USB: Ultrashort Binary Descriptor for Fast Visual Matching and Retrieval

Shiliang Zhang, Qi Tian, *Senior Member, IEEE*, Qingming Huang, *Senior Member, IEEE*, Wen Gao, *Fellow, IEEE*, and Yong Rui, *Fellow, IEEE*

***Abstract*— Currently, many local descriptors have been proposed to tackle a basic issue in computer vision: duplicate visual content matching. These descriptors either are represented as high-dimensional vectors relatively expensive to extract and compare or are binary codes limited in robustness. Bag-of-visual words (BoWs) model compresses local features into a compact representation that allows for fast matching and scalable indexing. However, the codebook training, high-dimensional feature extraction, and quantization significantly degrade the flexibility and efficiency of BoWs model. In this paper, we study an alternative to current local descriptors and BoWs model by extracting the ultrashort binary descriptor (USB) and a compact auxiliary spatial feature from each keypoint detected in images. A typical USB is a 24-bit binary descriptor, hence it directly quantizes visual clues of image keypoints to about 16 million unique IDs. USB allows fast image matching and indexing and avoids the expensive codebook training and feature quantization in BoWs model. The spatial feature complementarily captures the spatial configuration in neighbor region of each keypoint, hence is used to filter mismatched USBs in a cascade verification. In image matching task, USB shows promising accuracy and nearly one-order faster speed than SIFT. We also test USB in retrieval tasks on UKbench, Oxford5K, and 1.2 million distractor images. Comparisons with recent retrieval methods manifest the competitive accuracy, memory consumption, and significantly better efficiency of our approach.**

***Index Terms*— Image local descriptor, image matching, large-scale image retrieval, visual vocabulary.**

I. Introduction

MATCHING duplicate visual contents among images is a basic research issue in computer vision, and it serves as an important basis in various applications such as image alignment [13], [15], [26], 3D object reconstruction [2], duplicate detection [11], and large-scale partial-duplicate visual search [8], [9], [35], [37], [40]. To tackle this issue, currently two kinds of image local descriptors have been proposed, *i.e.*, the floating point descriptors and binary descriptors. As one of the most popular floating point descriptors including SIFT [15], SURF [17], and PCA-SIFT [10], the 128-dimensional SIFT [15] is computed by concatenating the 8-dimensional histograms of pixel gradient computed on 16 image sub-regions. VGA sized image commonly contains more than 1000 SIFT descriptors. Extracting and matching SIFT on such images would cost more than 1 second on modern CPUs. Binary descriptors like ORB [23], BRIEF [3], and LDAHash [28] are commonly generated by comparing the intensities of pixels sampled at different locations or mapping the local descriptor into the Hamming space, hence they are more efficient to compute. Compared with SIFT, the 256-bit ORB is about 1-order faster to extract and match, but is less discriminative and robust to image transformations.

A widely used solution to speed up SIFT matching is to generate Bag-of-visual Words (BoWs) model. The BoWs model quantizes SIFTs into compact visual words, hence allows for fast matching [27]. With information retrieval approaches like inverted file indexes, BoWs model is also well suited to large-scale image search task [19]. However, as discussed in lots of works [8], [20], and [35] traditional BoWs model shows limited discriminative power. This is partially because two important steps *i.e.*, local descriptor extraction and quantization, accumulatively and lossily compress visual clues in image patches into visual words. Hence, the final BoWs model could only preserve a small portion of visual clues in image. In addition, as illustrated in many works [7], [19], feature quantization depends on many factors like the number of visual words, the branch number in vocabulary tree, and the training data, *etc*. These make BoWs model not easy to tune for different applications. Besides that, BoWs model requires substantial computation. For instance, codebook training has to process a large number of high dimensional descriptors, which are expensive for extraction and similarity computation [19]. To improve the efficiency, recent binary descriptors like ORB [23] could be used as alternatives to SIFT for BoWs model. However, compared with SIFT, they preserve fewer visual clues and would result in larger quantization error with the binary representation. These defects result in limited flexibility and performance of BoWs model.

Manuscript received September 30, 2013; accepted June 5, 2014. Date of publication June 12, 2014; date of current version July 9, 2014. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the Army Research Office under Grant W911NF-12-1-0057, in part by the Faculty Research Awards through NEC Laboratories of America, in part by the 2012 UTSA START-R Research Award, and in part by the National Science Foundation of China under Grant 61128007, Grant 61025011, and Grant 61332016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Theo Gevers.

S. Zhang and Q. Tian are with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: slzhang.jdl@gmail.com; qitian@cs.utsa.edu).

Q. Huang is with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: qmhuang@jdl.ac.cn).

W. Gao is with Peking University, Beijing 100871, China (e-mail: wgao@pku.edu.cn).

Y. Rui is with Microsoft Research Asia, Beijing 100080, China (e-mail: yongrui@microsoft.com).

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.2014.2330794

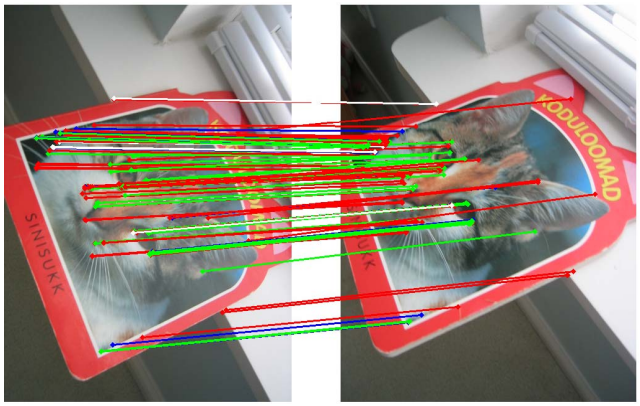


Fig. 1. An example of matched USBs between two images after cascade verification. red: 1-order match, green: 2-order match, blue: 3-order match, white: 4-order match. Higher order means more consistent spatial configurations in neighbor regions of two USBs, hence reasonably reflects larger confidence of matching correctness (refer to Section IV-A).

In this paper, we propose an alternative to current local descriptors and BoWs model. Our main motivation is to extract Ultra Short Binary Descriptors (USBs) from image patches to directly compress their visual clues into a representation as compact as the visual vocabulary, *e.g.*, a 24-bit binary code that corresponds to about 16 million unique IDs. Thus, USBs could be directly used as visual words but avoid the expensive codebook training, high dimensional descriptor extraction, and quantization. To make USB preserve more discriminative power, we select the most discriminative binary codes to construct USB. However, the 24-bit representation is still too short to be discriminative enough. We thus further extract another compact auxiliary spatial feature for each USB. The auxiliary spatial feature captures and quantizes the spatial configuration in the nearby region of each USB into a 64-bit binary code. With the auxiliary spatial feature, USB presents high discriminative power and fast speed for extraction and matching. We call the corresponding image representation as Bag-of-USBs (BoUs) model.

For USB matching, we propose a *cascade verification* strategy which firstly matches the 24-bit USBs with a relatively loose constraint to ensure high recall rate, then filters false matches using the 64-bit auxiliary spatial features in a cascade manner, *i.e.*, includes several steps, each discards a large portion of false matches by verifying a certain kind of spatial clues. Therefore, USB matching could be finished with high efficiency, recall rate, and precision. Moreover, cascade verification reasonably estimates the matching correctness by identifying different match orders (Section IV-A). Higher match order means more consistent spatial configuration between two USBs, thus corresponds to higher confidence of correctness. A typical example of matched USBs with different match orders is illustrated in Fig. 1, where higher order matches include fewer false matches. BoWs model based image retrieval essentially relies on finding a small set of matched local descriptors to retrieve the database images [19]. Therefore, USB plus cascade verification could be easily applied in large-scale image retrieval systems built on BoWs model.

By extracting USBs and the auxiliary spatial features, the final USB descriptor preserves two complementary clues: visual clues and spatial configurations of image keypoints. This is largely different from existing local descriptors like SIFT [15] and ORB [23]. Existing descriptors mainly focus on describing the visual clues surrounding image keypoints, but ignore their spatial configurations which have been proven important for visual matching and recognition [27], [35], [37], [41]–[43]. We jointly preserve the two clues in USB and prove doing so produces high discriminative power even with a ultra compact binary representation. This paper is an original work that studies how to preserve the visual and nearby spatial clues into a compact binary descriptor.

