Copyright ⓒ 2018 KSII

RLDB: Robust Local Difference Binary Descriptor with Integrated Learning-based Optimization

Huitao Sun[1,2*] and Muguo Li[2]
[1] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology
Dalian 116024, China
[e-mail: sht229@mail.dlut.edu.cn]
[2] State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology
Dalian 116024, China
[e-mail: lmuguo@126.com]
*Corresponding author: Huitao Sun

Received October 24, 2016; revised April 22, 2017; accepted April 24, 2018; published September 30, 2018

Abstract

Local binary descriptors are well-suited for many real-time and/or large-scale computer vision applications, while their low computational complexity is usually accompanied by the limitation of performance. In this paper, we propose a new optimization framework, RLDB (Robust-LDB), to improve a typical region-based binary descriptor LDB (local difference binary) and maintain its computational simplicity. RLDB extends the multi-feature strategy of LDB and applies a more complete region-comparing configuration. A cascade bit selection method is utilized to select the more representative patterns from massive comparison pairs and an online learning strategy further optimizes descriptor for each specific patch separately. They both incorporate LDP (linear discriminant projections) principle to jointly guarantee the robustness and distinctiveness of the features from various scales. Experimental results demonstrate that this integrated learning framework significantly enhances LDB. The improved descriptor achieves a performance comparable to floating-point descriptors on many benchmarks and retains a high computing speed similar to most binary descriptors, which better satisfies the demands of applications.

Keywords: Computer vision, local feature, binary descriptor, linear discriminant projections, image matching

This research was financially supported by the National Natural Science Foundation of China (Grant no. 61202253).

http://doi.org/10.3837/tiis.2018.09.017 **ISSN : 1976-7277**

1. Introduction

Local feature descriptors are crucial in numerous computer vision tasks, such as object recognition [1, 2], 3D reconstruction [3], image retrieval [4, 5], and panorama stitching [6]. These applications also promote the development of a plethora of descriptors [7-11]. However, to satisfy the increasing demands of various applying conditions, especially real-time processing and/or running on low-power devices, it is always a pressing yet challenging problem to creat a descriptor with high performance and low computational complexity.

Floating-point descriptors [7-9] usually provide higher performance, thus some approaches (e.g. PCA-sift [12], LDAHash [13], and LDP [14]) simplify the expressions of them to speed up matching with smaller descriptors. This not only enables low dimensional descriptors to be used, but also further enhances the original descriptors by selecting the features with better determinability in subspaces.

Binary descriptors [10, 11, 15, 16] have the simpler representations, and their performance is still closely related to the constructing complexity in general. Recent studies [17-19] have shown that generating binary strings by simply operating intensities of single-pixels leads to limited robustness and discriminative capabilities, though some such descriptors show considerable improvements [11, 16, 20]. Region-based descriptors [17-19, 21, 22] usually exploit flexible region selection, multiple features and advanced optimization methods, but these often increase computational costs [22], and even result in an extraction time similar to real-value descriptors.

LDB (local difference binary) computes multiple feature differences between grid cells within a patch, which is a typical representative of region-based descriptor with low computational complexity. However, to remain a better robustness, LDB contains no finer-level sampling regions. Such deficient pooling configuration limits its overall performance. This is unable to be improved by only adjusting the gridding choices, since a complete gridding strategy will generate much more binary tests and the features from finer-level grids cannot guarantee the robustness (see **Fig. 1**).

The objective of this work is to boost the performance of such region-based binary descriptor and maintain its computational simplicity. Considering the robustness of binary features from various scales, we introduce an integrated learning-based approach to optimize LDB descriptor. Inspired by the dimensionality reduction methods of floating-point descriptors, this approach leverages the idea of LDP (linear discriminant projections) [14] to select the more representative and robust binary features. Hence our main contributions are summarized as follows:

1) We propose a new optimization framework RLDB (Robust-LDB). This framework ensures the robustness and distinctiveness of binary features from various scales, which makes it possible to optimize multi-scale LDB features effectively. We demonstrate that coarse-level features and fine-level features are both indispensable for description. The pooling strategy of RLDB contains a more complete gridding configuration, which enables a patch to be expressed comprehensively.

2) We incorporate the LDP principle into the integrated learning-based framework to directly optimize the binary features. An offline learning algorithm is developed to select the more discriminative features (binary tests) efficiently, while an online optimization procedure is utilized to mask unstable information for each descriptor separately. These two learning strategies jointly increase the descriptive capabilities of descriptors.

3) We perform experiments on various public datasets to validate the proposed optimization framework. The results show that this framework can significantly improve the discriminative capability and the robustness of LDB. Moreover, the improved descriptor offers a fast extraction speed at the same level as most binary descriptors, and achieves higher precision and detection rate in recognition application.

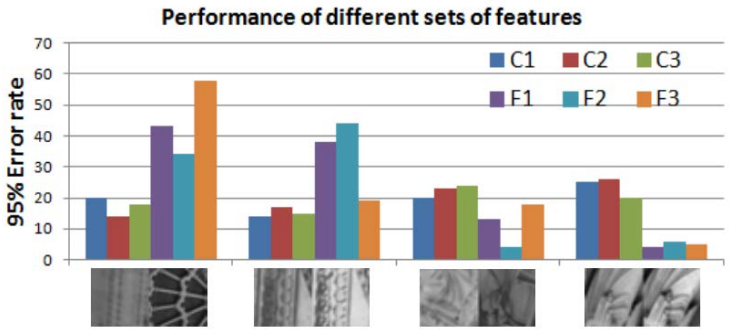


Fig. 1. 95% error rates [20, 22, 23] of different sets of binary tests for different patch pairs. Binary tests in sets C1, C2 and C3 are from coarse-level grids, and binary tests in sets F1, F2 and F3 are from fine-level grids. The performance of these sets of tests, especially the sets from fine-level grids, varies broadly with different types of patch pairs. Features from coarse-level grids respond better to relatively large-scale patterns, while fine-level features are more effective in distinguishing small-scale patterns.

2. Related Work

SIFT is generally considered as the most successful and widely-used local feature descriptor in the last decade. It computes local gradient histograms of multiple orientations around the keypoint to construct a 128D real-value vector for matching. SURF [8] is a very popular alternative to SIFT. Its efficiency was improved to some degree by utilizing Haar wavelet responses and integral images to approximate and accelerate the gradient computations. Recently, learning-based descriptors [22, 24-27] were proposed to fill the gaps in hand-crafted design of descriptors. Multiple aspects of feature description were optimized by learning, such as pooling pattern and feature selection. However, most floating-point descriptors involve intensive calculations, which are time-consuming and/or require many hardware resources.

