Int J Comput Vis
DOI 10.1007/s11263-011-0472-9

Location Discriminative Vocabulary Coding for Mobile Landmark Search

Rongrong Ji · Ling-Yu Duan · Jie Chen · Hongxun Yao · Junsong Yuan · Yong Rui · Wen Gao

Received: 13 October 2010 / Accepted: 1 June 2011

Abstract With the popularization of mobile devices, recent years have witnessed an emerging potential for mobile landmark search. In this scenario, the user experience heavily depends on the efficiency of query transmission over a wireless link. As sending a query photo is time consuming, recent works have proposed to extract compact visual descriptors directly on the mobile end towards low bit rate transmission. Typically, these descriptors are extracted based solely on the visual content of a query, and the location cues from the mobile end are rarely exploited. In this paper, we present a Location Discriminative Vocabulary Coding (LDVC) scheme, which achieves extremely low bit rate query transmission, discriminative landmark description, as well as scalable descriptor delivery in a unified framework. Our first contribution is a compact and location discriminative visual landmark descriptor, which is offline learnt in two-step: First, we adopt spectral clustering to segment a city map into distinct geographical regions, where both visual and geographical similarities are fused to optimize the partition of city-scale geo-tagged photos. Second, we propose to learn LDVC in each region with two schemes: (1) a Ranking Sensitive PCA and (2) a Ranking Sensitive Vocabulary Boosting. Both schemes embed location cues to learn a compact descriptor, which minimizes the retrieval ranking loss by replacing the original high-dimensional signatures. Our second contribution is a location aware online vocabulary adaption: We store a single vocabulary in the mobile end, which is efficiently adapted for a region specific LDVC coding once a mobile device enters a given region. The learnt LDVC landmark descriptor is extremely compact (typically 10–50 bits with arithmetical coding) and performs superior over state-of-the-art descriptors. We implemented the framework in a real-world mobile landmark search prototype, which is validated in a million-scale landmark database covering typical areas e.g. Beijing, New York City, Lhasa, Singapore, and Florence.

Keywords Mobile landmark search · Compact visual descriptor · Vocabulary compression · Two-way coding · Descriptor adaption · System applications

1 Introduction

With the popularization of camera embedded mobile devices, mobile landmark and location search have received a wide range of attentions from both academia and industry. For instance, mobile location recognition (Zhang and Kosecka 2006; Lee et al. 2008; Shao et al. 2003; Philbin et al. 2007; Crandall et al. 2009; Irschara et al. 2009), mobile landmark identification, online photographing recommendation (Kennedy et al. 2007; Hays and Efros 2008; Ji et al. 2009b; Zheng et al. 2009; Li et al. 2008), and content based advertising (Liu et al. 2009).

To this end, most existing mobile landmark search systems follow a client-server architecture. The remote server

R. Ji · L.-Y. Duan (✉) · J. Chen · W. Gao
Institute of Digital Media, Peking University, Beijing, China
e-mail: lingyu@pku.edu.cn

R. Ji · H. Yao
Visual Intelligence Laboratory, Harbin Institute of Technology, Harbin, China

J. Yuan
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Y. Rui
Microsoft China Research and Development Group, Beijing, China

Published online: 27 July 2011

maintains a landmark photo database, where each photo is bound with a location label e.g. GPS. A scalable near-duplicate visual search system is typically deployed based on a Bag-of-Words (BoW) model with inverted indexing structure (Nister and Stewenius 2006; Schindler and Brown 2007; Ji et al. 2008, 2009a; Irschara et al. 2009). In online search, the mobile user takes a query photo, which is transmitted to the remote server to identify its corresponding landmark through visual matching.[1] Subsequently, the server returns the search results including the geographical location, photograph viewpoints, tourism recommendation, or other value added information.

In many scenarios, the query photo is delivered over a bandwidth constrained wireless link. The user experience heavily depends on how much data to transmit. It is easy to imagine that sending the entire photo is time consuming and is not necessary indeed. The transmission overload turns out to be a bottleneck in most existing mobile visual search applications, especially for video rate reality augmentation.

1.1 State-of-The-Art Mobile Landmark Search Framework

The ever growing computational power motivates the research efforts to extract visual descriptors directly on a mobile device (Chen et al. 2009, 2010; Chandrasekhar et al. 2009a, 2009b; Makar et al. 2009). Instead of sending an entire photo, sending such descriptors are compact enough to enable the low bit rate search. Comparing with the previous works in low dimensional local descriptors such as PCA-SIFT (Ke and Sukthankar 2004), GLOH (Mikolajczyk and Schmid 2005), SURF (Bay et al. 2006), and MSR descriptors (Hua et al. 2007), works in (Chen et al. 2009, 2010; Chandrasekhar et al. 2009a, 2009b; Makar et al. 2009) target at intensive compactness as well as efficient extraction in a standard mobile end. They are expected to work well in mobile visual search scenarios.

Towards compact local visual descriptors, Chandrasekhar et al. proposed a Compressed Histogram of Gradient (CHoG) (Chandrasekhar et al. 2009a), which are further compressed by both Huffman Tree and Gagie Tree to reduce the size of each descriptor to approximate 50 bits. Works in Chandrasekhar et al. (2009b) employ Karhunen-Loeve transform to compress the SIFT descriptor, producing approximate 2 bits per SIFT dimension (128 dimensions in total). Tsai et al. (2010) proposed to transmit the spatial layouts of interest points to improve the precision of feature matching. Comparing with sending an entire query photo, sending above compact descriptors are much more efficient (Chandrasekhar et al. 2010). For instance, CHoG typically outputs only 50 bits per local feature. When 1,000 interest points are extracted per query (following the popular detector setting (Mikolajczyk et al. 2006)), the data amount to transmit is only 8 KB, much less than the entire query photo (typically over 20 KB with JPEG compression).

Chen et al. (2009) stepped forward to send the bag-of-features histogram (Chen et al. 2009, 2010) instead, which encodes the position difference of non-zero bins to yield approximate 2 KB per query photo using a one million vocabulary. It largely outperforms directly sending the compact local descriptors (more than 5 KB in reported works). Their successive work in Chen et al. (2010) further compressed the inverted indexing structure of visual vocabulary (Nister and Stewenius 2006) with arithmetic coding to reduce the memory and storage cost to maintain the scalable visual search system in server(s).

1.2 The Challenges

To the best of our knowledge, sending compressed bag-of-features still needs 2–4 KB descriptors per query in the state-of-the-arts (Chen et al. 2009, 2010). In real world 3G wireless environment, the unstable or limited upstream link would often delay the subsequent similarity ranking with the state-of-the-art compactness. The user experience is then degenerated. However, maintaining sufficient discriminability as well as desirable descriptor compactness is a sort of trade off, demanding elegant descriptor design to optimize both factors. Below we summarize four key challenges in the state-of-the-arts.

Location Insensitive Compression The existing compact descriptors are designed based on the visual statistics solely, the rich and cheaply available location cues (such as GPS or geographical tags) are left unexploited. More importantly, such location contexts have been widely available from either mobile devices or many landmark photo collections (Zheng et al. 2009; Schindler and Brown 2007; Irschara et al. 2009; Crandall et al. 2009; Ji et al. 2008, 2009a). We are inspired to develop more compact visual landmark descriptors that are location sensitive. That is, the landmark descriptor extraction should be "*location sensitive*", taking into account where the query happens.

Unscalable Landmark Description We argue that the compact descriptor could be scalable in length with respect to different regions, each of which maintains its own sufficient

[1] Vision based localization is a more generic choice comparing with solely GPS based localization: First, GPS signal is missing or noisy in many important locations, such as indoors, occluded skies, and the areas of dense buildings, which can only be identified by visual matching; Second, the pose and direction cannot be exactly identified by using GPS only. Third, the object of user interest cannot be located by GPS. Clearly, GPS only is not suitable for pervasive mobile search applications like reality argumentation or shopping recommendation. In addition, instead of relying on GPS signal, we may combine other types of location related cues (i.e. base station ID) with the visual query to perform context aware landmark search.

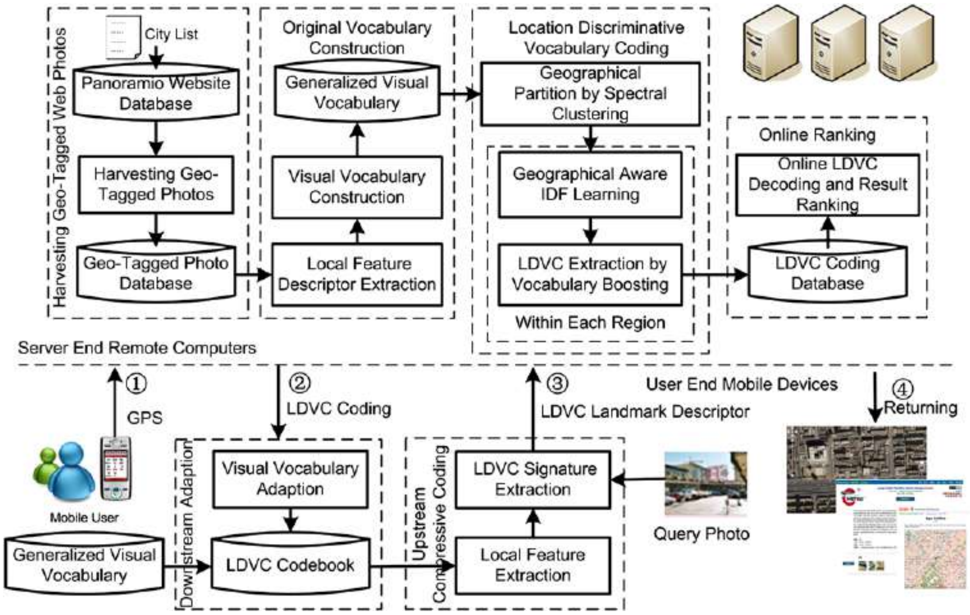

City List
Harvesting Geo-Tagged Web Photos
Panoramio Website Database
Harvesting Geo-Tagged Photos
Geo-Tagged Photo Database
Original Vocabulary Construction
Generalized Visual Vocabulary
Visual Vocabulary Construction
Local Feature Descriptor Extraction
Location Discriminative Vocabulary Coding
Geographical Partition by Spectral Clustering
Geographical Aware IDF Learning
LDVC Extraction by Vocabulary Boosting
Within Each Region
Online Ranking
Online LDVC Decoding and Result Ranking
LDVC Coding Database
Server End Remote Computers
① GPS
② LDVC Coding
③ LDVC Landmark Descriptor
User End Mobile Devices
④ Returning
Mobile User
Generalized Visual Vocabulary
Downstream Adaption
Visual Vocabulary Adaption
LDVC Codebook
Upstream Compressive Coding
LDVC Signature Extraction
Local Feature Extraction
Query Photo

Fig. 1 The proposed **L**ocation **D**iscriminative **V**ocabulary **C**oding (**LDVC**) framework for efficient and effective mobile landmark search in a bandwidth constrained wireless network environment

discriminability. The scalability depends on the visual complexity of landmark images at a given location. For instance, the descriptors in a location containing multiple landmarks with diverse appearances can be less compact.

Symmetric Vocabulary Maintaining The existing landmark search systems maintain a symmetric vocabulary between the remote server and the mobile end. This is unnecessary. As the visual statistics of different geographical locations are distinct, we may adaptively compress the vocabulary in the mobile end, with gains in both storage and computing. On the other hand, as maintaining a single vocabulary to search worldwide landmarks is unrealistic, existing works had to pre-load a vocabulary per city (Schindler and Brown 2007; Ji et al. 2008, 2009a), which is extremely time consuming in real world applications.

Coding Transmission Module The existing landmark search systems work in a straightforward pipeline, which directly extracts compact descriptors for upstream[2] query delivery. However, it is never constrained that only upstream transmission is allowed before the servers receive the visual query and perform ranking followed by resulting retrieval results. For instance, by leveraging the location of a mobile user, the remote sever may pre-deliver a compact downstream supervision to "*teach*" the mobile how to extract compact and discriminative descriptors.

In summary, we have two main concerns towards location aware mobile landmark search:

- An effective visual descriptor compression scheme: We aim to learn a location discriminative vocabulary coding that is discriminative in distinguishing visual content of different landmarks, extremely compact for wireless transmission, and scalable for representing and delivering queries subject to different visual complexity.
- A novel vocabulary maintaining and search framework: We aim to maintain a single vocabulary on a mobile end to search worldwide landmarks, which can be efficiently adapted with respect to different locations.

1.3 Our Contributions

In this paper, we present a location discriminative vocabulary coding framework as shown in Fig. 1, which enables efficient mobile landmark search even in a bandwidth constrained wireless link with two major contributions:

Our first contribution is an adaptive, asymmetric, and geographical aware vocabulary compression scheme, which works in the mobile end for location aware low bit rate landmark description. First, we present a visual aware spectral clustering to segment each city map into discrete geographical regions (locations) based on a large collection of geo-tagged photos in that city. Visual similarity is embedded to compensate the distortions in geographical tag (e.g. GPS) acquisition. Concretely, in each region, we introduce a geographical term weighting to evaluate location aware codeword discriminability. Then, we learn a **L**ocation **D**iscriminative **V**ocabulary **C**oding (**LDVC**) based on two proposed schemes: (1) a Ranking Sensitive PCA (*rs*PCA)

[2] In this paper, "upstream" denotes data delivery from a mobile to a remote server; "downstream" denotes data delivery from a remote server to a mobile end. Similarly, "upload" denotes a mobile end sends data to a remote server; "download" denotes a mobile end loads data from a remote server.

and (2) a Ranking Sensitive Vocabulary Boosting. The latter is a simplification to the former and both merit in the following aspects:

1. Specialized for each geographical region, which is downstream adapted to the mobile device to "teach" it how to extract the desirable descriptor within each region.
2. Extremely compact for both upstream and downstream wireless transmission, say only tens of bits per query. Such compactness is suitable for low latency search, especially in unstable or bandwidth-constraint 3G connections.
3. Sufficiently discriminative power to maintain the retrieval ranking precision comparable to that using high-dimensional vocabulary.
4. Scalable in its coding length, depending on the visual complexity in a region: For example, the region with many landmarks is supposed to yield less compact **LDVC** descriptor, and vice versa.

Our second contribution is a novel location aware landmark search framework. We break through the traditional upstream query pipeline: In online search, we first downstream deliver the **LDVC** set to adapt the vocabulary in the mobile device once a mobile user enters a given region. Then, once a landmark query occurs, the mobile device transforms the original Bag-of-Words (BoW) histogram into a compact **LDVC** descriptor, which is then upstream transmitted to the server with arithmetical coding. This scheme has achieved superior effectiveness (with the highest Mean Average Precision) and the lowest transmission cost (approximate 10–50 bits per query) comparing with the state-of-the-arts in Chen et al. (2009), Chandrasekhar et al. (2009a), Jegou et al. (2009, 2010a) to search landmarks in a million-scale landmark photo collection.

We review related work of landmark search and compact descriptors in Sect. 2. Our location discriminative vocabulary coding is introduced in Sect. 3. Then, Sect. 4 presents our online vocabulary adaption and systematic implementation in typical areas i.e. Beijing, New York City, Lhasa, Singapore, and Florence. We give quantitative experiments in Sect. 5, with comparisons to the state-of-the-art works in mobile visual search (Chen et al. 2009, 2009a) and compact image signatures (Jegou et al. 2010a, 2009).

2 Related Work

2.1 Landmark Search and Recognition

Near-Duplicate Landmark Matching Recently, scalable near-duplicate image retrieval (Nister and Stewenius 2006; Philbin et al. 2007) has been largely addressed by promising visual vocabulary models with inverted indexing e.g. K Means clustering (Sivic and Zisserman 2003), Vocabulary Tree (Nister and Stewenius 2006), and Approximate K-Means (Philbin et al. 2007) et al.

City-Scale Landmark/Location Search Towards city-scale landmark search and recognition, Schindler and Brown (2007) presented a location recognition system through geotagged video streams with multiple path search in the vocabulary tree. Eade and Drummond (2008) also adopted a vocabulary tree for real-time loop closing based on SIFT-like descriptors. Our previous works in Ji et al. (2009b) proposed a density-based metric learning to optimize the hierarchical structure of vocabulary tree (Nister and Stewenius 2006) for street view location recognition. Yeh et al. (2007) further adopted a hybrid color histogram to compensate the feature based ranking in mobile based location recognition applications. Cristani et al. (2008) learnt a global-to-local image matching for location recognition. And their consecutive work in Crandall et al. (2009) identified landmark buildings based on image data, metadata, and other photos taken within a consecutive 15-minute window. In addition, Irschara et al. (2009) further leverage structure-from-motion (SFM) to build 3D scene models for street views, combined with vocabulary tree for simultaneously scene modeling and location recognition. Xiao et al. (2008) proposed to combine bag-of-features with simultaneous localization and mapping (SLAM) to further improve the recognition precision. Incrementally vocabulary indexing is also explored in Ji et al. (2008) to maintain a landmark search system in a time varying database.

