Orientation and Scale Invariant Binary Descriptor Based on Haar Wavelet

Meng Yao*, Ke-Bin Jia* and Wan-Chi Siu†
*Beijing University of Technology, Beijing, China
E-mail: yaomeng@emails.bjut.edu.cn Tel: +86-10-67396691-20
E-mail: kebinj@bjut.edu.cn Tel: +86-10-67391090
† Hong Kong Polytechnic University. Hung Hom, Hong Kong
E-mail: enwcsiu@polyu.edu.hk

***Abstract*— In this paper, an orientation and scale invariant binary descriptor is proposed, which can be used in key-points matching systems. Conventionally, a binary descriptor is generated by comparing the intensities of pixels directly, such as those in Binary Robust Independent Elementary Features (BRIEF) and Oriented FAST and Rotated BRIEF (ORB). However, comparing intensities of pixels may lose the texture information in the region of interest, and lead to a high false match rate in a practical application. In our proposed method, the region of interest is segmented into grid cells and then the binary Haar wavelet responses are computed to store the texture information of the patch. Concretely, the texture information in each cell is expressed by the horizontal and vertical gradients and the polarity of intensity changes which are indicated by four components of Haar wavelet response. The binary descriptor is generated by comparing the Haar wavelet response in each pair of grid cells. Furthermore, to be scale and orientation invariant, the patch of key-points is rotated to the primary direction of the centroid vector in the image pyramid. Extensive experimental results show that our descriptor significantly outperforms other five state-of-the-art binary descriptors in key-point matching systems. The average percentage of correct matches of our method is 32.79% higher than that for FREAK and 5.31% higher than that for LDB.**

I. INTRODUCTION

Key-point matching can be used in many image processing systems, such as image mosaic, objects tracking and scene matching. In general, the feature matching process contains key-point detection, key-point description and descriptor matching. SIFT (Scale Invariant Feature Transform) [1] and SURF (Speeded Up Robust Features) [2] provide solutions to detect and match key-points for scaled and rotated images. MROGH (Multisupport Region Order-Based Gradient Histogram) [3], MRRID (Multisupport Region Rotation and Intensity Monotonic Invariant Descriptor) [3] and IOLD (interleaved order based local descriptor) [4] were proposed which have better matching performance and less matching time compared with SIFT. However, real value descriptors are matched by Euclidean distance with high time cost. To reduce the computational complexity, binary descriptor is introduced in BRIEF (Binary Robust Independent Elementary Features) [5] as the matching can be done using bitwise XOR in high speed. Some rotation-invariance binary descriptors were proposed with stronger robustness and they are widely used in real-time systems [6-11]. The general binary descriptor stores the result of comparing the intensity of pixels with 0 or 1, while ignoring the gradient changes information. Subsequently, the OSRI (Rotationally Invariant Binary Descriptor) [12] divides the region of interest into irregular sub-regions to generate the binary descriptor. The RFD (receptive fields descriptor) [13] uses the labeled image patches to train the descriptors, hence the performance is related to the training data. In reference [14], LDB (Local Difference Binary) is proposed to store the binary gradient information which has lower computational complexity compared with the original one. Nevertheless, the distinctiveness of LDB is not enough in sparse grid, and we will give a discussion of this in the next section.

This paper proposes a new way to generate binary descriptors based on the Haar wavelet. Haar wavelet responses were used in SURF to store the texture information within the regions of interest around the key-points. In our proposed method, the four components of Haar wavelet responses, including $\sum dx$, $\sum dy$, $\sum|dx|$ and $\sum|dy|$, are binarized and cascaded into bitstring to generate the descriptor. The experimental results show that the proposed method has better performance compared with other methods. The rest of this paper is organized as follows: Section 2 describes the binary descriptor generation algorithm, while Section 3 evaluates the performance of the proposed descriptor. The main conclusions are discussed in Section 4.

