OSRI: A Rotationally Invariant Binary Descriptor

Xianwei Xu, Lu Tian, Jianjiang Feng, *Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

Abstract—**Binary descriptors are becoming widely used in computer vision field because of their high matching efficiency and low memory requirements. Since conventional approaches, which first compute a floating-point descriptor then binarize it, are computationally expensive, some recent efforts have focused on directly computing binary descriptors from local image patches. Although these binary descriptors enable a significant speedup in processing time, their performances usually drop a lot due to orientation estimation errors and limited description abilities. To address these issues, we propose a novel binary descriptor based on the ordinal and spatial information of regional invariants (OSRIs) over a rotation invariant sampling pattern. Our main contributions are twofold: 1) each bit in OSRI is computed based on difference tests of regional invariants over pairwise sampling-regions instead of difference tests of pixel intensities commonly used in existing binary descriptors, which can significantly enhance the discriminative ability and 2) rotation and illumination changes are handled well by ordering pixels according to their intensities and gradient orientations, meanwhile, which is also more reliable than those methods that resort to a reference orientation for rotation invariance. Besides, a statistical analysis of discriminative abilities of different parts in the descriptor is conducted to design a cascade filter which can reject nonmatching descriptors at early stages by comparing just a small portion of the whole descriptor, further reducing the matching time. Extensive experiments on four challenging data sets (Oxford, 53 Objects, ZuBuD, and Kentucky) show that OSRI significantly outperforms two state-of-the-art binary descriptors (FREAK and ORB). The matching performance of OSRI with only 512 bits is also better than the well-known floating-point descriptor SIFT (4K bits) and is comparable with the state-of-the-art floating-point descriptor MROGH (6K bits), while it is two orders of magnitude faster to match than SIFT and MROGH.**

Index Terms—**Binary descriptor, rotation invariant, local order pattern, real-time matching, feature matching.**

I. Introduction

ESTABLISHING visual correspondences based on feature point descriptors is an essential component of many computer vision applications, such as image localization [1], 3D reconstruction from photo-collections [2], [3], large-scale partial-duplicate visual search [4], object recognition [5], [6], and panorama stitching [7]. Considering the fast developments of image acquisition devices and Internet/wireless network, these applications have to handle explosively increasing data or run on mobile devices with limited computational capabilities and storage space. This further necessitates that local descriptors should be discriminative, efficient, and compact.

As is well known to the computer vision community, the floating-point descriptor SIFT [8] and similar methods [9]–[12] have been widely accepted as the highest quality descriptors until now, with high distinctiveness and invariance to a variety of common image transformations. However, they still face drawbacks in terms of computation time, memory usage and matching efficiency, especially for large-scale or real-time applications. Consequently, there have been many recent attempts at compacting these floating-point descriptors to overcome these defects to a certain extent, which can be grouped into three categories: dimensionality reduction [9], [13], quantization [14]–[20], binarization [21]–[27]. Even though these approaches can improve the efficiency of storage and matching to various degrees, they all need first to compute the original descriptor then to shorten it, and generally require a training phase and/or a complex optimization scheme [18], [24]. The whole process of adopting the above approaches costs a massive amount of time-consuming computation. Note that these descriptors are still floating-point in nature even after being quantized or dimensionally reduced rather than binarized, and thus cannot be benefited from extremely fast similarity computation using the Hamming distance. In addition, all three classes of compacting techniques often result in matching performance degradation because they are lossy compression of the original floating-point descriptor.

To address the shortcomings of floating-point descriptors, recent works have primarily focused on directly computing binary descriptors from local image patches which require less storage and enable faster processing. BRIEF [28], ORB [29], BRISK [30], and FREAK [31] are good examples. Although these binary descriptors are highly efficient, their matching performance is still not comparable with best floating point descriptors. The main reasons can be summarized as:

- Limited distinctiveness. These binary descriptors are usually built upon a set of pairwise intensity comparisons where each sample point represents either a single pixel (e.g. BRIEF, ORB) or a Gaussian blurring of its surrounding pixels (e.g. BRISK, FREAK). However, this design is very sensitive to small disturbance to locations of sample points. Additionally, pairwise intensity comparisons capture very limited information of a local image region.
- Unreliable reference orientation. All these binary descriptors rely on a reference orientation estimated from the local region to achieve rotation invariance. It is very

Manuscript received September 7, 2013; revised December 25, 2013 and March 2, 2014; accepted April 29, 2014. Date of publication May 16, 2014; date of current version June 3, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61225008, Grant 61020106004, and Grant 61373074, in part by the National Basic Research Program of China under Grant 2014CB349304, in part by the Ministry of Education of China under Grant 20120002110033, and in part by the Tsinghua University Initiative Scientific Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitrios Tzovaras.

The authors are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xuxw09@mails.tsinghua.edu.cn; ltian1990@gmail.com; jfeng@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2324824

difficult to estimate the reliable reference orientation, especially under illumination changes, while the unreliable reference orientation is particularly harmful for the simple sampling schemes used in these binary descriptors (see Section V-A).

In this paper, a robust binary description framework is proposed to deal with the aforementioned problems, which has two essential differences comparing with the existing methods. Firstly, pairwise irregular subregions, generated by region division according to the orders of intensity and gradient orientation of pixels in one or more support regions, are taken as sampling units. Secondly, binary bits are computed by comparing pairwise regional invariants that represent appearance, shape and spatial geometry properties of subregions. This framework can capture more discriminative information of a local image region for feature description. Meanwhile, it is rotation-invariant without resorting to a reference orientation, and is also robust to monotonic illumination changes. Moreover, to obtain a compact descriptor, a learning method is used to select best bits from the raw binary string, leading to the better matching performance and the lower storage cost. The selected bits are further organized as a cascade filter so that non-matching descriptors can be rejected at early stages by comparing just a small portion of the whole descriptor, further reducing the matching time.

The proposed binary descriptor is termed as OSRI, namely, an abbriviatioin of Ordinal and Spatial information of Regional Invariants. Extensive experiments on four challenging data sets (Oxford, 53 Objects, ZuBuD, and Kentucky databases) show that OSRI outperforms the recent binary descriptors (ORB [29] and FREAK [31]) in terms of matching performance and efficiency. Compared with the existing state-of-the-art floating-point descriptors (SIFT [8] and MROGH [12]), OSRI has also better or comparable matching performance with significantly lower computational and storage complexity.

The rest of this paper is organized as follows. Section II gives a brief overview of the related works. Our proposed descriptor is elaborated in Section III. Analysis of properties of our descriptor are reported in Section IV. Section V presents the comparison of our descriptor against the state-of-the-art methods. Finally, we conclude this paper in Section VI.

II. Related Work

At present, there are two main classes of approaches toward building a binary descriptor.

A. Binarizing Floating-Point Descriptors

In view of high performance of some floating-point descriptors (e.g. SIFT [8], DAISY [11] and MROGH [12]), many recent efforts attempted to encode the robust descriptors into compact binary codes by resorting to hashing techniques. Locality-sensitive hashing (LSH) technique [32] and its variants [33], [34] are frequently used to find efficient binary representations of high-dimensional floating-point vectors maintaining their similarity in the new space, such as [21], [22], [35], [36]. These approaches are realized by first multiplying description vectors by a projection matrix and then thresholding the vectors to binary strings. Moreover, Linear Discriminant Analysis [24], K-means Hashing [25], Random Forest Hashing [26], Bilinear Projections [27], Nonlinear Neighborhood Component Analysis [37], Iterative Quantization [38], and Hamming Embedding [19] are also used for binarizing a floating point descriptor. However, the appropriate choice of hashing function is less well understood, whilst all these approaches are computationally expensive because the original floating-point descriptor must be computed before the hashing can occur, generally accompanying matching performance degradation due to the limitation of their lossy compression.

B. Directly Computing Binary Descriptors

As the necessity of first computing the full descriptor before further binarizing is a bottleneck for many large-scale or real-time applications, some researchers have paid increasing attention to directly computing binary descriptors from local image patches.

Calonder et al. [28] presented a simple method to directly build a binary descriptor (BRIEF) in which each bit is independently obtained by comparing the intensities of a pair of sample points. Notwithstanding with the lowest requirements for computation and storage, BRIEF has the notable defect in lack of rotation invariance. Therefore, Rublee et al. [29] proposed the Oriented Fast and Rotated BRIEF (ORB) descriptor, which is invariant to rotation changes and robust to noise. Meanwhile, ORB chooses a good subset of binary tests by a learning method that reduces correlation among the binary tests, improving the performance and scalability. Leutenegger et al. [30] also developed a binary descriptor called BRISK that is invariant to scale and rotation transformations. It turns away from the random sampling pattern of BRIEF, instead, and uses a symmetric sampling pattern in which each sample point represents a Gaussian blurring of its surrounding pixels. To gain more compact and robust performance, Alahi et al. [31] proposed a descriptor inspired by the human visual system recently, called Fast Retina Keypoint (FREAK). A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern.