To improve the discriminative power of BoWs model, state-of-the-art approaches either apply a spatial verification [20], [27] to the matched visual words, embed the quantization loss [8], or encode spatial configurations into visual words [25], [32], [41]. Most of them are developed upon the BoWs model and introduce extra computations. Differently, BoUs model is a novel alternative to BoWs model with obvious advantages. We prove that USB plus cascade verification outperforms several recent image search approaches in the aspects of accuracy, efficiency, and memory consumption. Moreover, state-of-the-art BoWs model based approaches could also be built upon BoUs model for better performance. To the best of our knowledge, this paper is also an original work that studies how to generate a better alternative to BoWs model by avoiding high dimensional descriptor extraction, codebook training, and descriptor quantization. The advantages of USB warrant further investigating of this research issue.

The remainder of this paper is organized as follows. Section II reviews related works on local descriptor, feature quantization, and spatial verification. Section III presents our USB and spatial feature extraction. Section IV discusses the cascade verification based image matching and retrieval. Section V analyzes experimental results, followed by the conclusions and future work in Section VI.

II. Related Work

This work studies an alternative to existing local descriptors and BoWs model and is closely related to local descriptor extraction, feature quantization, and spatial verification for image matching and retrieval. In this section, we briefly review the related works and emphasize our differences with them.

Local Descriptor is commonly computed from image patches by either summarizing the pixel gradients from different sub-regions into a high dimensional histogram like SIFT [19] and SURF [1], mapping the high dimensional descriptor into Hamming space like LDAHash [28], keeping the spatial clues of edges like Edge-SIFT [39], or comparing the intensities of pixels at different locations to form a series of binary code like ORB [23] and BRIEF [3]. The 128-dimensional SIFT [15] is one of the most popular image local descriptors in computer vision. SIFT is extracted from image keypoints detected with Difference-of-Gaussian (DoG) detector, which computes the scales, orientations, and

locations of keypoints. With the scale and orientation clues, image patches surrounding keypoints could be normalized into fixed orientation and size, hence scale and orientation invariant local descriptors could be extracted afterward. Based on SIFT, some other similar descriptors like SURF [1], PCA-SIFT [10], Gradient Location and Orientation Histogram (GLOH) [18] are proposed. According to the reported results in [18], SIFT and one of its extensions, *i.e.*, the GLOH generally outperform the other descriptors.

Other than the above mentioned floating point descriptors, many efforts have been made to design efficient and compact descriptors alternative to SIFT or SURF in recent years. Some researchers propose low-bitrate descriptors such as BRIEF [3], BRISK [12], CHoG [5], ORB [23], LDAHash [28], and Edge-SIFT [39] which are fast both to build and match.

BRIEF descriptor is proposed by Calonder et al. [3]. Each bit of BRIEF is computed by considering signs of simple intensity difference tests between pairs of points sampled from the image patches. Despite the clear advantage in speed, BRIEF suffers in terms of reliability and robustness as it has limited tolerance to image distortions and transformations. The BRISK descriptor [12] first efficiently detects keypoints in the scale-space pyramid based on a detector inspired by FAST [22] and AGAST [16]. Then given a set of detected keypoints, BRISK descriptor is composed as a binary string by concatenating the results of simple brightness comparison tests. The adopted detector of BRISK gets location, scale, and orientation clues for each keypoints. Hence BRISK achieves orientation-invariance and scale-invariance. ORB descriptor is built based on BRIEF but is extracted with a novel corner point detector, *i.e.*, o-FAST [23], hence is also robust to the rotation. The authors demonstrate that ORB is significantly faster than SIFT, while performs as well in many situations [23]. Compared with SIFT, despite of the clear advantage in speed, these compact descriptors show limitations in the aspects of descriptive power, robustness or generality.

Feature Quantization with a visual codebook is commonly used for BoWs model generation [8], [9], [19], [21], but easily suffers from expensive codebook training and large quantization error. A visual codebook is commonly obtained by hierarchical K-means clustering of hundreds of millions of local descriptors. The resulting codebook tree typically is deep and contains millions of leaf nodes [19]. Hashing approaches [29], [34] quantize descriptors into compact codes with hashing functions [33]. Although such codebook free approaches are more efficient, the authors of [4] conclude that they show limited discriminative ability. Strategies like soft quantization [21], Hamming Embedding [8], *etc.*, decrease the quantization error in feature quantization, but introduce extra computation. Generally, feature quantization is related to many factors and is hard to tune. Our work is superior because it does not need explicit visual codebook training and feature quantization.

Spatial Verification could be conducted by either verifying the matched visual words [6], [27], [38], [43] or embedding spatial clues directly into visual words [35]–[37], [41] which produce fewer false matches. It has been illustrated that single visual word can not preserve the spatial information in images, which has been proven important for visual matching and recognition [27], [35], [37], [41]–[43]. To combine BoWs with spatial information, lots of works are conducted [27], [35], [37], [41]–[43]. This may be achieved, for example, by using feature pursuit algorithms such as AdaBoosting [31], as demonstrated by Liu et al. [14]. Visual word correlogram and correlation [24], which are leveraged from the color correlogram, are utilized to model the spatial relationships among visual words for object recognition. Visual phrases preserve extra spatial information by involving multiple visual words [8], [41], [43]. For example, Descriptive Visual Phrase (DVP) is generated in [37] by grouping and selecting two nearby visual words. Similarly, visual phrases proposed in [41] are extracted by discovering multiple spatially stable visual words. Generally, considering visual words in groups rather than single visual word could effectively capture the spatial configuration among them, hence gets stronger discriminative power.

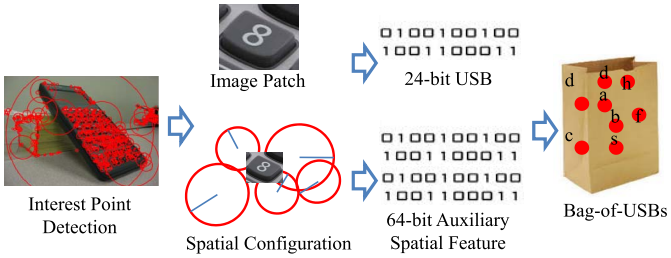


Fig. 2. The framework of Bag-of-USBs model generation.

Not withstanding the success of existing visual phrase features, they still show obvious issues. Currently, each visual phrase is commonly treated as a whole cell and two of them are matched only if they contain identical single visual words [37], [41]. Therefore, two visual phrases are regarded as unrelated, even if they contain partially similar visual and spatial clues. Meanwhile, because of quantization error, visually similar descriptors might be quantized into different visual words. Such quantization error would be aggregated in visual word combinations, and make visual phrase matching more difficult to occur. Thus, current visual phrases are obviously limited in flexility and repeatability.

The principle of our cascade verification is similar to the one of spatial verification in Video Google [27], which also verifies the spatial clues of two matched local features to check the correctness of this match. However, cascade verification is designed in significantly different way to ensure better recall rate, precision and significantly faster speed. Our approach matches USBs with different match orders *e.g.*, Fig. 1. It is more flexible than many works [35], [41], which generate and match features with fixed order, *e.g.*, two-order visual phrase that combines two visual words.

III. USB Descriptor Extraction

As illustrated in Fig. 2, USB descriptor is generated by extracting a USB from each keypoint and its auxiliary spatial feature. We hence define the USB descriptor as, *i.e.*,

$$\{U(p), S(p)\}_{p \in C_Q} \tag{1}$$

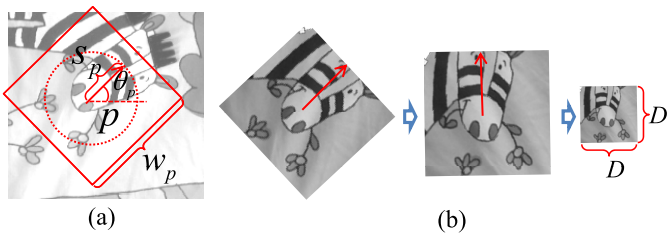


Fig. 3. Illustration of the image window extraction in (a) and image patch normalization in (b).

where, p is a keypoint in image Q, C_Q represents the collection of detected keypoints in Q. We use $U(p)$ and $S(p)$ to denote the USB and spatial feature extracted at keypoint p, respectively. Their extractions are introduced in the Section III-A and Section III-B, respectively.