II. BINARY DESCRIPTION BASED ON HAAR WAVELET

[1] This paper is supported by the Project for the National Key Technology R&D Program under Grant No. 2011BAC12B03, the National Natural Science Foundation of China under Grant No. 81370038，the Beijing Natural Science Foundation under Grant No. 7142012, the Beijing Nova Program under Grant No. Z141101001814107, the China Postdoctoral Science Foundation under Grant No. 2014M560032, the Science and Technology Project of Beijing Municipal Education Commission under Grant No. KZ201310005004, km201410005003, the Rixin Fund of Beijing University of Technology under Grant No. 2013-RX-L04 and the Basic Research Fund of Beijing University of Technology under Grant No. 002000514312015.
□ Corresponding author: Kebin Jia; Telephone: 8610-67391019; Fax: 8610-67391019.

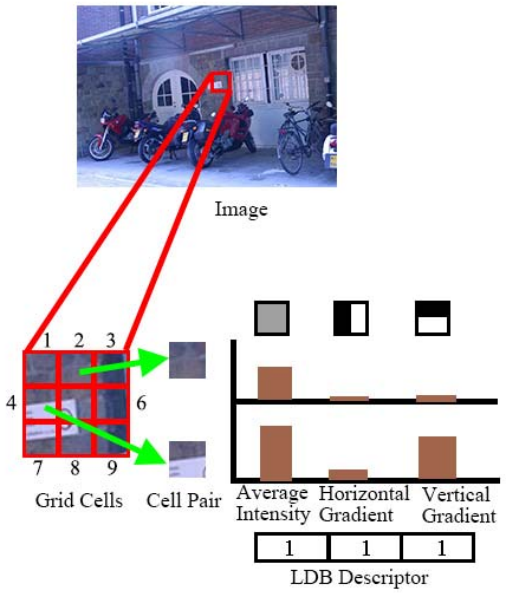


Fig. 1 The region of interest is divided into 3×3 grid cells. The feature of cells contains average intensity, horizontal gradient and vertical gradient.

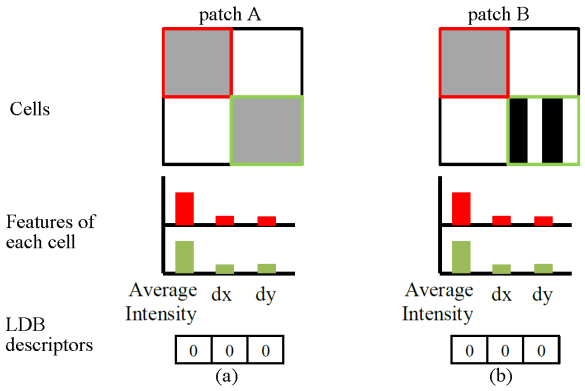


Fig. 2 Cells in image patches for which the LDB descriptors give the same description for two different patches.

Compared with real value descriptor, binary descriptors allow matching in high speed, since computing the Hamming distance between binary descriptors has lower time cost compared with computing the Euclidean distance between real value descriptors. Because of its outstanding efficiency, binary descriptor is mostly used in real-time systems.

A. LDB: Local Difference Binary Descriptor

In LDB descriptor, an image patch is firstly divided into grid cells. For example as shown in Fig 1, the patch is divided into 3×3 grid cells. The descriptor is generated with pairs of grid cells which contain two cells selected from the grid. As shown in Fig 1, No.2 and No.4 cells form a pair of cells. For each pair, the average intensity and average gradient are compared, and the relationship between two values is denoted by 0 or 1. In Fig 1, the average intensity of No.4 cell is larger than that for No.2 cell and the first bit of LDB descriptor is 1, otherwise it is 0. The average intensity is the average of all pixel intensities in one cell. To obtain more texture information in the cell, the average gradient is added into the descriptor. The average gradient contains two components, namely the horizontal gradient and vertical gradient. The horizontal gradient is the difference between the sum of pixel intensities in right half and that in left half within one cell, while the vertical gradient is the difference between the sum of pixel intensities in lower half and that in upper half, as shown in Fig 1.

Fig. 3 Haar wavelet responses in the cells

However, the LDB descriptor which is based on the average intensity and gradient is not distinctive enough for complex texture. For instance, as shown in Fig 2 that two patches (patch A and patch B) are split up into 2×2 grid cells, the cells in the green rectangle and red rectangle have the same texture in Fig 2(a) and different texture in Fig 2(b). They have the same features and the Hamming distance between the LDB descriptors of these two patches is 0, and it often leads to a high incorrect match rate. This problem can only be resolved by denser grid.

