Learning multiple local binary descriptors for image matching

Yongqiang Gao,*, Weilin Huang, Yu Qiao

1. Introduction

Local feature descriptor is a fundamental topic in computer vision and image processing. The performance of many computer vision applications is highly relied on an informative and robust descriptor, such as image matching, object recognition, object detection, image classification, face recognition, text localization, and so on. However, it is still highly challenging to design robust and discriminative descriptors, due to the significant variations caused by real-world changes in lighting or illumination, un-fixed viewpoints and different image qualities (blurring, noise, and low resolution), images with the same scene or object may exhibit significant appearance differences.

Current local feature descriptors can be roughly divided into two types, the floating point and binary descriptors. The former makes use of floating type values for vector feature representation. The widely-used floating point descriptors includes Scale-Invariant Feature Transform (SIFT) [1], Histograms of Gradients (HoG) [9], and GLOH [10] descriptors. In the recent years, various binary descriptors have been proposed to cater to the applications in low power mobile devices and the demands of fast computation. Recent binary descriptors include Binary Robust Independent Elementary Features (BRIEF) [11], Discriminative BRIEF (D-BRIEF) [12], BinBoost [13], Binary Robust Invariant Scalable Keypoints (BRISK) [14], ORB [15], Local Difference Binary (LDB) [16], Boosted Gradient Maps (BGM) [12], Local Ternary descriptor (LTD) [17], Ring-based Multi-Grouped Descriptor (RMGD) [18] and the latest Receptive Fields Descriptor (RFD) [19]. In contrast to the floating point ones, the binary descriptors encode patch information using binary strings and hamming distance is applied for measuring the similarity between patches by using fast XOR operator.

With numerous advantages in low memory storage, fast computing and matching strategies, the binary descriptors have been attracting increasingly attention recently. Generally, there are two types of methods to compute the binary descriptors. The first approach is to explore the quantization [20, 21] and hashing
techniques [20–26] to binarize the existing floating point features. Obviously, the performance of this approach is significantly limited by the intermediate floating point representations. The second method is to involve binary tests by computing the intensity differences between pairs of selected pixels or defined regions, and some learning based methods have been developed to optimize the selection of binary tests [11–17,19,27,28]. For example, the BinBoost [13] and BGM [29] apply the boosting-trick to learn compact binary strings for non-linear visual representation. The Boosted Similarity Sensitive Coding (SSC) [30] is adopted to maximize the similarity between similar patches, and to minimize it between dissimilar patches simultaneously. The RFD [19] defines a set of receptive fields in multiple gradient channels for binary tests, and generates the compact binary strings by learning to binarize the responses of defined receptive fields.

Though the binary descriptors have great advantage in speed and memory storage, they often focus overall on compact representation in the cost of significant information loss, leading to less discriminative power. This can be substantiated by the results on the challenging Brown’s datasets [31–33] where the learning based floating point descriptor developed by Simonyan et al. [34] achieves the best performance, with a large margin over the performance of existing binary descriptors.

In this paper, we propose a novel learning framework to effectively integrate multiple successful binary descriptors with the learnt weights, in an effort to bridge the performance gap between current binary and floating point descriptors. We refer the new descriptor as learning-based multiple binary descriptors (LMBD).

Two types of binary descriptors are explored as the basic units: the boosting-trick based binary descriptors, including BinBoost [13] and BGM [12], and receptive fields based descriptors, such as the RFDQ and RFDG described in [19]. Each binary descriptor is considered as a feature group. We develop a margin-based ranking optimization method to learn the weights optimally for multiple groups by leveraging the rankSVM algorithm [35]. Our optimization technique is inspirit similar to that of [18], but here we use different types of binary descriptors. We observe that the boosting-trick and receptive fields based descriptors are capable of capturing different characteristics of features, which can compensate strongly for each other, leading to a significant improvement on the discriminative power. The proposed LMBD was evaluated on the challenging Brown’s datasets [33]. It achieves excellent results which are comparable or even better than the state-of-the-art results achieved by the floating descriptor [34]. We also evaluated the LMBD descriptor on image matching task using the Mikolajczyk datasets [10]. The experimental results show that the LMBD outperforms other binary descriptors considerably for image matching.

