On-Device Mobile Landmark Recognition Using Binarized Descriptor with Multifeature Fusion

TAO GUAN and YUESONG WANG, Huazhong University of Science & Technology
LIYA DUAN, Institute of Oceanographic Instrumentation, Shandong Academy of Sciences
RONGRONG JI, Xiamen University

Along with the exponential growth of high-performance mobile devices, on-device Mobile Landmark Recognition (MLR) has recently attracted increasing research attention. However, the latency and accuracy of automatic recognition remain as bottlenecks against its real-world usage. In this article, we introduce a novel framework that combines interactive image segmentation with multifeature fusion to achieve improved MLR with high accuracy. First, we propose an effective vector binarization method to reduce the memory usage of image descriptors extracted on-device, which maintains comparable recognition accuracy to the original descriptors. Second, we design a location-aware fusion algorithm that can fuse multiple visual features into a compact yet discriminative image descriptor to improve on-device efficiency. Third, a user-friendly interaction scheme is developed that enables interactive foreground/background segmentation to largely improve recognition accuracy. Experimental results demonstrate the effectiveness of the proposed algorithms for on-device MLR applications.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.4.m [**Information Systems Applications**]: Miscellaneous; I.2.10 [**Vision and Scene Understanding**]: Vision

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Mobile landmark recognition, on-device, binarization, feature fusion, user interaction

ACM Reference Format:
Tao Guan, Yuesong Wang, Liya Duan, and Rongrong Ji. 2015. On-device mobile landmark recognition using binarized descriptor with multifeature fusion. ACM Trans. Intell. Syst. Technol. 7, 1, Article 12 (September 2015), 29 pages.
DOI: http://dx.doi.org/10.1145/2795234

1. INTRODUCTION

With the proliferation of mobile devices, Mobile Landmark Recognition (MLR) has attracted extensive research attention in the past decade. As a sort of Location Base Service (LBS) [Dey et al. 2010], MLR systems enable a mobile user to capture an image by using the embedded camera on a mobile device to recognize his or her current

This research is supported by the Special Fund for Earthquake Research in the Public Interest No. 201508025, National Natural Science Foundation of China (NSFC) under Grant No. 61272202, No. 61422210, and No. 61373076.
Authors' addresses: T. Guan and Y. Wang, College of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China; emails: {qd_gt, wwjqust}@126.com; L. Duan, Institute of Oceanographic Instrumentation, Shandong Academy of Sciences, Qingdao 266001, Shandong, China; email: jessduanjessduan@126.com; R. Ji (Corresponding Author), Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, Fujian, China; email: rrji@xmu.edu.cn.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2015 ACM 2157-6904/2015/09-ART12 $15.00
DOI: http://dx.doi.org/10.1145/2795234

geographical location. The recognized location and its corresponding information can be pushed to a mobile end device to enable various LBS applications (e.g., assisted city guide or navigational tools) [Yap et al. 2010]. For instance, a tourist can capture an image of his or her target landmark by using a mobile phone to search for related tips, such as name, history, or suggestions for dining, entertainment, shopping, and transportation.

The user experience of MLR applications depends greatly on two factors: recognition accuracy and response time. On one hand, recognition should be as accurate as possible to avoid trial-and-failure for mobile users. On the other hand, the returning results to users should be as timely as possible to avoid the loss of user patience.

To obtain accurate MLR, most current systems [Baatz et al. 2010; Chen et al. 2011a; Ji et al. 2012; Liu et al. 2012a; Sang et al. 2013; Yap et al. 2012] are deployed on a client-server architecture that stores reference images of locations (typically at city scale) at the server(s), and landmark recognition is accomplished via matching the visual features sent from the mobile end. Such an architecture faces a key limitation in response time, especially when using a low bit-rate wireless link. This response time highly depends on how much information is transferred up- and downstream. For example, recognition can fail in areas with no cell phone service. Moreover, the user's personal information, such as route of travel, can be disclosed easily if the captured images and their GPS information stored on the server(s) are leaked.

In view of these problems, on-device MLR has attracted increasing research interest recently thanks to the computational ability of modern mobile devices. Using these devices, the system can still work even in regions without wireless links. Moreover, personal information is not disclosed because no data are uploaded to a remote server(s). Although promising, it remains challenging to scale up recognition for an entire city due to the storage limitations of mobile devices. First, the search system must compact enough to be deployed directly on the mobile device's RAM. Second, to a ensure good user experience, recognition accuracy should be comparable to that found in client-server based MLR systems. Despite recent ongoing research [Chen et al. 2013; Guan et al. 2013; Schroth et al. 2011], these remain open issues in state-of-the-art systems. For example, Chen [2013] used hundreds of bytes to represent a single image, which is still not compact enough to manage millions of images directly on the mobile end. Guan's method [Guan et al. 2013] can compress each image descriptor into several bytes; however, the user experience is less than satisfactory because recognition accuracy is reduced due to the significant compression of image descriptors. In summary, the design of a compact and accurate city-scale landmark search method has become a vital problem for on-device MLR systems.

In this article, we tackle this problem from both system and user perspectives by designing more compact yet accurate descriptors on the system level, as well as designing intelligent interactive operations for the user level to improve on-device MLR:

(1) At the system level, we focus on designing a more discriminative visual descriptor to reduce memory consumption. To this end, existing systems often use only one single visual feature to perform MLR; this is, however, typically insufficient to characterize scene content. Instead, we propose a multifeature fusion scheme that innovates at the following points: First, we propose a simple and effective vector binarization method that can reduce the memory usage of image descriptors without significantly decreasing recognition accuracy. Second, we design a location-aware fusion algorithm to fuse multiple visual features into a compact and discriminative descriptor. Third, we jointly optimize feature fusion and indexing procedures to improve recognition accuracy.
(2) At the user level, we introduce an interactive foreground/background segmentation algorithm to improve both recognition accuracy and user experience. Our system

adopts interactive foreground segmentation to help users to formulate their intent query more conveniently, which avoids repeated trial-and-failure by identifying "relevant" visual descriptor for a target landmark. Noisy and irrelevant visual features like road, sky, and trees are eliminated from the subsequent feature extraction and similarity ranking procedures.

The rest of this article is organized as follows: Section 2 discusses the related work. Section 3 gives an overview of the proposed framework. Section 4 introduces the proposed visual feature binarization and fusion algorithm together with the index structure optimization scheme. Section 5 discusses the proposed interaction design. Section 6 shows experimental results. Finally, we discuss the open issues in Section 7.

2. RELATED WORK

Our work is closely related to MLR, image descriptor quantization and encoding, and multiple visual features fusion.

2.1. Mobile Landmark Recognition

State-of-art MLR systems [Biancalana et al. 2013; Chandrasekhar et al. 2011; Ji et al. 2014; Ji et al. 2012; Ji and Yao et al. 2012; Ji et al. 2011; Wu et al. 2012; Xia et al. 2012] commonly follow a client-server design that is highly focused on compressing the visual descriptor to be transmitted through the wireless link to reduce query delivery latency. For example, Chandrasekhar et al. [2011] propose a low bitrate Compressed Histogram of Gradients (CHoG) descriptor that can be matched in the compressed domain to facilitate mobile visual search applications. Ji et al. [2011, 2012] designed a location-aware encoding strategy to compress a Bag-of-Features (BOF) from different channels individually for low-bit-rate MLR. Wu et al. [2012] proposed a new mobile visual search scheme that can achieve a very low bit rate for transmission between a mobile client and a cloud. Xia et al. [2012] design a progressive geometric-preserving transmission method by dividing the query image into blocks and local features.

To improve the recognition accuracy of MLR systems, Chen et al. [2011a] fused facade-aligned and viewpoint-aligned street view images to improve recognition performance. Yap et al. [2012] proposed an efficient MLR system by incorporating saliency information into different stages of a Vocabulary Tree (VT)-based image search process. Tseng et al. [2012] used (DT) features to precisely capture the shape information to build a sketch-based mobile image search system. Chen et al. [2011b] fused GPS, a digital compass, and visual information to achieve good recognition performance.

Recently, on-device MLR architecture [Chen et al. 2013; Guan et al. 2013; Schroth et al. 2011] has attracted increasing attention. For example, Chen et al. [2013] compressed Residual Enhanced Visual Vector (REVV) descriptors by using Linear Discriminant Analysis (LDA) to obtain a compact image descriptor used for on-device MLR. Guan et al. [2013] proposed using compressed visual descriptors in combination with mobile sensors for fast on-device MLR.

Regarding interactive mobile visual search, Wang et al. [2011] proposed a system to formulate a visual query in a natural way. The system takes full advantage of multimodal and multitouch interactions on mobile devices, allowing users to easily formulate a composite image as their search intent by interacting with a phone through voice and multitouch. Sang et al. [2013] introduced four modes of gesture-based interactions (crop, line, lasso, and tap) and presented a prototype of interactive mobile visual search—TapTell—to help users formulate their visual intent more conveniently. The system leverages limited yet natural user interactions on a phone to improve search efficiency while maintaining a satisfactory user experience. In this research, we design a user-friendly interface by using interactive image matting to enable users to

formulate their search intention in a more efficient way. Experiments show that our interface can not only enhance the user experience but also improve search accuracy.

In terms of modeling and exploiting 3D landmarks, Xian et al. [2012] adopted 3D models to improve landmark recognition accuracy. And Min et al. [2014] further proposed adopting 3D models as the basic unit for landmark recognition.

2.2. Image Descriptor Quantization and Encoding

To deploy city-scale on-device MLR, image descriptors have to be extremely compressed to store the entire database on the RAM of a mobile device. To this end, the asymmetric distance-based methods [Brandt et al. 2010; Chen et al. 2010; Jegou et al. 2011] are especially suitable: These perform approximate nearest neighbor search directly in the compressed domain. For example, Guan et al. [2013] proposed an efficient encoding method by combining Transform Coding [Brandt et al. 2010] and Residual Vector Quantization [Chen et al. 2010] to obtain memory-light on-device MLR.

Another feasible method is to quantize visual descriptors into binary vectors, which can then be matched efficiently by using Hamming distance. Many binarization techniques have been proposed recently, including Locality Sensitive Hashing [Datar et al. 2004], Spectral Hashing [Weiss et al. 2008], Locality Sensitive Binary Codes [Raginsky et al. 2009], and Integrated Binary Encoding [Zhang et al. 2011]. However, reduction in memory usage and search time comes at the cost of sacrificing recognition accuracy. In this research, we design a simple and effective vector binarization method that makes use of cheaply available location cues such as GPS tags to maintain retrieval accuracy while reducing the memory footprint drastically.

2.3. Multiple Visual Features Fusion

In the field of Content-Based Image Retrieval (CBIR), extensive works [Douze et al. 2011; Fernando et al. 2012; Gehler et al. 2009; Liu et al. 2012b; Song et al. 2011; Yang et al. 2013; Ye et al. 2012] have improved retrieval accuracy by fusing multiple features; these methods can typically be categorized into late-fusion and early-fusion algorithms.

