Edge-SIFT: Discriminative Binary Descriptor for Scalable Partial-Duplicate Mobile Search

Shiliang Zhang, Qi Tian, *Senior Member, IEEE*, Ke Lu, Qingming Huang, *Senior Member, IEEE,* and Wen Gao, *Fellow, IEEE*

***Abstract*— As the basis of large-scale partial duplicate visual search on mobile devices, image local descriptor is expected to be discriminative, efficient, and compact. Our study shows that the popularly used histogram-based descriptors, such as scale invariant feature transform (SIFT) are not optimal for this task. This is mainly because histogram representation is relatively expensive to compute on mobile platforms and loses significant spatial clues, which are important for improving discriminative power and matching near-duplicate image patches. To address these issues, we propose to extract a novel binary local descriptor named Edge-SIFT from the binary edge maps of scale- and orientation-normalized image patches. By preserving both locations and orientations of edges and compressing the sparse binary edge maps with a boosting strategy, the final Edge-SIFT shows strong discriminative power with compact representation. Furthermore, we propose a fast similarity measurement and an indexing framework with flexible online verification. Hence, the Edge-SIFT allows an accurate and efficient image search and is ideal for computation sensitive scenarios such as a mobile image search. Experiments on a large-scale dataset manifest that the Edge-SIFT shows superior retrieval accuracy to Oriented BRIEF (ORB) and is superior to SIFT in the aspects of retrieval precision, efficiency, compactness, and transmission cost.**

***Index Terms*— Image local descriptor, large-scale image search, mobile vision.**

I. Introduction

WITH the fast development of RISC (Reduced Instruction Set Computer) processor, mobile camera, displaying technology, wireless network, mobile devices have become more powerful, ubiquitous and important to users. Compared with traditional computer, the advantages of mobile device, such as flexible and freshing user experience, convenience, easy access, *etc.*, are appealing to users. The fast development, obvious advantages, and growing popularity not only make mobile device generated multimedia data increase explosively, but also make mobile device an important and promising platform for multimedia applications and multimedia research.

In this paper, we study large-scale partial-duplicate image search on mobile platform. Compared with traditional image search, image search on mobile platform needs to take more factors into consideration, *e.g.*, the limited computational resource, storage and memory capacity, relatively expensive wireless data transmission, rich sensors like GPS, accelerometer, gyroscope, GPS, *etc*. Thus, there are many research topics need to be further explored, *e.g.*, efficient feature extraction, quantization, compression, effective utilization of multiple sensors, user preference acquisition and application, *etc*.

To achieve efficient and accurate partial-duplicate mobile visual search, we focus on extracting compact, discriminative, and efficient local descriptor which is commonly known as a basis for Bag-of-visual Words (BoWs) representation [1], [2]. As one of the most promising solutions in large-scale partial-duplicate image search, traditional BoWs based image retrieval generally can be divided into several key steps.

1) Local feature extraction that extracts feature vectors such as SIFT [1] from stable and repeatable interest points.
2) Visual vocabulary generation that commonly generates visual words by clustering a large number of local features. After clustering, each cluster center is taken as a visual word.
3) BoWs representation generation and offline image indexing: by replacing the contained local features with their nearest visual words, each image can be compressed into a compact BoWs representation [2]. With such representation, inverted file indexing strategy could be leveraged for image indexing.
4) Online image retrieval: each pair of identical visual words from two images is regarded as a pair of visually matched image patches.

Hence, similarity between two images can be computed according to the number of their identical visual words.

According to the four steps, the discriminative power of visual word largely decides the accuracy of local feature matching and hence is closely related to the quality of image retrieval. Meanwhile, the retrieval efficiency mainly relies on the speed of BoWs representation computation. Consequently,

Manuscript received September 20, 2012; revised January 20, 2013; accepted February 8, 2013. Date of publication March 7, 2013; date of current version May 22, 2013. This work was supported in part by ARO under Grant W911BF-12-1-0057 and Grant NSF IIS 1052851, the Faculty Research Awards by Google, FXPAL, NEC Laboratories of America, under Grant NSFC 61128007, the National Basic Research Program of China 973 Program under Grant 2012CB316400, the National Natural Science Foundation of China under Grant 61025011, the National Natural Science Foundation of China under Grant 61271435, the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under Grant IDHT20130225, and China Special Fund for Meteorological-scientific Research in the Public Interest under Grant GYHY201106044. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark H.-Y. Liao.

S. Zhang is with the Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China (e-mail: slzhang@jdl.ac.cn).

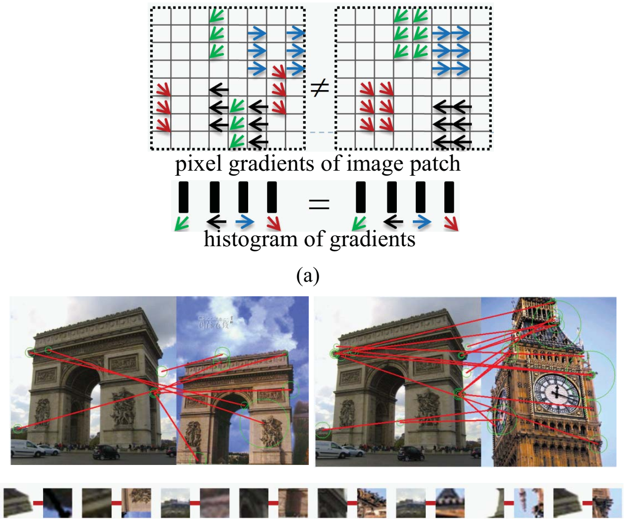
Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

K. Lu and Q. Huang are with the Graduate University of Chinese Academy of Sciences, Beijing 100080, China (e-mail: qmhuang@jdl.ac.cn; luk@gucas.ac.cn).

W. Gao is with the School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: wgao@pku.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.2013.2251650



(a)

(b)

Fig. 1. (a) Toy example showing the limited discriminative power of histogram-of-gradient-like feature in image patch matching. (b) Illustration of limited discriminative power of SIFT descriptor. SIFT descriptors between two images with similarity values (range between [0, 1]) larger than 0.75 are linked with red lines. The region pairs corresponding to the matched SIFT descriptors are also illustrated.

to achieve high performance in large-scale mobile partial-duplicate image search, efficient and discriminative BoWs representation is highly desired.

However, as illustrated in many works [3]–[12], traditional visual vocabulary shows poor discriminative ability, and needs extra efforts to improve its retrieval accuracy. Specifically, traditional visual vocabulary shows shortcomings such as containing lots of noisy visual words, losing spatial clues, large quantization errors, *etc*. To overcome these issues, existing works are mostly focused on three aspects, *i.e.*, visual vocabulary generation [3]–[6], indexing [7]–[9], and post verification in online retrieval [10]–[12].

For visual vocabulary generation, different algorithms have been proposed to depress noisy visual words and to improve the discriminative power. For example, rather than considering single local descriptors, Zhang *et al.*, propose to cluster local descriptor groups for visual vocabulary generation [3]. In this way, extra spatial contexts can be preserved in visual words. In some other works [4]–[6], visual word pairs or visual word combinations are extracted to improve the discriminative power of BoWs models. For image indexing, Jégou *et al.*, propose to compress local descriptors into compact codes for efficient image similarity computation [7]. For post verification in online retrieval stage, the goal is to identify and remove the mismatched visual words between images to improve the accuracy of similarity computation. For example in [10], Zhou *et al.*, propose a simple coding strategy to encode the spatial configurations of visual words in the index file. Then in the retrieval stage, an efficient verification is utilized to identify and remove the mismatched visual words according to the spatial consistency. Similarly, in some other works [11], [12], different spatial verification strategies are proposed to reduce the weights of mismatched visual words based on the spatial consistency.

Notwithstanding the demonstrated success of the above mentioned works, most of them are focused on the latter 3 steps, *i.e.*, visual vocabulary generation, offline image indexing, and online image retrieval, but pay relatively less attention to the potential issues of the local descriptors, such as SIFT [1], SURF [13], *etc*., which are the foundations of latter 3 steps. In addition, most of these existing works introduce additional computational burdens. For example, by combining multiple visual words, visual phrase [4]–[6] is more discriminative than single visual word, but is expensive to extract and select. Therefore, many of current works on large-scale partial-duplicate image search are not fitted for image search in computation sensitive scenarios.

As the basis for the above mentioned three steps, most of existing local descriptors are computed from the statistics of pixel gradients. For example, the 128-D SIFT is computed by dividing the local image patch into 16 (4 × 4) sub-regions, and extracting an 8D histogram of gradient from each [1]. Similarly, the 64D SURF is also extracted from 16 (4 × 4) sub-regions and each region produces a 4D descriptor by summarizing the wavelet responses in different orientations [13]. Although these descriptors divide the image patch into sub-regions to preserve spatial information, their statistical nature still loses significant spatial contexts and possibly results in low discriminative power for identifying near-duplicate patches.

