Local Multi-Grouped Binary Descriptor with Ring-based Pooling Configuration and Optimization

Yongqiang Gao, Weilin Huang, and Yu Qiao

Abstract—Local binary descriptors are attracting increasingly attention due to their great advantages in computational speed, which are able to achieve real-time performance in numerous image/vision applications. Various methods have been proposed to learn data-dependent binary descriptors. However, most existing binary descriptors aim overly at computational simplicity at the expense of significant information loss which causes ambiguity in similarity measure using Hamming distance. In this paper, by considering multiple features might share complementary information, we present a novel local binary descriptor, referred as Ring-based Multi-Grouped Descriptor (RMGD), to successfully bridge the performance gap between current binary and floated-point descriptors. Our contributions are two-fold. Firstly, we introduce a new pooling configuration based on spatial ring-region sampling, allowing for involving binary tests on the full set of pairwise regions with different shapes, scales and distances. This leads to a more meaningful description than existing methods which normally apply a limited set of pooling configurations. Then, an extended Adaboost is proposed for efficient bit selection by emphasizing high variance and low correlation, achieving a highly compact representation. Secondly, the RMGD is computed from multiple image properties where binary strings are extracted. We cast multi-grouped features integration as rankSVM or sparse SVM learning problem, so that different features can compensate strongly for each other, which is the key to discriminativeness and robustness. The performance of RMGD was evaluated on a number of publicly available benchmarks, where the RMGD outperforms the state-of-the-art binary descriptors significantly.

Index Terms—Local binary descriptors, ring-region, bit selection, Adaboost, convex optimization.

I. Introduction

LOCAL image description is a challenging yet important problem and serves as a fundamental component for broad image and vision applications, including object detection/recognition [1], [2], image classification [3], [4], face recognition [5]–[9] etc. With the increasing demands of advanced descriptors, a large number of local features have been developed in the last two decades. Typical examples include Scale Invariant Feature Transform [10], Local Binary Pattern [5], [11], Histogram of Orientated Gradient [12], Region Covariance Descriptor [13], [14]. SIFT [10] and SURF [15] are two successful and widely applied descriptors among them. A huge efforts have been devoted to improving their discriminative capabilities and robustness. However, most of

Y. Gao, W. Huang and Y. Qiao are with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 100864, China. The authors are also with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. E-mail: {yq.gao;wl.huang;yu.qiao}@siat.ac.cn.
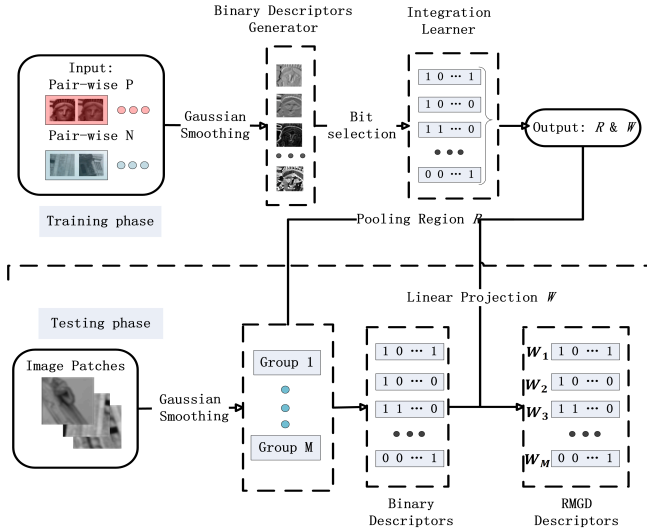
them are hand-crafted descriptors, which significantly limit their generality to various tasks or different databases by using pre-defined filters and unfeasible pooling configurations. Recently, learning-based descriptors have been proposed by optimising both local filters and pooling regions using training data, with promising improvements achieved [16]–[18]. The discriminative capability and computational complexity are two crucial but conflicted issues, which need to be balanced carefully in the learning processing.

Most high-performance image descriptors require float-point computation, to achieve promising distinctiveness and robustness by imposing a heavy computational cost. With the rapid growth of the vision applications in large-scale data sets or in low power mobile devices, binary descriptors have been attracting increasingly attentions, due to their numerous advantages, including low memory footprint, fast computation and simple matching strategy. In contrast to the float-point ones, the binary descriptors encode patch information using a string of bits and apply hamming distance for measuring similarity using fast XOR operation. They can achieve reasonable performance by comparing against the float-point ones, while only running in a fraction of the time required.

The binary descriptors can be generally categorized into two groups. In the first group, the binary strings are computed upon the float-point features in order to reduce computational cost without significantly compromising its performance. Quantization [19], and hashing techniques [20] are adopted to generate the bit strings from the float-point features. But their performance are largely limited by the qualities of the intermediate float representations. The other group of methods obtain the binary strings directly from raw image patches, mainly by measuring intensity differences between predefined pixel locations [21], [22] or pooling regions [23]. To improve the quality of the binary descriptor, learning methods are further applied for bit selection by optimizing the pixel locations or pooling regions [17], [24].

Although the binary descriptors have advantage in speed, they generally exhibit less discriminative power and robustness than the float-point equivalents. The quality of a binary descriptor is mainly determined by the pooling configuration and the strategy of binary tests. Current binary descriptors often suffer from several limitations. Firstly, a number of descriptors make use of intensity difference between two individual pixels for binary tests, which are sensitive to noise and spatial deformation. Secondly, region based binary tests would improve its stability and informativeness, but they are computed from a limited set of pre-defined pooling regions with fixed shapes and scales (e.g. rectangular or Gaussian pooling areas in

Fig. 1. Illustration of the proposed framework. The pooling region R and group weight W are learned from training data, and then are adopted for test.

[23]). This results in an uncompleted description by discarding a large amount of constrastive information between regions of different shapes, scales or distances. Thirdly, they mostly compute binary features from a single image property, such as intensity or gradient. Few of them extracts binary features from multiple image properties simultaneously, while an efficient algorithm for optimizing multiple groups of the binary features have not been explored in previous research.

The goal of this paper is to bridge the performance gap between binary and float-point descriptors by trading off distinctiveness, robustness and simplicity. Considering strong complementary information between various binary groups, we present a new local binary descriptor, referred as ring-based multiple-grouped descriptor (RMGD), to increase the informativeness and discriminative capability of current binary descriptors, while maintaining its computational feasibility to large-scale applications. The pipeline of the RMGD is described in Fig. 1. We first present a novel leaning based pooling configuration to encode more meaningful and compact information into multi-grouped binary strings. Then two powerful learning algorithms are proposed for effectively optimizing groups' weights, which further increase its discriminative ability. Our main contributions are summarized as follows.

First, we develop a new spatial pooling configuration by dividing an image patch into multi-scale ring regions, as shown in Fig. 3. A binary string is derived by computing the differences of all possible region pairs of various shapes, scales or distances. This yields a richer description than other alternatives. It is more nature for objects to appear in different sharps, scales or spatial distances in a natural image. With the pooling scheme, the descriptor is capable of encoding both coarse-level global feature and fine-level local information, which enhance its informativeness and distinctiveness.

Second, we introduce an efficient greedy algorithm for large-scale bit selection by extending the Adaboost algorithm. Motivated by Yang and Cheng's method [25], equal weight and accumulated error are adopted by our method to keep the

binary nature and joint optimization between the selected bits. Furthermore, our greedy algorithm leverages high variance and low correlation objectives to effectively handle a much larger-scale problem (with more than two orders of magnitudes in the number of bits), which not only results in a fast selection, but also leads to a compact and discriminative description.

Third, our binary descriptor is derived from multiple image properties, including intensity, multiple gradient channels. We propose two learning methods to effectively optimize the multi-group binary strings, so that they complement each other strongly, which is the key to discriminations and robustness. Firstly, we cast the multi-grouped optimization as a pair-wise ranking problem, and solve it effectively in a rankSVM framework. Secondly, the grouped weight learning is formulated as a convex optimization problem by penalizing the objective function with a $L1$ constraint to induce sparsity of the weights for optimization.

Finally, the RMGD outperforms the state-of-the-art binary descriptors significantly on a number of benchmarks, and achieves comparable performance of current float-point descriptors but with a fraction of computation and memory.

The rest of the paper is organized as follows. In Section II, we briefly review related studies on current local feature descriptors. Then details of the proposed RMGD are described in Section III, including a novel spatial pooling scheme and multi-grouped learning for feature optimization. The evaluation of the RMGD are detailed in Section IV. In section V, we investigate the performance of RMGD on two applications: image matching and object recognition, followed by the conclusion in Section VI.

II. RELATED WORK

SIFT [10] has been known as the most successful local image descriptor in the last decade. It extracts image feature by computing a number of local histograms from multiple oriented gradient channels, which enables it with highly descriptive power and strong robustness against multiple image distortions. With the goal of fast computation, SURF [15] was proposed by employing responses of Haar wavelets for approximating gradient orientations in the SIFT, and achieves great speed acceleration without significantly decreasing their performance. Recently, a number of learning based descriptors have been proposed in order to tackle hand-crafted limitations of the traditional descriptors [26]–[28]. Promising improvements have been achieved due to their data-driven properties, which learn to optimize the pooling configurations and the other aspects of the underlying representation [27]–[29]. However, by using the costly float-point operation, these descriptors are still too computationally expensive to extract and to match, making them prohibitively slow for many real-time applications.

