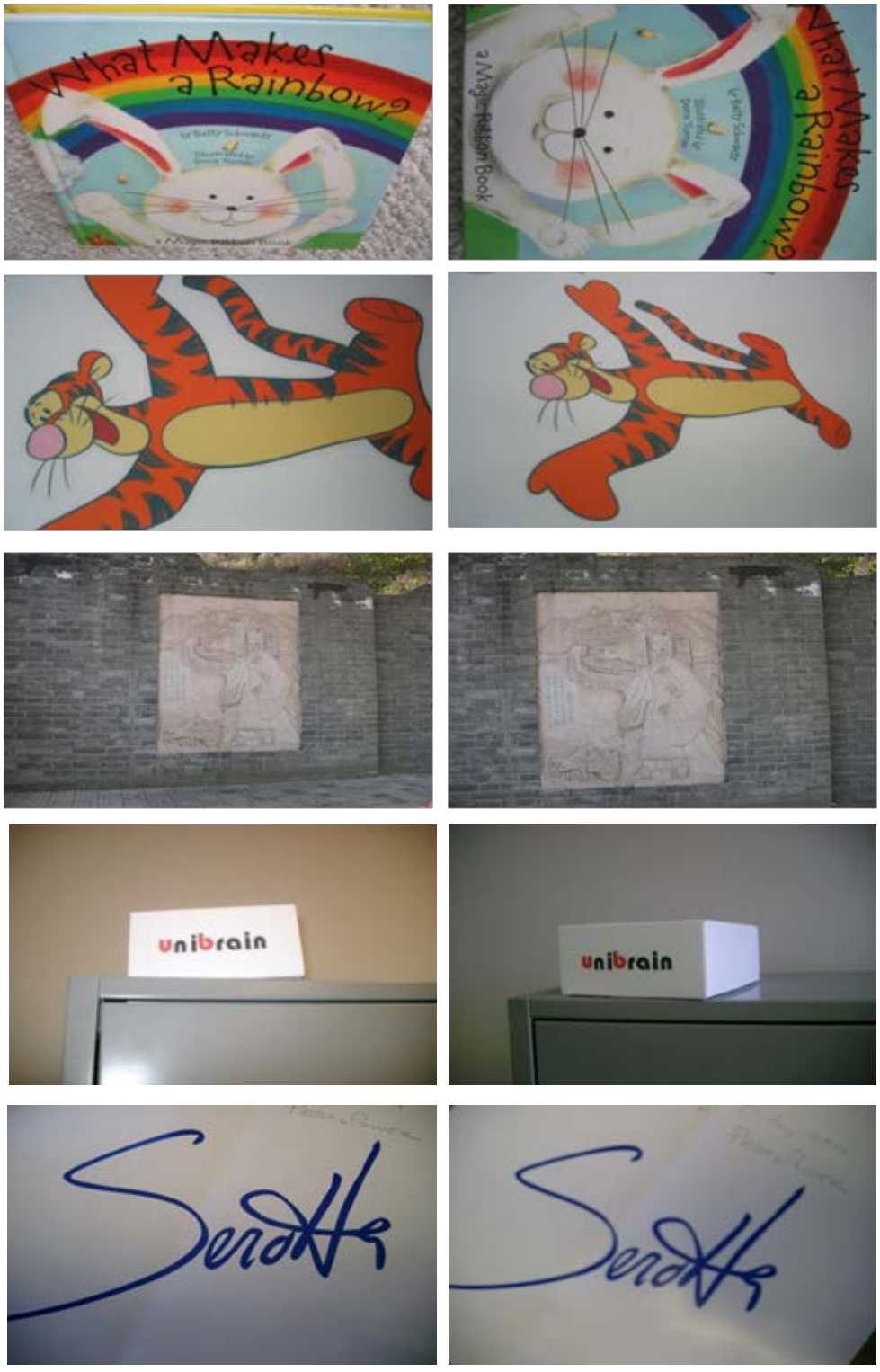

Fig 3. Examples of database image pairs

The representative image pairs are shown in Figure 3. Some of the results are presented with recall versus 1-precision [10, 20-22] , as defined in Eq.10 and Eq.11, where *tn* and *en* represent the number of correct matches and ground truth number of matches between the images, respectively and *fn* and *qn* represent the number of false matches and total number of matches between the images, respectively. To evaluate the performance of the image matching method, we need to determine matching pairs as much as possible with high accuracy [21], that is, when 1-precision is the same, the performance for the method with a higher recall is better.