The rest of the paper is organized as follows. In the next section, we present the details of our basic framework for multiple binary descriptors learning, and provide an efficient algorithm for solving it optimally. Experimental results on patch and image matching are reported in Section 3, with comparisons against recent existing binary and floating point descriptors. Section 4 concludes this manuscript.

2. Learning-based multiple binary descriptors

In this section, we present the details of our framework for learning multiple binary descriptors, including the boosting-trick and receptive fields based descriptors. Then a rankSVM algorithm is presented to solve the learning problem effectively.

2.1. Basic single binary descriptors

Given an image intensity patch x, a local descriptor \(C(x) = [C_1(x), \ldots, C_D(x)]\) maps the patch to a D-dimensional vector. For a binary descriptor, \(C_i(\cdot)\) denotes a binary function or a binary test. Various binary descriptors are different in computing the binary tests, C.

2.1.1. Boosting-trick binary descriptors

Both BinBoost [13] and BGM [29] learn compact binary strings using the boosting-trick. They apply the boosting to learn complex non-linear local binary representations. The employed weak learner family is capable of encoding specific design choices and meaningful descriptor properties. First, both methods construct a feature pooling \(H(x) = [h_i]_M^1\), which a collection of thresholded non-linear response functions of a intensity patch, \(h_i(x) \in [-1, 1]\). The size of the feature pooling \(M\) is generally large or possibly infinite. Then the compact binary strings and the discriminative mapping functions are learnt by minimizing the exponential loss of a defined similarity function \(f(m, n)\) over a set of image patch pairs [13]

\[
\mathcal{L} = \sum_{i=1}^{N} \exp \left( - l_i f(m_i, n_i) \right).
\]

where \(m_i, n_i\) are a pair of intensity patches, and \(l_i \in \{-1, 1\}\) is a label indicating whether it is a similar (1) or dissimilar (-1) pair. The Boosted Similarity Sensitive Coding (SSC) algorithm [30] is adopted. It defines a similarity function by using a simply weighted sum of the thresholded response functions

\[
f(x, y) = \sum_{i=1}^{M} a_i h_i(x) h_i(y).
\]

which defines a weighted hash function with the importance of each dimension \(i\) given by \(a_i\). The minimization of Eq. (1) aims to find an embedding that maximizes the similarity between similar patches, and at the same time, minimizes it between the dissimilar patches.

Computing the responses of the binary tests in multiple gradient domains leads to gradient-based BinBoost and BGM, which are different in the choice of weak learners. Each BinBoost is computed as a linear combination of many gradient orientation maps, while each BGM is constructed by a weak learner. The gradient-based weak learners applied by two descriptors are defined as

\[
h(x; R, c, T) = \begin{cases} 
1 & \text{if } \phi_{R,c}(x) \leq T \\
-1 & \text{otherwise} 
\end{cases}
\]

where

\[
\phi_{R,c}(x) = \sum_{m=0}^{R} \xi_c(x, m) / \sum_{c, \phi, m=0}^{R} \xi_{\phi}(x, m).
\]

where region \(\xi_c(x, m)\) is the gradient energy along an orientation \(e\) at location \(m\), and \(R\) defines a rectangular extent within the patch \(x\).

