LEARNING MULTIPLE CODEBOOKS FOR LOW BIT RATE MOBILE VISUAL SEARCH

Jie Lin[†‡] *Ling-Yu Duan*[‡⋆] *Jie Chen*[‡] *Rongrong Ji*[‡] *Siwei Luo*[†] *Wen Gao*[‡]

[†]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
[‡]The Institute of Digital Media,School of EE&CS, Peking University, Beijing, 100871, China
{jielin, lingyu, cjie, rrji}@pku.edu.cn swluo@bjtu.edu.cn wgao@pku.edu.cn

ABSTRACT

Compressing a query image's signature via vocabulary coding is an effective approach to low bit rate mobile visual search. State-of-the-art methods concentrate on offline learning a codebook from an initial large vocabulary. Over a large heterogeneous reference database, learning a single codebook may not suffice for maximally removing redundant codewords for vocabulary based compact descriptor. In this paper, we propose to learn multiple codebooks (m-Codebooks) for extremely compressing image signatures. A query-specific codebook (q-Codebook) is online generated at both client and server sides by adaptively weighting the off-line learned multiple codebooks. The q-Codebook is subsequently employed to quantize the query image for producing compact, discriminative, and scalable descriptors. As q-Codebook may be simultaneously generated at both sides, without transmitting the entire vocabulary, only small overhead (e.g. codebook ID and codeword 0/1 index) is incurred to reconstruct the query signature at the server end. To fulfill m-Codebooks and q-Codebook, we adopt a Bi-layer Sparse Coding method to learn the sparse relationships of codewords vs. codebooks as well as codebooks vs. query images via $l1$ regularization. Experiments on benchmarking datasets have demonstrated the extremely small descriptor's supervior performance in image retrieval.

Index Terms— Mobile visual search, visual vocabulary, universal quantization, compact descriptor

1. INTRODUCTION

With the increasing popularity of phone camera devices, mobile visual search becomes more and more attractive, such as mobile landmark search, mobile product search, and mobile CD/book cover search. In general, most existing mobile visual search systems follow a client-server architecture. In the server end, a visual search system is maintained, typically based on a Bag-of-Words (BoW) model [1] as well as a scalable inverted indexing on a visual vocabulary. In online search, a query is sent through the wireless network to the server end, where near-duplicated search is conducted to find out the best matched images.

Over a bandwidth constrained (3G) wireless network, sending an entire image may suffer from serious latency. Research efforts have been devoted to directly extracting visual descriptors on a mobile device for low bit rate query transmission. Beyond existing local descriptors (e.g. SIFT [2], SURF [3], PCA-SIFT [4]), recent works put more emphasis on the compactness of descriptors. The first group comes from direct compression of local descriptors [5][6]. The second group attempts to compress the BoW based signature [7][8][9][10] rather than local descriptors, which gains high compression rate without any serious loss of discriminability.

More recent vocabulary coding approaches [9][10] have reported promising search performance at extremely low bit rate, say hundreds of bits. Side information (e.g. GPS, RFID tags) associated with images are employed to define a channel (i.e. data partition over reference database), in which machine learning techniques (e.g. Boosting) are adopted to learn a codebook (a subset of initial BoW vocabulary) within each channel. The query image's compact signature is obtained by quantizing local descriptors with a single codebook (channel dependent). However, the codewords redundancy issue arises from learning a single codebook. That is, there would exist seriously redundant codewords in a single codebook for representing a query in its channel, especially when a channel involves heterogeneous images. Redundant codewords could degenerate the query descriptor's compactness and discriminative power. Furthermore, the codebook is often of fixed size, yielding less scalability.

In this paper, we formulate the issue of removing the codewords redundancy from learning a single codebook (s-Codebook) as a problem of learning multiple codebooks (m-Codebooks). Based on m-Codebooks, a query-specific codebook (q-Codebook) is online generated by adapting codebook weights to each incoming query image, and the compact descriptor is subsequently yielded by the resulting q-Codebook. To fulfill the compactness and discriminability, we adopt a Bi-layer Sparse Coding method to learn the sparse characteristics of codewords vs. codebooks and codebooks vs. images. Figure 1 illustrates the process with m-Codebooks and q-Codebook.

Our contributions are two-fold. First, we propose a novel m-Codebooks learning to further reduce the redundant codewords in generating vocabulary based descriptors [9][10]. Second, we introduce a bi-layer sparse coding method to learn the sparsity priors effectively, yielding more compact, discriminative and scalable descriptors to fulfill desirable image retrieval performance.