Binary descriptors are of particular interest with its distinct advantages on computational simplicity and low storage requirement. BRIEF realizes simplicity and fast speed by simply computing the binary tests from a set of randomly selected pairs [21]. However, it has been shown that binary information generated by such simple pixel-based operation is

highly sensitive to noise, yet not robust to rotation and scale changes [23], [25]. To alleviate these limitations, ORB [30] and BRISK [31] were proposed to enhance scale and rotation invariance by introducing an orientation operator and image pyramids. Both methods further increase their discriminative capabilities by improving their pooling configurations. The ORB selects highly uncorrelated pixel pairs for binary tests, while the BRISK emphasizes locality by computing intensity differences between two short-distance pixels in a predefined circular sampling pattern. FREAK [32] further improved the performance of the BRISK by defining a novel retinal based pooling scheme. These hand-designed descriptors, comparing raw intensities of pixels with manually-defined pooling configurations, may result in a significant information loss.

Obviously, the pooling configuration is crucial to the quality of binary descriptors. Recently, a few methods have been developed to optimize their pooling configurations using training data. D-BRIEF [16] projects an image patch onto a more discriminative subspace learned, and then builds the binary descriptor by thresholding their coordinates in the subspace. Trzcinski *et al.* [17] proposed Boosted Gradient Maps (BGM) by leveraging boosting-trick to optimize weights of the spatial pooling regions which are considered as weak learners in the boosting framework. Similarly, Binboost [24] computes a weak learner from each pooling region in the image gradient space, and then jointly optimizes the weak learners and their weights by a complex greedy algorithm. Yang and Cheng [33] proposed an ultra-fast binary descriptor, named Local Difference Binary (LDB). They computed the binary strings from pairs of equal-sized spatial regions in both intensity and gradient spaces. An efficient bit selection scheme extended from the AdaBoost [34] was applied for bit selection. Receptive fields descriptor (RFD) [23] computes the binary descriptors from defined receptive fields which are optimized using a simple greedy algorithm by sorting all the candidate pooling fields with their discriminative scores.

The proposed RMGD is related to the LDB [33] in using the region-based binary test to generate the binary strings, and proposing an extended Adaboost algorithm for fast bit selection. But our descriptor differs from it by proposing more principled approaches for both the pooling configuration and multiple-group optimization, which are the key to considerable performance improvement. Our work is also similar to the RFD [23]. We compute the binary strings directly from a raw image patch, while the RFD first extracts a float-point descriptor from a patch and then binary strings are generated by thresholding it. Furthermore, both the LDB and RFD involves binary tests with the equal-sized pooling configurations. By contrast, our descriptor is able to generate binary features from the full set of region pairs with different shapes, scales, and distances, leading to a complete and more meaningful representation by encoding both local and global information.

Recently, weighted Hamming distances for binary descriptors are studied. Fan et al. [35] claimed that the distinctiveness of each element is usually different and a more reasonable way is to learn weights for different elements of the binary descriptors. Feng et al. [36] defined Absolute Code Difference Vector (ACDV) and learned the weights of ACDV. The RFD

only computes binary features in gradient space [23], and the LDB simply combines binary descriptors computed from intensity and the first-order gradients [33]. For our RMGD, we learn weights for various groups of the binary descriptors. The RMGD not only considers the superiority of binary descriptors, simple computation and low memory, but also takes into account the strong complementary information between groups. We propose two learning methods, based on the rankSVM framework and the convex optimization algorithm respectively, to effectively optimize different feature groups derived from multiple image properties. This leads to a further improvement on discriminative capability and robustness by leveraging complementary properties among various feature groups.

III. RMGD: Ring-based Multi-Grouped Descriptor

This section presents the details of the proposed Ring-based Multi-Grouped Descriptor (RMGD), including the pooling configuration and multi-grouped binary features optimization. We introduce a new spatial pooling scheme by dividing an image patch into multi-scale ring regions, for which binary comparisons are calculated. Then, we present two leaning methods, based on the rankSVM and convex l_1-optimization, to solve the multi-grouped optimization problem effectively.

A. Problem Definition and Formulation

Given an image patch $\mathbf{x}$, we aim to generate a compact yet powerful binary descriptor $RMGD_M(\mathbf{x}) = \{B^m(\mathbf{x})\}_{m=1}^M$ which consists of M groups with N bits in each group. $B^m(\mathbf{x}) \in \mathscr{H}^N$ is in Hamming space, and is computed directly from a raw image property (e.g., intensity or gradient). Generally, the number of bits can be different for different groups. Here we use the same number of bits just for simplicity. Each bit is computed as,

$$B_n^m(\mathbf{x}; R_{n1}, R_{n2}) := \begin{cases} 1 & \text{if } f_m(\mathbf{x}, R_{n1}) < f_m(\mathbf{x}, R_{n2}) \\ 0 & \text{otherwise} \end{cases},$$

where $1 \le n \le N$, $1 \le m \le M$, $f_m(\mathbf{x}, R)$ denotes the operation of extracting certain image feature from a region R within the patch $\mathbf{x}$, we use the average property value (e.g. mean intensity) of a region as its feature. R_{n1} and R_{n2} denote a pair of spatial sampling regions in the patch. How to design these region pairs for binary tests is often referred as pooling configuration, which plays a key role in the performance of a local descriptor. It includes two main steps: a spatial sampling scheme for generating possible region candidates and an efficient learning algorithm for optimally selecting most distinctive and compact pairs for binary tests. We develop new methods for both steps.

Another novelty of our RMGD is its capability for generating binary strings from multiple image properties. The key factor for improving the performance is to optimally weight various binary groups. We define the weight vector as $W = \{w_1, w_2, \ldots, w_M\}$, corresponding to M groups of the binary strings. In the next, we will discuss how to learn these parameters from training data. Two algorithms are presented
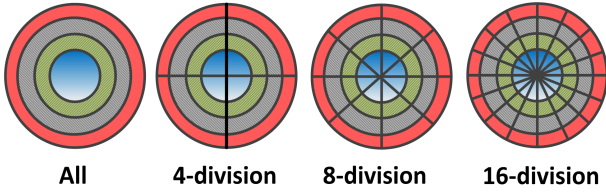
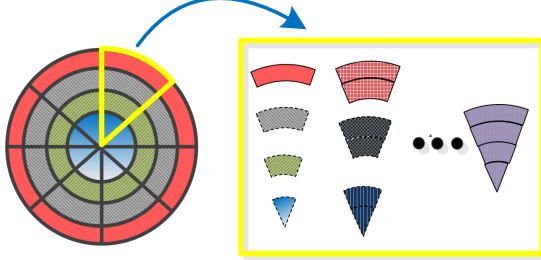Fig. 2. Four types of pooling regions: all, 4-division, 8-division and 16-division (Best viewed in color).

Fig. 3. 8-division ring-region sampling example (Best viewed in color) .

in our framework. Notice that the weights (W_m) can be float-point or integer values, since they are not assigned to the generated binary strings directly. In order to keep the binary nature of the descriptor, they are used to weight Hamming distance after XOR operation in matching processing.

B. Pooling Configuration

1) Spatial Ring-Region Sampling: As mentioned, randomly single-pixel comparisons may suffer from the problems of weak robustness and instability, we release this problem by using region-based sampling strategies as previous works [17], [25], [26], [37], [38]. It has been shown that the region-based intensity difference is robust to most of photometric changes, e.g. lighting/illumination changes, blurring, image noise, and compression artifacts [25], [39].

The design of the region sampling is crucial to the performance of binary descriptors. A number of issues should be considered and traded off carefully. The feature extracted from a small region is able to capture more detailed local information, which often includes more discriminative characteristic, but has low robustness and instability against noise and spatial distortions. While the feature computed from a large region would result in a more robust and stable representation by encoding more global information. But it has less distinctive power. Most exiting descriptors utilize fixed-size sampling regions, and hence are not powerful to integrate both local and global information effectively [17], [26], [37], [38].

To encode richer information, Yang and Cheng [25] proposed the Local Difference Binary descriptor (LDB) by sampling an image patch into multi-scales regions. The binary tests are conducted by comparing the intensity or gradient differences between paired regions of same scale and shape (e.g. rectangular). The multi-scale approach enables the LDB to capture both local and global features of the image, leading to considerable performance gains. To this end, we improve the LDB further by developing a more complete description that generates the region pairs in a full set of different scales, shapes and distances.