A. USB Extraction

USB is generated in three steps: 1) extracting image patches from detected keypoints, 2) generating a series of binary bins from each image patch, and 3) selecting discriminative bins as USB.

1) Image Patch Extraction and Normalization: USB is extracted from image patches surrounding keypoints. We use the approaches of Difference-of-Gaussian (DoG) [15] for keypoint detection. As introduced in [15], DoG mainly consists of three steps: 1) scale-space extrema detection, which identifies candidate keypoints on a series of Difference-of-Gaussian images at different scales, 2) keypoint localization, and 3) orientation assignment to keypoints. We use

$$((x_p, y_p), \theta_p, s_p)_{p \in C_Q} \tag{2}$$

to denote a keypoint p detected from image Q, where (x_p, y_p), θ_p, and s_p are the location, orientation, and scale of p, respectively. C_Q represents the collection of detected keypoints in Q. Note that some other fast detectors like o-FAST [23] also detect the location, orientation, and scale clues of keypoints. For instance, o-FAST detects corner keypoints at multi-scales and estimates the orientation of each keypoint by finding its intensity centroid [23]. Hence, USB is not limited to DoG detector. We also test USB extracted from o-FAST detector in the experiments in Section V.

As illustrated in Fig. 3 (a), for a keypoint p, we extract a $w_p \times w_p$ sized window. To make the descriptor invariant to scale change, we compute the size of the windows based on the scale of p, *i.e.*, $w_p = \alpha \times s_p$, where parameter α controls the size of the window. We set α as 12 for DoG detector, which is suggested for SIFT extraction in [15].

To achieve scale and rotation invariance, we normalize the image window to fixed orientation and size. As illustrated in Fig. 3 (b), the orientation is normalized by rotating the image windows to a fixed dominant orientation. As for the scale normalization, we resize each image window to a $D \times D$ sized patch. Hence, the normalized image patches would be robust to rotation and scale changes. In the following part, we proceed to introduce how we compute binary feature from the normalized patch.

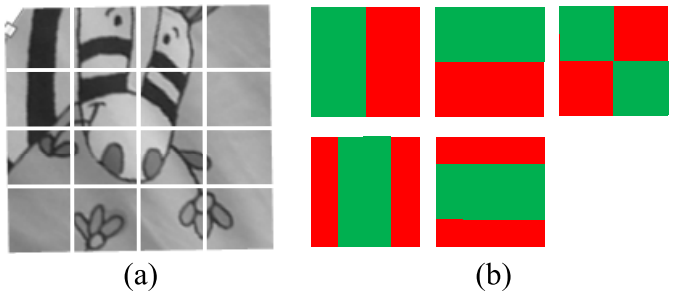


Fig. 4. Illustration of the $g \times g$ divided image patch in (a) and the five binary filters in (b).

2) Binary Bin Generation: Because USB quantizes image patches into compact codes, we should preserve more discriminative clues of the image patch in USB to decrease quantization error. We divide the normalized image patch into $g \times g$ sub-regions (Fig. 4(a)) and extract binary bins from each of them. This is inspired by the SIFT descriptor [15] to record more discriminative spatial clues in the patch. A binary descriptor is hence constructed by concatenating the binary bins extracted in each sub-region.

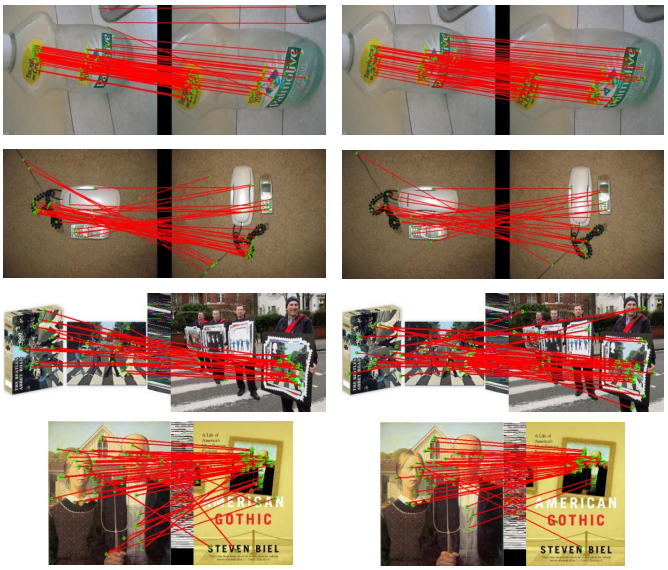
To preserve more visual clues in binary descriptor, as well as to ensure high efficiency, we generate binary bins by comparing the intensities of image patches. Specifically, for an sub-region with side length D/g, we generate 5 binary bins using 5 $D/g \times D/g$ sized binary filters illustrated in Fig. 4(b). Each binary filter contains two kinds of regions with the same size, *i.e.*, red region and green region in Fig. 4(b). The sums of pixel intensities in these two regions are compared to generate a binary bin, *i.e.*,

$$\mathbb{B}(fltr) = \begin{cases} 0, & if : \text{sum}(red_{fltr}) < \text{sum}(green_{fltr}) \\ 1, & if : \text{sum}(red_{fltr}) \geqslant \text{sum}(green_{fltr}), \end{cases} \tag{3}$$

where $fltr$ denotes one of the binary filters, sum$(\cdot)$ computes the sum of pixel intensities in red or green region of the filter.

The reasons why we select the above 5 binary filters are: 1) they show low correlation with each other and capture complementary visual clues; 2) patch-level comparison involves more visual clues and produces better robustness than the intensity comparison on pixel-level in ORB and BRIEF generation [3], [23]. Pixel-level intensity comparsion is easily affected by registration errors caused by affine transformations, inaccurate keypoint localization, lighting changes, *etc*. 3) Moreover, the computation of the five filters can be accelerated with Integrated Image, *i.e.*, the approach used in [30].

The number of visual words in existing large-scale image retrieval works ranges between 1 million to tens of millions [7], [8], [19]. Accordingly, we can extract 20 to 24 binary bins to compose USB, corresponding to 2^{20}, (about 1 million) to 2^{24}, (about 16 millions) unique IDs. We can set g as 2, which generates 4 sub-regions and 20 binary bins. However, as discussed in ORB extraction [23], many binary bins generated by intensity comparison are not informative enough and are correlated with each other. Consequently, we generate a larger number of binary bins, and select the most discriminative bins to compose USB.



(a) (b)

Fig. 5. Comparisons of image matching result between the raw 45-bit binary descriptor in (a) and SIFT descriptor in (b).

We hence reasonably set g as 3, which produces $3 \times 3 = 9$ sub-regions and a $9 \times 5 = 45$-bit raw descriptor. Accordingly, we set D as 36, which produces 12×12 sized sub-regions. We test the 45-bit raw descriptor in image matching task to check if it is a good basis for discriminative binary bin selection. The experimental result shows that the 45-bit descriptor produces surprising good performance, *i.e.*, comparable matching performance with SIFT. Some comparisons of matching result between the raw 45-bit binary descriptor and SIFT are illustrated in Fig. 5. In experimental part, we also show the performance of USB with $g = 4$, which produces a $4 \times 4 \times 5 = 80$-bit raw descriptor. In the following part, we select discriminative binary bins from the raw descriptor.

3) Discriminative Binary Bin Selection: To maximize the visual clue preserved in a limited number of binary bins, each bin should satisfy two criteria: 1) low correlation to the others, 2) high entropy, *i.e.*, evenly distributed on 0 and 1 on a large number of images, which produces small mean-to-0.5 distance in Eq. (5).

Suppose we extract n raw descriptors and organize them in a $45 \times n$ sized matrix M. Based on M, we compute the correlation of two binary bins i and j, *i.e.*, $CR(i, j)$ as:

$$CR(i, j) = \frac{\max(\text{Ham}(M_i, M_j), n - \text{Ham}(M_i, M_j)}{n} \quad (4)$$

where, M_i is the n-bit i-th row of the matrix M. Ham$(\cdot)$ computes the Hamming distance between two n-bit binary vectors. Note that Eq. (4) is different from the standard correlation computation, which treats binary codes as floating point values. We use Eq. (4) for its significantly better efficiency.