As illustrated in the toy example in Fig. 1(a), two image patches show different spatial configurations, but have identical histogram of gradient features. Another example illustrating this issue of SIFT is given in Fig. 1(b). It can be clearly observed that, although many SIFT descriptors are matched, their corresponding image regions are actually not near-duplicate with each other, *i.e.*, mismatched regions. This observation partially explains why extra efforts are needed to incorporate more spatial clues [3] in traditional BoWs representation and depress the mismatched visual words with post spatial verification [10]. Hence, we come to the conclusion that histogram based descriptors perform well in visual classification and recognition tasks [14], [15], which are generally built on statistical features and statistical models of the whole image, but might still not be the most optimal feature for large-scale partial-duplicate image search, which is based on the near-duplicate local image patch matching.

To produce more effective BoWs representation for large-scale partial-duplicate image search on mobile platform, the desirable local descriptors should have three properties: 1) high discriminative power, *i.e.*, preserve spatial and visual contexts in image patches; 2) high efficiency, *i.e.*, extraction, similarity measurement, and matching should be efficient to compute; and 3) compactness, *i.e.*, descriptor should be compact to store and transmit. To achieve the three goals, we propose to extract local descriptors from edges. This is because edge has been proven discriminative for large-scale image retrieval in many works [16], [17]. Meanwhile, edge keeps rich spatial clues for image identification or matching. In Fig. 2, the edge maps of four images are illustrated. Obviously, although the binary edge maps lose some details such as texture, color, *etc*., they still preserve the spatial configurations of images and can

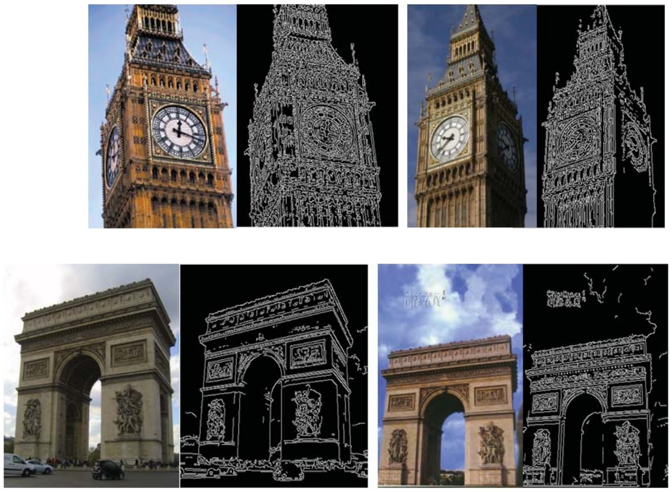


Fig. 2. Illustration of visually similar images and their binary edge maps.

easily identify the visually similar images. Hence, edge maps would be potentially valid for extracting discriminative local descriptor for image patch matching. Additionally, since edge map is binary and sparse, features extracted from it could be more efficient and compact, than the ones from original image patch.

Our proposed local edge descriptor is partially based on the work of SIFT [1], but is more discriminative, compact, and suitable for mobile partial-duplicate image search. Difference of Gaussian (DoG) detector in SIFT performs well in detecting repeatable and stable interest points [1], and also extracts reliable scale and orientation clues. Consequently, we first normalize the local image patches corresponding to the detected interest points into fixed scale to achieve scale-invariance. In similar way, we also achieve rotation invariance. On the normalized image patches, we first extract edges, and then record the locations and orientations of pixels on edges with a sparse binary descriptor. To depress the sparsity and make the binary descriptor more compact, we further propose a descriptor compression strategy, which selects the most discriminative bits and discards the sparse and noisy ones. To speed up the efficiency, a lookup table is constructed for fast descriptor similarity computation. Hence, the final local edge descriptor, namely *Edge-SIFT*, is designed to be discriminative, compact, and efficient.

Based on Edge-SIFT, we further develop an inverted file indexing framework which allows for fast online verification. We compare Edge-SIFT with SIFT and a recent binary local descriptor, *i.e.*, ORB [18] in large-scale partial-duplicate image retrieval tasks. Experimental results manifest that Edge-SIFT outperforms SIFT in the aspects of retrieval precision, efficiency, memory consumption, and transmission cost. Meanwhile, Edge-SIFT shows comparable retrieval efficiency and similar transmission cost with ORB [18], but achieves much better retrieval accuracy. Hence, we conclude that our work is more suitable for large-scale partial-duplicate image retrieval task on mobile platform than SIFT and ORB.

The contributions of this paper could be summarized into the following aspects.

1) The drawbacks of commonly used histogram based descriptors are analyzed. A novel binary descriptor is proposed for scalable partial-duplicate mobile visual search.
2) Edge-SIFT records the locations and orientations of edge pixels, thus preserves rich spatial clues and imposes more strict restriction on feature matching. Thus, Edge-SIFT performs better than SIFT in partial-duplicate image retrieval, which relies on accurate local feature matching.
3) Edge-SIFT is binary, compact and allows for fast similarity computation. It shows comparable efficiency and similar memory consumption with ORB but achieves much better retrieval accuracy.
4) A novel indexing framework with fast online verification is proposed. Experimental results verify the validity of our proposed feature and indexing algorithms.

The remainder of this paper is organized as follows. Section II introduces related works. Section III presents the Edge-SIFT descriptor extraction. Section IV presents descriptor compression and our proposed fast similarity computation strategy. Section V introduces the indexing and retrieval framework for Edge-SIFT. Section VI presents and discusses our experimental results, followed by the conclusions and future work in Section VII.

II. Related Work

Currently, mobile visual search has become a popular research topic for both the academic and industrial communities. The industrial community has already developed preliminary commercial systems such as Google “Goggles”, Amazon “Snaptell”, and Nokia “Point and Find”, *etc.* As for academic community, lots of efforts have been made to design mobile retrieval systems by compressing the commonly used image local descriptors or extracting novel compact descriptors. In the following part, we will review these works.

Proposed by Lowe in 1999, SIFT [1] descriptor is one of the most popular image local descriptor in computer vision. As a 128-D descriptor, SIFT is commonly represented as 1024 bits. A 400×300 sized JPEG image commonly contains more than 500 SIFT features. Obviously, the total size of the SIFT descriptors of one image (64 K-byte) is nearly the same as or even larger than the size of the image. Hence, SIFT and other similar descriptors like SURF [13], PCA-SIFT [19], Gradient Location and Orientation Histogram (GLOH) [20] are unsuitable for image search on mobile platform.

To design local descriptors for mobile applications, various methods have been proposed to compress original SIFT descriptors into compact codes that are faster and cheaper to transmit through wireless network. In [21], Chandrasekhar *et al.,* summarize different SIFT compression works published before 2010. The authors classify these works into three categories: vector quantization [3], [22], [23], transform coding [24], and hashing [25]–[29].

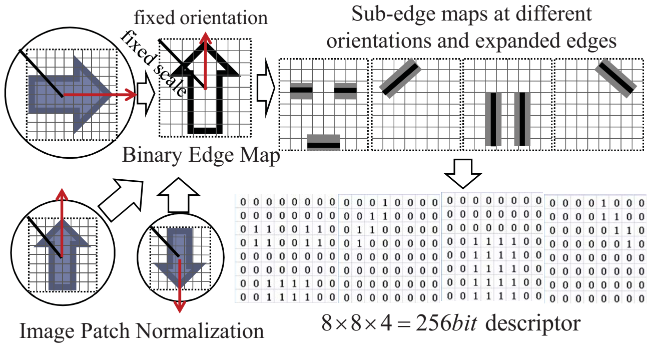
SIFT quantization with hierarchical visual vocabulary tree proposed by Nister *et al.,* is one of the most successful vector quantization strategy utilized in large-scale partial-duplicate image retrieval [3]. By searching hierarchically in the vocabulary tree, each SIFT descriptor could be represented as the ID code of its nearest visual word. For another example in [7], Jegou *et al.,* first aggregate the local descriptors into a long vector with a visual vocabulary. Then, they jointly optimize the dimension reduction as well as indexing with PCA and locality sensitive hashing to compress the long vector into compact codes which approximate the original neighborhood relationship. Compared with the other two compression strategies, vector quantization better preserves the discriminative power of SIFT. However, most vector quantization algorithms are relatively expensive to compute and need to load a visual vocabulary in the memory of mobile device, which is unfeasible for mobile applications if the visual vocabulary is too large.

Transform coding of SURF and SIFT descriptors is proposed by Chandrasekhar *et al.* [24]. Their compression pipeline first applies a Karhunen-Loeve Transform (KLT) to decorrelate different dimensions of the descriptor. This is followed by equal step size quantization of each dimension. Finally, entropy coding is applied to the quantized descriptor for storage or transmission. Authors verify that transform coding achieves 16× compression relative to original floating point representation [24]. However, transform coding schemes based on KLT shows relatively poor discriminative power. This might be due to the highly nonGaussian statistics of the SIFT descriptor [21].