We introduce a ring-region sampling scheme to generate a large number of pooling region candidates with multiple scales and shapes. As shown in Fig. 2, an image patch is first densely divided into a number of ring regions centered at central of the patch. Specifically, suppose a patch **x** with the size of $k \times k$, we compute radius of the patch as $r = floor(k/2)$, where $floor(y)$ is the maximum integer not larger than y. We generate r element ring-regions from this patch. All possible combinations of these element regions are considered to generate a complete set of pooling regions with multiple scales and shapes. Then each generated ring region is further divided into a number of sub divisions (e.g. 4, 8 or 16). Finally, it generates $z = t \times \frac{(r+1) \times r}{2}$ pooling regions in total. t is the number of divisions. The details of region combination and division are illustrated in Fig. 3. In contrast to most previous work comparing equal-sized regions, we compute the binary tests by comparing the mean intensity differences between all possible region pairs of different shapes, scales and spatial distances, so as to encode more meaningful and distinctive information. Hence, the number of the complete region pairs is $\binom{z}{2}$. For instance, given a smoothed patch **x** with resolution of 32×32, the single-division case (referred as "All''" in Fig. 2) includes 136 pooling regions that generates a total 9,180 region pairs for binary tests. The numbers of the generated pooling regions for the "4-division", "8-division" and "16-division" cases are 544, 1,088 and 2,176, corresponding to their full sets of region pairs of 147,696, 591,328 and 2,366,400, respectively. Comparing to the spatial sampling of the LDB [25], which divides a patch into a small number of large regions (e.g. from 2×2 to 5×5) and computes the binary tests by comparing two regions of the same scale and shape, our binary strings encode much more detailed features and important contrastive information between the regions of various scales and shapes, and hence stronger discriminative capability can be expected.

2) Boosted Bit Selection with Correlation Constraints: The proposed densely sampling and full comparison scheme generates a complete and meaningful description of an image. However, it also results in a huge number of region pairs to compare (e.g. 591,328 in the "8-division" case), making it prohibitively slow in practice. Moreover, the resulted long binary string may be highly redundant by including a large number of strongly correlated and noise (e.g. low variance) bits. In order to achieve a compact representation and fast computation, we aim to select a small number of the most informative region pairs for the binary tests.

Optimizing the binary tests over the full set of region pairs poses a difficult problem due to the huge number of the possible regions. Fortunately, the boosting methods are particularly well-adopted for this problem with good performance achieved [3], [24], [25], [40], [41]. Yang and Cheng [25] proposed an efficient greedy algorithm for bit selection by improving the original Adaboost [34], [41] at two aspects. (1) Forcing equal weights for all selected features to keep the binary nature of the descriptor; (2) Using the accumulated error as bit selection criterion to enhance the complementarity between the selected bits. However, our problem involves a

much larger-scale bit selection, where the candidate bit number is about two order of magnitude than that of the LDB [25], directly applying the algorithm to our problem may cause two problems. Firstly, Yang and Cheng's method [25] essentially does not strongly enhance the uncorrelation between bits by using the accumulated error instead of the single-bit error, and hence easily leads to a local minimum for our large-scale bit selection. Secondly, the computational cost can be increased substantially.

Motivated from [30], [42] which indicate that, for an efficient binary descriptor, each bit should have 50% chance of being 1 or 0, and different bits should have small correlation, we emphasize high variance and low correlation criterions to make the descriptor more discriminative. We develop a two-step bit section method. First, we implement a raw but fast selection scheme to generate a subset of bit candidates with low classification errors and high variance. Second, a correlation-constrained Adaboost algorithm is proposed to further optimize the selected bits. Although greedy, our algorithm is highly efficient for large-scale bit selection with the goal of searching for a small set of uncorrelated yet highly-variant bits, leading to a compact and discriminative representation. Details of the Boosted Bit Selection with Correlation Constraints (BBSCC) are described in Algorithm 1.

The numbers of matching and non-matching pairs for training are 1:3, since the non-matching cases are often much more than the matching cases in practice. Our experiments also show that this strategy outperforms that using equal numbers of them. The correlation between two bits is calculated as Pearson correlation through all training examples:

$$corr(b_{t_1}, b_{t_2}) = \frac{\sum_{i=1}^{|X|} b_{t_1}^i \bigoplus b_{t_2}^i}{\sqrt{\sum_{i=1}^{|X|} (b_{t_1}^i)^2} \sqrt{\sum_{k=1}^{|X|} (b_{t_2}^i)^2}}, \quad (2)$$

where $\bigoplus$ denotes XOR operation.

C. Multi-Grouped Features Optimization

The intensity is a fundamental image property and generally includes meaningful information for image description, so that most image descriptors extract their features from the intensity space. Great success of recent local descriptors show that image gradient space is capable of encoding inherent underlying structure of the image, which have been shown to be more powerful for image representation than the intensity in many applications, such as SIFT [10], HOG [12], GLOH [43] and BinBoost [37]. To this end, we aim to explore the advantages of both spaces to achieve a more robust and discriminative representation. Specially, we compute binary strings from multiple image properties (13 in total), including the intensity, x-partial, y-partial, gradient magnitude, orientation, and eight channels by soft assigning the gradient orientations. Finally, 13 groups of binary strings are generated from an image patch.

As expected, we achieved considerable performance improvements by simply combining multi-grouped binary strings, as indicated in our experiments in Section IV. B. It would be interesting to find the impacts of different binary groups which may have various contributions in the representation. And it can be expected that a good weighting on

---

Algorithm 1 Boosted Bit Selection with Correlation Constraints

Input:
A set of training data $T = \{X, Y\}$, where X_i is a pair of image patches. $Y_i = 1$ indicates a matching pair, while $Y_i = 0$ is for a non-matching pair;

Output:
The optimized bits or bit positions, $C = \{c_1, c_2, \ldots, c_n\}$, n is the number of the selected bits.
1: Compute N-bit descriptors for all patches in T.
2: Compute a matching error for each bit: $\frac{1}{M} \sum_{i=1}^{M} |Y_i - \hat{Y}_i|$ where $\hat{Y}_i$ is the predicted label of a pair, and is computed from our binary test,
   $M = |T|$ is the number of training pairs.
3: Order the errors in ascending and select the first $N/2$ bits.
4: Compute the mean of each bit through all training pairs: $\frac{1}{M} \sum_{i=1}^{M} Y_i$.
5: Choose $N/4$ bits whose means are mostly closed to 0.5, from the selected $N/2$ bits.
6: Set equal weight $d_i = 1/M$ to all training pairs X_i.
7: Set $C = \phi$
8: AdaBoost-based Bit Selection:
9: **for** $t = 1$ to n **do**
10:   Find a bit b_t with the minimum accumulated error:
      $b_t = \text{argmin } \varepsilon_{accu}(t), \varepsilon_{accu}(t) = \varepsilon_{accu}(t-1) + \varepsilon_t$,
      $\varepsilon_{accu}(0) = 0, \varepsilon_t = \sum_{i=1}^{M} d_i |Y_i - \hat{Y}_i|$.
11:   Compute the correlation rate, $corr(b_t, c_k), c_k \in C$.
12:   **if** $corr(b_t, c_j) < t_c, \forall c_j \in C$ **then**
13:     Set $C = C \cup b_t$
14:   **end if**
      t_c is the correlation threshold, which is set empirically.
15:   **if** $\varepsilon_t < 0.5$ **then**
16:     Update weights: $d_{t+1,i} = \frac{d_{t,i}}{Z_t} \times e^{\varphi_{t,i} \times \alpha_t}$,
        where $\varphi_{t,i} = 1$, if $\hat{Y}_i = Y_i$; $\varphi_{t,i} = -1$, otherwise.
        $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$, and Z_t is a normalizing factor.
17:   **else**
18:     Switch to a new training set and reset $d_i = 1/M$.
19:   **end if**
20: **end for**

---

the grouped features would achieve a better optimization, which may lead to a further improvement on the performance. Therefore, our goal is to learn the weights from provided training data. Specially, we define a weight vector as $W = [w_1, w_2, \ldots, w_M]$ for M groups of binary strings. We cast the weight learning as an optimization problem with an objective, which encourages that the distances of the non-matching pairs (**N**) are larger than those of the matching pairs (**P**):

$$d_W(x, y) + 1 < d_W(u, v), \quad \forall (x, y) \in P, \quad \forall (u, v) \in N, \quad (3)$$

the $d_W(x, y)$ is defined as:

$$\begin{aligned} d_W(x, y) &= \sum_{m=1}^{M} w_m d(B^m(x), B^m(y)) \\ &= W^T D(\mathbf{B}(x), \mathbf{B}(y)) \\ &= W^T D(x, y), \end{aligned} \quad (4)$$

where $d(B^m(x), B^m(y))$ denotes the Hamming distance computed from the m-th group of the binary strings $\{B(x), B(y)\}$. $D(\mathbf{B}(x), \mathbf{B}(y))$ is a distance vector with $\{d(B^m(x), B^m(y))\}_{m=1}^M$, and M is the number of groups. The Eq. 3 can be considered as a hinge loss in the formulation as: $\mathcal{L}(z) = max(z+1, 0)$, where $z = d_W(x, y) - d_W(u, v)$. Then we derive the following convex optimization problem by minimizing $\mathcal{L}(z)$:

$$\min_{W \geq 0} \sum_{\substack{(\mathbf{x}, \mathbf{y}) \in \mathbf{P} \\ (\mathbf{u}, \mathbf{v}) \in \mathbf{N}}} \mathcal{L}(W^T(d(x, y) - d(u, v))) + \mu_l ||W||_l, \quad (5)$$

where $||W||_l$ is the penalty on the learning weights, for which we adopt two forms: l_1 norm $||W||_1$ (lasso penalty [44]) and l_2 norm $||W||_2^2$ (ridge penalty [45]); $\mu_l > 0$ $(l = 1, 2)$, is a tuning parameter which balances the error loss and penalty. The W is a vector with non-negative elements. We present two methods to solve the convex optimization problem with different norms effectively.