We found that all the binary descriptors are built upon a set of pairwise intensity comparisons and resort to a reference orientation for rotation invariance. On one hand, pairwise intensity comparison is very sensitive to localization errors of sample points. On the other hand, dependence on a reference orientation can degrade matching performance (see Section V-A). As a result, these binary descriptors show lower description power than their floating-point competitors (e.g. SIFT, MROGH).

Recently, there are also some works to directly learn a compact descriptor from a local image patch [39]–[41]. However, the performance of those approaches depends on the training data sets, whilst those descriptors still require a reference orientation for rotation invariance.

III. Method

Suppose interest/support regions for feature description have been detected based on SIFT detector [8] or

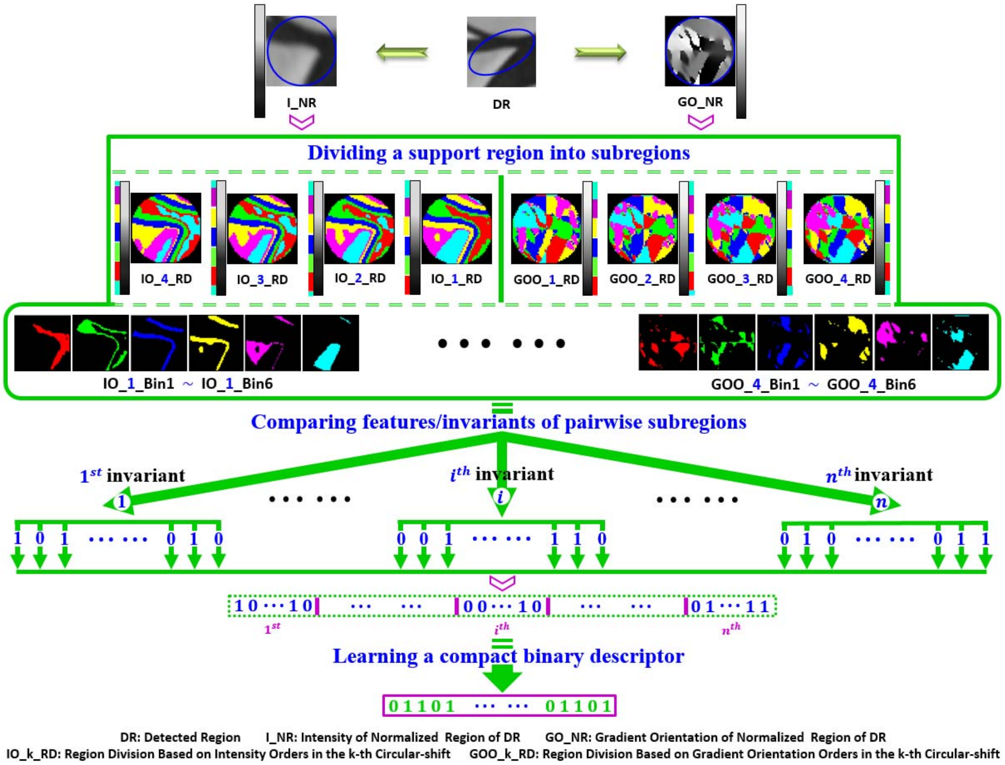

I_NR
DR
GO_NR
Dividing a support region into subregions
IO_4_RD
IO_3_RD
IO_2_RD
IO_1_RD
GOO_1_RD
GOO_2_RD
GOO_3_RD
GOO_4_RD
IO_1_Bin1 ~ IO_1_Bin6
GOO_4_Bin1 ~ GOO_4_Bin6
Comparing features/invariants of pairwise subregions
1st invariant
ith invariant
nth invariant
1 0 1 ⋯ ⋯ 0 1 0
0 0 1 ⋯ ⋯ 1 1 0
0 1 0 ⋯ ⋯ 0 1 1
10⋯10 ⋯ ⋯ 00⋯10 ⋯ ⋯ 01⋯11
1st
ith
nth
Learning a compact binary descriptor
01101 ⋯ ⋯ 01101
DR: Detected Region
I_NR: Intensity of Normalized Region of DR
GO_NR: Gradient Orientation of Normalized Region of DR
IO_k_RD: Region Division Based on Intensity Orders in the k-th Circular-shift
GOO_k_RD: Region Division Based on Gradient Orientation Orders in the k-th Circular-shift

Fig. 1. The workflow of constructing OSRI descriptor at the offline learning stage. At the online testing stage, only the compact binary descriptor needs to be computed.

Hessian/Harris-Affine detector [42]. Since the detected regions usually have different sizes and shapes, they are normalized to circular regions of a fixed radius for computing description vectors (please see [9] for normalizing interest regions).

The workflow of constructing the proposed OSRI descriptor is shown in Fig. 1, which involves four key steps: 1) rotation invariant sampling design for dividing a support region into a set of subregions, 2) binary description based on comparing high-level invariants in pairwise subregions, 3) learning compact binary codes from the lengthy bit-vector, and 4) cascade filtering design for speeding up matching. In the rest of this section, four steps will be described in details.

A. Sampling Pattern Based on Region Division

In the process of building OSRI, the first step is to produce many regional sampling-units (pairwise irregular subregions) for computing binary bits. These subregions should not only be rotation invariant without resorting to a reference orientation, but also contain rich information of appearance, shape and spatial geometry. In view of this, we propose an improved region dividing strategy based on the method in [12] with two main differences. Firstly, we use two types of information with complementary properties (intensity orders and gradient direction orders) for region division, which is more robust to noise while containing more appearance, shape, and spatial characteristics than by using either of two types of information separately (see Fig. 2). Secondly, we utilize circular-shift operation to group all pixels in different ways, which enables us to obtain more sampling units in a given support region for feature description. Moreover, our sampling pattern is also robust to monotonic illumination changes. Fig. 3 provides an illustration of our region division method in which each group/subregion is marked with a different color.

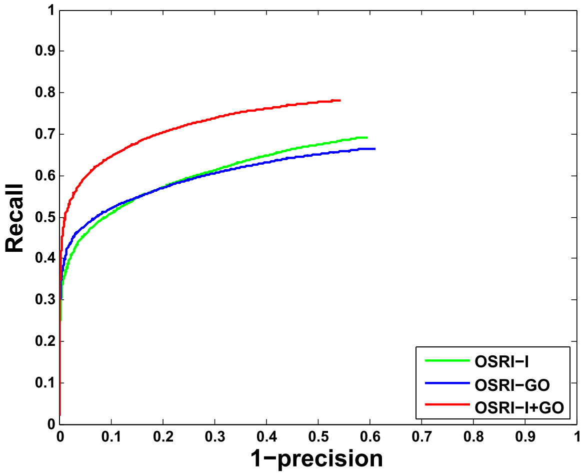


Fig. 2. The average performance of the proposed OSRI descriptor with three different methods of region division on the Oxford data set. OSRI-I, OSRI-GO, and OSRI-I+GO denote the corresponding OSRI with region division based on intensity orders, gradient direction orders, and the combination of the two respectively, in the settings of $N = 1$, $K = 4$, $k_u = k_v = 6$, and raw binary bits.

Suppose a support region Ω with all n pixels is denoted by $R = \{X_1, X_2, \ldots, X_n\}$, $I(X_i)$ is the intensity of pixel X_i, and $\theta(X_i)$ is the gradient direction of X_i in a rotation invariant coordinate system[1] [12], where $\theta(\cdot)$ is in the range of $[0, 2\pi)$.

[1] Let P denote the center of the support region. $\overrightarrow{PX_i}$ is the y-axis of this coordinate system.

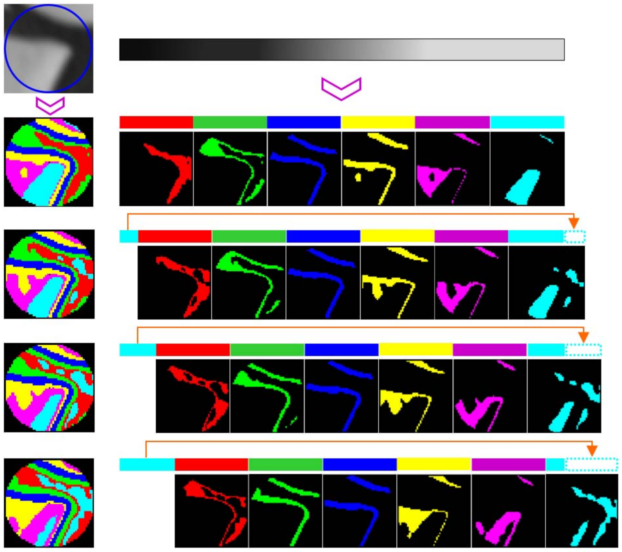


Fig. 3. The illustration of region division based on intensity orders. We utilize circular-shift operation to group all pixels in different ways, which can yield a set of pairwise subregions for computing a binary string. The procedure of region division based on gradient direction orders is similar.