The remaining of this paper is organized as follows. Section 2 formulates the m-Codebooks learning problem. In Section 3, we introduce bi-layer sparse coding to tackle the m-Codebooks learning process. Experimental evaluation is given in Section 4. Finally, we conclude this paper in Section 5.

2. M-CODEBOOKS LEARNING

Problem Formulation We aim to (1) reduce redundant codewords within the learned s-Codebook in state-of-the-art vocabulary coding approaches [9][10]; and (2) improve the weak scalability in s-Codebook by yielding q-Codebook over m-Codebooks. Suppose there are N images and the initial vocabulary $\mathbf{V}$ consists of M codewords, each image $\mathbf{I}_n$ is represented as $\mathbf{V}_n \in \mathbb{R}^M$, where the m^{th} entry of $\mathbf{V}_n$ denotes the frequency of the m^{th} codeword in $\mathbf{I}_n$.

[⋆] Corresponding Author

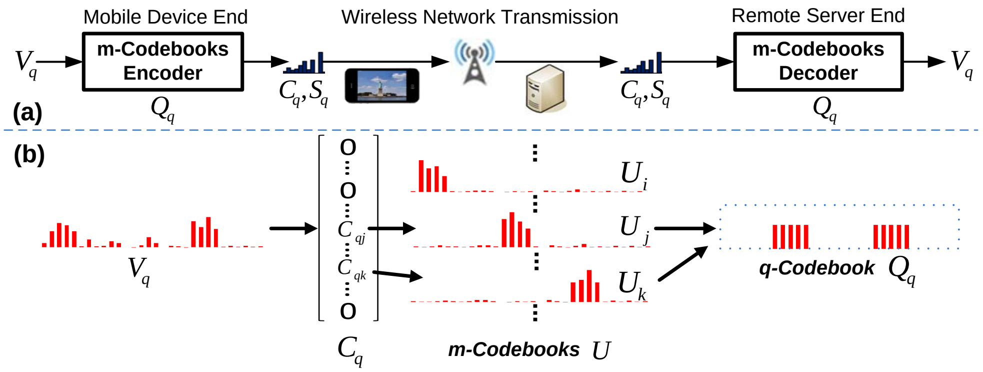

Fig. 1. Learning m-Codebooks and generating q-Codebook to yield compact, discriminative, and scalable descriptor: (a) mobile visual search pipeline, and (b) the generation process of q-Codebook based on m-Codebooks.

Previous vocabulary coding works [9][10] aim to learn a transform $\mathbf{M}_{M\times\overline{K}}$ ($\mathbf{M}^T$ is the encoder and $\mathbf{M}$ is the decoder) from $\mathbf{V}$ to a much more compact codebook $\overline{\mathbf{U}} \in \mathbb{R}^{\overline{K}}$ ($\overline{K} \ll M$), which transforms $\mathbf{V}_n$ into low cost ordinary code $\overline{\mathbf{U}}_n$:

$$\overline{\mathbf{U}}_n = f(\mathbf{V}_n) \triangleq \mathbf{M}^T\mathbf{V}_n \tag{1}$$

Towards low bit rate vocabulary coding, s-Codebook is generated by choosing $\overline{K}$ codewords from vocabulary $\mathbf{V}$.

To further reduce the codewords redundancy in Equation 1, we propose to learn m-Codebooks for generating a query-specific codebook, which is in spirit similar to the idea of universal quantization [11]. Universal quantization originates from running multiple lossless codes in parallel and choosing the one producing the fewest bits for a period of time, sending a small amount of overhead to inform the decoder which code the encoder was using [11]. Better performance tradeoffs can be achieved by allowing both rate and distortion to vary. In the scenario of visual search, rate means the descriptor's compactness, while distortion means the ranking loss in image retrieval. Motivated by universal quantization, codes with smaller dimension might be more efficient since separate codebooks can be used for distinct local behavior (e.g. query-specific behavior).