2.1.2. Receptive fields’ binary descriptors

The RFD [19] is computed by thresholding the responses of a set of defined receptive fields. It defines a feature pooling, which may also be large or possibly infinite. Specifically, the is computed in three steps: primary feature extraction, receptive fields pooling and binarization. Firstly, a patch is mapped into 8 feature channels with the same size as the patch, by soft assigning the gradient orientation of each pixel into 8 orientated bins: \([0.1 \times \pi/8, 2 \times \pi/8, \ldots, 7 \times \pi/8]\), each of which corresponds to a feature channel. Secondly, the floating-point responses of the receptive fields defined within a geometric area are calculated on each
feature channel. Two types of candidate receptive fields are designed in RFD [19]: rectangle receptive field

\[ g_R(x, y) = \sum_{x \in \text{Rect}(P_1, P_2)} F_i(x, X) / Z_R \]

(5)

and Gaussian receptive field

\[ g_G(x, y) = \sum_{x} \frac{1}{\sqrt{2\pi} \sigma^2} e^{-\frac{(x-c)^2}{2\sigma^2}} F_i(x, X) / Z_G \]

(6)

which are referred as RFDq and RDFG, respectively. \( X \) denotes the coordinate of a pixel in the patch. \( \text{Rect}(P_1, P_2) \) is the rectangular area defined by \( P_1 \) and \( P_2 \). \( F_i(x, X) \) is the feature value of pixel \( X \) at channel \( c \). \( Z_R \) and \( Z_G \) are normalization factors used to eliminate the influence of different gradient magnitudes. Finally, the responses of receptive fields are binarized by a learnt threshold. The response binarization is performed by thresholding \( g(x, y) \) at a learned value \( t \)

\[ h(x, y) = \text{sign}(g(x, y) - t). \]

(7)

where we assume \( \text{sign}() \in \{0,1\} \). The threshold \( t \) is optimized to give the best recognition rates.

It is well known that the central part of a given patch often includes more important information than those far from it. This observation can be verified in RFD, where Gaussian similarity function is adopted in RDFG. The response of a Gaussian receptive field is different from that of the rectangle receptive field.

2.2. Learning based multiple binary descriptors

Given an image patch \( x \), we define the \( m \)th binary string as \( B^m(x) \in \{0,1\}^d_m \), where \( d_m \) indicates the dimensionality of the \( B^m \) in hamming space \( \mathcal{H} \), \( m \in \{1, \ldots, M\} \). Each binary string is computed from an applied binary descriptor, and is referred as a group of bits. The multiple combined-group binary descriptor is expressed as \( [w_1B^1, w_2B^2, \ldots, w_MB^M] \), where the Learning based Multiple Binary Descriptors (LMBD) optimizes the weight of each group \( w_i \).

Suppose that \( \mathcal{P} \) and \( \mathcal{N} \) are the training datasets which contain similar and dissimilar pairs, respectively. What we expect to obtain a distance space in which the distances of matching pairs are less than those of non-matching pairs. It is noted that descriptor matching problem is usually obtained by searching the nearest neighbor in the descriptor space for the given descriptor, and the distance ranking is much more significant. To be specific, the LMBD can be learnt such that the matching and non-matching pairs are separated by a margin in descriptor spaces. We formulate the optimization of the LMBD as the following margin-based constraints on the hamming distance in the descriptor space

\[ d_w(x, y) + c < d_w(u, v). \]

\( \forall (x, y) \in \mathcal{P}, \forall (u, v) \in \mathcal{N} \]

(8)

where \( d_w(x, y) \) is the distance between patch \( x \) and \( y \).

\[ d_w(x, y) = \sum_{m=1}^{M} w_m \times \text{Ham}_m(x, y) = w^T \text{Ham}(x, y) \]

(9)

and the Hamming distance of the \( m \)th group between \( x \) and \( y \) is computed as

\[ \text{Ham}_m(x, y) = \sum_{d=1}^{d_m} B^m_d \otimes B^m_d \]

(10)

where \( \otimes \) denotes bitwise XOR operation. \( \mathbf{w} = [w_1, \ldots, w_M]^T \), \( \text{Ham}(x, y) = [\text{Ham}_1(x, y), \ldots, \text{Ham}_M(x, y)]^T \), and \( c \) is a constant.