1) l_2 norm: The objective is to learn the W which makes as many as possible of the pairs to satisfy the Eq. 3. With the l_2 constraint on the W, the convex optimization of Eq. 5 can be reduced to a ranking SVM problem [46]–[48] by introducing the (non-negative) slack variables ξ:

$$
\begin{aligned}
W^* = \arg\min_W ||W||^2 + \frac{1}{\mu_2} \sum \xi_{i,j} \quad &s.t. \\
W^T d(u_i, v_i) - W^T d(x_j, y_j) \geq 1 - \xi_{i,j} & \\
\forall i, \forall j: \quad \xi_{i,j} \geq 0 & \\
(u_i, v_i) \in N, (x_j, y_j) \in P &
\end{aligned}
\quad (6)
$$

If a training example lies on "wrong" side of the hyperplane, the corresponding $\xi_{i,j}$ is greater than 1. Therefore $\sum \xi_{i,j}$ yields an upper bound on the number of training errors. This means that the rankSVM finds a hyperplane classifier that optimizes an approximation of the training error regularized by the l_2 norm of the weight vector [46], [48]–[50].

2) l_1 norm: If the l_1 norm is adopted, we derive the following non-smooth convex optimization problem:

$$\min_{W \geq 0} \sum_{\substack{(\mathbf{x}, \mathbf{y}) \in \mathbf{P} \\ (\mathbf{u}, \mathbf{v}) \in \mathbf{N}}} \mathcal{L}(W^T(d(x, y) - d(u, v))) + \mu_1 ||W||_1, \quad (7)$$

This is intrinsically similar to sparse support vector machines [44], [51]–[53]. which are highly effective in variable ranking and selection [51]. Our objective is to select most informative feature maps by using the sparsity-inducing regulariser, and weight the selected groups optimally. Simonyan et. al [26] proposed a method for learning pooling regions based on Euclidean distance in the descriptor space, while we adopt the Eq. 7 for multi-grouped features ranking and selection by computing the Hamming distance between the binary strings.

Typically, an extreme large number of the pairs is employed for learning, e.g. 500,000 in our experiment. It makes conventional interior point methods infeasible. By following [26], we adopt Regularized Dual Averaging (RDA) [54] to optimize the Eq. 7, formulating the objective function into an online setting. Typically, this objective function contains the sum of two convex terms: one is the loss function of the learning

task and the other is a simple regularization term. In our case, the second regularization term is the "soft" l_1-regularization, $\mu_1 ||w||_1$. By following the principles of RDA, an auxiliary function $h(w) = \frac{1}{2}||w||_2^2$ is applied. Given a nonnegative and nondecreasing sequence $\beta_t = \gamma\sqrt{t}$ (t denotes the iteration), the specific form of the RDA update term for the Eq. 7 is,

$$w_{m,t+1} = \max\{-\frac{\sqrt{t}}{\gamma}(\bar{g}_m + \mu_1), 0\}, \quad (8)$$

where $\bar{g} = \frac{1}{t}\sum_{i=1}^t g_i$ is the average sub-gradient of the corresponding hing loss function at iteration t, μ_1 is the parameter in Eq. 7, and w_m can be fine-tuned by different values of μ_1.

IV. EXPERIMENTS

In this section, we present extensive experimental results to evaluate efficiency of the RMGD, and investigate the properties of our method by showing the performance improvements by each independent component. The experiments were conducted on three challenging and widely-used local image patch datasets [27], [55]: Liberty, Yosemite and Notre Dame. Each dataset contains over 400k scale- and rotation-normalized 64×64 image patches, which are detected by Difference of Gaussian (DoG) maxima or multi-scale Harris corners. The ground truth for each dataset is available with patch pairs of 100k, 200k, and 500k, where 50% for matching and the other 50% for non-matching pairs. In this paper, we resize the training and test patches into 32×32, and all patches are smoothed by a Gaussian kernel with standard deviation. These pre-processing are the standard steps by following previous work in [16], [21], [24], [30]. We report results of the evaluation in terms of ROC curves and false positive rate at 95% recall (FPR @ 95%) which is the percentage of incorrect matches obtained when 95% of true matches are found, as in [27] and [37]. More details can be found in project webpage[1].

Assume that "8-division" ring-region sampling scheme is adopted. There are totally 591,328 tests for a local patch of size 32×32. We compute the binary strings from thirteen different feature maps. Fig. 4 shows an example channel for each map, and we denote them as "Int.", "X-part.", "Y-part." "Mag.", "Ori.", "Chan.1" $\sim$ "Chan.8" for short. Intuitively, the "Int." and "Mag." maps consist of more local details comparing to the "X-part." and "Y-part.", while eight different originated maps exhibit to be not only discriminative but strongly complementary information to each others.

A. Evaluation of the Proposed Pooling Configuration

We conduct extensive experiments to evaluate our ring-based pooling configuration. For fair comparisons, the experiments were only implemented on the "Int." map. We investigate the efficiency of each independent component of it. The performance of the spatial ring-region sampling scheme is compared to that of the BRIEF without any learning process, then the efficiency of our bit selection method is
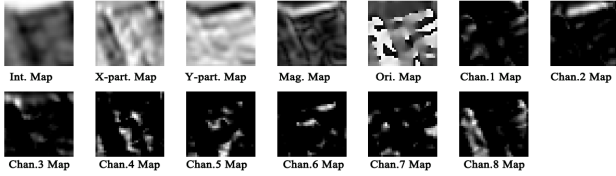
[1]http://mmlab.siat.ac.cn/yqgao/RMGD/

Fig. 4. Multiple feature maps of an image patch. The sequence of feature maps are intensity map (Int.), x-partial (X-part.), y-partial (Y-part.), gradient magnitude (Mag.), gradient orientation (Ori.), and soft assigning gradient maps with eight orientations from $[0, \pi/4]$ to $[7\pi/4, 2\pi]$.
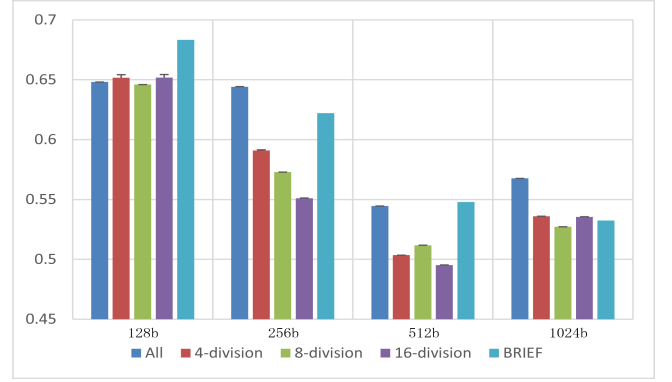

Fig. 5. False positive rate at 95% (FPR @ 95%) for ring-region features with different divisions on the datasets of 100k Notre Dame. Results are obtained by averaging 5 loops and all bits selected by uniform random and error bar indicates variance for each division with related bits.

evaluated. Finally, the whole pooling configuration including both components is further compared to recent methods.

The spatial ring-region sampling. A group of experiments were conducted to compare the performance of different division strategies of our spatial ring-region sampling method. In each case, we generated a set of binary descriptors by randomly selecting increasing numbers of the region pairs for the binary tests, e.g. $N = 128, 256, 512, 1024$. The generated binary descriptors were tested on the 100k Notre Dame database, and the false positive rates (at the *95%* recall) were reported as in [29]. Each error rate presented in the Fig. 5 is the average value of five independent random selections of the region pairs. Noting that most of variances locate in $10^{-4} \sim 10^{-6}$ which means they are relatively stable through our experiments. We adopt BRIEF [21][2] as the baseline. As shown in Fig.5, the "8-division" scheme achieves reasonable performance among the four cases by trading off their performance and the numbers of bit candidates. Although the "16-division" case generates a larger set of region pairs by dividing the patch into finer regions. These regions may be too small to encode enough global and robust information, and hence it dose not lead to a further large improvement, while doubling the number of the bit candidates. Therefore, we use the "8-division" for the RMGD in all our following experiments. In the "8-division" case, our method achieves 51.16% and 49.51% error rates at the 512- and 1024-dimensions respectively, which outperform the BRIEF at 56.78% and 53.25% considerably, even by randomly selecting a small number of bits from the generated binary strings. This indicates that the proposed ring-region sampling is highly beneficial, and is powerful for capturing meaningful local image feature.

it with recent BSB [25] which is also extended from the Adaboost for bit selection. For fair comparisons, both the BBSCC and BSB were implemented upon our ring-region sampling for selecting 256 bits from the total $591, 328$ bits. They were trained by using the "Liberty" dataset with two different scales: 4k and 40k pairs (both with 1:3 matching and non-matching pairs). They were test in the "Notre Dame" with 100k pairs (50k matching and 50k non-matching). The results are compared in Table I.

