

Dehashing: Server-Side Context-Aware Feature Reconstruction for Mobile Visual Search

Yin-Hsi Kuo and Winston H. Hsu, *Senior Member, IEEE*

Abstract—Due to the prevalence of mobile devices, mobile search becomes a more convenient way than desktop search. Different from the traditional desktop search, mobile visual search needs more consideration for the limited resources on mobile devices (e.g., bandwidth, computing power, and memory consumption). The state-of-the-art approaches show that the bag-of-words (BoW) model is robust for image and video retrieval; however, the large vocabulary tree might not be able to be loaded on the mobile device. We observe that recent works mainly focus on designing compact feature representations on mobile devices for bandwidth-limited network (e.g., 3G) and directly adopt feature matching on remote servers (cloud). However, the compact (binary) representation might fail to retrieve target objects (images and videos). Based on the hashed binary codes, we propose a dehashing process that reconstructs BoW by leveraging the computing power of remote servers. To mitigate the information loss from binary codes, we further utilize contextual information (e.g., GPS) to reconstruct a context-aware BoW for better retrieval results. Experiment results show that the proposed method can achieve competitive retrieval accuracy as BoW while only transmitting few bits from mobile devices.

Index Terms—Bag-of-words (BoW), binary codes, mobile visual search (MVS), vector of locally aggregated descriptors (VLAD).

I. INTRODUCTION

WITH the explosive growth of mobile devices, the needs for mobile visual search (MVS) are emerging. Because of the limited computing power and memory usage, it becomes a challenging problem for MVS [1]. Different from the traditional content-based image/video retrieval [2], [3], MVS requires lightweight computing and small data transmission. Hence, recent works focus on generating compact representations before transmitting the query. In order to achieve good retrieval accuracy, they will extract local features and compress them into binary codes for different applications,

Manuscript received August 14, 2015; revised December 28, 2015 and March 23, 2016; accepted July 6, 2016. Date of publication July 9, 2016; date of current version January 5, 2017. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 104-2622-8-002-002 and Grant MOST 104-2221-E-002-049, and in part by MediaTek Inc., and Microsoft Research Asia. The work of Y.-H. Kuo was supported by the Microsoft Research Asia Fellowship. This paper was recommended by Associate Editor W. Zeng. (*Corresponding author: Winston H. Hsu.*)

Y.-H. Kuo is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (e-mail: kuonini@cmlab.csie.ntu.edu.tw).

W. H. Hsu is with the Department of Computer Science and Information Engineering, Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (e-mail: whsu@ntu.edu.tw).

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/TCSVT.2016.2589679

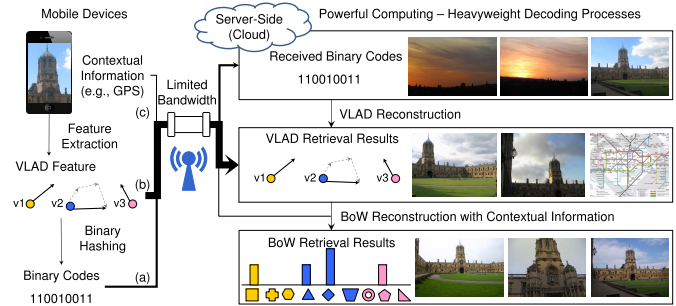


Fig. 1. For MVS, the state-of-the-art approaches usually send hashed features through the bandwidth-limited network (3G) and apply feature matching on the server-side (cloud). We observe that we can utilize the computing power of remote servers to reconstruct a better feature representation from the hashed feature for MVS. We propose a context-aware feature reconstruction to achieve better retrieval results. The thickness of the lines that pass through the wireless network represents the amount of data transmission. We can transmit (a) binary codes or (b) a VLAD feature to the remote server. Meanwhile, we can integrate the proposed method with (c) contextual information to reconstruct a context-aware bag-of-words (BoW).

such as product search [4], landmark retrieval [5], image/video retrieval [6], [7], and interactive image exploring system [8]. Moreover, some works, such as [9], further aim at on-device image matching.

The state-of-the-art visual feature—vector of locally aggregated descriptors (VLAD) [10]—has been shown promising for image/video retrieval, which has similar retrieval performance as the BoW model [2], [11]. However, it might suffer from object queries [12], because database images usually contain not only the target object but also cluttered backgrounds, as shown in Fig. 1. These background features will affect the final representation of an image, because VLAD aggregates features into a single representation (Section III-A). For object retrieval, it is necessary to utilize BoW to mitigate the effect of noisy or cluttered backgrounds and provide better retrieval accuracy [13]. Otherwise, we need to generate and match multiple VLADs from different sizes of database images [12]. However, BoW requires a large vocabulary tree, which might not satisfy the memory constraint on mobile devices. Hence, He *et al.* [4] propose bag of hash bits for image retrieval. They only consume a small amount of memory to generate compact binary codes for each local feature. Nevertheless, if the image/video contains many local features, the transmission time (cost) is still large.

Compared with mobile devices, remote servers (cloud) have stronger computing power and storage. We find that the state-of-the-art approaches usually apply feature matching directly on the received features. To leverage the computing

TABLE I

COMPARISON OF THE STATE-OF-THE-ART METHODS WITH OUR PROPOSED METHOD. OUR PROPOSED METHOD ONLY CONSUMES A SMALL AMOUNT OF MEMORY ON MOBILE DEVICES. MOREOVER, WE CAN RECONSTRUCT CONTEXT-AWARE FEATURE REPRESENTATIONS FROM THE RECEIVED BINARY CODES ON REMOTE SERVERS AND ACHIEVE BETTER RETRIEVAL ACCURACY

	Transmission (KBytes)	Transmission Size	Memory Consumption on Mobile (KBytes)	Meta-Information
BoW (1M) [14]	0.36 - 0.6	Dependent (~ 13 bits/feature)	68,000 (quantization tree)	8-bit per dimension
CHoG [15]	0.38 - 5.3	Dependent (~ 60 bits/feature)	-	Feature location
Product Search [4]	0.5 - 5	Dependent (80 bits/feature)	20 (64x80x4, projection matrix)	Boundary information
REVV [16], [17]	- (0.35)	Independent	264 (tree + matrix)	On-device matching
MCVD [5]	0.025 - 0.4	Independent	690 - 3,440	Contextual information
IMShare [18]	8 (2 + 6)	Dependent	-	Thumbnail
Our proposed method	0.8 - 1.6	Independent	45 - 121.9 (tree + matrix)	Contextual information

power of remote servers, we utilize VLAD as an intermediate feature and generate a better feature representation before feature matching. To tackle this challenge, we observe the relation between VLAD and BoW (Section III), and propose to estimate possible visual words (VWs) from VLAD on remote servers. Meanwhile, motivated by the on-device image matching [9], we generate compact and fixed length binary codes from VLAD on mobile devices. As Fig. 1 shows, given a query (image or video), we adopt a lightweight encoding process that hashes features into binary codes on mobile devices, and design a novel decoding process on the server-side called dehashing. To the best of our knowledge, this is the first work that attempts to reconstruct BoW from VLAD or binary codes. Hence, the proposed method only transmits a single compact representation (with contextual information) from a huge amount of local features in the mobile query.