Hashing based algorithms commonly convert the original SIFT descriptors into compact hash codes with multiple predefined hashing functions. Compared with vocabulary tree based compression, hashing functions consume less memory and are more efficient to compute than searching the nearest visual words hierarchically in the vocabulary tree. However, the authors of [21] conclude that hashing schemes like Locality Sensitive Hashing (LSH) [25], Similarity Sensitive Coding [26] or Spectral Hashing [27] show limited discriminative ability at low bitrates. Some recent works have been proposed to tackle this issue. For instance, Mu *et al.,* integrate the ideas of LSH and Random Forest to generate hashing codes. They verify that generating and aggregating multiple random hashing projections get comparable accuracy with vocabulary tree based algorithms, but with better space and time performance [28]. In a recent work [29], He *et al.,* add extra operations like geometric verification and boundary reranking to the hashing based search framework to improve the retrieval accuracy.

Other than the above compression strategies, many efforts have been made to design efficient and compact descriptors alternative to SIFT or SURF. Some researchers propose low-bitrate descriptors such as BRIEF [30], BRISK [31], CHoG [32], and ORB [18] which are fast both to build and match.

BRIEF descriptor is proposed by Calonder *et al.* [30]. 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 [31] first efficiently detects interest points in the scale-space pyramid based on a detector inspired by FAST [32] and AGAST [33]. 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 interest points. Hence BRISK achieves orientation-invariance and scale-invariance. Proposed in 2009, CHoG [34] first extracts histogram-of-gradient vector from a local neighborhood of each interest point. Then, the vector is further compressed to reduce the size. Moreover, the authors also propose an efficient similarity measurement strategy, with which the descriptors could be efficiently matched at the compressed domain. ORB descriptor is built based on BRIEF, and is more robust to the rotation and image noises. The authors demonstrate that ORB is two orders of magnitude faster than SIFT, while performs as well in many situations [18].


fixed orientation
fixed scale
Sub-edge maps at different orientations and expanded edges
Binary Edge Map
Image Patch Normalization
$8 \times 8 \times 4 = 256 bit$ descriptor

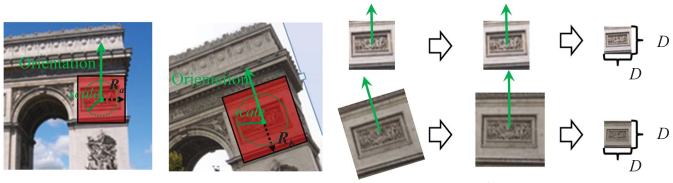
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Fig. 3. Proposed framework for Edge-SIFT extraction. We first normalize image patches into fixed scale and orientation for binary edge map extraction. Then, the positions and orientations of extracted edges are preserved in subedge maps, where edges are expanded to improve the robustness to registration errors. The conjunction of subedge maps is hence taken as the initial binary Edge-SIFT.

Compared with SIFT, despite of the clear advantage in speed, these compact descriptors show limitations in the aspects of descriptive power, robustness or generality. Moreover, similar to SIFT, many of the existing compact descriptors are also based on the statistic clues in the local image patches, hence also loose spatial clues that are important for discriminative power. Therefore, discriminative, efficient, and compact local descriptors are still high desired for mobile applications.

III. Edge-SIFT Extraction

The framework for Edge-SIFT extraction is illustrated in Fig. 3. As illustrated in the figure, we first extract image patches surrounding the interest points. Then, based on the scales and orientations of interest points, we normalize the image patches into fixed scale and orientation for edge map extraction. We then decompose the edge map into different sub-edge maps, according to the directions of edges. To make Edge-SIFT more robust to registration errors caused by inaccurate interest point localization, affine transformations, *etc*., we expand the edges in each sub-edge map to make edges in nearby locations can be considered for similarity computation. The conjunction of resulting sub-edge maps is hence taken as the initial Edge-SIFT.



(a) (b)

Fig. 4. (a) Illustration of image patch extraction. (b) Illustration of orientation normalization and scale normalization.

A. Interest Point Detection

We leverage the approach of SIFT for interest point detection. As illustrated by Lowe [1], this approach consists of three steps.

1) Scale-space extrema detection: a series of DoG images with different scales are computed. The sample points in DoG images that are larger or smaller than all of their 26 neighbors in both the current image and two adjacent images in scale space, are identified as candidate interest points. In this step, candidate interest points and their scale clues can be extracted.
2) Keypoint localization: more accurate interest point locations are assigned. Meanwhile, unstable points are discarded.
3) Orientation assignment: dominant orientations of interest points are computed by summarizing the pixel gradients in their corresponding image patches and selecting the most dominant directions. After interest point detection, we get the location, scale, and orientation of each interest point.

In the following steps, we introduce how we extract image patches and achieve scale invariance and rotation invariance.

B. Image Patch Extraction and Normalization

Based on the detected interest points, we first extract image patches around interest points, *i.e.*, as illustrated in Fig. 4(a). The size of extracted image patch corresponding to interest point i is defined as:

$$R_i = r \cdot \text{scale}_i \tag{1}$$

where *scale* denotes the scale of an interest point. It can be inferred that, larger r corresponds to larger image patches, which contain more edges and richer spatial clues, thus it is helpful to improve the discriminative power of the edge descriptor. However, larger r also increases the computational cost. In our experiment, we set r as 2.5. More details about r selection will be discussed in Section VI.

To achieve scale and rotation invariance, we normalize the extracted image patches into fixed orientation and scale. As illustrated in Fig. 4(b), the orientation is normalized by rotating the image patches to make sure their dominant orientations are identical. As for the scale normalization, we resize each image patch into a fixed size, *i.e.*, $D \times D$ sized patch. Larger D keeps more details, while smaller D corresponds to more compact descriptor. Therefore, the D should be carefully selected to achieve a good balance between discriminative power and compactness. We present more details about D selection in Section VI. Based on the normalized image patch, we introduce our edge descriptor computation in the following part.

C. Edge Descriptor Computation

From the $D \times D$ sized image patch, we first utilize Canny detector [35] for edge map extraction for its high efficiency and reasonably good performance. After edge extraction, image patches become $D \times D$ bit binary edge maps, where values of edge pixels are 1 or 0, otherwise. The edge map can be regarded as a binary local descriptor containing D^2 bits. Note that, extracting edge maps from scale normalized patches makes the response of Canny detector more stable and robust to image blur and scale changes than direct edge extraction from original images. Canny detector can be regarded as an edge filter with fixed scale and hence detecting edges on patches with similar scales results in more stable detection performance.

According to the edge pixel matching criteria, the similarity measure can be formalized as:

$$\text{Sim}(A, B) = 2 \cdot \sum_{i=1}^{D^2} (a_i \cdot b_i) \Big/ (\mathbb{N}_A + \mathbb{N}_B) \tag{2}$$

where A and B are two binary descriptors, a_i, b_i are the values of the ith bit in these two descriptors. $\mathbb{N}$ is the number of edge pixels *i.e.*, the nonzero bits, in a descriptor.

According to Eq. (2), the bits in the same locations of the edge map are matched for similarity computation without considering the orientation constraints, *i.e.*, edge pixels with different orientations can be matched. Intuitively, this strategy degrades the discriminative power of the binary local descriptor in identifying the near-duplicate image patches, which show nearly identical spatial configurations. Therefore, a better similarity measure should take the orientation constraint into consideration *i.e.*,

$$\begin{aligned} \widehat{\text{Sim}}(A, B) &= 2 \cdot \sum_{i=1}^{D^2} \widehat{\text{Hit}}(a_i, b_i) \Big/ (\mathbb{N}_A + \mathbb{N}_B) \\ \widehat{\text{Hit}}(a_i, b_i) &= \begin{cases} 1, & \text{if } a_i \cdot b_i = 1, \ \left|\theta_a^{(i)} - \theta_b^{(i)}\right| \le \varepsilon \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

where θ is the orientation of the edge pixel, and ε is a threshold. Intuitively, edge pixels in the same location with orientation difference smaller than ε would be considered as a match. In Eq. (3), the orientation of each edge pixel needs to be computed online, making it expensive to compute. One possible solution to speed up without losing orientation constraint, is to quantize the edge pixels in sub-vectors representing different orientations. Hence, similarity can by computed by summarizing the similarities between sub-vectors with similar orientations.

As illustrated in Fig. 3, we decompose the edge map into k ($k = 4$) sub-edge maps, representing k orientation ranges, *i.e.*, [0, 45), [45, 90), [90, 135), and [135, 180), respectively. Edge pixels in the original edge map will be assigned to

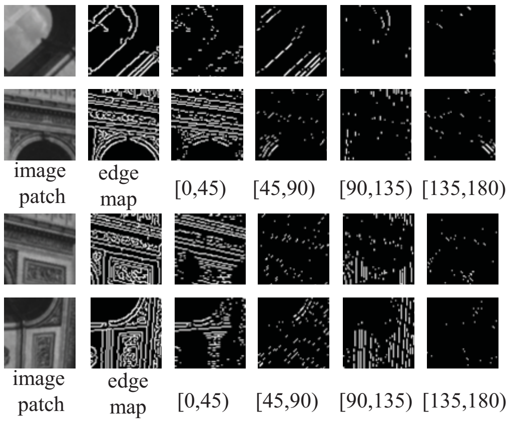


Fig. 5. Examples of extracted image patches, edge maps, and corresponding four subedge maps.