Worldwide Landmark Search Towards worldwide landmark search and recognition, the IM2GPS system (Hays and Efros 2008) inferred possible location distributions of a given query by visual matching in a worldwide, geo-tagged landmark dataset. As a consecutive work, Kalogerakis et al. (2009) further demonstrated how to combine single image matching with sequential data to improve matching accuracy. Zheng et al. (2009) developed a worldwide landmark recognition system, which used a predefined landmark list to query online image search engines to selected candidate images, followed by re-clustering and pruning to locate the final landmark location.

2.2 Mobile Visual Search with Compact Visual Descriptors

Even with the increasing wireless bandwidth, compact visual descriptors are desirable:

Firstly, it remains a long way to provide a stable and high-speed (3G) wireless coverage everywhere, especially for those touristic landmarks that are often far away from urban areas or in developing countries, e.g., Lhasa, Tibet in our experiments. So it is unrealistic to guarantee the bandwidth is good enough to reliably and fast send a query photo. In particular, the recently established MPEG Ad Hoc Group **CDVS** (reference number: N11688) is bringing together the academia and industry practitioners to explore the next MPEG standard of **C**ompact **D**escriptor for **V**isual **S**earch.

Secondly, from the server perspective, the network capability of receiving a batch of entire photo queries is by no doubt limited for a more powerful cloud platform that may handle intensive search at the server end. From the industry practice, a clear fact is that receiving multiple query photos is much more challenging than receiving texts in the state-of-the-art search engines. More importantly, with compact upstream queries, more bandwidth can be saved up to downstream return the actually valuable searched information (in rich forms of text, images and video). That is one of the reasons why many Internet service providers often set a smaller uplink bandwidth to save bandwidth for fast browsing.

Finally, sending large amount of data via 3G wireless definitely causes serious battery energy consumption. Empirical evidence shows that compressing the query photo into a compact signature and sending the signature through the mobile is much more power saving.

In summary, the promising research efforts in compact visual descriptors (as reviewed in Sect. 1.1) are bringing great benefits in lightening the battery consumption, the cost of bandwidth and memory, which undoubtedly contribute to efficient and effective visual query delivery in mobile visual search, especially in the scenarios of video rate reality augmentation.

It is worth mentioning that, building a vocabulary for each city is impractical for transmitting selected codewords at low bit rate, which requires the maintenance of those vocabularies in the mobile end. As proven, to ensure desirable search precision, the vocabulary size should be large enough (e.g. at a million scale) (Nister and Stewenius 2006; Philbin et al. 2007; Schindler and Brown 2007). But even for a very small vocabulary (approximate 1,000 words), to deliver all codewords through a bandwidth constrained wireless link is still burdensome. However, to maintain numerous codebooks (each for a city) in the mobile is definitely unacceptable, due to its limited storage and constrained memory.

2.3 Compact Image Signature

Beyond the context of mobile visual search, compact image signatures are recently investigated in Yeo et al. (2008), Weiss et al. (2009), Jegou et al. (2010a, 2010b). For instance, Jegou et al. proposed a product quantization scheme (Jegou et al. 2010b) to learn a compact image descriptor that approximates the square distance of original Bag-of-Words histograms. The same authors also proposed a miniBOF feature (Jegou et al. 2009) by packing the bag-of-features. Their recent work in Jegou et al. (2010a) further aggregated local descriptors with PCA and locality sensitive hashing to produce a compact descriptor of approximate 32 bits in length. Weiss et al. (2009) used spectral hashing to compress GIST descriptor (Torralba et al. 2008) into tens of bits. Wang et al. (2010) proposed a locality-constrained linear coding (LLC) scheme over the Bag-of-Words histogram to improve the spatial pyramid matching. Finally, in multi-view coding, Yeo et al. (2008) proposed a rate-efficient correspondence learning scheme to randomly project descriptors to build a minHashing code.

2.4 Image Compression

Image compression aims to minimize the visual distortion between the original image pixels and the recovered ones from the compressed signals. Many methods such as Run-Length Coding, Predictive Coding, Entropy Encoding, DCT Coding, and Dictionary Learning are well explored in the literature. In contrast, our vocabulary coding aims to obtain a compact image signature for searching near-duplicate photos rather than lossy or lossless image recovery. In other words, we focus on maximizing the descriptor discriminability with minimal coding cost, rather than the perceptual consistence in recovering the original image content. In the subsequent sections, whereas rate distortion in image compression is employed to evaluate the coding gains, we emphasize the distortion of search precision with reduced rates. In our experiments (Sect. 5), the term rate distortion ("distortion" is in terms of ranking precision) is applied to study the trade off between descriptor compactness and ranking precision.

3 Location Discriminative Vocabulary Coding

3.1 Scalable Vocabulary Tree Search

Towards scalable near-duplicate visual search, the Scalable Vocabulary Tree (SVT) model (Nister and Stewenius 2006) is well exploited in the state-of-the-art works (Chen et al. 2009, 2010; Schindler and Brown 2007; Irschara et al. 2009). SVT uses hierarchical k means to quantize local descriptors into discrete codewords. An H-depth B-branch SVT produces $M = B^H$ codewords. And most scalable search systems typically have $H = 6$ and $B = 10$ (Nister and Stewenius 2006). Given a query photo I_q with J local descriptors $\mathbf{S}_q = [S_1^q, \dots, S_J^q]$, each descriptor is traversed in the SVT hierarchy to find the nearest codeword, which quantizes $\mathbf{S}_q$ into a Bag-of-Words (BoW) histogram $\mathbf{V}_q = [V_1^q, \dots, V_M^q]$.

For an N-photo database, an optimized ranking using $\mathbf{S}_q$ is the one that minimizes the following ranking loss:

$$Loss_{Rank} = \sum_{x=1}^{N} R(x) D_{descriptors}(I_x, I_q) \quad (1)$$

where $R(x) = \exp(-rank(x))$ is the ranking position weight of I_x with respect to I_q, such that a higher rank corresponds to a larger weight. $R(x)$ puts a constraint that a photo more similar to I_q should be ranked higher. $D_{descriptors}(I_x, I_q)$ stands for the sum of L2 distances for pairwise descriptor matching between $\mathbf{S}_q$ and $\mathbf{S}_x$[3]:

$$D_{descriptors}(I_x, I_q) = \sum_{j=1}^{J} \left(\|S^x_{i'}, S^q_j\|_2 \quad \text{s.t.} \quad i' = \arg\min_i \|S^x_i, S^q_j\|_2 \right) \tag{2}$$

Minimizing (1) with respect to (2) cannot scale up due to its linear complexity to the image volume N. SVT (Nister and Stewenius 2006) addresses the scalability by inverted indexing (Witten et al. 1999) each I_x to codeword V_i that involves descriptor(s) from $\mathbf{S}_x$. Subsequently, SVT only compares those images indexed by each non-zero codeword V^q_i for the given query I_q:

$$\begin{aligned} D_{descriptors}(I_x, I_q) &\approx \sum_{i=1}^{M} \|Count(V^x_i), Count(V^q_i)\|_2 \\ \text{s.t.} \quad & Count(V^x_i) = |S^x_j | Q(S^x_j) = V_i| \\ & Count(V^q_i) = |S^q_j | Q(S^q_j) = V_i| \neq 0 \end{aligned} \tag{3}$$

where $Q(S_j) = V_i$ means the SVT quantizes descriptor S_j into codeword V_i; $Count(V^x_i)$ denotes the number of local descriptors falling into V_i of photo I_x. Equation (3) approximates the optimal matching between I_q and I_x by counting their concurrent codewords. Using L2 distance, the inverted indexing based search is identical to the L2 similarity ranking between two BoW histograms. Therefore, the ranking loss can be approximated using the following form:

$$Loss_{Rank} \approx \sum_{x=1}^{N} R(I_x) \sum_{i=1}^{M} \|Count(V^x_i), Count(V^q_i)\|_2 \tag{4}$$

Many existing works (Nister and Stewenius 2006; Chen et al. 2009, 2010; Schindler and Brown 2007; Irschara et al. 2009) also involve Term Frequency and Inverted Document Frequency (TF-IDF) weighting $\mathbf{W}_x$ with Cosine distance as:

$$Loss_{Rank} = \sum_{x=1}^{N} R(I_x) \mathbf{W}_x \|\mathbf{V}_x, \mathbf{V}_q\|_{Cosine} \tag{5}$$

where TF-IDF weighting $\mathbf{W}_x$ is calculated similar to its original form in Salton and Buckley (1988) as:

$$\mathbf{W}_x = \left[\frac{n^x_1}{n^x} \log\left(\frac{N}{N_{V_1}}\right), \ldots, \frac{n^x_i}{n^x} \log\left(\frac{N}{N_{V_i}}\right), \ldots, \frac{n^x_M}{n^x} \log\left(\frac{N}{N_{V_M}}\right) \right] \tag{6}$$

n^x denotes the number of local descriptors in I_x; $n^x_{V_i}$ denotes the number of local descriptors in I_x quantized into V_i; N denotes the total number of images in the database; N_{V_i} denotes the number of images containing V_i; $\frac{n^x_i}{n^x}$ is the Term Frequency (TF) (Salton and Buckley 1988) of V_i in I_x; and $\log(\frac{N}{N_{V_i}})$ is the Inverted Document Frequency (IDF) (Salton and Buckley 1988) of V_i in the entire dataset.

3.2 Compact Descriptor Learning Formulation

We aim to learn a coding matrix $\mathbf{M}_{M \times K}$ from the original vocabulary $\mathbf{V} \in \mathbb{R}_M$ to a compact codebook $\mathbf{C} \in \mathbb{R}_K$. This matrix transforms an original BoW histogram $\mathbf{V}_x$ of I_x to a much more compact descriptor $\mathbf{C}_x$. In other words, $\mathbf{M}^T$ is the coder and $\mathbf{M}$ is the decoder as in Fig. 2:

$$\mathbf{C}_x = f(\mathbf{V}_x) = \mathbf{M}^T \mathbf{V}_x \tag{7}$$

We formulate the following cost to seek the trade-off between the descriptor compactness and the search precision:

$$Cost_{Compression} = |\mathbf{M}^T \mathbf{V}| + \alpha Loss_{Rank} \tag{8}$$

$|\cdot|$ denotes the size of $\mathbf{M}^T\mathbf{V}$ (equal to the new codebook $\mathbf{C}$). We aim to minimize $Cost_{Compression}$ in terms of $\mathbf{M}$, where a low dimensional transform $\mathbf{M}^T\mathbf{V}$ is favored. The ranking loss $Loss_{Rank}$ in (8) is formulated as:

$$Loss_{Rank} = \sum_{x=1}^{N} R(I_x) \mathbf{W}_x \|\mathbf{V}_x, \mathbf{MC}_q\|_{Cosine} \tag{9}$$

where $\mathbf{C}_q$ is the compact descriptor extracted from the query I_q. $\mathbf{MC}_q$ is the decoded BoW histogram in the server. Seeking an optimal $\mathbf{C}$ in a large scale landmark dataset is infeasible due to the extreme difficulty in modeling all ranking constraints for every database photo into (9), for which we resort to a region-specific sampling scheme latter.

Location Discriminative Compression To enable efficient optimization in a million scale database, we learn $\mathbf{M}$ within each geographical region as a *local optimal* compression function. Subsequently, once the mobile end enters a given region, its vocabulary is downstream adapted using a region specific $\mathbf{M}_{Region}$. (The geographical partition is introduced in Sect. 4.1.)

[3] A typical matching allows only one descriptor in $\mathbf{S}_x$ to be matched against one descriptor in $\mathbf{S}_q$. But SVT model counts the quantization concurrence between $\mathbf{S}_x$ and $\mathbf{S}_q$. So one-to-many matching is allowed in (2).

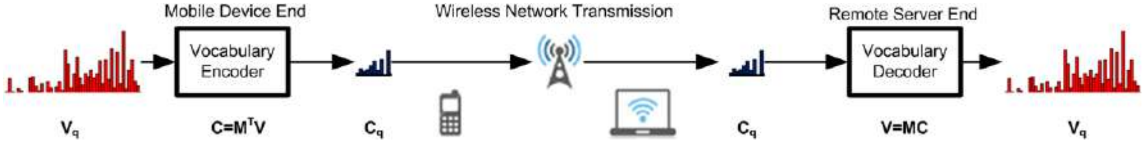


Fig. 2 The proposed vocabulary coding and wireless upstream query transmission pipeline

More formally, given a geographical region containing images $[I_1, \ldots, I_N]$, we try to learn the region specific $\mathbf{M}_{Region}$. This can be solved more efficiently and effectively as:

$$Cost^{Region}_{Compression} = |\mathbf{M}^T_{Region}\mathbf{V}| + \alpha Loss^{Region}_{Rank} \tag{10}$$

$Loss^{Region}_{Rank}$ denotes the sum of ranking loss only in $[I_1, \ldots, I_N]$ instead of in the entire N-photo database $(N \gg n)$ as:

$$Loss^{Region}_{Rank} = \sum_{x=1}^{n} R(I_x)\mathbf{W}_x \|\mathbf{V}_x, \mathbf{M}_{Region}\mathbf{C}_q\|_{Cosine} \tag{11}$$

We propose two learning schemes to learn the optimal **C** based on the renewed loss in (11): Our first scheme is a Ranking Sensitive Principle Component Analysis in Sect. 3.4 to learn $\mathbf{M}_{Region}$ that maximally preserves the retrieval ranking orders of the original **V**. Our second scheme is a less precise but more effective solution: Instead of the nonlinear transformation of $\mathbf{M}_{Region}$, we propose to boost a compact codeword subset, referred to as "Ranking Sensitive Vocabulary Boosting" in Sect. 3.5, which is also discriminative to nicely maintain the ranking precision of **V**.

3.3 Building Training Set via Conjunctive Ranking

Given a geographical region containing n landmark photos $[I_1, \ldots, I_n]$, the first step of both schemes is to sample a subset of photos $[I'_1, I'_2, \ldots, I'_{n_{sample}}]$ to conduct n_{sample} times conjunctive query,[4] which outputs the following ranking lists:

$$\begin{aligned} &Query(I'_1) = [A^1_1, A^1_2, \ldots, A^1_R] \\ &\vdots \\ &Query(I'_{n_{sample}}) = [A^{n_{sample}}_1, A^{n_{sample}}_2, \ldots, A^{n_{sample}}_R] \end{aligned} \tag{12}$$

where A^j_i is the ith returning of the jth query. $[A^j_1, A^j_2, \ldots, A^j_R]$ are R top ranked images based on the original BoW histogram given the jth query ($j \in [1, n_{sample}]$).

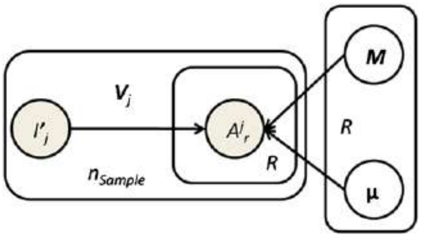


Fig. 3 The probabilistic graph model of Ranking Sensitive PCA (*rs*PCA) to learn LDVC descriptor. The LDVC learning is treated as a generative and supervised Probabilistic PCA, which is guided by K-dimensional latent variables **z** with $M \times K$ transformation $\mathbf{M}_{M \times K}$ and mean $\boldsymbol{\mu}$. What are observable are the n_{Sample} conjunctive rankings with the BoW histograms $\mathbf{V}_j$, $j \in [1, n_{Sample}]$, corresponding to a list of R returning results $\{A^j_r\}$, $r \in [1, R]$

We aim to maximally preserve the ranking orders of above conjunctive queries by compact descriptor **C** instead of **V**. To this end, ranking lists of queries $[I_1, \ldots, I_n]$ are treated as training data to learn **M** in (9), (10), and (11).