In this paper, we aim to leverage both the mobile- and cloud-based configurations (two heterogeneous platforms) for balancing effectiveness and efficiency for image/video retrieval (also extendable for image/video classification). In a novel way, we exploit the (unbalanced) computation capabilities between the two heterogeneous platforms (i.e., mobile devices and cloud servers) and seek new feature learning and representations friendly with the whole path from mobiles, through communication channel, and to the cloud. The primary contributions of this paper include the following.

- 1) Proposing a dehashing process that leverages contextual information to approximate BoW from a compact feature representation on the server-side (Section III).
- 2) Investigating the memory consumption of binary hashing on mobile devices (Section IV).
- 3) Conducting experiments on two retrieval benchmarks and demonstrating that the proposed method can achieve better retrieval accuracy compared with the original binary codes (Section V).

II. RELATED WORKS

We aim to provide better search results for MVS; hence, we introduce recent works and compare them with our proposed method. The state-of-the-art image/video retrieval systems usually extract BoW from a query; however, it might be infeasible to transmit the query image or video from mobile devices under the unstable network connection [1]. Recent works focus on low-bitrate MVS [4], [9], [19]. They propose a lightweight encoding process on the extracted features before transmission. These approaches can roughly divide into four compression methods—BoW-based [14], compressed

histogram of gradients (CHoG)-based [15], hash-based [4], and VLAD-based [17] methods.

For the BoW-based method, they attempt to prune the large vocabulary tree, so that it can fit in the mobile memory [5] and apply standard encoding methods (e.g., run-length encoding or arithmetic coding) to further compress the BoW histogram [14]. Instead of applying BoW, Chandrasekhar *et al.* [15] propose a novel descriptor—CHoG—by considering the limited resources on mobile devices. Different from vector quantization (or tree-based) approach, the hash-based method utilizes a small amount of memory consumption (projection matrix) to efficiently generate compact binary codes [4], [20]–[22]. Moreover, recent works further propose more compact and discriminative binary descriptors [23], [24] to achieve similar performance as floating-point descriptors. These approaches have shown promising results on different benchmarks; however, the transmission cost highly depends on the number of extracted local features. For high-resolution images, it might exceed 1000, but these methods usually extract few hundreds of features to perform image or video matching.

To tackle this problem, Chen *et al.* [17] propose a novel compact global signature called residual enhanced visual vector (REVV), which compresses VLAD feature into binary codes. Hence, their proposed method only needs a few bits to represent each image. It is very suitable for on-device image retrieval for personal photos, because the whole process can be done on the mobile device. Similarly, Ji *et al.* [5] propose a multiple-channel coding-based compact visual descriptor (MCVD), which compresses the BoW histogram into a reversible binary signature on mobile devices, and restore MCVD to BoW in a lossy manner on the remote server. However, their method needs to retain different compression functions for different locations on mobile devices. Instead of utilizing individual compression function, we attempt to provide a universal compression method on mobile devices and investigate different reconstruction methods with contextual information adaptively on remote servers.

Rather than focusing on mobile devices, recent works utilize the computing power of remote servers (cloud) to reconstruct an image from its (compressed) local features [18], [25], [26], or generate distinctive image representations [27], [28]. Motivated by aforementioned works, our proposed framework considers both the limited resources on mobile devices and the stronger computing power on remote servers for MVS. Table I shows the overall comparison of the proposed method and prior works. We only consume a small amount of memory

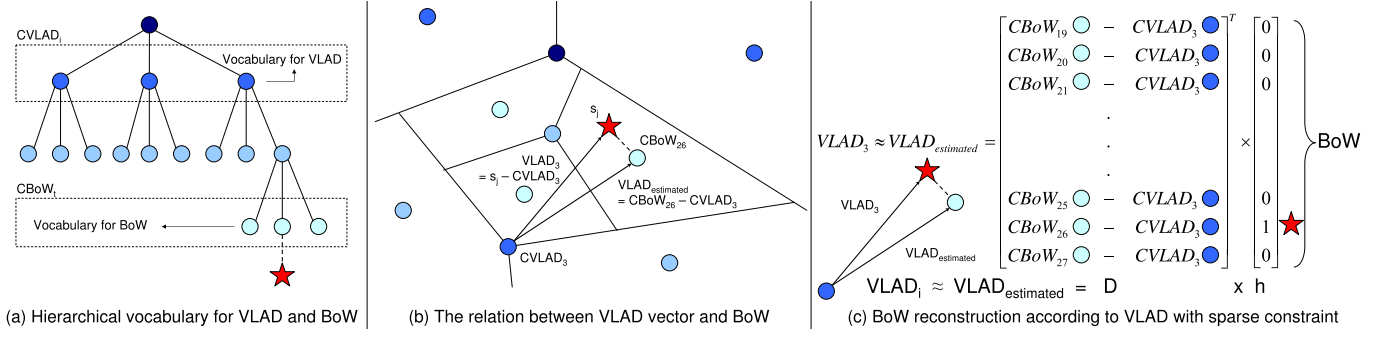


Fig. 2. Illustration of BoW reconstruction from VLAD. (a) We utilize an HKM to maintain both VLAD and BoW centers. Therefore, given a local feature (red star), we can estimate its possible BoW center (light blue circle) from its VLAD center (dark blue circle). (b) Due to the large vocabulary of BoW, we propose to calculate $VLAD_i$ by replacing the local feature (red star) with a BoW center. (c) Hence, with the connection between BoW and VLAD, we can roughly approximate BoW from VLAD.

to generate binary codes, and further reconstruct them into a context-aware BoW for better retrieval results.

III. CONTEXT-AWARE BoW RECONSTRUCTION

To leverage both the mobile- and cloud-based configurations for MVS, we utilize VLAD as an intermediate feature for compression and reconstruction. This is because VLAD only requires a small amount of memory for quantization tree, which is much smaller than the BoW model (the fourth column in Table I) and is suitable for mobile devices.¹ Hence, we aim to reconstruct BoW from VLAD for better retrieval accuracy (e.g., object queries). By observing the relation between VLAD and BoW in the HKM, we are able to approximate BoW from VLAD detailed in Section III-A. Besides, we can utilize contextual information to reconstruct different BoW histograms from the same VLAD, or obtain a better BoW even if the VLAD is approximated from binary codes (see Section IV). We investigate different reconstruction approaches in Section III-B. Moreover, we integrate the reconstruction method with a prior knowledge from the initial search results for better BoW reconstruction in Section III-C.

A. BoW Reconstruction From VLAD With Sparse Constraint

To reconstruct BoW from VLAD, we observe that they can be connected by their vocabulary trees. As shown in Fig. 2(a), we utilize a hierarchical vocabulary (e.g., hierarchical k -means (HKM) [30]) to construct the relation between VLAD and BoW. The top-layered vocabulary is used for VLAD, and the leaf nodes are BoW vocabulary. Therefore, if a local feature (red star) belongs to a VLAD center, we are able to estimate possible BoW centers (VWs) it belongs to (i.e., the subtree of the VLAD center).

