Towards Codebook-Free: Scalable Cascaded Hashing for Mobile Image Search

Wengang Zhou, Ming Yang, Houqiang Li, Xiaoyu Wang, Yuanqing Lin, and Qi Tian, *Senior Member, IEEE*

***Abstract*—State-of-the-art image retrieval algorithms using local invariant features mostly rely on a large visual codebook to accelerate the feature quantization and matching. This codebook typically contains millions of visual words, which not only demands for considerable resources to train offline but also consumes large amount of memory at the online retrieval stage. This is hardly affordable in resource limited scenarios such as mobile image search applications. To address this issue, we propose a codebook-free algorithm for large scale mobile image search. In our method, we first employ a novel scalable cascaded hashing scheme to ensure the recall rate of local feature matching. Afterwards, we enhance the matching precision by an efficient verification with the binary signatures of these local features. Consequently, our method achieves fast and accurate feature matching free of a huge visual codebook. Moreover, the quantization and binarizing functions in the proposed scheme are independent of small collections of training images and generalize well for diverse image datasets. Evaluated on two public datasets with a million distractor images, the proposed algorithm demonstrates competitive retrieval accuracy and scalability against four recent retrieval methods in literature.**

***Index Terms*—Binary signature, cascaded hashing, matching verification, mobile image search.**

I. Introduction

With the ever increasing popularity of smart phones and tablets, billions of people are projected to use mobile as primary internet access points since 2012. These mobile devices generally equip cameras which become one of the most natural and convenient portals from the physical space to the digital world. Thus, the ubiquitous access to both digital photos and internet sheds bright light on many emerging applications based on mobile image search. For instances, searching similar landmark or product images among visual media resource online may allow users to explore valuable information such as reviews or discounts on the spot. These applications demand efficient and scalable content-based image retrieval.

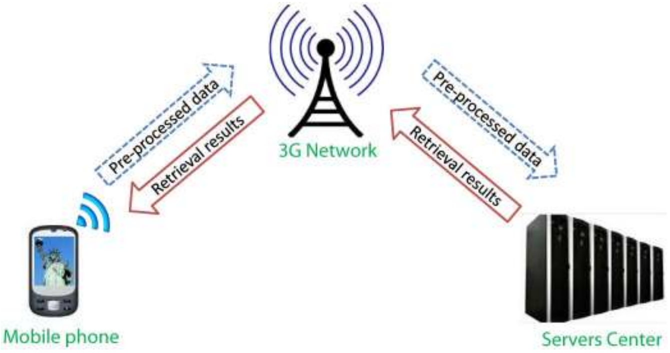


Fig. 1. An illustration of mobile visual search in wireless environment. The special concerns include the transmission latency and the limited resource in the mobile device.

Fig. 1 illustrates the mobile visual search scenario in wireless environment. Generally, the mobile phone is used as an interface to take photos, transmit the photo data, and receive the retrieval results from the computing servers via 3G network. The index file of the image database is stored in the servers and the searching processing is also conducted in the computing servers. Considering the data transmission latency, it is desirable to transit some compact pre-processed data such as quantization results instead of the raw image data or the extracted features. Due to the limited memory resource in a mobile phone, it is infeasible to a large visual codebook locally and a codebook-free retrieval paradigm is desired.

Recent years have witnessed significant advances in content-based image retrieval on large-scale image databases. In particular, the approaches [1]–[11] that utilize invariant local features [12], [13] to represent images and leverage the bag-of-visual-words (BoW) model [1] to index large-scale image datasets have demonstrated excellent retrieval precision and scalability. Further, post-processing techniques, such as spatial verification [2], [3], [8], [14] and query expansion [4], [15]–[17], have been explored to boost the retrieval accuracy. In essence, the fundamental problem of content-based image search is how to perform visual matching between images reliably and efficiently.

In image retrieval algorithms using local invariant features, the image matching is achieved via local feature matching

Manuscript received February 01, 2013; revised May 07, 2013 and August 01, 2013; accepted August 13, 2013. Date of publication January 22, 2014; date of current version March 13, 2014. This work was supported in part to Dr. Zhou by the Fundamental Research Funds for the Central Universities under contract No. WK2100060014 and the start-up funding from the University of Science and Technology of China under contract No. KY2100000036, in part to Dr. Li by NSFC under contract No. 61325009, No. 61390514, and No. 61272316, and in part to Dr. Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, Google, and FXPAL, and 2012 UTSA START-R Research Award, respectively. This work was supported in part by National Science Foundation of China (NSFC) under contract No. 61128007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yu-Gang Jiang.

W. Zhou and H. Li are with the Electronic Engineering and Information Science Department, University of Science and Technology of China, Hefei 230027, China (e-mail: zhwgeeis@gmail.com; lihq@ustc.edu.cn).

M. Yang, X. Wang, and Y. Lin are with NEC Laboratories America, Inc., Cupertino, CA 95014 USA (e-mail: myang@nec-labs.com; xwang@nec-labs.com; ylin@nec-labs.com).

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

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/TMM.2014.2301979

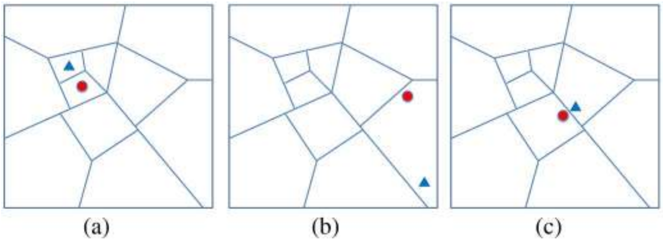


Fig. 2. Mock-up illustration of the issues in the codebook based feature quantization. The feature space is split into multiple small cells, each of which corresponds to a visual word. The query feature and candidate database feature are denoted by a red circle and a green triangle, respectively. (a) Two features at a small distance are quantized to different visual words; (b) Two features at a large distance are quantized to the same visual word; (c) Two features close to the cell boundary are quantized to different visual words.

between images. The most straightforward way is to explicitly compare and threshold the distance between two feature descriptors, which can be hardly generalized beyond pairwise image matching or registration. Using the conventional BoW model, feature matching is implicitly determined by whether two descriptors are quantized to the same visual word [1] in a codebook, which essentially matches local features by their hashing values (*i.e.*, the visual word ID) and thus lowers the complexity of feature *matching* from quadratic to linear. Furthermore, a large vocabulary tree with millions of leaf nodes [5], *i.e.*, a hierarchical codebook instead of a flat one, is adopted to reduce the complexity of feature *quantization* or hashing from linear to logarithm. This approach significantly enhances the scalability of local feature quantization and matching.

However, there are two issues led by a large visual codebook. First, the codebook requires considerable resources to train offline and consumes large amount of memory online. For example, a hierarchical codebook with a million visual words for 128D SIFT descriptors [12] is generally learned from tens of million training descriptors, and requires hundreds of megabytes to store at the runtime. These requirements prohibit its usage in resource limited scenarios such as visual search in mobile device. Second, the feature quantization error using a hierarchical vector quantization is not easy to control. Depending on the training descriptors, a large codebook divides the feature space to multiple small cells (*i.e.*, the hashing cells) with a variable coverage. It is not rare to observe such cases shown in Fig. 2: features at a small distance from each other are quantized to different words in Fig. 2(a); features at a large distance from each other are quantized to the same word in Fig. 2(b); or features close to quantization cell boundaries are separated to different words due to the hard-decision strategy, no matter how small their distances are, as illustrated in Fig. 2(c). These cases easily cause the false positive or true negative matches of local features, leading to less precise retrieval performance.