We formulate the problem of learning K m-Codebooks $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_K]$ in parallel and choosing the most relevant codebooks for query image $\mathbf{I}_n$ using its codebook weights $\mathbf{C}_n \in \mathbb{R}^K$ to produce as small distortion as possible in search performance. The m^{th} entry of $\mathbf{U}_k \in \mathbb{R}^M$ denotes the weight of the m^{th} codeword in k^{th} codebook. Intuitively, the codewords with larger weights are more representative in the corresponding codebook. The k^{th} entry of $\mathbf{C}_n$ represents the weight of codebook $\mathbf{U}_k$ for image $\mathbf{I}_n$. The larger codebook weight, the more important role the codebook plays in representing the query image. Each codebook $\mathbf{U}_k$ actually represents a different type of local behavior. Ideally, each query should involve as fewer codebooks as possible; meanwhile each codebook contains a small number of codewords in the entire vocabulary. This assumption is subsequently validated by our experiments. Therefore, we inject sparsity constraints to both m-Codebooks matrix $\mathbf{U}$ (each $\mathbf{U}_k$ should be sparse, i. e. with just a few non-zero entries) and codebook weights vector $\mathbf{C}_n$ for each image $\mathbf{I}_n$ as well. The m-Codebooks based vocabulary coding is then defined as follows:

$$\mathbf{S}_n \triangleq \mathbf{U}\mathbf{C}_n \tag{2}$$

from the perspective of universal quantization, $\mathbf{S}_n \in \mathbb{R}^M$ may be considered as a sort of universal code of image $\mathbf{I}_n$ in the low bit rate visual search. Our objective is to use $\mathbf{S}_n$ to approximate $\mathbf{V}_n$ with less information loss. As with a small number of non-zero entries, $\mathbf{S}_n$ can be also regarded as the BoW quantization of image $\mathbf{I}_n$ based on tailored q-Codebook with $\overline{M}$ codewords ($\overline{M} \ll M$).

Learning Goal Given database images $\mathbf{I} = [\mathbf{V}_1, \ldots, \mathbf{V}_N] \in \mathbb{R}^{M\times N}$, we aim to learn m-Codebooks $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_K] \in \mathbb{R}^{M\times K}$ from initial vocabulary $\mathbf{V}$ as well as the codebook weights $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_N] \in \mathbb{R}^{K\times N}$ for image sets $\mathbf{I}$.

q-Codebook Generation Given a query $\mathbf{I}_q$, we generate a q-Codebook $\mathbf{Q}_q \in \mathbb{R}^{\overline{M}}$ for BoW quantization based on the learned m-Codebooks $\mathbf{U}$ and $\mathbf{C}_q$ (see Figure 1(b)). We first represent query $\mathbf{I}_q$ with codebook weights $\mathbf{C}_q$ (using Equation 5). As $\mathbf{C}_q$ tends to be sparse, we assume there are k non-zero codebook weights. We select the columns of $\mathbf{U}$ corresponding to the k codebooks and denote them as $\{\widehat{\mathbf{U}}_i\}_{i=1}^k$, each $\widehat{\mathbf{U}}_i$ is also sparse. Then we put the codewords with non-zero weights in $\{\widehat{\mathbf{U}}_i\}_{i=1}^k$ into a union set and use this union set as q-Codebook $\mathbf{Q}_q$. This procedure ensures that the codebook generation is adaptive to each incoming query. Furthermore, $\overline{M}$ will be much smaller than M due to the sparsity of $\{\widehat{\mathbf{U}}_i\}_{i=1}^k$ and $\mathbf{C}_q$.

Compared to [9][10], learning m-Codebooks brings about two advantages: (1) the codewords in $\mathbf{Q}_q$ from vocabulary $\mathbf{V}$ is tailored to each query $\mathbf{I}_q$ with the guidance of m-Codebooks, which not only produces a more compact descriptor, but promotes discriminability due to sparsity property; (2) the descriptor is more scalable, rather than relying on a codebook of fixed size $\overline{K}$ for all query images. Moreover, the descriptor size $\overline{M}$ is adaptive to the query difficulty. For instance, an image with complex background may involve more codebooks and codewords than a plain image.

3. BI-LAYER SPARSE CODING

In this section, we aim to learn $\mathbf{U}$ and $\mathbf{C}$. Our objective is to optimally approximate $\mathbf{V}_n$ using universal code $\mathbf{S}_n$. We choose the squared $l2$-norm of the difference between $\mathbf{V}_n$ and $\mathbf{S}_n$ to measure the BoW reconstruction error: $\| \mathbf{V}_n - \mathbf{U}\mathbf{C}_n \|_2^2$. Suppose that the codeword-codebook matrix $\mathbf{U}$ and codebook-image matrix $\mathbf{C}$ are sparse. We use $l1$ norm regularization on both $\mathbf{U}$ and $\mathbf{C}$ to fulfill the sparsity constraint as follows:

$$\min_{\mathbf{U},\{\mathbf{C}_n\}} \sum_{n=1}^{N} \| \mathbf{V}_n - \mathbf{U}\mathbf{C}_n \|_2^2 + \lambda \sum_{k=1}^{K} \| \mathbf{U}_k \|_1 + \beta \sum_{n=1}^{N} \| \mathbf{C}_n \|_1 \tag{3}$$

where $\lambda \geq 0$ and $\beta \geq 0$ are the parameters controlling the regularization on $\mathbf{U}$ and $\mathbf{C}$ respectively. The larger values λ or β, the more