We first define the generation process for VLAD. For each $VLAD_i$, we collect all the local features (s_j) quantized into the same VLAD center ($CVLAD_i$), and aggregate the difference between local features and the center as

$$VLAD_i = \sum_{s_j \in CVLAD_i} (s_j - CVLAD_i). \quad (1)$$

¹In our prototype, it takes 7.6 ms to extract VLAD from speeded-up robust feature (SURF) [29] for a 320×240 image in iPhone 5.

As an example shown in Fig. 2(b), we assume that the original local feature is the red star and circles are codeword centers. Different colors represent different levels of hierarchical vocabulary. Hence, for each local feature (s_j), it will contribute to $VLAD_i$ by the difference between the red star and the dark blue circle ($s_j - CVLAD_i$). The final VLAD will concatenate all the $VLAD_i$ values into a single feature.

By observing large vocabulary of BoW for image retrieval, we propose to substitute a BoW center ($CBoW_t$, light blue circle near the red star) for the local feature (red star). In other words, the difference ($s_j - CVLAD_i$) of $VLAD_i$ can be approximated by the light blue circle minus the dark blue circle ($CBoW_t - CVLAD_i$). Therefore, as shown in Fig. 2(c), $VLAD_i$ (1) can be approximated by Dh , where h represents BoW histogram and D is generated by the difference between BoW centers (subtree of $CVLAD_i$) and VLAD centers (i.e., $D = [CBoW_t - CVLAD_i]^T$).

Meanwhile, BoW histogram is usually a sparse vector, because the number of local features is relatively small. The sparse constraint also provides an opportunity to correctly reconstruct BoW histogram from VLAD. Therefore, given $VLAD_i$ (v), BoW (h) reconstruction can be formulated as

$$f_h = \min_h \|v - Dh\|_2^2 + \lambda \|h\|_1, \quad \text{subject to } h > 0 \quad (2)$$

where the first term measures the reconstruction quality between $VLAD_i$ (v) and approximated VLAD (Dh). λ is a regularization parameter that controls the sparsity of BoW (h). The formulation is similar to the sparse coding problem [31] but the dictionary (D) is predefined according to the difference between the centers of BoW and VLAD. Hence, we do not need to train the dictionary and can utilize SPArse Modeling Software [32], [33] for solving the above formulation. Note that this idea is also similar to compressive sensing that reconstructs sparse signals [34].

B. Context-Aware Dictionary Selection

It is reasonable, since mobile is augmented with geoinformation; hence, we further propose a context-aware dictionary selection (CADS) for BoW reconstruction. By utilizing contextual information, we are able to reconstruct different BoW histograms from the same VLAD or binary codes, as shown in Fig. 3. We utilize a single and universal vocabulary tree for

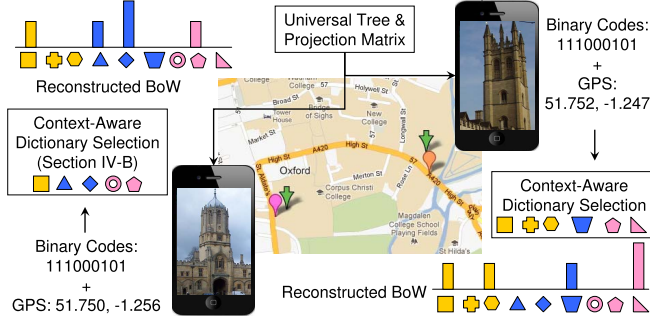


Fig. 3. Illustration of dictionary selection based on different contextual information. We utilize the same projection matrix and small quantization tree on mobile devices, and reconstruct different BoW representations on the server-side according to the contextual information (e.g., GPS).

BoW reconstruction; however, we dynamically select possible dictionary $[D_{\text{context}}$ in (2)] based on different contextual cues. Instead of using all VWs (e.g., 9 BoW centers [CBoW_19 to CBoW_27] in Fig. 2), we only retain few relevant VWs (e.g., top-ranked candidates by binary codes or GPS similarity). Hence, we consider both visual and contextual information for generating more discriminative BoW representations. Note that we can record possible VWs (subvocabularies) in an offline manner, or select them on the fly (e.g., calculating GPS similarities). Moreover, the selection process also leads to faster solving time, because the hypotheses of VWs are greatly reduced.

C. BoW Reconstruction With Prior Knowledge

In addition to CADS, we further utilize initial ranking results as prior knowledge to estimate a more powerful BoW representation. We know that the reranking process is a common query expansion method to provide a better performance. Hence, we generate a pseudo-BoW from top-ranked search results (i.e., obtained from binary codes in our experiments).² Based on the prior knowledge, our goal is to reconstruct BoW representation, which is similar to the approximated VLAD (Section IV-C) and the pseudo-BoW. Hence, the reconstruction operation can be formulated as

$$f_h = \min_h \alpha \frac{\|v - Dh\|_2^2}{N_1} + (1 - \alpha) \frac{\|h - h_0\|_2^2}{N_2} \quad (3)$$

where α stands for the influence between the first and the second terms. $N_1 = \|v\|_2^2$ and $N_2 = \|h_0\|_2^2$ are the normalization terms. The first term is followed by Section III-B to reconstruct BoW representation. The second term is to ensure that the reconstructed BoW is similar to the pseudo-BoW (h_0) that is generated from top-ranked results. By considering two different criteria, the proposed BoW reconstruction with prior knowledge (BRPK) can obtain a better BoW for MVS.³

²By considering the limited bandwidth for MVS, we only transmit a small amount of data (binary codes) to remote servers. It is efficient to retrieve images via binary codes. Nevertheless, we can also transmit VLAD and retrieve superior ranking results for better BoW reconstruction.

³Note that here we choose L2 regularization to speed up the reconstruction process, because the response time for image retrieval systems is also an important factor. Hence, with prior knowledge from ranking results, we can roughly estimate possible VWs and round down the reconstructed BoW histogram to enforce the sparsity in the final BoW representation.

To solve the proposed formulation, we find that it is equivalent to the generalized Tikhonov regularization and exists an optimal solution. Hence, we are able to compute the unique solution h^* of (3) analytically. Let $\alpha_1 = (\alpha/N_1)$ and $\alpha_2 = (1-\alpha)/N_2$, and a direct solution would lead to

$$h^* = (\alpha_1 D^T D + \alpha_2 I)^{-1} (\alpha_1 D^T v + \alpha_2 h_0). \quad (4)$$

However, if the inverse matrix is large (e.g., 10000×10000 , possible VWs), it is time-consuming to compute the solution. We can transform $D^T D$ to DD^T , which is related to the feature dimension (e.g., 128×128 , SIFT [35]). The transformation [36] is based on the identity of the inverse function $(I + AB)^{-1}A = A(I + BA)^{-1}$ and $(I + AB)^{-1} = I - A(I + BA)^{-1}B$. Then, we can rewrite (4) as

$$\begin{aligned} h^* &= \alpha_1 D^T (\alpha_1 DD^T + \alpha_2 I)^{-1} v \\ &\quad + [I - \alpha_1 D^T (\alpha_1 DD^T + \alpha_2 I)^{-1} D] h_0 \\ &= \alpha_1 D^T (\alpha_1 DD^T + \alpha_2 I)^{-1} (v - Dh_0) + h_0. \end{aligned} \quad (5)$$

We can efficiently reconstruct BoW histogram based on the above solution. Moreover, we can achieve better retrieval accuracy, because we not only consider the initial ranking results but also utilize both visual and contextual information.