To avoid the above issues, in this paper we investigate how to hash local features for near-duplicated image retrieval without involving any codebook training and vector quantization. We propose to first ensure the recall rate of feature matching with a scalable cascaded hashing (SCH) scheme which conducts scalar quantization on the principle components of local descriptors in a cascaded manner. Then, we improve the precision rate of feature matching by verifying the binary signatures of local descriptors, which effectively bounds the quantization error to the same hashing value. The recall and precision achieved by the above two steps will eventually boost the retrieval accuracy.

This cascaded hashing scheme plus the binary signature verification is capable of balancing the needs for the recall and precision of feature matching. On one hand, the quantization on the principle components largely assures similar descriptors fall to the same hashing cell; on the other hand, the verification bounds the dissimilarity of local features linked to one inverted index, consequently boosting the retrieval accuracy performance. Since the proposed hashing scheme employs no codebook at all, we bypass the time-consuming codebook training at the offline stage and avoid the storage of a large codebook in memory during online retrieval, which is particularly beneficial to mobile devices with a strict limit on memory usage. Last but not the least, as both our hashing and binary signature generation schemes are independent of small collections of training images, our method readily generalizes to handle diverse image datasets.

We evaluate the retrieval performance and the scalability on two public benchmark datasets, *i.e.*, the *UKBench* dataset [5] and the *DupImage* dataset [8], and compare with 4 recent retrieval methods, including the visual vocabulary tree [5], Hamming embedding [7], soft assignment [6], and scalar quantization [18]. The experiments validate that our method achieves a comparable accuracy with yet a much higher efficiency than the soft assignment [6], while significantly outperforms the other three algorithms in terms of the accuracy on the *UKBench* dataset. On the *DupImage* dataset mixed with one million distractor images, our approach achieves the highest mAP over all the other 4 comparison algorithms.

The paper is organized as follows. We first review the image retrieval framework using local features and different feature quantization and hashing schemes in Section II, then present our approach in details in Section III. After that, we provide experimental results and comparisons with 4 recent methods in terms of retrieval accuracy, efficiency and memory cost in Section IV, followed by the conclusions in Section V.

II. Related Work

Content-based image search or retrieval has been a core problem in multimedia for years. In recent literature, many approaches adopt invariant local features to represent images, which exploit the bag-of-visual-words model [1] and the classic inverted index structure for scalable image search. Generally, such an image search framework consists of four necessary key modules, including feature extraction, feature quantization, image indexing, and image ranking. For feature extraction, the most popular and effective local descriptor is the SIFT [12], which is extracted on key points or regions detected by Difference of Gaussian (DoG) [12], MSER [19], or Hessian affine detector [20], *etc*. Later on, there have been lots of efforts on designing local descriptors with a higher efficiency and comparable discriminability, *e.g.*, the SURF [13] and edge-SIFT [21]. At feature quantization, each local descriptor is mapped or hashed to one or multiple visual words and then an image is represented by a group of visual words [1], [6]. After that, inverted index structures are readily adopted to index large scale image databases for image search [1]. At the online retrieval stage, the shared visual words between a query image and database images can be easily identified by looking up the inverted index lists. The similarity between the query and database images is measured by a weighted formulation [5] based on those shared visual words. Finally, those relevant database images are ranked by their similarity scores and presented to users. The initial retrieval results may be re-ranked by some post-processing techniques, such as the query expansion [4], [17], feature augmentation [16], or geometric verification [2], [3], [14].

The feature quantization, essentially a hashing function of local features, plays a key role in identifying the feature matching between images, where a visual codebook is usually trained beforehand by clustering techniques, such as k-means [1], hierarchical k-means [5], approximate k-means (AKM) [2], or affinity propagation [22]. A visual codebook may contain thousands or millions of visual words. Thus, after feature vectors are quantized to their nearest visual words, two feature vectors from different images are regarded as a match if they fall to the same word. Apparently, the vector quantization with a hard-decision strategy inevitably incurs some quantization error and leads to missing matches. To reduce the vector quantization error, the soft-quantization [6], [23] applies a soft-decision strategy and quantizes a SIFT descriptor to multiple visual words based on the proximity in the feature space. Although this method significantly lowers the chance of missing matches, identifying multiple nearby visual words from a large visual codebook is computationally expensive. To address this issue, Zhou *et al.* proposed a visual word expansion scheme to identify those nearest visual words with a pre-computed supporting visual word table [24].

Hamming Embedding [7], [25] enriches the visual word with more information from its original local descriptor and filters out false matches by Hamming distance which is efficient in computation. Spatial context of local features is a natural clue to reduce the false matches in the feature quantization [7], [26]. So either the geometric contexts and SIFT descriptors are jointly considered in the quantization [27] or separate codebooks are trained for quantization of spatial relationship between local features [28]. The spatial context can generally improve the retrieval performance considerably.

Recently, there emerges another trend to perform image search without codebook training. Zhou *et al.* [18] proposed to transform SIFT descriptors to 256-bit binary vectors by a scalar quantization scheme. Without training a codebook, this method selects 32 bits from the 256-bit vector as a codeword for indexing and search. The drawback of this approach is that the rest 224-bit per feature has to be stored in the inverted indexes, which casts a heavy memory burden. In [29], Zhang *et al.* proposed a novel query-sensitive ranking algorithm to rank PCA-based binary hash codes to search for ϵ-neighbors for image retrieval, which effectively improves the precision of feature matching but at the risk of missing some true matches.

The quantization of local descriptors is closely related to approximate nearest neighbor search. In literature, there are many hashing algorithms for approximate nearest neighbor search, *e.g.*, LSH [30], kernelized locality sensitive hashing [31], semi-supervised hashing method (SSH) [32], spectral hashing [33], min-Hashing [9], iterative quantization [34]. These hashing methods, however, are mostly applied to global image features such as GIST or BoW features at the image level, or to feature retrieval only at the local feature level. There is few work on image level search based on local feature hashing [10]. This is mainly due to the fact that those hashing schemes are capable of achieving a high precision but without guarantee on the recall rate, which may not benefit the final retrieval accuracy. To obtain a relatively high recall rate of feature matching, those hashing schemes have to generate tens of or hundreds of hashing tables, which will require a heavy memory cost for each indexed local feature of database images and meanwhile consume much more time during retrieval.

Inspired by [18], [29], in this paper, we target on codebook-free mobile image search. After studying the statistics of a large number of matched local descriptor pairs, we propose an effective scalable cascaded hashing scheme for feature quantization, which first assures the recall rate of local feature matching. Then we verify the binary signatures of SIFT features to improve the precision. The proposed approach focuses on the feature quantization and hashing step. Therefore, this codebook-free method can also be flexibly integrated with many other techniques, such as weak geometric consistency (WGC) [7], fast spatial matching [2], spatial/geometric coding [3], [8], [14], and query expansion [4], [15]–[17], contextual weighting [35], or geo-preserving visual phrase [36], *etc.*, to further boost the performance.

III. Our Approach

Our approach follows the strategy of first ensuring a relatively high recall rate of local feature matching and then refining the matching to improve the precision rate. We first conduct a PCA for dimension reduction on SIFT features [12] in Section III-A. We ensure the recall rate of local feature matching by cascaded hashing of the principal components of SIFTs as discussed in Section III-B. Ensuring the recall rate inevitably incurs some false positive feature matches. To address this issue, in Section III-C, we propose to verify the candidate feature matches by compact binary signatures of SIFT descriptors, which effectively removes a large portion of false positive matches and greatly improves the precision rate. Consequently, our method achieves promising retrieval accuracy based on reliable feature matching, free of any codebook.