Binary descriptors have been developed in recent years to meet the demands of real-time and low-power-device applications. BRIEF [10] generates binary strings for matching by comparing the intensities of arbitrarily selected pair-wise pixels. To improve the performance of such binary strings, ORB [11], BRISK [16] and FREAK [15] respectively selected the more effective binary tests with different sampling strategies, and realized rotation and scale invariance of binary descriptors. Recently, ORB was further optimized in BOLD [20] and [28] by online learning. Even so, it was shown in [17, 19] that the descriptors only based on raw intensities of several single-points in a patch are not distinctive and robust enough. A number of algorithms [17-19, 22] use multiple features from sampling regions instead of simple pixel intensities. LDB [17] constructs a binary string by comparing the intensities and gradients of different grid cells within a patch, which achieves good robustness while maintaining high extracting efficiency. However, there is still much room for performance improvement of LDB due to its limited gridding choices.

Another category of approaches [12-14, 23, 29] transforms the descriptor vectors into low-dimensional representations to simplify matching process, and some of them learn optimal transformation parameters from training examples. PCA-sift [12] reduced the dimensionality for SIFT descriptor with PCA technique. Cai et al. presented Linear Discriminant Projections (LDP) [14], which learns the discriminative dimensions during the mapping process by maximizing the ratio of the sum of distances between differently-labeled training examples to the sum of distances between same-labeled examples. Strecha et al. [13] mapped real-value descriptor vectors into Hamming space with a LDA-like approach and Hashing technique. Our work is closely related to [17], but differs from it by applying a new learning framework and employing the idea of LDP to optimize the binary descriptor.

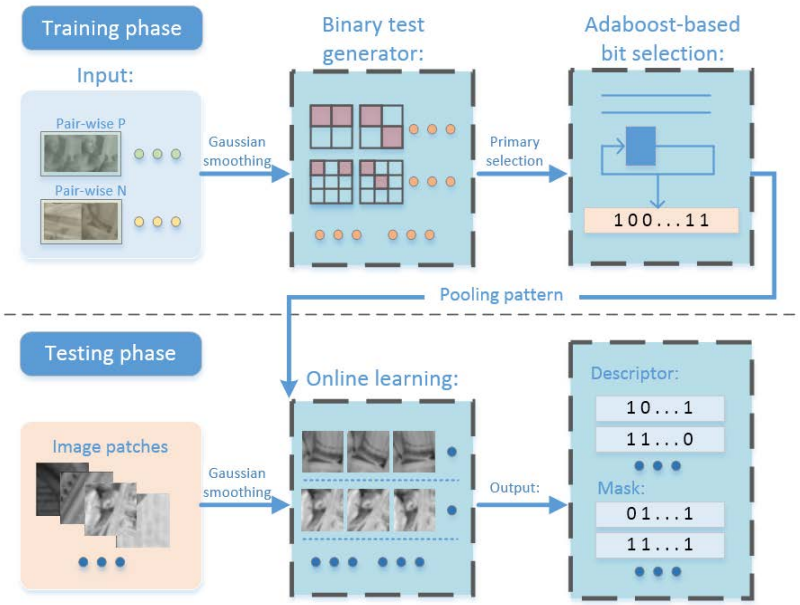


Fig. 2. Overview of the proposed framework RLDB

3. RLDB

The pipeline of the proposed RLDB is described in **Fig. 2**. RLDB utilizes a multi-stage bit selection algorithm to learn the more representative patterns by training, and then further optimizes descriptor for each patch by an online learning strategy in testing phase. Since LDB feature keeps resilient to most of the photometric changes by computing differences between two sub-regions, we also generate a binary string first based on the tests between every possible pair of grid cells. More specifically, for a certain grid scale, an image patch p is divided into $s \times s$ equal-sized grid cells, and each bit (b_t) of the binary string represents a test result on a pair of such grid cells (C_{t1} and C_{t2}):

$$b_t(p; C_{t1}, C_{t2}) = \begin{cases} 1, & \text{if } Func(C_{t1}) < Func(C_{t2}) \text{ and } C_{t1} \neq C_{t2} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $1 \le t \le N$, and N is the number of the bits that compose the binary string (N -dimensional Hamming space $\boldsymbol{H}^N$). In formula (1), $Func(C)$ denotes the function for extracting intensity and gradient information from a grid cell C . All the N test-bits of the binary string can be represented as a set B with $b_t \in B$. A more complete gridding configuration (Section 3.1) will yield more binary tests, and we will select the more meaningful and stable ones from B by the LDP principle. Given a set of labeled example pairs for training, methods like [14, 27] seek to learn a direction vector $\boldsymbol{u}$ to project high dimensional vectors into the subspace with better discriminability. During this process, LDP is designed to maximize the distances between differently-labeled examples (D) and minimize the distances between same-labeled examples (S). In our proposed framework, we exploit this theory just to select a set of binary tests from plenty of candidates ($\boldsymbol{H}^N \to \boldsymbol{H}^M, M < N$). Hence, $\boldsymbol{u}$ becomes a vector that consists of 1s and 0s, where 1s indicate which dimensions (tests) are selected. $\boldsymbol{u}$ can be formulated as follows

$$\boldsymbol{u}_{\boldsymbol{BLDP}} = \arg\max_{\boldsymbol{u}} \frac{\sum_{(p_a,p_b)\in D} \left\| \boldsymbol{u}^T \boldsymbol{b}(p_a) - \boldsymbol{u}^T \boldsymbol{b}(p_b) \right\|^2}{\sum_{(p_a,p_b)\in S} \left\| \boldsymbol{u}^T \boldsymbol{b}(p_a) - \boldsymbol{u}^T \boldsymbol{b}(p_b) \right\|^2} = \arg\max_{\boldsymbol{u}} \frac{\boldsymbol{u}^T C_D \boldsymbol{u}}{\boldsymbol{u}^T C_S \boldsymbol{u}} \tag{2}$$

where $\boldsymbol{b}$ is a vector that is composed of 0s and 1s as well, and $\boldsymbol{b}(p)$ represents the results of all the binary tests from set B on patch p . C_D and C_S respectively represent the inter-class and intra-class covariance matrices. With this formula, the Local Difference Binary description will be optimized by selecting a discriminative subset of bits from B, which we will detail in Section 3.2 and Section 3.3.

3.1 Complete gridding strategy

In multiple-gridding strategy, features from different grid scales (sizes) influence the robustness and distinctiveness of a descriptor (see **Fig. 1**). However, to ensure a better robustness of features, LDB contains no finer level grids, such that each patch is partitioned into a small number (e.g. from 2×2 to 5×5) of relatively large regions. This leads to a stable descriptor but limits its distinctiveness. By contrast, since we employ an integrated learning method for selecting discriminative features (Section 3.2) and removing unstable features (Section 3.3), the choice of grid sizes in our framework can be more comprehensive, as shown in **Fig. 3**. Specifically, for a patch p of radius r , we obtain $r-1$ ($2 \times 2, 3 \times 3, \ldots r \times r$) gridding choices, and all possible grid cells from these choices constitute a complete set of sampling regions from multiple scales. With this set, $(-2r - 5r^2 + 5r^4 + 2r^5)/20$ grid-cell pairs for comparison will be generated. For an image patch of size 64×64, in practice, we generate ~440K comparison pairs under several reasonable constraints (e.g. the widths of grid cells are required to be integers). These comparison pairs can represent the patch more completely, thus the descriptor will be enhanced comprehensively.