To this end, we cast the weight learning problem as a minimization of a defined empirical loss with a soft constraint. The empirical loss is computed as the weighted Hamming distance between matching and non-matching pairs over a training set

\[ \min_{\mathbf{w} \neq \mathbf{0}} \sum_{(x,y) \in \mathcal{P}(x,v) \in \mathcal{N}} \mathcal{L}(\mathbf{w}^T (\text{Ham}(x, y) - \text{Ham}(u, v))) + \mu \mathbf{w} \]

(11)

where \( \mathcal{L}(z) = \max(z + c, 0) \) is the hinge loss function. \( \mathbf{R}(\mathbf{w}) \) is a regularization term to prevent overfitting and \( \mu \) is a parameter trading-off between the empirical loss and regularization. The aim is to select the strongest complementary groups, we use \( L_1 \) norm for the regularization in our case, i.e., \( \mathbf{R}(\mathbf{w}) = ||\mathbf{w}||_1 \). This optimization problem is differentiable, and thus Eq. (11) can be solved by the subgradient descent algorithm.

The formulation Eq. (11) can be seen as an instance of a pairwise RankSVM problem \([18,35-37]\) where two classes are separated by a margin. However, our object function is penalized by the \( L_1 \) regularization, which makes it different from conventional RankSVM \([35]\). The \( L_1 \) norm \( ||\mathbf{w}||_1 \) is sparsity-induced, which encourages the weights of irrelevant groups to be zero, while setting large weights to the groups with strong complementarity.

Our optimization problem involves training a large number of image pairs (e.g., 500,000 iterations in our experiment), where the conventional interior point methods is infeasible. Xiao [38] proposed a Regularized Dual Averaging (RDA), which aims to formulate a typical objective function into an online setting. This objective function often contains the sum of two convex terms: a loss function of the learning task and a regularization term. In our case, the second regularization term is the “soft” \( L_1 \)-regularization, \( \mu ||\mathbf{w}||_1 \), following the pipeline of the RDA. The auxiliary function \( h(w) = \frac{1}{2} ||\mathbf{w}||^2 \), which is strongly convex, is considered. Similar techniques have been exploited for learning binary descriptors in \([18]\). Given a nonnegative and nondecreasing sequence \( \mathbf{g} = g / T (t \) denotes the iteration), the specific form of the RDA update term for Eq. (11) is obtained

\[ w_{m_t+1} = \max \left\{ 0, \frac{\sqrt{t}}{\gamma} (g_m + \mu) \right\}. \]

(12)

where \( g = \frac{1}{T} \sum_{t=1}^{T} g_t \) is the average sub-gradient of the corresponding hinge loss function at iteration \( t \), \( \mu \) is the parameter in Eq. (11). And the results can be fine-tuned by running the optimization with different values of \( \mu \).

Considering low memory storage and simple computation, the floating weights \( \mathbf{w} \) are converted to integer data by

\[ w_m = 256 \times \frac{w_m}{\sqrt{\sum_{m=1}^{M} w^2_m}} \]

and

\[ w_m = \text{round}(w_m). \]

(13)

It ensures that each weight \( w_m \) is less than 256 which is the maximum value that one byte (256) can represent. \( \text{round}(w_m) \) denotes the rounded value of \( w_m \).

3. Experimental results

In this section, we present experimental results of the proposed LMBD, and compare them with most recent results on the challenging Brown’s \([33]\) and Mikolajczyk \([10]\) datasets for patch/image matching 1.

3.1. Experiment setup

The challenging Brown’s dataset \([31,33]\) with three widely-used subsets, Liberty, Yosemite and NotreDame, is adopted in our experiments (Fig. 1). Each subset contains over 400 k scale- and rotation-normalized 64 \times 64 image patches. All of the local patches are de-

1 More details can be found at http://mmlab.siat.ac.cn/ygao/com8/.
tected as Difference of Gaussian (DoG) maxima or multi-scale Harris corners. The ground truth is available for each of these datasets which include 100 k, 200 k, and 500 k pairs of patches, 50% of them correspond to match pairs, and 50% to non-match pairs. We report the results of the evaluation in terms of ROC curves and false positive rate at 95% recall (FPR @ 95%) which is the percent of incorrect matches obtained when 95% of the true matches are found on the 100 k datasets, by following [31] and [13].