A. Dimension Reduction on SIFT by a PCA

Before conducting the cascaded hashing, we reduce the dimension of SIFT feature by a PCA. The major benefit from a PCA is that the top k principal dimensions of PCA preserve most energy of the original descriptor. Therefore, we can only focus on those top dimensions instead of all 128 dimensions to reduce processing complexity. These low dimensional features facilitate a high recall rate with a limited number of hashing operations. We have collected 5 million SIFT features for the PCA training. Those training features are randomly sampled from a 50-million feature set, which are extracted from an independent

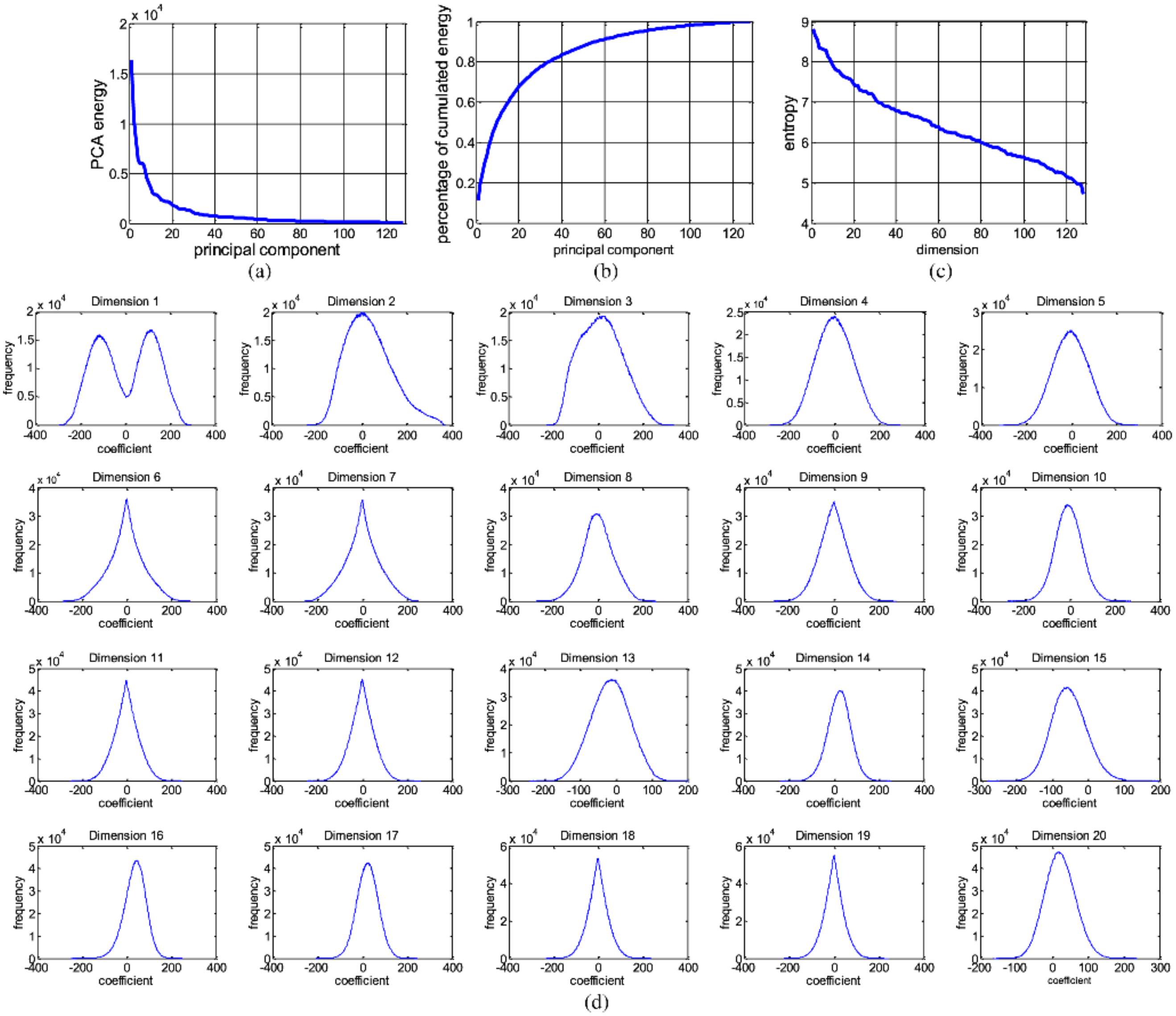


Fig. 3. PCA results on 5 million SIFT training samples. (a) The energy (eigenvalues of PCA) corresponding to each principal component; (b) The cumulated energy distribution over principal components; (c) The entropy of each dimension on the coefficient distribution after the dimension reduction; (d) The coefficient distributions of the top 20 dimensions in PSIFT.

image database. We denote the dimension-reduced SIFT feature as *PSIFT*. Some interesting observations are revealed from the results shown in Fig. 3.

Not surprisingly, the energy (eigenvalue of PCA) of SIFT feature is concentrated on a few dimensions, as illustrated in Fig. 3(a) and (b). Fig. 3(d) shows the coefficient distributions for the top 20 dimensions. It is worth noting that the coefficient distribution for the top 1 dimension exhibits a mixture of two Gaussian-like distributions, while for other dimensions the coefficient distribution presents a single Gaussian-like distribution. Besides, the coefficient range in each dimension is also different from each other. Some existing works, such as [7], [29], [34], assign one bit independently to quantize each dimension of the transformed SIFTs. However, as demonstrated in Fig. 3(c) and Fig. 3(d), one bit may be far from enough to encode a dimension if the hashing is conducted independently for each dimension. For instance, the entropy of dimension 2 is about 8.7, which means at least 8.7 bits are required to encode the distribution with small quantization error. Although some error is allowed, it is too rough to independently describe such a single Gaussian-like distribution with only one bit. In other words, these statistics suggest us to assign multiple bits to each dimension, which means multiple quantization steps for one dimension in case of a scalar quantization, unless the context among different dimensions are explored as in [18]. Based on the above observations, in the following section, we propose a cascaded hashing scheme splitting the value ranges of PSIFT's top dimensions, which largely assures the feature matching at certain given recall rates.

B. Scalable Cascaded Hashing

The general strategy of our approach is to first ensure recall rate and then improve the precision rate in feature matching. We propose a scalable cascaded hashing (SCH) to sequentially perform scalar quantization on the principal components of SIFT such that the accumulative recall rate is relatively high while the false positive rate is low. Our SCH scheme can be regarded as an approximate nearest neighbor search method focusing on ensuring the recall rate of local feature matching. Denote a PSIFT data point as $\mathbf{y} \in \mathcal{R}^d$ and a PSIFT query as $\mathbf{q} \in \mathcal{R}^d$, then $\mathbf{q}$'s ϵ-neighborhood is given as,

$$NN(\mathbf{q}, \epsilon) = \{\mathbf{y} | \|\mathbf{q} - \mathbf{y}\|_2 < \epsilon\}. \quad (1)$$

Since the PCA projection is orthogonal and preserves the L_2-distance of original SIFT descriptors [29], Eq. (1) is an approximation of the ϵ-neighborhood of the corresponding feature in the original SIFT space.