IV. REVERSIBLE BINARY CODE GENERATION

Under the unstable wireless network, transmitting VLAD (e.g., 8192 dimensions, 32 kB) back to the remote server might still be infeasible. To further compress VLAD, we apply hash-based methods to generate binary codes on mobile devices for reducing the transmission cost. The binary hashing functions $[bh_k(\cdot)]$ can be formulated as

$$bh_k(x) = (\text{sgn}(w_k^T x) + 1)/2, \quad k = 1, \dots, K \quad (6)$$

where $x \in \mathbb{R}^d$ (zero mean) is the original (visual) feature and $w_k \in \mathbb{R}^d$ is a projection vector sampled from Gaussian distribution [random projection (RP)] [37] or learned from training data [38]. K is the total number of hashing functions (to generate K bits for each feature). Although it is compact for transmission, the limited memory of mobile devices enforces the projection matrix ($W \in \mathbb{R}^{d \times K}$) should be small as well. Hence, we introduce principal component analysis hashing (PCAH) in Section IV-A and investigate various ways for generating compact codes on mobile devices in Section IV-B. In Section IV-C, we reverse binary codes for obtaining approximated VLAD, which is used for BoW reconstruction.

A. Principal Component Analysis Hashing (Joint PCAH)

Recent works demonstrate that PCAH provides high binarization quality and retrieval performance [39], [40]. The projection matrix (W) of PCAH is learned from the covariance matrix (XX^T) of training data (X). By selecting the largest K eigenvectors to form W , we are able to generate binary codes based on (6).⁴ However, the projection matrix might be very

⁴We can apply other binarization or (semi-) supervised hashing [38] strategies, or compute weights for each bit (dimension) to improve retrieval accuracy [41]. However, we are to investigate BoW reconstruction from binary codes so we only utilize the standard PCAH in our experiments.

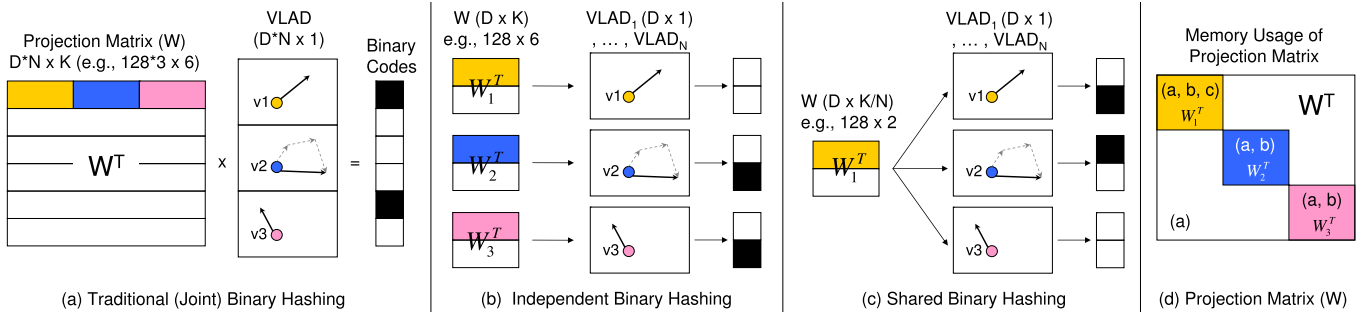


Fig. 4. Illustration and comparison of different binary hashing methods. (a) Traditional (joint) binary hashing projects the original (high-dimensional) feature space onto a reduced feature space. However, memory usage is proportional to the original and reduced features' dimensionality. (b) We can split the original feature (VLAD) into subfeatures (VLAD_{*i*}) so that the projection matrix can be reduced. (c) Moreover, we can learn a uniform (shared) projection matrix for all the subfeatures. The shared method needs more bits to achieve competitive retrieval accuracy compared with the traditional one (Section V-E); hence, it is a tradeoff between the performance and memory usage. (d) Memory usage of projection matrix on three approaches. The symbols in different rectangles represent the consumed memory. For example, the shared method (c) only requires the top-left (yellow) rectangle for the projection matrix.

huge if we directly apply PCAH on the high-dimensional feature space called joint PCAH in our experiments. For example, to generate 1024-b binary codes (K) from 12800D VLAD ($D \times N$), the projection matrix requires around 50 MB [i.e., $12800 \text{ B} \times 1024 \text{ B} \times 4 \text{ B}$, $D \times N \times K$ in Fig. 4(a)] for memory usage.⁵ As a result, the memory consumption might be similar to 1M vocabulary tree (e.g., 128 MB = 1M centers $\times$ 128 B). In other words, if we increase the dimension of the original or reduced features for better retrieval accuracy, the memory cost will be infeasible on mobile devices. Hence, we further consider memory-efficient binary hashing for MVS.

B. Memory-Efficient Binary Hashing (Ind. and Shared PCAH)

Followed by product quantization [42] and on-device image matching [17], we split high-dimensional feature (VLAD) into subfeatures (VLAD_{*i*}) and learn the projection matrix individually called independent (Ind.) PCAH. Moreover, we can learn a shared projection matrix for all the VLAD_{*i*} values called shared PCAH. As shown in Fig. 4(b) and (c), the memory cost of Ind. PCAH and shared PCAH is $D \times K$ and $D \times K/N$, respectively. Fig. 4(d) shows the overall comparison of memory consumption by different approaches. Compared with others, the shared approach only requires relatively small memory usage (W_1^T , top-left yellow rectangle). It is a tradeoff between the performance (bit length) and memory consumption on mobile devices. Note that these approaches are complementary to sparse learning (e.g., sparse PCA [43]), bilinear [44], or circulant approaches [45]. We can combine them with our proposed method to further reduce the memory consumption. Besides, as mentioned in [46], a simple componentwise sign binarization on VLAD can achieve good results. Hence, we will also discuss the results in Section V-E.

C. VLAD Approximation From Binary Codes

Based on Section IV-B, we can efficiently generate binary codes via PCA-based methods (i.e., $W^T x$) on

mobile devices. To reconstruct BoW from VLAD on remote servers (Section III), we have to reverse the retrieved binary codes and obtain approximated VLAD for the reconstruction. PCA-based methods have the orthogonal projection matrix property ($W^T W = I$); therefore, we can reverse binary codes by multiplying the same projection matrix (i.e., $x \sim W W^T x$). However, for binary hashing [i.e., $\text{sgn}(W^T x)$], we will lose more information due to the binarization process. This is also the reason why we propose to utilize contextual information for better BoW reconstruction (Section III-B). Besides, as mentioned in [10] and [39], we can apply an orthogonal transformation [e.g., random rotation (RR)] to mitigate the quantization error of binarization, and obtain better binary codes and the reversed feature. It is essential for the high-dimensional feature; hence, we will apply random orthogonal rotation on joint PCAH (joint PCAH-RR) in Section V-E.