3.4 Learning LDVC by Ranking Sensitive PCA

Our first solution comes from learning the principle components from **V** to form the LDVC descriptor in each region. Different from the original PCA, we resort to its probabilistic version (Tipping and Bishop 1997). We propose a *Ranking Sensitive* PCA (*rs*PCA) to embed the ranking discriminability into the principle component extraction. Figure 3 shows the graphical representation of the proposed *rs*PCA.

Given a set of BoW histograms $\{\mathbf{V}_i\} \in \mathbb{R}_M$ where $i \in [1, n]$, PCA learns an optimal linear projection set to map $\{\mathbf{V}_i\}$ into $\{\mathbf{C}_i\} \in \mathbb{R}_K$, where $\mathbb{R}_K$ is a low-dimensional space of principal components. The mapping minimizes the average projection cost defined by the mean square distances between $\{\mathbf{V}_i\}$ and $\{\mathbf{C}_i\}$. To this end, an $M \times M$ covariance matrix **S** is first defined:

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{V}_i - \overline{\mathbf{V}})(\mathbf{V}_i - \overline{\mathbf{V}})^T \tag{13}$$

where $\overline{\mathbf{V}}$ denotes the mean vector of $\{\mathbf{V}_i\}$. Subsequently, PCA extracts the top K eigenvectors $\mathbf{c}_1, \ldots, \mathbf{c}_k$ from **S** to define a K-dimensional linear projection **C**, such that $\mathbf{S}\mathbf{c}_i = \lambda_i \mathbf{c}_i$, where the ith dimension corresponds to the ith largest eigenvalue λ_i with eigenvector $\mathbf{c}_i$.

[4] The term "conjunctive" denotes that queries are randomly selected from each region to simulate the possible queries posed by a mobile user. The "conjunctive" query is used to train our LDVC descriptor in each region, which is different from the queries for evaluation. One advantage is that the conjunctive query does not need any user labeling.

We do not apply PCA directly in its original form due to: (1) Computing the full eigenvector decomposition for an $M \times M$ matrix $\mathbf{S}$ needs $O(M^3)$, which is extremely time consuming for a large vocabulary. (2) It is hard to embed supervised information (such as the ranking preservation capability) into the eigenvector decomposition. Instead, we learn $\mathbf{C}$ by an iterative probabilistic optimization.

p*PCA for Vocabulary Coding* The *probabilistic* PCA (pPCA) addresses the computational inefficiency of PCA from the perspective of probabilistic latent Gaussian distribution (Tipping and Bishop 1997), which adopts Expectation Maximization (EM) to learn an optimal $\mathbf{M}_{M\times K}$ with time complexity $O(M)$ rather than $O(M^3)$.

Following the principle of Tipping and Bishop (1997), we first give an explicit latent variable $\mathbf{z}$ corresponding to the principal component subspace. Then, we define a zero-mean unit-covariance Gaussian prior distribution $p(\mathbf{z})$ and a conditional Gaussian distribution $p(\mathbf{V}|\mathbf{z})$ over $\mathbf{z}$ as:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, I) = \frac{1}{2\pi}\exp\left\{-\frac{\mathbf{z}^T\mathbf{z}}{2}\right\} \tag{14}$$

$$\begin{aligned} p(\mathbf{V}|\mathbf{z}) &= \mathcal{N}(\mathbf{V}|\mathbf{Mz}+\boldsymbol{\mu}, \sigma^2 I) \\ &= \frac{1}{2\pi\sigma^2}\exp\left\{-\frac{\|\mathbf{V}-\mathbf{Mz}-\boldsymbol{\mu}\|_2}{2\sigma^2}\right\} \end{aligned} \tag{15}$$

Therefore, with an assumption of V_i being independent between each other, the original signature $\mathbf{V}$ is a generative output of this linear projection with a M-dimensional zero-mean Gaussian noise $\boldsymbol{\epsilon}$ and covariance $\sigma^2 I$:

$$\mathbf{V} = \mathbf{Mz} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{16}$$

We aim to determine $\mathbf{M}$, $\boldsymbol{\mu}$, and σ^2 using maximum likelihood estimation. To this end, we give a marginal distribution $p(\mathbf{V})$ from $\mathbf{z}$ as follows:

$$p(\mathbf{V}) = \int(\mathbf{V}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{17}$$

which is again a Gaussian distribution with:

$$\mathbb{E}[\mathbf{V}] = \mathbb{E}[\mathbf{Mz}+\boldsymbol{\mu}+\boldsymbol{\epsilon}] = \boldsymbol{\mu} \tag{18}$$

$$\begin{aligned} Cov[\mathbf{V}] &= \mathbb{E}[(\mathbf{Mz}+\boldsymbol{\epsilon})(\mathbf{Mz}+\boldsymbol{\epsilon})^T] \\ &= \mathbb{E}[\mathbf{Mzz}^T\mathbf{M}^T] + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \mathbf{MM}^T + \sigma^2 I \end{aligned} \tag{19}$$

Therefore, each original histogram $\mathbf{V}_i$ corresponds to a latent variable $\mathbf{z}_i$. And an EM algorithm can be adopted to find the maximum likelihood of $\mathbf{M}$, $\boldsymbol{\mu}$, and σ^2, with a discrete log likelihood estimation of (17) as:

$$\ln p(\mathbf{V}, \mathbf{z}|\mathbf{M}, \boldsymbol{\mu}, \sigma^2) = \sum_{i=1}^{n}\{p(\mathbf{V}_i|\mathbf{z}_i) + p(\mathbf{z}_i)\} \tag{20}$$

where $\boldsymbol{\mu}$ is equal to $\overline{\mathbf{V}}$.

In the *Expectation* step, using (14) and (15), we calculate the expectation of (20) by expanding the log likelihood into:

$$\begin{aligned} &\mathbb{E}[\ln p(\mathbf{V}, \mathbf{z}|\mathbf{M}, \boldsymbol{\mu}, \sigma^2)] \\ &= -\sum_{i=1}^{n}\left\{\frac{M}{2}\ln(2\pi\sigma^2) + \frac{1}{2}\mathrm{Tr}(\mathbb{E}[\mathbf{z}_i\mathbf{z}_i{}^T])\right. \\ &\quad + \frac{1}{2\sigma^2}\|\mathbf{V}_i - \overline{\mathbf{V}}\|_2 - \frac{1}{\sigma^2}\mathbb{E}[\mathbf{z}_i]^T\mathbf{M}^T(\mathbf{V}_i - \overline{\mathbf{V}}) \\ &\quad \left. + \frac{K}{2}\ln(2\pi) + \frac{1}{2\sigma^2}\mathrm{Tr}(\mathbb{E}[\mathbf{z}_i\mathbf{z}_i{}^T]\mathbf{M}^T\mathbf{M})\right\} \end{aligned} \tag{21}$$

$\mathbb{E}[\mathbf{z}_i\mathbf{z}_i{}^T] = \sigma^2\mathbf{W} + \mathbb{E}[\mathbf{z}_i]\mathbb{E}[\mathbf{z}_i]^T$, and $\mathbf{W}$ is learnt based on $\mathbf{W} = \mathbf{M}^T\mathbf{M} + \sigma^2 I$. We then only need to estimate $\mathbb{E}[\mathbf{z}_n]$ as:

$$\mathbb{E}[\mathbf{z}_i] = \mathbf{W}^{-1}\mathbf{M}^T(\mathbf{V}_i - \overline{\mathbf{V}}) \tag{22}$$

In the *Maximization* step, we maximize the estimation in (21) with respect to renewed both $\mathbf{M}$ and σ^2 as:

$$\mathbf{M}_{new} = \left[\sum_{i=1}^{n}(\mathbf{V}_i - \overline{\mathbf{V}})\mathbb{E}[\mathbf{z}_i]^T\right]\left[\mathbb{E}[\mathbf{z}_i\mathbf{z}_i{}^T]\right]^{-1} \tag{23}$$

$$\begin{aligned} \sigma^2_{new} = \frac{1}{MK}&\sum_{i=1}^{n}\{\|\mathbf{V}_i - \overline{\mathbf{V}}\|_2 - 2\mathbb{E}[\mathbf{z}_i]^T\mathbf{M}^T_{new}(\mathbf{V}_i - \overline{\mathbf{V}}) \\ &+ \mathrm{Tr}(\mathbb{E}[\mathbf{z}_i\mathbf{z}_i{}^T]\mathbf{M}_{new}\mathbf{M}^T_{new})\} \end{aligned} \tag{24}$$

Using the above EM iteration, pPCA improves the PCA computational complexity from $O(M^3)$ to $O(n \times M \times K)$, which is much more suitable for scalable learning.

*Ranking Sensitive PCA (*rs*PCA)* We propose a novel *Ranking Sensitive* PCA (*rs*PCA) to incorporate the ranking preservation supervision into PCA learning. *rs*PCA embeds the conjunctive rankings into the estimation of $p(\mathbf{z})$ as:

$$\mathbb{E}[\mathbf{z}_i] = \mathbf{W}^{-1}\mathbf{M}^T(\mathbf{V}_i - \overline{\mathbf{V}}) + Loss^i_{Rank} \tag{25}$$

where $Loss^i_{Rank}$ estimates whether the learnt $\mathbf{z}_i$ can preserve the ranking orders of its original BoW $\mathbf{V}_i$:

$$Loss^i_{Rank} = \sum_{r=1}^{R} R(A^i_r)\|M^T\mathbf{z}_i, \mathbf{V}_{A^i_r}\|_2 \tag{26}$$

$R(A^i_r)$ is the current ranking position of the original rth returning result for the ith conjunctive query $\mathbf{V}_i$ (see (27)).[5] By embedding the ranking loss in (26), we replace (22) with (25) in the EM estimation.

[5] To unify the learning labels, we set $Loss^i_{Rank} = 0$ for any $\mathbf{V}_i$ outside the conjunctive query set.

3.5 Learning LDVC by Ranking Sensitive Boosting

We further propose a Ranking Sensitive Vocabulary Boosting to improve the efficiency of LDVC learning, which approximates the nonlinear coding matrix $\mathbf{M}_{Region}$ learnt by *rs*PCA. Our boosting treats vocabulary coding as an AdaBoost (Freund and Schapire 1994) based codeword selection. The weak learner is each single codeword, and the learning is to minimize the ranking loss with more compact descriptors. This coding scheme can be also interpreted as a linear and greedy dimension reduction.

More formally, we define a unified error weighting vector $[w_1, \ldots, w_{n_{sample}}]$ to measure the ranking consistency loss for n_{sample} conjunctive rankings. To unify mathematical formulations, in vocabulary boosting, $\mathbf{M}_{Region}\mathbf{M}_{Region}^{T}$ is a diagonal matrix. Suppose at the tth iteration, we got the current $\mathbf{M}_{Region}^{t-1}$ with $(t-1)$ non-zero diagonal elements to indicate the selection of $(t-1)$ codewords. To select the tth codeword, we first estimate the ranking preservation of $\mathbf{M}_{Region}^{t-1}$:

$$Loss(I'_i) = w_i^{t-1}\sum_{r=1}^{R} R(A_r^i)\mathbf{W}_{A_r^i}\|\mathbf{M}_{Region}^{t-1}\mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i}\|_{Cosine} \tag{27}$$

$R(A_r^i)$ denotes the current ranking of the original rth returning of query I'_i; w_i^{t-1} is the $(t-1)$th error weighting of query I'_i to measure its ranking loss. Subsequently, we have:

$$\begin{aligned} Loss_{Rank}^{Region} &= \sum_{i=1}^{n_{sample}} Loss(I'_i) \\ &= \sum_{i=1}^{n_{sample}} w_i^{t-1}\sum_{r=1}^{R} R(A_r^i)\mathbf{W}_{A_r^i}\|\mathbf{M}_{Region}^{t-1} \\ &\quad \times \mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i}\|_{Cosine} \end{aligned} \tag{28}$$

for which the best new codeword C_t is selected as the one that minimizes the following loss:

$$\begin{aligned} C_t = \arg\min_j \sum_{i=1}^{n_{sample}} w_i^{t-1}\sum_{r=1}^{R} Rank(A_r^i)\mathbf{W}_{A_r^i}\|[\mathbf{M}_{Region}^{t-1} \\ + [0, \ldots, pos(j), \ldots, 0]_M [0, \ldots, pos(t), \ldots, 0]_K^T] \\ \times \mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i}\| \end{aligned} \tag{29}$$

$[0, \ldots, pos(j), \ldots, 0]_M$ is a $M \times 1$ selection vector to select the jth column into the linear projection; $[0, \ldots, pos(t), \ldots, 0]_K$ is a $K \times 1$ position vector to map this column into the selected word C_t. Subsequently, we update the error weighting of each w_i^{t-1}:

$$\begin{aligned} w_i^t = \sum_{r=1}^{R} R(A_r^i)\mathbf{W}_{A_r^i}\|[\mathbf{M}_{Region}^{t-1} \\ + [0, \ldots, pos(j), \ldots, 0]_M [0, \ldots, pos(t), \ldots, 0]_K^T] \\ \times \mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i}\| \end{aligned} \tag{30}$$

which updates the contribution of different conjunctive queries in selecting the next codeword. Also, the $\mathbf{M}_{Region}$ at the tth round is updated as follows:

$$\begin{aligned} \mathbf{M}_{Region}^{t} = \mathbf{M}_{Region}^{t-1} \\ + [0, \ldots, pos(j), \ldots, 0]_M [0, \ldots, pos(t), \ldots, 0]_K^T \end{aligned} \tag{31}$$

The codeword boosting is finalized when $\sum_{i=1}^{n_{sample}} w_i^t \leq \tau$, which results in a scalable LDVC coding length that may adapt with the visual complexity in the current region: For regions that contain complicated landmarks, our LDVC coding would be less compact to maintain its discriminability; for regions that contain visually simple landmarks, the LDVC would be more compact towards low bit rate wireless transmission. Algorithm 1 summarizes our Boosting based LDVC coding.

Algorithm 1: Ranking Sensitive Vocabulary Boosting for LDVC Construction in Each Region

```
Input: BoW histograms $\mathbf{V} = \{\mathbf{V}_i\}_{i=1}^{n}$; conjunctive rankings $\{Query(I'_r)\}_{r=1}^{R}$; boosting threshold $\tau$; error weighting vector $[w_1, \ldots, w_{n_{sample}}]$; boosting iteration $t = 0$.
Pre-Computing: Calculate $Loss_{Rank}^{Region}$ using (28);
while $\{\sum_{i=1}^{n_{sample}} w_i^t \leq \tau\}$ do
    Loss Estimation: Calculate $Loss_{Rank}^{Region}$ using (28).
    Codeword Selection: Select the codeword $C_t$ that minimizes the loss in (29).
    Error Re-weighting: Update $[w_1, \ldots, w_{n_{sample}}]$ using (30);
    Transform Update: Update $\mathbf{M}_{Region}^{t-1}$ using (31).
    $t++$;
end
Output: The learnt transformation $\mathbf{M}_{Region}$, the LDVC codebook $\mathbf{C}_{region} = \mathbf{M}_{Region}^{T}\mathbf{V}_{region}$.
```

Learning Convergence Proven Our LDVC aims to preserve the conjunctive rankings (see (27)) of the original BoW histogram with as fewer codewords as possible. In the worst case (which is almost impossible), by setting the threshold τ as $\sum_{i=1}^{n_{sample}} w_i^t = 0$, the maximum length of

LDVC is proportional to the volume of all non-zero codewords in this region, which is degenerated to Tree Histogram coding. Therefore, our boosting always converges.

3.6 Model Degeneration Analysis

rs*PCA vs. Ranking Sensitive Vocabulary Boosting* By maintaining a diagonal matrix $\mathbf{M}_{Region}\mathbf{M}_{Region}^T$, our Ranking Sensitive Vocabulary Boosting serves as a linear simplification of *rs*PCA in optimal $\mathbf{M}_{Region}$ learning as follows:

$$\begin{aligned} &\mathbf{M}_{Boosting} \\ &\text{s.t.} \quad \forall_{k\in[1,K]} \textstyle\sum_{i=1}^{M} \mathbf{M}_{i,k} = \mathbf{M}_{j,k} | \mathbf{M}_{j,k} \neq 0; \\ &\qquad \forall_{m\in[1,M]} \textstyle\sum_{k=1}^{K} |\mathbf{M}_{m,k}| \leq 1 \end{aligned} \tag{32}$$

Each column in $\mathbf{M}_{Boosting}$ has only one non-zero item; each row in $\mathbf{M}_{Boosting}$ contains only one non-zero item if any.