The late-fusion algorithms [Liu et al. 2012b; Ye et al. 2012] search different features separately and then combine the obtained results. For example, Tsai et al. [2011] proposed a mobile book spine recognition system by combining text- and image-based spine recognition pipelines. Chen et al. [2013] used weighted sum fusion to fuse CHoG and SURF features to improve retrieval performance. Although promising, the late-fusion algorithms significantly increase search time and memory usage.

The early-fusion algorithms [Douze et al. 2011; Fernando et al. 2012; Gehler et al. 2009] fuse multiple features *before* recognition. For example, Fernando et al. [2012] presented a Logistic Regression-Based Feature Fusion (LRFF) method that takes advantages of different cues. Douze et al. [2011] proposed fusing visual attributes and Fisher vectors followed by a product quantization-based feature compression for fast and compact recognition. Although promising, most early-fusion strategies rely on a Support Vector Machine (SVM), which is computationally inefficient for mobile devices. In this article, we design a location-aware fusion algorithm that directly performs early fusion over the multiple visual features and produces a compact and discriminative image descriptor, which is then leveraged to perform location recognition on-device. To the best of our knowledge, our work is the first attempt to use an early-fusion strategy to perform city-scale visual search directly for on-device MLR applications.

The proposed method is an extension of our previous work [Guan et al. 2013], with key new innovations:

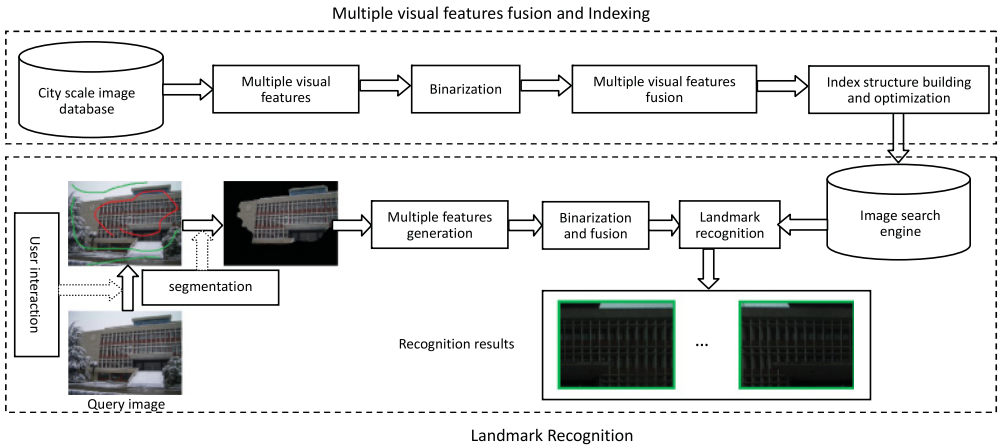


Fig. 1. Architecture of the proposed on-device MLR system.

(1) A novel interactive scheme is designed for city-scale on-device MLR. This scheme adopts efficient interactive technology to improve the accuracy and user experience of MLR.
(2) We propose to add and fuse multiple features beyond BOF to improve the recognition accuracy of on-device MLR, which has been quantitatively shown to significantly outperform our previous scheme.
(3) In this work, we further propose an effective, supervised binarization scheme to further compress features, which provides comparable retrieval accuracy while largely reducing the storage requirement for on-device MLR.

2.4. Interactive Mobile Application

It is widely admitted that a smart and flexible interactive design is the key to a successful mobile application. To this end, in the literature, interactive design for mobile-based applications has been widely studied [Rother et al. 2004; Li et al. 2004; Wang et al. 2013; Fails et al. 2003; Yeh et al. 2005; White et al. 2007]. For instance, Wang et al. [2013] developed a lazy snapping algorithm that facilitates on-device snapping in a user-friendly manner with minimum required operations. The work in Fails et al. [2005] targets designing an interactive technology that allows users to train, classify/view, and correct the classifications. And finally, works in Yeh et al. [2005] and White et al. [2007] have proposed an interactive species identification technique specifically for user interaction on a mobile device.

3. OVERVIEW OF THE PROPOSED FRAMEWORK

As shown in Figure 1, the proposed on-device MLR framework can be divided into following two stages:

Offline multiple feature fusion and indexing: The task of this stage is to build an image search engine that will be deployed for on-device MLR. To do this, we first extract multiple visual features from each database image, which are then converted into binary vectors using the method introduced in Section 4.2. We then propose a location-aware fusion scheme to fuse these binary features in Section 4.3. Subsequently, we build and optimize an index structure for on-device MLR in Section 4.4.

Online landmark recognition: In this stage, user interaction is needed to segment the target landmark from a query image first. Then, multiple binary visual features of the segmented landmark are fused to generate a visual descriptor for the search.

Finally, GPS and pose information from the query image is used to find a candidate image set that will be searched by the method given in Section 5 for landmark recognition purposes.

4. MULTIPLE VISUAL FEATURES GENERATION AND FUSION

4.1. Visual Features

In this section, we briefly introduce the three image descriptors used in our system: the Vector of Locally Aggregated Descriptors (VLAD) [Jegou et al. 2012], the BOF [Nister et al. 2006], and the Pyramid of Histograms of Oriented Gradients (PHOG) [Bosch et al. 2012]. Why we do not use other descriptors, such as color [Wengert et al. 2011], is discussed in Section 8.

VLAD: We detect 64-dimensional gravity-aligned SURF [Bay et al. 2008; Kurz et al. 2011] features from the database images and then use a K-means algorithm to generate 64 cluster centroids. To generate the VLAD descriptor of an image, we first assign each detected gravity-aligned SURF feature to its nearest cluster centroid. Then, a 4,096-dimensional VLAD descriptor is obtained by directly concatenating the aggregated residual vector in all the centroids.

BOF: We build a vocabulary tree (depth = 4 and branch = 10) with 1K visual words using the same set of gravity-aligned SURF features used in VLAD. Using the built vocabulary tree, we generate the BOF descriptor of each image using the standard TF-IDF [Sivic et al. 2003] method.

PHOG: We first extract canny edges and then quantize the gradient orientation on the edges (0° to 180°) into 20 bins. Three spatial pyramid levels are used (1×1, 2×2, and 4×4). We also align each input image according to the direction of gravity to make the generated PHOG descriptors rotation-invariant.

4.2. Binarization Method

We quantize each visual descriptor into a binary vector to reduce memory usage. Most existing methods (e.g., BoF or Fisher vector) rely on analyzing the distribution of the training set with a time-consuming learning process to achieve binarization. In this research, we design a simple binarization method targeted at finding an optimum threshold for each descriptor.

In the field of CBIR, normalized visual descriptors are commonly used. In the query stage, the visual descriptor of query image is compared with the database descriptors to find the target image with the minimum descriptor distance to the query image. This operation is based on the principle that visual descriptors from the same scene should be as similar as possible and vice versa. Without loss of generality, it is reasonable to assume that such descriptors should have similar values on the majority of their corresponding dimensions. We can then reasonably imagine that if we can obtain *similar thresholds* for visual descriptors from the same scene, we can then obtain similar binarized descriptors for them. In fact, several approaches are proposed to determine the value of the needed threshold. For example, we can set the value of the threshold as the mean of the normalized descriptor. However, in this article, we design a more stable threshold computation and binarization method as

$$b_t = \begin{cases} 1 & if\ |\sqrt{v_t}| \geq \sum_{t=1}^{T} |\sqrt{v_t}|/T \\ 0 & Otherwise \end{cases}, \tag{1}$$

where $V = (v_1\, v_2 \ldots v_T)$ is the normalized input image descriptor to be binarized, $B = (b_1\, b_2 \ldots b_T)$ is the obtained binary vector corresponding to V, and T is the

dimensionality of the image descriptor. The square root operation in Equation (1) is used to reduce the influence of peaky dimensions. Equation (1) can be treated as an adaptive threshold operator implemented onto every dimension of a given vector V. Note that since we adapt a SQRT operation to the value of each dimension, the influence of a peaky dimension is reduced. By Equation (1), the difference between two descriptors in the peaky dimension would be unified once they are higher than the given threshold. This is based on our practical observation that such a "dominant" dimension typically wrongly dominates the subsequent similarity measurement.

In addition, the binarization threshold for each dimension is determined based on the data distribution projected on this dimension. This differs from random projection because we do not change the original feature representation. One very important issue is that, because we need to do this binarization on a mobile device, we need a very fast scheme; thus, random projection is less effective due to its need for a large projection data matrix that costs time and memory. In addition, because we subsequently adopt boosting, we need the original features to be more meaningful, rather than random. In our future work, we may try random projection to see if it can replace our binarization + boosting-based scheme.

Our binarization method has the following advantages: (i) While the traditional binarization methods commonly decrease retrieval accuracy, our method can significantly improve accuracy, as shown in the experiments described in Section 6.1, mainly due to the use of boosting-based supervised feature selection to filter out features from foreground clutter (people, trees, vehicles). (ii) Our method is computationally efficient because the time-consuming codebook training process is completely avoided. (iii) The method can deal with different kinds of image descriptors. We not only rely on visual features to solely "reduce" the information, but we also integrate ranking lists as supervised labels to "add" labeled information. Thus, although we cannot achieve better results compared to "feature + label," we can reasonably outperform "feature" only.

4.3. Location Aware Multifeature Fusion

Although the memory usage of binarized image descriptors is much lower than that of the original ones, it is still impossible to store millions of binarized descriptors directly on a mobile device. Moreover, not all visual features are useful for recognizing the target landmark due to the complexity and diversity of landmark appearance. In view of that, we design an early-fusion strategy to fuse multiple binary visual descriptors into a compact and discriminative binary descriptor. Our method is based on the following two observations: (i) For each landmark, there will be some representative features or dimensions that can be used to efficiently distinguish this landmark from others; and (ii) in MLR, the whole area is commonly partitioned into different regions to facilitate the landmark recognition process. Inspired by the first observation, we propose carrying out feature fusion by selecting the most important dimensions from multiple features. Inspired by the second observation, the feature fusion is carried out in different regions respectively to overcome interference from irrelevant scenes.

In view of this, we design a location-aware multiple-feature fusion method using a boosting algorithm to individually select the most discriminative dimensions from multiple features in each region. Before giving the detailed fusion algorithm, we first introduce the symbols used in our algorithm:

(1) The concatenation of multiple normalized visual descriptors of an image is represented as

$$V = (V_{VLAD}, V_{BOF}, V_{PHOG}) = (\underbrace{v_1, \ldots, v_{4096}}_{V_{VLAD}}, \underbrace{v_{4097}, \ldots, v_{5096}}_{V_{BOF}}, \underbrace{v_{5097}, \ldots, v_{5516}}_{V_{PHOG}}), \quad (2)$$

where V_{VLAD}, V_{BOF}, and V_{PHOG} are the normalized VLAD, BOF, and PHOG descriptors, respectively.