Let K be the number of times that R^I (or R^G) is partitioned by circular-shift operation. Our aim is to divide R into k_u and k_v subregions according to the intensity orders and the gradient direction orders of all pixels in each circular-shift operation respectively. First, all pixels in R are sorted by their intensities and gradient directions in nondescending order respectively, and two sets of sorted pixels are obtained as

$$\begin{aligned} R^I &= \{X_{f_1}, X_{f_2}, \ldots, X_{f_n} : I(X_{f_1}) \\ &\qquad \le I(X_{f_2}) \le \cdots \le I(X_{f_n})\}, \qquad (1)\\ R^G &= \{X_{g_1}, X_{g_2}, \ldots, X_{g_n} : \theta(X_{g_1}) \\ &\qquad \le \theta(X_{g_2}) \le \cdots \le \theta(X_{g_n})\}, \qquad (2) \end{aligned}$$

where $f_1, f_2, \ldots, f_n$ and $g_1, g_2, \ldots, g_n$ are two different permutations of $1, 2, \ldots, n$. Then, we can take $(k_u + 1) \times K$ intensities from R^I and $(k_v + 1) \times K$ gradient directions from R^G as follows:

$$\mathbb{I}_i^s = I(X_{f_{u_i^s}}) : \mathbb{I}_0^s \le \mathbb{I}_1^s \le \cdots \le \mathbb{I}_{k_u}^s, \qquad (3)$$

$$\Theta_j^t = \theta(X_{g_{v_j^t}}) : \Theta_0^t \le \Theta_1^t \le \cdots \le \Theta_{k_v}^t, \qquad (4)$$

where

$$u_i^s = \begin{cases} \left\lceil \dfrac{n}{k_u} i \right\rceil + \left\lceil \dfrac{n}{k_u} \right\rceil \times \dfrac{s-1}{K}, & i \ne 0 \lor s \ne 1, \\ 1, & i = 0 \land s = 1, \end{cases} \qquad (5)$$

$$v_j^t = \begin{cases} \left\lceil \dfrac{n}{k_v} j \right\rceil + \left\lceil \dfrac{n}{k_v} \right\rceil \times \dfrac{t-1}{K}, & j \ne 0 \lor t \ne 1, \\ 1, & j = 0 \land t = 1, \end{cases} \qquad (6)$$

where $i = 1, 2, \ldots, k_u, s = 1, 2, \ldots, K, j = 1, 2, \ldots, k_v$, $t = 1, 2, \ldots, K$. Finally, R^I and R^G are respectively partitioned into $k_u \times K$ and $k_v \times K$ groups as

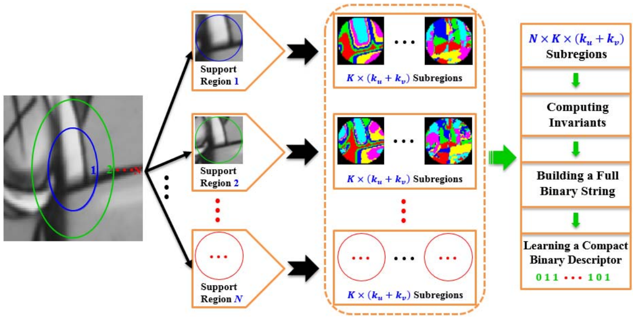

1
2
N
Support Region 1
Support Region 2
Support Region N
$K \times (k_u + k_v)$ Subregions
$K \times (k_u + k_v)$ Subregions
$K \times (k_u + k_v)$ Subregions
$N \times K \times (k_u + k_v)$ Subregions
Computing Invariants
Building a Full Binary String
Learning a Compact Binary Descriptor
011 ··· 101

Fig. 4. The workflow of our proposed method in the multiple support regions framework. Each ellipse corresponds to the boundary of one support region. Suppose that the detected region is taken as the minimal support region whose radius is r. The radius of the other support regions are defined as $r_i = i \cdot r$, $i = 2, 3, \ldots, N$.

$$R_i^s = \begin{cases} \left\{ X_m \middle| \mathbb{I}_{i-1}^s \le I(X_m) \le \mathbb{I}_{i+1}^s \right\}, & i = 1, 2, \ldots, k_u - 1, \\ R^I - \displaystyle\bigcup_{k=1}^{k_u-1} R_k^s, & i = k_u, \end{cases} \qquad (7)$$

$$R_j^t = \begin{cases} \left\{ X_l \middle| \Theta_{j-1}^t \le \theta(X_l) \le \Theta_{j+1}^t \right\}, & j = 1, 2, \ldots, k_v - 1, \\ R^G - \displaystyle\bigcup_{k=1}^{k_v-1} R_k^t, & j = k_v. \end{cases} \qquad (8)$$

It has been shown in [12] and [43] that multiple support regions of different sizes can provide more discriminative information to handle the mismatching problem better than a single support region. Therefore, multiple support regions are also used for constructing our descriptor to further improve its discriminative ability (see Fig. 4 for the scheme of using multiple support regions and Fig. 11 for the matching performance of different numbers of support regions).

B. Computing Raw Binary Descriptor

After subregion division, the following step is to build regional invariants and compare them to produce binary bits. We notice that the corresponding subregions tend to be similar in appearance, shape and spatial geometry properties between matching interest regions, while tend to be dissimilar between non-matching interest regions. Therefore, the built invariants should be able to not only tolerate detection errors and monotonic illumination changes, but also capture these local cues for a rich description. In view of this, we design the following three types of regional invariants: moment invariants, invariants of spatial distribution of pixels, and spatial order invariants of the geometric centroids of subregions (summarized in Table I).

1) Extracting Regional Invariants:

a) Moment invariants: Since moment invariants can capture appearance and shape characteristics of an image region, we use two algebraic moment invariants (intensity variance and gradient magnitude variance) to represent appearance characteristics of subregions, and utilize several geometric

TABLE I
STATISTICAL INVARIANTS USED IN OUR DESCRIPTOR

Invariant		Definition
MI*	aMI*	$\mathbf{V_I} = \frac{1}{n}\sum_{(x,y)\in\Omega}\left(I(x,y)-\overline{I}\right)^2$
		$\mathbf{V_{Gm}} = \frac{1}{n}\sum_{(x,y)\in\Omega}\left(Gm(x,y)-\overline{Gm}\right)^2$
	gMI*	$\psi_\mathbf{1} = \eta_{20}+\eta_{02}$
		$\psi_\mathbf{2} = (\eta_{20}-\eta_{02})^2+4\eta_{11}^2$
		$\psi_\mathbf{3} = (\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2$
		$\psi_\mathbf{4} = (\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2$
		$\psi_\mathbf{5} = (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2$ $-3(\eta_{21}+\eta_{03})^2]+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})$ $\times\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$
		$\psi_\mathbf{6} = (\eta_{20}-\eta_{02})\left[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$ $+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})$
		$\psi_\mathbf{7} = (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2$ $-3(\eta_{21}+\eta_{03})^2]-(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})$ $\times\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$
pSDI*		$\mathbf{\Phi_{ij}} = \frac{1}{2j-1}\left\vert\,\{(x,y)\mid(x,y)\in SubR_i\ \&\ (x,y)\in\mathbb{O}_j\}\,\right\vert$
gcSOI*		$\theta_{\overrightarrow{P_iC'_l}}$

MI*: Moment invariants, aMI*: Algebraic moment invariants, gMI*: Geometric moment invariants, pSDI*: Invariants of spatial distribution of pixels, gcSOI*: Spatial order invariants of geometric centroids.

moment[2] invariants (seven Hu moment invariants [44]) to embody shape information of subregions. The moment invariants are shown in Table I where the normalized central moment η_{pq} is defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = \frac{p+q}{2}+1, \tag{9}$$

where

$$\mu_{pq} = \iint_{\Omega}(x-\overline{x})^p(y-\overline{y})^q I(x,y)\,d(x-\overline{x})\,d(y-\overline{y}) \tag{10}$$

$$\overline{x} = m_{10}/m_{00}, \qquad \overline{y} = m_{01}/m_{00}, \tag{11}$$

$$m_{pq} = \iint_{\Omega}x^p y^q I(x,y)\,dx\,dy, \qquad p,q = 0,1,2,\ldots, \tag{12}$$

where Ω is a subregion, $I(x,y)$ is the intensity of pixel in (x,y). Note that we normalize $I(X)$ ($X \in \left\{X\middle|I(X_{f_1}) \le I(X) \le \mathbb{I}_0^s \bigwedge X \in R^I\right\}$) as $I(X)+\mathbb{I}_{k_u-1}^s-\mathbb{I}_0^s$ when computing the intensity variance of $R_{k_u}^s$ group.

b) The invariants of spatial distribution of pixels: To capture the spatial distribution information of pixels in each subregion, we first divide the support region into several concentric rings of equal space, then build a normalized histogram based on the spatial distribution of each subregion in the concentric rings. As shown in Fig. 5, each bin of the histogram can be computed as

$$\mathbf{\Phi_{ij}} = \frac{1}{2j-1}\left\vert\left\{(x,y)\mid(x,y)\in SubR_i\ \&\ (x,y)\in\mathbb{O}_j\right\}\right\vert, \tag{13}$$

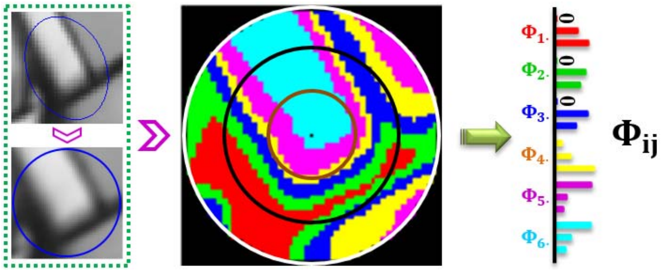


Fig. 5. The normalized histogram based on the spatial distribution of each subregion in the concentric rings.

where (x,y) is a pixel, $|\cdot|$ is the cardinality of a set, $SubR_i$ is the i^{th} subregion, $\mathbb{O}_j$ denotes the j^{th} ($j = 1,2,3.$) concentric ring in a support region.

c) The spatial order invariants of geometric centroids of subregions: For two matching interest regions, the spatial geometric relationship of their subregions should be similar; on the contrary, they should be dissimilar between non-matching interest regions. So we utilize the spatial order of regional geometric centroids to reveal the spatial geometric information.