With this simplified matrix $\mathbf{M}_{Boosting}$ and its corresponding $\mathbf{z}$ transform, the learning of optimal $\mathbf{M}_{region}$ is simplified by a greedy gradient descendent approach: It learns only one column in $\mathbf{M}_{Boosting}$ at the kth boosting ($k = [1, K]$), which simulates the learning of $\mathbf{M}_{new}$ for *rs*PCA in (23) as:

$$\begin{aligned} \mathbf{M}_{Boosting}^{new} &= \mathbf{M}_{Boosting}^{old} + [0, \ldots, pos(t), \ldots, 0]_M \\ &\quad \times [0, \ldots, pos(k), \ldots, 0]_K^T \\ \text{s.t.} & \\ c_t &= \arg\min_j \sum_{i=1}^{n_{sample}} w_i^{k'} \sum_{r=1}^{R} R(A_r^i) \mathbf{W}_{A_r^i} \| [\mathbf{M}_{Boosting}^{old} \\ &\quad + [0, \ldots, pos(t), \ldots, 0]_M [0, \ldots, pos(k), \ldots, 0]_K^T] \\ &\quad \times \mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i}) \| \end{aligned} \tag{33}$$

which is similar to (29) that selects the tth codeword into the current boosted codeword set.

On Correlation to Word Frequency Thresholding and Tree Histogram Coding (Chen et al. 2009) The word frequency thresholding (i.e. keeping those codewords with the t highest IDF) can be interpreted by simplifying our Vocabulary Boosting as:

$$\begin{aligned} \mathbf{M}_{Boosting}^{new} &= \mathbf{M}_{Boosting}^{old} + [0, \ldots pos(t), \ldots 0]_M \\ &\quad \times [0, \ldots pos(k), \ldots 0]_K^T \\ \text{s.t.} & \\ C_t &= \arg\min_j \sum_{i=1}^{n_{sample}} w_i^{k} \sum_{r=1}^{R} \mathbf{W}_{A_r^i} \| [I_{m\times k} \\ &\quad + [0, \ldots, pos(t), \ldots, 0]_M [0, \ldots, pos(k), \ldots, 0]_K^T] \\ &\quad \times \mathbf{C}_{I'_i}, \mathbf{V}_{A_r^i} \| \end{aligned} \tag{34}$$

$[0, \ldots, pos(t), \ldots, 0]_M$ is an $M \times 1$ selection vector, which selects the jth column into the linear projection of boosting. $[0, \ldots, pos(k), \ldots, 0]_K$ is a $K \times 1$ position vector to map v_j into new codeword C_t. Equation (34) selects the codeword with the highest IDF into $\mathbf{M}_{new}^{Boosting}$. Obviously, it's suboptimal to our vocabulary boosting in two-fold:

(1) The Word Frequency Thresholding in (34) does not consider the effects of previously selected codewords in each new round of discriminative codeword selection;

(2) It also discards the ranking position $R()$ in loss function, where choosing words present in the top returning results is regardless of their positions.

Similarly, Tree Histogram Coding (Chen et al. 2009) also attempt to choose the non-zero codewords with the similar principle of (34) to incorporate all codewords with non-zero IDF into optimization. In contrast, although they are both more compact and lossy, our *rs*PCA and Ranking Sensitive Vocabulary Boosting can better preserve the retrieval ranking capability of the original BoW histograms, which would be proven subsequently in Sect. 5.5.

4 A Novel Mobile Landmark Search Framework

We further present a novel mobile landmark search framework using our LDVC descriptor, which handles two unaddressed issues: (1) how to determine the best geographical partition to extract LDVC in each region. It is addressed by a Visual Aware Spectral Clustering in Sect. 4.1; (2) how to efficiently update the visual vocabulary maintained in the mobile end for the subsequent LDVC extraction. It is addressed by a Location Based Vocabulary Adaption in Sect. 4.3, with a two-way LDVC transmission in Fig. 1.

4.1 Visual Aware Geographical Segmentation

We use the geographical tags (latitude, longitude) of landmark photos to segment each city into geographical regions. To avoid incorrect partition of the images from an identical landmark, we present a Visual Aware Spectral Clustering, which models the fact that only those geographically nearby and visually similar photos should be assigned to the same partition.

Suppose there are in total N photos in a given city, we first formulate an N-node fully connected graph $\mathbf{G}$. Each node g_i represents a photo, and a link l_{ij} denotes the visual and geographical distance between g_i and g_j. Subsequently, we aim to partition $\mathbf{G}$ into L subgraphs $\{\mathbf{G}'_l\}_{l=1}^{L}$. While direct optimal graph partition is NP hard, we resort to a spectral clustering to achieve this goal, which is proven to be equivalent to the normalized cut in Ng et al. (2001).

To strictly ensure only geographically nearby photos are partitioned into the identical region, we leverage a ε-ball operation to disconnect far-away photos in $\mathbf{G}_{N\times N}$ as:

$$\mathbf{G}_{N\times N} = \begin{cases} G_{i,j}, & G_{i,j} < \varepsilon \\ \infty, & G_{i,j} \geq \varepsilon \end{cases} \tag{35}$$

Algorithm 2: Visual Aware Spectral Clustering to Segment Geographical Regions

```
Input: Geographical Similarity Graph G.
Output: Spectrum Clustering Graph S.
$\varepsilon$-Ball Operation on G using (35);
Build Laplacian Graph $\mathbf{L} = I - \mathbf{D}^{-1/2}\mathbf{G}\mathbf{D}^{-1/2}$;
Spectral Graph Construction by SVD over L into
    $\mathbf{S}_{N\times K}$, with eigenvectors $[e_1, e_2, \ldots, e_L]$;
Clustering $N$ rows in $\mathbf{S}_{N\times L}$ into $L$ clusters using (37);
Return Spectrum clustering S as graph partition $\mathbf{G}'$;
```

Then, we build a diagonal matrix $\mathbf{D}$ whose (i, i)-element is the sum of $\mathbf{G}$'s ith row ($d_k = \sum_{n=1}^{N} G_{k,n}$), based on which a Laplacian matrix $\mathbf{L}$ is built:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{G}\mathbf{D}^{-1/2} \tag{36}$$

Subsequently, we extract the L largest eigenvectors (we denote them as $e_1, e_2, \ldots, e_L$) from $\mathbf{L}_{N\times N}$. They transform $\mathbf{L}_{N\times N}$ into a spectral matrix $\mathbf{S}_{N\times L}$, in which each row $\mathbf{S}_i$ is a L-dimensional normalized eigenvector $[e_1, e_2, \ldots, e_L] \in \mathbb{R}_L$.

We incorporate visual similarity into the clustering of the rows in $\mathbf{S}_{N\times K}$ with the similarity as:

$$Sim(\mathbf{S}_i, \mathbf{S}_j) = \|BoW_i, BoW_j\|_{Cosine} \cdot \|\mathbf{S}_i, \mathbf{S}_j\|_2 \tag{37}$$

where $\mathbf{S}_i$ and $\mathbf{S}_j$ denote two rows in $\mathbf{S}_{N\times K}$ (K-dimensional).

Based on (37), the visual similarity is integrated into the spectral graph, rather than directly in $\mathbf{G}$, because $\mathbf{G}$ is operated on with a ε-ball to disconnect geographical distant nodes, hence is hard to control the visual similarity threshold. Algorithm 2 summarizes the overall clustering; Figs. 4 and 5 show two exemplar partitions in Beijing and New York City respectively; and Fig. 9 shows a typical geographical scale distribution of regions in Beijing.

4.2 Geographical IDF Evaluation

We propose a refined word frequency measurement to replace the IDF weighting of $\mathbf{W}_{A_r^i}$ in (27), (28), (29), (30), (33), and (34), which significantly improves our *rs*PCA and Ranking Sensitive Vocabulary Boosting by distinguish the contributions of discriminative codewords better.

To facilitate landmark search, we aim to distinguish the contributions of codewords together with their spatial cues. When the descriptors falling into a codeword are geographically scatted over the entire region, the codeword is less discriminative than those concentrated in this region, even an identical IDF is produced. Hence, we incorporate the geographical codeword distribution to refine the codeword discriminability beyond the original IDF.

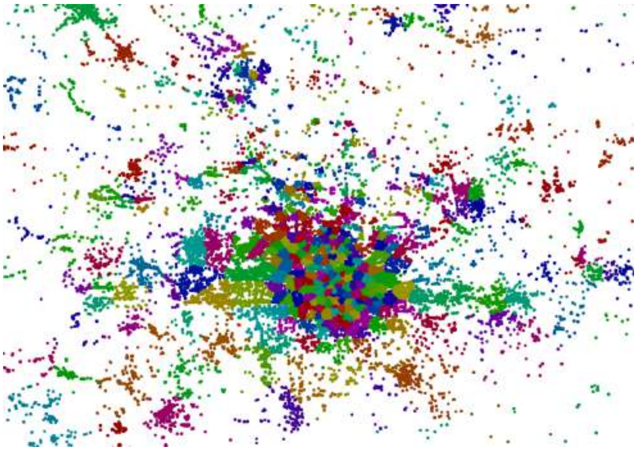


Fig. 4 (Color online) The visual aware spectral clustering to partition Beijing into geographical regions. Different colors denote different clusters

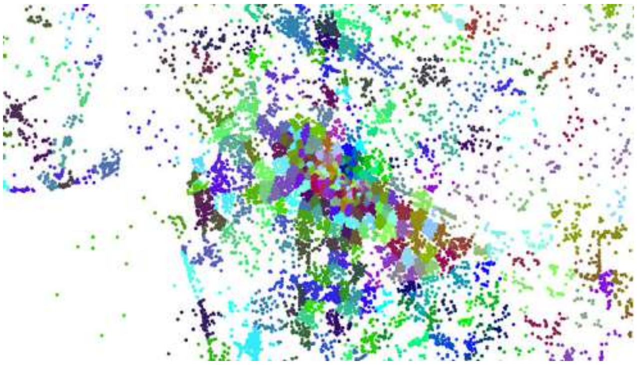


Fig. 5 The visual aware spectral clustering to partition New York into geographical regions. Different colors denote different clusters

For a given codeword V_i, we incorporate the geographical distances amongst the images containing V_i to re-estimate the original IDF $IDF^i_{Original}$ of V_i in G'_j as:

$$IDF^i_{Original} = \log\frac{N_{G'_j}}{N_i} \tag{38}$$

$N_{G'_j}$ is the number of photos in region G'_j; and N_i is the number of photos in region G'_j that contain codeword V_i. We propose a novel Geographical IDF IDF^i_{Geo} on V_i:

$$IDF^i_{Geo} = \log\left(\frac{\sum_{I_m\in G'_j}\sum_{I_n\in G'_j} GeoDis(I_m, I_n)}{\sum_{I_m\in G'_j,\ V_i\in I_m}\sum_{I_n\in G'_j,\ V_i\in I_n} GeoDis(I_m, I_n)}\right) \tag{39}$$

where $GeoDis(I_m, I_n)$ denotes their geographical distance, which is measured by the L2 distance of their corresponding

Fig. 6 The typical photo collections within Beijing, New York City, Singapore, and Florence

geographical location[6]; $I_m \in G'_j$ denotes image I_m falling into region G'_j; $V_i \in I_m$ denotes that image I_m contains V_i.

From (39), a codeword that is distributed in a more concentrated geographical scale is more likely to produce a higher IDF, and vice versa. This IDF measure is location sensitive. Towards efficient LDVC learning, the geographical IDF should be applied in each region to pre-filter out less discriminative codewords. However, since the number of non-zero codewords in each region is limited (Fig. 9), the directly LDVC learning is also efficient.

4.3 Location based Online Vocabulary Adaption

We present our location-based online vocabulary adaption in Fig. 1, in which steps ①–④ show the interleaved upstream and downstream operations in adaption:

Step ①: In a typical scenario, once a mobile user activates the landmark search functionality, the geographical location of this mobile device is sent to the remote server. This location guides the remote server to locate and assign the mobile user to one of geographical regions in this city.

Step ②: Then, the server downstream transmits $\mathbf{M}_{Region}$ of a current region to the mobile device, where the original BoW histogram is to be online updated. In other words, the location of the mobile device serves as an indicator to decide whether to update the compression settings of compact landmark descriptors.

Step ③: Once a mobile user takes a query photo as shown in Fig. 7, either SIFT (Lowe 2004) or CHoG (Chandrasekhar et al. 2009a) are directly extracted on the mobile device. Then, these local descriptors are quantized into an initial BoW histogram, which is subsequently compressed using $\mathbf{M}_{Region}$, typically producing a 10–50 bits LDVC landmark descriptor that is upstream delivered to the server.

Step ④: The server decodes the received LDVC descriptor into the original BoW histogram, and then searches near-duplicate landmark photos in the inverted indexing system. Consequently, the remote server downstream delivers the top-returning photos as well as their locations to the mobile end, as shown in Fig. 7.

Fig. 7 User interface of our mobile landmark search, which is developed in HTC DESIRE G7 smart phone (512 MB ROM + 576 MB RAM memory, an 8G extended storage, a 528 MHz processor, and an embedded camera with maximal 2592 × 1944 resolution)

5 Quantitative Evaluations

5.1 Datasets and Evaluation Criteria

Data Collection We collected over 10 million geo-tagged photos from photo sharing websites of Flickr (http://www.Flickr.com) and Panoramio (http://www.Panoramio.com). Our data covers typical areas i.e. Beijing, New York City, Lhasa, Singapore, and Florence. Figure 6 shows the exemplar photos in Beijing, New York City, Singapore, and Florence. Figure 8 shows the geographical photo distribution in Beijing, which may delineate the photograph activity of mobile users from a geographical point of view. For instance, these photos are basically distributed along roads or around landmarks. In addition, popular landmarks are more likely to have more near-duplicate photos. As more and more popular photograph viewpoints result from the consensus of photo contributors, compact LDVC descriptors are more likely to be learnt at such locations.

Generating Conjunctive Query Our system randomly selects n_{sample} photos from every region in each city as the conjunctive queries to train region-specific LDVC. Our system collects the top R returning photos for each query using the original BoW histogram, which finally form a conjunctive query set for LDVC learning subsequently.

Ground Truth Labeling for Evaluation We invite volunteers to label landmark queries and their correct matching:

[6] In our implementation, we adopt GPS to infer the location. Other location cues, such as base station information, can be also used.

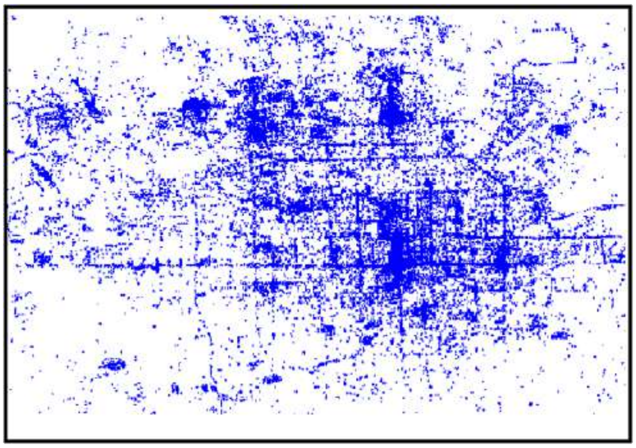


Fig. 8 The geographical distribution of community contributed photos with geographical tagging in Beijing, for which the location distribution can reveal human activities in their photographing manners

- For each city, we select the top 30 most dense regions as well as 30 random selected regions based on the geographical partition.
- Since manually identifying the complete list of related photos (query and reference) is intensive, we prefer to identify one or more dominant landmark views from each of these 60 regions. In practice, all their near-duplicate photos are manually labeled in its belonging and nearby regions.
- Finally, 300 queries as well as their ground truth of ranking lists in each city are yielded. In total, we have 300 × 5 (cities) queries in the overall evaluation.

Building Scalable Vocabulary Tree For the landmark photo collection in each city, we extract both SIFT (Lowe 2004) and CHoG (Chandrasekhar et al. 2009a) features from each photo. Using all features in this city, we build a Scalable Vocabulary Tree model (Nister and Stewenius 2006) $\mathbf{V}$ using hierarchical k means clustering, which generates a BoW histogram $\mathbf{V}_i$ for each database photo I_i. We denote the hierarchical layers as H and the branching factor as B. We stop the further quantization division when a leaf node contains less than 1,000 descriptors, and this leaf node becomes a codeword. This settlement gives at most $M = B^H$ words at the finest level. In a typical setting, we have $H = 5$ and $B = 10$ to produce approximate 100,000 codewords. For Boosting and *rs*PCA, we built up and maintain a single SVT tree based on Beijing photo database, which is used in all touristic cities i.e. Beijing, New York City, Lhasa, Singapore, and Florence.[7] Our subsequent experiments will show that our LDVC adaptation ensures the promising search effectiveness and efficiency when a mobile user enters any geographical region.