(2) The concatenation of multiple binary visual descriptors of V is represented as

$$B = (B_{VLAD}, B_{BOF}, B_{PHOG}) = (\underbrace{b_1, \ldots, b_{4096}}_{B_{VLAD}}, \underbrace{b_{4097}, \ldots, b_{5096}}_{B_{BOF}}, \underbrace{b_{5097}, \ldots, b_{5516}}_{B_{PHOG}}), \quad (3)$$

where B_{VLAD}, B_{BOF}, and B_{PHOG} are obtained using the method discussed in Section 4.2.

(3) $\mathbf{Q} = (Q_1, \ldots, Q_N)$ denotes the set of sample query images within the region.

(4) $\mathbf{D}$ denotes the set of database images within the region.

(5) $\mathbf{I}_n = [I_n^1, I_n^2, \ldots, I_n^R]$ denotes the manually applied labels of correct matches to the sample query Q_n within $\mathbf{D}$.

(6) $\mathbf{M}_n = [M_n^1, M_n^2, \ldots, M_n^S]$ denotes the set of mismatches to the sample query Q_n. To obtain $\mathbf{M}_n$, we first search the set $\mathbf{D}$ by using B_n (the concatenated multiple binary descriptors of Q_n) to get a result set, and then we remove the correct matches (images in set $\mathbf{I}_n$) from the result set to form $\mathbf{M}_n$.

Using these definitions, the detailed boosting-based fusion algorithm selects a set of discriminative dimensions from the vector space of B. The idea is to evaluate the ranking loss of correct matches and mismatches of all the sample queries. Since our algorithm is performed on binary vectors by using fused binary vectors, the Hamming distance is adopted to rank the search results.

We treat each dimension as a weak ranker. Then, in the *k–th* iteration of our algorithm, we select the dimension that minimizes the ranking loss of correct matches:

$$\sum_{n=1}^{N} \alpha_n^k \sum_{r=1}^{R} Rank_H(I_n^r) \cdot D_H(Q_n, I_n^r), \quad (4)$$

where $Rank_H(I_n^r)$ is the current ranking position of I_n^r considering $D_H(\cdot)$, $D_H(\cdot)$ is the Hamming distance computed using the current k selected dimensions from B, and α_n^k is the error weighting of sample query Q_n that produces a dimension selection that benefits the entire set of training images.

Since Hamming distance is used in this process, it may be the case that multiple dimensions output identical minimal ranking loss. We further leverage knowledge from mismatches to eliminate such duplication. Let $\tilde{S}_1$ be the set of such dimensions. We then select the needed dimension from $\tilde{S}_1$ by maximizing the ranking loss of mismatches as

$$\sum_{n=1}^{N} \beta_n^k \sum_{s=1}^{S} Rank_H(M_n^s) \cdot D_H(Q_n, M_n^s), \quad (5)$$

where $Rank_H(M_n^s)$ is the current ranking position of M_n^s considering $D_H(\cdot)$, and β_n^k is the error weighting used to make the dimension-selecting process punishing all mismatches, instead of a small part of mismatches.

If more than one dimension continues to maximize Equation (5), the algorithm instead selects the result dimension from $\tilde{S}_2$ (i.e., the set of dimensions selected from $\tilde{S}_1$ by maximizing Equation (5), which is done by minimizing the ranking loss of correct matches based on Euclidean distance as

$$\sum_{n=1}^{N} \sum_{r=1}^{R} Rank_E(I_n^r) \cdot D_E(Q_n, I_n^r), \quad (6)$$

where $Rank_E(I_n^r)$ is the current ranking position of I_n^r considering $D_E(\cdot)$, and $D_E(\cdot)$ is the Euclidean distance computed using the current dimension selected from the vector space of V.

With the *k–th* dimension selected, we compute α_n^{k+1} and β_n^{k+1} to be used in the $(k+1)$*–th* iteration as

$$\alpha_n^{k+1} = \sum_{r=1}^{R} Rank_H\left(I_n^r\right) \cdot D_H\left(Q_n, I_n^r\right) \tag{7}$$

$$\beta_n^{k+1} = \frac{1}{\sum_{s=1}^{S} Rank_H\left(M_n^s\right) \cdot D_H\left(Q_n, M_n^s\right)}, \tag{8}$$

where $Rank_H(\cdot)$ and $D_H(\cdot)$ are computed using the k dimensions selected in the previous k iterations.

The complete algorithm of the *k–th* iteration in our dimension selection method is summarized as follows:

Step 1: Select a dimension from the vector space of B to minimize the ranking loss of correct matches given by Equation (4).

Step 2: If multiple dimensions give the same minimal ranking loss of correct matches in Step 1, construct a set $\tilde{S}_1$ by using all these dimensions and turn to the next step. Otherwise, go to Step 6.

Step 3: Select a dimension from $\tilde{S}_1$ to maximize the ranking loss of mismatches given by Equation (5).

Step 4: If multiple dimensions give the same maximal ranking loss of mismatches in Step 3, construct a set $\tilde{S}_2$ by using all these dimensions and turn to the next step. Otherwise, go to Step 6.

Step 5: Select a dimension from $\tilde{S}_2$ to minimize the Euclidean distance-based ranking loss of correct matches given by Equation (6).

Step 6: Add the selected dimension to the result set and compute α_n^{k+1} and β_n^{k+1}, which will be used in the $(k+1)$*–th* iteration by using Equations (7) and (8), respectively.

This iteration is performed K times to obtain a fused K dimensional binary descriptor for each database image, and the (initial) values of α_n^1 and β_n^1 are both set to 1.

In the literature, Ji et al. [2012] proposed a boosting-based dimensionality selecting method by minimizing the ranking loss of correct matches. However, the method is not suitable for dealing with binary vectors because it is designed for visual word selection in Euclidean space. Also, it does not consider the case where multiple dimensions may give the same ranking loss. As described earlier, our method can cope with these issues efficiently, finding the optimal dimension from the returned set by penalizing the mismatches (through maximizing Equation (5)), as well as minimizing the Euclidean distance-based ranking loss of correct matches. As shown by the experiments presented in Section 6.2, our method can obviously provide more accurate searching results than Ji's method [Ji et al. 2012].

4.4. Joint Optimization of Feature Fusion and Index Structure

To facilitate on-device MLR, we build an inverted index structure (shown in Figure 2) for each geographical region. We divide each region into 12 groups in practice by considering the heading information (i.e., the direction in which the front of the mobile phone is pointed) from a digital compass to narrow the search range in the online landmark recognition process. For each image, we store the image ID, GPS code, and

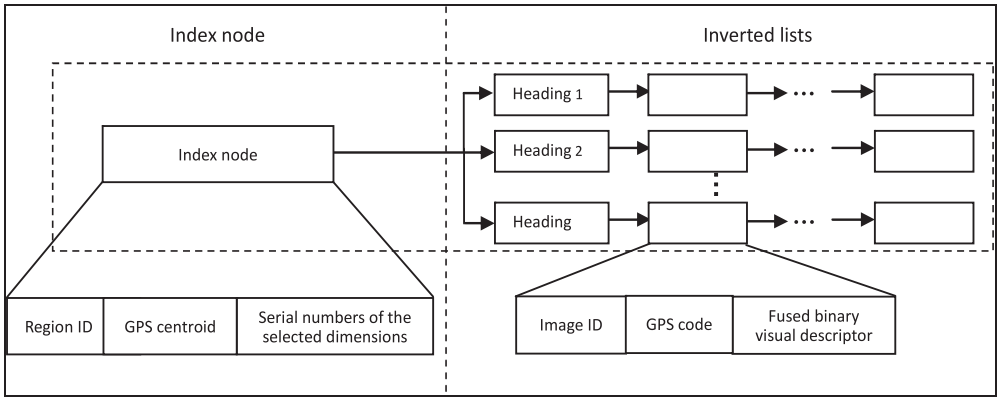


Fig. 2. Index structure of each region.

fused binary visual descriptor in the corresponding inverted list. The GPS code is generated using the RVQ algorithm [Chen et al. 2010; Guan et al. 2013]. The fused binary visual descriptor is generated using the method proposed in Section 4.3.

We now consider how to optimize the number (N_R) of regions to improve recognition accuracy given a fixed length (K) for each fused binary visual descriptor (e.g., $K = 80$ (10 bytes)). In previous works [Guan et al. 2013; Ji et al. 2011], N_R is determined by testing the accuracy of each query image. However, obtaining all the possible queries in advance is usually impractical for real-world MLR. Instead, we determine N_R by using a small set of training images (the training set used in Section 4.3). One possibility is to determine N_R by maximizing the search accuracy of training queries. However, this will lead to the problem of overpartitioning because the number of correct matches to each training query in the corresponding c region will reduce with increasing N_R.

We optimize N_R based on the following observations: If N_R is large, increasing the discriminability of the fused descriptors may improve search accuracy; however, the number of marginal queries will increase, which will decrease the search accuracy. On the other hand, using a small number of regions will reduce the number of marginal queries at the cost of sacrificing the discriminability of the fused visual descriptors.

In our method, we assume that only the nearest region will be searched for each query image in the recognition process. Then, we define the Average Fusion Loss (AFL) of each training query as

$$AFL(Q_n) = \frac{\sum_{r=1}^{\hat{R}_n} D_H\left(\hat{B}_{Q_n}, \hat{B}_{I_n^r}\right) + \tilde{R}_n K}{\hat{R}_n + \tilde{R}_n}, \tag{9}$$

where $\hat{B}_{Q_n}$ is the fused binary descriptor of sample query Q_n, $\hat{B}_{I_n^r}$ is the fused binary descriptor of I_n^r, $(I_n^1, I_n^2, \ldots, I_n^{\hat{R}_n})$ is the set of correct matches of Q_n within the corresponding search region, $D_H(\cdot)$ is Hamming distance, $\tilde{R}_n$ is the number of correct matches of Q_n outside the corresponding search region, and K is the dimensionality of the fused visual descriptor. We adopt Equation (9) to optimize the region selection of N_R to achieve the best recognition accuracy with respect to N_R. In the numerator of $\sum_{r=1}^{\hat{R}_n} D_H(\hat{B}_{Q_n}, \hat{B}_{I_n^r}) + \tilde{R}_n K$, the part $\sum_{r=1}^{\hat{R}_n} D_H(\hat{B}_{Q_n}, \hat{B}_{I_n^r})$ is used to force descriptors from similar views of this region to be as small as possible during optimization. Basically speaking, the smaller the region, the small this value is, which corresponds to lower loss. $\tilde{R}_n K$ serves as a penalty term to avoid the case of selecting the smallest regions

possible. This reduces the possibility that descriptors or queries fall in the margins. In sum, the numerator controls the number of regions, while the denominator $\hat{R}_n + \tilde{R}_n$ serves overall more like a regularizer.