For each subregion, the geometric centroid can be computed according to Eq. (14) and (15)

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right), \tag{14}$$

where

$$m_{pq} = \sum_{x,y} x^p y^q. \tag{15}$$

As shown in Fig. 7, the k centroids are denoted by $\mathbb{C}(P_i) = \{C_1, C_2, \ldots, C_k\}$. Let the direction of $\overrightarrow{P_iC_1}$ connecting P_i and C_1 be 0°, $\theta_{\overrightarrow{P_iC_l}} \in [0, 2\pi)$, $l = 1,2,\ldots,k$, is taken as the relative direction angle between $\overrightarrow{P_iC_l}$ and $\overrightarrow{P_iC_1}$ anticlockwise. For a pair of matching keypoints, $\theta_{\overrightarrow{P_iC_l}}$ is a discriminative invariant. However, the geometric centroid is unstable when the corresponding subregion is approximately centrosymmetric with respect to the center of support region, such as C_4 and C_6 in Fig. 7. To tackle the problem, we compute $\theta_{\overrightarrow{P_iC'_l}}$ instead of $\theta_{\overrightarrow{P_iC_l}}$ based on the geometric centroids C'_l of the maximal connected components in all subregions. As can be seen in Fig. 7, it is obvious that this improvement is very effective.

2) Computing the Binary String:

Suppose that we have obtained the above different types of invariants adding up to 11 invariants (as shown in Table I) for each subregion in every support region. The sets of different types of invariants are denoted respectively by:

$$\mathbb{V}_1 = \left\{V_{11}^1, \ldots, V_{ij}^1, \ldots, V_{nk}^1\right\}, \tag{16}$$

$$\mathbb{V}_2 = \left\{V_{11}^2, \ldots, V_{ij}^2, \ldots, V_{km}^2\right\}, \tag{17}$$

$$\mathbb{V}_3 = \left\{V_1^3, \ldots, V_l^3, \ldots, V_k^3\right\}, \tag{18}$$

where V_{ij}^1 is the i^{th} moment invariant computed for the j^{th} subregion, $V_{ij}^2 = \Phi_{ij}$, $V_l^3 = \theta_{\overrightarrow{P_iC'_l}}$, $n = 9$, k is the number of

[2] The reason for considering geometric invariance is that there is an unknown rotation between matching regions even after interest-region normalization.

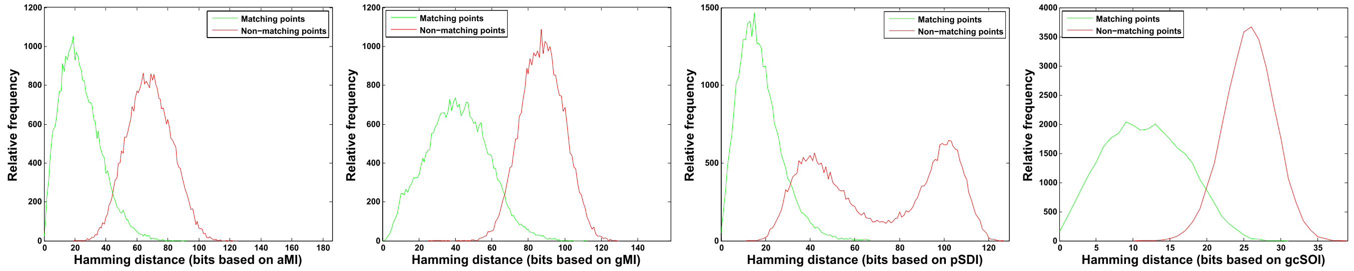


Fig. 6. Distributions of Hamming distances of 4 parts of the OSRI for matching interest points (green lines) and for non-matching interest points (red lines). We extract all ground-truth matching points and equal number of non-matching points from Oxford data set.

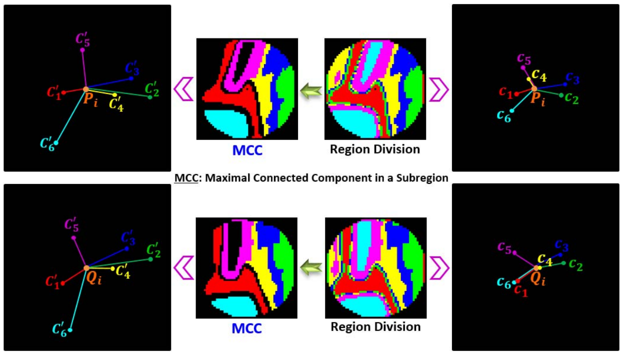

MCC
Region Division
MCC: Maximal Connected Component in a Subregion
P_i
Q_i
c'_1 c'_2 c'_3 c'_4 c'_5 c'_6
c_1 c_2 c_3 c_4 c_5 c_6

Fig. 7. Geometric centroids c_l of 6 subregions and Geometric centroids c'_l of maximal connected components (MCC) in the subregions for two matching regions. The orderings of c'_l are consistent between the two matching keypoints, while the orderings of c_l are inconsistent.

subregions, m is the number of concentric rings. The sets of different types of pairwise invariants are as follows:

$$\mathbb{A}_1 = \left\{(V^1_{is}, V^1_{it}) | V^1_{is} \in \mathbb{V}_1 \wedge V^1_{it} \in \mathbb{V}_1 \wedge s \neq t\right\}, \tag{19}$$

$$\mathbb{A}_2 = \left\{(V^2_{is}, V^2_{jt}) | V^2_{is} \in \mathbb{V}_2 \wedge V^2_{jt} \in \mathbb{V}_2 \wedge (i \neq j \vee s \neq t)\right\}, \tag{20}$$

$$\mathbb{A}_3 = \left\{(V^3_{i}, V^3_{j}) | V^3_{i} \in \mathbb{V}_3 \wedge V^3_{j} \in \mathbb{V}_3 \wedge i \neq j\right\}. \tag{21}$$

We construct our bit-vector descriptor by performing all the comparisons between pairwise invariants $(V_u, V_v) \in \mathbb{A}_m$, $m = 1, 2, 3$, such that each bit $\mathbf{b}$ corresponds to:

$$\mathbf{b} = \begin{cases} 1 & \text{if } \forall (V_u, V_v) \in \mathbb{A}_m, V_u > V_v, \\ 0 & \text{otherwise}. \end{cases} \tag{22}$$

C. Learning a Compact Binary Descriptor

The raw descriptor is a lengthy bit-vector ($21576 = 2256 + 7896 + 10296 + 1128$ bits) generated by comparing all possible pairwise subregions in a support region. Intuitively, there are many redundant bits that are not effective to describe the support region. To reduce correlation among the binary tests, the authors of ORB [29] and FREAK [31] collect many descriptors and organize them as a matrix where the row is viewed as training data and the column as features. Then a greedy forward feature selection is performed to select the least correlated columns. This is actually an unsupervised learning technique. In this paper, we adopt this algorithm to select most discriminative d bits from the raw OSRI based on the PASCAL VOC Challenge 2007 data set.

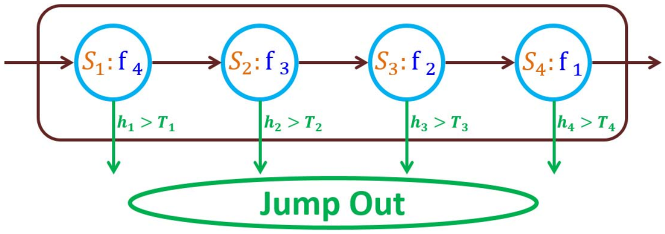


Fig. 8. The flowchart of our cascade filtering.

D. Cascade Filtering for Speeding up Matching

The learned d bits descriptor OSRI are from 4 different categories of pairwise invariants (aMI, gMI, pSDI, and gcSOI) explained in Table I. Formally, we denote by f the d–dimensional feature vector and by $f_i, i = 1, \ldots, 4$ the 4 categories, where $f = f_1 f_2 f_3 f_4$. In the learned OSRI ($N = 1, d = 512$), $|f_1| = 186$, $|f_2| = 158$, $|f_3| = 129$, $|f_4| = 39$. The statistical results in Fig. 6 show that all 4 parts of the OSRI have comparable and high discriminating power. Considering this fact, we resort to the cascade structure [45] to speed up the descriptor matching process, which can make the non-match decision without revealing all the 512 bits.