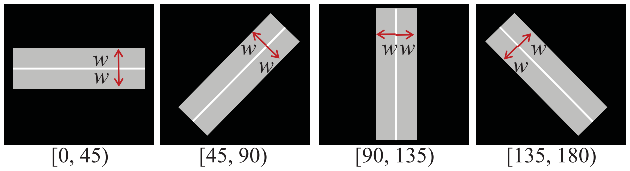


Fig. 6. Illustration of the proposed edge expansion strategy.

corresponding sub-edge maps according to their orientations. The concatenation of the four binary sub-edge maps would hence be regarded as a $D \times D \times 4$ bit local edge descriptor. The corresponding similarity measure between such two descriptors can be computed efficiently in similar way as Eq. (2). Note that, in descriptor extraction, we normalize the orientations of image patches. In addition, the orientation is coarsely quantized into 4 bins. This operation improves discriminative power without causing serious quantization errors. Thus the resulting Edge-SIFT is not sensitive to orientation changes.

Fig. 5 illustrates four image patches, their edge maps, and four sub-edge maps. It can be observed that, edge pixels are assigned to different sub-edge maps because they show different orientations. Therefore, during similarity computation, only edge pixels with both same locations and similar orientations are considered, which enhances the discriminative power of the edge descriptor.

Because the defined similarity measurement matches edge pixels in the same location, the proposed edge descriptor is sensitive to registration errors [1], which can be caused by affine transformations, inaccurate interest point localization, image noises, lighting changes, *etc*. Intuitively, if the centers of two near-duplicate patches are not aligned, their corresponding binary edge descriptors cannot be correctly matched. In the following step, we will show how we improve the robustness of the edge descriptor.

D. Robustness Enhancement

Edge-SIFT records the location and orientation of edge pixels, thus preserves rich spatial clues and imposes strict restriction on feature matching. However, strict restriction results in high precision but low recall rate in near-duplicate image patch matching. Thus, we relax the restriction with edge expansion to find a good trade-off between robustness (*i.e.*, recall) and discriminative power (*i.e.*, precision). One possible way to improve robustness to registration error is to loose the location constraint in similarity measurement, *i.e.*, edges from nearby locations could be also matched. The corresponding similarity measurement can be formalized as:

$$\widehat{\mathrm{Sim}}(A, B) = 2 \cdot \sum_{i=1}^{4\times D^2} \widehat{\mathrm{Hit}}(a_i, b_i) \Big/ (\mathbb{N}_A + \mathbb{N}_B)$$

$$\widehat{\mathrm{Hit}}(a_i, b_i) = \begin{cases} 1, & \text{if } a_i \cdot b_i = 1, \ \left| l_a^{(i)} - l_b^{(i)} \right| \le w \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

where l denotes the location of an edge pixel, and w is a threshold controlling the strictness of location constraint. It can be inferred that, (4) looses the location constraint. However, (4) is also more expensive to compute than (2). To achieve similar effects as Eg. (4) without degrading the efficiency, we propose an edge expansion strategy to acquire an improved edge descriptor.

The proposed edge expansion is illustrated in Fig. 6. As shown in the figure, the edge pixels in each sub-edge map are expanded in thickness to $2 \cdot w$ adjunct pixels in the directions *vertical* to their orientations. This is because edge matching is more sensitive to the registration errors occurred in the vertical directions of edges. The parameter w controls the strictness of location constraint, *i.e.*, number of expanded pixels. Intuitively, if w is too large, nearby edges can be mixed together, which degrades the discriminative power. Thus, w should be carefully selected. We set w as 1 and 2 for 1024 bit ($16 \times 16 \times 4$) and 4096 bit ($32 \times 32 \times 4$) edge descriptors, respectively. More details about w selection are given in Section VI.

After edge expansion, the resulting edge descriptor, *i.e.*, *initial Edge-SIFT*, would be a $D \times D \times 4$ bit binary vector, where D is the size of the edge map, and 4 denotes the number of sub-edge maps with different orientations. The similarity can be efficiently computed based on Eq. (2) with orientation constraint and loosen location constraint.

The effectiveness of the initial Edge-SIFT is illustrated in Fig. 7. It can be clearly observed that, most of the initial Edge-SIFT descriptors among the images are correctly matched. Some false matched initial Edge-SIFT pairs occur because their corresponding image patches contain similar edge clues. It is also obvious that the number of mismatched SIFT descriptors is generally more than the number of mismatched initial Edge-SIFT descriptors, which intuitively illustrates the discriminative power of initial Edge-SIFT. More evaluations and comparisons will be given in Section VI.

IV. Edge-SIFT Compression and Similarity Computation

From Fig. 5, it can be observed that the extracted initial Edge-SIFT is *sparse*, *i.e.*, lots of bins (bits) are 0-value. To make the final Edge-SIFT more compact and hence improve the efficiency of similarity computation, we propose to compress it.

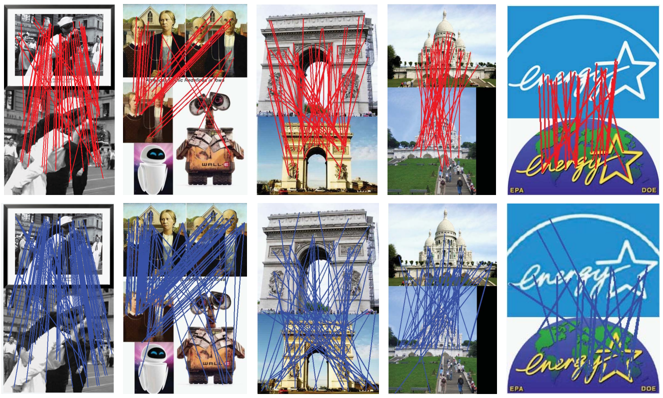
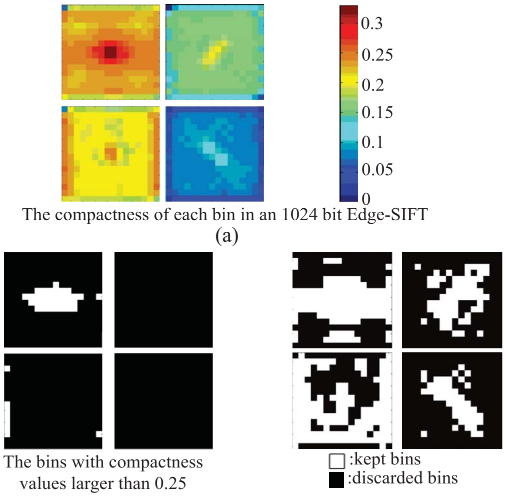


Fig. 7. Red lines: matched 1024-bit initial Edge-SIFT descriptors. Blue lines: matched SIFT descriptors. Between visually similar images.



(a)

(b)

Fig. 8. (a) Illustration of compactness of a 1024-bit initial Edge-SIFT computed from 6412 images. (b) 384-bit Edge-SIFT compressed from an 1024-bit initial one with discriminative bins selection.

Intuitively, a binary vector can be compressed by removing the sparse bins, which consistently shows 0-value. We define the compactness of the kth bin in the initial Edge-SIFT descriptor with Eq. (5), *i.e.*,

$$\chi_k = \sum_{i=1}^{N} v_k^{(i)} \Big/ N \tag{5}$$

where N denotes the total number of collected edge descriptors from a dataset, and $v_k^{(i)}$ is the value of the kth bin in the ith descriptor. Therefore, we can set a threshold for descriptor compression. Specifically, bins with compactness below the threshold will be discarded. The compactness of each bin computed from a dataset containing 6412 images is illustrated in Fig. 8(a). It can be seen that, if the threshold is set as 0.25, only 41 bins could survive the compression operation.

A. Discriminative Bins Selection

According to the above mentioned strategy, to compress initial Edge-SIFT as well as to preserve its discriminative power, we need to choose an ideal threshold. However, such threshold is hard to decide. In addition, it is necessary to point out that, some of the sparse bins might be discriminative for certain kinds of image patches, and they cannot be kept by simply setting a threshold. To conquer this issue, we first select several initial bins with high compactness from initial Edge-SIFT, and then identify and add discriminative bins to get the final compressed Edge-SIFT.

We use RankBoost [36] to select the discriminative bins. Similar to the feature selection strategy in [37], we collect a dataset set, where the relevance degrees between images are labeled. Based on this dataset set, we construct many lists of ordered images, which are hence adopted as training data. In each iteration of discriminative bins selection, RankBoost finds a most discriminative bin and constantly adds it to the initial bins. To test the discriminative power of a certain bin to be selected, we add it to the descriptor containing the already selected bins to get a new descriptor. The new descriptor is hence utilized to update relevance degrees computed between images. The discriminative bins should be the ones with which the number of disordered training images are decreased. More details about RankBoost can be found in [36].