The BBSCC achieves the FPR at 38.46% and 28.27% for 4K- and 40K- training sets. Obviously, the proposed BBSCC improves the performance of the ring-region sampling with random bit selection and the BRIEF substantially by leveraging the training data, as shown in Fig. 5, and more training data would lead to a considerable reduction on the FPR. The BSB gets higher FPRs at 39.97% and 29.45% based on the same pooling configuration scheme, indicating that our algorithm with strong enhancements on high variance and low correlation leads to a more discriminative binary representation. Furthermore, in our experiment, we found that the BSB requires much more training time to optimize the compact bits from our large-scale binary strings, which is about four times of our methods. This indicates that our multi-step scheme is more efficient to handle the large-scale bit selection problem, and achieves a more compact representation.

TABLE I
COMPARISON RESULTS OF VARIOUS POOLING CONFIGURATION

Pooling Configuration	FPR @ 95%-4K	FPR @ 95%-40K
ORB	63.72	
RFD_R	40.56	-
RFD_G	41.34	-
Ring-BSB	39.97	29.45
Ring-BBSCC	38.46	28.27

The BBSCC bit selection strategy. We further show that the performance of RMGD can be improved considerably with the proposed bit selection methods (the BBSCC). We compare

TABLE II
AVERAGE TIME COSTS OF DIFFERENT DESCRIPTORS

Descriptor	Extracting time (ns)	Matching time (ns)
SIFT	235	940
BinBoost	6.48	36.45
BRIEF	12.54	35.46
RMGD	10.46	32.46

The whole pooling configuration. We further compare our full pooling configuration with the ORB [30] and RFD [23] in Table I. The ORB was improved from the BRIEF by learning the oriented BRIEF features from a given dataset. In the Liberty, Yosemite and Notre Dame datasets, the principal orientations of the image patches were normalised by the original authors [27], [55]. Thus the ORB is equivalent to

[2]codes: http://cvlab.epfl.ch/research/detect/brief, and the patch size and kernel size are 32 and 7, respectively.

the BRIEF when the principal orientation is given [16]. As can be seen, our pooling configuration has obvious advantages over the ORB by reducing the error rates considerably. Our method also outperforms recently-developed RFD [23], which achieves a higher error rates at 40.56% with the 4K training data. This improvement may be benefited from our meaningful binary representation generated from the full set of region pairs, and the efficiency of the bit selection algorithm. Besides, learning the RFD descriptor is highly memory demanding, making it prohibitive to be implemented in a large training set.

Computational complexity The main advantage of binary descriptors is that they require less computational and storage cost compared with float descriptors. We introduce circle integral image to speed up the computation of ring-based descriptor. The circle integral image is computed independently within each image patch or around each keypoint center, which is different from the integral image exploited by the SURF for rectangular regions [15]. Details of our circle integral image can be found in the Appendix. To validate the computational efficiency of the RMGD, we estimate the average time costs of feature extraction and feature matching on "wall" dataset [43]. Experiments are conducted on a PC with Intel (R) Core(TM) 2 Duo CPU E7500 @ 2.93 GHZ, 2.94 GHZ, 6.00 GB of RAM. Tab. II gives the average time costs of different descriptors, where the model of the RMGD is trained on the "Liberty" with 256 bits. The reported times include the computation times of the integral images. Notice that the BinBoost and BRIEF were run with provided C/C++ codes, while our method was implemented in Matlab, which could be further speeded up with more engineering work involved. Obviously, the RMGD still requires less time than the BRIEF in both extracting and matching times, while achieving substantial improvements on the performance. Note that RMGD is obtained by only one channel and it is calculated on circle integral image.

B. Evaluation of Multi-Grouped Feature Optimization

We evaluate the multi-grouped binary features optimization by comparing the performance of two proposed optimization methods (referred as "l_1-opt" and "l_2-opt") with single-group binary feature and direct combination of them with equal weights (the "No-opt"). The "l_1-opt" and "l_2-opt" are computed by using l_1 (Eq. 7) and l_2 norm (Eq. 6), respectively. We analyse insights of the proposed descriptor for performance improvements and discuss the contribution of each grouped feature and interaction between them. Our pooling configuration is applied on 13 different image properties (as shown in Fig. 4) for extracting 13 groups of compact binary strings (e.g. 128 or 256 bits) by Algorithm 1. In this experiment, we trained 60k patch pairs from the "liberty"(abbr. Lib), and tested on 100K pairs from the "Notre Dame" (abbr. NoD). Both datasets include 50% matching and the other 50% non-matching pairs. The results of feature combinations (including the "No-opt", "l_1-opt" and "l_2-opt") are reported, comparing to the performance of each single binary group. The comparisons are summarized in Fig. 6, and more experimental results by using the "Notre Dame" and "Yosemite" datasets as the training data are presented in our project webpage.

Three observations can be found from the left of Fig. 6. First, the binary feature from the "Int." map achieves the lowest FPR (at about 25%) among 13 single-grouped features, while the other single features get much larger FPRs independently. This correctly matches the fact that intensity map generally encodes main image information, and serves as a basic image property for general feature extractions. Second, direct combination of 13 grouped features leads to a large improvement with 5% reduction of the FPR (the "No-opt") over the best performance of the single feature. This indicates that, although gradient based binary features do not include as much detailed information as the intensity, they are able to capture robust global information which provides strong complementary to the local information. Third, the proposed optimization methods for group weights learning (the "l_1-opt" and "l_2-opt") lead to a further considerable improvements over the "No-opt", which finally reaches at about 15% on the FPR. These results clearly show importance of multi-grouped features combination for performance improvements, and efficiency of the proposed methods for weights learning.

We further compare three combination methods with varied bit numbers in the right sub-figure of Fig. 6. The test were conducted on the "Notre Dame" (abbr. NoD) and "Yosemite" (abbr. Yos) datasets. The performance of three methods are generally improved by increasing the numbers of bits. This is because more bits can encode more meaningful binary feature in each group, and thus make the final integrated descriptor more discriminative. However, as shown in the right sub-figure of the Fig. 6, the speeds of the performance improvements by the direct combinations are slowed down considerably when the numbers of the bits are increased from 512 to 1024. A similar phenomenon is shown in the Fig. 5, where the performance of the single-group binary descriptor is not always increased by increasing the number of the bits. The descriptor with more bits may easily include redundant information without an optimal selection. By contrast, the improvements of both optimization methods are consistently significant in this case, indicating that our group weight methods can enhance the complementarity between different grouped features. Furthermore, the proposed weight learning algorithms (the "l_1-opt" and "l_2-opt") consistently outperform the direct combination with a large margin. And among the two learning-based methods, we notice that the "l_1-opt" obtains slightly lower error rates than the "l_2-opt". This indicates that the "l_1-opt", which utilizes a sparsity-inducing regularizer, is more powerful for learning the weights of multiple groups. Another advantage of using "l_1-opt" is that it encourages the elements of W to be zero, which not only discards the redundant features, but also reduces the final computational cost in the testing phase. Therefore, the "l_1-opt" is adopted in our following experiments.

We develop two strategies to further improve the performance. Firstly, we enlarge the number of feature groups by dividing each group into eight subgroups with equal intervals, and hence we finally have $13 \times 8 = 104$ groups in total. This finer weighting scheme increases the discriminative power of our descriptor, and the 104 sub groups were chosen empirically by trading off the performance and its learning cost. Further
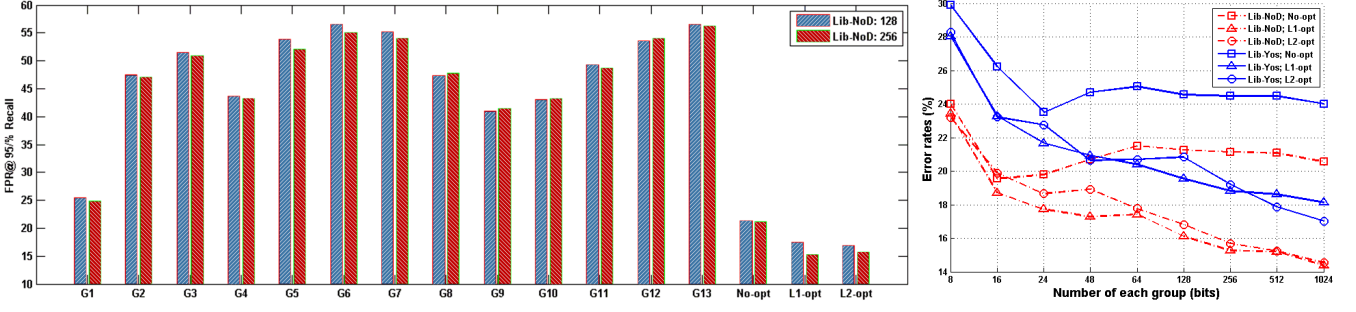
Fig. 6. Left: the results of 13 single-grouped binary features, compared to the multi-grouped binary features with direct combination (No-opt) and learning with l_1 (l_1-opt) and l_2 (l_2-opt) norm; train on the "liberty" and test on the "Notre Dame" with 128 and 256 selected bits. Right: Comparison of multi-grouped features with various numbers of bits.
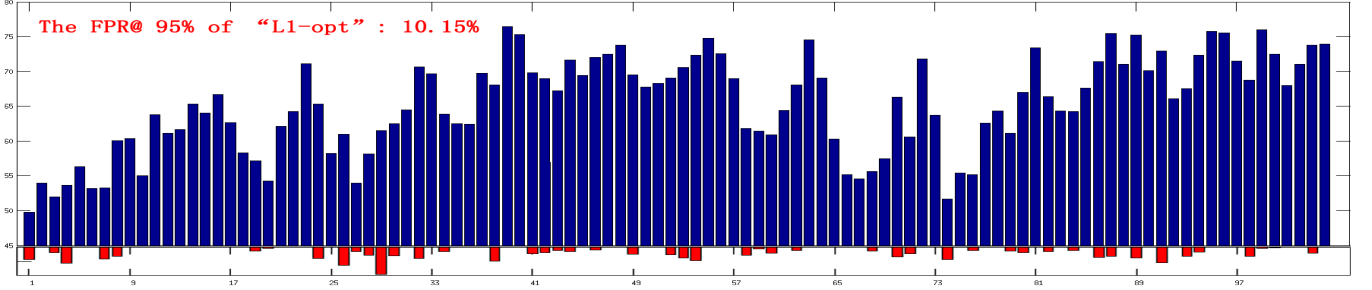