In Fig. 8 we give the flowchart of the cascade filtering. At the k-th stage S_k ($k = 1, 2, 3, 4$), one of f_i is revealed, denoted by $S_k : f_i$. Then we calculate the accumulative Hamming distance h_k up to the k-th stage and compare it with a threshold T_k. We allow jumping out if $h_k > T_k$. Intuitively, we should place more discriminant f_i at earlier stages. However we experimentally observe that there is no distinct differences in the discriminative power among the 4 parts of bits (see Fig. 6). Considering computational efficiency, we design the order of cascade filtering according to the size of the 4 parts. The threshold T_k is set by letting all the positive training examples pass stage S_k (i.e., 0% False Negative Rate, in this paper, $T_1 = 30$, $T_2 = 85$, $T_3 = 161$, $T_4 = 203$).

IV. Analysis

In this section, we analyze the influence of various parameter settings in the OSRI on the matching performance. Five key parameters (i.e. the number of subregions k_u and

TABLE II

PARAMETER SETTINGS OF THE DETECTOR AND THE FIVE DESCRIPTORS

Hessian-Affine detector	**SIFT descriptor**	**MROGH**	**ORB**	**FREAK**	**OSRI**
thres = 500	magnification = 3	nPartitions = 6	firstLevel = 0	nOctaves = 4	nPartitions = 6
	isNormalized = true	nOrientationBins = 8	nlevels = 8	patternScale = 22.0	K = 4
	recalculateAngles = true	N = 1, 4	scaleFactor = 1.3	oNormalized = true	N = 1,2,3,4
				sNormalized = true	

TABLE III

PARAMETERS OF THE PROPOSED DESCRIPTORS

	Denotation	Parameter settings	Description
OSRI	k_u	4, 5, 6, 7, 8	number of subregions in RD-I
	k_v	4, 5, 6, 7, 8	number of subregions in RD-GO
	K	2, 3, 4, 5, 6	frequency of circular-shift operation
	N	1, 2, 3, 4	number of support regions
	d	128, 256, 512, 768, 1024	learned dimensionality of OSRI

RD-I and RD-GO: region divisions based on intensity orders and gradient orientation orders respectively.

k_v, the frequency of circular-shift operation K, the number of support regions N, and the learned dimensions of OSRI d[3]) are evaluated to test their influence. For the remaining parameters, the empirically determined values are provided in the text where they first appear.

For each parameter, we conduct image matching experiments with the corresponding setting values as listed in Table III (other parameters are set as the optimal values) on the Oxford data set. The evaluation procedure is the same as that of Section V-B.1. The *average recall* versus *average 1-precision* curve is used to show the matching performance. The matching strategy used here is the nearest neighbor distance ratio [9]. Note that we evaluate k_u (or k_v) by merely using region division based on intensity (or gradient orientation) orders for feature description.

It can be seen from Fig. 9 that the parameters k_u and k_v show the same changing trend in the influence on the matching performance. Intuitively, the larger k_u and k_v, the more pairwise sample patches, which can provide more information for feature description. However, when the sample patches are too small, the descriptor will be sensitive to the pixel location errors. This explains why OSRI performs the best at the parameter setting of $k_u = 6$ and $k_v = 6$ rather than $k_u = 8$ and $k_v = 8$. In addition, we can observe from Fig. 9 that OSRI is not sensitive to the changes of parameters k_u and k_v, especially in high precision region.

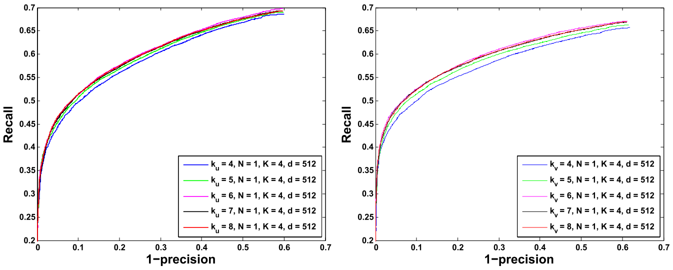


Fig. 9. The average performance of the proposed descriptor with different setting values of parameters k_u and k_v on Oxford data set.

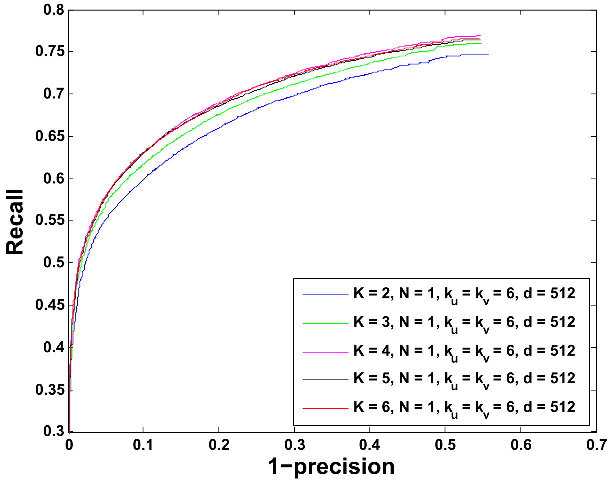


Fig. 10. The average performance comparison under different frequencies of circular-shift operation on Oxford data set.

Fig. 10 shows the influence on the matching performance with different setting values of parameter K. Intuitively, the larger K, the more subregions, which can provide more sampling units for computing binary bits. However, the larger K, the smaller the difference between two neighboring subregions, which will add more redundancy. This explains why OSRI performs the best at $K = 4$ rather than $K = 6$. Meanwhile, similar to k_u and k_v, OSRI is also not sensitive to the changes of K, especially in high precision region.

As shown in Fig. 11, OSRI has better performance of building upon multi-support regions than a single support region under the same dimensionality. However, we also observe that there are no significant differences in the discriminative power at different settings of $N = 2, 3, 4$, especially for $N = 3$ and $N = 4$. Considering computational efficiency, we can use two or three support regions to build OSRI for different vision tasks. While we are also surprised to find that a single support region similarly shows competitive performance, especially in high precision region.

[3] In the OSRI, N and d are independent. However, $d = \bigcup_{i=1}^{N} d_i$ in the MROGH, where d_i is the dimensionality in i^{th} support region. In addition, we denote OSRI$_{N=i}$(128/256/512/768/1024/2048 bits), ORB (256 bits), FREAK (512 bits), SIFT (128–D floating point), and MROGH$_{N=i}$($i \times 48$–D floating point) by OSRI$_{N=i}$–4/8/16/24/32/64, ORB–8, FREAK–16, SIFT–128, and MROGH$_{N=i}$–48 × i, respectively, where $i = 1, 2, 3, 4$.

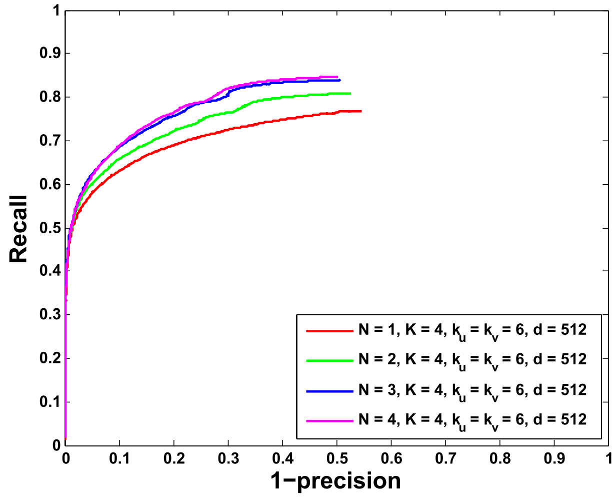


Fig. 11. The average performance comparison between multisupport regions and a single support region for OSRI description on Oxford data set.

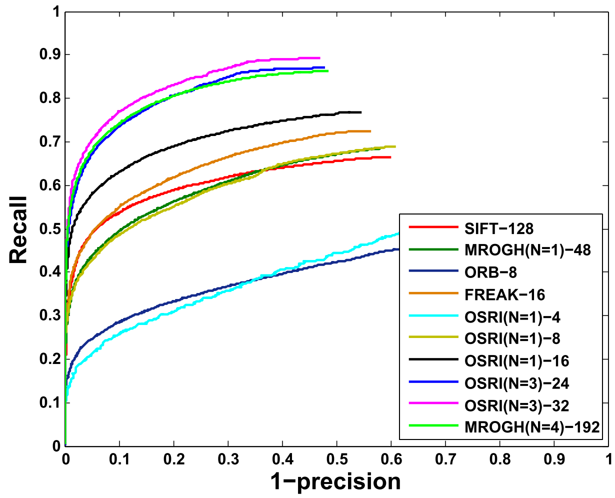


Fig. 12. The average performance comparison among OSRI with different dimensions and the state-of-the-art descriptors on Oxford data set.