The gridding strategy produces steerable square regions. Features from such regions are easier to compute by using integral images, compared with the ring-based region sampling method in [19]. The integral images can be computed for a whole image as well as for each patch separately, which avoids a linear increase in calculations as the detected features get denser. Besides, with the square region sampling, we can conveniently check the unstable representations in each descriptor (Section 3.3).

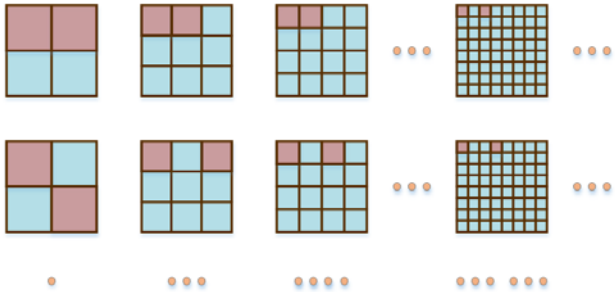


Fig. 3. Grid-cell pairs for comparison are generated from a more complete gridding strategy.

3.2 Offline bit selection

The complete sampling strategy can extract information from various scales, but it also yields a huge number of binary tests. Moreover, some of these tests are redundant or less representative. Hence, we express the test results of all the region pairs as a long bit string and select a set of discriminatory bits with formula (2) to construct a compact representation. From formula (2), the selected bits are expected to have two attributes: 1) minimizing the distance between matching patches while maximizing the distance between non-matching patches, 2) low correlations between these bits.

Boosting methods have proved to be highly effective in selecting a set of features for classifying different image patches [17, 19, 22, 30]. Yang and Cheng applied the Adaboost-based method to the bit selection problem in [17] by assigning uniform feature weights and introducing accumulated error. Intuitively, their algorithm can select the bits that have the first attribute mentioned above. However, their algorithm lacks explicit correlation constraint and is inefficient for our large scale bit selection. Therefore, we propose Algorithm 1, a novel multi-stage bit selection algorithm with constraints, to address these limitations.

Algorithm 1 Multi-stage bit selection with constraints

Input: Training data $T = \{(X_i, Y_i) \mid 1 \le i \le |T|\}$, where $X_i = (p_{ia}, p_{ib})$ is a pair of patches, and $Y_i = 1$ if X_i is a match; otherwise $Y_i = 0$. A complete set of candidate bits: $U_B = \{b_1, b_2, \cdots b_N\}$.

Output: A set of selected bits: $\hat{U} = \{\hat{b}_1, \hat{b}_2, \cdots \hat{b}_n\}$, $n = |\hat{U}|$.

1. Compute N-bit strings for all patches in T.
2. Compute a test error $\sum_{i=1}^{|T|} |Y_i - \overline{Y_i}|$ for each candidate bit, where $\overline{Y_i}$ represents the test result on patch pair X_i.
3. Select $N/2$ bits whose errors are smaller.
4. Compute an evaluation coefficient $\xi(b_j, T)$, $1 \le j \le N/2$ for each selected bit.
5. Sort these bits by their $\xi(b_j, T)$ s in descending order and choose the first $N/10$ bits to compose a candidate bit table.
6. Adaboost-based bit selection with constraints:
7. **Initialize:** assign uniform weight $d_i = 1/|T_s|$ to all training data from subset T_s, and set $\hat{U} = \phi$.
8. **for** $t = 1$ to n

   Find a bit b_t in the candidate table, which gives a minimum accumulated error:

$b_t = \arg\min \varepsilon_{acc}(t)$, $\varepsilon_{acc}(t) = \varepsilon_{acc}(t-1) + \varepsilon_t$, $\varepsilon_{acc}(0) = 0$, $\varepsilon_t = \sum_{i=1}^{|T_s|} d_{t,i} \left| Y_i - \overline{Y_i} \right|$.

Compute the correlation: $corr(b_t, \hat{b}_k)$, $\hat{b}_k \in \hat{U}$.

if $corr(b_t, \hat{b}_k) < t_c$, $\forall \hat{b}_k \in \hat{U}$, **then** add b_t into $\hat{U}$;

else if find a b_t' in the table s.t. $\frac{\varepsilon_{acc}(b_t')}{\varepsilon_{acc}(b_t)} < t_e$, $corr(b_t', \hat{b}_k) < t_c$, **then** add b_t' into $\hat{U}$;

otherwise add b_t into $\hat{U}$.

if $\varepsilon_t < 0.5$, **then** update weights of training data;

otherwise switch to a new training subset T_s' and reset $d_i = 1/\left|T_s'\right|$.

9. **return** $\hat{U}$.

This algorithm is inspired by LDP. First, an appropriate discrimination measure is proposed to implement an efficient selection procedure. For a labeled training dataset T, C_D and C_S in formula (2) are computed as follows

$$C_D = \sum_{X_i \in T_0} (\boldsymbol{b}(p_{ia}) - \boldsymbol{b}(p_{ib}))(\boldsymbol{b}(p_{ia}) - \boldsymbol{b}(p_{ib}))^T \tag{3}$$

$$C_S = \sum_{X_i \in T_1} (\boldsymbol{b}(p_{ia}) - \boldsymbol{b}(p_{ib}))(\boldsymbol{b}(p_{ia}) - \boldsymbol{b}(p_{ib}))^T \tag{4}$$

where $T_0 = \{T \mid Y_i = 0\}$ and $T_1 = \{T \mid Y_i = 1\}$. Vector $\boldsymbol{u}$ is determined by the eigenvector corresponding to the largest generalized eigenvalue of matrix $C_S^{-1} C_D$. However, computing the eigenvector requires a mass of high-dimensional matrix operations due to the large number of candidate bits and training examples. Fortunately, for this large scale bit selection problem, we observe that the diagonal elements of C_D and C_S in formula (3) and formula (4) are far greater than off-diagonal elements. (The variances of diagonal elements are also far greater than those of off-diagonal elements). Thus, we can approximately compute a coefficient ξ for each bit using formula (5) to evaluate the discrimination.

$$\xi(b_j, T) = \frac{\sum_{X_i \in T_0} [b_j(p_{ia}) - b_j(p_{ib})]^2}{\sum_{X_i \in T_1} [b_j(p_{ia}) - b_j(p_{ib})]^2} \tag{5}$$

where b_j ($b_j \in U$) represents an arbitrary test bit, and $b_j(p)$ is its test result on a patch p. Obviously, the bits whose ξ s are larger should be picked out. They have better discriminative power that increases inter-class distances and decreases intra-class distances. Second, adding the correlation constraint enhances the distinctiveness for the whole descriptor, which increases the inter-class distances integrally. The correlation between two tests (b_t and $\hat{b}_k$) can be calculated as formula (6).

$$corr(b_t, \hat{b}_k) = \frac{2|T| - \sum_{i=1}^{|T|} b_t(p_{ia}) \oplus \hat{b}_k(p_{ia}) + b_t(p_{ib}) \oplus \hat{b}_k(p_{ib})}{2|T|} \tag{6}$$

where $\oplus$ denotes XOR operation. For the Adaboost-based method, if we directly filter out some of the selected bits by adding a simple constraint, the rest of these bits (weak-classifiers) will constitute a defective strong-classifier, which will spoil the effect of the whole algorithm. Therefore, thresholds t_c and t_e are introduced together to ensure the selected bits simultaneously satisfy the correlation constraint and accumulated error constraint, and they are set empirically (e.g. $t_c = 0.45$ and $t_e = 1.2$).