$$recall = \frac{tn}{en} \tag{10}$$

$$1 - precision = \frac{fn}{qn} \tag{11}$$

Image matching experiments are conducted by randomly selected 200 image pairs with rotation, scale, viewport, illumination and blur variance from the UKBench data set. In order to show respectively the advantage of proposed binarization and distance metric, BI-SIFT1 includes the proposed binarization method and Hamming distance, and BI-SIFT includes the proposed binarization method and distance metric method. The two methods are compared with SIFT method, Chen's method, and Zhou's method in the image matching experiments. It should be noted that, the binarization methods in Chen's and Zhou's are exploited the methods in [12] and [13], respectively, the proposed distance metric method is used to evaluate the similarity of two binarized descriptors.

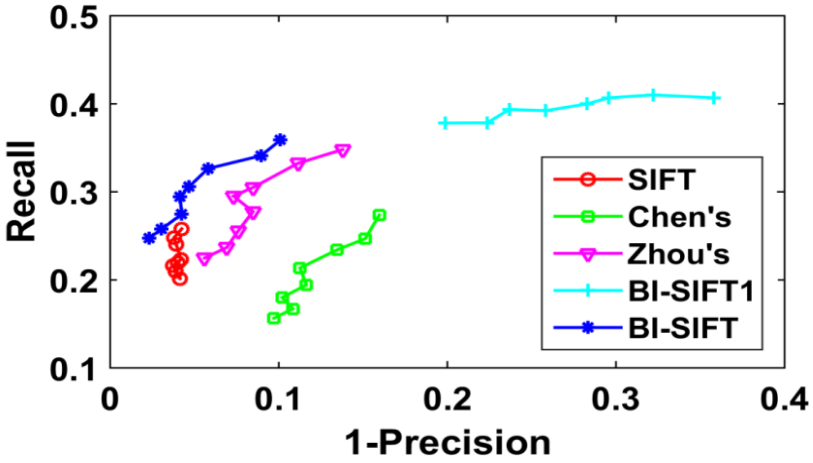

Fig 4. Recall versus 1 minus Precision for Five Methods

The matching results of BI-SIFT, BI-SIFT1, SIFT, Chen's, and Zhou's methods are presented in Figure 4 with recall versus 1 minus precision. Distratio in Eq.9 is in the interval [0.58, 0.65] for SIFT method and [0.83, 0.90] for other methods, respectively. It can be seen from Figure 4 that BI-SIFT has the best performance for accuracy. Therefore, it can be proven that the proposed binarization method and distance metric has better performance in image matching.

Table 1. The Efficiency for Five Methods

-	SIFT	Chen's	Zhou's	BI-SIFT1	BI-SIFT
Avg. feature binarization time for an image	-	0.150	0.213	0.480	0.480
Avg. matching time for an image	33.49	11.91	22.25	17.74	22.30
Matching speedup ratio (relative to SIFT)	1	2.81	1.51	1.89	1.50

It can be seen from the second row in table 1, in terms of the average feature binarization time for an image, our method spent the longest time than the other methods. This is due to the fact that, compared with Chen's and Zhou's, it is required to calculate the mean and standard deviation of the 128-D SIFT vector in the binarization process. In addition, Table 1 also shows that SIFT spent the longest than the other methods in matching.

In summary, the proposed method is superior to SIFT method in matching speed, accuracy and recall. Compared with Chen's and Zhou's methods, the proposed method can significantly improve accuracy as well as ensure the recall and the matching speed.

4. Conclusion

A novel SIFT descriptors binarization method has been presented in this paper. 128-D SIFT descriptor is converted into 256-bit binary string. Modified Hamming distance has been proposed to measure the similarity between binary strings. Image matching experiments are conducted by randomly select some image pairs with rotation, scale, viewport, illumination and blur variance from the UKBench data set. The results show the reliable matching performance. However, image matching performance is slightly lower than SIFT method. In the future, how to improve the recall and guarantee the high accuracy will be further studied.