Evaluation Criterion Both Precision@N and Mean Average Precision at N (MAP@N) are used to evaluate our performance. Both are widely used in state-of-the-arts (Sivic and Zisserman 2003; Philbin et al. 2007; Schindler and Brown 2007; Chandrasekhar et al. 2009a; Jegou et al. 2010a, 2010b). MAP reveals the position-sensitive ranking precision of the queries based on the returning lists as:

$$MAP@N = \frac{1}{N_q}\sum_{i=1}^{N_q}\left(\frac{\sum_{r=1}^{N} P(r)rel(r)}{min(N, \#\text{-}relevant\text{-}images)}\right) \tag{40}$$

N_q is the number of queries; r is the rank, N the number of related images for query i; $rel(r)$ a binary function on the relevance of r; and $P(r)$ precision at the cut-off rank of r.

Note that here we have a min operation between the top N returning and #-relevant-images. In a large scale search system, there are always over hundreds of ground truth relevant images to each query. Therefore, dividing by #-relevant-images would result in a very small MAP. Alternatively, a better choice is the division by the number of returning images. We use min(N, #-relevant-images) to calculate MAP@N.

As N is at most 20 in our evaluation and always smaller than the number of labeled ground truth, we simply replace min(N, #-relevant-images) with N in subsequent calculation.[8]

It is worth mentioning that, in many cases the complete labels are not available due to the incomplete manual labeling. But the practice of labeling partial images from the nearby regions of each query has satisfied our location sensitive experimental set-up. In addition, noisy or wrong GPS tags may give imprecise or incorrect location, which leads to the ground truth matching falling outside the nearby regions. So some effective measurements (like recall@precision 1) would be inaccurate, as we are prohibited from labeling the ground truth over the entire database for exhaustive efforts otherwise.

5.2 Validating Motivations of our LDVC Coding

Geographical Distribution Sparsity of Codewords First, we show in Fig. 9 that the visual appearances of landmark images basically produce compactness or consistency within each geographical region, which investigates whether the visual vocabulary in a given region is compressible.

From Fig. 9, it is easy to figure out the sparse distribution of words among geographically nearby photos, typically 500–800 words from the original large vocabulary (say

[7] For comparison baselines, we still built the vocabulary within each city respectively.

[8] The min(N, #-relevant-images) operation is a common evaluation in the TRECVID evaluation (http://trecvid.nist.gov/).

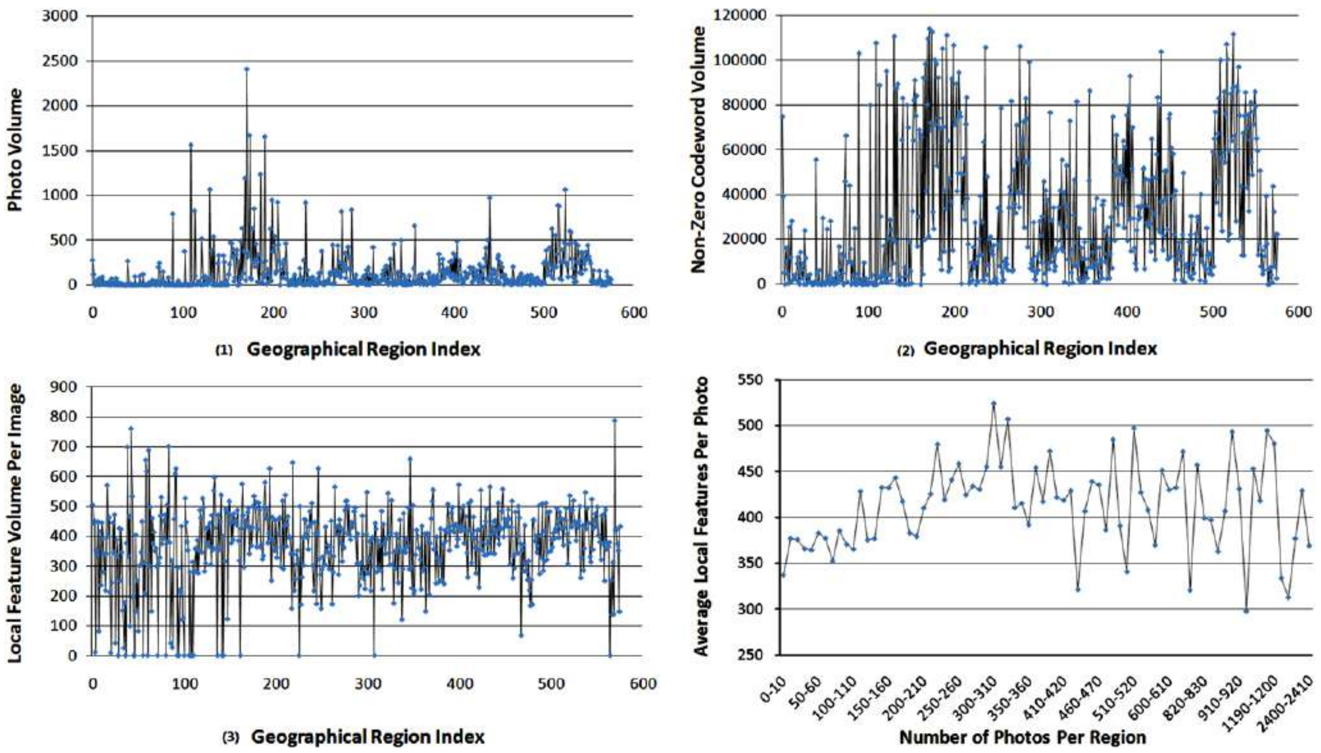


Fig. 9 (**1**) the photo volume of each geographical region; (**2**) the number of non-zero codewords in a region (the region-level visual content complexity may be revealed, and more diverse images more words); (**3**) the average number of local features per photo in a region; and (**4**) the average number of local features per photo vs. number of photos per region in Beijing (totally 576), which is an indicator that image-level visual content complexity is independent of the image number in a region

how many photos and non-zero codewords are there). As shown in Fig. 9, this sparsity depends on the scene complexity in a region. For different regions, the image volume and the number of non-zero codewords would vary, which allows us to develop scalable vocabulary coding in the coding length (or compression rate) to flexibly handle the visual content variances in different regions.

Simplifying Codeword Frequency into Occurrence Second, we investigate whether it is feasible to transmit the Hits/No-Hits (0-1) occurrence histogram instead of the original frequency histogram of BoW. If true, one dimensional signal compression methods (such as run-length coding or arithmetic coding) can be applied to compress this occurrence histogram. Figure 11 shows the performance degeneration with our Hits/No-Hits replacement, which empirically demonstrate that this simplification does not significantly degenerate precision. Hence, we prefer the replacement of Hits/Non-Hits in our LDVC coding.

Codeword Frequency Thresholding Third, we investigate the feasibility of transmitting only the most frequent codewords in each region. From a lossy compression viewpoint, this practice can discard "unimportant" codewords in the original BoW histogram that serves the subsequent low bit coding. In Fig. 13, we adjust the codeword maintenance percentages of top 10–90% by thresholding IDF or Geo-IDF (see Sect. 4.2) to produce the retrieval MAP degeneration.

Figure 13 shows that IDF thresholding produces acceptable MAP degeneration, even when keeping top 10% frequent codewords only (which leads to an $1:10$ compression rate). However, as shown in the subsequent vocabulary boosting, our LDVC can further achieve up to $1:10,000$ compression rate while maintaining over 90% MAP. Another important finding is that Geographical IDF is much more discriminative than IDF: maintaining less codewords by using Geographical IDF can yield good search performance comparable to that by maintaining more codewords using IDF. In addition, the queries in different regions often produce diverse performances in this simple compression scheme, which also validates our basic motivation of scalable descriptor compression in different regions.

Geographical Scales of Different Landmarks Finally we validate that the geographical scales of landmarks are diverse. Figure 10 shows that different queries should be performed in different scales, depending on both visual content statistics and scene constitution. For instance, the queries of "Summer Palace" are with a larger geographical scale, comparing with the queries of "Temple of Heaven". Hence, the

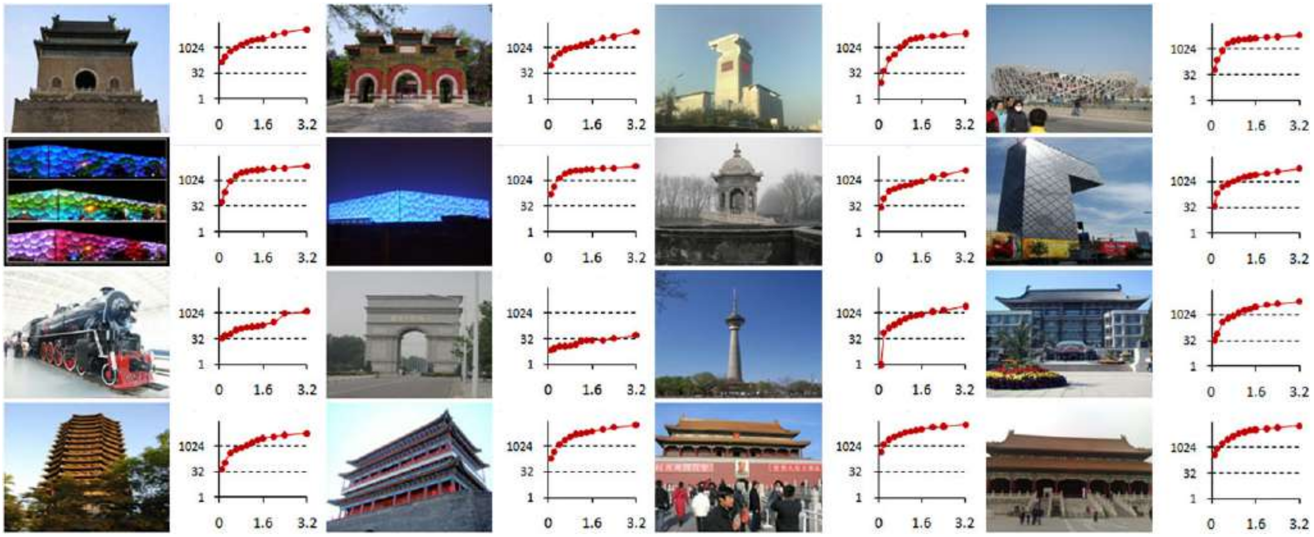


Fig. 10 Geographical photo distribution function with respect to their geographical distances to the query. x-axis: km; y-axis: photo volume

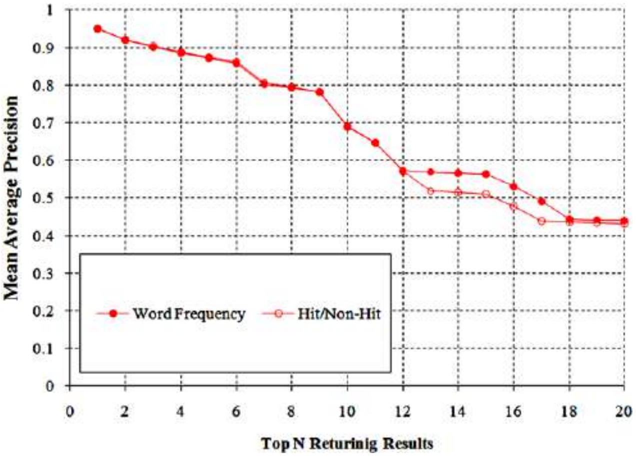


Fig. 11 Precision@N degeneration by using Hits/No-Hits (0-1) histogram instead of the original frequency histogram

region scale depends not only on the geographical distance but also the visual diversity of landmark photos.

Average Landmarks per Region For addition information on the subsequent LDVC evaluation, we give the average number of landmarks of 60 geographical regions of each city. The average number of 24 landmarks per region is much less than the maximum representation capability of LDVC (say 10–50 bits). Undoubtedly, LDVC representation has been much less redundant. Another empirical finding is that more diverse images in a region are often with longer LDVC descriptors.

5.3 Parameters Tuning

Visual Embedding and Eigenvector Selection in the Spectral Clustering This tuning relates to the geographical partition in two aspects: (1) embedding the visual discriminability; (2) selecting a proper number of eigenvectors, which are both reflected in our visual aware spectral clustering.

Figure 12 presents the results of tuning both factors. With visual embedding (see (37)), we achieved better performance of searching with less codewords in both IDF thresholding and Vocabulary Boosting for codeword compression. In practice, validation experiments of tuning above parameters can be done for each city to figure out the best number of regions in geographical partitioning. Figure 12 shows the influence of selecting the number of regions in the visual aware spectral clustering. There is a trade off between setting the number of regions and better LDVC performance. That is, smaller regions typically have higher LDVC compression rates, but demand more frequent downstream vocabulary adaption. Meanwhile, incorrect matching happening on the region margins would be more probable. On the contrary, larger regions typically produce a less compact LDVC set due to the large visual complexity in that region, with less mismatching happening on the margin.

Geographical IDF vs. Original IDF Comparing with the original IDF, we further reveal our geographical IDF is more discriminative to identity important codewords and to preserve the ranking capability of the original BoW in each region. In Fig. 14, with the selected top 10%–90% codewords by thresholding IDF or Geographical IDF, we measure MAP degeneration. Clearly, our Geographical IDF is better in identifying the most discriminative codewords towards a compact landmark descriptor.

One explanation lies in that the traditional IDF completely ignores the spatial distribution of visual words. An intuitive fact is a frequent codeword would be less discriminative when it is with diverse geographical distribution. On

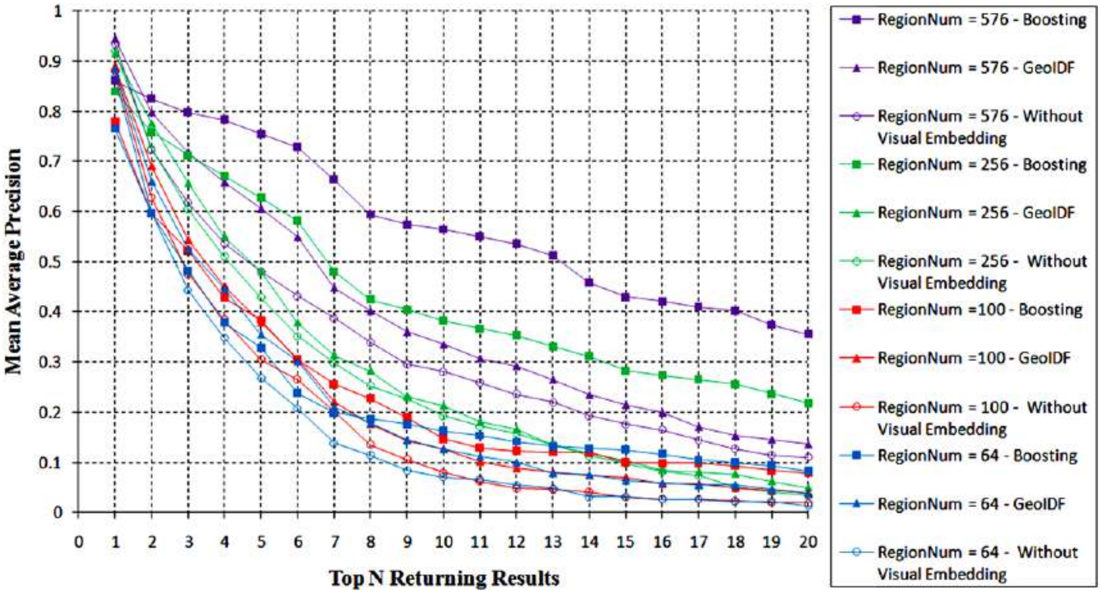


Fig. 12 The influence of both visual discriminability embedding and different region numbers in Visual Aware Spectral Clustering, measured by Precision@N using our ground truth query set

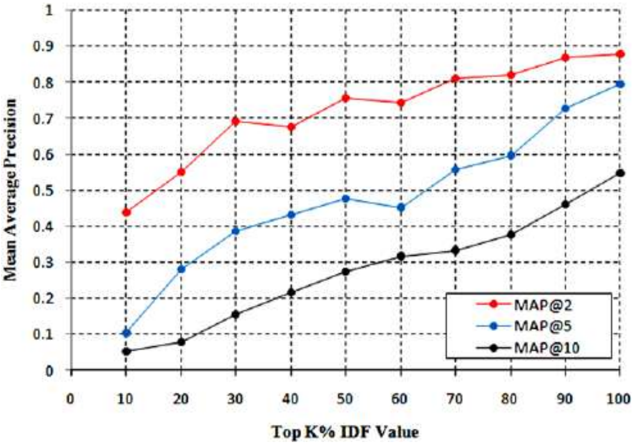


Fig. 13 The retrieval MAP loss by maintaining codewords with the top $k\%$ IDF or top $k\%$ Geographical IDF values

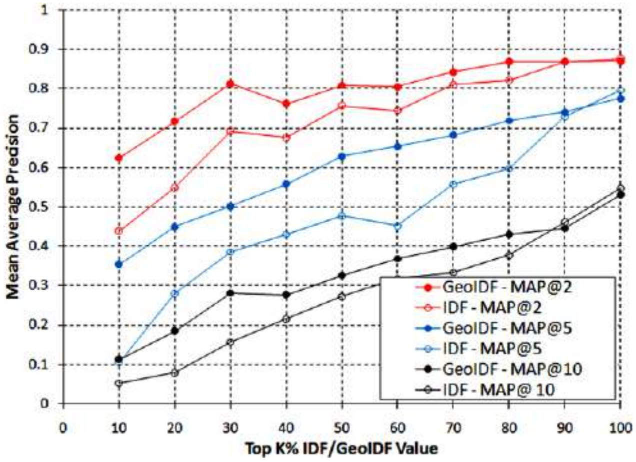


Fig. 14 Geographical IDF comparison with the original IDF by maintaining only the top K highest IDF using our ground truth query set

the contrary, once a codeword is more concentrated in a given region it could be more discriminative with identical or even less (original) frequency. This is a commonsense in image distribution at landmarks as many near-duplicate photos are geographically nearby. Those concentrated codewords with higher geographically IDF have higher priority to be selected into a LDVC set.