Denote $\mathbf{y}^k$ as the vector of the top k dimensions of $\mathbf{y}$ and define $\mathbf{q}^k$ for $\mathbf{q}$ in the same way. We relax Eq. (1) in the following way denoting the approximate nearest neighbor set,

$$AN(\mathbf{q}^k, \mathbf{t}) = \{\mathbf{y}^k | \|q_i - y_i\|_2 < t_i, i = 1, 2, \ldots, k\}, \quad (2)$$

where a series of thresholds $\mathbf{t} = \{t_i\}$ on each dimension are critical in our cascaded hashing scheme. We determine the thresholds by extensively empirical study with the constraints on the recall rate of local feature matching. The threshold t_i for the i-th dimension is sequentially determined with the expected recall rate of the candidate results:

$$t_i = \arg\min_n \int_0^n \mathrm{p}_i(x) dx > r_i, \quad (3)$$

where $\mathrm{p}_i(x)$ denotes the probability density function of absolute coefficient distance between relevant features (truly matched features under some criteria) for the i-th dimension, r_i is the relative recall rate for the i-th dimension defined as,

$$r_i = \begin{cases} \dfrac{|NN(\mathbf{q}, \epsilon) \bigcap AN(\mathbf{q}^i, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon)|}, & \text{for } i = 1 \\ \dfrac{|NN(\mathbf{q}, \epsilon) \bigcap AN(\mathbf{q}^i, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon) \bigcap AN(\mathbf{q}^{i-1}, \mathbf{t})|}, & \text{for } i > 1. \end{cases} \quad (4)$$

The relative false positive rate in the i-th dimension is defined as,

$$f_i = \begin{cases} \dfrac{|AN(\mathbf{q}^i, \mathbf{t}) \backslash NN(\mathbf{q}, \epsilon)|}{|\mathcal{S}|}, & \text{for } i = 1 \\ \dfrac{|AN(\mathbf{q}^i, \mathbf{t}) \backslash NN(\mathbf{q}, \epsilon)|}{|AN(\mathbf{q}^{i-1}, \mathbf{t}) \backslash NN(\mathbf{q}, \epsilon)|}, & \text{for } i > 1, \end{cases} \quad (5)$$

where $\mathcal{S}$ denotes the set of all potential feature matches. In the SCH, we do not explicitly constrain the false positive rate but focus on the recall rate.

So the overall recall after cascaded quantizing c dimensions ($c < k$) is expressed as

$$recall(c) = \prod_{i=1}^{c} r_i = \frac{|NN(\mathbf{q}, \epsilon) \bigcap AN(\mathbf{q}^c, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon)|}. \quad (6)$$

Thus the overall false positive rate after quantizing c dimensions ($c < k$) is

$$FP(c) = \prod_{i=1}^{c} f_i = \frac{|AN(\mathbf{q}^c, \mathbf{t}) \backslash NN(\mathbf{q}, \epsilon)|}{|\mathcal{S}|}. \quad (7)$$

To ensure the overall recall in Eq. (6) large enough, we impose the constraint on the recall rate of each of the c dimensions:

$$r_i > \xi. \quad (8)$$

Therefore, we have $recall(c) > \xi^c$. The selection of ξ impacts both the overall recall rate $recall(c)$ and the overall false positive rate $FP(c)$. For instance, if we select $\xi = 0.95$ (which means 95% true relevant matches are kept from the previous round of scalar quantization) and c = 10, we have $recall(c) > \xi^c \approx 0.60$. Meanwhile, the selection of ξ should remove the vast majority of irrelevant false matches after the sequential filtering. This strategy shares some insights with the face detection algorithm [37], which filters irrelevant feature samples gradually in a cascaded manner. Distinguished from other hashing methods which improve the precision rate but cannot adaptively control the recall rate, our approach explicitly improves the recall rate by learning from the statistical distribution of visual feature matches.

To select the threshold t_i in Eq. (2) with the predefined constraint on the recall rate, we need to explicitly identify the probability density function $\mathrm{p}_i(x)$ in Eq. (3). To achieve the goal, we pair-wisely select two images from relevant image groups to generate about 15 K image pairs, leading to 1.54 million relevant feature pairs. Based on those feature pairs, we conduct the feature transformation by a PCA and build the probability density function $\mathrm{p}_i(x)$ on the absolute coefficient distance between relevant features for each dimension of PSIFT.

Fig. 4(a) illustrates the probability density function for the first dimension. The accumulated probability over $\mathrm{p}_i(x)$ is shown in Fig. 4(b). From Fig. 4(b), we observe that about 94% relevant true matches are kept if we set the difference threshold as 40. Considering the large range of the coefficient in the corresponding dimension, the portion of false matches within the threshold is relatively small. Based on such observations, if we sequentially cascade k dimensions and select a proper threshold at each dimension to keep a high recall from the previous round of scalar quantization, both the overall recall and false positive rate decrease exponentially. Importantly, the false positive rate decreases much faster, which benefits narrowing down the search scope of candidate feature samples.

In the following, we present our cascaded hashing scheme for the indexed features and query feature, respectively. Given a feature vector $\mathbf{q}^c$, which is the top c dimensions from a PSIFT, its cascaded hashing result, referred as the SCH vector, is defined as the concatenation of scalar quantization result at each dimension, *i.e.*,

$$SCH(\mathbf{q}^c) = (h_1, \ldots, h_c), \quad (9)$$

where the quantization result in the i-th dimension is defined as,

$$h_i = \left\lfloor \frac{q_i - m_i}{s_i} \right\rfloor, 1 \le i \le c, \quad (10)$$

where m_i denotes the minimum coefficient in the i-th dimension, and s_i denotes the quantization step in the i-th dimension.

If the feature vector is from a database image, a hash key is generated from the cascaded quantization result in Eq. (9)

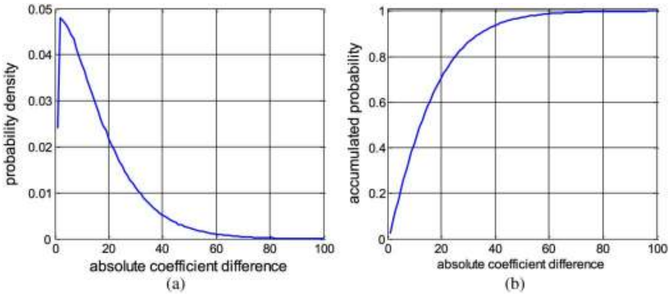


Fig. 4. Distribution of the absolute coefficient difference for the top 1 dimension of the PSIFT on the 1.54 million pairs of relevant feature pairs. (a) Probability density distribution of the absolute coefficient difference; (b) The accumulated probability integrated based on (a).

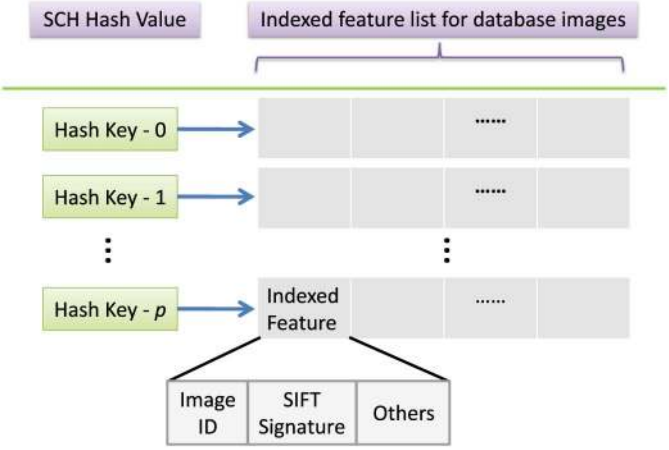


Fig. 5. The inverted index structure in our approach.

for indexing based on the inverted index structure, as shown in Fig. 5. If the feature vector is from a query image, we will take a different strategy to reduce the quantization error as discussed below.