With the AFL of each sample query defined, we optimize the feature fusion and index structure jointly by selecting a value of N_R that minimizes

$$\frac{\sum_{n=1}^{N_T} AFL(Q_n)}{N_T}, \tag{10}$$

where N_T is the total number of sample queries.

5. INTERACTIVE DESIGN

This section discusses the problem of the interactive design of our system. Most existing MLR systems only provide a single kind of user interaction, such as capturing a query image of the target landmark. However, due to the real-world diversity of locations, the captured query image often contains irrelevant content (such as road, sky, and trees). Thus, it is difficult to understand a user's intention by directly using the captured image for recognition. In view of this, Sang et al. [2013] introduced four modes of gesture-based interactions (*crop, line, lasso*, and *tap*) and proved that *lasso* (inspired by the Lasso selection tool for interactive Bing search on iPad) is the most natural and effective interaction mode. Although promising, there are some problems. For example, it is inconvenient to use lasso to segment objects with complex contours (the third row of Figure 3(a) and Figure 3(b)) or objects obscured by plants (the first two rows of Figure 3(a) and Figure 3(b)) since the process is tedious, and existing touchscreen technologies cannot support such a precise response [Sang et al. 2013]. Moreover, when dealing with hollow objects (the last two rows of Figure 3(a) and Figure 3(b)), *lasso* cannot eliminate the irrelevant scenes efficiently.

In view of these issues, we design an efficient interface using interactive image segmentation (Figure 3(c)) to enable users to segment interested objects in a natural and engaging way. As shown in Figure 4, to segment a foreground object, a few lines are marked manually through the mobile device's touchscreen. The red and green lines indicate the foreground/background markers, respectively. The segmentation process begins once the user clicks the "Segment" button after drawing each marking line. If the segmentation result is not satisfactory, the user can add more lines and re-click the "Segment" button to obtain more precise segmentation. The segmentation algorithm used in our system is the interactive graph-cut algorithm proposed in Tian et al. [2009]. We do not discuss the detailed segmentation algorithm in this article, and we advise readers to refer to Tian et al. [2009] and Boykov et al. [2001] for more details. The design of our interface is inspired by PC-based interactive image segmentation systems [Li et al. 2004; Tian et al. 2009]. In these systems, the left and right mouse buttons are used to mark the lines that indicate the foreground and background, respectively. However, we added two buttons (i.e., "Foreground" and "Background") to our system because we can only use touchscreens as the interactive tools for smartphones.

The detailed algorithm is as follows:

Step 1: Capture a query image, segment the target object using the interface introduced in Section 5.

Step 2: Find a candidate search region by using the GPS of the query image, then return four inverted lists for the candidate region by considering the heading information.

Step 3: Extract the VLAD, BOF, and PHOG descriptors of the target object and compute the corresponding binarized descriptors using Equation (1).

(a) (b) (c)

Fig. 3. Illustration of Lasso and interactive image segmentation.

Fig. 4. Landmark recognition process of our prototype system.

Step 4: Select K dimensions from the concatenated multiple binary visual descriptors of the target object according to the serial numbers stored in the index node of the found region. Find a set ($\mathbf{S}_1$) of candidate images from the four candidate inverted lists (returned in Step 1) by using Hamming distance.

Step 5: Reconstruct the GPS value of each candidate image in set $\mathbf{S}_1$ by using the corresponding GPS code. Re-rank set $\mathbf{S}_1$ by computing the distance between the query object and each candidate image in $\mathbf{S}_1$ as

$$Dis = \frac{D_{GPS}}{D_{GPS-\max}} + \frac{D_H}{D_{H-\max}}, \tag{11}$$

where D_{GPS} is the location distance computed using GPS, D_H is the visual distance computed using the fused visual descriptors and Hamming distance, and $D_{GPS-\max}$ and $D_{H-\max}$ are the maximum location and visual distance between the query object and the candidate images in $\mathbf{S}_1$, respectively.

6. RESULTS

We built a prototype MLR system using the proposed algorithms to enable users to perform landmark recognition on a city scale. The Wuhan dataset [Guan et al. 2013] is used, which contains about 1.295 million images. The GPS, gravity, and heading information of each image is recorded using GPS and IMU, respectively. The database also provides a query set in which 680 queries were captured by different people several months after the database was built, and another 169 queries were collected from Google street view. We further collected another set of 1k images to expand the query set. Thus, the query set currently contains about 1.84K test queries, corresponding to about 300 landmarks. The dataset and the expanded query set are publicly available at: http://media.hust.edu.cn/guantao/guantao.html, and readers can get downloading information (FTP site) by contacting (via email) the authors of this article. We only store the image thumbnail to be shown as the search result; we do not store the original images.

To build the index structure, we divide the whole geographical location into 608 regions (according to the results given in Section 6.3). For each region, one-third of the sample queries are selected to form the training set. The foreground of each selected sample query image is segmented using Tian's method [Tian et al. 2009]. We generate the VLAD, BOF, and PHOG descriptors of each sample query image by using the segmented foreground and then perform the fusion algorithm introduced in Section 4.3 in different regions individually to obtain a 10-byte visual descriptor for each image. These fused descriptors are then inserted into the index structure (Figure 2) for on-device MLR.

The landmark recognition process is illustrated in Figure 4. A user captures a query image and segments the foreground using the interface introduced in Section 5. Then, the landmark recognition method given in Section 6 is performed to search a result landmark from the database. Finally, the related information is shown to the user for browsing.

Several experiments are carried out in this section to prove the effectiveness of the proposed algorithms. We measure the performance of different algorithms using *recall@R*, which is defined as the ratio of query images for which the correct match is ranked within the top R returned results. It is worth noting that the recall rate is one of the most frequently used criterion for retrieval and recognition. In our case, since online user interaction is further involved, users can simply browse the search results (as lists of snapshots) to check whether the system has return their target of interest.

From this point of view, it is more important to ensure that true results are within the returned list. Therefore, we adopt the recall rate for our interactive on-device MLR design.

The Wuhan database and San Francisco database are used in our experiments. Detailed information about the Wuhan database was given in Section 6; some detailed information about San Francisco database [Chen et al. 2011a] is as follows:

The dataset was collected in San Francisco and contains two parts: (i) the perspective Central Images (PCI) set contains approximately 1.06 million images generated from the center of the panoramas, and (ii) the perspective Frontal Images (PFI) set contains approximately 0.638 million images, each of which is generated by shooting a ray through the center of a PCI projection and computing the ray intersection point with the scene geometry. The database provides 803 query images of landmarks in San Francisco taken with several different camera phones by various people several months after the database images were collected.

6.1. Performance of the Proposed Binarization Method

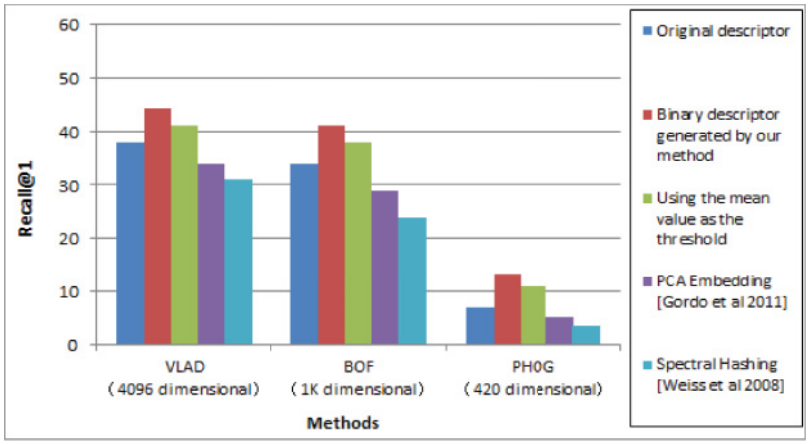
In this section, we discuss the performance of the proposed vector binarization method. We also implement the following three methods for comparison: (i) The PCA embedding method proposed in Gordo et al. [2011], (ii) the Spectral Hashing method proposed in Weiss et al. [2008], and (iii) the method using mean value as the threshold to do binarization. To make a fair comparison, we generate the binary codes with the same length (identical to the dimensionality of input vectors) when using different methods to binarize input vectors. We also divide the whole area into 64 sub-areas by considering the GPS information. In this experiment, we directly use the original query images instead of the segmented ones to test the performance of different methods. The heading information is also ignored.

We tested the performance of different methods on normalized VLAD, BOF, and PHOG, respectively. The results are shown in Figure 5. We can see that the binary descriptor obtained by using the method given in Section 4.2 can obviously improve the recognition accuracy over that of the others. Moreover, the binary descriptors obtained by Equation (1) can provide more accurate search results than those using mean value as the threshold. These results prove the effectiveness of the proposed descriptor binarization method.

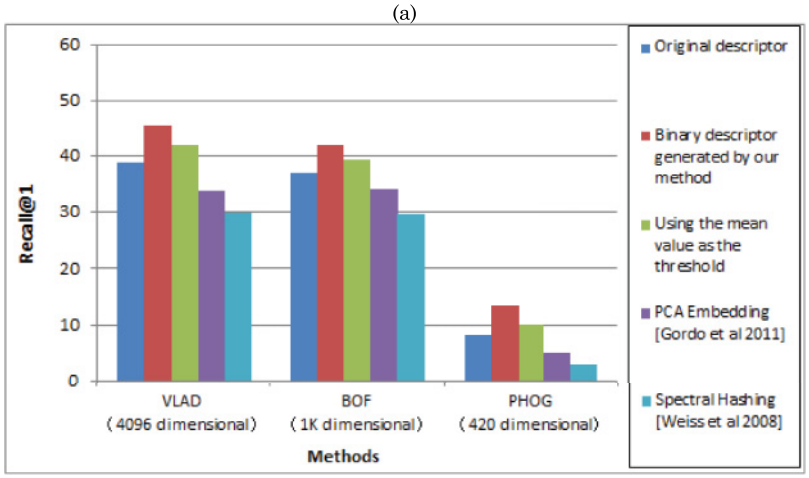
6.2. Performance of the Proposed Feature Fusion Method

We test the performance of the proposed feature fusion method. We divide the whole area (geographical location) into 64 areas. For each region, one-third of the sample queries are selected to form the training set, and the others will be used for validation. We use the method proposed in Section 4.3 to fuse multiple binary visual descriptors extracted in the respective regions. For each query image, we use GPS to find a candidate region and then obtain the search results by computing the Hamming distances between the fused binary visual descriptors. We also test the performance of the following strategies for comparison use: (i) Boosting and searching different kinds of binary visual descriptors individually—the searching results are obtained using Hamming distance; (ii) using Ji's method [Ji et al. 2012] to boost the concatenation of multiple normalized visual descriptors (Equation (2)); and (iii) boosting and searching binary VLAD and BOF descriptors using our method (the search results are obtained using Euclidean distance). To make a fair comparison, we do not use the heading information in these methods. Moreover, the training and searching processes are performed using the original query images instead of the segmented ones.