B. Haar Wavelet Binarization

To obtain more distinctive texture information, binary descriptors with Haar wavelet is proposed in this paper. Similarly to SURF descriptor, the region of interest around the key-point is segmented into $n \times n$ sub-regions (called grid cells). Haar wavelet responses in the horizontal and vertical direction are represented by dx and dy, which are the horizontal and vertical gradients of each pixel. For each cell unit, dx and dy are summed respectively as two members of texture feature, which are denoted by Σdx and Σdy. The absolute values of dx and dy are also summed to obtain the polarity of intensity changes in each cell [2], which are denoted by $\Sigma|dx|$ and $\Sigma|dy|$. As shown in Fig 3, the texture feature vector of each cell contains four dimensions and cells with different textures can be distinguished.

However, the Haar wavelet feature vector is a real value vector which is computationally expensive for the matching. Hence we extract the binary feature between cells by a binary test, which is defined as:

$$\tau(F(i),F(j)) = \begin{cases} 1 & if(F(i)>F(j)) \ \ and \ i \neq j \\ 0 & other) \end{cases} \quad (1)$$

where F(·)s are the four members of Haar wavelet feature of one cell, $\sum dx$, $\sum dy$, $\sum|dx|$ and $\sum|dy|$.

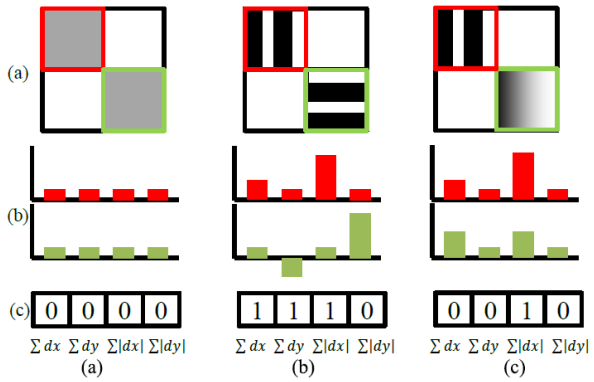


Fig. 4 Binary descriptors based on Haar wavelet

Comparing with LDB, the binary Haar wavelet feature vector is more distinctive. In Fig 4, as an example, a patch is segmented into 2×2 grid cells. The left-upper cell, denoted by a red rectangle, is compared with the right-lower cell which is denoted by a green rectangle. Fig 4(b) shows the Haar wavelet features of the two cells respectively and the binary test result is shown in Fig 4(c). It is seen that the binary descriptors can distinguish the cells with different textures.

The binary results of each pair of grid cells are cascaded into a bit-string. Besides this, we also compare the average intensities of each pair of grid cells. For a patch which is divided into $n \times n$ grid cells, the length of its bit-string is $5n(n-1)/2$. When n becomes large, the bit-string increases rapidly and requires a huge complexity for the generation and matching of descriptor. Hence the most satisfied n and pairs of grid cells that have maximum amount of information should be determined.

C. *Choice of Parameters*

The number of cells in the patch denoted by $n \times n$ influences the distinctiveness and the robustness of the descriptor. As n increases, the size of cell becomes smaller if the size of patch is fixed. Then more texture details will be descripted and the descriptor will be more distinguishable. However, dense grid may lead to lower stability as a result of more sensitive descriptors generated on smaller cells compared with larger ones. A test of different dense grid was taken and the result is shown in the Section III A. When the grid becomes denser, the computational complexity becomes higher and the length of bit-string also gets longer. Matching long descriptors is always time-consuming and it affects the performance of a matching algorithm. By analyzing the statistics on distances in Section III B, most bits in proposed binary descriptor have high correlation. The significant pairs can be identified by computing the correlation or entropy and the length of descriptor can be reduced. Further research can be done in the future to choose the most satisfactory factor n and to find out significant pairs with a training data set.