For a binary bin i, we compute its mean-to-0.5 distance $MD(i)$ as:

$$MD(i) = |mean(M_i) - 0.5|. \quad (5)$$

Algorithm 1 Binary Bin Selection

Input: the matrix M containing n 45-bit raw descriptors; the desired number of binary bins in USB: d; the threshold of maximum correlation: ω.
Output: the d-bit USB
Initialize two pools: empty P_1 and P_2 containing 45 bins.
From P_2, move a binary bin with minimum mean-to-0.5 distance to P_1, hence $|P_1| = 1$, $|P_2| = 44$
while $|P_1| < d \&\& |P_2| > 0$ **do**
  Find a bin i with minimum $MD(i)$ (Eq. (5)) in P_2
  if $\max_{j \in P_1}(CR(i, j)) < \omega$ **then**
    move i from P_2 to P_1
  else
    delete i from P_2
  end if
end while
Output P_1 as the d-bit USB

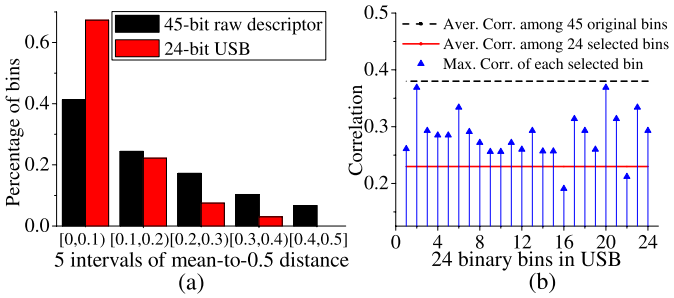


Fig. 6. (a) Percentage of binary bins with different values of mean-to-0.5 distance. (b) Maximum correlation of each selected bin and comparison of average correlations.

Considering the above two criteria, we extract a large number of raw descriptors, then select binary bins iteratively with greedy search like the one in ORB extraction [23]. In each iteration, we select a bin with small mean-to-0.5 distance and low correlations with the already selected bins. This algorithm is summarized in Algorithms 1.

We extract about 5 million raw descriptors from 10K *Flickr* images and set the correlation constraint ω in Algorithm 1 as 0.35. To show the validity of Algorithm 1, we divide the mean-to-0.5 distance into five intervals. On 5K independent images, we compute the mean-to-0.5 distance of each bin and count the percentage of bins falling in each interval. The comparison between the 45 original bins and the 24 selected bins in USB is illustrated in Fig. 6(a), from which we conclude the selected bins distribute more evenly on 0 and 1 than the original bins. On the 5K images, we also test the correlation among the 24 selected bins. For each bin, its maximum correlation to the other 23 bins is summarized in Fig. 6(b), which also compares the average correlations among original bins and selected bins. Obviously, the selected bins are less correlated with each other. More evaluations will be presented in Section V.

B. Auxiliary Spatial Feature Extraction

Besides USB, we extract an auxiliary spatial feature from each keypoint to describe its spatial relationship with neighbor

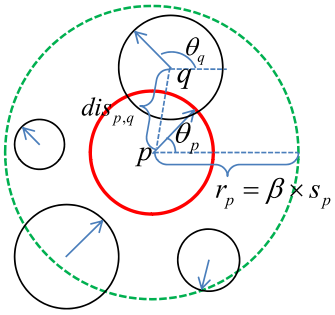


Fig. 7. Illustration of the spatial feature extraction.

keypoints. We define the spatial feature of a keypoint p as:

$$S(p) = \{v_{p,q}, ori_{p,q}, dis_{p,q}\}\,|_{q=1}^{\mathbb{C}} \tag{6}$$

where q is another keypoint in the neighborhood of p, $v_{p,q}$ is the visual clue of q, $ori_{p,q}$ and $dis_{p,q}$ are the orientation and distance relationships between p and q, respectively. $\mathbb{C}$ is the maximum number of neighbor keypoints considered for spatial feature extraction. Because correct matches show similar spatial configurations [6], [27], [43], we could compare the spatial features of two USBs to check if they are correctly matched (Section IV-A).

As illustrated in Fig. 7, for an keypoint p, we consider its neighbor region to extract the spatial feature. To achieve scale invariance, we compute the radius of the region based on the scale of p, *i.e.*, $r_p = \beta \times s_p$, where parameter β controls the size of the region. Larger β involves more keypoints and captures richer spatial information. However, too large β means large search space and degrades the efficiency. We set β as 12, which produces a good tradeoff in our experiments.

Within the neighbor region of p, we select at most $\mathbb{C}$ keypoints with closest scales and DoG responses to the ones of p for spatial feature extraction. DoG keypoints with similar scales and responses commonly locate on the nearby scale spaces with similar repeatability [15]. Therefore, the co-occurrence relationships between them could be more stable. For each selected nearby keypoint q, we hence extract the three aspects of features in Eq. (6). We use the first *8* bits of the USB extracted at q as the visual feature $v_{p,q}$. For $ori_{p,q}$ computation, we quantize the orientation difference between p and q into m intervals, *i.e.*,

$$ori_{p,q} = \lfloor \theta_{p,q} \times m / 2\pi \rfloor, \tag{7}$$

where $\theta_{p,q}$ is the difference of orientation between p and q (Fig. 7) within $[0, 2\pi)$. $dis_{p,q}$ reflects the distance between p and q. It is computed as,

$$dis_{p,q} = \left\lfloor \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \times m \Big/ r_p \right\rfloor, \tag{8}$$

where $dis_{p,q}$ reflects the distance between p and q illustrated in Fig. 7. In Eq. (8), we also quantize $dis_{p,q}$ into m intervals. We set m as *16*, corresponding to 4 bits. Hence, the three spatial features are preserved by $8 + 4 + 4 = 16$ bits.

According to Eq. (6), the size of final spatial feature would be $\mathbb{C} \times 16$ bits. For two matched USBs, if k of their neighbor keypoints show similar spatial features to each other, we call this match as a k-order match. An illustration of match order is presented in Fig. 8. Obviously, higher order match needs to satisfy more strict spatial constraint, making it more confident but more difficult to occur. Therefore, it is not necessary to set $\mathbb{C}$ as large values, which also consume more memory and storage space. We set the maximum of $\mathbb{C}$ as 4, which at most produces 4-order matches. In Section IV-A, we show how we use USB and spatial feature in cascade verification.

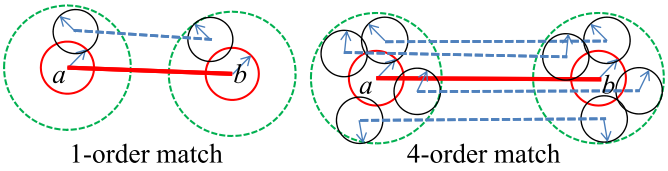


Fig. 8. Illustration of the match order.

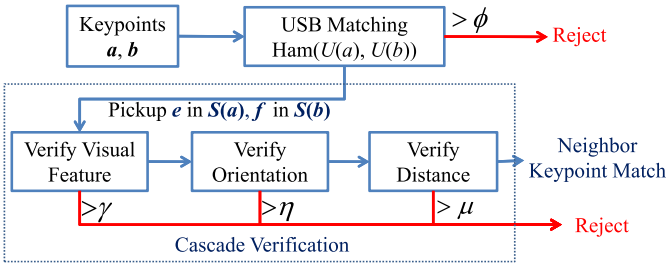


Fig. 9. Illustration of the cascade verification for USB matching.

IV. Application

A. Cascade Verification for Image Matching

Recall that from a keypoint p, we extract two features, USB: $U(p)$ and spatial feature: $S(p)$. Suppose C_Q and C_D are two collections of keypoints extracted from images Q and D, respectively. Then, we match the keypoints between C_Q and C_D.

For two keypoints $a \in C_Q$ and $b \in C_D$, we first check if their USB descriptors are close enough, *i.e.*,

$$\text{Ham}(U(a), U(b)) \leqslant \phi \tag{9}$$

where ϕ controls strictness of USB matching. We call USBs passed Eq. (9) as candidate matches. To assure high recall rate, we set ϕ a relatively large value to generate more candidates. Then, we utilize spatial feature to filter the false matches. ϕ will be discussed in Section V-B.