3.3 Online optimization

The offline bit selection algorithm is a global optimization strategy, which learns the salient binary tests over the whole training dataset. However, such strategy cannot ensure that the descriptor is locally optimized as well. As shown in **Fig. 1**, different sets of tests perform very differently on different patch pairs. If we optimize the selected binary tests again for each patch independently, the resulting descriptor will perform better.

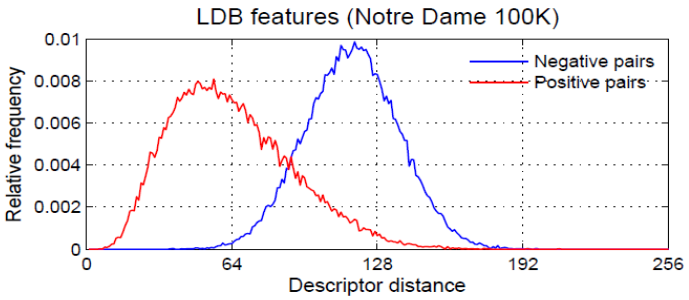
Compared with global optimization, the local optimization strategy learns specific description pattern on each individual class of samples. Thus, according to formula (2), the local optimization process is expected to decrease the distances between patches from the same class S_i and to ascertain the more stable bits in set $\hat{U}_B$ for S_i. For an efficient learning process, the number of patches from S_i is generally limited. Since $\boldsymbol{u}$ and $\boldsymbol{b}$ in formula (2) are both composed of 0s and 1s, we can obtain:

$$\boldsymbol{u}_i = \neg(|\boldsymbol{b}_i(p_i) - \boldsymbol{b}_i(p_{ia})| \vee |\boldsymbol{b}_i(p_i) - \boldsymbol{b}_i(p_{ib})| \vee \cdots) \tag{7}$$

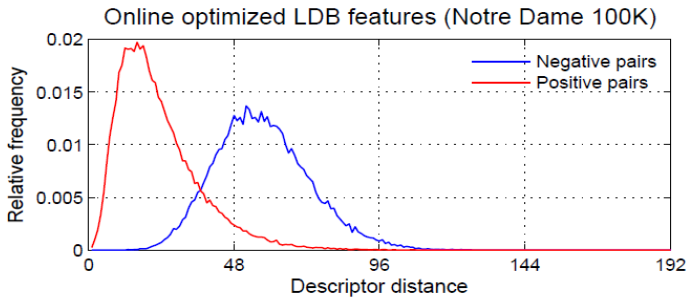
where $\neg$ and $\vee$ denote logic NOT and OR operations, and $\boldsymbol{b}_i(p_i)$ represents the binary string that describes patch p_i. $\{p_{ia}, p_{ib}, \cdots\}$ is a set of patches from the same class as p_i. $\boldsymbol{u}_i$ in formula (7) can be seen as a mask $\boldsymbol{m}_i$ for $\boldsymbol{b}_i(p_i)$, and 1s in $\boldsymbol{u}_i$ indicate which bits in $\boldsymbol{b}_i(p_i)$ are stable. From another perspective, choosing the more stable binary tests for each class can be considered as changing the binary weights of the selected features. Since we optimize the descriptor for each specific patch, $\boldsymbol{u}_i$ has to be obtained by online learning [14, 20, 28, 31], which performs during descriptor extraction. In this case, only one instance p_i in class S_i is available, while additional examples $\{p_{ia}, p_{ib}, \cdots\}$ for learning need to be synthetically generated.

Affine projections are effective in producing such simulated data. By following [14, 20, 32, 33], we compute affine transformations for patch p_i to approximate various geometric changes and generate set $\{p_{ia}, p_{ib}, \cdots\}$ for learning. However, for real-time applications or low-power devices, it is time-consuming to handle the online process. Fortunately, with our sampling strategy, the affine transformations can be applied directly to sampling locations $\{L(C_{t1}), L(C_{t2}) | 1 \le t \le n\}$ to avoid warping the whole patch, where $L(C)$ denotes the central location of grid cell C. Furthermore, given that globally optimized tests have been fixed after

offline learning, we can construct a lookup table for the transformed locations $\{L_{Affine}(C_{t1}), L_{Affine}(C_{t2}) | 1 \le t \le n\}$ of these tests to reduce online calculations. Thus, more information is acquired efficiently for each feature to detemine the stability of the feature. Specifically, after trying various parameters of affine transformations, we also choose the transformations with rotations of $10°$ to $20°$, which usually yield the better results [14].



(a)



(b)

Fig. 4. Changes of positive pair and negative pair distance distribution. (a) Distance distribution of LDB descriptors; (b) distance distribution of online optimized LDB descriptors.

To analyze the effect of the online optimization, we observe the changes of distance distribution of pair-wise patches. **Fig. 4** shows the distributions of original LDB descriptors (a) and online optimized LDB descriptors (b). The image dataset (Notre Dame 100K, see section 4.1 for details) for test contains both matching pairs (Positive pairs) and non-matching pairs (Negative pairs), and the overlapping area between their distributions is reduced from 13.26% to 10.95%, which means that the indistinguishable patch pairs become fewer.

In practice, the original descriptor $\boldsymbol{b}$ consists of two parts $\boldsymbol{b}_c$ and $\boldsymbol{b}_f$, which separately represent the binary tests from coarse-level grids and the binary tests from fine-level grids. Since the coarse-level features themselves are relatively robust, the mask $\boldsymbol{m}$ of $\boldsymbol{b}$ has the same dimensionality n_f as $\boldsymbol{b}_f$. We only compare the more stable parts of two descriptors, and therefore Hamming distance with masks is used as a metric of matching:

$$Hm(\boldsymbol{b}_i, \boldsymbol{b}_j) = \frac{n_f}{2D_i} \boldsymbol{m}_i \wedge \boldsymbol{b}_{if} \oplus \boldsymbol{b}_{jf} + \frac{n_f}{2D_j} \boldsymbol{m}_j \wedge \boldsymbol{b}_{if} \oplus \boldsymbol{b}_{jf} + \boldsymbol{b}_{ic} \oplus \boldsymbol{b}_{jc} \tag{8}$$

where $\wedge$ denotes logic AND operation. In formula (8), $\boldsymbol{m}_i$ is the mask for original binary descriptor $\boldsymbol{b}_i$. 1s in $\boldsymbol{m}_i$ indicate which bits in $\boldsymbol{b}_i$ are valid for patch p_i, and D_i is the number of 1s.