The results are shown in Figure 6, from which we can draw the following conclusions: (i) the boosting of multiple features can achieve better accuracy than boosting a single



(a)



(b)

Fig. 5. Search accuracy of the proposed binarization method; (a) and (b) are the results on Wuhan and San Francisco PFI datasets, respectively.

kind of feature; and (ii) compared with Ji's method [Ji et al. 2012], our method not only reduces memory usage by a factor of 32, but it also improves search accuracy significantly. These results demonstrate the effectiveness of the proposed feature fusion method. Also, Figure 5 shows the performance of the proposed binarization scheme.

6.3. Results of Jointly Optimizing Feature Fusion and Index Structure

In this section, we carry out an experiment to prove the effectiveness of jointly optimizing the feature fusion and index structure. We use the Wuhan dataset in this experiment and select one-third of the sample queries for each landmark to form the training set. The method given in Section 6 is used to perform landmark recognition. We record the average AFL (Equation (10)) and search accuracy (training query set and test query set) when we change the number (N_R) of divided regions. In this experiment, the training and search processes are performed using the original query images instead of the segmented ones.

The results are shown in Figure 7: The optimal value of$N_R$returned by minimizing the average AFL can significantly improve the recognition accuracy over the optimal

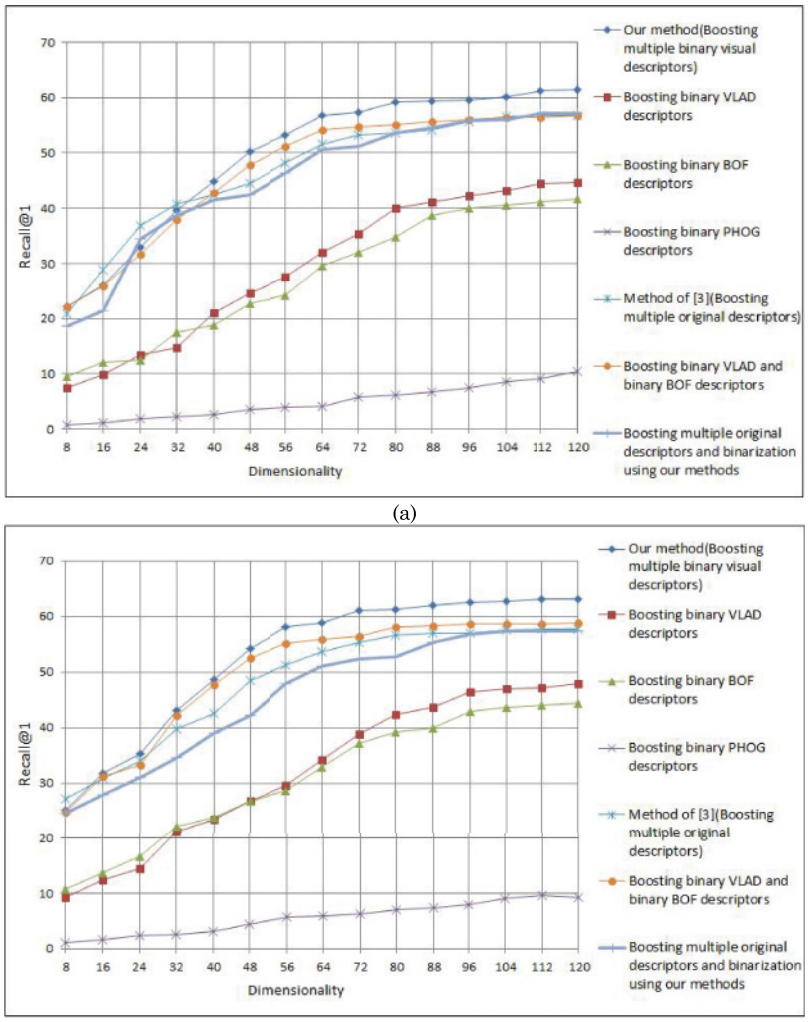


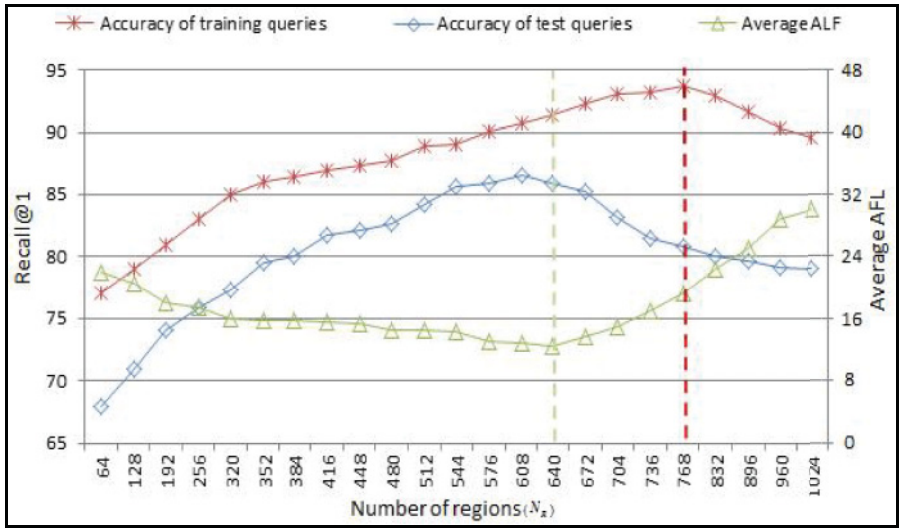
Fig. 6. Search accuracy of the proposed feature fusion method. (a) and (b) are the results on Wuhan and San Francisco PFI datasets respectively.

value obtained by maximizing the search accuracy of training queries. These results demonstrate the effectiveness of combining the feature fusion and index structure optimization scheme.

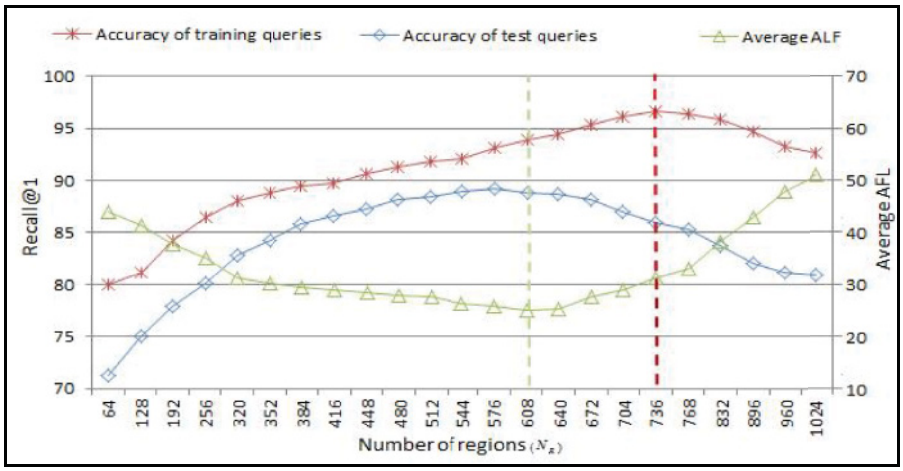
6.4. Comparison with Previous Methods

We carry out an experiment to compare the performance of the proposed multiple-feature–based landmark recognition with previous methods, including:

(1) The late fusion strategy proposed by Chen [Chen et al. 2013]: Both SURF and CHoG features are extracted to generate two different VLAD descriptors for each image. The index structure of Figure 2 is used to index the generated descriptors. The weighted sum fusion method [Chen et al. 2013] is used to fulfill the landmark recognition task.



(a)



(b)

Fig. 7. Results of optimizing feature fusion and index structure jointly: (a) and (b) are the results when we set the length (K) of each fused binary visual descriptor to 48 (6 bytes) and 80 (10 bytes), respectively. The scale on the left is searching accuracy of training queries and test queries. The scale on the right is average AFL. The green and red dotted lines mark the optimal values of N_R returned by minimizing the average AFL and maximizing the search accuracy of training queries, respectively.

(2) The method of Chen et al. [2011a]: We build a vocabulary tree (depth = 6 and branch factor = 10, 64-dimensional SURF features) with 1 million leaf nodes to generate the BOF descriptor of each image. To perform landmark recognition, only those database images that are within 300 meters of the query image will be scored. Finally, a geometric verification (RANSAC with a 2D affine model) process is carried out to refine the recognition results.
(3) Guan's method [Guan et al. 2013]: We use the TC-RVQ method [Guan et al. 2013] to encode each PCA compressed VLAD descriptor (256 dimensional) into 10 bytes and

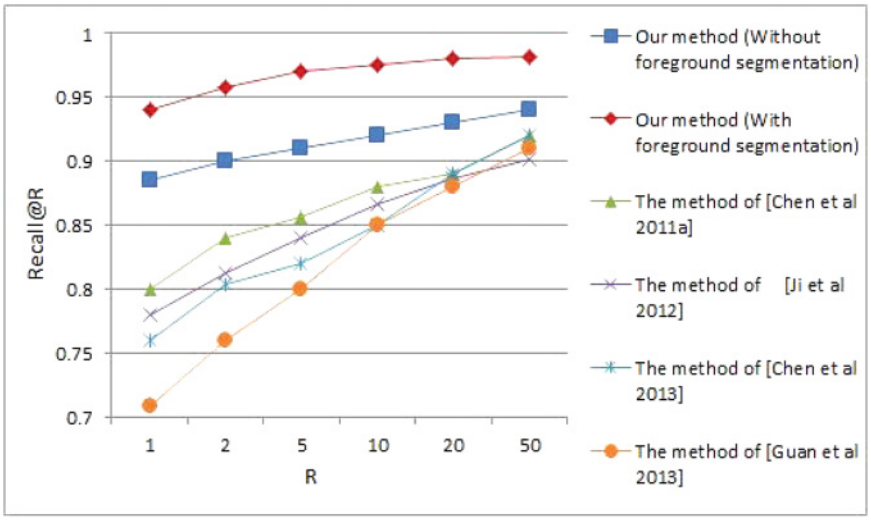


Fig. 8. Comparison with previous methods.

then use the index structure of Figure 2 to index the generated codes. Equation (11) is used to fulfill the landmark recognition task. The visual distance in Equation (11) is obtained by computing asymmetric distance.

(4) Ji's method [Ji et al. 2012]: We use the boosting method proposed in Ji et al. [2012] to compress each 1-million dimensional BOF descriptor to 80 dimensionalities and use the index structure of Figure 2 to index the compressed descriptors for landmark recognition use. Equation (11) is used to fulfill the landmark recognition task. The visual distance in Equation (11) is obtained by computing Euclidean distance.

When using our method, we set the length (K) of each fused binary visual descriptor to 80 (10 bytes) and use the method described in Section 6 to perform landmark recognition. Both the original and segmented query images are used to evaluate the performance improvement of the proposed interaction scheme.