Based on Eq. (10), each dimension is uniformly split into multiple cells. Denote the center of the j-th cell in the i-th dimension is $h_i^{(j)}$, which is a scalar. To tolerate the quantization error from the hard-decision strategy in quantization by Eq. (10), we define the following criterion to identify the candidate feature matches of a query feature. Given a PSIFT query feature $\mathbf{q}^c \in \mathcal{R}^c$, we perform quantization on each dimension q_i of $\mathbf{q}^c$ with a soft-decision strategy, by selecting those cells close to q_i as follows:

$$|q_i - h_i^{(j)}| < s_i, \forall j, 1 \leq j \leq c. \tag{11}$$

To ensure the recall performance in Eq. (8), in our implementation, we define $s_i = 2 \cdot t_i$, because, in image search, all features of database images have to be quantized to a hash key for the convenience of index and scalable retrieval. After the feature indexing, the original features of the database images are discarded. Therefore, for each indexed feature, we only know which cell it is located in the i-th dimension by the scalar quantization result in Eq. (10), but do not know the precise location in the corresponding cell. Give a query feature, we relate the i-th dimension with two closest cells. To make sure that the two closest cells contain all relevant indexed features with the absolute difference in the i-th dimension less than t_i, the best choice is to set s_i to be $2 \cdot t_i$.

Using Eq. (11), each dimension of the query feature $\mathbf{q}^c$ is assigned the ID of at most 2 cells. That is, there are at most two alternative quantization results in each dimension for the query feature. Then the final SCH result of $\mathbf{q}^c$ is obtained by alternatively select one quantization cell in each dimension. So each query feature is quantized to at most 2^c SCH vectors. For a query feature, all features indexed to any of these SCH vectors are considered as candidate matches.

C. Matching Verification by Binary Signatures

In Section III-B, we have discussed the cascaded hashing scheme which guarantees the rate of true positive results but inevitably leads to some false positive matches. To identify and remove those false positive results, it is necessary to perform matching verification on these candidates. To make the verification fast enough, it is preferable to transform the feature to binary code and verify the matching with the efficient Hamming distance measurement. Motivated by this, we propose to generate some binary signatures for those dimensions in PSIFT after the top c dimensions. In other words, for the PSIFT as $\mathbf{y} \in \mathcal{R}^d$, we select a sub-set of elements in $\mathbf{y}$ and obtain the vector $\mathbf{z} \in \mathcal{R}^e$, with $z_i = y_{c+i}$, considering that the top c dimensions have already been used in the scalable cascaded hashing in Section III-B. Then, we transform the vector $\mathbf{z}$ to a binary vector $\mathbf{b} = (b_1, \ldots, b_e)$ by comparing each coefficient with an individual threshold [18] as follows,

$$b_i = \begin{cases} 1, & \text{if } z_i > \hat{z} \\ 0, & \text{if } z_i \leq \hat{z} \end{cases}, \tag{12}$$

where $\hat{z}$ is the median of all dimensions for an individual vector $\mathbf{z}$, as suggested in [18]. Different feature vectors will have different median values, and the median value $\hat{z}$ of each feature $\mathbf{z}$ is computed online. The rationale behind Eq. (12) is that the relative coefficient differences between different dimensions are assumed to be stable. Unlike [7], [29] where each dimension of feature vector is considered independently, the context of relative magnitudes among different dimensions is implicitly and weakly encoded by Eq. (12).

With the PSIFT features represented by these binary signatures, the comparison between different features can be efficiently conducted by checking the Hamming distance between their binary signatures. Given the PSIFT query $\mathbf{q} \in \mathcal{R}^d$, we regard the candidate feature $\mathbf{y} \in \mathcal{R}^d$ given by the SCH result of $\mathbf{q}$ as a valid match if it satisfies the following criterion,

$$H(\mathbf{b}^{(\mathbf{q})}, \mathbf{b}^{(\mathbf{y})}) \leq \tau, \tag{13}$$

where $\mathbf{b}^{(\mathbf{q})}$ and $\mathbf{b}^{(\mathbf{y})}$ denote the binary signatures of $\mathbf{q}$ and $\mathbf{y}$, respectively, $H(\cdot,\cdot)$ denote the Hamming distance between two binary vectors, τ is a threshold. We will study the impact of parameter e and τ in the experiments in Section IV-A.

D. Index and Retrieval

In this section, we discuss how to perform the indexing and retrieval based on the proposed scalable cascaded hashing and matching verification. Generally, an image search system consists of an offline image indexing stage and an online retrieval stage. In the offline indexing stage, for each database image, we first conduct SIFT feature extraction and dimension reduction by a PCA to generate the PSIFT. Then, for each PSIFT, we perform the cascaded hashing and binary signature generation, as discussed in Section III-B and Section III-C, respectively. Afterwards, we identify the hash values of the scalable cascaded hashing and store the feature information into the inverted indexes, as shown in Fig. 5. For each indexed feature, we store the ID of the image that this feature belongs to, the binary signature, and other information such as geometric clues if necessary. After indexing the features of all database images, we can perform on-line retrieval for a query image.

At the online retrieval stage, given a query image, we conduct the same SIFT feature extraction and generate the PSIFT. Then, for each PSIFT, we perform the cascaded hashing and calculate the binary signature. We identify those hash keys with the SCH vectors satisfying Eq. (11) as the candidates. All candidate features linked to the inverted indexes of those hash values are verified with their binary signatures. Only those whose binary signatures satisfy Eq. (13) are regarded as the valid matches to the query feature. Then, each matched feature casts a vote to the corresponding image ID. Finally, the indexed images are ranked by their voting scores and returned to users as retrieval results.

The three major technical components in our approach, *i.e.*, the PCA, cascaded hashing, and binary signature verification, are all important and contribute to the success of the final algorithm. The PCA dimension reduction of SIFT descriptors enables the cascaded hashing to focus on fewer dimensions and reduce the computational complexity. The cascaded hashing preserves a large portion of true feature matching and removes the majority of false ones, leading to a relatively high recall rate. The binary signature verification removes those false positive feature matches after the cascaded hashing, leading to a high precision result.

IV. Experiments

We evaluate the proposed approach on two public benchmark datasets, *i.e.*, *UKBench* dataset [5] and *DupImage* dataset [8]. In Section IV-A, we study the impact and sensitivity of the key parameters on the *UKBench* dataset. Following [5], the average 4 times top-4 accuracy over this dataset is adopted to measure the retrieval performance, which counts the number of relevant results in the top 4 returned images.

In Section IV-B, we mix the *DupImage* dataset [8] with a distractor dataset, which contains one million images randomly crawled from the Web. To evaluate the scalability with respect to the size of dataset, we construct three smaller datasets (50 K, 200 K, and 500 K) by sampling the distractor dataset. From the *DupImage* dataset, 108 representative query images are randomly selected for evaluation. Mean average precision (mAP) is selected to evaluate the retrieval accuracy of all methods. The retrieval results compared with four recent image retrieval approaches using different feature quantization schemes are discussed afterwards.

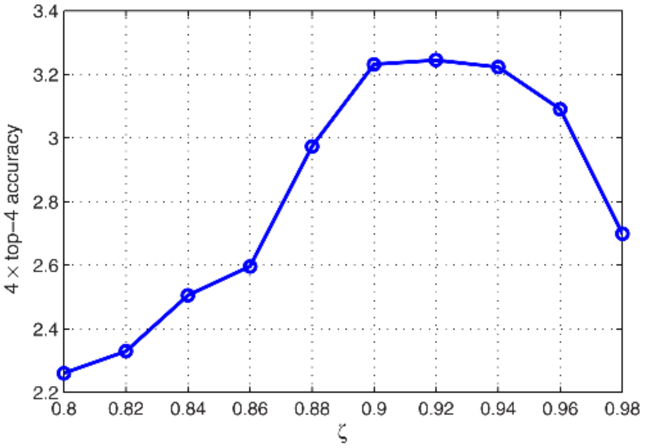


Fig. 6. The performance of the 4 times top-4 accuracy on the *UKBench* dataset with the 64-bit signature using various values of ξ, with $\tau = 10$.