4. Experiments

In this section, we present the experimental results to evaluate the proposed RLDB descriptor and compare its performance with state-of-the-art descriptors on public datasets. For LDB [17], Binboost [22], BRIEF [10] and BOLD [20], the available implementations from their authors were used in the experiments. For SURF [8] and ORB [11], we used the implementations from OpenCV. For a fairer comparison, all the binary descriptors for test are 256 bits.

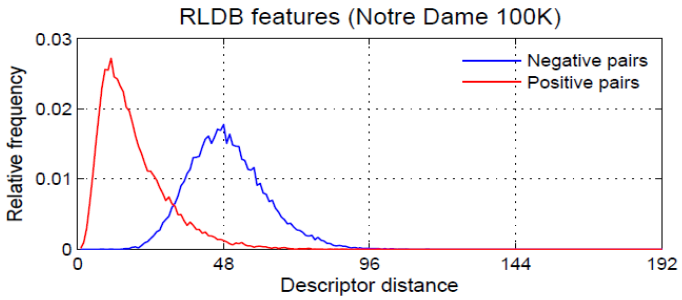


Fig. 5. Positive-pair and negative-pair distance distribution of RLDB descriptors

For RLDB, we randomly chose 50K matching pairs and 200K non-matching pairs from Brown patch datasets [27, 34, 35] to perform the supervised training in Section 3.2, and set parameters t_c and t_e to 0.45 and 1.2. Since the complete gridding strategy is applied, we first observe the distance distribution of RLDB descriptors on patch-pair dataset. **Fig. 5** shows the distributions of positive pairs and negative pairs from dataset Notre Dame, and the overlapping area between them is further reduced to 9.33%. This illustrates that features from fine-level grids and from coarse-level grids are both very important for description, and our integrated learning framework can effectively choose the more robust information from the features.

Fig. 6. Examples from patch datesets

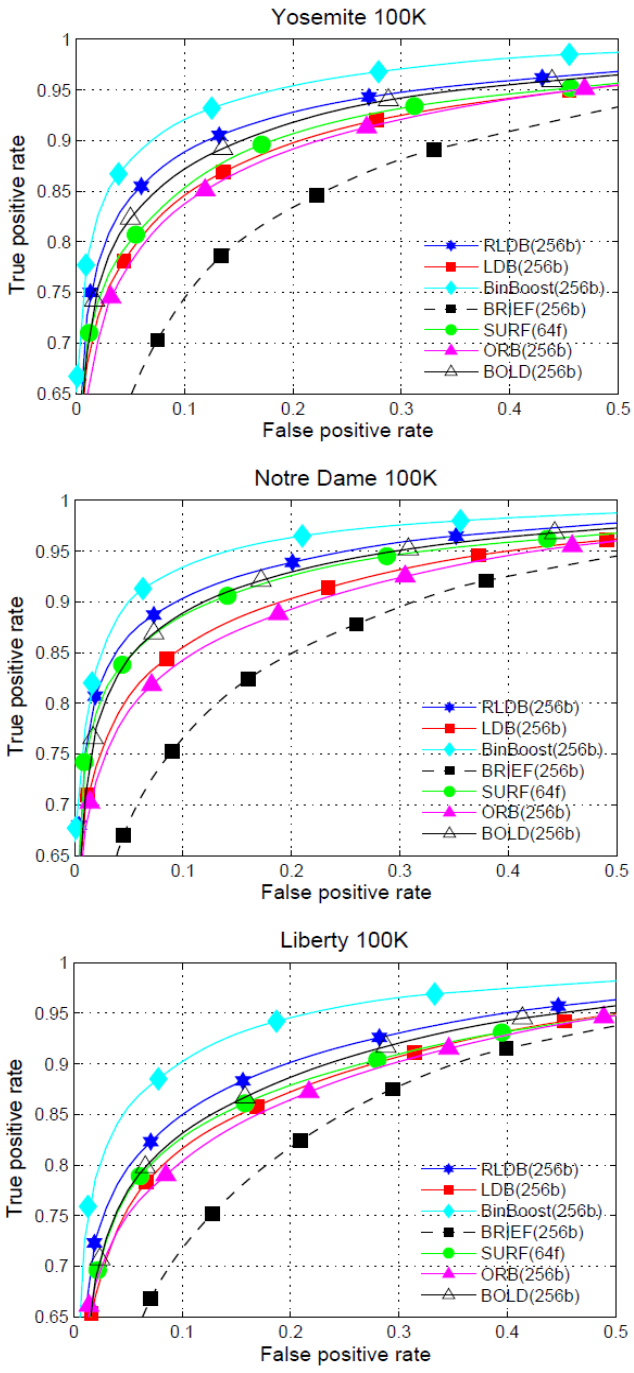


Fig. 7. Performance of different descriptors on patch datasets

4.1 Patch dataset

We first perform an evaluation using the benchmark from [22, 34] to investigate the discriminative capability of RLDB descriptor. The three patch datasets [34, 35] that we used in

the experiment are Yosemite, Notre Dame and Liberty. Patches in these datasets are sampled from 3D reconstruction of three real scenes. All the patches are of size 64×64 and arranged in 1024×1024 bitmap images. Examples are given in **Fig. 6**. Their centres are determined by real feature point detections, and their orientations and scales are normalized. Since these datasets also include associated files that list "matches" and "non-matches" of the patch pairs as ground truth, they are frequently employed in descriptor training and testing in researches [17-23, 26, 27]. We used 100K patch pairs of each dataset to perform our tests. The test results of different descriptors on these datasets are presented in **Fig. 7** in terms of ROC curves.

As **Fig. 7** shows, RLDB descriptor outperforms original LDB distinctly. This corresponds to the result that **Fig. 5** reflects. Particularly, the bit selection algorithm in Section 3.2 enhances the expressive power of the bit-combined descriptor. Although the descriptor dimensionality is reduced by the method in Section 3.3, the rest of the dimensions always constitute a more definite and stable subset. These two points jointly ensure the complete pooling strategy to function effectively. Furthermore, features from fine-level grids are crucial in distinguishing the details of a large quantity of patches, which is an important reason why the results of RLDB from all the three datasets are better than those of LDB.