V. EXPERIMENT RESULTS AND DISCUSSION

A. Experiment Setting

In our experiments, we construct 1M HKM with six levels and ten branches [30] for the BoW model, and choose the second level for VLAD (i.e., 100 centers). The distance measurement is L1 for BoW and L2 for VLAD in the retrieval process. We conduct our proposed method on two data sets. For mobile scenario, we choose Stanford MVS (SMVS) data set [47], [48], which contains 1193 single reference images with 3269 mobile queries across eight image categories (i.e., CD, DVD, Books, Video Clips, Landmarks, Business Cards, Text Documents, and Paintings). We resize images to small resolution (maximum height or width is 640) and use SURFs [29] on mobile devices and generate VLAD with 6400 dimensions. The evaluation metric is recall at Num (R@Num) and NDCG. We set the relevance score of ground truth images to be 1. Each query only has one reference (correct) image; hence, the ideal DCG is 1 and normalized discounted cumulative gain (NDCG) is equivalent to $1/\log_2(r+1)$, where r is the rank number of the reference image in the retrieval results. As reported in [17], BoW and REVV can achieve around 75% recall with spatial verification on top 50 candidates. To demonstrate the effect of our proposed method, we do not apply spatial verification on the image search results (i.e., evaluating on the initial ranking results).

⁵ D is the dimension of local features (e.g., SIFT and SURF), and N is the number of VLAD centers. $D \times N$ means the concatenated dimension of VLAD.

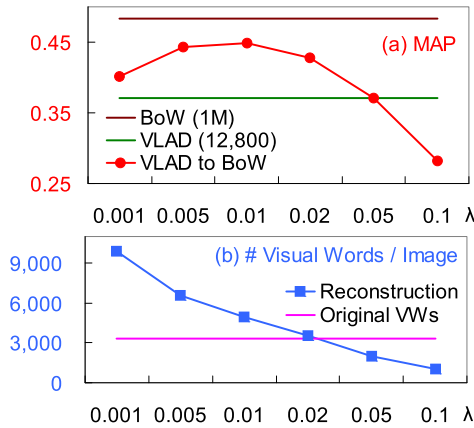


Fig. 5. Parameter sensitivity of λ in (2) on Oxford Buildings data set. (a) It shows that we can achieve similar results as the original BoW when selecting a proper value of λ . (b) Moreover, the value should be adjusted by the number of VWs we are to reconstruct. For Oxford data set, the averaged number of VWs is around 3350. Hence, we set $\lambda = 0.02$ in the experiments.

Moreover, for demonstrating the effect of image object retrieval, we choose Oxford Buildings (Oxford) data set [49], which contains 5062 images with 55 object queries. We use rootSIFT [35], [50] to generate 12 800D VLAD with intra normalization [12].⁶ In order to provide contextual information on Oxford, we generate GPS for each image followed by [51] that utilizes Gaussian error model to approximate GPS from the true location. As mentioned in [49], the data set is downloaded from 17 Flickr queries (keywords). Hence, we can obtain 16 buildings' GPS information (true location) from Wikipedia. The remaining Oxford keyword is randomly assigned to the other 16 keywords. The evaluation metric is mean average precision (MAP). The MAP of VLAD on Oxford is 0.371, and is similar to [10], which reports that the MAP of VLAD with 4096 dimensions is 0.304. The MAP of BoW and GPS on Oxford is 0.483 and 0.168, respectively. As reported in [49], the MAP of SIFT with HKM for BoW is 0.439. In our experiments, the original SIFT can achieve 0.422, which is similar to their paper. Hence, we utilize the more robust rootSIFT (0.483) for constructing vocabulary in our experiments. Meanwhile, for evaluating the effect of training data, we further utilize two additional data sets—Paris [52] and Landmarks [53]—for learning the vocabulary tree and projection matrix. For Landmarks data set, we randomly sample 30 images from 721 landmarks, and utilize around 22 000 images for training.

B. Experiments on the Choice of Lambda (λ)

The most important parameter of our proposed context-aware BoW reconstruction is λ in (2), because it will affect the reconstruction quality for image retrieval. Hence, we conduct experiments with different values of λ on Oxford data set. As Fig. 5(a) shows, the reconstructed BoW (VLAD to BoW) has similar retrieval accuracy as BoW when λ ranges between 0.005 and 0.02. We find that the number of reconstructed VWs (6500–3500) is larger than the original VWs (around 3350), as shown in Fig. 5(b). The reconstruction step may include

⁶Note that the proposed method can also apply to video retrieval, because we can extract features from each frame and aggregate them to form VLAD.

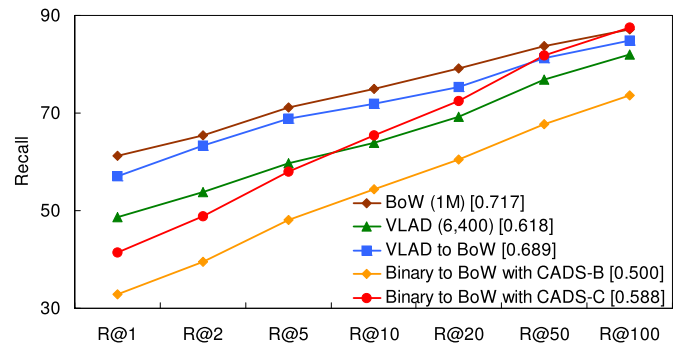


Fig. 6. BoW reconstruction from the original retrieved feature (i.e., VLAD or binary codes) on SMVS data set. It shows that the reconstructed BoW (VLAD to BoW) can achieve similar retrieval accuracy as BoW. However, if we apply CADs based on binary ranking results to estimate BoW from binary codes (Binary to BoW with CADs-B), the retrieval accuracy is worse than VLAD and BoW, because the binarization step loses too much information. By utilizing additional contextual information (category types) to select the most possible VWs as mentioned in Section III-B, we can achieve better retrieval results. The values in the square brackets represent NDCG scores.

noisy VWs; however, it can also be viewed as a by-product of soft (multiple) assignment [52]. When λ is small (0.001), we reconstruct too many noisy VWs and the retrieval accuracy decreases. Conversely, if λ is larger than 0.05, the MAP is worse than VLAD, because we only reconstruct few VWs and it is hard to retrieve similar images.

We observe that a proper value for λ is related to image resolution, because the number of extracted local features (VWs) depends on it. Therefore, for Oxford data set with 1024×768 pixels, we set λ to be 0.02, because the averaged number of VWs is around 3350. Note that the choice of λ for Oxford is not based on the highest MAP ($\lambda = 0.01$). Similarly, based on the experiments in Fig. 5(b), for SMVS data set with small resolution (maximum height or width is 640), we can directly adjust the value of λ to 0.1 without evaluating the performance (i.e., viewing Oxford as an independent data set for SMVS), because the averaged number of VWs is around 830. In other words, we can decide λ based on the desired number of reconstructed VWs, as shown in Fig. 5(b).