Suppose keypoints a and b compose a candidate match. To check if it is correct, we see if the spatial configurations of a and b are similar enough. We hence use the three aspects of spatial feature extracted in Eq. (6) to conduct this checking. As illustrated in Fig. 9, we first verify the distance between visual features because it can be computed efficiently with Hamming distance, then we check the orientation and distance relationships. Denote e and f as two neighbor keypoints considered in $S(a)$ and $S(b)$ computation. The verifications

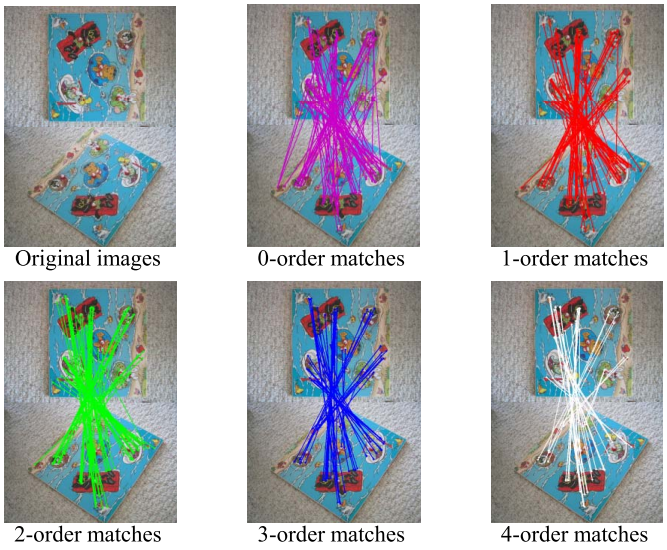


Fig. 10. Examples of matched 24-bit USBs after cascade verification with different match orders.

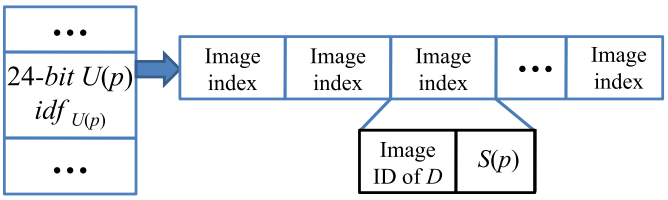


Fig. 11. The proposed index structure of BoUs models.

are computed with:

$$\text{Ham}\left(v_{a,e}, v_{b,f}\right) \leqslant \gamma, \quad (10)$$

$$\min\left(m - \left|ori_{a,e} - ori_{b,f}\right|, \left|ori_{a,e} - ori_{b,f}\right|\right) \leqslant \eta, \quad (11)$$

$$\left|dis_{a,e} - dis_{b,f}\right| \leqslant \mu, \quad (12)$$

where the three parameters γ, η, and μ control the strictness of the cascade verification.

Since we set $\mathbb{C}$ as 4, between a and b, at most 4 of their neighbor keypoints could pass Eq. (10), Eq. (11), and Eq. (12). Because high order matches are difficult to occur, we avoid setting strict γ, η, and μ to assure the recall rate of high order matches. Due to limited space, we omit detailed discussions, and set them as 2, 2, and 3, respectively, which produce reliable matching with different match orders as illustrated in Fig. 10.

To verify two USBs, we at most conduct $4 \times 4 = 16$ times of cascade verification. However, this rarely occurs because Eq. (9) could filter a large portion of false matches with a properly selected ϕ. Besides that, cascade verification is also very fast due to its cascade structure.

B. Image Indexing and Retrieval

USB corresponds to a large number of unique IDs. Hence BoUs model is applicable in most BoWs model based retrieval systems. We use inverted file [19] to index the BoUs models of database images. As illustrated in Fig. 11, for a USB $U(p)$ extracted from keypoint p in image D, we record the ID of D and the spatial feature of p, *i.e.*, $S(p)$, in the inverted index corresponding to $U(p)$. This is slightly different from traditional approach, which records the image ID and the TF (Term Frequency) of visual word in inverted index [19]. Compared with the TF, the size of the recorded $S(p)$ is adjustable, *i.e.*, $\mathbb{C} \times 16$ bits. Hence, the memory consumption of USB based image search is closely related to the maximum number of recorded neighbor keypoints, *i.e.*, the $\mathbb{C}$. We will discuss this parameter and make comparisons of memory consumption with several recent approaches in the experimental part.

Based on the inverted indexes of USBs, we propose our retrieval strategy. We also use IDF (Inverted Document Frequency) to weight the image similarity computation as in [19]. Instead of using TF, we use match order computed by cascade verification to measure the importance of matched USBs to image similarity, *i.e.*, high order matches between query Q and a database image D are more important for image similarity computation than the low order matches. Accordingly, image similarity is computed as:

$$\text{sim}(Q, D) = \sum_{a,b} idf_{U(b)} \times (1+\sigma)^{\text{order}(S(a),S(b))}$$
$$a \in C_Q, b \in C_D, \text{Ham}(U(a), U(b)) \leqslant \phi \quad (13)$$

where a and b are two keypoints from Q and D, whose USBs are similar enough, *i.e.*, with Hamming distance less than ϕ. $U(a)$, $S(a)$, $U(b)$, and $S(b)$, are their USBs and spatial features, respectively. Similar to the IDF of visual word in [19], $idf_{U(b)}$ is the IDF of $U(b)$. order $(S(a), S(b))$ returns the match order of a and b computed by cascade verification. Larger σ sets larger importance to high-order matches in image similarity computation. We will test this parameter in our experiments.

In Eq. (13), USBs in inverted indexes are scanned by USBs in query, if their Hamming distance is less than ϕ. During retrieval, this constraint can be satisfied by query expansion, *i.e.*, exhaustively select 0 to ϕ bins in each USB of query and reverse their values. For a 24-bit USB in Q, query expansion produces $\mathbf{C}_{24}^{\phi}$ new query features. We use these new features to scan the inverted lists. Hence, larger ϕ produces higher recall rate, but may degrade the retrieval accuracy by introducing more noises and may degrade the retrieval efficiency. This parameter will be discussed in Section V-B.

V. Experiments

A. Dataset and Experimental Setup

We first use image matching to test the discriminative power of USB. Both DoG and o-FAST detectors compute location, scale, and orientation clues for each keypoint. Hence, we can extract USB from both DoG and o-FAST keypoints, and easily compare USB with SIFT and ORB in image matching tasks. We call the USB extracted with DoG as ***D-USB***, and the USB extracted with o-FAST as ***o-USB***. Besides that, in the image matching experiment, we also test the effects of several parameters, *i.e.*, parameter g which controls the number of sub-regions in Fig. 4(a), $\mathbb{C}$: the maximum number of recorded

neighbor keypoints in the spatial feature (*i.e.*, Eq. (6)), and the threshold ϕ controlling the strictness of USB matching in Eq. (9). *UKbench* [19] dataset contains 2,550 objects under 4 different viewpoints. In *UKbench*, there commonly exists obvious transformations among the 4 images of the same object, like viewpoint changes, illumination changes, *etc*. These properties make *UKbench* suitable to test the robustness of local descriptors. We hence use *UKbench* as the test set for image matching.

We evaluate the proposed retrieval approaches on 2 public datasets: *UKbench* [19] and *Oxford5K* [20]. *Oxford5K* contains 55 queries and 5,063 annotated landmark images. We test our retrieval approach on both *UKbench* and *Oxford5K* because all of their images have been annotated with groundtruth. Besides that, we also collect a partial-duplicate image dataset, containing 25 categories and about 800 images. Each category is collected by firstly searching keywords like "Abbey Road," "Olympic Logo" in web search engines, then manually downloading the partial-duplicate images in returned results. To conduct the large-scale image search, we mix these three datasets with 1.2 million images collected from *Flickr*, which are used as distractors. All experiments are conducted on a PC with a 4-core 3.4GHz processor and 16GB memory.

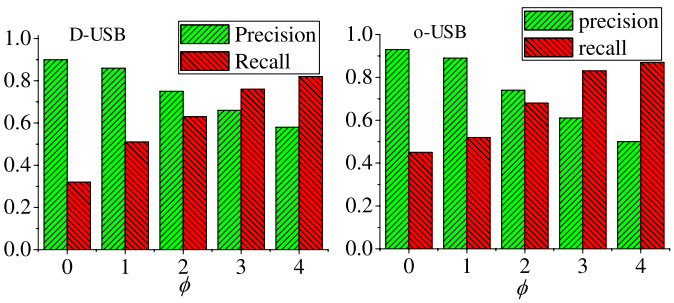


Fig. 12. The matching performance of 24-bit USB extracted with DoG and oFAST with different values of ϕ.