A. Impact of Key Parameters

There are 4 key parameters in our algorithm, *i.e.*, the recall threshold ξ, the dimensionality c, binary signature length e and Hamming threshold τ. Since it is difficult to study the impact of their combinations, in the following, we study the impact of each parameter to the retrieval accuracy separately.

The recall threshold ξ in Eq. (8) determines the fraction of relevant feature matches that can be preserved in hashing each dimension from the previous round. With a specific dimension number c, the overall recall rate is proportional to ξ^c. As shown in Fig. 6, when ξ increases, the top-4 accuracy first grows sharply, then keeps stable and drops sharply. This indicates that when ξ is relatively small, a high recall is obtained with the increase of ξ. However, when ξ becomes too large, *e.g.*, over 0.94, we also keep too many false positive feature matches, leading to an increasing number of false matches that fail to be removed by the binary signature verification in Eq. (13). As a result, such false positive results degrade the accuracy performance. According to Fig. 6, in the following experiments, we empirically set ξ as 0.90.

The second key parameter is the dimensionality c in Eq. (6). Since the overall recall of feature matching in SCH is proportional to ξ^c, given ξ is predefined, increasing c gradually decreases the recall, but not necessarily degrades the retrieval accuracy. As revealed in Fig. 7, when c grows, the retrieval accuracy first gradually increases and then decreases sharply after c becomes larger than 11. This is because when c is relatively small, increasing c degrades the recall performance, which, however, still passes sufficient true feature matches to the verification stage. On the other hand, when c becomes too large, the recall performance drops too much that few true features matches remain after the hashing. In our following experiments, we select c as 10 which generally works well.

The other two parameters e and τ is related to the binary signature of PSIFT. Since the binary signature is stored in the inverted index list, its length is expected to be small, so as to reduce the memory cost. Therefore, in our experiments, we only test two alternatives of e, *i.e.*, 32 and 64. For each type of binary signature, we test the retrieval performance using different

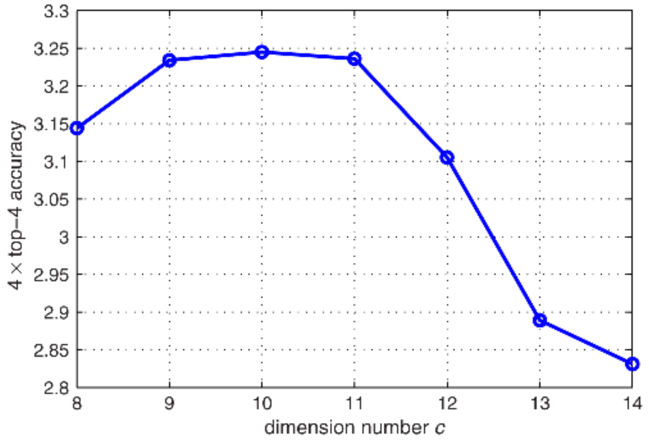


Fig. 7. The performance of the 4 times top-4 accuracy on the *UKBench* dataset with the 64-bit signature using various values of dimension c, with $\tau = 10$ and $\xi = 0.90$.

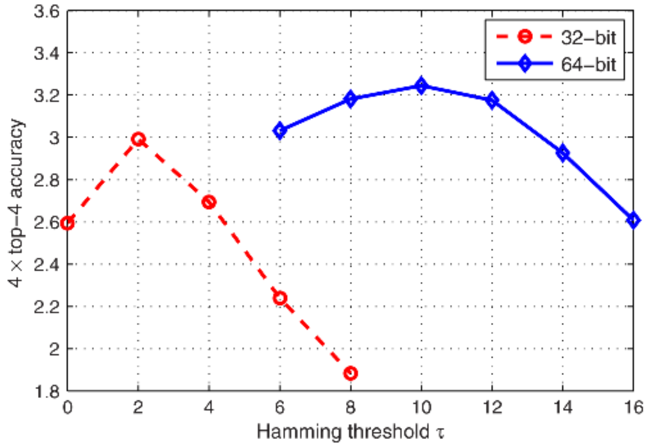


Fig. 8. The performance of the 4 times top-4 accuracy on the *UKBench* dataset with 32-bit or 64-bit signatures using various values of Hamming threshold τ, with $c = 10$ and $\xi = 0.90$.

selections of the Hamming threshold τ. From Fig. 8, we observed that for each kind of binary signature, the top-4 accuracy first increases to a peak and then decreases, when the threshold τ grows. This is due to that a smaller threshold removes more false positives, but also may filter out more relevant matches. On the other hand, when the threshold τ becomes larger, more relevant results survive the verification, but more false positives live as well, which degrades the retrieval accuracy. It can be deduced that the retrieval accuracy will be much poor when the threshold τ takes the value of the bit length of the binary signature, which corresponds to the case that the binary signature verification is ignored. This fact partially reveals that both the cascaded hashing and binary signature verification are important to the success of the final algorithm. In our following experiments, as a tradeoff we select e as 64 and τ as 10.

The retrieval performance of the top-4 accuracy and the query efficiency of four other recent algorithms, introduced below, are shown in Table I. Considering both accuracy and efficiency, our approach achieves the best overall performance.

B. Performance Evaluation

Comparison algorithms: We compare our approach with four feature quantization methods for large-scale image search.

TABLE I
COMPARISON IN TERMS OF THE 4 TIMES TOP-4 ACCURACY AND THE AVERAGE QUERY TIME ON THE *UKBENCH* DATASET [5]

Method	baseline [5]	HE [7]	SA [6]	SQ [18]	our approach
4×top-4 accuracy	2.90	3.04	3.26	2.99	3.25
query time (second)	0.036	0.038	0.631	0.050	0.146

The baseline method is the BoW approach using a large visual vocabulary tree [5]. We test various sizes of the visual vocabulary and select the tree with one million (10^6) leaf nodes which yields the best overall performance. Two extensions of the baseline using soft assignment [6] and Hamming embedding [7] are also evaluated in the comparison.

Soft assignment [6] represents a local feature by a weighted combination of three nearest visual words. We implement this method using the default parameters in [6]. The nearest visual words for a given feature are found by the k-d tree algorithm [23], [38] in a public library for approximate nearest neighbor (ANN) searching [39] in our experiments. The error bound parameter [39] is set as 5 as a tradeoff between the accuracy and efficiency. We denote this soft assignment method as "SA".

The Hamming embedding method [7] generates additional binary codes (64 bits) to filter out candidate features which are quantized to the same visual word but with a large hamming distance to the query feature. We denote this method as "HE". We have tested different thresholds for the Hamming distance in the HE and select the one that achieves the best performance. Since the main focus of this paper is feature quantization, the weak geometric consistency (WGC) scheme [7] is not involved in the experiments.

Different from the above three algorithms based on vector quantization using visual codebook, the 4th algorithm adopts the scalar quantization [18] to generate a 256-bit binary signature for each SIFT feature and takes out 32 bits as code word to index image feature. This method does not involve a codebook neither which is denoted as "SQ" in the evaluation.