Finally, we evaluate the influence of different dimensionality on the matching performance of OSRI, given certain N. The experimental results in Fig. 12 show that the higher dimensionality, the higher distinctiveness. Under the same dimensionality, OSRI$_{N=1}$–8 and OSRI$_{N=1}$–16 significantly perform better than its binary competitors (ORB–8 and FREAK–16) respectively. Meanwhile, OSRI$_{N=1}$–8 is comparable to SIFT–128 and MROGH$_{N=1}$–48, however, OSRI$_{N=1}$–16 evidently outperforms them. In addition, OSRI$_{N=3}$–24 is comparable to MROGH$_{N=4}$–192, however, OSRI$_{N=3}$–32 is already better than it.

V. Experiments

To evaluate the proposed descriptor, we design the following four groups of experiments, using several state-of-the-art descriptors (floating-point: SIFT [8] and MROGH [12], binary: ORB [29] and FREAK [31]) as a baseline:

- *Reliability evaluation of rotation invariance*. We assess the reliability of rotation invariance designs of different methods.
- *Image matching*. We compare our descriptor with several competing descriptors in the matching performance.
- *Object recognition*. We conduct experiments on object recognition to further show the effectiveness and versatility of the proposed descriptor.
- *Computational costs and storage requirements*. We compare extraction and matching costs and storage requirements of different descriptors.

For the fairness and effectiveness of comparisons, we use the Hessian-Affine [46] detector which is more robust to complex image transformations for all descriptors on four challenging real-world data sets (Oxford, 53 Objects, ZuBuD, and Kentucky) in our experiments. For the sake of consistency with results presented in other works, the implementations of different descriptors are as follows: ORB and FREAK are provided with the OpenCV library[4]; two publicly available SIFT implementations can be used from Rob Hess[5] and Andrea Vedaldi[6] (the former is used for our experiment); MROGH[7] and Hessian-Affine[8] were obtained from the authors. To enable the replication of our experimental results, the details of parameter settings are listed in Table II.

A. Reliability Evaluation of Rotation Invariance

To achieve rotation invariance, many floating-point descriptors and binary descriptors (such as BRISK, ORB, and FREAK) all resort to an estimated orientation to design their respective sampling patterns. Fan et al. [12], however, have experimentally proven that the orientation estimation of SIFT descriptor is an error-prone process, which makes many true matches missed due to the estimation errors. In this study, we find that the orientation estimation error[9] results in the matching performance degradation for ORB (the orientation is estimated using the intensity centroid) and FREAK (several long-distance pairwise sampling-points with symmetric receptive fields with respect to the center are utilized to estimate an orientation), based on the Oxford data set shown in Fig. 15.

As shown in Fig. 13, for ORB and FREAK, only 75.62 and 71.84 percent of corresponding points (ground-truth matching points can be determined by the given homography H between two images) have orientation estimation errors in the range of $[-20°, 20°]$ (defined as $\leq 20°$ in this paper) respectively. In other words, 24.38 and 28.16 percent of ground-truth matching points may not be recalled by comparing their descriptors, mainly due to their large orientation estimation errors out of the range of $[-20°, 20°]$ (defined as $> 20°$ in this paper). In conclusion, the designs that resort to an estimated orientation for rotation invariance are unreliable and may result in matching performance degradation for a descriptor.

[4] http://sourceforge.net/projects/opencvlibrary/

[5] http://blogs.oregonstate.edu/hess/

[6] http://www.vlfeat.org/~vedaldi/code/siftpp.html [47]

[7] http://vision.ia.ac.cn/Students/bfan/index.htm

[8] http://www.robots.ox.ac.uk/~vgg/research/affine/

[9] For a pair of matching points (x, y, θ) and (x', y', θ'), the orientation estimation error $\varepsilon_\theta = \theta' - \Theta(\theta; H)$ where $\Theta(\theta; H)$ is the ground-truth orientation of θ by warping the first image to the second image according to the homography H between the two images.

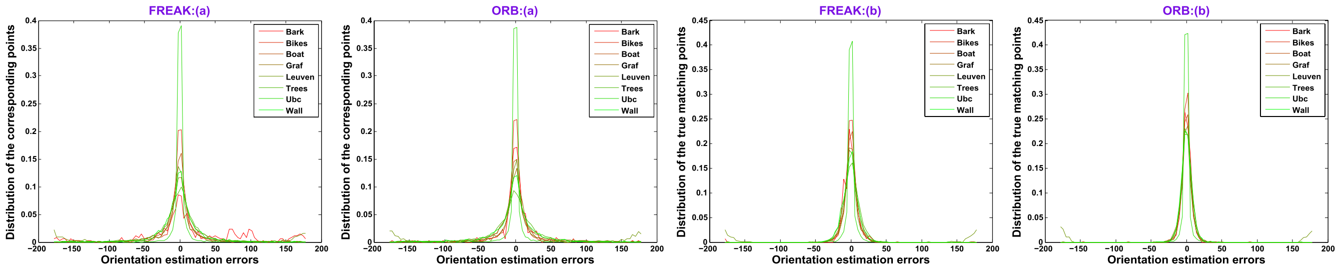


Fig. 13. Distributions of the orientation estimation errors among feature correspondences. FREAK:(a) and ORB:(a) are the distributions among all ground-truth corresponding points based on FREAK and ORB descriptors respectively. FREAK:(b) and ORB:(b) are the distributions among true matching points by comparing their descriptors respectively (for each interest point, we take its nearest neighbor as the matching point).

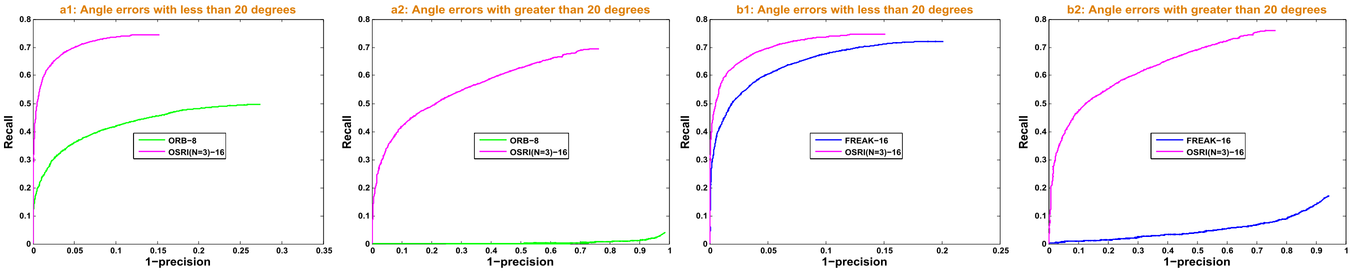


Fig. 14. Matching results of two types of true corresponding points which have orientation estimation errors (≤ 20°) and (> 20°) by the orientation estimation methods of ORB (a1 and a2) and FREAK (b1 and b2) respectively.

Compared with ORB and FREAK (see Fig. 14), the proposed descriptor (OSRI) can recall more effectively the missing true matches whose orientation estimation errors are greater than 20° by the orientation estimation method of either ORB or FREAK. Meanwhile, OSRI has also an excellent performance in recalling the correct correspondences whose orientation estimation errors are less than 20°.

B. Image Matching

1) Data Sets: To ensure the compatibility of our work with existing analyses, we conduct image matching experiments on the Oxford data set[8] which is widely used for evaluating the matching performance of local descriptors [8], [9], [12], [30], [31], [48]. This data set contains a set of benchmark image sequences depicted in Fig. 15 with different geometric and photometric transformations of structured and textured scenes, which are viewpoint changes (*Graffiti* and *Wall*), zoom+rotation changes (*Bark* and *Boat*), image blur (*Bikes* and *Trees*), illumination changes (*Leuven*) and JPEG compression (*Ubc*). The first image in each category is treated as the model image, and the others are the corresponding warped images. Moreover, the ground-truth homography matrix between the model image and each warped image is also given.

2) Evaluation Criterion: We follow the evaluation procedure in [9] to plot the *recall* versus *1-precision* curves. The definitions of *recall* and *1-precision* are as follows:

$$recall = \frac{\#correct\ matches}{\#correspondences},$$
$$1 - precision = \frac{\#false\ matches}{\#putative\ matches}$$

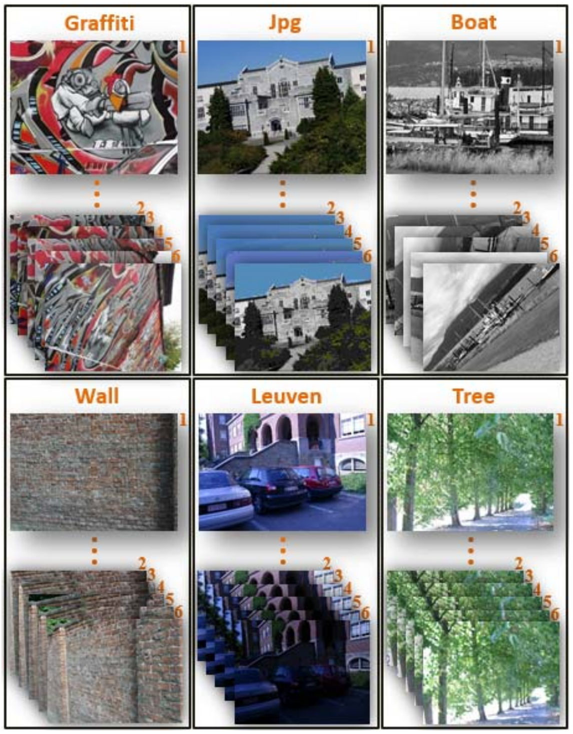


Fig. 15. Image sequences in the Oxford dataset used in our experiments. Each sequence contains six images that are sorted in ascending difficulty, corresponding index of 1 to 6. Hence we consider five image pairs per sequence by matching the first one against the other images. See text for details.