Compression Rates in both rs*PCA and Ranking Sensitive Vocabulary Boosting* There are other parameters in our ranking sensitive LDVC descriptors. In both *rs*PCA and Ranking Sensitive Vocabulary Boosting, one important parameter is the compression rate with respect to the MAP degeneration: For *rs*PCA, we need to decide how many principle components in the learnt $\mathbf{M}_{region}$; For the Ranking Sensitive Vocabulary Boosting, it means how many codewords are boosted from training in the conjunctive queries. Section 5.5.3 shows the quantitative comparisons of applying different types of rate distortion.

Average Energy Consumption On a mobile device, we are constrained by the battery life. Therefore, energy conserving is critical for mobile applications. One interesting study is to compare the average mobile energy consumption in: (1) extracting and sending compact descriptors, (2) extracting and sending the BoW signature, and (3) sending the original query image. We empirically test the number of image queries that the mobile can send before the battery

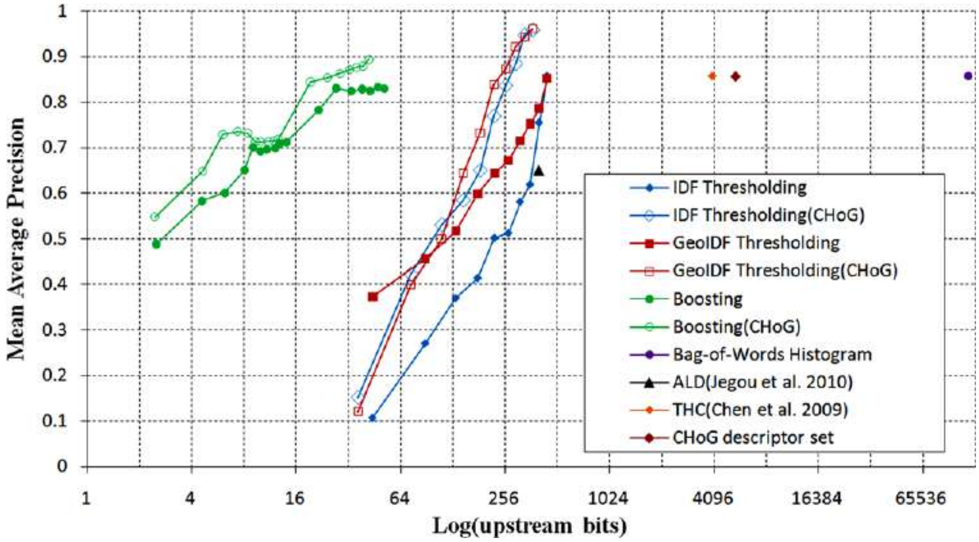


Fig. 15 Compression rate and ranking distortion analysis comparing with approaches in Chen et al. (2009), Chandrasekhar et al. (2009a), Jegou et al. (2010a) using our ground truth query set. The corresponding downstream LDVC set cost are shown in Fig. 19

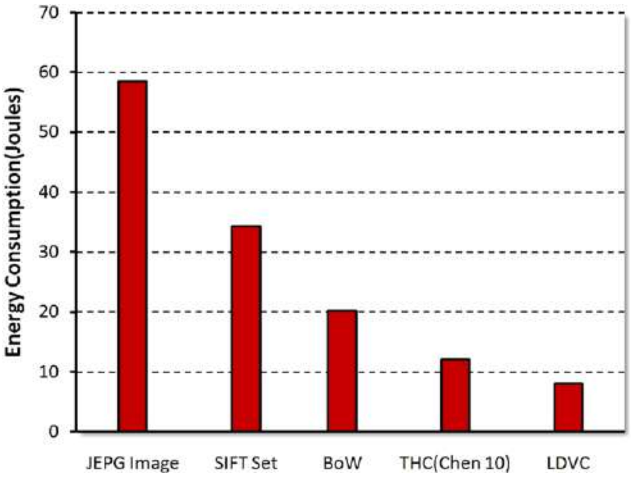


Fig. 16 Comparison of average energy consumption through 3G wireless link, through transmitting an entire query image, extracting and transmitting LDVC, and other compact descriptors

runs out of power for 3G network connections. A typical phone battery has a voltage of 4.0 V and a capacity of 1400 mAh (or 20.2 k joules). Hence, for 3G connections, the maximum number of images that the mobile can send is 20.2 k joules/52.4 joules = 385 total queries. For the extraction and transmitting of our proposed LDVC, we are able to perform 20.2 k joules/8.1 joules = 2494 total queries, which are 6 times as many queries as transmitting the entire query image.

Figure 16 reveals that sending either original query images or the high-dimensional descriptors would cause serious energy consumption, comparing with performing visual descriptor compression on the mobile and then sending the compact descriptors instead.

The study on Concurrent Query Capability At the server end, a mobile visual search system is constrained by available wireless bandwidth, which limits the capability of concurrent upstream queries as well as the concurrent downstream deliveries of search results. To run an online visual search service, we have to apply for the wireless bandwidth from the network service providers and its maintenance is often of high cost. The relations of concurrent query, bandwidth (upstream/downstream), as well as query delivery delay may be briefed as follows:

$$Bandwidth_{Overall} = Bandwidth_{Upstream} + Bandwidth_{Downstream} \tag{41}$$

$$\begin{aligned} |Query_{Concurrent}| &= \frac{Bandwidth_{Upstream}}{Delivery_{Upstream}/Delay_{Upstream}} \\ |Result_{Concurrent}| &= \frac{Bandwidth_{Downstream}}{Delivery_{Downstream}/Delay_{Downstream}} \end{aligned} \tag{42}$$

$|Query_{Concurrent}|$ denotes the number of concurrent (upstream) queries that the server end can receive; $|Result_{Concurrent}|$ the number of (downstream) concurrent results that the server can deliver; and $Delivery_{Upstream}$ and $Delivery_{Downstream}$ denote the upstream and downstream delays respectively.

At the server end, the capability of receiving concurrent queries depends on both upstream query delivery size and the wireless delivery delay. It is practically useful to make the upstream data delivery as small as possible. For instance, suppose that a server is with a 10 Mbps link, and the wireless network delivery delay is 2 second on average, sending one query photo of 100 KB would lead to 100 KB × 8/2 = 400 Kbps bandwidth cost, and then the server end can concurrently receive 10 Mbps/400 Kbps = 25.6 queries in total. If the size of each upstream query is reduced from 100 KB to 100 B (by LDVC), the capability of receiving multiple queries would be scaled up by 1,000 (approximate 25,000 queries), regardless of the searching capability of the backbone of a cluster of servers.

In practice, due to the technical restrictions or running cost considerations, the significant enhancement in 3G wireless linking cannot be guaranteed. With the ever increasing mobile computing capability, our compact LDVC descriptor can in principle improve the throughput for most state-of-the-art mobile visual search frameworks. More importantly, once the upstream bandwidth cost is much reduced, more bandwidth can be saved up to improve the amount of concurrent delivery of search results.

5.4 Comparison Baselines

1. *Original BoW Histogram*: Transmitting the original BoW histogram has the lowest compression rate (without any compression). It is supposed to provide an upper bound of search performance without any word information loss. However, as shown in our experiments, the noisy codewords should be removed in the learning process.
2. *Word Frequency Compression*: This baseline serves as the most straightforward solution to compress the histogram representation of a landmark, which retains those codewords with the highest IDF values. The subsequent experiment will show that it is suboptimal comparing with LDVC in terms of both compression rate and ranking precision.
3. *Geographical Word Frequency Compression*: This baseline serves as an alternative scheme for lossy codeword compression. We will show that the geographical IDF yield better performance than the original IDF.
4. *Vocabulary Boosting Regardless of Ranking Positions*: This baseline is to show the effectiveness of applying the ranking position to our ranking loss model. We degenerate the Rank function by dismissing the influence of ranking positions in Vocabulary Boosting based LDVC learning.
5. *Probabilistic PCA (*p*PCA)*: To quantize the effectiveness of ranking supervision in our *rs*PCA, we use unsupervised probabilistic PCA to learn the best vocabulary coding $\mathbf{M}_{region}$ in each region.
6. *Ranking Sensitive PCA (*rs*PCA)*: This is an optimal solution of our LDVC descriptor, which learns a non-linear vocabulary transformation with minimal ranking loss. One negative point is low efficiency due to the EM style learning, which is addressed by Baseline (7).
7. *Ranking Sensitive Vocabulary Boosting*: This is a simplified version of *rs*PCA, which adopts Boosting to learn LDVC by taking into account the ranking position loss. We will show good performance comparable to rsPCA, but with superior computational efficiency. Comparing with Baseline (3), much better performance results with ranking positions.
8. *Aggregating Local Descriptors* (Jegou et al. 2010a): To the best of our knowledge, the work in Jegou et al. (2010a) reported the most comparable performance to our LDVC in terms of compactness, which employs subspace quantization to obtain a low bit rate signature that is to approximate the square distance between original BoW histograms. For fair comparison to *rs*PCA and Ranking Sensitive Vocabulary Boosting, we fed the BoW histogram of the original vocabulary **V** as the input.
9. *Tree Histogram Coding (THC)* (Chen et al. 2009): Chen et al. (2009) compressed the SVT histogram, which serves as the most related work to ours. Their scheme encoded the position difference of non-zero bins in BoW histogram, which produces an approximate 2 KB coding length per image for a vocabulary with 1 M words (much less than directly sending the BoW histogram that costs more than 5 KB).
10. *CHoG* (Chandrasekhar et al. 2009a): As an alternative approach, we replace the SIFT descriptor with CHoG, which produces a more compact local descriptor with approximate 50 bits, This may reduce the storage cost in a mobile device. In subsequent experiments, we report better performance (rate distortion) by replacing SIFT with CHoG. The reason is that, landmark queries are often taken carefully and may not produce serious scale and rotation variations, hence descriptor invariance is not very elementary (indeed would occasionally degenerate the precision by introducing description synonymy). Finally, we choose CHoG as the local descriptor in our landmark search prototype systems.

5.5 Quantitative Results and Comparisons

5.5.1 Ranking Effectiveness Comparisons

Figures 15 and 18 show that our proposed *rs*PCA and Ranking Sensitive Vocabulary Boosting won superior performance over Baselines (3)–(5). As a linear simplification, our Ranking Sensitive Vocabulary Boosting achieves the promising performance comparable to *rs*PCA. In our landmark search prototype systems, Ranking Sensitive Vocabulary Boosting is finally employed.

We would like to compare Baselines (8)(9)(10) in Fig. 15 subsequently, since Fig. 18 maintains only the top 300 most discriminative codewords to show the 2D Precision@N comparison graph. For Baseline (8), we directly apply the code available in Jegou et al. (2010c), and then we fix the upstream rate at 512 bits. For Baseline (9), the Tree Histogram Coding (THC) maintains all non-zero codewords, which differs from the comparison in Fig. 18 in maintaining the top 300 codewords. The comparison of Baseline (10) is well shown in Fig. 15.

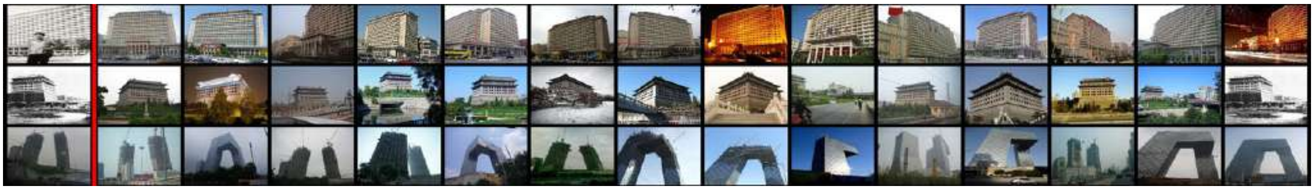
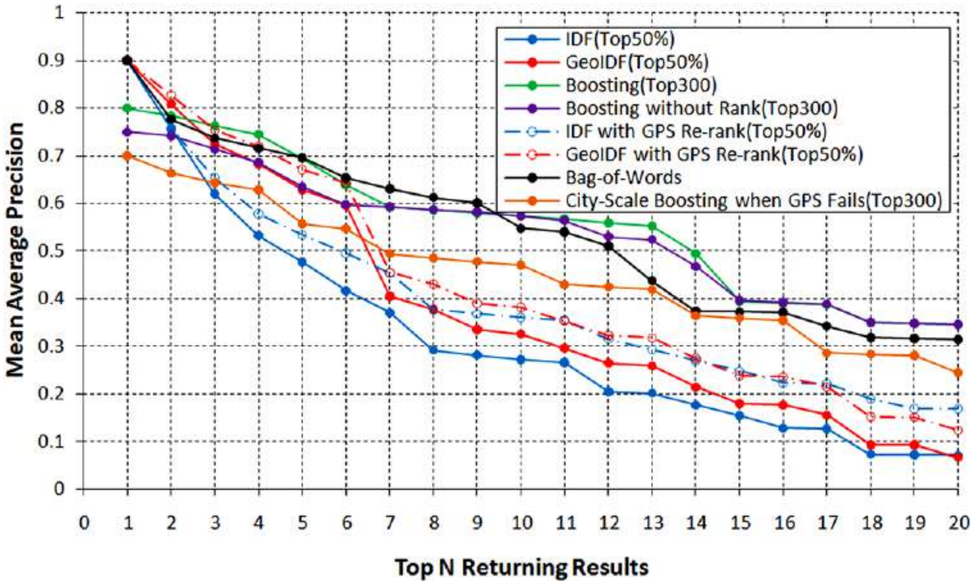


Fig. 17 *Top*: query of the Peking Hotel taken at the early of 1950 right after its completion. *Middle*: A query of Northeast Wall in Beijing, taken at 1860 during the Qing Dynasty of China. *Down*: A fake query of the CCTV Building in construction

Fig. 18 MAP comparisons with respect to the top N ranking positions using our ground truth query set. In this figure, "Top300" denotes we maintain the top 300 codewords by settling $K = 300$ in $\mathbf{M}_{M\times K}$, with comparisons to using GPS to re-rank IDF, GeoIDF, original vocabulary, and THC (Chen et al. 2009). "Top50%" denotes we maintain the top 50% codewords with the highest IDF or GeoIDF



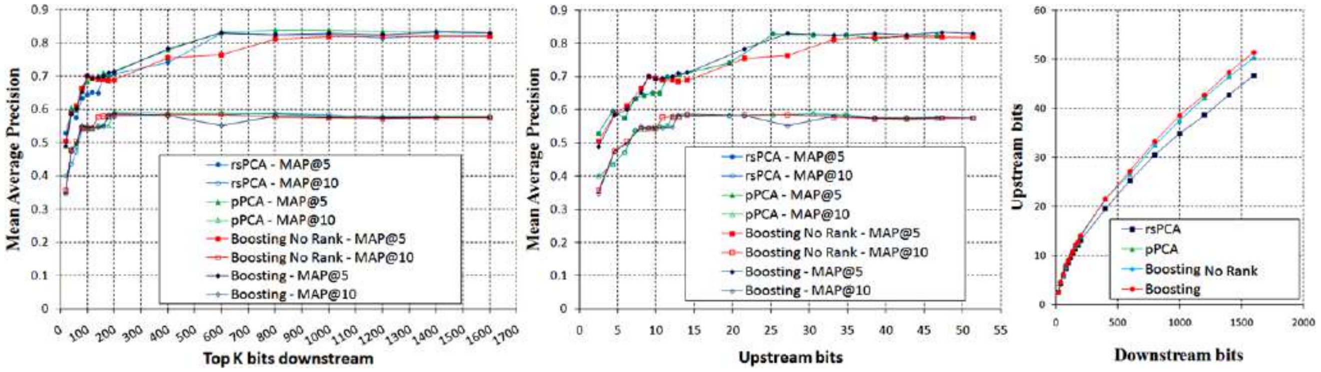


Fig. 19 The upstream and downstream transmission cost with respect to the ranking distortions

5.5.2 *Comparing with GPS based Visual Search Re-ranking*

We did comparisons with the alternative approaches that use GPS to re-rank the visual search results (including baselines of IDF, GeoIDF, original vocabulary, and THC (Chen et al. 2009)). Results show that our Boosting outperforms those schemes with GPS based visual search re-ranking. One explanation goes to the performance degeneration of noisy GPS signals or location tags (Fig. 25).