The relevance degrees between two images A and B is computed according to the number of matched descriptors between them, *i.e.*,

$$\begin{aligned}\mathrm{Rel}(A, B) = \frac{1}{2} \cdot \Bigg(& \sum_{k=1}^{N_A} \mathrm{Match}(d_A^{(k)}, B) \\ & + \sum_{k=1}^{N_B} \mathrm{Match}(d_B^{(k)}, A) \Bigg)\end{aligned} \tag{6}$$

$$\mathrm{Match}(d_A, B) = \begin{cases} 1, & \text{if } \frac{\mathrm{closestsim}(d_A, B)}{\mathrm{secondsim}(d_A, B)} \geq \phi \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where N is the total number of local descriptors, *i.e.*, d, in an image. $\mathrm{closestsim}(d_A, B) \big/ \mathrm{secondsim}(d_A, B)$ denotes the similarity comparison between the closest neighbor and the second-closest neighbor of a descriptor d_A from image A in image B. ϕ, which is set as 1.5, is the threshold controlling the strictness of feature matching. As discussed in [1], the matching criterion in Eq. (7) is better than setting a global threshold on similarity to the closest neighbor.

Note that, the iteration terminates if no more discriminative bins can be found. Fig. 8(b) illustrates a 384 bit Edge-SIFT compressed form a 1024 bit initial one. It can be observed that, many bins with small compactness values are also selected. The validity of compressed Edge-SIFT will be tested in Section VI.

Suppose we select n initial compact bins, by running the iteration for m times, we obtain a $m+n$ bit compressed Edge-SIFT. The number of m can be flexibly adjusted to seek a tradeoff between compactness and discriminative power. In our experiment, we set the values of $m + n$ as integer multiples of 8, *i.e.*, one byte. Therefore, each compressed Edge-SIFT can be represented with several bytes without wasting storage space. In the next step, we will present an efficient lookup table based similarity computation for compressed Edge-SIFT matching.

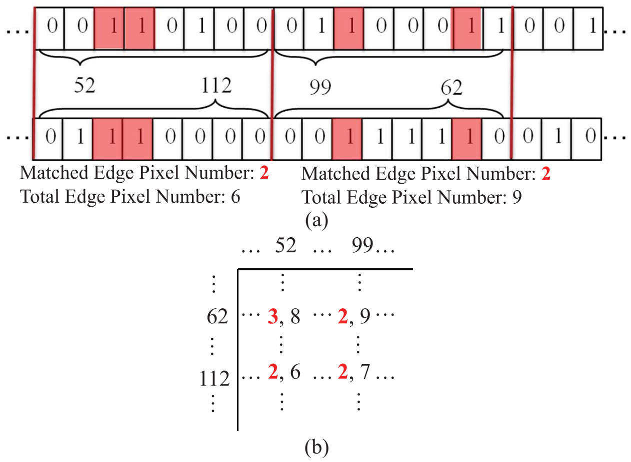


Fig. 9. (a) Illustration of the code representation and corresponding similarity computation. (b) Illustration of the lookup table.

B. Lookup Table-Based Similarity Computation

As mentioned above, each compressed binary Edge-SIFT can be represented as a list of basic units, *i.e.*, bytes. Each byte can be represented as an integer code with value ranges between [0, 255]. Therefore, as illustrated in Fig. 9(a), the similarity computation between two Edge-SIFT descriptors is transformed as the similarity computation between two lists of integer codes. We hence formulate the similarity computation as, *i.e.*,

$$\mathrm{FastSim}(A, B) = \frac{2 \cdot \sum_{i=1}^{U} \mathrm{MEPN}\left(C_A^{(i)}, C_B^{(i)}\right)}{\sum_{i=1}^{U} \mathrm{TEPN}\left(C_A^{(i)}, C_B^{(i)}\right)} \qquad (8)$$

where U is the number of integer codes, *i.e.*, C in Edge-SIFT. Suppose the size of an Edge-SIFT is 384 bit, its U would be 48. MEPN(·,·) and TEPN(·,·) return the Matched Edge Pixel Number and Total Edge Pixel Number in two codes, respectively.

According to its formalization, (8) is based on two aspects of clues, *i.e.*, MEPN and TEPN (illustrated in Fig. 9(a)). Because the value of each code ranges between [0, 255], these two aspects of information can be stored in a lookup table (*i.e.*, illustrated in Fig. 9(b)) for fast similarity computation. Specifically in Fig. 9(b), we construct a 256×256 sized table, where the item in 62-th row and 52-th column contains the Matched Edge Pixel Number and the Total Edge Pixel Number of code pair (62, 52). Therefore, the similarity between two Edge-SIFT descriptors can be efficiently computed by looking up the table with a list of integer codes, and making comparison between the accumulated MEPN values and TEPN values. Because there is no need to scan each bit in the descriptor, the computation efficiency is expected to be largely improved.

V. Indexing and Retrieval

In this section, we study the way to index large-scale images based on the compressed Edge-SIFT. BoWs representation and inverted file indexing have demonstrated great success in large-scale image search [3]–[12]. Therefore, we also design our indexing and retrieval system based on the BoWs and inverted file indexing framework.

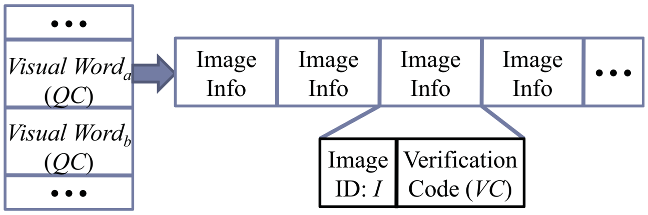


Fig. 10. Proposed indexing framework.

To generate BoWs representation, we first quantize Edge-SIFT into code words.To achieve this, we could utilize different strategies such as Hashing [26], vocabulary tree [2], *etc*. Compared with traditional vocabulary tree, the advantages of Hashing are low memory consumption and high efficiency. However, many experiments manifest that vocabulary tree is simpler, more robust, and performs better than hashing in many situations [21], [25], [27]. Because Edge-SIFT is compact, *i.e.,* 384 bit, and is efficient for similarity computation, we expect to get a compact and efficient vocabulary tree suitable for mobile applications.

Visual vocabulary tree can be generated through clustering Edge-SIFT descriptors with the defined similarity measurement. As a popular clustering algorithm, hierarchical K-means is generally efficient for visual word generation. However, with the lookup table based similarity measurement, it is hard to compute the cluster centers for K-means clustering. Therefore, we use K-centers clustering instead. Different form K-means, the cluster center of K-centers is simply updated as the data point having the maximum similarities with the other data points in the same cluster.

BoWs representation is computed by quantizing local features into visual words. Hence, quantization error is inevitable and may degrade the retrieval performance. To decrease quantization error, we divide Edge-SIFT after discriminative bins selection into two parts: the former selected α bins are called as *Quantization Code* (*QC*) and the latter selected β bins are called as *Verification Code* (*VC*). QC is utilized for visual vocabulary tree generation and Edge-SIFT quantization, *i.e.*, BoWs representation computation. VC is kept in the index file for online verification. Fig. 10 shows the corresponding indexing framework.

As illustrated in Fig. 10, our indexing strategy is based on the standard inverted file indexing framework. Differently, each term in the index list contains extra verification code for online verification. Our corresponding online image similarity computation can be represented as:

$$S(Q, D) = \sum_i \mathrm{IDF}_i \cdot \mathrm{FastSim}(VC_i^Q, VC_i^D) \qquad (9)$$

where Q and D are query and one of the database images respectively. i denotes one of their matched visual words. IDF_i means the Inverse Document Frequency of visual word i in the image index. VC_i^Q is the auxiliary code of the Edge-SIFT descriptor in image Q, whose main code is quantized as visual word i.

Collected images with keyword "Tengwang Tower"
Collected images with keyword "Coliseum"
Collected images with keyword "Confucius Temple"
Collected images with keyword "Dongguan Qingzhen Temple"
Collected images with keyword "Eiffel Tower"

Fig. 11. Illustration of the collected landmark dataset.

Note that in our indexing framework, we do not record the Term Frequency (TF) clue. This is because local features from the same image can rarely be quantized into the same visual word, if a large visual codebook (*i.e.*, 1M visual words) is utilized. Similar strategy is also adopted in many other works [10], [38]. In our experimental setting, if multiple features in the same image are quantized into the same visual word, we pick up the local feature with the largest scale for indexing or retrieving, and discard the rest ones.

Suppose the mobile image retrieval is implemented based on a client-server architecture, where the server maintains an image index and the mobile device uploads queries and receives retrieval results. With the proposed retrieval framework, two kinds of information should be sent for query from mobile devices, *i.e.*, visual word ID and VC of each Edge-SIFT. Hence, keeping a larger VC would potentially improve the retrieval performance, but produces more transmission cost. In addition, larger VC may make QC too sparse to generate valid visual vocabulary and BoWs representation. Hence, parameters α and β can be flexibly adjusted to chase a reasonable trade-off between retrieval accuracy and transmission cost. In the next section, we will test Edge-SIFT and retrieval framework.