The Wuhan database is used in this experiment, and the whole geographical location is divided into 608 regions to test our method and the other methods [Chen et al. 2013; Guan et al. 2013; Ji et al. 2012]. For each landmark, we randomly select one-third of sample queries as the training set that will be used in our method and Ji's method [Ji et al. 2012]. The heading information is used in different methods to refine the candidate set obtained using GPS information. As shown in Figure 8, our multiple-feature–based landmark recognition method can provide better recognition accuracy than the others. Moreover, compared with searching original query images, searching foreground segmented queries can further improve recognition accuracy. These results prove the effectiveness of our interactive image segmentation and multiple-feature fusion-based landmark recognition method.

Figures 9, 10, and 11 present some exemplar landmark recognition results. Figures 9 and 10 give the results of querying the original images. Figure 11 gives the results of querying the foreground-segmented images. We only compare our method with the methods of Chen [Chen et al. 2011a] and Ji [Ji et al. 2012] because both are more accurate than the others, as shown in Figure 9. From Figures 9 and 10, we can see that our method preserves the ranking precision better than the others even in cases of illumination variations, blur, and occlusion. From Figure 11, we can see that, compared

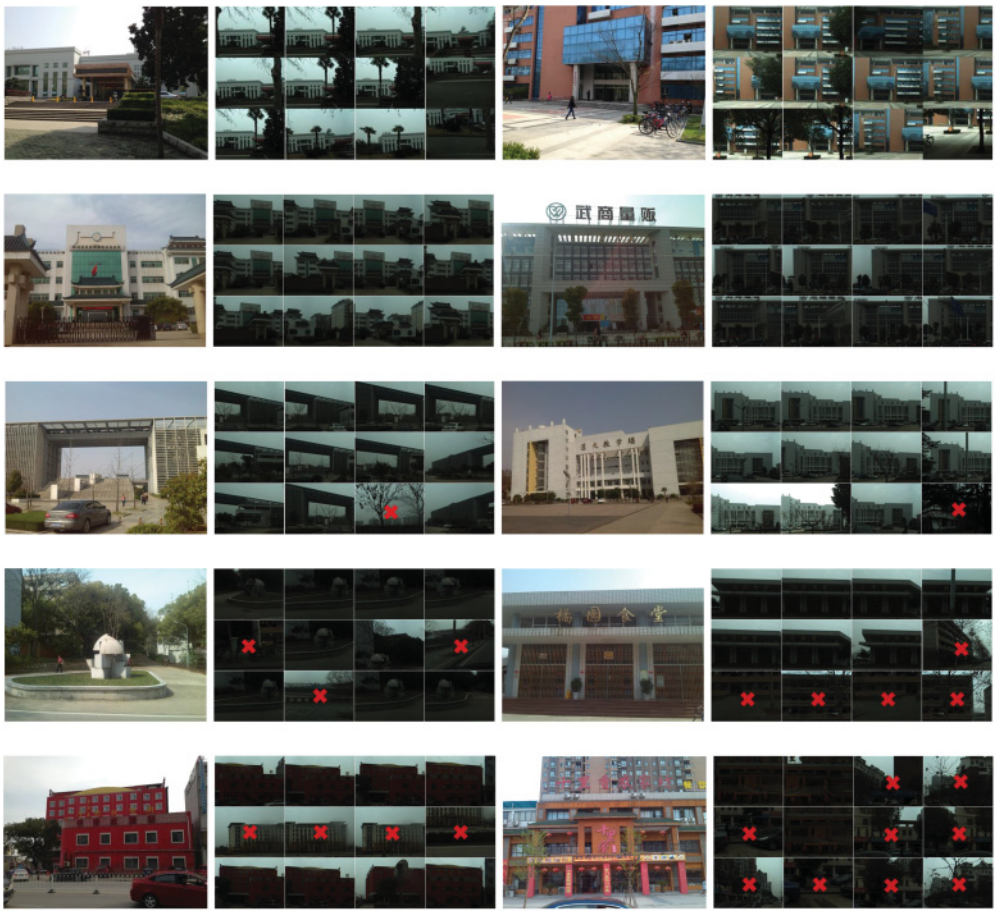


Fig. 9. Landmark recognition results. Each row gives two groups of recognition results. In each group, the query image is shown on the left and each line on the right corresponds to an approach. Top: Our method; Middle: method of Chen et al. [2011a]; Bottom: method of Ji et al. [2012]. The incorrect results are marked with red semitransparent tags.

with the use of original queries, the use of foreground-segmented query images can further improve the ranking precision.

6.5. Computation Timings and Memory Usage

The computation time of the proposed algorithms is recorded in Table I. Our method can fulfill multifeature–based landmark recognition within 1.71s. The computation time of our method is slightly longer than that (1.3s) of the method proposed in Guan et al. [2013]. This is mainly because we need to spend about 0.5s to generate the BOF and PHOG descriptors. In our current system, no optimization is done to accelerate image descriptor generation. However, there exists the possibility of accelerating the extraction in our further work, for example by using the Graphical Processing Units (GPU) in mobile devices. Table II further shows the time analysis of using both our method and the lasso-based approach in image segmentation. It is obvious that by using our approach, the time complexity can be largely reduced.

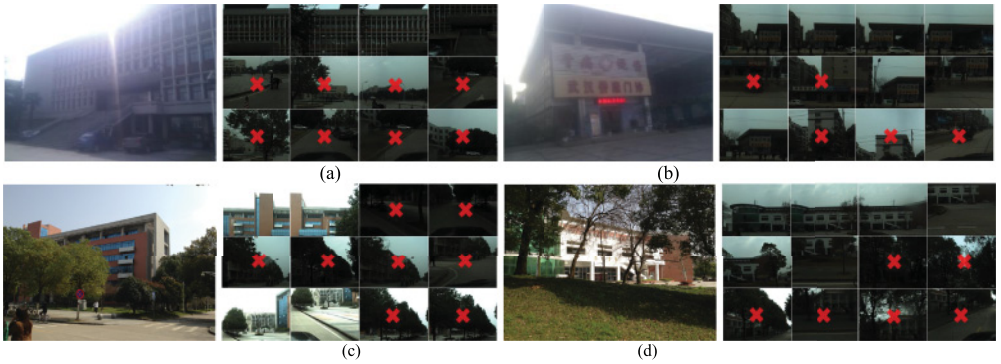


Fig. 10. Landmark recognition results in case of illumination variations, blur as well as occlusion: (a) the results in a case of illumination variations, (b) the results in a case of image blur, (c) and (d) are the results in cases of occlusion. In each group, the query image is shown on the left, and each line on the right corresponds to an approach. Top: Our method; Middle: Chen's method [Chen et al. 2011a]; Bottom: Ji's method [Ji et al. 2012]. The incorrect results are marked with red semitransparent tags.

We also record the memory usage of our method. The index nodes take about 108K bytes (608 index nodes, each region ID takes 2 bytes, the serial numbers take $80 \times 2 = 160$ bytes, each GPS centroid takes 8 bytes, and the 12 heading nodes take 12 bytes). Each image descriptor takes 15 bytes (10 bytes for fused visual descriptor, 3 bytes for image ID, and 2 bytes for GPS code). The vocabulary tree used for BOF generation takes 271K bytes. Therefore, our image searching engine can load the Wuhan database (1.295 million images) into the RAM of a mobile device at the cost of about 18.87M bytes. As a comparison, the Vocabulary Tree (with 1 million leaf nodes built using 64-dimensional SURF features) together with its inverted indexing files used in the methods of Chen et al. [2011a] and Ji et al. [2012] will consume about 270 M bytes of memory, which makes these two methods not suitable for on-device MLR applications. Note that the proposed method costs 1.71s to fulfill one query, which is slower than our previous work (1.3s). With BOF and PHOG descriptors, the proposed method is slightly faster than the previous work, which is mainly due to some optimization in our implementation. From the engineering side, we have reimplemented both the BoF and PHoG descriptors. We accelerate several places, including gradient calculating, histogram calculation, and sharing gradient maps among descriptors.

6.6. Usability Evaluation

We also carry out an experiment to evaluate the usability of the interaction method introduced in Section 5 by comparing it with the lasso method [Sang et al. 2013]. We selected 16 landmarks from the Wuhan database and divided them into four groups, as shown in Figure 12. We posted the images of 16 landmarks on the walls around our testing room. Users were asked to perform landmark recognition by capturing the images of these colorful pictures.

We recruited 100 users with no previous knowledge of MLR to test the usability of the designed prototype. For each user, we run two tests: The first is to perform MLR using the lasso method, and the other is to perform MLR using our interactive method. After each group of tests, the users were asked to fill in the questionnaires shown in Table III as the quantitative feedback on usability. Users were asked to evaluate the different methods from three perspectives: *ease of operation, clear search intention,* and *flexibility*.

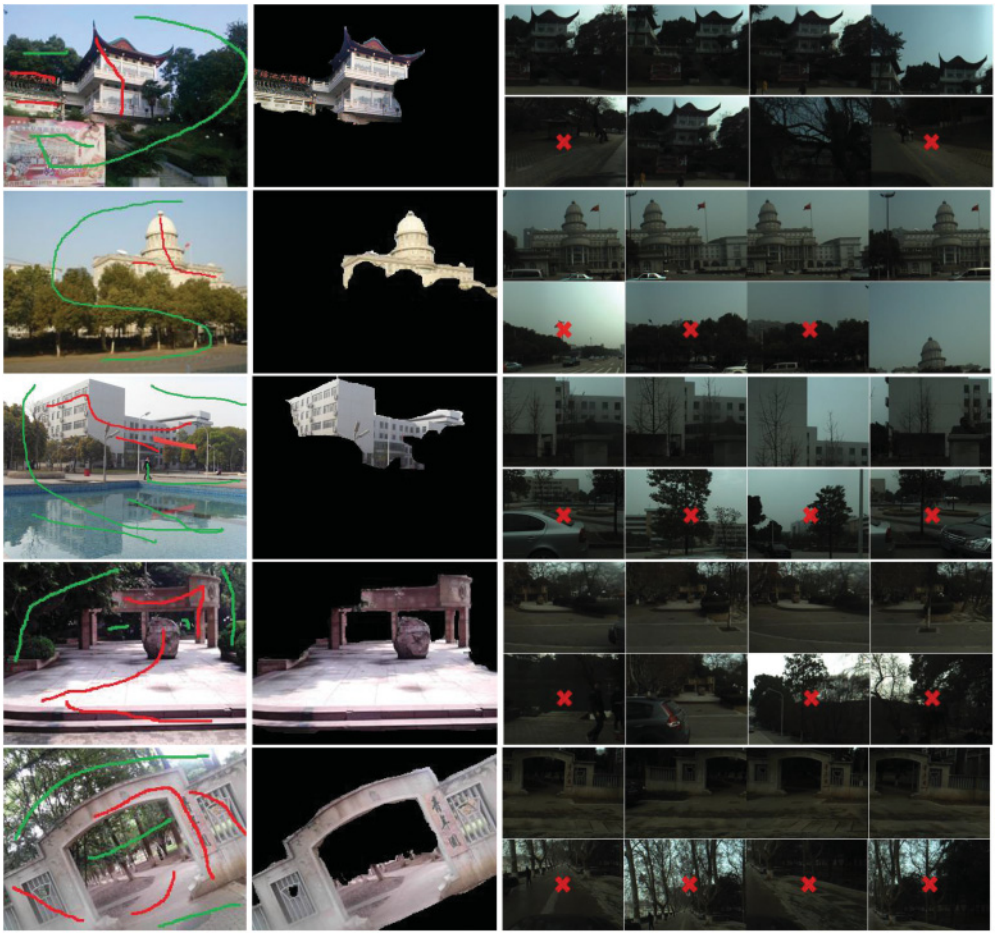


Fig. 11. Landmark recognition results when using foreground segmented queries. Each row gives one group of recognition results. In each group, the marked query and foreground segmented images are shown on the left, and each line on the right corresponds to an approach. Top: The results of using our method to search foreground segmented query images. Bottom: The results of using our method to search original query images. The incorrect results are marked with red semitransparent tags.