In addition, we provide the performance of applying Boosting at the city-scale. This is to study the cases when the exact location of a query is missing except the correct identity of a given city by parsing the base station information. Now the task is to offline learn a compact codebook by using the images from an entire city. Once the server identifies the missing GPS information from the mobile upstream delivery, this city-level codebook will be sent to the mobile for coding landmark descriptors. As shown in Fig. 18, although the performance degenerates when non region-specific Boosting is applied, our Boosting scheme at the entire city sill outperform those approaches (IDF when $N > 3$ and GeoIDF when $N > 7$). However, to achieve desirable precision, a less compact LDVC results in the absence of GPS usage.

5.5.3 *Rate Distortion Analysis*

To compare our LDVC descriptors to the baselines (Chen et al. 2009; Chandrasekhar et al. 2009a; Jegou et al. 2010a), we give the rate distortion analysis. Both *rs*PCA and Ranking Sensitive Vocabulary Boosting have achieved the best performances. From Fig. 15, we can see the highest compression rates with comparable distortion (horizontal), as well as the highest ranking performance with comparable compression rates (vertical). In addition, without using ranking position embedding (Baseline (4)(5)), we have achieved better performance than almost all alternatives and state-of-the-art approaches.

In some cases, our LDVC descriptor can even outperform the original BoW histogram (Baseline (1)). Our *rs*PCA and Ranking Sensitive Vocabulary Boosting attempt to extract the most discriminative words by learning from the conjunctive ranking list. Therefore, both schemes may largely filter out the incorrect matching from cluttered, occlusion, etc. This mechanism is in spirit similar to "Query Expansion" in information retrieval, where incorrect ranking of a given query can be well improved from those conjunctive rankings (in the current region). This is one important reason why more compact LDVC yields more robustness than the original BoW.

5.5.4 *Upstream and Downstream Transmission Cost*

We further study both upstream and downstream transmission cost from the location based vocabulary adaptation. As a mobile users enter a new region, this region's LDVC set is downstream transmitted to the mobile device. Note that a mobile user may deliver multiple queries in a region, so the downstream adaptation and the upstream query deliveries do not have an one-to-one relationship, but an one-to-many correspondence. However, based on the rate distortion analysis in Fig. 19, even for one-to-one correspondence, we can still achieve the lowest bit rate, as well as the highest ranking MAP, comparing with all state-of-the-art baselines (Chen et al. 2009; Jegou et al. 2010a) in Fig 15. Furthermore, the right subfigure of Fig. 19 shows that the data amount of upstream delivery is almost linear to downstream delivery.

5.5.5 *Memory and Time Cost in Mobile Devices*

We deploy the prototype system on HTC Desire G7 as a software App. HTC DESIRE G7 is equipped with an embedded camera with maximal 2592 × 1944 resolution, a Qualcomm MSM7201A processor at 528 MHz, a 512 M ROM + 576 M RAM memory, 8G extended storage, and an embedded GPS. Table 1 further shows the memory and time part of our search system with comparisons to state-of-the-art works in Chen et al. (2009), Chandrasekhar et al. (2009a), Jegou et al. (2010a). In our LDVC descriptor extraction, the most time-consuming cost comes from the local feature extraction, which can be further accelerated by random sampling, instead of using the interest point detectors (Chandrasekhar et al. 2009a; Lowe 2004).

5.5.6 *Case Study*

Figure 20 gives some examples of our Ranking Sensitive Vocabulary Boosting comparing with (1) state-of-the-art works in Chen et al. (2009) and the original BoW histogram (both have identical performance), and (2) IDF Thresholding. We can observe that the current LDVC descriptor achieves comparable (or even better) performance to the

Table 1 Memory (MB) and time (Second) requirements for SVT and the availability on several mobile phones

Tree Structure	Memory Requirement
SIFT SVT, $H = 6$, $B = 10$	59 MB
CHoG SVT, $H = 6$, $B = 10$	24 MB
Aggregate Local Descriptors (Jegou et al. 2010a)	728 MB

Compression Methods	Feature Extraction	BoW Generation	Vocabulary Coding
BoW Histogram	1.25S	0.14S	≈0S
GeoIDF Compression	1.25S	0.14S	≈0S
Aggregate Local Descriptors (Jegou et al. 2010a)	1.25S	0.14S	1.5S
Tree Histogram Coding (Chen et al. 2009)	1.25S	0.14S	≈0S
Vocabulary Boosting	1.25S	0.14S	≈0S
*rs*PCA	1.25S	0.14S	≈0S

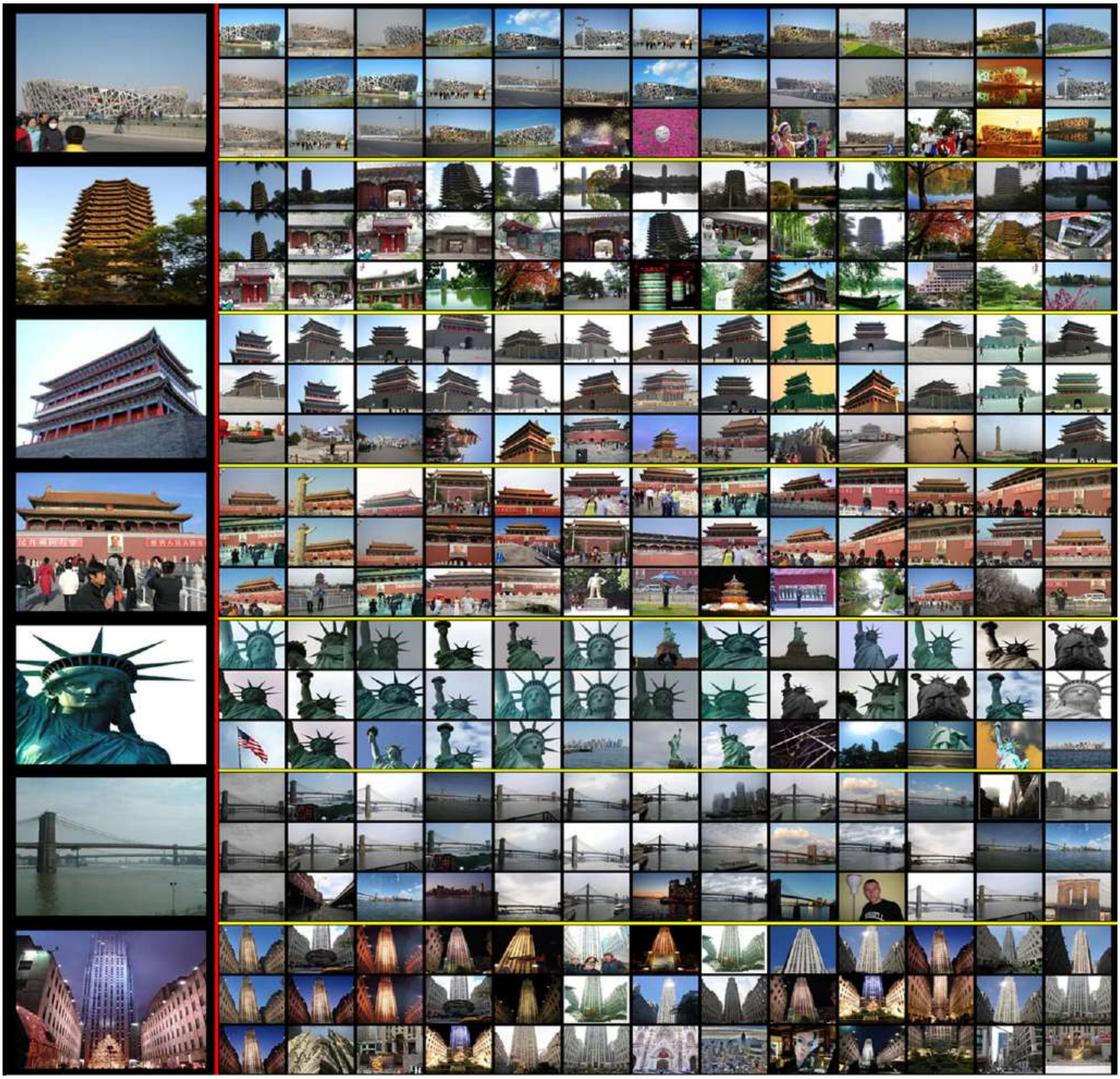


Fig. 20 The visualized ranking performance of Ranking Sensitive Vocabulary Boosting based LDVC landmark descriptor in comparison to the alternative approach in Chen et al. (2009). Each photo on the *left* is the query, each line of returning results corresponds to an approach. *Top*: LDVC; Middle: Original BoW feature or Tree Histogram Coding (Chen et al. 2009) (since work in Chen et al. 2009 is a lossless compression scheme, it produces identical returning results to the BoW feature); *Bottom*: IDF Thresholding

original BoW histogram and the state-of-the-arts (Chen et al. 2009).

We further study 6 groups of query scenarios to visualize the performance comparison with Baselines (1)(2)(9) (Chen et al. 2009). These cases are most likely to cause mismatches in our empirical study.

Scene Variations Since our landmark photos are collected from photo sharing websites Flickr and Panoramio, our database involves extensive landmark appearance diversity from changes in seasons and building appearances during times. For instance, Fig. 17 shows several unrealistic landmark queries taken long time ago, together with their search results using LDVC based descriptors.

Illumination, Scale, and Blurring Variations Some queries happen at night; some queries occur in different scales (from either nearby views or distant views); and mobile phones

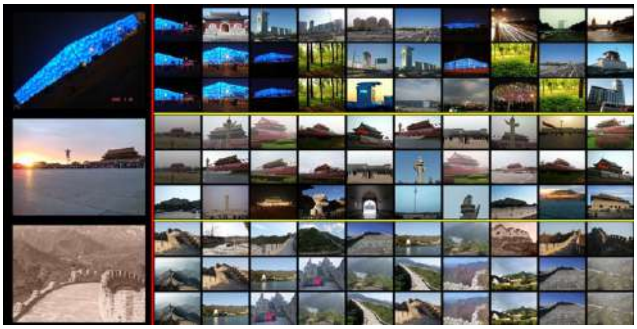


Fig. 21 Case study of landmark search with illumination changes, scale changes, and blurred photographing. Each photo on the *left* is the query, each line of returning results corresponds to an approach. *Top*: LDVC; *Middle*: Original BoW feature or Tree Histogram Coding (Chen et al. 2009); *Bottom*: IDF Thresholding

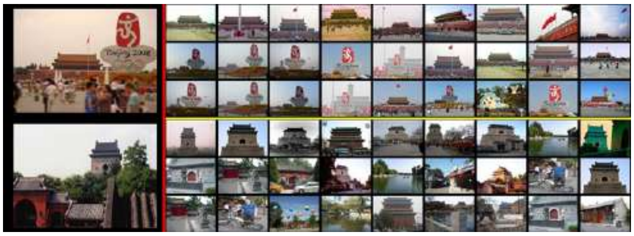


Fig. 22 Case study of landmark search with occlusions and partial landmark queries. Each photo on the *left* is the query, each line of returning results corresponds to an approach. *Top*: LDVC; *Middle*: Original BoW or Tree Histogram Coding (Chen et al. 2009); *Bottom*: IDF Thresholding

often produce blurred queries. Figure 21 shows the list of ranking results where LDVC descriptors better preserve the ranking precision, comparing with Baseline (1)(2)(9) (Chen et al. 2009).

Photographing Occlusions and Partial Matching We selected a set of suboptimal queries containing partial occlusions from foreground objects or peoples, as well as with partial landmark views. Figure 22 shows that our LDVC descriptor outperforms Baseline (1)(2)(9) (Chen et al. 2009). Currently, geometry consistency is not used to verify the top returning images. The reason is that we are validating the LDVC descriptor in compactness and effectiveness in representing visual content. So all alternative approaches are performed without geometry verification. However, spatial re-ranking as well as geometry consistency verification are very useful complements to our approach. We may transmit both LDVC Hits/Non-hit and the spatial layout of words. A spatial pyramid or max/averaged pooling (Wang et al. 2010) can be employed to encode spatial layout in the BoW histogram representation.

Location Distortions in Our Database We evaluate the robustness of LDVC by adding Gaussian random distortion to the GPS of database images. With the same pipeline, we re-learn LDVC descriptors over the distorted image datasets. Figure 22 shows the results of different distortions in the upstream GPS location: (1) The original GPS location is distorted with Gaussian Noise scale = 30 m (a typical case that GPS is obtained within a building or nearby dense buildings). Our LDVC yields promising performance comparable to that using precise GPS, more robust than alternative approaches in Fig. 15 and Fig. 18; (2) GPS falls outside the current region (thereby a wrong LDVC set is assigned to the current query), which leads to performance degeneration. To address this, we can further adopt a larger geographical scale in city map segmentation to reduce the probability that a query falls outside its true region.

Exemplar Transmission Rate Figure 23 shows several groups of exemplar upstream transmission rates of Baseline (2)(3)(7). LDVC produces variant coding lengths for different landmark queries. In general, LDVC generates the most compact descriptor, comparing to simply maintaining the most discriminative codewords by using IDF or Geographical IDF thresholding. The latter (GeoIDF) yields a compression rate that is most comparable to LDVC.

Where LDVC Descriptors are Matched Figure 24 illustrates where our LDVC descriptors yield the matches from a query photo to the database images, where different colors show that those descriptors are quantized into different codewords in LDVC. Not all the detected local features in a query (on average 300–400) are kept track of BoW. In contrast, LDVC deals with the most discriminative local features only.

Which LDVC Codewords are Transmitted Finally, we investigate which LDVC codewords are transmitted the upstream query. Figure 26 illustrates that LDVC selects the most discriminative local patches for a given landmark query. By visualizing the centroid patch of the codeword to which a LDVC landmark descriptor is assigned, we can easily identify different non-zero codewords in different queries. In our LDVC coding, these words are delivered as the most important cue to represent and distinguish the visual appearance of a query.