Acknowledgements

This work was supported by the National Nature Science Foundation of China (No. 61501297, 61373004).

References

[1] Ping BI, "A Binocular Vision System for Object Distance Detection with SIFT Descriptors", International Journal of Hybrid Information Technology. vol. 5, no. 3, (2012), pp. 67-74.

[2] Amin Sedaghat, Hamid Ebadi, "Remote Sensing Image Matching Based on Adaptive Binning SIFT Descriptor", IEEE Transactions on Geoscience and Remote Sensing. vol. 53, no. 10, (2015), pp. 5283-5293.

[3] Yan Lin, Bo Liu, "Underwater Image Bidirectional Matching for Localization Based on SIFT", Journal of Marine Science and Application. vol. 13, no. 2, (2014), pp. 225-229.

[4] Amin Sedaghat, Hamid Ebadi, "Very High Resolution Image Matching Based on Local Features and K-Means Clustering", The Photogrammetric Record. vol. 30, no. 150, (2015), pp. 166-186.

[5] Wen Zhou, Chunheng Wang, "SLD: A Novel Robust Descriptor for Image Matching", IEEE Signal Processing Letters. vol. 21, no. 3, (2014), pp. 339-342.

[6] L.Seidenari, G. Serra, AD. Bagdanov, BA, Del, "Local pyramidal Descriptors for Image Recognnition", IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 36, no. 5, (2014), pp. 1033-1040.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision. vol. 60, no. 2, (2004), pp. 91-110.

[8] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 32, no. 5, (2010), pp. 815–830.

[9] R.J. Alitappeh, K.J. Saravi, F. Mahmoudi, "Key point reduction In SIFT descriptor used by subtractive clustering", Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications, (2012) July 906–911.

[10] Y. Ke, R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2004) July 503–513.

[11] Z.S. Ni, "B-SIFT: a binary SIFT based local image feature descriptor", Proceedings of the Fourth International Conference on Digital Home, (2012) November 117–121.

[12] Chun-Che Chen, Shang-Lin Hsieh, “Using binarization and hashing for efficient SIFT matching”, Journal of Visual Communication and Image Representation. vol. 30, (2015), pp. 86–93.

[13] Wengang Zhou, Houqiang Li, “BSIFT:Toward data-independent codebook for large scale image search”, IEEE Transactions on Image Processing. vol. 24, no. 3, (2015), pp. 967–979.

[14] M. Su, Y. Ma, X. Zhang, Y. Wang, and Y. Zhang, “MBR-SIFT: A mirror reflected invariant feature descriptor using a binary representation for image matching,” *PLOS ONE*, vol. 12, no. 5, p. e0178090, May 2017.

[15] A. M. Nickfarjam, H. Ebrahimpour-komleh, and A. A. A. Tehrani, “Binary image matching using scale invariant feature and hough transforms,” in *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, 2018, pp. 1–5.

[16] Zhen Liu, Houqiang Li, Liyan Zhang, “Cross-indexing of binary SIFT codes for large-scale image search”, IEEE Transaction on Image Processing. vol. 23, no. 5, (2014), pp. 2047–2057.

[17] Y. Usui, K. Kondo, “The SIFT image feature reduction method using the histogram intersection kernel”, Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, (2009) January 517–520.

[18] Andreas Persson, Amy Loutfi, “Fast Matching of Binary Descriptors for Large-scale Applications in Robot Vision”, International Journal of Advanced Robotic Systems. vol. 13, (2016).

[19] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recogniton, (2006) Sept. 2161–2168.

[20] Mikolajczyk K, Schmid C, “A performance evaluation of local descriptors”, IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 27, no. 10, (2005), pp. 1615–1630.

[21] Y.B. Zheng, X.S. Huang, S.J. Feng, “An Image Matching Algorithm Based on Combination of SIFT and the Rotation of Invariant LBP”, Journal of Computer-Aided Design and Computer Graphics. no.2, (2010), pp. 286–292.

[22] Kaiyang Liao, Guizhong Liu, Youshi Hui, “An improvement to the SIFT descriptor for image representation and matching”, Pattern Recognition Letters. vol. 34, no. 11, (2013), pp. 1211-1220.