D. *Scale and Orientation Invariance*

As key-points are always detected in images with different scales, a patch should be scaled with the same factor of the key-points to achieve the scale invariance property. Also, to be orientation invariant, a patch should be rotated to normalize the direction of the patch. In our method, the direction of the centroid vector is defined as the primary direction of the patch, similarly to ORB. The centroid vector starts from the center of the patch and points towards the intensity centroid. The position of intensity centroid can be

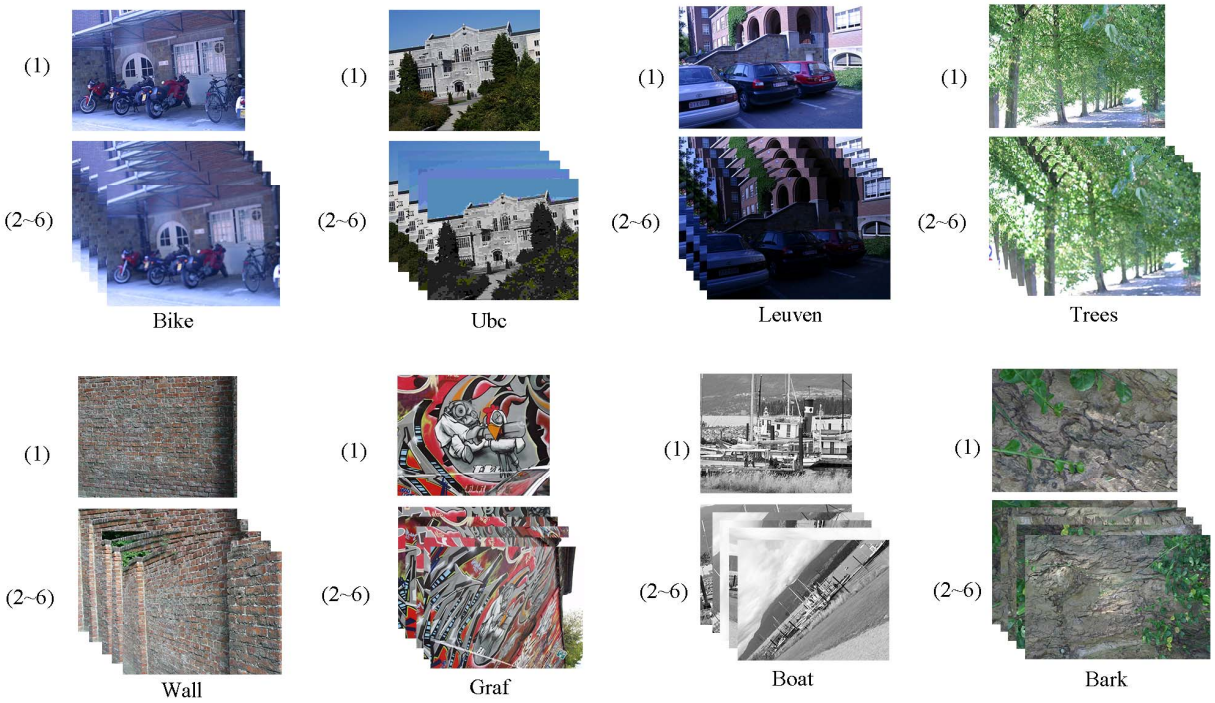


Fig. 5 Images in Image Sequences

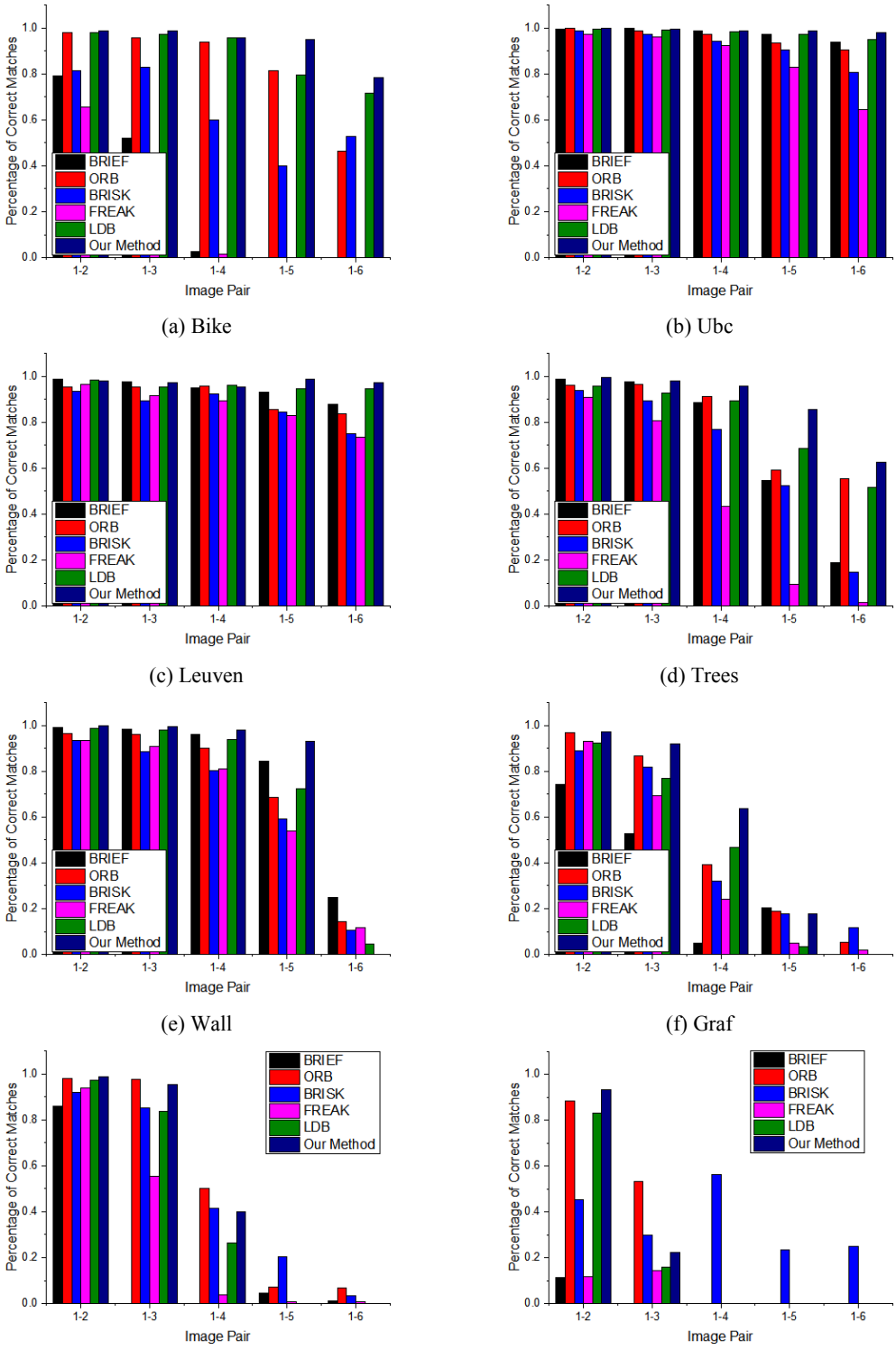
computed by the weighted sum of positions of all pixels in the patch. The position of intensity centroid is denoted as (G_x, G_y) and it can be computed as follows:

$$G_x = (\sum_{x,y} x\, I(x,y)) / (\sum_{x,y} I(x,y)) \quad (2)$$

$$G_y = (\sum_{x,y} y\, I(x,y)) / (\sum_{x,y} I(x,y)) \quad (3)$$

where $I(x,y)$ is the intensity of the pixel which is located at (x,y).