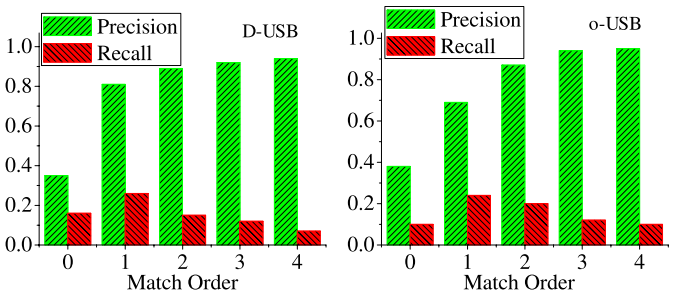


Fig. 13. The matching performance of 24-bit USB extracted with DoG and oFAST plus cascade verification.

B. Image Matching

To test the image matching performance of different descriptors, we first collect all the correctly matched DoG and o-FAST keypoints as a groundtruth set. To collect this groundtruth set, we first match raw SIFT descriptors and raw ORB descriptors with relatively loose constraints. Then, we use RANSAC [6] to filter the false matches. Specifically, among the 4 images of each object in *UKbench*, we first match their SIFT and ORB descriptors between the first image to the other three images with cosine similarity threshold 0.83 and Hamming distance threshold 40, respectively. Then, we conduct RANSAC to the initial matching results. All the matches passing the RANSAC are recorded in the groundtruth set. In the subsequent experiments, we also match the first image with the other three images with different algorithms, and then compute the precision and recall rate based on the collected groundtruth set. Note that, raw SIFT descriptor could produce more reliable matching than the quantized BoWs model. However, SIFT needs more than 1000 milliseconds (ms) to match two images, which is too slow for real world applications.

1) Discriminative Power of 24-Bit USB: In this experiment, we fix g as 3, hence extract a 24-bit USB from the 45-bit raw descriptor. We test the 24-bit USB without cascade verification in image matching to show its discriminative power. Experimental results with different matching thresholds, *i.e.*, ϕ in Eq. (9) are illustrated in Fig. 12. From the figure, we observe that small ϕ reasonably produces high precision rate, but low recall rate. As we increase ϕ, the recall rate is obviously improved. We observe that setting ϕ as 3 produces a good trade-off between precision and recall rate for both o-USB and D-USB, *i.e.*, high recall rate and reasonably good precision. In the followings, we fix ϕ as 3 and further test the performance of USB plus cascade verification.

2) USB Plus Cascade Verification: Cascade verification identifies matched 24-bit USBs with different match orders. Cascade verification is closely related to the number of preserved neighbor keypoints, *i.e.*, $\mathbb{C}$ in spatial feature, because $\mathbb{C}$ defines the maximum match order between two USBs. In Fig. 13, we present the individual precision and recall rate of matched USBs with different match orders. Obviously, high order matches are constantly more accurate than the low order ones. Meanwhile, the 0-order matches present both poor precision and recall rate, hence record most of mismatched USBs. This result reasonably implies that most of the correct matches could pass the cascade verification, while the false matches are effectively rejected. Thus, the validities of the spatial feature and cascade verification are clearly illustrated. Besides that, in the figure, the high order (*e.g.*, 4-order) matches show lower recall rate than the low order (*e.g.*, 1-order) matches. This validates that it is not necessary to set $\mathbb{C}$ in Eq. (6) as a large value because higher order matches are more difficult to occur. Hence, we think fixing $\mathbb{C}$ as 4 is a reasonable choice for USB descriptor.

3) Comparisons: To intuitively show the matching performance of USB plus cascade verification, we compare it with the BoWs model, ORB, and SIFT in the aspects of precision, recall, and matching efficiency (does not include feature extraction time), respectively. We only count the USBs passing cascade verification, *i.e.*, with match order larger than 0. BoWs model is computed by quantizing SIFT descriptors with 1 million visual words, which are generated by hierarchically clustering a large number of SIFT descriptors extracted from

TABLE I
COMPARISON OF MATCHING PRECISION, RECALL, AND EFFICIENCY FOR D-USB IN (A) AND O-USB IN (B)

(A)

Methods	SIFT	BoWs	D-USB g=3	D-USB g=4
Precision	79.3	76.1	90.3	89.2
Recall	100	64.9	72.5	72.1
Time (ms)	421.2	113.1	41.6	41.6

(B)

Methods	ORB	o-USB g=3	o-USB g=4
Precision	78.6	91.6	90.1
Recall	100	81.7	81.3
Time (ms)	69.4	39.3	39.3

an independent dataset. The experimental results are illustrated in Table I.

In the table, SIFT and ORB are utilized for groundtruth set generation, thus they show 100% recall rates but present low precision rates. Both D-USB and o-USB outperform SIFT and ORB by large margins in the aspect of precision. It can be observed that D-USB outperforms the BoWs model significantly in the aspects of precision and recall rate. The o-USB also shows competitive precision and recall rate. Moreover, D-USB is about 10 times faster than SIFT in image matching. Because of the shorter descriptor and the cascade verification which allows early termination of comparison, D-USB and o-USB are even faster than the 256-bit ORB. This experiment clearly demonstrates the effectiveness of our proposed USB and cascade verification.

In Table I, we also show the effects of g. We observe that selecting the 24-bit USB from a 45-bit raw descriptor (*i.e.*, $g = 3$) produces stronger discriminative power than selecting it from a 80-bit raw descriptor (*i.e.*, $g = 4$). This might be because setting g as 3 preserves more visual clues in image patch.

Several comparisons of matching results among SIFT, D-USB, BoWs model and ORB are illustrated in Fig. 14. We also test D-USB in a more challenging vehicle matching task in surveillance video. We found that D-USB produces surprisingly good performance in matching the identical vehicle taken under different cameras, time, and viewpoints. Some examples of matched vehicles based on D-USB are illustrated in Fig. 15.

C. Image Retrieval

In this experiment, we implement the BoWs model based image retrieval [19] as the baseline. To generate the BoWs model, a vocabulary tree is trained by hierarchically clustering about 50 million SIFT descriptors extracted from an independent dataset [19]. By setting branch number as 10 and layer number as 6, we finally get about 1 million visual words. ORB is extracted with o-FAST detector, which also extracts scale and orientation clues [23], therefore the first 24 bits of ORB could also be applied in our retrieval approach. We hence also compare USB with the first 24 bits in ORB, which are the most discriminative 24 bits, according to the extraction strategy of ORB [23].

Fig. 14. Comparisons of image matching result among SIFT, D-USB, BoWs model, and ORB. red: 1-order match, green: 2-order match, blue: 3-order match, white: 4-order match.

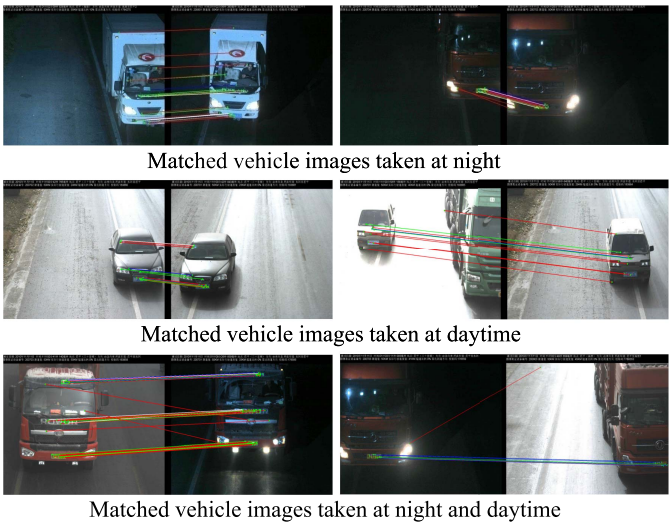


Fig. 15. Illustration of matched vehicles in surveillance video using D-USB.

We use *UKbench* and *Oxford5K* as test sets because it is convenient to make comparisons with state-of-the-arts on them. We adopt the metrics in the original papers of the 2 datasets to evaluate the retrieval performance: the recall rate for the top-4 returned candidates (referred as the N-S score) for *UKBench*, and the mAP (mean Average Precision) for *Oxford5K*.