Accuracy: We test the retrieval performance of all algorithms on the *DupImage* dataset with different number of distractor images. As shown in Fig. 9, for each algorithm, the mAP drops as the database size grows. Our approach consistently achieves the best mAP performance at different database sizes. On the one-million image database, the baseline approach [5] achieves an mAP of 0.38. Hamming Embedding (HE) [7] and soft assignment (SA) [6] improve the mAP to 0.43 and 0.48, respectively. The scalar quantization (SQ) approach [18] pushes the mAP to 0.54. Our approach achieves the mAP 0.60, which improves over the SA by about 25.0%. Another observation from Fig. 9 is that our approach exhibits a mild decline in mAP as the database size increases compared with the baseline, HE and SA, demonstrating a better scalability. This is partly due to the fact that the discriminative power of a fixed visual codebook generally decreases along the growth of the image database. Although a larger visual codebook may alleviate this issue, it may also introduce larger quantization error to the local feature.

Efficiency: We compare the efficiency in both offline indexing and online query. Table II shows that our approach is comparable with the baseline and HE approach in terms of the efficiency of offline indexing. The major computational cost of

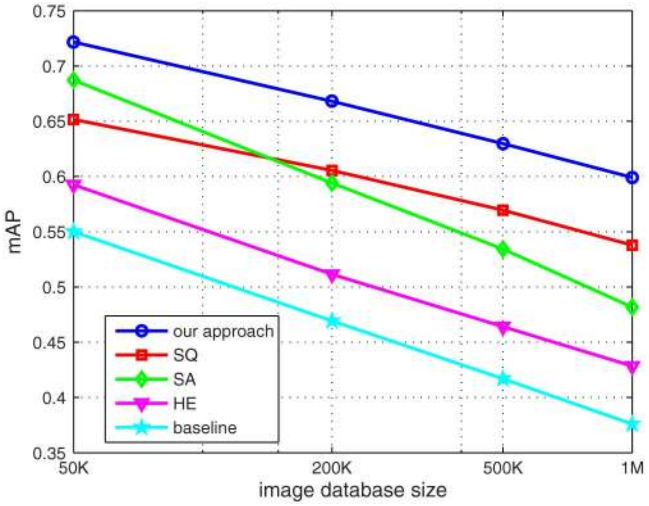


Fig. 9. Comparison of the mAP for 5 approaches on the *DupImage* with different numbers of image database images. SQ: scalar quantization [18]; SA: soft assignment [6]; HE: Hamming Embedding [7].

TABLE II
COMPARISON OF THE INDEXING TIME IN SECOND OF 1 MILLION SIFT FEATURES AT THE OFFLINE STAGE

Method	baseline [5]	HE [7]	SA [6]	SQ [18]	our approach
Time (second)	53.7	64.8	771.1	18.9	68.1

our approach in feature quantization is on the PCA projection and binary signature generation. SQ [18] is very efficient in indexing one million SIFT features within only 18.9 seconds. SA [6] is quite time-consuming, about 10 times slower than our approach, since the soft assignment requires to identify the three nearest visual words from a large visual codebook with the k-d tree for every feature in database images.

On the one-million image database, the average query time of all algorithms is compared in Fig. 10. The baseline approach is very efficient, using about 0.12 second on average. The HE [7] approach is even more efficient, with the average query time less than 0.1 second. The other three algorithms are more time consuming. The average query time of the SA approach [6] and the SQ is 0.54 second and 0.49 second, respectively. Our approach requires about 0.69 second per query, mainly due to that the soft quantization in the cascaded hashing may result in many candidate features in the database to be verified by their binary signatures. The retrieval efficiency of our approach can be improved if the number of hash buckets is reduced. In this paper, we perform quantization in each dimension independently. Therefore, with c dimensions, the maximum number of non-empty hash buckets is 2^c. However, if in some dimension the query coefficient is located close to the cell center, we may only check one cell instead of two, with minor sacrifice of accuracy. If m out of c dimensions in a query feature satisfy the above condition, it is possible to reduce the bucket number from 2^c to $2^{(c-m)}$, which causes some minor reduction in accuracy but may greatly improve the efficiency.

Memory Cost: We compare the memory cost in terms of both the indexed database features and the quantization or hashing function in all the 5 approaches, as listed in Table III. In terms of the memory cost per indexed feature, the baseline method [5] needs 4 bytes to store one image ID and another 4 bytes to store the *tf-idf* weight. The SA [6] has to store the image ID of each indexed feature in three visual word lists, therefore it costs 24 bytes, which is three times the memory cost of the baseline approach. In the HE [7], it allocates 4 bytes on one image ID and 8 bytes on the 64-bit Hamming code. Similar to the HE, our approach requires 12 bytes for each indexed feature, including 4 bytes for an image ID and another 8 bytes to store the binary signature. Compared with the above four methods, the SQ [18] consumes more memory, which needs 28 bytes to store a 224-bit binary feature besides the 4 bytes for an image ID.

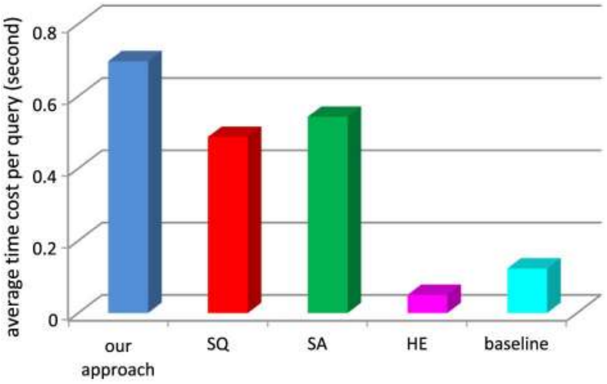


Fig. 10. Comparison on the query time for the 5 approaches on the one-million image database. The query time does not include the SIFT feature extraction.

TABLE III
MEMORY COST FOR EACH INDEXED FEATURE AND THE QUANTIZATION FUNCTION FOR THE 5 APPROACHES

Method	baseline [5]	HE [7]	SA [6]	SQ [18]	SCH
Per indexed feature (byte)	8	12	24	32	12
Quantization (megabyte)	560	816	920	0	0.00012

Besides the indexed features for database images, the retrieval algorithms also need to load the quantization or hashing function into the memory during online query as the overhead. A hierarchical visual vocabulary tree with 10^6 leaf nodes is required for both the baseline [5] and HE [7], which needs about 560M bytes to save 1.11M 128D floating-point SIFT vectors. Besides, the HE method stores additional 64D median vectors for each leaf node which adds 256M extra storage. As for the SA [6], based on the 1M visual words (only leaf nodes of the vocabulary tree), it needs to generate a large k-d tree, which costs about 920 M bytes memory in total in our implementation. In contrast, our SCH scheme spends much less memory cost to save the scalar quantization functions. It just needs to save the quantization step (4 bytes) and coefficient range (8 bytes) for 10 dimensions, that costs only 120 bytes in total. The SQ [18] does not involve any explicit quantizer and has no memory cost for quantization. The light requirement of the runtime memory enables the codebook-free methods to be readily applied in resource limited scenarios, such as mobile phone based visual search.

Transmission data: In the scenario of mobile image search via wireless network, the quantization can be conducted in the mobile device locally and the quantization results are transmitted as pre-processed data to the server for retrieval. In the four comparison algorithms and our approach, each 128-D floating-point SIFT feature is quantized to a visual word ID or hash code, which can be represented with a 4-byte integer. That is, for each SIFT feature, only 4-byte data needs to be transmitted via the wireless network.