To generate orientation normalized descriptor, the patch will be rotated by the angle (denoted by θ) between the centroid vector and horizontal coordinate. All pixels in the patch are multiplied by the rotation matrix R to get the rotated patch:



(a) Bike (b) Ubc

(c) Leuven (d) Trees

(e) Wall (f) Graf

(g) Boat (h) Bark

Fig 6 Results of matching each scene

$$\theta = tan^{-1}\left(G_y / G_x \right) \quad (4)$$

$$R = \begin{pmatrix} cos\,\theta & -sin\,\theta \\ sin\,\theta & cos\,\theta \end{pmatrix} \quad (5)$$

III. EXPERIMENT

A. Percentage of Correct Matches

The database used in this study is Image Sequences [15], which contains 8 sequences. It is used to test the robustness to typical image disturbances which occur in real-world, including rotation changes, scale changes, image blur, illumination changes, viewpoint changes and compression. Each sequence contains 6 images, as shown in Fig 5, and we matched the first image with the remaining 5 images, respectively. Therefore, we have 5 image pairs, e.g. 1-2, 1-3, 1-4, 1-5 or 1-6 for each sequence. The 5 pairs are sorted in ascending order of matching difficulty. The database also provides the ground truth homography matrixes of image pairs to evaluate the matching results.