Four binary descriptors are used as the basic descriptors for learning the LMBD. They are the BGM, BinBoost \(^2\), RFD\(_C\) and RFD\(_R\) \(^3\), which are described briefly in Section II. Codes for computing these binary descriptors are available from the corresponding authors. To be specific, the BinBoost and BGM use 256 bits binary descriptors on the 32 × 32 local patches, while the RFD\(_C\) and RFD apply 406 bits and 293 bits for describing the local patches of size 64 × 64. In our experiments, we used the subset of 500 k pairs, including 250 k matching and 250 k non-matching pairs, as training data for learning the weights \(\mathbf{w}\) in Eq. (11).

2 http://cvlab.epfl.ch/research/detect/binboost.

The performance of the proposed LMBD and four basic binary descriptors (the BinBoost, BGM, RFD and RFD) are compared in Table 1. By following the evaluation protocol in [19,33], the 100 k subsets of “Notre Dame” and “Yosemite” are used for the test, and

<table>
<thead>
<tr>
<th>Test dataset</th>
<th>groups</th>
<th>64(b)</th>
<th>128(b)</th>
<th>256(b)</th>
<th>Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notre Dame (%)</td>
<td>BinBoost</td>
<td>30.66</td>
<td>19.98</td>
<td>19.52</td>
<td>19.52</td>
</tr>
<tr>
<td>BGM</td>
<td>27.01</td>
<td>21.06</td>
<td>18.43</td>
<td>18.43</td>
<td></td>
</tr>
<tr>
<td>RFD(_R)</td>
<td>21.50</td>
<td>16.83</td>
<td>13.02</td>
<td>12.04</td>
<td></td>
</tr>
<tr>
<td>RFD(_C)</td>
<td>24.30</td>
<td>18.95</td>
<td>12.70</td>
<td>12.24</td>
<td></td>
</tr>
<tr>
<td>LMBD</td>
<td>15.96</td>
<td>12.65</td>
<td><strong>9.86</strong></td>
<td><strong>9.52</strong></td>
<td></td>
</tr>
</tbody>
</table>

| Yosemite (%) | BinBoost | 31.72 | 26.31 | 24.64 | 24.64 |
| BGM | 29.18 | 23.23 | 22.03 | 22.03 |
| RFD\(_R\) | 26.08 | 21.11 | 16.25 | 15.12 |
| RFD\(_C\) | 28.14 | 19.74 | 16.30 | 16.66 |
| LMBD | **18.66** | **15.24** | **12.64** | **12.44** |
false positive rates at 95% recall are reported. Four groups of experiments were conducted by using various number of bits of the basic descriptors: 64, 128, 256 and the optimized bit numbers provided by the original authors (Opt.). As shown, the proposed LMBD descriptor achieves excellent results in all groups by improving the best performance of basic descriptors at least about 3% in most cases. This indicates that four basic descriptors include important complementary information to each other and the proposed learning framework is highly efficient to learn them. In our experiments, we suggest that 128 bits for each basic binary descriptor is sufficient to achieve reasonable performance in practice.

We further compare the performance of the LMBD against the state-of-the-art results of existing binary descriptors. Fig. 2 shows the comparative results of our proposed LMBD descriptor and state-of-the-art binary descriptors. The “Liberty” dataset is used for training, and we test on the other two datasets (100K). In Table 2, we follow the protocol in previous work [33], and report the false positive rate at 95% recall. The number of binary dimensions (bits) for each descriptor is also presented. The best performance are **bolded**, and the best performance of the floating point and binary descriptors are indicated by an asterisk and a #, respectively. Again, the results show that the proposed LMBD descriptor obtains the best performance among all compared binary descriptors in both test sets. Its improvements over the other binary descriptors are significant, with about 3% and 5% over the closest results in the “Notre Dame” and “Yosemite” subsets respectively. Furthermore, the LMBD even outperforms the recent proposed floating point descriptor, L-BGM [29] substantially, with about 5% and 7% improvements on two tests. Our results nearly match the best results on these datasets, which are achieved by a powerful learning based floating point descriptor developed by Simonyan et al. [34].