Algorithm 1 Bi-layer Sparse Coding

```
**Input:** $\mathbf{I} \in \mathbb{R}^{M \times N}$
Generate random matrix $\mathbf{C}^0 \in \mathbb{R}^{K \times N}$
for t = 1 : T do
    $\mathbf{U}^t \leftarrow UpdateU(\mathbf{I}, \mathbf{C}^{t-1})$
    $\mathbf{C}^t \leftarrow UpdateC(\mathbf{I}, \mathbf{U}^t)$
end for
**Output:** $\mathbf{U}^t$, $\mathbf{C}^t$
```

sparse $\mathbf{U}$ and $\mathbf{C}$ are. Equation 3 is regarded as a sparse coding technique [12]. Here we have two layers of sparsity constraints (named as Bi-layer Sparse Coding in this paper).

Optimization The optimization problem of Equation 3 is non-convex. But fixing one variable (either $\mathbf{U}$ or $\mathbf{C}$), the objective function with respect to the other is convex. So we alternately minimize Equation 3 with respect to $\mathbf{U}$ or $\mathbf{C}$, as showed in Algorithm 1.

Update $\mathbf{U}$. When $\mathbf{C}$ is fixed, the update of $\mathbf{U}$ can be decomposed into M independent problems, each corresponding to one row of $\mathbf{U}$:

$$\min_{\mathbf{U}_m^T} \parallel \mathbf{V}_m^T - \mathbf{C}^T \mathbf{U}_m^T \parallel_2^2 + \lambda \parallel \mathbf{U}_m^T \parallel_1 \tag{4}$$

where $\mathbf{V}_m$ and $\mathbf{U}_m$ are the mth row of $\mathbf{I}$ and $\mathbf{U}$, $m = 1, \dots, M$. Then we choose coordinate descent technique [13] to solve Equation 4, resulting the following update rule:

$$w_{mk} \leftarrow r_{mk} - \sum_{l \neq k} q_{kl} u_{ml}$$

$$u_{mk} \leftarrow \frac{(\mid w_{mk} \mid - \frac{1}{2}\lambda)_+ sign(w_{mk})}{q_{kk}}$$

where q_{ij} and r_{ij} are the $(ij)^{th}$ entries of $K \times K$ matrix $Q = CC^T$ and $M \times K$ matrix $R = IC^T$ respectively, $k = 1, \dots, K$.

Update $\mathbf{C}$. Likewise, the update of $\mathbf{C}$ with $\mathbf{U}$ fixed can be also decomposed into N independent problems, each corresponding to one column of $\mathbf{C}$:

$$\min_{\mathbf{C}_n} \parallel \mathbf{V}_n - \mathbf{U}\mathbf{C}_n \parallel_2^2 + \beta \parallel \mathbf{C}_n \parallel_1 \tag{5}$$

which executes the similar procedure as Equation 4.

4. EXPERIMENTAL RESULTS

Datasets (1) *The PKU Landmark Benchmark Subset (PKUBench)*: The PKUBench subset consists of 5007 scene photos, organized into 170 landmark locations from the Peking University Campus. There are in total 567 queries from 170 landmarks, and on average 34 reference images per query. Rich photograph scenarios (e.g. diverse angles, shots, blurring, night, etc.) are involved. (2) *Zubud Database*: Zubud contains 1005 color images of 201 buildings or scenes (5 images per object) and 115 queries. (3)*UKBench Database*: UKBench contains 10,000 images with 2500 objects, including general indoors objects (CD Covers, books, etc.) or scenes. There are four images per object involving sufficient variances in viewpoints, rotations, lighting conditions, scales, occlusions and affine transforms. Note that these datasets have been included in MPEG CDVS benchmarking datasets.

Parameters and Evaluation We choose m-Codebooks parameter K from a set of values $\{30, 50, 80, 100, 150, 200\}$. Regularization parameters λ and β are adjusted in the interval $[0.01, 1]$ and $[0.01, 1]$, respectively. In subsequent retrieval experimental results, we select the optimal parameters $K = 50$, $\lambda = 0.5$, and $\beta = 1.0$. The effects of different parameters are not discussed due to space limit. How to effective and efficient select multiple parameters is included in our next work.