Fig. 7. The results of 104 single-subgrouped binary features (blue bar), comparing to the "l_1-opt" (red text), along with its sparse weights (red bar).
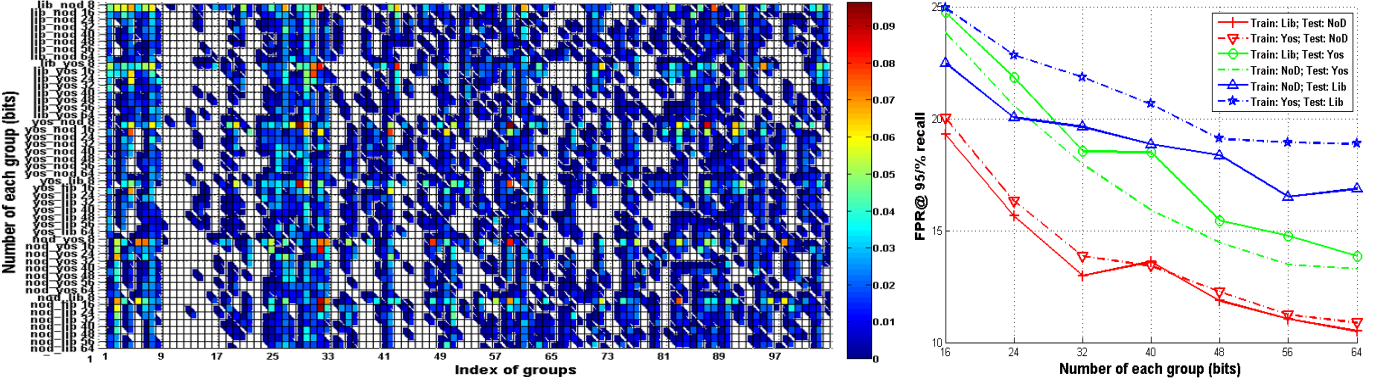

Fig. 8. Left: the distribution of learned weight by the "l_1-opt" on different splits of the training and test datasets with various numbers of the selected bits; The white blocks denote the groups with zero weights which are not applied for the test processing. Right: the FPR @95% recall on different numbers of the selected bits (Best viewed in color).

increasing the number of sub groups did not lead to an obvious improvement on the performance. Secondly, we increase the number of training image pairs to 500K to learn a better weight for each group. The test data is the same as previous experiment on the 13-grouped learning. The FPR values for each single-grouped binary feature (in Blue bar) and the "l_1-opt" combination are presented in Fig. 7, along with the learned weights (in Red bar) by the "l_1-opt".

By comparing Fig. 7 with the left of Fig. 6, we can find that the FPR value for each single feature in the 104-groups is generally higher than that in the 13-groups. It is natural that a grouped feature by a more finer division may loss some important information when it is used independently. However, the result is boosted substantially when we integrate all 104 groups together by using the "l_1-opt" optimization, and finally

get an appealing result at 10.15% of the FPR, which further improves the result of the "l_1-opt" with 13 groups (about 15% of the FPR) significantly.

By investigating the distribution of learned weights, we can find that large weights are converged on the "Int." and "Mag." maps, while they are highly sparse in the "X-part." and "Y-part." maps. It means that the "Int." and "Mag." maps include the most important information for patch representation, while the "X-part." and "Y-part." maps may have redundant or overlapped information. Another observation is that, although most single sub-group features from the "Chan.1" $\sim$ "Chan.8" do not achieve reasonable performance independently, some of them server as good complementary information for patch description (as indicated by their weight values), which is also a key factor to performance boosting. Furthermore, the

distributions of learned weights are highly consistent when we verified the training datasets with various numbers of the selected bits, which is visualized in the left of Fig. 8. This is a appealing property as it means that the learned weights obtained by our method dose not heavily depend on special training datasets, and hence has good generality.

The right of Fig.8 shows the performance of the "l_1-opt" with various numbers of the selected bits across all splits of training and test datasets. As can be expected, the FPRs drop as the numbers of bits increase. The matching time for the RMGD_{104} is increased slightly at $45.34ns$, which is only a fraction of that of the SIFT (at $940ns$).

As showed theoretically in Section III.C, we derive the margin-based objective with the form of $loss + penalty$, Eq. 5. The μ_l is the tuning parameter that controls the tradeoff between loss and penalty. Two forms of penalty are used: the $ridge$ penalty with the idea of penalizing by the sum-of-squares of the parameters and the $lasso$ penalty stressed feature selection. In our experiments, we formulate the objective with $ridge$ penalty as a rankSVM problem by introducing the slack variable ξ (Eq. 6); while, the objective with $lasso$ penalty (Eq. 7) can be considered as one instance of the sparse support vector machines which can be solved with a non-smooth convex optimization approach, e.g. subgradient algorithm.

It is interesting to note that the performance of our grouped-feature optimization is considerably better than the direct combination, as shown in Fig. 6. The $ridge$ penalty shrinks the fitted coefficients W towards zero, and this shrinkage has the effect of controlling the variances of W which possibly improves the fitted model's prediction accuracy, especially when the feature maps are highly correlated [44]. In our method, we generate 13 feature maps from the original image, which may be strongly correlated to each other. That is the reason that feature optimization by Eq. 6 consistently outperforms the direct combination method, and the tendencies are significant in Fig. 6. The l_1 norm of W, i.e. the $lasso$ penalty, can be considered as a feature selection by inducing the sparsity. It corresponds to a double-exponential prior for the W, while the $ridge$ penalty corresponds to a Gaussian prior [44]. It is well known that the double-exponential density has heavier tails than the Gaussian density. Friedman et al. [56] showed the comparison results that the $lasso$ penalty works better than the $ridge$ penalty in the sparse scenario, similar results can be seen in our experiments (Fig. 6). Hence, we extend the sparse scenario to the 104 groups learning to achieve great performance improvements.

C. Comparisons with the state-of-the-art methods

We further compare performance of the RMGD against the state-of-the-art binary descriptors on the Brown datasets [27], including RFD [23], BinBoost [37], BGM [16], ITQ-SIFT [19], D-BRIEF [16], BRIEF [21] and BRISK [31]. Besides, comparisons with recent float descriptors, such as SIFT [10], Brown et al. [29] and Simonyan et al. [26] are also provided. For the RMGD, we report results of the 104-grouped optimization by the "l_1-opt". In order to reach a

fair comparison, we follow the protocol proposed in [29] by reporting the ROC curves and false positive rates at 95% recall. The experiments were conducted on the benchmark dataset (Local Patch Datasets [29]) which contains three subsets of patch pairs: the "liberty", "Notre Dame" and "Yosemite". We use crossing combinations of three subsets by training on one (with 500K pairs of local image patches attached in the datasets) and testing on one of the remained two. For the "l_1-opt", we only select a small number of groups (e.g. 50) for testing based on the numbers of non-zeros weights learned. The results are compared in Tab. II.

It can be found that the RMGD descriptor yields the best performance among all the binary descriptors listed. It outperforms the most closed one (the RFD) by a large margin with over 2% of the FPR in average. Furthermore, it is appealing that the RMGD also achieves competitive or even better results than recent float descriptors (Simonyan et al. [26]), which indicates that our binary method may narrow the performance gap between binary and floated-point descriptors.

V. APPLICATIONS

A. Image matching

We evaluate image matching performance of the RMGD on Oxford dataset [43] which contains six image sequences with different variations such as, 2D viewpoints (wall), compression artifacts (ubc), illumination changes (leuven), zoom and rotation (boat), and images blur (bikes and trees). Each image sequence is sorted in order of an increasing degree of distortions with respect to the first image corresponding to their changes. We followed the evaluation protocol of [47] to compare the descriptors. For each image, we compute 1,000 keypoints using oFAST [24] detector and then calculate their corresponding binary descriptors. The matching keypoints in two images are determined by nearest neighbor search. Since homography matrix between two images is given, the ground truth of correct matches can be estimated.