### 3.2. Image matching

The proposed LMBD was further evaluated on the task of image matching. Experiments were conducted on the Mikolajczyk dataset [10], which is designed to investigate the robustness to 2D viewpoints (wall), compression artifacts (ubc), illumination changes (leuven), zoom and rotation (boat) and blur images (bikes and trees). Each dataset contains six image sequences sorted by an increasing degree of distortions with respect to the first image.

We followed the evaluation protocol described in [10,11,16,17]: (a) Pick \(n_1, n_2\) interest points from two images and get the \(n\) matching pairs estimated by homography matrix; (b) For each point in the first set, find the nearest neighbor in the second one by different descriptors; (c) Count the number of correct matches \(n_c\) and obtain the recognition rate with \(r = n_c/n\). In our experiments, Hessian–Affine detector is adopted for extracting region of interest

![Fig. 3. Mikolajczyk datasets used for image matching, it contains six image sequences with different variations.](http://www.robots.ox.ac.uk/~vgg/research/affine/).
and the principal orientation is calculated 5, and then it is normalized to the provided size for each descriptor.

Fig. 4 illustrates the recognition rate achieved by ORB-32, SURF-64, BinBoost-256, BGM-256, RFD$_R$, RFD$_G$ and the proposed LMBD descriptor, which is learning from the BinBoost with 256 bits, BGM with 256 bits, RFD$_R$ with 293 bits and RFD$_G$ with 406 bits. The optimized bit numbers for these descriptors are provided by the original authors. For ORB-32 and SURF-64, we use the latest openCV [39] and the implementations of RFD, BGM and BinBoost are available from the authors. As can be found, the LMBD achieves highest performance among all descriptors in all image sequences, following by the RFD$_R$ and RFD$_G$. These results further verify the efficiency of the LMBD.

4. Conclusion

We have presented a novel learning framework to efficiently learn the important complementary information from multiple binary descriptors. We cast the multi-groups learning as a margin-based constraints optimization problem with $L_1$-regularization and solved it effectively in the rankSVM framework. Four high-performance binary descriptors, BinBoost, BGM, RFD$_R$ and RFD$_G$, were used as basic descriptors for learning. We show experimentally that the learned LMBD descriptor is highly discriminative and achieves significant improvements over the current results of existing binary descriptors on two benchmark datasets. Furthermore, our results are also comparable against the best results achieved by the floating point descriptors [34] on the Brown's

5 http://www.robots.ox.ac.uk/~vgg/research/affine/descriptors.html.
database. These excellent results verify the efficiency of our descriptor convincingly.

Acknowledgment

This work is partly supported by National High-Tech Research and Development Program of China (2015AA042303) and Guangdong Reserarch Program (2014B050505017,2015B010129013).

References

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. And he obtained Ph.D. degree in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences in 2015. He now is working Tencent company. His current research interests include computer vision machine learning.

Weilin Huang, received Ph.D degree in electronic engineering from the University of Manchester (UK) in 2012. He got his B.Sc. in computer science and M.Sc. 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 published about 10 papers in peer-reviewed conferences and journals, and a PC member or reviewer of ICCV, CVPR, ECCV, AAAI, IJCAI etc. He is a member of IEEE. (SMC)-Part B and Pattern Recognition. He is a member of IEEE.

Yu Qiao (S’05–M’07) 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 University 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, multimedia, image processing, and machine learning. He has published more than 90 papers. He received the Lu Ji-axi Young Researcher Award from the Chinese Academy of Sciences in 2012.