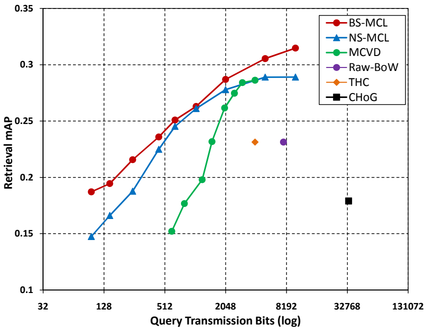

Fig. 2. Compression rate versus ranking distortion of our m-Codebooks learning and the comparison with the state-of-the-arts on PKUBench (the most challenging MPEG CDVS dataset).

We use mean Average Precision (mAP) to evaluate the image retrieval performance. Given in total Q queries, mAP is defined as follows:

$$mAP = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(\frac{\sum_{r=1}^{N} P(r)}{\# - of - relevant - images} \right)$$

where N_q is the number of queries; N the number of relevant images for the i^{th} query; $P(r)$ is the precision at rank r.

Baselines (1) *Raw Bag-of-Words (Raw-Bow)*: Transmitting the entire BoW has the lowest compression rate. However, it provides an upper bound in mAP with traditional TF-IDF indexing scheme. (2) *Tree Histogram Coding (THC)* [7]: Chen et al. applied residual coding to compress the BoW histogram. (3) *Compressed Histogram of Gradients (CHoG)* [5]: CHoG is the state-of-the-art compact local descriptor. As m-Codebooks and q-Codebook work on the quantized descriptors, learning multiple codebooks can be applied to CHoG. The subsequent comparison with CHoG actually presents the comparison between pre-quantization and post-quantization compact descriptor scheme. (4) *Multiple-Channel Coding based compact Visual Descriptor (MCVD)* [10]: MCVD presents the state-of-the-art vocabulary coding descriptor, which belongs to post-quantization methods. (5) *Non-Sparse m-Codebooks Learning (NS-MCL)*: This is a variant of our model, which relaxes the bi-layer $l1$ norm regularization. We introduce NS-MCL to investigate the sparsity property of m-Codebooks. (6) *Bi-layer Sparse m-Codebooks Learning (BS-MCL)*: BS-MCL is our proposed m-Codebooks learning method.

Rate Distortion Analysis We perform comparison with state-of-the-art methods over extensive datasets. As illustrated in Fig.2, over the challenging PKUBench, our method achieves the highest compression rates subject to a given mAP, and the best retrieval performance at a fixed compression rate. In addition, BS-MCL outperforms NS-MCL, which shows that the bi-layer sparsity constraints over m-Codebooks bring about more discriminative descriptors at the same bits.

Sparsity Analysis Table 1 shows the sparsity comparison of codewords vs. codebooks (matrix U) / codebooks vs. images (matrix C) between NS-MCL and BS-MCL models on PKUBench, using different m-Codebooks parameters K. We estimate the sparsity of U or C by computing the average ratio of non-zero entries in each column of U or C. From Table 1, the non-zero entries of BS-MCL

Table 1. Comparison of codewords vs. codebooks (matrix U) / codebooks vs. images (matrix C) sparsity between the NS-MCL and BS-MCL models on PKUBench, with different m-Codebooks parameter K.

K	30	50	80	100	150	200
NS-MCL (%)	15.73/19.49	20.51/19.72	27.41/20.71	38.35/23.12	48.46/25.81	46.03/26.88
BS-MCL (%)	2.69/1.75	2.34/1.56	3.18/1.44	2.54/1.41	2.76/1.53	2.73/1.57

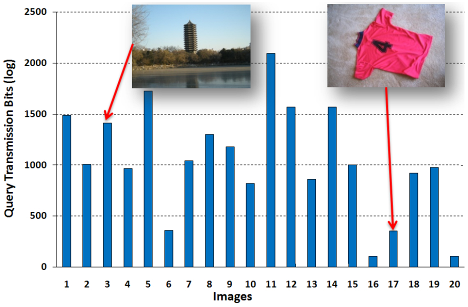

Fig. 3. Evaluation of descriptor's scalability based on our m-Codebooks learning.(The horizontal axis: image index, 20 images from PKUBench, Zubud, and UKBench; The vertical axis: upstream transmission bits per query)

are much smaller than NS-MCL on both U and C. We argue that the more sparse U and C are, the more compactness of descriptors we may achieve. That is, BS-MCL tends to yield more compact descriptors than NS-MCL.