where the number of correct matches and correspondences (ground-truth matches) is determined with the overlap error [46] (i.e. $\zeta_s = 1 - (A \cap H^T BH)/(A \cup H^T BH)$), where

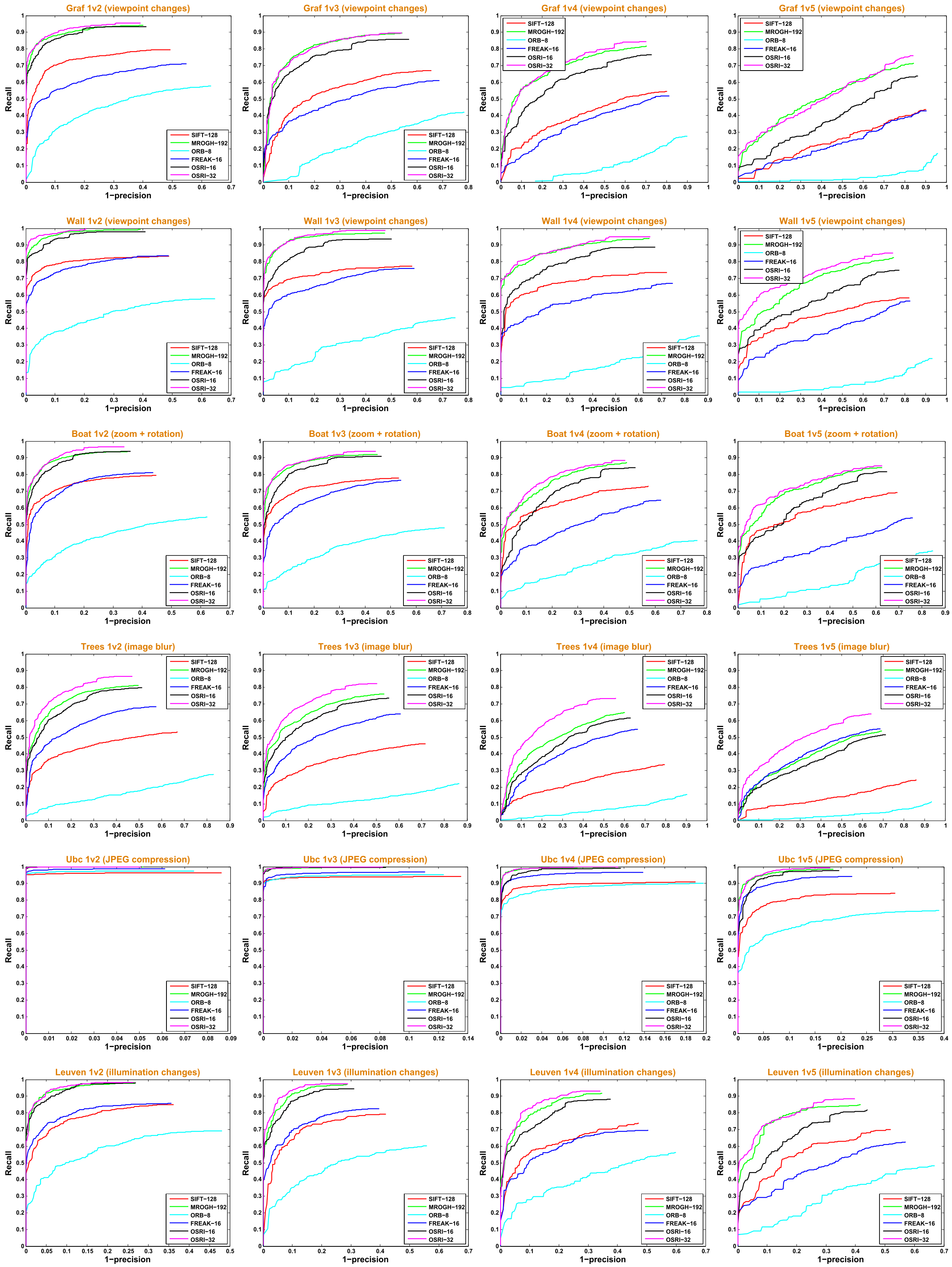


Fig. 16. Experimental results under various image transformations in the Oxford data set for Hessian-Affine detector.

A and B are the detected/interest regions and H is the homography between the images). A match is correct if the overlap error is less than 50%. A *putative match* is defined as a single pair of keypoints/interest-regions whose descriptors meet the matching measure, and a keypoint cannot be matched to more than one keypoint. The set of *correct matches* is the intersection of *putative matches* and ground-truth matches.

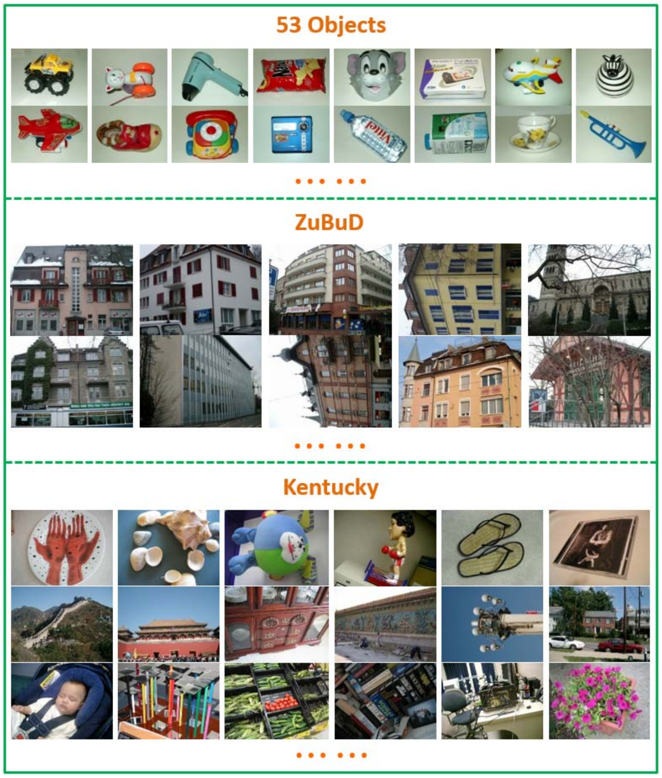


Fig. 17. Some example images of 53 objects, ZuBuD, and Kentucky data sets. More details in the text.

3) Results and Analysis: As shown in Fig. 16, OSRI$_{N=3}$–32 performs the best in all types of image deformations, meanwhile, OSRI$_{N=3}$–32 and OSRI$_{N=3}$–16 all shows a clear advantage comparing with its binary competitors and SIFT. It is also worth noting that OSRI$_{N=3}$–32 has an outstanding performance in both high precision region and high recall region. For instance, 91.6% of tested images rank OSRI as the best descriptor at precisions (≥ 0.95), while all test images rank OSRI as the best in the high recall region.

C. Object Recognition

1) Data Sets: We also conduct object recognition experiments on three publicly available real-world data sets for different descriptors: i) 53 Objects[10], which contains 53 objects with five images taken from different viewpoints for each object; ii) *ZuBuD*[10], which has 1005 images of 201 buildings of historical or architectural interest in Zurich (five different images were taken for each building from different viewpoints); iii) *Kentucky*[11], which contains 10200 images of 2550 objects (CDs, flowers, household objects, keyboards, etc.) where each object has exactly four images. For the sake of fair comparison with [12], we also select the first 4,000 images (1,000 objects) from the Kentucky in our experiments. Please see Fig. 17 for some example images of each data set.

2) Evaluation Criterion: Suppose that I_Q is query image and I_F is reference image. Let $\{f_1^Q, f_2^Q, \ldots, f_m^Q\}$ and $\{f_1^F, f_2^F, \ldots, f_n^F\}$ be two sets of feature descriptors extracted from I_Q and I_F respectively. Similar to [12], the similarity between I_Q and I_F is defined as

$$Sim\left(I_Q, I_F\right) = \frac{\sum_{i,j} \Gamma(f_i^Q, f_j^F)}{m \times n}, \tag{23}$$

where

$$\Gamma(f_i^Q, f_j^F) = \begin{cases} 1 & \text{if } dist\left(f_i^Q, f_j^F\right) \leq T \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

in which $dist\left(f_i^Q, f_j^F\right)$ is the Euclidean distance of f_i^Q and f_j^F, and T is a threshold which is set to provide the best result for each evaluated descriptor. The evaluation criterion depends merely on the the distinctiveness of the local descriptor.