C. BoW Reconstruction on SMVS Data Set

First, we conduct experiments on SMVS data set. As Fig. 6 shows, the retrieval accuracy of BoW (even the reconstructed BoW) is better than VLAD (0.717 or 0.689 versus 0.618). The results confirm that it is necessary to reconstruct BoW from VLAD on the server-side. As mentioned in Section III, we can reconstruct BoW from VLAD or (reversible) binary codes. The blue curve (rectangle) in Fig. 6 shows that the reconstructed BoW (VLAD to BoW) can achieve competitive retrieval accuracy as the original BoW. This represents that the proposed method can successfully approximate the original BoW. However, if we reconstruct BoW directly from binary codes and utilize the approximated VLAD for the reconstruction, the results might be worse than the original binary codes, because the binarization process loses too much detailed information to reconstruct the original VLAD.⁷

⁷For better reconstruction results from binary codes to VLAD, we also adopt iterative quantization as proposed in [39].

TABLE II

COMPARISON OF RETRIEVAL ACCURACY WITH DIFFERENT BIT LENGTHS AND CONTEXTUAL CUES. THE MAP OF VLAD AND BoW IS 0.371 AND 0.483, RESPECTIVELY. IT SHOWS THAT JOINT BINARY HASHING (JOINT PCAH-RR) CAN PERFORM BETTER WHEN THE REDUCED DIMENSION IS LOW. MOREOVER, BY UTILIZING CONTEXTUAL INFORMATION FOR BoW RECONSTRUCTION, WE CAN ACHIEVE SIMILAR RETRIEVAL ACCURACY AS THE ORIGINAL BoW WHILE ONLY TRANSMITTING FEW BITS AND CONSUMING A SMALL AMOUNT OF MEMORY ON MOBILE DEVICES. SEE MORE EXPLANATIONS IN SECTION V-E. G INDICATES WE UTILIZE GPS INFORMATION IN THE RECONSTRUCTION STEP WHEREAS B ONLY CONSIDERS THE ORIGINAL BINARY RANKING RESULTS. BIN. INDICATES THE FINAL RANKING RESULTS ARE CONDUCTED FROM BINARY CODES AND OTHERS ARE RANKED BY BoW

Oxford Dataset (Bits)	1,000	2,000	4,000	8,000	12,800
Joint PCAH-RR [Bin.]	0.323	0.335	0.344	-	-
Ind. PCAH [Bin.]	0.302	0.354	0.368	0.382	0.370
Shared PCAH [Bin.]	0.252	0.306	0.346	0.375	0.390
Joint to BoW	0.291	0.314	0.314	-	-
Ind. to BoW	0.121	0.200	0.267	0.312	0.336
Shared to BoW	0.066	0.138	0.233	0.313	0.343
Joint to BoW [G]	0.398	0.410	0.413	-	-
Ind. to BoW [G]	0.267	0.314	0.364	0.390	0.405
Shared to BoW [G]	0.231	0.276	0.341	0.403	0.437
Joint to BoW [B]	0.362	0.378	0.384	-	-
Ind. to BoW [B]	0.240	0.307	0.363	0.404	0.404
Shared to BoW [B]	0.194	0.257	0.337	0.382	0.421
Joint to BoW [G+B]	0.373	0.397	0.406	-	-
Ind. to BoW [G+B]	0.228	0.309	0.380	0.419	0.431
Shared to BoW [G+B]	0.184	0.249	0.350	0.394	0.439
Joint [G+B] + BRPK [B]	0.423	0.456	0.445	-	-
Ind. [G+B] + BRPK [B]	0.400	0.465	0.469	0.481	0.454
Shared [G+B] + BRPK [B]	0.385	0.430	0.442	0.468	0.455

In order to demonstrate the effect of contextual information for BoW reconstruction, we assume that the class information is known (i.e., book, cd, painting, and so on, given by SMVS data set) for CADs. For real applications, we can apply classification for obtaining possible class information. Hence, for fair comparison, we only utilize GPS information and binary codes for Oxford data set in Section V-E and Table II. As shown in Fig. 6, when utilizing binary ranking results as a contextual cue for BoW reconstruction (Binary to BoW with CADs-B), the retrieval accuracy is worse than the original VLAD. However, when utilizing class information as an additional cue (Binary to BoW with CADs-C), we can achieve better retrieval accuracy (0.5–0.588). This is because the selection process can filter out those impossible (irrelevant) VWs. Note that we only utilize contextual information for BoW reconstruction (i.e., selecting D_{context} in Section III-B); hence, the proposed method can further combine with other reranking methods (e.g., late fusion) in the retrieval process.

D. VLAD Approximation From Reversible Binary Codes

As demonstrated in Section V-C, BoW reconstruction from binary codes might not be able to provide a good reconstruction results. Therefore, we investigate the intermediate step—VLAD approximation—in this section. As Fig. 7 shows, the approximated VLAD from binary codes (PCAH to VLAD [0.596]) has competitive performance as the original VLAD [0.618] and is better than binary codes (PCAH, 800 b [0.483]). The results confirm again that we

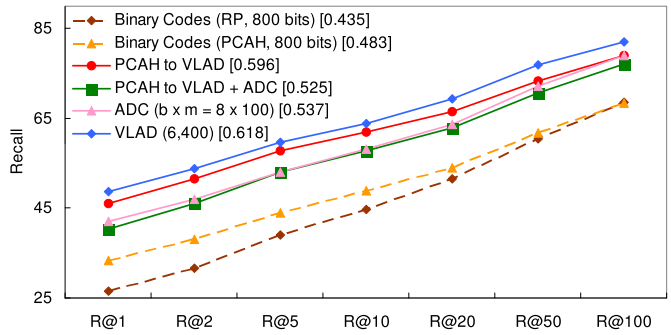


Fig. 7. Performance comparison of the original retrieved features (e.g., binary codes) and the approximated features on SMVS data set. The approximated VLAD (PCAH to VLAD) can achieve competitive retrieval accuracy as the original VLAD. Moreover, we can combine the approximated VLAD with ADC [42] to further reduce the memory consumption on the server-side. The values in the square brackets represent NDCG scores.

should utilize the computing power of remote servers to generate a better feature representation for image retrieval rather than using the original received feature. Another alternative is to increase the number of bits for binary codes with RP. Based on our experiments, when we utilize 6400 b with RP, the performance is similar to the approximated VLAD. However, it might be hard to obtain approximated VLAD from RP so we cannot further perform BoW reconstruction. By using PCA-based hashing methods, we can have better reconstruction results when increasing the number of bits and we will demonstrate the results in Section V-E.

We also compare and integrate our proposed method with asymmetric distance computation (ADC) as adopted in the original VLAD paper [10]. They compress VLAD into binary codes for database images and utilize the original VLAD for the query image to calculate the distance. Hence, they can greatly reduce the memory consumption in the database while retaining similar performance as the original approach. As Fig. 7 shows, the performance of the ADC method with 256 ($b = 8$, 2^8) centers for 100 subvectors ($m = 100$, VLAD_i) only slightly decreases. However, by using the ADC method, we still need to transmit the original VLAD to the remote server. Conversely, the proposed reversible binary codes (PCAH to VLAD) can achieve better retrieval accuracy. For fair comparison of memory usage on the server-side, we adopt the same approach as ADC that compresses database images into binary codes ($8 \text{ b} \times 100 \text{ b}$) but we utilize the approximated VLAD as query (PCAH to VLAD + ADC). It shows that we can achieve similar recall rate as the original ADC while only transmitting few bits (0.525 versus 0.537). Moreover, the ADC approach can further combine with inverted indexing as proposed in [42].