VI. Experiments and Evaluations

A. Dataset and Experimental Setup

We use Oxford Building [39] for testing the effects of different parameters and evaluating the validity of Edge-SIFT compression. This dataset contains 55 queries and 5062 images. The relevance degrees between queries and dataset images have been manually annotated. To train discriminative bins selection for Edge-SIFT compression, we use the Paris dataset [40] as the training set. This dataset contains 6412 images. Similar to the Oxford Building, the queries and corresponding ground truth are also available in the Pairs dataset.

To test the performance of Edge-SIFT in large-scale image retrieval, we collect a dataset containing 1 million images. This is finished with a similar process of the one in bundled feature [11]. Besides that, we also collect a landmark dataset from the Google Image by searching landmarks such as "Eiffel Tower", "Big Ben", *etc.* From the downloaded images, we select 40 categories, within each category we keep 150 relevant images. Therefore, we get a dataset containing 6000 images with ground truth. This dataset is mixed with the 1 million Internet images for large-scale image retrieval. During the retrieval process, the landmark images are adopted as queries, and are considered to evaluate the retrieval performance. Examples of the landmark dataset are illustrated in Fig. 11.

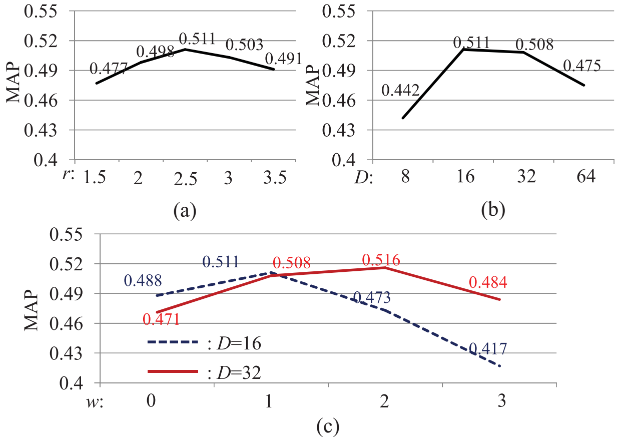


Fig. 12. Illustration of the effects of (a) r, (b) D, and (c) w.

We conduct our experiments on a server with 8-core 2.4 GHz CPU and 8 GB memory. Mean Average Precision (MAP), which takes the average precision across all different recall levels is adopted to evaluate the retrieval performance.

B. Parameter Selection

The initial Edge-SIFT is related to the three parameters: r, which controls the size of the extracted image patch; D, which decides the size of the edge map; and w which controls the edge expansion. We test the effects of these parameters in image retrieval tasks. Specifically, we use initial Edge-SIFT, BoWs representation, classic inverted file, and TF-IDF weighting for image indexing and retrieval, rather than the proposed framework in Section V. We set the number of visual words as 53 K, *i.e.*, a four-level visual vocabulary tree, each level has 27 branches [2]. In the followings, we will test the effects of these parameters, *i.e.*, 5 values for r, 4 values for D, and 4 values for w. It is desirable to repeat the experiment for $5 \times 4 \times 4$ times to seek the optimal parameter combination. However such experiment is too time consuming. We refer to the experimental setting of SIFT [1] and assume the three parameters are relatively independent of each other. Hence, when a parameter is discussed, we fix the other two to default values, which are set as 2.5, 16, and 1, respectively.

The effects of r are illustrated in Fig. 12(a). From the figure, we can observe that larger r is helpful for performance improvement, this is because larger image patches contains richer edge clues, which make edge descriptor more discriminative. However, increasing r does not consistently improve the performance. This might be because larger image patches also include more unstable edges which are far away from the stable interest points, which would introduce more noises. Therefore, we set the parameter r as 2.5, which shows reasonably good performance and high efficiency.

The effects of D are illustrated in Fig. 12(b). It is clear in the figure that, the retrieval performance degrades, if D is too large or too small. Intuitively, small D results in compact descriptor, however also loses too much information in image patches. When D is too large, the performance also drops. This might be because, if the descriptor dimension is too high, the descriptor would be more sensitive to image noises and registration errors.

The effects of w are illustrated in Fig. 12(c). It can be observed that edge expansion is helpful to improve the performance. However, when w is too large, the adjacent edges can be mixed with each other, which degrades the discriminative power. Meanwhile, the validity of edge expansion is closely related to the edge map size. In the following experiments, for $16 \times 16 \times 4$ bit initial descriptor, we set the value of w as 1; while for $32 \times 32 \times 4$ bit initial descriptor, we set the value of w as 2.

C. Validity of Edge-SIFT Compression

After selecting the parameters, we hence compress the initial Edge-SIFT and select the discriminative bins. We test two types of initial Edge-SIFT descriptors: a 1024 bit one whose r, D, and w are 2.5, 16, and 1, and another 4096 bit one whose r, D, and w are 2.5, 32, and 2, respectively. We first compress the two descriptors to 32 bit and 64 bit by selecting compact bins. Then, we run the discriminative bins selection from the left 992 bins and 4032 bins, respectively. The discriminative bin selection process finally converged and the final sizes of compressed Edge-SIFT descriptors are 412 bit and 832 bit, respectively. In the followings, compressed Edge-SIFT with different numbers of selected bins are tested on Oxford Building dataset.

We generate 53 K visual words for Edge-SIFT descriptors with different sizes. One Edge-SIFT with different sizes: 32, 64, 128, 256, and 384 bits, compressed from the initial 1024 bit one, and the other Edge-SIFT with 64, 128, 256, 384, 512, 640, and 768 bits, compressed from the initial 4096 bit one are compared in Fig. 13.

Clearly from the figure, as more bins are added to the compressed descriptors, their retrieval performances are improved remarkably. This proves the validity of our discriminative bins selection strategy. The compressed Edge-SIFT from the 4096 bit descriptor finally outperforms the one from the 1024 bit descriptor. This shows that larger descriptor contains richer clues, thus more discriminative bins can be selected. It can be also observed that, two compressed descriptors finally outperform their initial descriptors with more compact sizes. In addition, it is necessary to note that the MAP of SIFT descriptor with 53 K visual word is 0.513. Thus, the compressed Edge-SIFT descriptors also finally outperform the SIFT. More detailed comparisons with SIFT will be given in the next part.

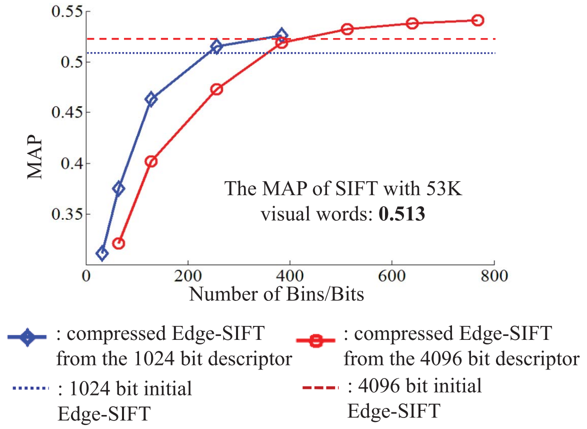


Fig. 13. Illustration of the validity of the descriptor compression.

Generally, we can conclude that our descriptor compression strategy produces discriminative and compact Edge-SIFT. It should be noted that, the performance of Edge-SIFT on Oxford Building is not the state-of-the-art. However, the focus of this paper is different from the ones of most image retrieval works which are built upon SIFT and introduce extra operations such as spatial verification. It is interesting to combine Edge-SIFT with the other retrieval algorithms in our future work for a better performance.

D. Experiments on Large-Scale Image Retrieval

In this experiment, we test compressed Edge-SIFT in large-scale image retrieval tasks with our proposed indexing framework and make comparisons with SIFT [1] and ORB [18]. We make comparisons with SIFT because it is the most commonly used descriptor for large-scale image retrieval [2], [3], [5]–[12], [38]–[40], and it shows superior performance to most of the other descriptors [20], such as PCA-SIFT [19], shape context [17], SURF [13], *etc.* ORB is a recent scale and rotation invariant binary descriptor and it shows superior performance than the BRIEF [30] and comparable performance with SIFT in many cases [30]. We first compare the following four features in terms of MAP and efficiency:

F1: 1024 bit SIFT descriptor, *i.e.*, 8 bit ×128 dimensions

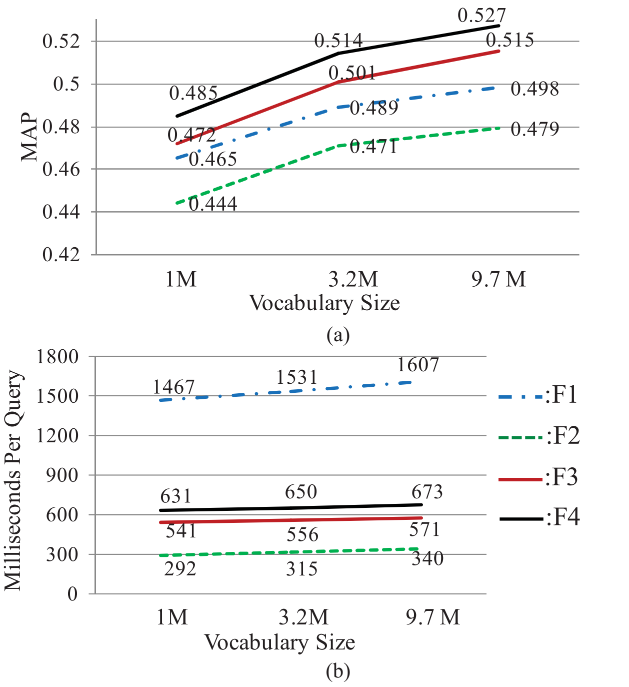
F2: 256 bit ORB descriptor

F3: 384 bit Edge-SIFT compressed from the 1024 bit initial descriptor. We use 256 bits QC and 128 bits VC, *i.e.*, $\alpha = 256$, $\beta = 128$.