In our user study, we removed indicators that identified our algorithm and the lasso algorithm. In addition, the user was given no background information on either the approach in Sang et al. [2013] or our approach. Also, in this indoor evaluation, we predefined the GPS location as the actual GPS location of the targeted landmark. Note that even with the GPS tag, it is still quite coarse at geographical scale, which therefore required us to do another round of similarity ranking to search the actual query. For example, a location might contain several surrounding landmarks.

We recorded the feedback of each user, and we give the average scores in Figure 13. We can see that our interface receives better responses compared with the lasso method. Some users' comments illustrate the obtained scores:

—*"It's really difficult to segment the objects with irregular shapes or obscured by plants by using lasso since the touchscreen is not precise enough to response my input."*

Table I. Computation Time on a HTC Mobile Phone with a 1GHz Processor

Step	Time(ms)
SURF features detecting and local descriptors generating (64 dimensional)	~1,100
VLAD generating	~40
BOF generating	~120
PHOG generating	~400
Feature fusion and location recognition	~50
Total	**~1,710**

Table II. Average Time Evaluation for Object Segmentation

	Lasso	Our Interface
Group 1	11.5	8.3
Group 2	21.7	10.4
Group 3	29.2	11.7
Group 4	19.3	10.9

Fig. 12. Query images used in usability evaluation. Each row gives one group of query images. The first row gives query images with regular foregrounds. The second row gives query images with irregular foregrounds. The third row gives query images with landmarks obscured by plants. The fourth row gives query images with hollow objects.

Table III. Questionnaire Used in Our Experiment

Landmarks		Score of Lasso			Score of our Interface		
		Ease of operation	*Clear search intention*	*Flexibility*	*Ease of operation*	*Clear search intention*	*Flexibility*
Group 1: Regular foregrounds	*Landmark 1*						
	Landmark 2						
	Landmark 3						
	Landmark 4						
Group 2: Irregular foregrounds	*Landmark 5*						
	Landmark 6						
	Landmark 7						
	Landmark 8						
Group 3: Landmarks obacured by plants	*Landmark 9*						
	Landmark 10						
	Landmark 11						
	Landmark 12						
Group 4: Hollow landmarks	*Landmark 13*						
	Landmark 14						
	Landmark 15						
	Landmark 16						
Guidance: Please give your score (0∼100) after each group of test							

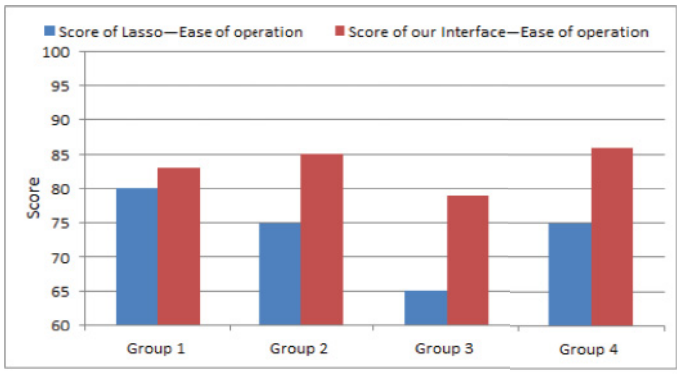
—*"Lasso is too restrictive. I cannot mark the exact contours of the foregrounds with irregular shapes."*
—*"It is convenient to use lasso to segment objects with regular shapes. However, interactive foreground segmentation is more effective in dealing with the other three cases."*
—*"Interactive foreground segmentation is more attractive since it can deal with all the cases conveniently."*
—*"Interactive foreground segmentation itself is a very interesting function. It enables me to express my search intent effectively in dealing with all the landmarks."*
—*"It is flexible to use interactive image segmentation to express my search intent since I do not need to care about the exact contours of the foregrounds."*

Table IV(a) further shows the variation of user evaluation scores for our approach. It is obvious that the user scores are indeed consistent, which means that our approach can achieve consistently high evaluation scores from different users.

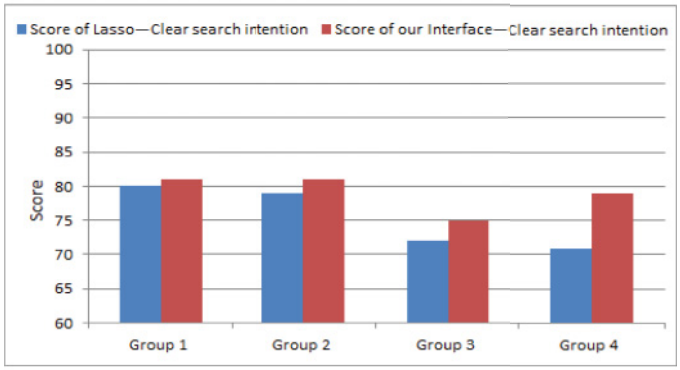
And finally, as shown in Figure 14 and Table IV(b), we have also added a group of additional experiments regarding the comparison between our approach and lasso-based segmentation in the setting of a pen-based user interface (we adopted Samsung Note 3). It is obvious that our method still significantly outperforms the lasso-based method. Finally, for more comparisons of our interaction scheme and pen based interaction scheme, please refer to Figure 14.

Here, for the combination of multifeature fusion and interactive design, we summarize the overall experimental analysis:

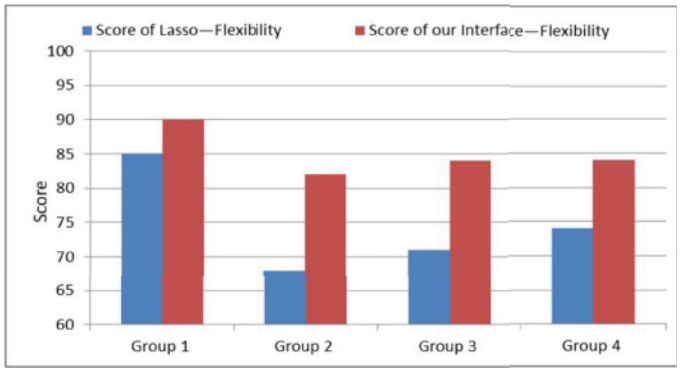
(1) For efficiency, the combination scheme is timely, as shown in Table I and Table II.
(2) For accuracy, by combining feature selection and interactive search interface, we significantly outperform the lasso-based interactive segmentation scheme proposed in Sang et al. [2013], as shown in Table IV.
(3) For comfort level, we also provided several groups of user studies on how well users are satisfied with the proposed combination scheme, as shown in Figure 13 for touch-based and pen-based devices.



(a)



(b)



(c)

Fig. 13. Average scores for the user experience study on HTC mobile phone: (a) scores on ease of operation; (b) scores on clear search intention; (c) scores on flexibility.

Table IV. Accuracy Variation Comparison

(a)

	Ease of Operation		Clear Search Intention		Flexibility	
	Lasso	Our Interface	Lasso	Our Interface	Lasso	Our Interface
Group 1	3.9	3.84	3.45	3.41	3.85	3.81
Group 2	3.91	3.57	3.52	3.51	3.67	3.22
Group 3	3.68	3.36	4.02	3.93	3.52	3.43
Group 4	3.57	3.27	3.54	3.19	3.78	3.64

(b)

	Ease of Operation		Clear Search Intention		Flexibility	
	Lasso	Our Interface	Lasso	Our Interface	Lasso	Our Interface
Group 1	3.78	3.72	3.21	3.09	3.59	3.37
Group 2	3.41	3.12	2.92	2.87	3.08	3.07
Group 3	3.27	3.16	4.33	3.83	4.32	3.89
Group 4	3.44	3.39	4.05	3.92	4.01	3.88

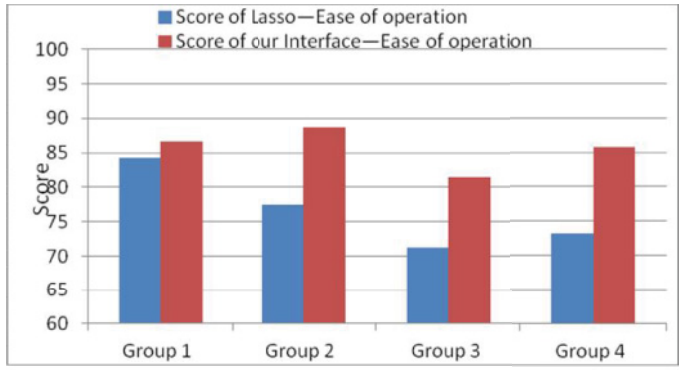
Tables (a) and (b) are the variations on HTC and Samsung Note 3 Mobile Phones, respectively.

7. CONCLUSION

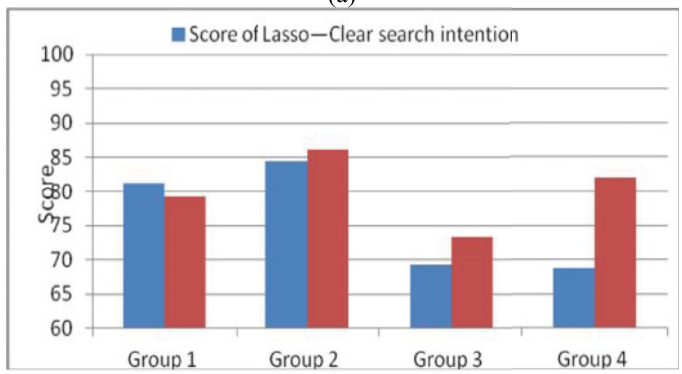
This article targets fusing multiple visual features in a compact manner with an intelligent interactive interface design to enhance the user experience of city-scale on-device MLR. Several innovations are introduced in this article to facilitate multiple visual feature-based MLR : binarizing visual descriptors to reduce memory usage, fusing multiple visual features to get more compact and discriminative image descriptors, and optimizing feature fusion and indexing jointly to obtain accurate landmark recognition. We also prove that the use of interactive foreground/background segmentation can improve recognition accuracy and the user experience significantly. Experimental results demonstrate the effectiveness of the proposed methods.