E. BoW Reconstruction on Oxford Buildings Data Set

Besides conducting experiments on SMVS data set, we also evaluate our proposed method on Oxford Buildings data set. We not only compare the results by using different bit lengths and binarization methods but also utilize different contextual information (e.g., binary ranking information and GPS) for context-aware BoW reconstruction. As shown in the second

to the fourth rows of Table II, the traditional binary hashing (joint PCAH-RR [Bin.]) can perform well in low dimension (e.g., 1000); however, the reduced dimension is limited to the number of images or features.⁸ This phenomenon is also demonstrated in [40] and [42] that joint dimension reduction will provide better compact representations and retrieval performance. As the number of bits increases (1000–12800 b), independent binary hashing (Ind. PCAH [Bin.]) and shared binary hashing (Shared PCAH [Bin.]) can have competitive or even better retrieval accuracy than the traditional method. Note that the MAP might be low on Oxford; nevertheless, it is still similar to [10] (i.e., VLAD with 64 VWs: 0.304 and PCA with 128D: 0.257).

As reported in Section V-D, the approximated VLAD (PCAH to VLAD) can achieve similar results as the original VLAD. Hence, we only focus on BoW reconstruction (via VLAD) from binary codes on Oxford data set. As shown in the fifth to the seventh rows of Table II, for fair comparison, we show the results without utilizing contextual information. The reconstructed BoW from joint PCAH-RR (joint to BoW) is better than others, because each bit is generated from (and can represent) a high-dimensional projection vector (i.e., 12800D) whereas other methods only consider (reconstruct) few dimensions (i.e., 128D). However, it might consume too much memory usage on mobile devices. An alternative way is to utilize independent or shared binary hashing and increase the number of bits (still few bits for transmission). As the number of bits is 8000 or 12800, we can achieve similar results as the joint one.

To mitigate the loss in binarization and achieve better results, we further utilize different contextual information (i.e., [G]PS, [B]inary ranking results) for context-aware BoW reconstruction. As Table II shows, the BoW reconstruction results are much better (i.e., better than the original VLAD: 0.371), and only slightly below the original BoW (0.483). This means that we only transmit few bits and consume a small amount of memory on mobile devices to achieve competitive results as BoW (especially for challenging object queries). Moreover, as mentioned in Section III-C, we can also utilize pseudo-BoW from top-ranked results to roughly estimate possible VWs for a given query (binary codes). As shown in the last three rows of Table II, we can further improve the retrieval accuracy by utilizing prior knowledge from the initial binary ranking results (BRPK [B]). However, if the top-ranked images are irrelevant to the target query, as the binary ranking results in Fig. 1, the improvement by utilizing this method might be limited.

We find that shared PCAH might be the most suitable way for MVS, because it only consumes 121.9 kB [projection matrix: $128D \times 128 \text{ b} \times 4 \text{ B}$ + hierarchical tree: $128D \times (10+100) \times 4 \text{ B}$] on mobile devices and 1.6 kB (12800 b) for transmission cost. As Table III shows, for fair comparison, we only utilize image content for retrieval and compare memory cost in an uncompressed manner. It shows that the BoW

TABLE III
COMPARISON OF THE MEMORY COST ON MOBILE DEVICES AND RETRIEVAL ACCURACY ON OXFORD DATA SET. * INDICATES EACH DIMENSION ONLY CONSUMES 8 b

Bytes	Transmission	Memory	MAP
BoW (1M) [14]	~5.4K	136M*	-
BoW (1M) [uncompressed]	13.4K	569M	0.483
VLAD (12,800)	51.2K	56K	0.371
Binarized VLAD [46]	1.6K	56K	0.331
REVV [16]	- (0.35K)	264K*	-
Ind. PCAH [Bin.] (2,000 bits)	0.25K	1080K	0.354
Shared PCAH [Bin.] (2,000 bits)	0.25K	66K	0.306
Shared PCAH [Bin.] (12,800 bits)	1.6K	122K	0.390
Shared to BoW [B]	1.6K	122K	0.421

TABLE IV
RETRIEVAL ACCURACY ON OXFORD DATA SET WITH DIFFERENT TRAINING DATA. AS DEMONSTRATED IN PRIOR WORK, THE PERFORMANCE WILL SLIGHTLY DECREASE WHEN WE TRAIN THE VOCABULARY ON DIFFERENT DATA SETS

MAP	Training data		
	Oxford	Paris	Landmarks
BoW (1M)	0.483	0.411	0.404
VLAD (12,800)	0.371	0.378	0.354
Binarized VLAD [46] (100 x 129 bits)	0.331	0.338	0.314
Shared PCAH [Bin.] (12,800 bits)	0.390	0.374	0.353
Shared to BoW [G+B] + BRPK [B]	0.455	0.391	0.373

method can achieve the best retrieval accuracy; however, it consumes more memory and transmission cost. VLAD-like approaches can greatly reduce memory consumption while the accuracy may slightly decrease. Note that Ind. PCAH can be viewed as a simplified version of REVV. Hence, we only increase a small amount of memory and transmission cost for better reconstruction results and retrieval accuracy.

We further utilize two additional data sets—Paris [52] and Landmarks [53]—for evaluating the effect of training data. As shown in Table IV, the retrieval accuracy slightly decreases when the vocabulary is trained on other data sets. This phenomenon has been demonstrated in prior work. Besides, similar to [46], the results of binarized VLAD are slightly below the original VLAD. In this paper, we assume that we can roughly replace the original local features with BoW centers for VLAD generation. Hence, we choose the best vocabulary tree (better BoW centers), and evaluate the effect of PCAH (projection matrix in Section IV) and reconstruction on independent data sets. As shown in Table V, we can achieve similar retrieval accuracy as training on the database images (Oxford). Based on these experiments, we observe that a suitable vocabulary tree is essential. Hence, we can further apply the concept of fine quantization [55] or vocabulary adaptation [12] to our proposed method.

For retrieval time, as demonstrated in prior work, binary matching and BoW matching with inverted indexing are very efficiency for real-time retrieval systems. Hence, in our experiments, we consume more time on the reconstruction step. From binary codes to approximated VLAD, it only contains a matrix multiplication. However, the BoW reconstruction step with sparse constraint (2) takes around 0.14 s

⁸Although VLAD has 12800 dimensions, the total number of Oxford data set only contains 5062 images. Therefore, we cannot generate more than 5061 dimensions by standard PCAH unless we apply random Fourier feature mapping [54] as adopted in [39].