In **Fig. 7**, we also observe that RLDB descriptor performs better than most binary descriptors except Binboost. After all, the extraction procedure of Binboost is closer to that of a real-value descriptor. However, the real-value descriptor, SURF, just performs slightly better than LDB and ORB, and doesn't show the distinct superiority over these binary descriptors in this experiment. It has to be noted, that these binary descriptors were usually optimized by learning from patch datasets, while SURF was designed without any learning methods. Therefore, effective learning strategies are significant for the performance improvements of descriptors. Another notable property of RLDB descriptor is that it reaches relatively high true-positive-rates when false-positive-rates are low in the left half of the curve. This means that in situations where high matching accuracy is required, more corresponding points can be correctly matched by using RLDB descriptor.

Fig. 8. Example images from Oxford Vgg-Affine dataset

4.2 Keypoint matching

In this section, we evaluate the matching performance of RLDB descriptor on Oxford Vgg-Affine dataset [36, 37]. This dataset consists of sequences of images affected by various factors (e.g., scale changes, illumination changes), and therefore is often used to test expressiveness of descriptors under different image variations [10-12, 16-20]. **Fig. 8** shows some example images of this dataset. In our experiment, we used six image sequences, including Bikes (image blur, 1000×700 pixels), Leuven (illumination changes, 900×600 pixels), UBC (compression artifacts, 800×640 pixels), Wall (viewpoint changes, 1000×700 pixels), Boat (rotation and scale changes, 850×680 pixels), and Bark (rotation and scale changes, 765×512 pixels). Each sequence contains a set of images, sorted in increasing order of image distortion. We detected 1000 keypoints for each of the images and matched the descriptors of the first image to those of other images. By following [17], oFAST was employed as the keypoint detector. Homographies between the images can provide the ground truth for the matching test.

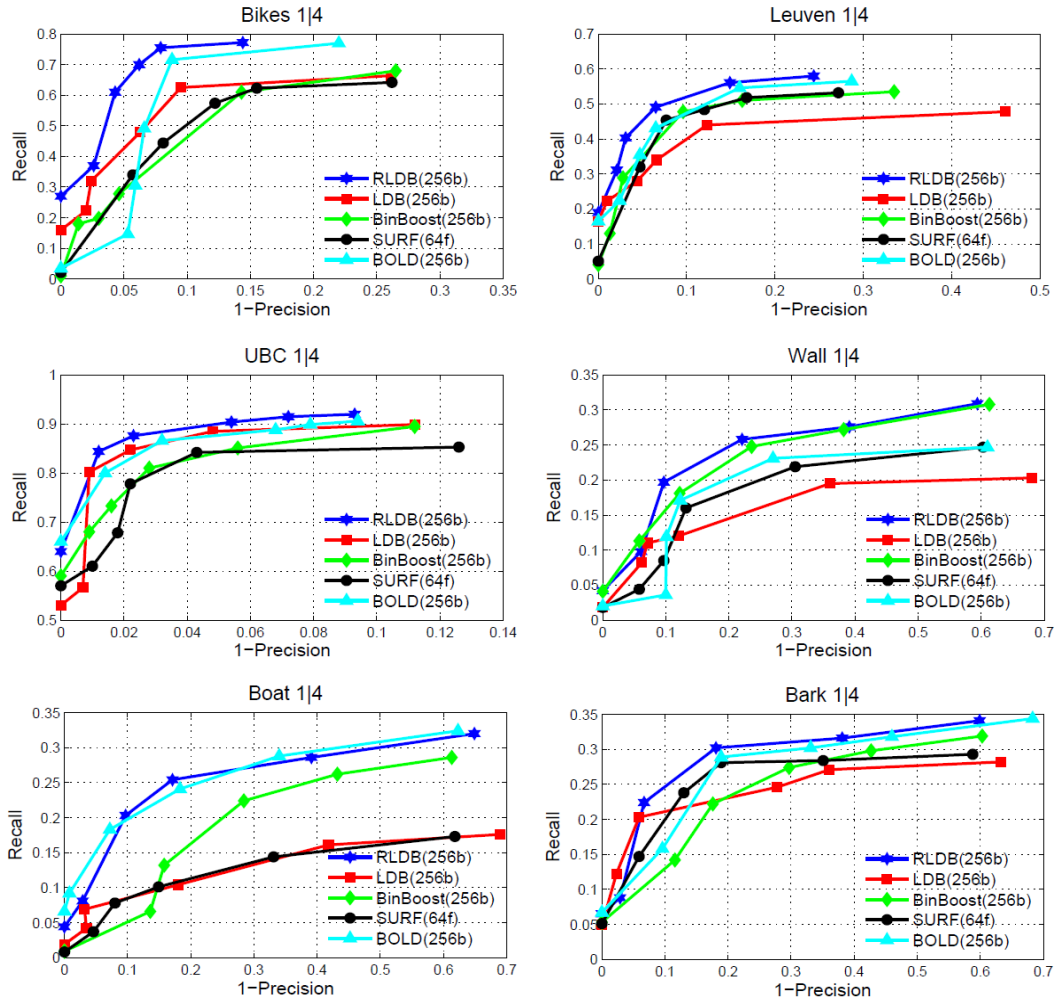


Fig. 9. Keypoint matching results of image pairs 1|4 in Vgg-Affine dataset

Fig. 9 shows the matching results of the image pairs 1|4 in terms of the Recall versus 1-Precision, which is computed based on a series of thresholds. Evidently, the proposed RLDB descriptor (blue) outperforms LDB (red) again. To compare RLDB and LDB in a different way, we specially list their recognition rates on the six sequences in **Table 1**. The results show that the recognition rates of RLDB surpass those of LDB for all image pairs. Adopting the finer grids enables the distinctive details of the images to be described, and our optimization framework offers the robustness for the expressions of these details. This point is reflected from a different perspective as well by the close results of RLDB and LDB on UBC sequence (compression artifacts). It is worth mentioning that BOLD performs slightly worse on Bikes and Wall sequences than RLDB. A potential reason is that the bit string of BOLD is generated by purely comparing intensities of single pixels, which neglects the expressions for relatively large scale patterns in images, even if these bits can maintain some robustness through online learning. Consequently, we believe that a comprehensive sampling strategy is necessary.

Surprisingly, although Binboost performs best among all the descriptors in patch dataset experiment, it seems less effective in keypoint matching and ranks behind RLDB and BOLD in several cases. This may be related to the training data and the pooling mode of Binboost. Besides, the detector should also be considered, which is an important factor for the float descriptor SURF not performing as good as reported before. Some recent studies [16, 17] also demonstrated this observation. More specifically, SURF descriptor was used with a blob detector in the original design [8], while oFAST, as a corner detector, may limit some abilities of it.