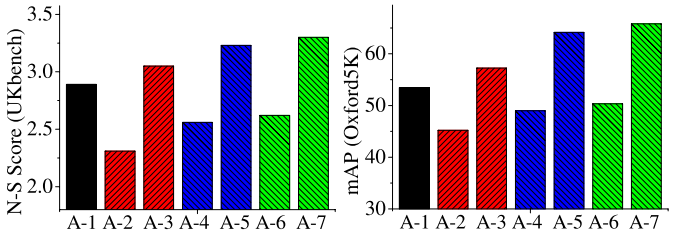


Fig. 16. The comparison of retrieval results among A-1 to A-7 on *UKbench* and *Oxford5K*.

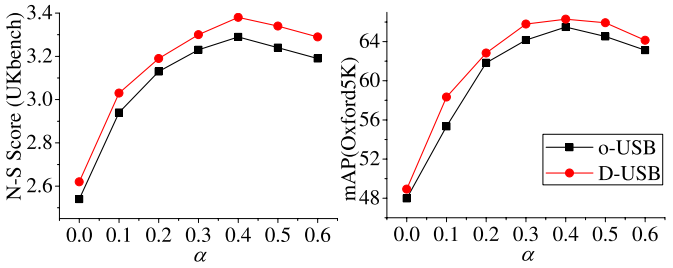


Fig. 17. The influence of σ on the performance of 24-bit USB plus cascade verification.

Our retrieval approach is related to two parameters in Eq. (13), *i.e.*, 1) ϕ which controls the maximum number of expanded bits in USBs of query and 2) σ which controls the weight of high order matches in image similarity computation. Referring to the image matching result in Section V-B, we fix ϕ as 3. In the followings, we first make comparison with other algorithms by fixing σ as 0.3 for both D-USB and o-USB. Then, we tune σ to seek the best performance. The compared algorithms are:

- A-1: BoWs model
- A-2: 24-bit ORB (the first 24 most discriminative bits [23])
- A-3: 24-bit ORB plus cascade verification
- A-4: 24-bit o-USB
- A-5: 24-bit o-USB plus cascade verification
- A-6: 24-bit D-USB
- A-7: 24-bit D-USB plus cascade verification

The retrieval performances of these algorithms are compared in Fig. 16, which obviously shows that our cascade verification significantly improves the retrieval performances of ORB, o-USB, and D-USB. Using the same keypoint detector, o-USB outperforms ORB by large margins on the two datasets. This verifies that our patch-level intensity comparison strategy is superior to the pixel-level intensity comparison of ORB for *short* binary code extraction. The comparison among A4 to A7 manifests that D-USB outperforms o-USB. This is reasonable because DoG detector computes more reliable location, scale and orientation clues that the o-FAST detector.

We also observe that BoWs model outperforms 24-bit ORB, o-USB, and D-USB if no cascade verification is applied to them. We conclude that BoWs model is still more discriminative than the 24-bit USB descriptor, even it is carefully designed. This might be because 128-dimensional SIFT preserves far more visual clues than the 24-bit binary code. Meanwhile, the trained vocabulary tree reasonably captures the distribution of SIFT descriptors in test set, thus produces relatively small quantization loss. With cascade verification, the 24-bit ORB, o-USB, and D-USB all outperform the BoWs model by large margins. This clearly demonstrates the validity of our retrieval approach.

In another experiment, we test the influence of σ on the performance of the 24-bit USB plus cascade verification. As shown in Fig. 17, a proper σ significantly improves the retrieval performances of both o-USB and D-USB on the two datasets. This is reasonable because high order matches are more accurate than the low order matches, hence they should be emphasized in image similarity computation. However, overemphasizing the high-order match by setting a too large σ also degrades the performance. According to Fig. 13, this might be because high order matches are more sparse than the low order matches, thus they cannot perform well alone. We observe that setting σ as 0.4 achieves the best performance on both the two datasets. We hence set σ as 0.4 for the following large-scale image retrieval experiments.

TABLE II
RETRIEVAL PERFORMANCE COMPARISON AMONG OURS WITH STATE-OF-THE-ARTS

Methods	**Ours**	[20]	[8]	[41]	[35]
UKbench, N-S Score	**3.38**	3.45	3.42	3.26	3.23
Oxford5K, mAP(%)	**66.29**	64.5	61.5	71.3	64.21

The comparisons of retrieval performance among our approach with several recent methods are shown in Table II. The comparisons demonstrate that although our approach is based on a very compact binary descriptor (24+64-bit), it still shows competitive retrieval precision. Note that the compared methods introduce extra computations to the BoWs model by either fusing the retrieval results of multiple features [32], or performing complicated spatial clue computation [35] and online spatial verification [41]. In the following large-scale image retrieval experiments, we manifest that USB is significantly more efficient than BoWs model. Hence, it is reasonable to infer that our approach would be far more efficient than these compared algorithms. Another advantage of our approach is it does not need codebook training, hence it is more flexible and easy to use. More comparisons of efficiency and memory consumption will be made in the following experiments.

D. Large-Scale Image Retrieval

In large-scale experiment, we use 1.2 million *Flickr* images as distractors and conduct the retrieval with the original queries in *UKbench*, *Oxford5K*, and our partial-duplicate image dataset. We compare with several recent image retrieval approaches in the aspects of retrieval precision, efficiency (including feature extraction time), and memory consumption. The compared methods are:

- B-1: o-USB plus cascade verification, with $\mathbb{C} = 2$
- B-2: D-USB plus cascade verification, with $\mathbb{C} = 2$
- B-3: o-USB plus cascade verification, with $\mathbb{C} = 4$
- B-4: D-USB plus cascade verification, with $\mathbb{C} = 4$

TABLE III
THE EXPERIMENTAL RESULTS OF LARGE-SCALE IMAGE SEARCH ON UKBENCH (A), OXFORD5K (B), AND PARTIAL-DUPLICATE IMAGE DATASET (C) PLUS 1.2 MILLION FLICKR IMAGES

(A)

Methods	B-1	B-2	B-3	B-4	HE [8]	DVP [37]
N-S score	2.95	3.03	3.03	3.12	3.10	3.04
Time (ms)	101.5	312.2	118.8	321.9	712.3	738.4
Index Size (GB)	3.61	3.72	5.83	5.91	6.13	6.34

(B)

Methods	B-1	B-2	B-3	B-4	HE [8]	DVP [37]
mAP	59.21	62.14	61.97	63.11	61.34	61.07
Time (ms)	111.9	316.2	121.4	329.3	813.1	825.7
Index Size (GB)	3.63	3.81	5.93	6.02	6.21	6.62

(C)

Methods	B-1	B-2	B-3	B-4	HE [8]	DVP [37]
mAP	71.21	73.15	73.39	76.62	72.62	73.22
Time (ms)	98.45	302.3	101.5	311.2	701.1	721.5
Index Size (GB)	3.59	3.62	5.81	5.88	6.03	6.21

- HE: Hamming Embedding [8]
- DVP: Descriptive visual word and descriptive visual phrase [37].

Note that the parameter $\mathbb{C}$ in Eq. (6) defines the maximum number of neighbor keypoints kept in the spatial feature of USB. Hence, $\mathbb{C}$ largely decides the memory consumption of USB in large-scale image search.

We summarize the experimental results on the three datasets in Table III. In the table, it is clear that D-USB plus cascade verification with $\mathbb{C} = 4$ outperforms HE [8] and DVP [37] in the aspect of both retrieval precision and efficiency. The better efficiency is mainly because USB does not need to extract and quantize the high dimensional SIFT descriptors, which commonly need about 300 ms to finish. It is necessary to point out that o-USB also shows competitive retrieval performance. For example, with $\mathbb{C} = 4$ o-USB outperforms both HE and DVP on *Oxford5K* and the partial-duplicate image dataset. Because o-USB uses a faster keypoint detector, it demonstrates significantly faster retrieval speed than D-USB, HE, and DVP. Our experiments show that o-USB is about 7 times faster than HE and DVP, and about 3 times faster than D-USB. We believe this is a significant progress in large-scale image search.