TABLE V
RETRIEVAL ACCURACY ON OXFORD DATA SET WITH DIFFERENT
TRAINING DATA ON PCAH. WE CAN ACHIEVE SIMILAR
PERFORMANCE AS TRAINING ON OXFORD

MAP	Training data		
	Oxford	Paris	Landmarks
VLAD (12,800)		0.371	
Shared PCAH [Bin.] (12,800 bits)	0.390	0.393	0.384
Shared to BoW [G+B] (12,800 bits)	0.439	0.430	0.433
Shared to BoW [G+B] + BRPK [B]	0.455	0.450	0.449
BoW (1M)		0.483	

for each $VLAD_i$ (128D) when we consider all the possible VWs. To further apply dictionary selection, we can reduce it to 0.05 s (i.e., around 5 s per query). For integrating with prior knowledge (Section III-C), we relax the sparse constraint and thus reduce the reconstruction time to 0.51 s per query. Moreover, taking advantages of cloud, we can parallel the reconstruction step for each $VLAD_i$ and utilize multiple cloud servers to further reduce the reconstruction time.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose context-aware BoW reconstruction that utilizes the computing power of remote servers for MVS. We focus on generating reversible and memory-efficient binary codes on mobile devices, and attempt to reconstruct them to a better BoW representation on remote servers (cloud). Hence, the proposed method only transmits few bits to the remote server. By observing the relation between VLAD and BoW, we can reconstruct BoW from VLAD or binary codes. Moreover, we can select possible visual features (VWs) according to the contextual information (e.g., top-ranked images, category, and GPS), and further incorporate with prior knowledge from the initial (binary) ranking results. Experimental results show that the proposed method can achieve better retrieval results compared with the original retrieved feature (e.g., VLAD or binary codes). In the future, we will investigate how to utilize extra information on the server-side and adopt better binary reconstruction methods, such as [56].

REFERENCES

- [1] B. Girod *et al.*, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, Jul. 2011.
- [2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2003, pp. 1470–1477.
- [3] J.-M. Guo, H. Prasetyo, and J.-H. Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 466–481, Mar. 2015.
- [4] J. He *et al.*, "Mobile product search with bag of hash bits and boundary reranking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3005–3012.
- [5] R. Ji *et al.*, "Towards low bit rate mobile visual search with multiple-channel coding," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 573–582.
- [6] Z. Liu, H. Li, L. Zhang, W. Zhou, and Q. Tian, "Cross-indexing of binary SIFT codes for large-scale image search," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2047–2057, May 2014.
- [7] C.-Y. Yeh *et al.*, "Me-link: Link me to the media—Fusing audio and visual cues for robust and efficient mobile media interaction," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 147–150.
- [8] S. Lu, T. Mei, J. Wang, J. Zhang, Z. Wang, and S. Li, "Exploratory product image search with circle-to-search interaction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1190–1202, Jul. 2015.
- [9] D. M. Chen and B. Girod, "Memory-efficient image databases for mobile visual search," *IEEE MultiMedia*, vol. 21, no. 1, pp. 14–23, Jan./Mar. 2014.
- [10] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
- [11] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 653–656.
- [12] R. Arandjelović and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1578–1585.
- [13] O. Chum, A. Mikulík, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. CVPR*, 2011, pp. 889–896.
- [14] D. M. Chen, S. S. Tsai, V. R. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, "Tree histogram coding for mobile image matching," in *Proc. Data Compress. Conf.*, 2009, pp. 143–152.
- [15] V. R. Chandrasekhar *et al.*, "Compressed histogram of gradients: A low-bitrate descriptor," *Int. J. Comput. Vis.*, vol. 96, no. 3, pp. 384–399, Feb. 2012.
- [16] D. Chen *et al.*, "Residual enhanced visual vectors for on-device image matching," in *Proc. Conf. Rec. 45th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2011, pp. 850–854.
- [17] D. M. Chen *et al.*, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Process.*, vol. 93, no. 8, pp. 2316–2327, Aug. 2013.
- [18] L. Dai, H. Yue, X. Sun, and F. Wu, "IMShare: Instantly sharing your mobile landmark images by search-based reconstruction," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 579–588.
- [19] D. M. Chen and B. Girod, "A hybrid mobile visual search system with compact global signatures," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 1019–1030, Jul. 2015.
- [20] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, and Q. Tian, "Towards codebook-free: Scalable cascaded hashing for mobile image search," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 601–611, Apr. 2014.
- [21] Z. Liu, H. Li, W. Zhou, R. Zhao, and Q. Tian, "Contextual hashing for large-scale image search," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1606–1614, Apr. 2014.
- [22] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—A survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.
- [23] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua, "Boosting binary keypoint descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2874–2881.
- [24] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 597–610, Mar. 2015.
- [25] P. Weinzaepfel, H. Jégou, and P. Pérez, "Reconstructing an image from its local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 337–344.
- [26] H. Yue, X. Sun, J. Yang, and F. Wu, "Cloud-based image coding for mobile devices—Toward thousands to one compression," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 845–857, Jun. 2013.
- [27] Z. Liu, H. Li, W. Zhou, T. Rui, and Q. Tian, "Making residual vector distribution uniform for distinctive image representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 375–384, Feb. 2016.
- [28] R. Hong, Y. Yang, M. Wang, and X.-S. Hua, "Learning visual semantic relationships for efficient visual retrieval," *IEEE Trans. Big Data*, vol. 1, no. 4, pp. 152–161, Dec. 2015.
- [29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [30] D. Nistér and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2161–2168.
- [31] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 801–808.
- [32] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.
- [33] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

- [34] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
- [35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [36] K. B. Petersen and M. S. Pedersen. (Nov. 2012). *The Matrix Cookbook, Version 20121115*. [Online]. Available: <http://www2.imm.dtu.dk/pubdb/p.php?3274>
- [37] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. 34th Annu. ACM Symp. Theory Comput. (STOC)*, 2002, pp. 380–388.
- [38] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.
- [39] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2012.
- [40] H. Jégou and O. Chum, "Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 774–787.
- [41] Y.-G. Jiang, J. Wang, X. Xue, and S.-F. Chang, "Query-adaptive image search with hash codes," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 442–453, Feb. 2013.
- [42] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
- [43] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, Mar. 2010.
- [44] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik, "Learning binary codes for high-dimensional data using bilinear projections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 484–491.
- [45] F. X. Yu, S. Kumar, Y. Gong, and S.-F. Chang, "Circulant binary embedding," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 946–954.
- [46] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 3384–3391.
- [47] V. R. Chandrasekhar *et al.*, "The Stanford mobile visual search data set," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 117–122.
- [48] V. R. Chandrasekhar *et al.* (2013). Stanford mobile visual search dataset. Stanford Digital Repository. [Online]. Available: <http://purl.stanford.edu/rb470rw0983>
- [49] 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.
- [50] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2911–2918.
- [51] D. M. Chen *et al.*, "City-scale landmark identification on mobile devices," in *Proc. IEEE CVPR*, Jun. 2011, pp. 737–744.
- [52] 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.
- [53] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 584–599.
- [54] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. NIPS*, 2007, pp. 1177–1184.
- [55] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning vocabularies over a fine quantization," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 163–175, 2013.
- [56] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.



Yin-Hsi Kuo received the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, where she is currently pursuing the Ph.D. degree with the Graduate Institute of Networking and Multimedia.

Her current research interests include multimedia content analysis and image retrieval.



Winston H. Hsu (S'03–M'07–SM'12) received the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA.

He was with the multimedia software industry. Since 2007, he has been a Professor with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, where he is also with the Department of Computer Science and Information Engineering. His current research interests include computer vision, machine intelligence, image/video indexing and retrieval, and mining over large-scale databases.

Dr. Hsu serves as an Associate Editor of the *IEEE TRANSACTIONS ON MULTIMEDIA* and on the Editorial Board of the *IEEE Multimedia Magazine*.