TABLE IV

Object Recognition Results on the Three Datasets (Da*) With Different Local Descriptors (De*)

De* \ Da*	53 Objects	ZuBuD	Kentucky
✧ SIFT–128	52.45%	75.67%	48.83%
✧ MROGH$_{N=1}$–48	69.91%	76.29%	63.17%
✧ MROGH$_{N=4}$–192	**72.50%**	88.10%	74.00%
✍ ORB–8	39.81%	56.27%	34.13%
✍ FREAK–16	34.62%	70.92%	42.81%
✍ **OSRI**$_{N=1}$–16	66.98%	82.56%	69.25%
✍ **OSRI**$_{N=4}$–16	72.26%	**89.18%**	**77.30%**

✧ : Floating-point Descriptors, ✍ : Binary Descriptors.

3) Results and Analysis: For each image, we calculate its similarities to the remaining images in the corresponding data set, and return either the top four images (for 53 Objects and ZuBuD) or the top three ones (for Kentucky) with the largest similarities. We define *the number of correctly returned images/the total number of returned images* as the recognition accuracy. Table IV shows the recognition results. Compared with SIFT–128, ORB–8 and FREAK–16, OSRI$_{N=1}$–16 and OSRI$_{N=4}$–16 all impressively outperform them a lot in recognition accuracy, especially OSRI$_{N=4}$–16. Meanwhile, they also outperform MROGH$_{N=1}$–48 and MROGH$_{N=4}$–192 on two (ZuBuD and Kentucky) of the three data sets respectively.

D. Computational Costs and Storage Requirements

A descriptor should not only exhibit the best possible matching performance but also be as efficient in computation and storage as possible when computational and storage resources are paid much attention in practice. Table V gives timing results measured on an Intel Core2 CPU/2.40GHz using a single core and storage requirements for different descriptors. Although OSRI is slower than ORB and FREAK in the description phase, it is comparable to SIFT and MROGH. The comparison of matching times shows a clear advantage of OSRI over all the other four descriptors. Meanwhile, all binary descriptors have lower storage requirements than SIFT and MROGH.

[10] http://www.vision.ee.ethz.ch/datasets/

[11] http://www.vis.uky.edu/~stewe/ukbench/

TABLE V

COMPUTATION COSTS AND STORAGE REQUIREMENTS OF DIFFERENT DESCRIPTORS. WE RANDOMLY SELECT 100 PAIRWISE IMAGES FROM ZUBUD DATA SET TO ESTIMATE THE MEAN TIME COST OF DESCRIBING AND MATCHING A SINGLE DESCRIPTOR. FOR EACH DESCRIPTION METHOD, A TOTAL OF 106,055 DESCRIPTION VECTORS ARE BUILT, AND 28,051,457 MATCHING OPERATIONS ARE CONDUCTED. STORAGE REQUIREMENT CORRESPONDS TO THE MEMORY FOOTPRINT OF A SINGLE DESCRIPTOR.

	SIFT	**MROGH**	**ORB**	**FREAK**	**OSRI**$_{N=1}$
Description (in ms)	2.93	5.28	0.021	0.028	2.97
Matching (in ns)	1102	1637	17	34	11
Storage (in bytes)	128*4	192*4	32	64	64

VI. CONCLUSION

In this paper, we present a novel method of directly computing a binary descriptor. The key idea is to utilize the ordinal and spatial information of regional invariants to provide more discriminative description over a rotation invariant sampling pattern. The important properties of our method include:

- Our binary descriptor is computed by comparing discriminative regional invariants over rotation invariant sample patches, rather than by comparing smoothed intensities at sample points which is a popular method used in existing binary descriptors.
- We develop a novel sampling pattern to extract a set of pairwise sample patches for pooling our binary descriptor, which is inherently rotation invariant without resorting to a reference orientation for rotation invariance, while being robust to monotonic illumination changes.
- We utilize a learning method to select the best bits for building a compact descriptor. Moreover, we design an effective cascade filter to reject non-matching descriptors at early stages by comparing just a small portion of the whole descriptor.

Our method (OSRI) gains good results by pooling the spatial, shape and appearance properties of subregions over a rotation invariant sampling pattern. Extensive experiments on the challenging data sets show that OSRI achieves better matching performance than state-of-the-art binary descriptors (ORB and FREAK), whilst performing similarly to the best floating-point descriptor MROGH at a fraction of the matching time (two orders of magnitude faster than MROGH) and memory footprint (64 bytes vs. 768 bytes). It is worth mentioning that ORSI is also three times faster to match than ORB and FREAK in the same dimensionality due to its cascade filtering design.

REFERENCES

[1] A. R. Zamir and M. Shah, "Accurate image localization based on Google maps street view," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 255–268.

[2] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[3] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proc. 12th IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 72–79.

[4] S. Zhang, Q. Tian, K. Lu, Q. Huang, and W. Gao, "Edge-SIFT: Discriminative binary descriptor for scalable partial-duplicate mobile search," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 799–813, Jul. 2013.

[5] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. CVPR*, Jul. 2006, pp. 2161–2168.

[6] G. Takacs *et al.*, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proc. ACM Int. Conf. Multimedia Inform. Retr.*, Oct. 2008, pp. 427–434.

[7] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, 2007.

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

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

[10] 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.

[11] 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, May 2010.

[12] B. Fan, F. Wu, and Z. Hu, "Rotationally invariant descriptors using intensity order pooling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2031–2045, Oct. 2012.

[13] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors," in *Proc. 11th IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[14] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Proc. 11th IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[15] S. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 178–185.

[16] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2504–2511.

[17] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3304–3311.

[18] 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.

[19] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[20] J. He *et al.*, "Mobile product search with bag of hash bits and boundary reranking," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 3005–3012.

[21] G. Shakhnarovich, "Learning task-specific similarity," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.

[22] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[23] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 316–336, 2010.

[24] C. Strecha, A. M. Bronstein, M. 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.

[25] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2938–2945.

[26] X. Li, C. Shen, A. Dick, and A. van den Hengel, "Learning compact binary codes for visual tracking," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2419–2426.

[27] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik, "Learning binary codes for high-dimensional data using bilinear projections," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 484–491.

[28] M. Calonder, V. Lepetit, M. Özuysal, 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, Jul. 2012.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[30] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[31] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 510–517.

[32] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” in *Proc. Int. Conf. VLDB*, Sep. 1999, pp. 518–529.

[33] M. Bawa, T. Condie, and P. Ganesan, “LSH forest: Self-tuning indexes for similarity search,” in *Proc. 14th Int. Conf. WWW*, May 2005, pp. 651–660.

[34] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” *Commun. ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[35] M. Raginsky and S. Lazebnik, “Locality-sensitive binary codes from shift-invariant kernels,” in *Proc. Adv. NIPS*, 2009, pp. 1509–1517.

[36] B. Kulis and T. Darrell, “Learning to hash with binary reconstructive embeddings,” in *Proc. Adv. NIPS*, 2009, pp. 1042–1050.

[37] R. Salakhutdinov and G. E. Hinton, “Learning a nonlinear embedding by preserving class neighbourhood structure,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007, pp. 412–419.

[38] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 817–824.

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

[40] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua, “Learning image descriptors with the boosting-trick,” in *Proc. Adv. NIPS*, 2012, pp. 278–286.

[41] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, “Boosting binary keypoint descriptors,” in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2874–2881.

[42] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[43] H. Cheng, Z. Liu, N. Zheng, and J. Yang, “A deformable local image descriptor,” in *Proc. IEEE Conf. CVPR*, 2008, pp. 1–8.

[44] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Trans. Inform. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[45] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Conf. CVPR*, Dec. 2001, pp. 511–518.

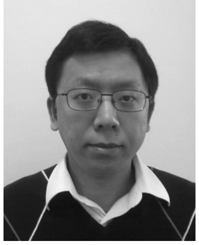
[46] K. Mikolajczyk *et al.*, “A comparison of affine region detectors,” *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, 2005.

[47] A. Vedaldi, “An open implementation of the SIFT detector and descriptor,” Dept. Comput. Sci., UCLA, Los Angeles, CA, USA, Tech. Rep. 070012, 2007.

[48] J. Heinly, E. Dunn, and J. M. Frahm, “Comparative evaluation of binary features,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 759–773.

Xianwei Xu received the M.S. degree from the School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2008. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China. His research interests include image processing, image matching, and feature description.

Lu Tian received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2012, where he is currently pursuing the M.S. degree with the Department of Automation. His research interests are in computer vision and pattern recognition.



Jianjiang Feng (M’10) is an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. He received the B.S. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, in 2000 and 2007, respectively. From 2008 to 2009, he was a Post-Doctoral Researcher with the Pattern Recognition and Image Processing Laboratory, Michigan State University, East Lansing, MI, USA. His research interests include fingerprint recognition, palmprint recognition, and structural matching.

Jie Zhou (M’01–SM’04) was born in 1968. He received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China, where he has been a Full Professor with the Department of Automation since 2003. His research area includes computer vision, pattern recognition, and image processing. He has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 30 papers have been published in top journals and conferences such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the IEEE Conference on Computer Vision and Pattern Recognition. He is an Associate Editor of *International Journal of Robotics and Automation*, *Acta Automatica*, and two other journals. He was a recipient of the National Outstanding Youth Foundation of China.