Table 1. Comparisons of LDB and RLDB descriptor on matching accuracy for the image sequences from Vgg-Affine dataset

Categories		Bikes	Leuven	UBC	Wall	Boat	Bark
1|2	LDB	79.5	70.0	95.9	50.7	67.8	68.1
	RLDB	89.8	80.6	97.0	68.9	73.3	74.0
1|3	LDB	77.0	58.2	94.0	41.0	48.4	43.7
	RLDB	86.1	68.9	95.1	60.1	61.0	57.8
1|4	LDB	67.1	49.3	89.6	21.0	16.2	28.5
	RLDB	78.9	59.1	93.2	32.2	32.3	35.2
1|5	LDB	58.7	46.4	80.3	10.1	4.2	7.9
	RLDB	76.2	55.5	89.1	16.3	11.2	14.9
1|6	LDB	47.9	41.1	65.0	2.4	2.3	2.1
	RLDB	67.8	51.7	79.7	5.2	4.6	3.0

4.3 Efficiency

To compare the computational efficiency, the above experiment in section 4.2 is extended. We perform the image matching with image sequence Bikes (size: 1000×700 pixels), and record the average time requirements of different descriptors on constructing and matching. All the computations were done on a laptop with an Inter Core-I5 processer running at 2.3 GHz.

The test results are shown in **Table 2**. Extracting a RLDB descriptor takes only $80\,\mu s$ on average, at the same level as most binary descriptors. This is because RLDB inherits the sampling strategy of LDB, and features from square regions can be computed conveniently by the assist of integral images. Admittedly, the online learning increases the computation time inevitably. However, with our approach in Section 3.3, the additional time is limited, making RLDB only $1.63\times$ slower than LDB. Moreover, the extra time of RLDB is mainly used for

eliminating unstable expressions from finer grids, which is very worthwhile for the higher quality. Among these descriptors, LDB and ORB have the faster extracting speeds, at around 50 μs /descriptor. Despite the great performance on patch dataset, Binboost takes much more time for constructing, similar to the float-point descriptors. It is noteworthy that the time data in **Table 2** is based on image matching experiment rather than simple patch matching, which we believe is closer to real applications.

Table 2. Average processing time per operation of various descriptors

Descriptor	Extraction (μs)	Matching (μs)
LDB	49	0.12
RLDB	80	0.27
Binboost	758	0.12
BOLD	78	0.35
ORB	51	0.12
SURF	676	5.20

In **Table 2**, the matching time denotes the average cost for computing distance between two descriptors. Although the masked Hamming distance increases the computations of matching, all these logic operations can be done quickly on a common processer with the avx instruction set (e.g. popcount). As a result, compared with real-value descriptor, the matching times of all the binary descriptors are still very short.

4.4 Object recognition

In this section, we test the RLDB descriptor with an object recognition task to demonstrate its effectiveness on applications. Images for this experiment are from four datasets: 1) ZuBud dataset [38], 2) Kentucky dataset [39, 40], 3) COE underwater dataset, 4) dataset generated from Flickr100K with the method in [17]. ZuBud dataset consists of 1005 images of 201 Zurich buildings. Each building/scene is represented by a group of 5 images acquired at different viewpoints. All the images in ZuBud were captured under various photometric conditions by two cameras at 640×480 resolution. Kentucky dataset contains abundant images about daily-life objects with 4 images for each object. The recognition difficulty of its subset varies broadly, depending on the selected objects and image sharpness. The sizes of the images in Kentucky dataset are 640×480. COE dataset includes 102 groups of underwater images captured in laboratory by industrial cameras. These images are generally affected by the complex imaging circumstance (e.g., refraction, scattering, and low contrast). Each group consists of 4 images of an underwater object/scene. Following [17], we also artificially capture pictures from Flickr images for our recognition test. For each randomly selected Flickr image, 4 additional synthetic pictures (including multiple transformations, e.g., rotation, scaling, and brightness changes) are generated to collectively constitute a group of the dataset. Examples from the four datasets above are shown in **Fig. 10**.

We follow the evaluation protocol in [17], and randomly select 100 groups of images from every dataset to constitute our database (1800 images in total). All the images in it are normalized to the size of 640×480. We use different descriptors to describe the features in them, and oFAST is still adopted as a keypoint detector. For each image in the database, we query N-1 top-ranked (N is the number of images in its group) similar images from the whole database based on the matching results of descriptors. Since a large number of descriptors would be generated with the database, searching with an index structure is more appropriate for such problem than simple brute-force matching, and appears more often in applications.

For binary descriptors, we leveraged locality sensitive hashing (LSH) to perform an efficient approximate nearest neighbors (ANN) search in this experiment, and we set the LSH key size as 20 and the number of hash table as 6. For comparison purposes, we matched SURF descriptors using kd-tree and also set the number of kd-trees as 6.

Fig. 10. Examples from the datasets for object recognition experiment

Based on the query results of all the images, we calculate the average detection rates and precision for the descriptors. Experimental results are presented in **Table 3**. In general, LDB is still well suited for such object recognition task due to its high precision and less compution time. (The precision denotes the ratio between the number of correctly recognized objects and the total number of recognized objects.) However, the proposed RLDB descriptor achieves a higher detection rate (85.2 percent), clearly surpassing that of LDB (74.3 percent). Furthermore, the precision of RLDB is higher than those of other descriptors in this experiment. In fact, RLDB, LDB and Binboost all have advantages in precision, compared with BOLD and ORB which are based on single-pixel comparisons, since the multiple features from multi-scale sampling regions offer better robustness for the recognition. In addition, we can see from the table that RLDB has relatively fewer large-sized LSH buckets, which is an attribute of LDB. This leads to the faster matching speed in ANN search. (Matching time in **Table 3** is the average cost for searching the ANN of a descriptor from the database.)

Table 3. Comparison of experimental results for object recognition

Descriptor	Detection Rate (%)	Precision (%)	Extraction time (ms)	Matching time (ms)
LDB	74.3	88.1	0.043	1.03
RLDB	85.2	91.0	0.064	2.28
Binboost	83.4	88.8	0.597	2.16
BOLD	83.9	87.9	0.063	4.27
ORB	73.8	81.0	0.044	2.61
SURF	73.6	85.2	0.541	-

5. Conclusion

We have presented a new optimization framework, RLDB, for improving the performance of local difference binary descriptor. This framework implements an integrated learning approach by incorporating the LDP principle, which guarantees the robustness of features from various grid levels, so that a more complete gridding configuration can be applied. The more representative binary tests are selected globally by using our cascade bit selection

algorithm through offline learning. Online learning further optimizes descriptor for each specific patch locally, and masks the unstable features. We evaluate our RLDB extensively on public datasets. The experimental results show that using this framework leads to significant improvements over original LDB and maintains its advantage in efficiency. Furthermore, LDB is a typical region-based descriptor and a number of binary descriptors [41, 42] have similar structure with it, thus the proposed framework is also meaningful or illuminating for the design and usage of other binary descriptors.

References

[1] M. U. Kim and K. Yoon, "Performance evaluation of large-scale object recognition system using bag-of-visual words model," *Multimedia Tools & Applications*, vol. 74, no. 7, pp. 2499-2517, April, 2015. Article (CrossRef Link).