V. Conclusion

In this paper, we propose a novel codebook-free image search algorithm. We present a scalable cascaded hashing scheme for local feature quantization which first ensures the matching recall rate, followed by a verification step using compact binary signatures to remove false positive matches. Therefore, this method achieves a balanced recall and precision for the feature matching. Thorough study of the parameter sensitivity and the scalability on two public datasets has validated that this method achieves competitive retrieval performance against 4 recent image retrieval algorithms. More importantly, unlike conventional BoW based image search algorithms, the proposed approach involves neither training large visual codebooks nor loading them into memory during online query. Thus, the merits of a high retrieval precision and low memory footprint make the method a good fit to mobile phone based image search.

The future work will include investigation of other dimension reduction schemes to improve the feature matching precision of the scalable cascaded hashing which may relax the requirement of the verification step and accelerate the overall retrieval speed.

References

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 1470–1477.

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[3] 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.

[4] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. Int. Conf. Computer Vision*, 2007, pp. 1–8.

[5] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2161–2168.

[6] 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. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[7] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Computer Vision*, 2008, pp. 304–317.

[8] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Large scale image search with geometric coding," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1349–1352.

[9] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: Min-hash and tf-idf weighting," in *Proc. British Machine Vision Conf.*, 2008, vol. 3, p. 4.

[10] 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. Computer Vision and Pattern Recognition*, 2012, pp. 3005–3012.

[11] Q. Tian, S. Zhang, W. Zhou, R. Ji, B. Ni, and N. Sebe, "Building descriptive and discriminative visual codebook for large-scale image applications," *Multimedia Tools Applicat.*, vol. 51, no. 2, pp. 441–477, 2011.

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

[13] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Computer Vision*, 2006, pp. 404–417.

[14] W. Zhou, H. Li, Y. Lu, and Q. Tian, "SIFT match verification by geometric coding for large-scale partial-duplicate web image search," *ACM Trans. Multimedia Comput., Commun., Applicat. (TOMCCAP)*, vol. 9, no. 1, p. 4, 2013.

[15] Y. Kuo, K. Chen, C. Chiang, and W. Hsu, "Query expansion for hash-based image object retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2009, pp. 65–74.

[16] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918.

[17] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 889–896.

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

[19] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vision Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

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

[21] 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.

[22] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.

[23] J. Friedman, J. Bentley, and R. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[24] W. Zhou, H. Li, Y. Lu, M. Wang, and Q. Tian, "Visual word expansion and BSIFT verification for large-scale image search," *Multimedia Syst.*, pp. 1–10, 2013.

[25] M. Jain, H. Jégou, and P. Gros, "Asymmetric hamming embedding: Taking the best of our bits for large scale image search," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1441–1444.

[26] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Large scale partial-duplicate image retrieval with bi-space quantization and geometric consistency," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, 2010, pp. 2394–2397.

[27] 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.

[28] Z. Liu, H. Li, W. Zhou, and Q. Tian, "Embedding spatial context information into inverted file for large-scale image retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 199–208.

[29] X. Zhang, L. Zhang, and H. Shum, "QsRank: Query-sensitive hash code ranking for efficient ϵ-neighbor search," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 2058–2065.

[30] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. IEEE Symp. Foundations of Computer Science*, 2006, pp. 459–468.

[31] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. Int. Conf. Computer Vision*, 2009, pp. 2130–2137.

[32] J. Wang, S. Kumar, and S. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 3424–3431.

[33] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Advances in Neural Information Processing Systems*, 2008, pp. 3424–3431.

[34] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 817–824.

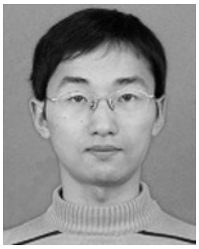
[35] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proc. Int. Conf. Computer Vision*, 2011, pp. 209–216.

[36] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 809–816.

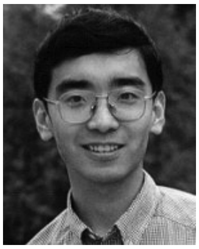
[37] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[38] J. Bentley, "K-d trees for semidynamic point sets," in *Proc. Annual Symp. Computational Geometry*, 1990, pp. 187–197.

[39] S. Arya and D. M. Ann, *Ann: Library for approximate nearest neighbor searching* [Online]. Available: http://www.cs.umd.edu/mount/ANN/



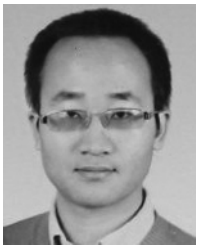
Wengang Zhou received the B.E. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronic engineering and information science from University of Science and Technology of China, China, in 2011. He was a research intern in Internet Media Group in Microsoft Research Asia from December 2008 to August 2009. From September 2011 to 2013, he works as a post-doc researcher in Computer Science Department in University of Texas at San Antonio. He is currently an associate professor at the Department of Electronic Engineering and Information Science, USTC. His research interest is mainly focused on multimedia information retrieval. He received the best paper award in ACM ICIMCS 2012.



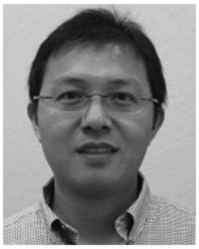
Ming Yang received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, Illinois, in June 2008. From 2004 to 2008, he was a research assistant in the computer vision group of Northwestern University. After his graduation, he joined NEC Laboratories America, Cupertino, California, where he is currently a senior researcher. His research interests include computer vision, machine learning, video communication, large-scale image retrieval, and intelligent multimedia content analysis. He is a member of the IEEE.

Houqiang Li received the B.S., M. Eng., and Ph.D. degree from University of Science and Technology of China (USTC) in 1992, 1997, and 2000, respectively, all in electronic engineering. He is currently a professor at the Department of Electronic Engineering and Information Science, USTC. His research interests include video coding and communications, image/video analysis, and computer vision. His research has been supported by National Natural Science Foundation of China (NSFC), State High-Tech Development Plan of China (863 program), Microsoft, Nokia, and Huawei.

He is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and in the Editorial Board of *Journal of Multimedia*. He has served on technical/program committees, organizing committees, and as program co-chair, track/session chair for over 10 international conferences.



Xiaoyu Wang is a researcher in the department of Media Analytics at NEC Laboratories America. He obtained his Ph.D. in electrical and computer engineering as well as M.A. in statistics from University of Missouri in 2012. Before that, he is a research assistant in the department of electrical engineering and information science in University of Science and Technology of China from 2006 to 2008. He obtained his bachelor in Electrical Engineering and Information Science from University of Science and Technology of China in 2006.



Yuanqing Lin received his Ph.D. degree in electrical engineering from University of Pennsylvania in 2008. After that, he joined NEC Labs America as a Research Staff Member, working on feature learning and large-scale classification. In 2010, he was the team leader of NEC-UIUC team and the team won the No. 1 place in ImageNet Large Scale Visual Recognition Challenge. In April 2012, he became the head of the Media Analytics Department of NEC Labs America.

Qi Tian (M'96–SM'03) received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University in 1996 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana–Champaign in 2002. He is currently a Professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA).

Dr. Tian's research interests include multimedia information retrieval and computer vision. He has published over 190 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA and he received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008-2009. He was the co-author of an ACM ICMCS 2012 Best Paper, an MMM 2013 Best Paper, a Top 10% Paper Award in MMSP 2011, a Best Student Paper in ICASSP 2006, and co-author of a Best Paper Candidate in PCM 2007. He received 2010 ACM Service Award.

Dr. Tian has been serving as Program Chairs, Organization Committee Members and TPCs for numerous IEEE and ACM Conferences including ACM Multimedia, SIGIR, ICCV, ICME, etc. He is the Guest Editor of 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 IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), *Journal of Multimedia* (JMM) and *Journal of Machine Visions and Applications* (MVA).