F4: 768 bit Edge-SIFT compressed from the 4096 bit initial descriptor. We use 640 bits QC and 128 bits VC, *i.e.*, $\alpha = 640$, $\beta = 128$.

We generate visual vocabularies by hierarchically clustering [2] the descriptors into 5-level vocabulary trees. We set the branch numbers as 16, 20, and 25, corresponding to three vocabulary sizes: 1M, 3.2M and 9.7 M, respectively. Comparisons among the four features are illustrated in Fig. 14.

Fig. 14(a) shows the MAPs of the four features. From the figure, we can observe that the proposed Edge-SIFT performs better than SIFT in large-scale image retrieval with more compact descriptor size. This is mainly because of three reasons: 1) more spatial clues are preserved in Edge-SIFT by recording both locations and orientations of edges which impose more strict restriction on feature matching; 2) discriminative bins



(a)

(b)

Fig. 14. Comparisons of MAP and efficiency with different vocabulary sizes.

are preserved and noisy ones are discarded in the descriptor compression step; and 3) we keep VC in the index file for online verification, which decreases the quantization error in BoWs model and potentially decreases the mismatched local features. It also can be seen that ORB performs worse than SIFT and Edge-SIFT. This might be because ORB uses relatively weak interest point detector and loses too many image details with very compact representation. It is clear that, as we increase the branch number in the vocabulary tree, the performance of BoWs based image retrieval improves. This is because the feature space is quantized to finer scales, which increases the discriminative ability of visual words.

Fig. 14(b) shows the average retrieval time based on the four descriptors, which mainly contains four parts: interest point detection, descriptor computation, BoWs representation computation, and image ranking. Obviously from the figure, Edge-SIFT is remarkably more efficient than SIFT. This is mainly because Edge-SIFT needs less time for BoWs representation computation and is faster to extract than SIFT. In order to compute the BoWs representation, the QC of each Edge-SIFT has to find a nearest visual word in the hierarchical vocabulary tree. In this process, the similarities between QC and visual words can be computed efficiently with lookup table. This is faster than computing the Euclidean distance for SIFT similarity computation. It also can be observed that ORB is the fastest feature. This is largely because it uses faster interest point detector. However, ORB loses important image clues and gets poor retrieval accuracy.

Because mobile application is sensitive to the transmission cost, in the followings, we further compare the four features in the aspect of the size of transmitted data for each local feature. Note that, in traditional BoWs image retrieval framework, for each local feature, local device needs to transmit two kinds of clues: visual word ID (32 bits) and the Term Frequency (TF, 32 bits) to the server. Differently, in our proposed retrieval system, we need to transmit visual word ID (32 bits) and the VC (β bits) to the server. Hence, as more bits are preserved in VC, the transmission cost would increase. Fig. 15 compares the retrieval performance of SIFT, ORB, and Edge-SIFT with different VC settings. In this figure, we set the vocabulary size as 3.2 M. Fig. 15 also reasonably reflects the comparison of time for data transmission among three features, *i.e.,* the larger transmission cost, the more time would be required.

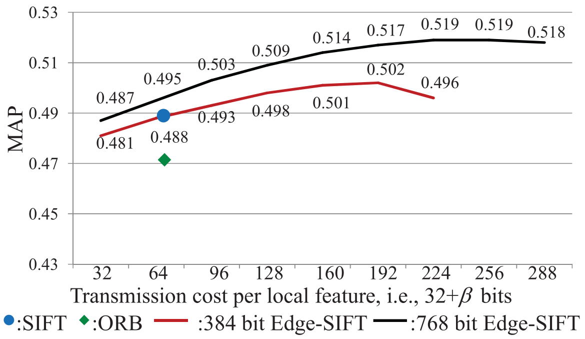


Fig. 15. Comparisons of transmission cost with a 3.2 M visual vocabulary.

From the figure, it is clear that our propose retrieval system shows comparable performance with SIFT and obvious advantage over ORB, when the VC is set to 0 bit, *i.e.*, the entire Edge-SIFT descriptor is utilized for vocabulary tree generation and BoWs representation computation. In this case, we achieve lower transmission cost than SIFT and ORB. It is obvious that, as we increase the size of VC, the performance of our algorithm increases remarkably. Hence, with our proposed retrieval system, the trade-off between transmission cost and retrieval accuracy can be flexibly adjusted according to users' requirements. From the figure we also observe that, the retrieval performance would remain the same or even drop, if VC is too large. This might be because too compact QC would be unstable and may cause large quantization error during the visual vocabulary generation process, which cannot be compensated by the latter online verification step.

From the above comparisons, we conclude that Edge-SIFT gets comparable efficiency and transmission cost with ORB but with much better retrieval accuracy. Moreover, we also conclude that our proposed retrieval framework based on Edge-SIFT outperforms the traditional BoWs retrieval system based on SIFT, in the aspects of retrieval accuracy, efficiency, transmission cost, and memory consumption.

Fig. 16 shows some examples of Edge-SIFT based partial-duplicate image retrieval. It can be observed that, although the images are edited by affine transformations, cropping, and cutting, *etc*., they still can be retrieved by Edge-SIFT. The images that can not be retrieved by SIFT are highlighted by color boxes. It can be inferred that SIFT fails with large cropping, cutting or obvious transformations, which may introduce more cutter background, and noisy visual words.

E. Discussions and Future Work

In addition to the advantages, we must address the limitations and challenging issues with our schemes, as well as provide feasible directions for solutions in our future work.

One of the limitations is that, the adopted DoG interest point detector is relatively expensive to compute, which degrades

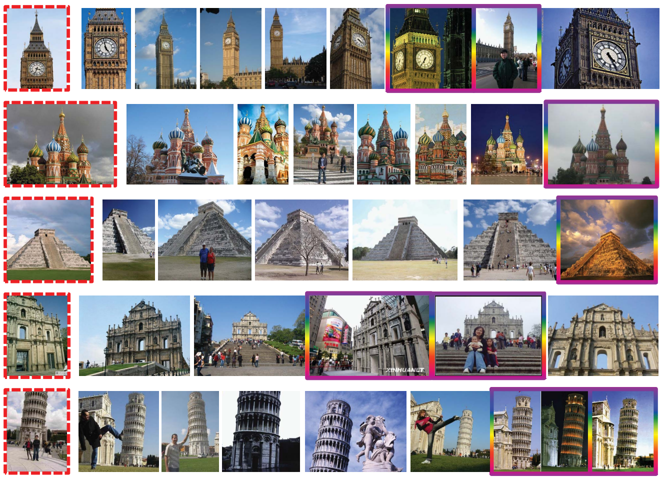


Fig. 16. Results of Edge-SIFT-based partial duplicate image retrieval. The most left images are the queries. Returned images before the first false positive are shown. Images cannot be retrieved by SIFT are highlighted by color boxes.

the efficiency of Edge-SIFT extraction. Therefore, Edge-SIFT is not as efficient as ORB, which is based on faster FAST detector [30]. We use DoG mainly because it produces more reliable position, scale and orientation clues, which largely decrease the registration error. To further improve the efficiency of Edge-SIFT, it is necessary to study more efficient interest point detectors that identify stable regions with rich edge clues. The desirable detector is also expected to extract scale and orientation clues and be able to estimate the affine transformations. Hence, the final Edge-SIFT could be adjusted to be more robust to affine changes. One possible solution is to detect stable corner points from the binary edge maps.

Edge-SIFT is built on binary edge maps, therefore its performance is also related to the robustness of Canny edge detector. Canny detector can be regarded as a detector with fixed scale, thus it is sensitive to image blur and scale changes. In Edge-SIFT extraction, we extract edges from scale-normalized patches, rather than original images to depress the effects of image blur and scale change. Moreover, Canny detector could overcome image blur and illumination change to some extent with nonmaximum suppression, which effectively converts the blurred edges into sharp edges and hysteresis thresholding, which avoids identifying edge pixels simply based on their gradient magnitudes [35]. However, image blur and significant illumination change are still harmful to Edge-SIFT. Intuitively, image blur not only erases edge details but also decreases the number of detected interest points. Significant illumination change may also degrade the performance of DoG detector and change the appearance of image patches. Actually, image blur and significant illumination change are challenging issues for most of existing image descriptors. It is still a challenging research topic to develop interest point detector and descriptor robust to these two issues.