Our method makes use of the result of ORB key-point detector, and the dense of grid was 4×4. All image pairs were matched by Brute Force Algorithm which computes the distances of every descriptor pairs between two images. The corresponding point pairs between images generated by the known homography matrix were regarded as the ground truth. Concretely, for each point in image 1, the known homography matrixes were used to obtain the corresponding points in the remaining 5 images. If the matched point pair is similar with the ground truth, it is considered as a correct match. Then, the number of correct match n_{good} and the total number of match N were calculated. To quantitatively evaluate the robustness of a descriptor, the percentage of correct matches (PCM) can be used, which is defined as n_{good}/N.

To validate our proposed method, we implemented it with OpenCV 2.4.9, and compared with some other algorithms, including BRIEF [5], ORB [6], BRISK [7], FREAK [8] and LDB [14]. Fig 6 shows the percentage of correct matches of each algorithm. The horizontal axis is the serial number of image pairs and the vertical axis shows the percentage of correct matches.

Generally, our method achieves a higher percentage of correct matches as compared with other methods. The first 4 charts in Fig 6 show the robustness of images with blur, illumination change and compression. Our method has outstanding performance in image blur change, as shown in Fig 6(a) and (d). For example, the average percentage of correct matches is 93.37% in Bike, while the case in LDB is 88.46% and in ORB is 83.07%. The proposed method also has the best performance in Fig 6(b) and (c) which show high robustness of image compression and illumination change. The average percentage of correct matches is 1.15% higher than that for LDB and 2.94% higher than that for ORB in compression sequence, while it is 1.54% higher than that for LDB and 6.29% higher than that for ORB in illumination change sequence. Moreover, the proposed method has outstanding performance when images have large view point change, as shown in Fig 6(e) and (f). The average percentage of correct matches is 4.84%~4.99% higher than that for ORB.

However, note in Fig 6(g) and (h), that most binary descriptors have low robustness for scale and rotation changes.