Table III also compares the memory consumption. Note that HE [8] needs to record a binary vector for each visual word to preserve its quantization loss. DVP [37] needs to build the inverted indexes for both visual words and visual phrases. We observe that with $\mathbb{C} = 4$, *i.e.*, recording a 64-bit spatial feature for each indexed USB, both D-USB and o-USB show slightly better memory consumption than HE and DVP. However, the memory consumption of USB based image retrieval could be flexibly adjusted by setting different values of $\mathbb{C}$. It is clear in the table that the memory consumption of USB is significantly decreased with $\mathbb{C} = 2$. In this case, we observe o-USB and D-USB still show competitive performances. Hence, we could conclude that USB is more flexible in memory consumption sensitive applications like mobile application than these compared methods.

Fig. 18. Examples of retrieved images before the first false positive and matched D-USBs between query and returned images. The images highlighted with bounding boxes are queries.

In Fig. 18, we show several examples of D-USB based partial-duplicate image search and matched D-USBs between query and retrieved images. It is clear that D-USBs in query and returned images could be reliably matched despite of obvious image changes in illumination, occlusion, affine transformations, *etc.*

VI. CONCLUSIONS

In this paper, we study an alternative to current local descriptors and BoWs model by jointly extracting binary USBs and compact auxiliary spatial features from keypoints. USB directly compresses the visual clues of keypoints into an ultra compact 24-bit representation that allows for fast matching and indexing. The auxiliary spatial feature effectively captures the spatial configuration in the neighbor region of the keypoint into a 64-bit representation, hence could be utilized to remove the mismatched USBs in cascade verification. The USB and spatial feature are complementary to each other and both of them can be extracted and matched efficiently. USB is also compatible with o-FAST detector, which further improves its efficiency. Experimental results of image matching reveal that 24-bit USB plus cascade verification shows promising precision, recall rate, and speed. Image search experiments show our retrieval approach based on USB also consistently outperforms BoWs model and several recent image search approaches in the aspects of accuracy and efficiency.

In this paper, we generate the USB descriptor by carefully designing some rules that are helpful to preserve more visual and spatial clues in image patches. However, our experiments show that without the cascade verification, the 24-bit USB still can not outperform the BoWs model generated on 1 million visual words. In our future work, we will design a learning strategy to further maximize the discriminative clues preserved in USB. One possible strategy is to optimize the 24-bit descriptor by making it produce similar distance measurements with the raw SIFT descriptor.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, May 2006.

[2] M. Brown and D. Lowe, "Unsupervised 3D object recognition and reconstruction in unordered datasets," in *Proc. IEEE Int. Conf. 3-D Digit. Imag. Model.*, Jun. 2005, pp. 56–63.

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "Brief: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010.

[4] V. Chandrasekhar *et al.*, "Survey of SIFT compression schemes," in *Proc. Int. Mobile Multimedia Workshop (ICPR)*, 2010.

[5] 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. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2504–2511.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] J. Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.

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

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptor into a compact image representation," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3304–3311.

[10] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jul. 2004, pp. II-506–II-513.

[11] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicated detection and sub-image retrieval," in *Proc. ACM Multimedia*, Oct. 2004.

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

[13] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 377–389.

[14] D. Liu, G. Hua, P. Viola, and T. Chen, "Integrated feature selection and higher-order spatial feature extraction for object categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2008, pp. 1–8.

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

[16] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 183–196.

[17] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf.*, 2002.

[18] 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.

[19] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2006, pp. 161–2168.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[22] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An effcient alternative to SIFT or surf," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.

[24] S. Savarese, J. Winn, and A. Criminisi, "Discriminative object class models of appearance and shape by correlatons," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. 2006, pp. 2033–2040.

[25] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-NN reranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3013–3020.

[26] H. Y. Shum and R. Szeliski, "Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment," *Int. J. Comput. Vis.*, vol. 36, no. 2, pp. 101–130, 2000.

[27] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[28] 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. 2010.

[29] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[30] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. 2001, pp. 511–518.

[31] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, Feb. 2004.

[32] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 209–216.

[33] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inform. Process. Syst.*, 2008.

[34] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2008, pp. 217–220.

[35] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 501–510.

[36] S. Zhang, Q. Huang, Y. Lu, W. Gao, and Q. Tian, "Building pair-wise visual word tree for efficient image re-ranking," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2010, pp. 794–797.

[37] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li, "Descriptive visual words and visual phrases for image applications,"in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 75–84.

[38] S. Zhang, Q. Tian, Q. Huang, and Y. Rui, "Embedding multi-order spatial clues for scalable visual matching and retrieval," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 130–141, Jan. 2014.

[39] 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. 22, no. 7, pp. 2889–2902, Jul. 2013.

[40] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1673–1680.

[41] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 809–816.

[42] Y.-T. Zheng, M. Zhao, S.-Y. Neo, T.-S. Chua, and Q. Tian, "Visual synset: Towards a higher-level visual representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[43] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 511–520.

Shiliang Zhang received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He was a Visiting Researcher with the Media Analysis Group, NEC Laboratories America, Cupertino, CA, USA. He is currently a Post-Doctoral Research Fellow with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA.

Dr. Zhang's research interests include large-scale image and video retrieval and multimedia content affective analysis. He was a recipient of the CCF Excellent Doctoral Dissertation by the China Computer Federation, the Outstanding Doctoral Dissertation Award by the Chinese Academy of Sciences, the President Scholarship by the Chinese Academy of Sciences, the ACM Multimedia Student Travel Grants, and the Microsoft Research Asia Fellowship in 2010. He has authored and co-authored over 20 papers in journals and conferences, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, COMPUTER VISION AND IMAGE UNDERSTANDING, the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, *ACM Multimedia*, and the International Conference on Computer Vision. He was also a recipient of the Top 10% Paper Award at the IEEE International Workshop on Multimedia Signal Processing in 2011.

Qi Tian (M'96–SM'03) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University, Philadelphia, PA, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2002. He is currently a Professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. He took one-year faculty leave at Microsoft Research Asia, Beijing, from 2008 to 2009.

Dr. Tian's research interests include multimedia information retrieval and computer vision. He has authored over 210 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA, and he also received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He was a recipient of the Best Paper Awards in PCM 2013, MMM 2013, and ICIMCS 2012, the Top 10% Paper Award in MMSP 2011, the Best Student Paper in ICASSP 2006, the Best Paper Candidate in PCM 2007, and the ACM Service Award in 2010. He is the Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, *Journal of Computer Vision and Image Understanding*, *Pattern Recognition Letter*, *EURASIP Journal on Advances in Signal Processing*, *Journal of Visual Communication and Image Representation*, and is in the Editorial Board of the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY, *Multimedia Systems Journal*, *Journal of Multimedia*, and *Journal of Machine Visions and Applications*.

Qingming Huang (M'04–SM'08) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1994.

He was a Post-Doctoral Fellow with the National University of Singapore, Singapore, from 1995 to 1996, and was with the Institute for Infocomm Research, Singapore, as a member of the research staff from 1996 to 2002. He joined the Chinese Academy of Sciences, Beijing, China, under Science100 Talent Plan in 2003, where he is currently a Professor with the Graduate University. His current research areas are image and video analysis, video coding, pattern recognition, and computer vision.

Wen Gao (M'92–SM'05–F'08) received the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1985 and 1988, respectively, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Research Fellow with the Institute of Medical Electronics Engineering, University of Tokyo, in 1992, and a Visiting Professor with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, in 1993. From 1994 to 1995, he was a Visiting Professor with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, and a Professor of Computer Science with the Harbin Institute of Technology.

Yong Rui (F'10) received the B.S. degree from Southeast University, Nanjing, China, the M.S. degree from Tsinghua University, Beijing, China, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He is currently a Senior Director with Microsoft Research Asia, Beijing.

Dr. Rui is a fellow of the International Association for Pattern Recognition and the International Society of Optics and Photonics, and a Distinguished Scientist of the Association for Computing Machinery (ACM). He is an Associate Editor-in-Chief of the IEEE MULTIMEDIA MAGAZINE, an Associate Editor of the *ACM Transactions on Multimedia Computing, Communication and Applications*, and the Founding Editor of the *International Journal of Multimedia Information Retrieval*. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA from 2004 to 2008, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2006 to 2010, the *ACM/Springer Multimedia Systems Journal* from 2004 to 2006, and the *International Journal of Multimedia Tools and Applications* from 2004 to 2006.