Canny detector is controlled by three parameters, *i.e.*, low hysteresis threshold, high hysteresis threshold, and size of Gaussian kernel which controls the amount of smoothing [35]. In this paper, we do not specifically tune the three parameters and set them to fixed values, *i.e.*, 30, 90, and 3, respectively, by referring to related works and the recommendation of OpenCV [35], [41]. However, as discussed by Heath, *et al.* [41], tuning parameters largely improves the performance of Canny detector. Moreover, Nalwa–Binford detector could be a better choice if a fixed set of parameters is required [41]. Hence carefully tuning the edge detector would further improve the performance of Edge-SIFT.

Experimental results show that the proposed robustness enhancement strategy improves the robustness of Edge-SIFT without degrading the efficiency. However, simply expanding the edge might be not the optimal solution and it would be harmful for the discriminative power. In our future work, better solutions will be further explored.

Based on 384 bit Edge-SIFT, the size of vocabulary with 1M visual words is about 40MB, which is manageable for most of mobile devices. However, if the vocabulary size is too large, the size of vocabulary would beyond the memory capacity of mobile devices. Hence, strategies should be studied to conquer this issue. One possible solution is to design hashing based vector quantization. Another solution is to further compress Edge-SIFT into more compact binary codes, *e.g.*, 20 bits that represent 1.05 M unique IDs. Hence, the values of the code could be the straightforwardly utilized as visual word IDs.

Conducting experiments on mobile platforms requires lots of engineering implementation and optimization, which is beyond the scope of this paper. Consequently, we use standard PC to simulate the mobile platform to compare Edge-SIFT with SIFT and ORB in the aspects of retrieval accuracy, efficiency, and data transmission. Because transporting these descriptors to the mobile platform does not change the computation or memory complexity, we could reasonable draw the conclusion that Edge-SIFT is superior to the other two descriptors in mobile visual search. However, it is still desirable to test Edge-SIFT on real mobile platforms. This will be our future work.

VII. Conclusion

In this paper, we propose a novel edge based local descriptor called Edge-SIFT. Different from traditional histogram of gradient based descriptor, Edge-SIFT is built upon edge maps of local image patches and keeps both locations and orientations of edges. In order to make Edge-SIFT more robust and more compact, we further study edge expansion and discriminative bins selection strategies. To utilize Edge-SIFT in large-scale partial-duplicate mobile search, we further propose an inverted file based indexing framework, which allows for flexible online verification. The proposed descriptor is compared with SIFT and ORB in large-scale partial-duplicate mobile search tasks. Experimental results reveal that, Edge-SIFT shows better efficiency, retrieval precision, and compactness than SIFT. Edge-SIFT also shows remarkably better retrieval precision than ORB with similar transmission cost and comparable retrieval efficiency. Hence, we conclude that, Edge-SIFT is compact, efficient, and discriminative, our retrieval system is accurate and efficient for large-scale mobile partial-duplicate image retrieval.

With rich spatial clues and compact representation, Edge-SIFT allows for accurate and efficient near-duplicate image patch matching. Hence we naturally utilize it in large- scale partial-duplicate mobile search. Although it is not as general as SIFT and SURF, which are capable for various tasks, *i.e.*, visual recognition, classification, *etc*., Edge-SIFT still can be further utilized in more tasks relying on image patch matching. For example, Edge-SIFT can serve as a better alternative to existing descriptors for image matching, the basis of generating panoramic views from images, like the ones of Google Street View. It also can be used for efficient landmark 3D construction, which builds 3D models by collecting and matching partial-duplicate landmark images [42]. The properties of Edge-SIFT, *i.e.*, fast and accurate image patch matching, warrant further study and utilization of Edge-SIFT in different tasks in the future.

References

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

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

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

[4] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[5] 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.

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

[7] H. Jégou, M. Douze, C. A. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[8] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognit. Lett.*, vol. 31, no. 11, pp. 1348–1357, Aug. 2010.

[9] Y. Mu, J. Sun, T. Han, L. Cheong, and S. Yan, "Randomized locality sensitive vocabularies for bag-of-features model," in *Proc. 11th Eur. Conf. Comput. Vis. Conf. Comput. Vis.*, 2010, pp. 748–761.

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

[11] Z. Wu, Q. F. Ke, and J. Sun, "Bundling features for large-scale partial-duplicate web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 25–32.

[12] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large-scale image search," in *Proc. 10th Eur. Conf. Comput. Vis., Part 1*, 2008, pp. 304–317.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Mar. 2008.

[14] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2169–2178.

[15] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2005, pp. 1458–1465.

[16] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Edgel index for large-scale sketch-based image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 761–768.

[17] A. Belongie, J. Malik, J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

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

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

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

[21] V. Chandrasekhar, M. Makar, G. Takacs, D. Chen, S. Tsai, M. Cheung, R. Grzeszczuk, Y. Reznik, and B. Girod, "Survey of SIFT compression schemes," in *Proc. Int. Mobile Multimedia Workshop, IEEE Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1–8.

[22] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 304–317.

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

[24] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, and B. Girod, "Transform coding of feature descriptors," *Proc. SPIE, Int. Soc. Opt. Photon.*, vol. 7257, p. 725710, Jan. 2009.

[25] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 217–220.

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

[27] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, Dec. 2008.

[28] Y. Mu, J. Sun, T. Han, L. Cheong, and S. Yan, "Randomized locality sensitive vocabularies for bag-of-features model," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 748–761.

[29] J. He, J. Feng, X. Liu, T. Cheng, T. Lin, H. Chung, and S. Chang, "Mobile product search with bag of hash bits and boundary reranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2012, pp. 3005–3012.

[30] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.

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

[32] E. Rosten and T. Drummond, "Machine learning for highspeed corner detection," in *Proc. 9th Eur. Conf. Comput. Vis.*, May. 2006, pp. 430–443.

[33] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 183–196.

[34] 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.

[35] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[36] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, Nov. 2003.

[37] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-bag-of-features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3352–3359.

[38] W. Zhou, Y. Lu, H. Li, and Q. Tian, "Scalar quantization for large scale image search," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 169–178.

[39] 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.

[40] 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.*, Jun. 2007, pp. 1–8.

[41] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors: A methodology and initial study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1996, pp. 143–148.

[42] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," in *Proc. 10th Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 427–440.

Shiliang Zhang received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently a Post-Doctoral Research Fellow with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. His current research interests include large-scale image and video retrieval, image/video processing, and multimedia content affective analysis.

Dr. Zhang was the recipient of the Excellent Graduate Award from the Chinese Academy of Sciences, ACM Multimedia Student Travel Grants, and the Microsoft Research Asia Fellowship 2010.

Qi Tian (M'96–SM'03) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, the M.S. degree in electrical and computer engineering from Drexel University, Philadelphia, PA, USA, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, Urbana, IL, USA, in 1992, 1996, and 2002, respectively.

He is currently an Associate Professor with the Department of Computer Science, University of Texas at San Antonio (UTSA), San Antonio, TX, USA. He took a one-year faculty leave with Microsoft Research Asia from 2008 to 2009. He has authored over 180 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA. His current research interests include multimedia information retrieval and computer vision.

Dr. Tian was a recipient of the faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He was also a recipient of the Best Paper Awards in 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, the 2010 ACM Service Award. He is the Guest Editors of the IEEE TRANSACTIONS ON MULTIMEDIA, the *Journal of Computer Vision and Image Understanding*, *Pattern Recognition Letter*, the *EURASIP Journal on Advances in Signal Processing*, the *Journal of Visual Communication and Image Representation*, and is on the Editorial Board of IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the *Multimedia Systems Journal*, the *Journal of Multimedia*, and the *Journal of Machine Visions and Applications*.

Ke Lu was born in Ningxia, China on March 13, 1971. He received the Master's and Ph.D. degrees from the Department of Mathematics and Department of Computer Science, Northwest University, Xi'an, China, in 1998 and 2003, respectively.

He was a Post-Doctoral Fellow with the Institute of Automation Chinese Academy of Sciences, Beijing, China, from 2003 to 2005. He is currently a Professor with the University of the Chinese Academy of Sciences, Beijing, China. His current research interests include curve matching, 3D image reconstruction, and computer graphics.

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 with the Institute for Infocomm Research, Singapore, as a Research Staff Member from 1996 to 2002. He joined the Chinese Academy of Sciences, Beijing, China, under Science100 Talent Plan in 2003, and is currently a Professor with the Graduate University of Chinese Academy of Sciences. His current research interests include image and video analysis, video coding, and 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 AI Lab, Massachusetts Institute of Technology. He is currently a Professor with the School of Electronic Engineering and Computer Science, Peking University, Peking, China, and a Professor in computer science with the Harbin Institute of Technology. He is also the Honor Professor in computer science with the City University of Hong Kong, Hong Kong, and the External Fellow of International Computer Science Institute, University of California, Berkeley, CA, USA. His current research interests include signal processing, image and video communication, computer vision, and artificial intelligence.