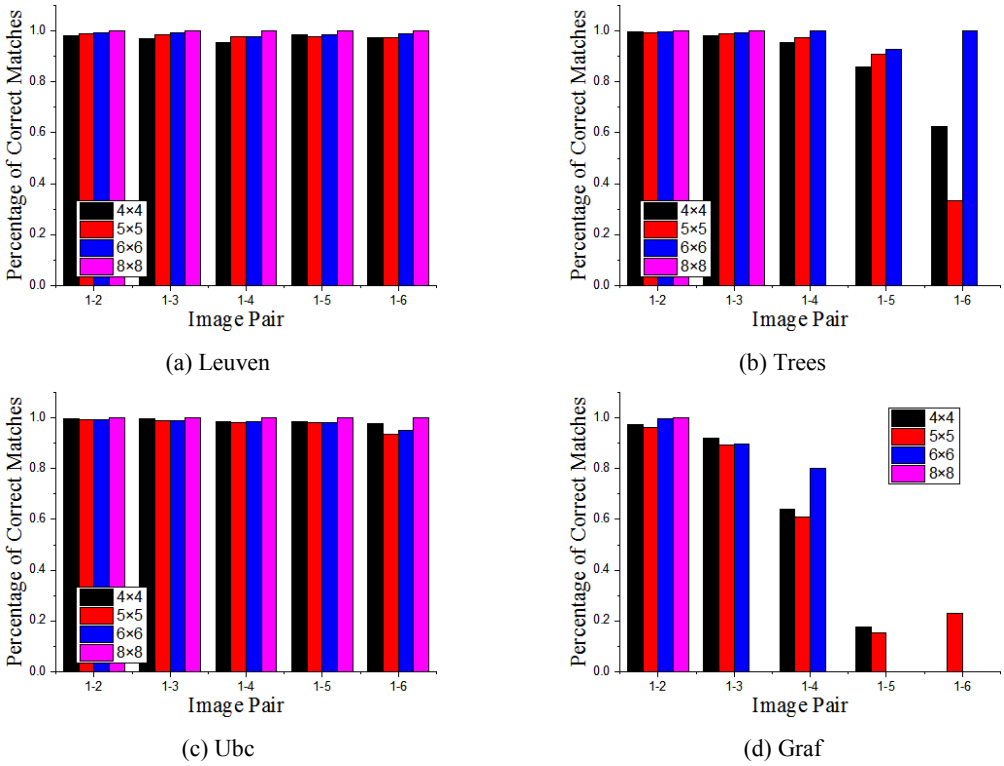


Fig. 7 Percentage of correct matches of different density grids

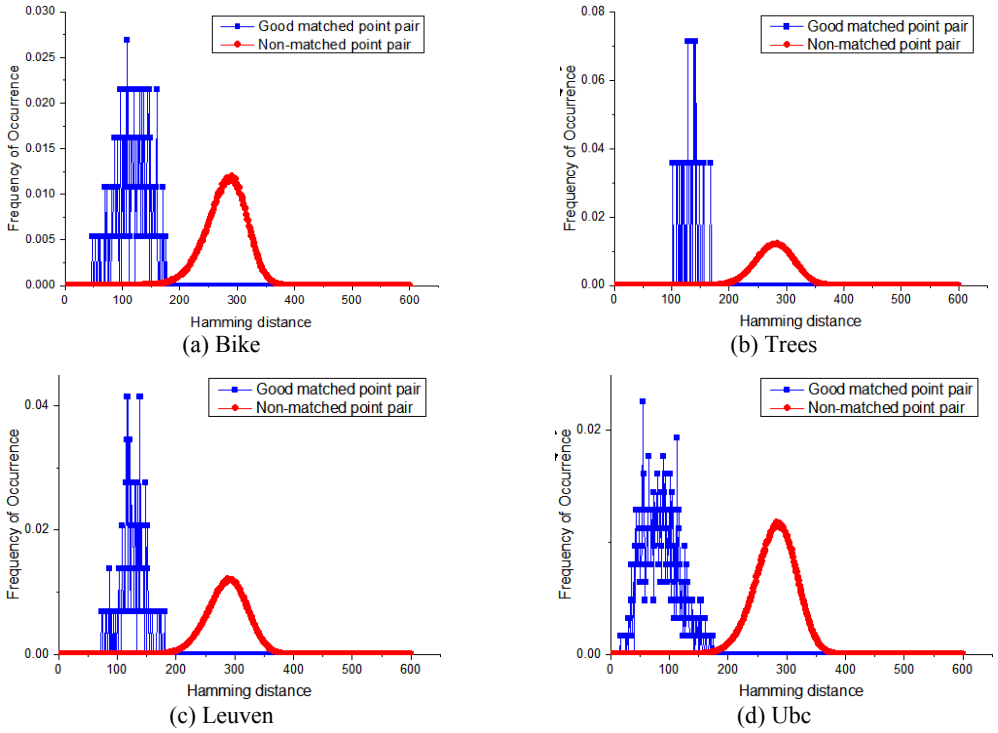


Fig. 8 Distribution of Hamming distance

This may due to the limitation of scale factor in image pyramid. The scale factors between test images pairs, such as 1-5 or 1-6, are often more than 2, but the largest scale factors between two layers in image pyramid is 1.73. This problem can be resolved by adding layers or enlarging the scale factor of image pyramid.

In order to evaluate the influence of grid density, we tested our method with different density grids, including 4×4, 5×5, 6×6 and 8×8. Fig 7 shows the percentage of correct matches with different densities of girds. The horizontal axis is the serial numbers of image pairs and the vertical axis is the percentage of correct matches. For Leuven and Trees, denser grid has better performance, as shown in Fig 7(a) and (b). On the other hand, denser grid makes descriptor more sensitive to image compression and viewpoint change, as shown in Fig 7(c) and (d).

B. Distance Distribution

In this section, an analysis of the distribution of Hamming distances of the proposed descriptors between image pairs is given. Fig 8 shows the normalized distance distribution of image pairs 1-5 in 4 sequences, Bike, Trees, Leuven and Ubc. The horizontal axis is the Hamming distance, while the vertical axis is the frequency of each distance occurrence. The distribution of matched point pairs (in blue) and non-matched point pairs (in red) are separated around 180. This value can be regarded as the threshold to distinguish good matched point pairs and non-matched point pairs. On the other hand, the distances, including matched point pairs and non-matched point pairs, vary between 0 and 350, while the length of descriptor is 600bits (the patches were split up into 4×4). This means that most bits in descriptors have high correlation with each other and they can be discarded. Hence significant bits with high information entropy can be filtered by training to accelerate the speed of descriptor generation and matching.