6 Conclusions and Future Works

We have leveraged the mobile-end visual descriptor extraction to reduce the latency of visual query delivery over the (3G) wireless link. Distinct from previous works, our proposed Location Discriminative Vocabulary Coding (LDVC) exploits the pervasive location context to generate extremely

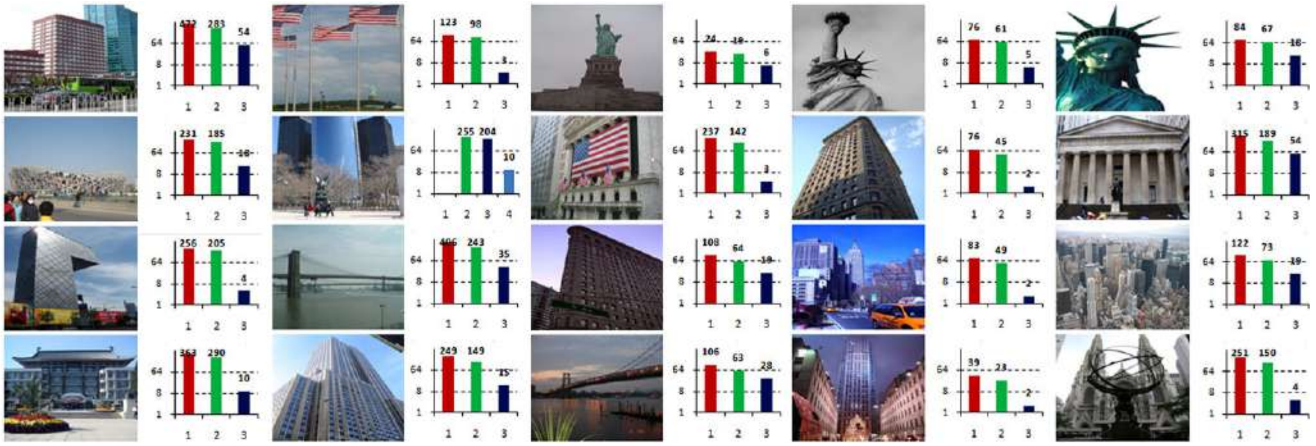


Fig. 23 (Color online) Case study of the upstream transmission rates for representative landmark queries in Beijing and New York. Each photo on the *left* is the query, its left histogram is the transmission rate of Baseline (2) (*Red*) Baseline (3) (*Green*), and Baseline (7) (*Blue*)

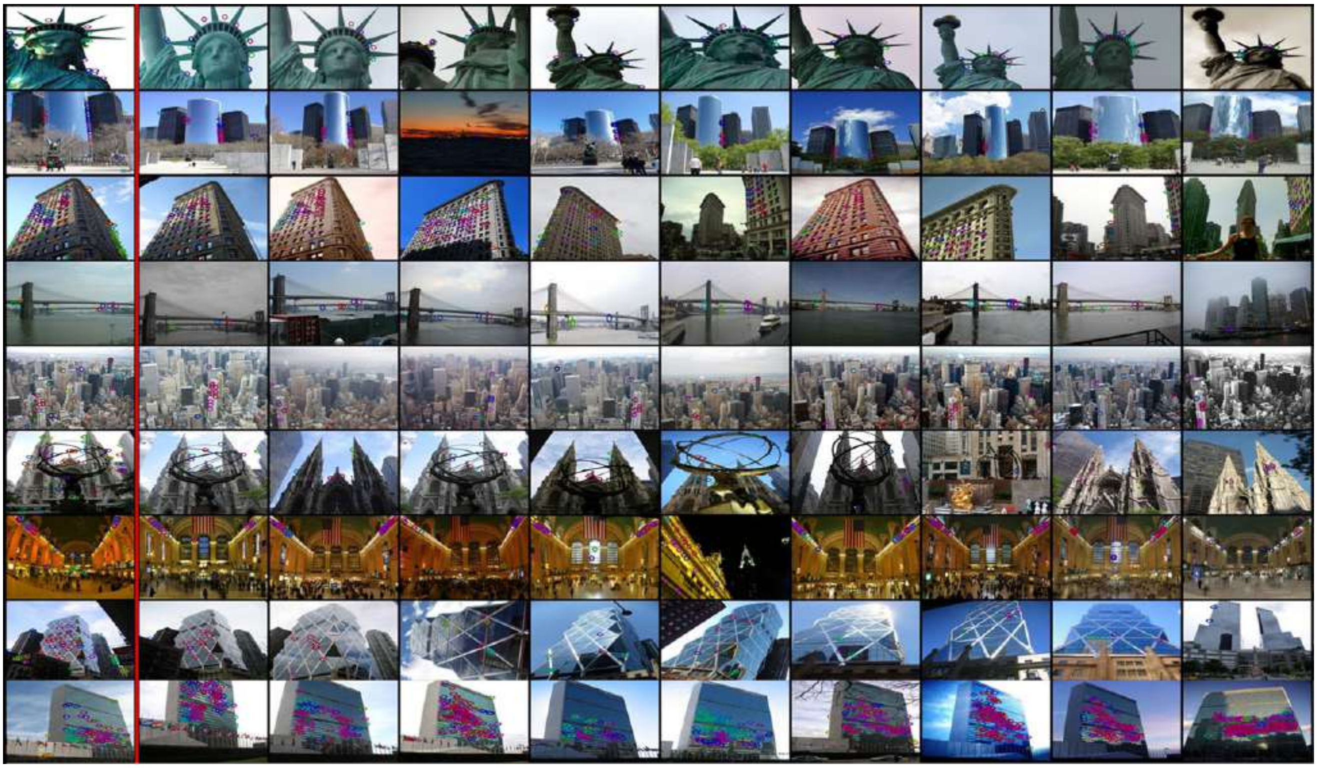


Fig. 24 (Color online) Case study of the spatial matching for LDVC descriptors between query (*left photo*) and the top returning results. Different colors denote different codewords

compact visual descriptors. We have come up with a unified framework to accomplish promising landmark search from three aspects: low transmission cost, discriminative description, as well as scalable descriptor delivery. In particular, our proposed location based vocabulary adaptation breakthroughs the traditional one-way upstream query delivery pipeline. By using a preliminary downstream adaptation with respect to different locations, the mobile device is to extract a region-specific LDVC descriptor. Such teaching mode is especially suitable for two way communication of a mobile device. We have successfully deployed the mobile landmark search system in a million scale landmark database covering five typical areas like Beijing, New York City, Lhasa, Singapore, and Florence. The extensive experimental results have shown that our LDVC descriptor has significantly outperformed state-of-the-art compact visual

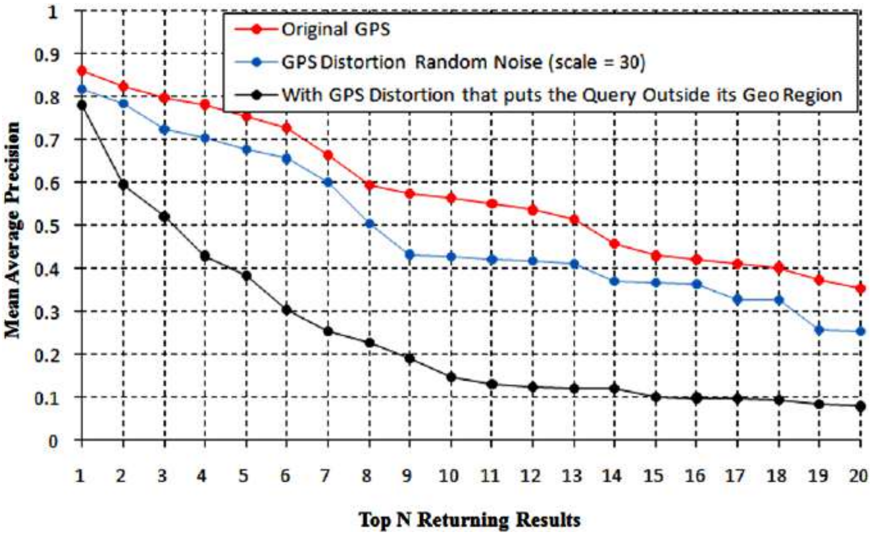


Fig. 25 Precision@N degeneration by adding location distortions in our ground truth query set

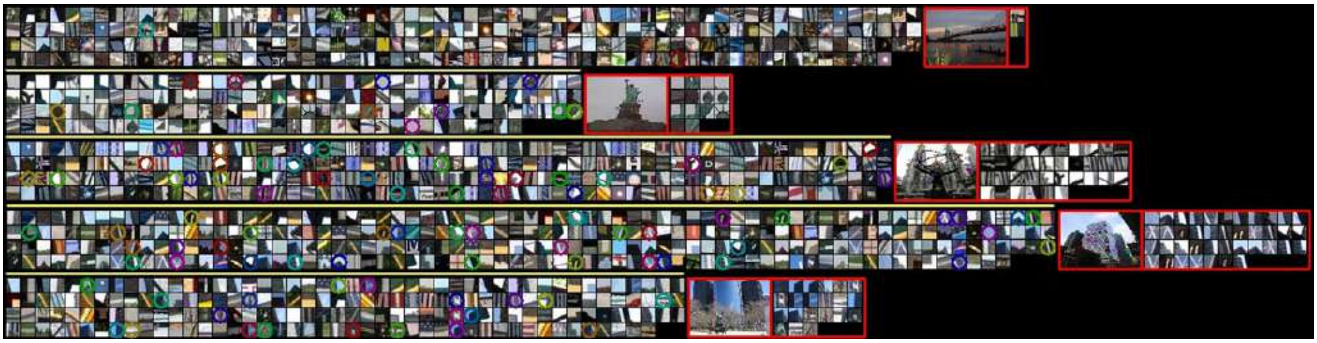


Fig. 26 Case study of the LDVC set as well as the transmitted LDVC landmark descriptor in the several landmark queries in New York. *Left*: the LDVC set within its current geographical region; *Middle*: the detected codewords that are preserved after LDVC coding; *Right*: the corresponding codeword centers of the non-zero LDVC codeword bins

descriptors (Nister and Stewenius 2006; Chen et al. 2009; Chandrasekhar et al. 2009a; Jegou et al. 2010a) in terms of compression rate (10–50 bits with arithmetical coding) and retrieval MAP in mobile landmark search.

Two interesting issues remain open: First, as imprecise GPS does not significantly degenerate the LDVC performance, it is worthy to explore the capability of other coarse but easily available location information (like base station identity) to learn our LDVC descriptor. Since this can be directly accessed from the query the requirement of initial upstream GPS updates can be removed. Second, as a natural extension, to allow the landmark search system to scale up from million-scale to billion-scale, we need to make further efforts in both parallel computing and distributed indexing techniques that are relevant to more comprehensive research of scalability issues in the Web scale search applications.

Acknowledgements This work was supported by National Basic Research Program of China under contract No. 2009CB320902, in part by grants from the Chinese National Natural Science Foundation under contract No. 60902057, and in part by the CADAL Project Program.

References

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: speed up robust features. In *ECCV* (pp. 450–459).

Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., & Girod, B. (2009a). CHoG: Compressed histogram of gradients a low bit-rate feature descriptor. In *CVPR* (pp. 2504–2511).

Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Singh, J., & Girod, B. (2009b). Transform coding of image feature descriptors. In *VCIP*. doi:10.1117/12.805982.

Chandrasekhar, V., Chen, D., Lin, A., Takacs, G., Tsai, S., Cheung, N., Reznik, Y., Grzeszczuk, R., & Girod, B. (2010). Comparison of local feature descriptors for mobile visual search. In *ICIP* (pp. 3885–3888).

Chen, D., Tsai, S., & Chandrasekhar, V. (2009). Tree histogram coding for mobile image matching. In *DCC* (pp. 143–152).

Chen, D., Tsai, S., Chandrasekhar, V., Takacs, G., Vedantham, R., Grzeszczuk, R., & Girod, B. (2010). Inverted index compression for scalable image matching. In *DCC* (pp. 525–552).

Crandall, D., Backstrom, L., Huttenlocher, D., & Kleinberg, J. (2009). Mapping the world's photos. In *WWW* (pp. 761–770).

Cristani, M., Perina, A., Castellani, U., & Murino, V. (2008). Geolocated image analysis using latent representations. In *CVPR* (pp. 1–9).

Eade, E.-D., & Drummond, T.-W. (2008). Unified loop closing and recovery for real time monocular SLAM. In *BMVC*

Freund, Y., & Schapire, R. (1994). A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory* (Vol. 904, pp. 23–37).

Hays, J., & Efros, A. (2008). IMG2GPS: estimating geographic information from a single image. In *CVPR* (pp. 1–8).

Hua, G., Brown, M., & Winder, S. (2007). Discriminant embedding for local image descriptors. In *ICCV* (pp. 1–8).

Irschara, A., Zach, C., Frahm, J., & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *CVPR* (pp. 2599–2606).

Jegou, H., Douze, M., & Schmid, C. (2009). Packing bag-of-features. In *ICCV* (pp. 1–8).

Jegou, H., Douze, M., Schmid, C., & Perez, P. (2010a). Aggregating local descriptors into a compact image representation. In *CVPR* (pp. 3304–3311).

Jegou, H., Douze, M., & Schmid, C. (2010b). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *99*, 1.

Jegou et al. (2010c). http://www.irisa.fr/texmex/people/jegou/src/compactimgcodes/index.php.

Ji, R., Xie, X., Yao, H., Ma, W.-Y., & Wu, Y. (2008). Vocabulary tree incremental indexing for scalable scene recognition. In *ICME* (pp. 869–872).

Ji, R., Xie, X., Yao, H., & Ma, W.-Y. (2009a). Hierarchical optimization of visual vocabulary for effective and transferable retrieval. In *CVPR* (pp. 1161–1168).

Ji, R., Xie, X., Yao, H., & Ma, W.-Y. (2009b). Mining city landmarks from blogs by graph modeling. In *ACM Multimedia* (pp. 105–114).

Kalogerakis, E., Vesselova, O., Hays, J., Efros, A., & Hertzmann, A. (2009). Image sequence geolocation with human travel priors. In *CVPR* (pp. 1–8).

Ke, Y., & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR* (pp. II-506–II-513).

Kennedy, L., Naaman, M., Ahern, S., Nail, R., & Rattenbury, T. (2007). How Flickr helps us make sense of the world: context and content in community-contributed media collections. In *ACM Multimedia* (pp. 631–640).

Lee, J.-A., Yow, K.-C., & Sluzek, A. (2008). Image-based information guide on mobile devices. In *Advances in Visual Computing* (pp. 346–355).

Li, X., Wu, C., Zach, C., Lazebnik, S., & Frahm, J.-M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV* (pp. 427–440).

Liu, D., Scott, M., Ji, R., Yao, H., & Xie, X. (2009). Location sensitive indexing for image-based advertising. In *ACM Multimedia* (pp. 793–796).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Makar, M., Chang, C., Chen, D., Tsai, S., & Girod, B. (2009). Compression of image patches for local feature extraction. In *ICASSP* (pp. 821–824).

Mikolajczyk, K., & Schmid, C. (2005). Performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2006). A comparison of affine region detectors. *International Journal of Computer Vision*, *29*(11), 1735–1783.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In *NIPS* (pp. 849–856).

Nister, D., & Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR* (pp. 2161–2168).

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabulary and fast spatial matching. In *CVPR* (pp. 1–8).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*.

Schindler, G., & Brown, M. (2007). City-scale location recognition. In *CVPR* (pp. 1–7).

Shao, H., Svoboda, T., Tuytelaars, T., & Van Gool, L. (2003). Hpat indexing for fast object/scene recognition based on local appearance. In *CIVR*, (Vol. *2728*, pp. 71–80).

Sivic, J., & Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *ICCV* (pp. 1470–1477).

Tipping, M., & Bishop, C. (1997). Probabilistic principle component analysis. Technical Report, Neural Computing Research Group, Aston University.

Torralba, A., Fergus, R., & Weiss, Y. (2008). Small codes and large databases for recognition. In *CVPR* (pp. 1–8).

Tsai, S., Chen, D., Takacs, G., & Chandrasekhar, V. (2010). Location coding for mobile image retrieval. In *MobileMedia*

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2010). Locality-constrained linear coding for image classification. In *CVPR* (pp. 3360–3367).

Weiss, Y., Torralba, A., & Fergus, R. (2009). Spectral hashing. In *NIPS* (pp. 1753–1760).

Witten, I., Moffat, A., & Bell, T. (1999). *Managing gigabytes: compressing and indexing documents and images* (2nd edn.). San Francisco: Morgan Kaufmann.

Xiao, J.-X., Chen, J.-N., Yeung, D.-Y., & Quan, L. (2008). Structuring visual words in 3D for arbitrary-view object localization. In *ECCV* (pp. 725–737).

Yeh, T., Lee, J., & Darell, T. (2007). Adaptive vocabulary forest for dynamic indexing and category learning. In *CVPR* (pp. 1–8).

Yeo, C., Ahammad, P., & Ramchandran, K. (2008). Rate-efficient visual correspondences using random projections. In *ICIP* (pp. 217–220).

Zhang, W., & Kosecka, J. (2006). Image based localization in urban environments. In *3DVT* (pp. 33–40).

Zheng, Y. T., Zhao, M., Song, Y., & Adam, H. (2009). Tour the world: building a web-scale landmark recognition engine. In *CVPR* (pp. 1085–1092).