Fig. 9 illustrates the correct matching rates obtained by ORB-32 [30], BRISK-64 [31], SURF-64 [15], BinBoost-128 [37], RFD ($\text{RFD}_\mathbf{R}$, $\text{RFD}_\mathbf{G}$) and the proposed RMGD. For ORB-32 and SURF-64, we use the latest openCV implementation [23] [58]. And the implementations of the RFD, BRISK and BinBoost are available from the authors. In general, the RMGD_{104} achieves better performance than the other descriptors in all image sequences, and followed by the $\text{RFD}_\mathbf{R}$ and $\text{RFD}_\mathbf{G}$. These results are in consistency with our previous experiments, indicating that our learned descriptor RMGD can deal with various image variations effectively.

B. Object Recognition

We further evaluate the RMGD on object recognition task. Specially, we test them on two image recognition/retrieval benchmarks, the ZuBud dataset and Kentucky dataset [2]. The Kentucky dataset consists of object images with the resolution of 640×480 (see the middle of Fig. 10). It includes 255 indoor and outdoor objects in total. The ZuBuD dataset contains 1,005 images of Zurich building with 5 images for each of

TABLE III
COMPARISONS OF THE PROPOSED RMGD WITH THE STATE-OF-THE-ART BINARY AND FLOATED-POINT DESCRIPTORS WITH THE FPR @95%. THE
NUMBER OF BITS (B), OR DIMENSIONS (F), OR GROUPS AND SELECTED BITS FOR RMGD, ARE DESCRIBED IN PARENTHESES.

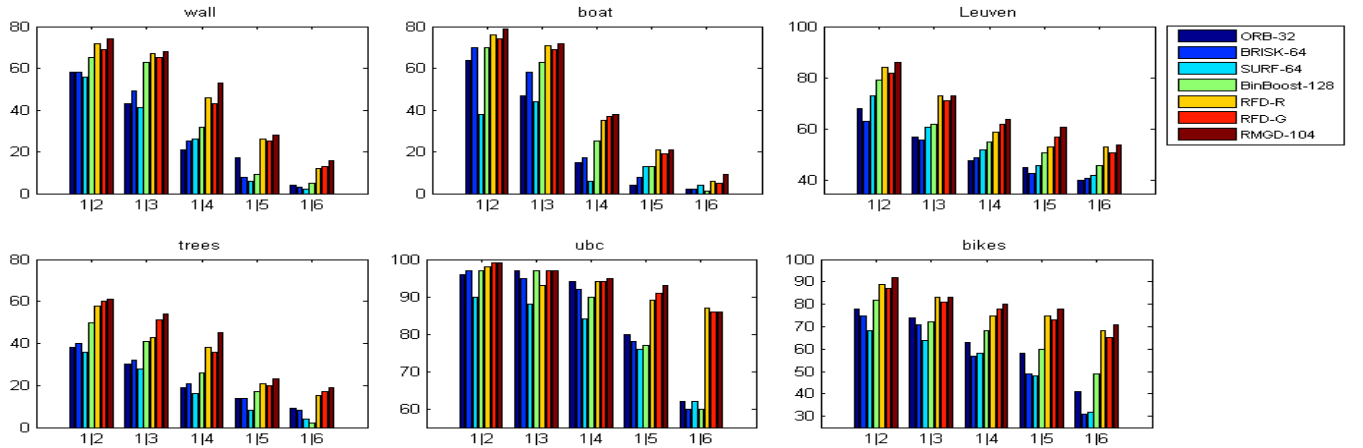| Train	Yosemite	Notre Dame	Yosemite	Liberty	Notre Dame	Liberty
Test	Liberty		Notre Dame		Yosemite	
The Binary Descriptors						
BRIEF [21]	54.01 (512b)		48.64 (512b)		52.69 (512b)	
BRISK [31]	79.36 (1024b)		74.88 (1024b)		73.21 (1024b)	
FREAK [32]	58.14 (512b)		50.62 (512b)		52.95 (512b)	
D-BRIEF [16]	53.39 (32b)	51.30 (32b)	43.96 (32b)	43.10 (32b)	46.22 (32b)	47.29 (32b)
ITQ-SIFT [19]	37.11 (64b)	36.95 (64b)	30.56 (64b)	31.07 (64b)	34.34 (64b)	34.43 (64b)
BGM [16]	22.18 (256b)	21.62 (256b)	14.69 (256b)	15.99 (256b)	18.42 (256b)	21.11 (256b)
SIFT-KSH [57]	44.87 (128b)	44.71 (128b)	35.73 (128b)	34.84 (128b)	37.59 (128b)	36.31(128b)
RFD$_G$ [23]	19.03 (563b)	17.77 (542b)	11.37 (563b)	12.49 (406b)	15.14 (542b)	17.62 (406b)
RFD$_R$ [23]	19.40 (598b)	19.35 (446b)	11.68 (598b)	13.23 (293b)	14.50 (446b)	16.99 (293b)
RMGD$_{104}$	**17.42(50x32b)**	**15.09(44x32b)**	**10.86(45x32b)**	**10.15(50x32b)**	**13.82(44x32b)**	**14.64(43x32b)**
The Floated-point Descriptors						
SIFT [10]	32.46 (128f)		26.44 (128f)		30.84 (128f)	
Brown et al. [29]	18.27(29f)	16.85(36f)	11.98(29f)	-	13.55(36f)	-
Simonyan et al. [26]	16.7(32f)	14.26(32f)	9.99(32f)	9.07(32f)	13.4(32f))	14.32(32f)

Fig. 9. Comparisons of the RMGD with recent binary descriptors on the matching accuracy of six image sequences from the "Oxford" dataset. $1|x$ denotes the matching pair between image 1 and image x, $x = 2, 3, 4, 5, 6$.

the 201 buildings. The sizes of the images are 640×480. Some examples are shown in the bottom of Fig. 10.

To compare the performance of our descriptor with existing results, we follow the same evaluation protocol in [23]. The DoG detector is adopted for extracting keypoints. Then the local descriptors are calculated, including SIFT [10], BGM [17], BinBoost [37], RFD [23] and RMGD$_{104}$. The implementation codes of these descriptors are from OpenCV or the authors. For each image we query its top 4 similar images for the ZuBuD dataset or 3 similar images for the Kentucky dataset. We report the ratios between the number of correctly retrieved images to the number of all returned images as an accuracy metric. Tab. IV summarizes the results of different local descriptors on two datasets. Again, the RMGD$_{104}$ obtains the best performance among all descriptors compared, and its improvements over the others are significant with about 3% higher than the most closed one (the RFD$_G$).

VI. CONCLUSION

We have presented a novel local binary descriptor (the RMGD) for image description. Our key contribution includes

TABLE IV
OBJECT RECOGNITION ACCURACY ON THE ZUBUD AND KENTUCKY.

	ZuBuD	Kentucky
SIFT [10]	75.5%	48.2%
BGM [17]	67.3%	36.3%
BinBoost-256 [37]	62.3%	19.2%
BRIEF [21]	70.5%	41.6%
FREAK [32]	48.8%	21.9%
SIFT-KSH [57]	64.6%	29.8%
RFD$_G$ [23]	82.5%	65.1%
RFD$_R$ [23]	80.7%	62.5%
RMGD$_{104}$	**85.4%**	**67.3%**

a novel pooling configuration, which generates meaningful binary strings from pairwise ring-regions with various shapes, scales and distances, and achieve compact representation by developing a new Adaboost based algorithm for fast bit selection with enhancements on variance and correlation. Furthermore, we showed the performance can be improved considerably by computing the binary strings from multiple image properties, and proposed two efficient learning algo-

(a) The Oxford dataset: *wall*, *trees* and *ubc*

(b) The Kentucky dataset with different viewpoints

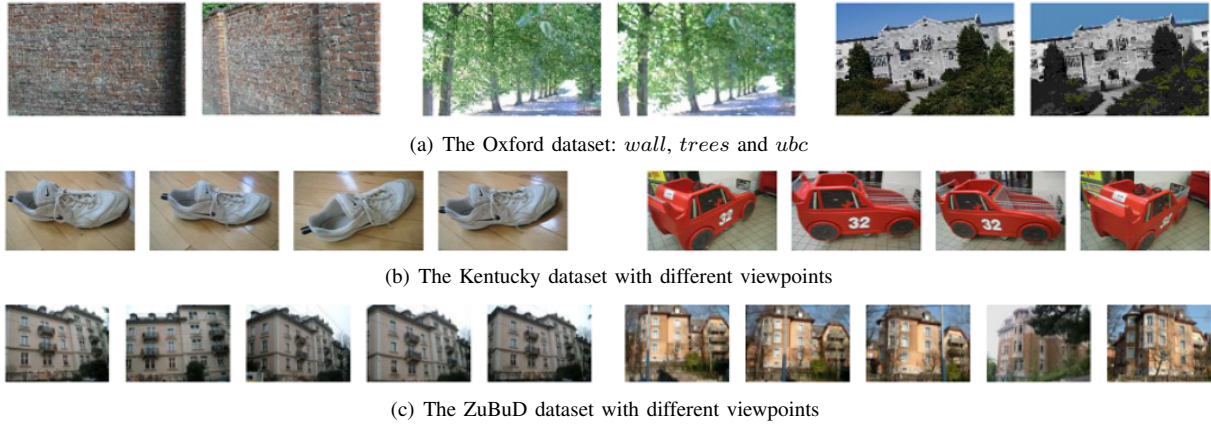(c) The ZuBuD dataset with different viewpoints

Fig. 10. Several examples from the Oxford, Kentucky and ZuBuD datasets.

rithms to effectively optimize multi-grouped binary features, allowing them for compensating for each other. This leads to a further performance boosting. Extensive experimental results on a number of benchmarks verify the effectiveness and usefulness of the RMGD convincingly by achieving significant performance improvements over current binary descriptors.