Scalability We qualitatively study the descriptor scalability in length. 20 images are randomly selected from the datasets and we evaluate their upstream transmission bits, respectively. As illustrated in Figure 3, the descriptors of different query images vary in coding length of (0/1) bits, where each bit indicates hit/non-hit of a codeword. As the size of q-codebook is adaptive to each query, our m-Codebooks descriptor yield more flexible scalability while previous works are only with the codebook of fixed length [9][10].

Case Study We collect a few real-world challenging queries (different scales, illumination changes, occlusions, or blurring). Figure 4 shows that our m-Codebooks method can better preserve the ranking precision over the original Raw-Bow of high dimensions, based on more compact q-Codebook and m-Codebooks.

5. CONCLUSIONS

We have proposed a novel m-Codebooks learning and q-Codebook generation approach to reduce redundant codewords in arising vocabulary coding in very low bit rate mobile visual search. With a bi-layer sparse coding method, we have successfully incorporate the sparsity priors for generating more compact, discriminative, and scalable descriptors. Comprehensive experiments have validated the sparsity assumption and show significant improvements over the state-of-the-art BoW compression techniques. More investigation on universal quantization in vocabulary coding will be included in future work. Based on reference databases, how to better tune performance tradeoffs between rate and distortion need further study.

6. ACKNOWLEDGEMENT

This work was supported by National Basic Research Program of China (2009CB320902), in part by Chinese National Nature Science Foundation (60902057) and CADAL Project Program.

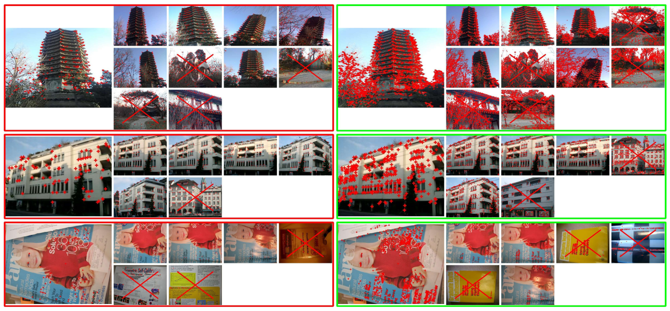

Fig. 4. Case Study of queries in challenging photograph scenarios. In each box, the left photo is the query and the returning results are listed on the right. The red framed box corresponds to BS-MCL, while the green framed box corresponds to Raw-Bow. The query examples are from Top: PKUBench; Middle: Zubud; Bottom: UKBench. The descriptors involved in matching between query and reference images are shown in red "+" sign. BS-MCL outperforms Raw-Bow with much fewer local descriptors, derived from the compact q-Codebook based on m-Codebooks, which consequently yields more compact query descriptor with BS-MCL.

7. REFERENCES

[1] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.

[2] D. G. Lowe, "Distinctive image features from scale scale invariant keypoints," in *IJCV*, 2004.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*, 2006.

[4] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive rep. for local image descriptors," in *CVPR*, 2004.

[5] V. Chandrasekhar, G. Takacs, and D. Chen et. al., "Chog: Compressed histogram of gradients a low bit-rate feature descriptor," in *CVPR*, 2009.

[6] V. Chandrasekhar, G. Takacs, and D. Chen et. al., "Transform coding of image feature descriptors," in *VCIP*, 2009.

[7] D. Chen, S. Tsai, and V. Ch et. al., "Tree histogram coding for mobile image matching," in *DCC*, 2009.

[8] D. Chen, S. Tsai, and V. Ch et. al., "Inverted index compression for scalable image matching," in *DCC*, 2010.

[9] R. Ji, L. Duan, and J. Chen et. al., "Learning compact visual descriptor for low bit rate mobile landmark search," in *IJCAI*, 2011.

[10] R. Ji, L. Duan, and J. Chen et. al., "Towards low bit rate mobile visual search with multiple channel coding," in *ACM Multimedia*, 2011.

[11] Robert M. Gray and David L. Neuhoff, "Quantization," *IEEE Trans. on Information Theory*, 1998.

[12] R. Raina et. al. H. Lee, A. Battle, "Efficient sparse coding algorithms," in *NIPS*, 2007.

[13] H. Hofling et. al. J. Friedman, T. Hastie, "Pathwise coordinate optimization," in *ANN APPL STAT*, 2007.