[2] L. Mansourian, M. T. Abdullah, L. N. Abdullah, A. Azman and M. R. Mustaffa, "A Salient Based Bag of Visual Word model (SBBoVW): improvements toward difficult object recognition and object location in image retrieval," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 2, pp. 769-786, February, 2016. Article (CrossRef Link).

[3] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 8, pp. 1362-1376, August, 2010. Article (CrossRef Link).

[4] H. Li, Y. Guan, L. Liu, F. Wang and L. Wang, "Re-ranking for microblog retrieval via multiple graph model," *Multimedia Tools & Applications*, vol. 75, no. 15, pp. 8939-8954, August, 2016. Article (CrossRef Link).

[5] X. Zhang, B. Guo and Y. Yan, "Image retrieval method based on IPDSH and SRIP," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 5, pp. 1676-1689, May, 2014. Article (CrossRef Link).

[6] Y. Guo, G. Zhao, Z. Zhou and M. Pietikainen, "Video texture synthesis with multi-frame LBP-TOP and diffeomorphic growth model," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3879-3891, October, 2013. Article (CrossRef Link).

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

[8] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision & Image Understanding*, vol. 110, no. 3, pp. 346-359, June, 2008. Article (CrossRef Link).

[9] E. Tola, V. Lepetit and P. Fua, "DAISY: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 5, pp. 815-830, May, 2010. Article (CrossRef Link).

[10] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Proc. of European Conference on Computer Vision*, pp. 778-792, September 5-11, 2010. Article (CrossRef Link).

[11] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. of IEEE International Conference on Computer Vision*, pp. 2564-2571, November 6-13, 2011. Article (CrossRef Link).

[12] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 506-513, June 27-July 2, 2004. Article (CrossRef Link).

[13] C. Strecha, A. Bronstein, M. Bronstein and P. Fua, "LDAHash: improved matching with smaller descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 1, pp. 66-78, January 2012. Article (CrossRef Link).

[14] H. Cai, K. Mikolajczyk and J. Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 2, pp. 338-352, February, 2011. Article (CrossRef Link).

[15] R. Ortiz, "FREAK: fast retina keypoint," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 510-517, June 16-21, 2012. Article (CrossRef Link).
[16] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proc. of IEEE International Conference on Computer Vision*, pp. 2548-2555, November 6-13, 2011. Article (CrossRef Link).
[17] X. Yang and K. T. T. Cheng, "Local difference binary for ultrafast and distinctive feature description," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 1, pp. 188-194, January, 2014. Article (CrossRef Link).
[18] B. Fan, Q. Kong, T. Trzcinski, Z. Wang and C. Pan, "Receptive fields selection for binary feature description," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2583-2595, June, 2014. Article (CrossRef Link).
[19] Y. Gao, W. Huang and Y. Qiao, "Local multi-grouped binary descriptor with ring-based pooling configuration and optimization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4820-4833, December, 2015. Article (CrossRef Link).
[20] V. Balntas, L. Tang and K. Mikolajczyk, "BOLD-binary online learned descriptor for efficient image matching," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2367-2375, June 7-12, 2015. Article (CrossRef Link).
[21] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. of European Conference on Computer Vision*, pp. 228-242, October 7-13, 2012. Article (CrossRef Link).
[22] T. Trzcinski, M. Christoudias and V. Lepetit, "Learning image descriptors with boosting," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 3, pp. 597-610, March, 2015. Article (CrossRef Link).
[23] S. Winder, G. Hua and M. Brown, "Picking the best daisy," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 178-185, June 20-26, 2009. Article (CrossRef Link).
[24] V. Balntas, E. Johns, L. Tang and K. Mikolajczyk, "PN-Net: conjoined triple deep network for learning local image descriptors," *arXiv preprint*, January, 2016. Article (CrossRef Link).
[25] X. Han, T. Leung, Y. Jia, R. Sukthankar and A. C. Berg, "MatchNet: unifying feature and metric learning for patch-based matching," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3279-3286, June 7-12, 2015. Article (CrossRef Link).
[26] K. Simonyan, A. Vedaldi and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 8, pp. 1573-1585, August, 2014. Article (CrossRef Link).
[27] M. Brown, G. Hua and S. Winder, "Discriminative learning of local image descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 1, pp. 43-57, January, 2011. Article (CrossRef Link).
[28] A. Richardson and E. Olson, "TailoredBRIEF: online per-feature descriptor customization," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 28-October 2, 2015. Article (CrossRef Link).
[29] G. Hua, M. Brown and S. Winder, "Discriminant embedding for local image descriptors," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, October 14-20, 2007. Article (CrossRef Link).
[30] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, May, 2004. Article (CrossRef Link).
[31] B. Guo and J. Liu, "Real-time keypoint-based object tracking via online learning," in *Proc. of IEEE International Conference on Information Science and Technology*, pp. 907-911, March 23-25, 2013. Article (CrossRef Link).
[32] J. M. Morel and G. Yu, "ASIFT: a new framework for fully affine invariant image comparison," *Siam Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438-469, April, 2009. Article (CrossRef Link).
[33] M. Ozuysal, M. Calonder, V. Lepetit and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 3, pp. 448-61, March,

2010. Article (CrossRef Link).
[34] S. A. J. Winder and M. Brown, "Learning local image descriptors," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 17-22, 2007. Article (CrossRef Link).
[35] M. Brown, "Multi-view stereo correspondence dataset," Article (CrossRef Link).
[36] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, October 2005. Article (CrossRef Link).
[37] Visual Geometry Group, Department of Engineering Science, University of Oxford, "Affine covariant regions datasets," Article (CrossRef Link).
[38] H. Shao, T. Svoboda and L. V. Gool, "Zubud-zurich buildings database for image based recognition, " Article (CrossRef Link).
[39] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2161-2168, June 17-22, 2006. Article (CrossRef Link).
[40] D. Nistér and H. Stewénius, "Recognition benchmark images," Article (CrossRef Link).
[41] M. Oszust, "An optimisation approach to the design of a fast, compact and distinctive binary descriptor," *Signal Image & Video Processing*, vol. 10, no. 8, pp. 1-8, November 2016. Article (CrossRef Link).
[42] S. Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Proc. of International Conference on Biometrics*, pp. 828-837, August 27-29, 2007. Article (CrossRef Link).

Huitao Sun received his B.S. degree in automation from Yanshan University, Hebei, China, in 2010. He is currently working toward the Ph.D. degree at Dalian University of Technology. His research interests include image processing and computer vision. Email: sht229@mail.dlut.edu.cn

Muguo Li received his B.S. degree in automatic engineering from Dalian University of Technology, Dalian, China, in 1978. He is currently a Professor with the State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology. His research interests include computer vision, image processing, and automatic control. Email: lmuguo@126.com