APPENDIX
CIRCLE INTEGRAL IMAGE

We propose the circle integral image for fast calculation of our binary descriptor. Integral image [41] has been proved to be highly effective in computing various low-level features. We generalize it to compute the ring features. Supposing the original point located in the center of a patch $\mathbf{x}$, the circle integral image can be defined as:

$$q(r', \theta') = \sum_{r \leq r'} \sum_{0 \leq \theta < 2\pi} i(r, \theta) + \sum_{0 \leq \theta \leq \theta'} i(r', \theta), \qquad (9)$$

where $i(r, \theta)$ is the intensity of the polar coordinate (r, θ), $r, r' \leq R$, $\theta, \theta' \in [0, 2\pi)$, and R denotes radius of the patch $\mathbf{x}$. Once the circle integral image is obtained, annular-
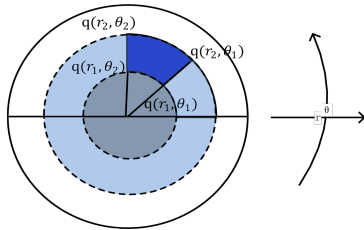

Fig. 11. Illustration of circle integral image. The sum of blue ring-region can be computed with four points located in the circle polar integral image.

sector can be calculated with $s = q(r_2, \theta_2) - q(r_2, \theta_1) - q(r_1, \theta_2) + q(r_1, \theta_1)$. As shown in Fig. 11, it speeds up the calculation of the summation of the pixel values over a ring-region substantially.

REFERENCES

[1] Y. Gao, J. H. Zhang, and L. Zhang, "Finding objects at indoor environment combined with depth information," in *Int. Conf. Mech. and Automat.*, 2011, pp. 687–692.

[2] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2161–2168.

[3] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 91.1–91.11.

[4] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, 2009.

[5] A. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 2037–2041, 2006.

[6] W. Huang and H. Yin, "Robust face recognition with structural binary gradient patterns," *arXiv:1506.00481*, 2015.

[7] Z. Li, D. Gong, X. Li, and D. Tao, "Learning compact feature descriptor and adaptive matching framework for face recognition," *IEEE Trans. Image Process.*, pp. 2736–2745, 2015.

[8] Z. Li, D. Gong, Y. Qiao, and D. Tao, "Common feature discriminant analysis for matching infrared face images to optical face images," *IEEE Trans. Image Process.*, pp. 2436–2445, 2014.

[9] W. Huang and H. Yin, "A dissimilarity kernel with local features for robust facial recognition," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 3785–3788.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[11] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 971–987, 2002.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[13] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classication," in *European Conference on Computer Vision (ECCV)*, 2006.

[14] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[16] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. IEEE Int. Conf. Eur. Conf. Comput. Vis.*, 2012, pp. 228–242.

[17] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Learning image descriptors with the boosting-trick," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 2874–2881.
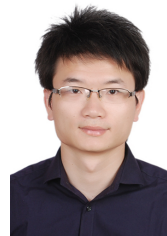
[18] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan 2012.

[19] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec 2013.

[20] W. Liu, J. Wang, and S. fu Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learn.*, June 2011, pp. 1–8.

[21] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a Local Binary Descriptor Very Fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, 2012.

[22] Y. Gao, Y. Qiao, Z. Li, and C. Xu, "Ltd: Local ternary descriptor for image matching," in *IEEE Int. Conf. Info. Autom.*, Aug 2013, pp. 1375–1380.

[23] B. Fan, Q. Kong, T. Trzcinski, Z. H. Wang, C. Pan, and P. Fua, "Receptive fields selectioni for binary feature description," *IEEE Trans. Image Process.*, pp. 2583–2595, 2014.

[24] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 597–610, March 2015.

[25] X. Yang and K. Cheng, "Local difference binary for ultrafast and distinctive feature description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 188–194, Jan 2014.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, Aug 2014.

[27] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors," in *Proc. IEEE Int. Conf. Comput*, Oct 2007, pp. 1–8.

[28] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2009, pp. 178–185.

[29] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan 2011.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput*, 2011, pp. 2564–2571.

[31] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput*, 2011, pp. 2548–2555.

[32] R. O. Alahi, Alexandre and P. Vandergheynst, "Freak: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2564–2571.

[33] X. Yang and K.-T. Cheng, "Ldb: An ultra-fast feature for scalable augmented reality on mobile devices," in *IEEE Int. Sym. Mixed Aug. Real.*, 2012, pp. 49–57.

[34] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *ECCLT*, 1995, pp. 23–37.

[35] B. Fan, Q. Kong, X. Yuan, Z. Wang, and C. Pan, "Learning weighted hamming distance for binary descriptors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013.

[36] J. Feang, W. Liu, and Y. Wang, "Learning to rank binary codes," in *arXiv:1410.5524 [cs.CV]*, 2014.

[37] L. V. Trzcinski T., Christoudias M. and P. Fua, "Boosting binary keypoint descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2874–2881.

[38] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, 2010.

[39] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2007, pp. 1–8.

[40] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," vol. 55, no. 1, Aug. 1997, pp. 119–139.

[41] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. 1511–1518.

[42] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Adv. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.

[43] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[44] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, "1-norm support vector machines," in *Adv. Neural Inf. Process. Syst.*, 2004, pp. 49–56.

[45] R. Tibshirani, "Regression shrinkage and selection via the lasso," in *J.R.S.S.B.*, 1996, pp. 267–288.

[46] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM Int. Conf. Knowl. Dis. Data Mining*, 2002, pp. 133–142.

[47] X. Wu, B. Xu, Y. Qiao, and X. Tang, "Automatic music video generation: cross matching of music and image." in *ACM Multimedia*. ACM, 2012, pp. 1381–1382.

[48] T. Joachims, "Training linear svms in linear time," in *Proc. ACM Int. Conf. Knowl. Dis. Data Mining*, 2006, pp. 217–226.

[49] ——, "A support vector method for multivariate performance measures," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 377–384.

[50] T. Joachims, T. Finley, and C.-N. Yu, "Cutting-plane training of structural svms," *Machine Learning Journal*, vol. 77, no. 1, pp. 27–59, 2009.

[51] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *J. Mach. Learn. Res.*, vol. 3, pp. 1229–1243, 2003.

[52] P. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 82–90.

[53] A. Cotter, S. Shalev-shwartz, and N. Srebro, "Learning optimally sparse support vector machines," in *Proc. Int. Conf. Mach. Learn.*, vol. 28, 2013, pp. 266–274.

[54] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," vol. 11, October 2010, pp. 2543–2596.

[55] S. Winder and M. Brown, "Learning local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2007, pp. 1–8.

[56] J. Friedman, T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "Discussion of boosting papers," *Ann. Statist*, pp. 102–107, July 2004.

[57] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2012, pp. 2074–2081.

[58] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Yongqiang Gao received the B.Sc. degree in School of Mathematics and Information sciences from Yantai University, Yantai, China, in 2009, the M.S. degree in School of Computer Science and Technology from University of South China, Hengyang, China, in 2012. He is currently pursuing the Ph.D. degree in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His current research interests include computer vision and machine learning.

Weilin Huang (M'13) received PhD degree in electronic engineering from the University of Manchester (UK) in December 2012. He got his BSc in computer science and MSc in internet computing from the University of Shandong (China) and University of Surrey (UK), respectively. Currently, he is working as a Research Assistant Professor at Chinese Academy of Science, and a joint member in the Multimedia Laboratory, Chinese University of Hong Kong. His research interests include computer vision, machine learning and pattern recognition. He has served as reviewers for several journals, such as IEEE Transactions on Image Processing, IEEE Transactions on Systems, Man, and Cybernetics (SMC)-Part B and Pattern Recognition. He is a member of IEEE.

Yu Qiao (SM'13) received the Ph.D. degree from the University of Electro-Communications, Japan, in 2006. He was a JSPS Fellow and Project Assistant Professor with the Unversity of Tokyo from 2007 to 2010. He is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include pattern recognition, computer vision, multi-media, image processing, and machine learning. He has published more than 90 papers. He received the Lu Jiaxi Young Researcher Award from the Chinese Academy of Sciences in 2012.