TABLE I
AVERAGE OF TIME COST

	Matching Pairs	Total time of Matching (ms)	Average Matching Time of Each Pair (μs)
SIFT	34592589	41094	1.185
SURF	18749432	11675	0.6237
BRIEF	10049196	858	0.0854
Proposed Method	878667	125	0.1423

C. Time Costs

To evaluate the time cost, we computed the time of each descriptor matching for SIFT, SURF and our method. SIFT and SURF were implemented with OpenCV2.3.1. The experiments were carried out on an Intel Core Duo CPU E7500 system at 2.93GHz clock rate with 8G RAM.

As shown in Table I, compared with SIFT and SURF, the matching procedure of the binary descriptor based on Haar Wavelet reduces the computation time greatly. Both SIFT and SURF have a good performance in image scale change and rotation change but with high time cost in key-point matching, so they can hardly be used in real-time system. For each descriptor pair, the time cost of our method is 87.99% less than SIFT and 77.18% less than SURF. However, because of

the non-optimized description, the length of descriptor in our method is large with some redundant information. The time cost of BRIEF is less than our method, while the length of BRIEF descriptor is 256bits. So if the descriptors can be shorten to 128bits or 256bits, our method will have even better performance.

IV. CONCLUSION

This paper proposes a new kind of binary descriptor based on the Haar wavelet. Comparing with other binary descriptors, binary Haar wavelet response saves more texture information. The proposed method is scale and orientation invariant and can achieve higher performance as compared with FREAK and LDB, especially for images with blur and viewpoint changes. The optimization of training and sampling is a good research direction in the future.

REFERENCES

[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints". *International journal of computer vision*, vol. 60, pp. 91-110, November 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features". *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, June 2008.

[3] B. Fan, F. Wu, and Z. Hu, "Rotationally Invariant Descriptors Using Intensity Order Pooling". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, pp. 2031-2045, 2012.

[4] S. R. Dubey, S. K. Singh, and R. K. Singh, "Rotation and Illumination Invariant Interleaved Intensity Order-Based Local Descriptor." *Image Processing, IEEE Transactions on*, vol. 23, pp. 5323-5333, 2014.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features". *Computer Vision - ECCV 2010*, vol. 6314, pp. 778-792, 2010.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF". *IEEE International Conference on Computer Vision*, pp. 2564-2571, November 2011.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints." *Computer Vision (ICCV), IEEE International Conference on*, pp. 2548-2555, 2011.

[8] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510-517, 2012.

[9] Y. Wang, K. Lu, and R. Zhai, "Challenge of multi-camera tracking", *Image and Signal Processing (CISP), International Congress on*, pp. 32-37, 2014.

[10] E.D. Bello and P. Salvadeo, "An image descriptors extraction hardware-architecture inspired on Human Retina", *Programmable Logic (SPL), Southern Conference on*, pp. 1-6, 2014.

[11] H. Li, Y. Wang, T. Mei, J. Wang, and S. Li, "Interactive Multimodal Visual Search on Mobile Device", *Multimedia, IEEE Transactions on*, vol. 15, pp. 594-607, 2013.

[12] X. Xu, L. Tian, J. Feng, and J. Zhou, "OSRI: A Rotationally Invariant Binary Descriptor." *Image Processing, IEEE Transactions on*, vol.23, pp. 2983-2995, 2014.

[13] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua, "Receptive fields selection for binary feature description." *Image Processing, IEEE Transactions on*, vol.23, pp. 2583-2595, 2014.

[14] X. Yang and K.T. Cheng, "LDB: An ultra-fast feature for scalable augmented reality on mobile devices". *IEEE International Symposium on Mixed and Augmented Reality*, pp. 49-57. November 2012.

[15] Image Sequences[DB/OL], http://www.robots.ox.ac.uk/~vgg/research/affine, 2007-07-15.