Some issues that should be further studied are discussed here:

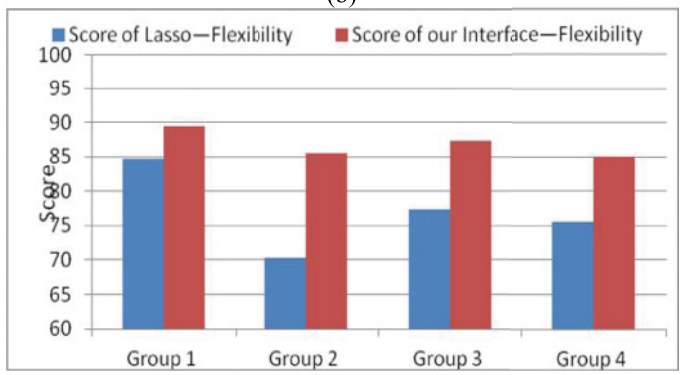
(1) In our method, we use sample query images to facilitate the processes of multiple features fusion and index structure optimization. It may be feasible to use preselected images, which can be obtained conveniently from resources such as Flickr, Facebook, and Google Earth. As new landmarks are added into the system, feature fusion and index optimization has to be rerun to ensure good recognition accuracy. In our future work, we will design a method to enable the system to incrementally accommodate new landmarks efficiently.
(2) More features can be added to further improve the accuracy of MLR. However, it is noted that the response time will deteriorate with increasing features. For example, we tested the search accuracy when adding BOC [Wengert et al. 2011] descriptors, in which we only obtain less than 1.5% improvement in accuracy (on both the San Francisco and Wuhan databases) at the cost of increasing the response time by about 10%. This might due to the color difference introduced by different imaging devices. In our future work, we will try to develop a set of evaluation criteria to determine the optimal selection of new features.
(3) This research focuses on searching multiple visual features directly on mobile devices in a fully unsupervised manner. To design a practical MLR system, such supervision is in turn encouraged. In our future work, we will also test leveraging such information with the help of cloud storage and computing techniques. In this way, a mobile user can first perform city-scale landmark recognition directly on a



(a)



(b)



(c)

Fig. 14. Experimental comparison when using pen-based user interface (Samsung Note 3); (a) scores on ease of operation; (b) scores on clear search intention; (c) scores on flexibility.

mobile device, then corresponding information on the recognized landmark can be further downloaded to provide additional location-based services.

REFERENCES

G. Baatz, K. Koeser, D. Chen, R. Grzeszczuk, and M. Pollefeys. 2010. Handling urban location recognition as a 2D homothetic problem. In *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*.

H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. 2008. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)* 110, 3, 346–359.

C. Biancalana, F. Gasparetti, and A. Micarelli. 2013. An approach to social recommendation for context-aware mobile services. *ACM Transactions on Intelligent Systems and Technology* 4, 1.

A. Bosch, A. Zisserman, and X. Munoz. 2007. Representing shape with a spatial pyramid kernel. In *CIVR*.

Y. Boykov and M. P. Jolly. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of ICCV 2001*.

J. Brandt. 2010. Transform coding for fast approximate nearest neighbor search in high dimensions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 1815–1822.

V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod. 2011. Compressed histogram of gradients: A low bitrate descriptor. *International Journal on Computer Vision* 94, 5, 384–399.

D. Chen, G. Baatz, K. Koeser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. 2011a. City-scale landmark identification on mobile devices. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 737–744.

D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod. 2013. Residual enhanced visual vector as a compact signature for mobile visual search. *Signal Processing* 93, 8, 2316–2327.

T. Chen, K. H. Yap, and L. P. Chau. 2011b. Integrated content and context analysis for mobile landmark recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 10, 1476–1486.

Y. Chen, T. Guan, and C. Wang. 2010. Approximate nearest neighbor search by residual vector quantization. *Sensors* 10, 12, 11259–11273.

M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the 20th Annual Symposium on Computational Geometry*. 253–262.

A. Dey, J. Hightower, E. de Lara, and N. Davies. 2010. Location-based services. *IEEE Pervasive Computing* 9, 1, 11–12.

M. Douze, A. Ramisa, and C. Schmid. 2011. Combining attributes and Fisher vectors for efficient image retrieval. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 745–752.

J. A. Fails and D. R. Olsen. 2003. Interactive machine learning. *ACM IUI*.

B. Fernando, E. Fromont, D. Muselet, and M. Sebban. 2012. Discriminative feature fusion for image classification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 3434–3441.

P. Gehler and S. Nowozin. 2009. On feature combination for multiclass object classification. In *Proceedings of the IEEE International Conference on Computer Vision*. 221–228.

Y. C. Gong and S. Lazebnik. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 817–824.

A. Gordo and F. Perronnin. 2011. Asymmetric distances for binary embeddings. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 729–736.

T. Guan, Y. F. He, J. Gao, J. Z. Yang, and J. Q. Yu. 2013. On-device mobile visual location recognition by integrating vision and inertial sensors. *IEEE Transactions on Multimedia*.

H. Jegou, M. Douze, and C. Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1, 117–128.

H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. 2012. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 9, 1704–1716.

R. R. Ji, L. Y. Duan, J. Chen, H. X. Yao, Y. Rui, S. F. Chang, and W. Gao. 2011. Towards low bit rate mobile visual search with multiple-channel coding. In *Proceedings of the ACM International Conference on Multimedia*. 573–582.

R. R. Ji, L. Y. Duan, J. Chen, H. X. Yao, J. S. Yuan, and Y. W. Rui. Gao. 2012. Location discriminative vocabulary coding for mobile landmark search. *International Journal of Computer Vision* 96, 3, 290–314.

R. R. Ji, Y. Gao, W. Liu, X. Xie, Q. Tian, and X. L. Li. 2014. When location meets social multimedia: A comprehensive survey on location-aware social multimedia. *ACM Transactions on Intelligent System and Technology* 6, 1.

R. R. Ji, H. X. Yao, Q. Tian, P. F. Xu, X. S. Sun, and X. M. Liu. 2012. Context-aware semi-local feature detector. *ACM Transactions on Intelligent System and Technology* 3, 3.

D. Kurz and S. Benhimane. 2011. Inertial sensor-aligned visual feature descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Y. Li, J. Sun, C. K. Tang, and H. Y. Shum. 2004. Lazy snapping. *ACM Transactions on Graphics* 23, 3, 303–308.

H. Liu, T. Mei, J. B. Luo, H. Q. Li, and S. P. Li. 2012a. Finding perfect rendezvous on the go: Accurate mobile visual localization and its applications to routing. In *Proceedings of the ACM Multimedia (ACM MM)*.

N. N. Liu, E. Dellandrea, C. Zhu, C. E. Bichot, and L. M. Chen. 2012b. A selective weighted late fusion for visual concept recognition. In *Proceedings of the 12th International Conference on Computer Vision (ECCV'12)*. 426–435.

W. Min, C. Xu, M. Xu, X. Xiao, and B. Bao. 2014. Mobile landmark search with 3D models. *IEEE Transactions on Multimedia* 16, 3, 623–636.

D. Nister and H. Stewenius. 2006. Scalable recognition with a vocabulary tree. In *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 2161–2168.

M. Raginsky and S. Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *Proceedings of the Conference on Neural Information Processing Systems*. 1509–1517.

C. Rother, V. Kolmogorov, and A. Blake. 2004. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23, 3, 309–314.

J. T. Sang, T. Mei, Y. Q. Xu, C. Zhao, C. S. Xu, and S. P. Li. 2013. Interaction design for mobile visual search. *IEEE Transactions on Multimedia* 15, 7, 1665–1676.

G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach. 2011. Mobile visual location recognition. *IEEE Signal Processing Magazine* 28, 4, 77–89.

J. Sivic and A. Zisserman. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*. 1470–1477.

J. Song, Y. Yang, Z. Huang, H. Shen, and R. Hong. 2011. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the ACM International Conference on Multimedia*. 423–432.

Y. Tian, T. Guan, C. Wang, L. J. Li, and W. Liu. 2009. Interactive foreground segmentation method using mean shift and graph cuts. *Sensor Review* 29, 157–162.

S. Tsai, D. Chen, H. Chen, C. H. Hsu, K. H. Kim, J. P. Singh, and B. Girod. 2011. Combining image and text features: A hybrid approach to mobile book spine recognition. In *Proceedings of the ACM International Conference on Multimedia*. 1029–1032.

K. Y. Tseng, Y. L. Lin, C. Y. Hsiu, and W. H. Hsu. 2012. Sketch-based image retrieval on mobile devices using compact hash bits. In *Proceedings of the ACM International Conference on Multimedia*. 913–916.

T. Wang et al. 2013. TouchCut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding* 120, 14–30.

Y. Wang, T. Mei, J. D. Wang, H. Q. Li, and S. P. Li. 2011. JIGSAW: Interactive mobile visual search with multimodal queries. In *Proceedings of the ACM International Conference on Multimedia*. 73–82.

Y. Weiss, A. Torralba, and R. Fergus. 2008. Spectral hashing. In *Advances in Neural Information Processing Systems*. 1–8.

C. Wengert, M. Douze, and M. Douze. 2011. Bag-of-colors for improved image search. In *Proceedings of the ACM International Conference on Multimedia*. 1437–1440.

S. White, D. Marino, and S. Feiner. 2007. Designing a mobile user interface for automated species identification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (ACM CHI)*. 291–294.

Y. Wu, S. Y. Lu, T. Mei, J. Zhang, and S. P. Li. 2012. Local visual words coding for low bit rate mobile visual search. In *Proceedings of the ACM International Conference on Multimedia*. 989–992.

J. H. Xia, K. Gao, D. M. Zhang, and Z. D. Mao. 2012. Geometric context-preserving progressive transmission in mobile visual search. *Proceedings of the ACM International Conference on Multimedia*. 953–956.

X. Xian, C. Xu, J. Wang, and M. Xu. 2012. Enhanced 3D modeling for landmark image classification. *IEEE Transactions on Multimedia* 14, 4, 1246–1258.

K. H. Yap, T. Chen, Z. Li, and K. Wu. 2010. A comparative study of mobile-based landmark recognition techniques. *IEEE Intelligent Systems* 25, 1, 48–57.

K. H. Yap, Z. Li, D. J. Zhang, and Z. K. Ng. 2012. Efficient mobile landmark recognition based on saliency-aware scalable vocabulary tree. In *Proceedings of the ACM International Conference on Multimedia*. 1001–1004.

Y. Yang, J. K. Song, Z. Huang, Z. G. Ma, N. Sebe, and A. G. Hauptmann. 2013. Multifeature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia* 15, 3, 572–581.

G. N. Ye, D. Liu, I. H. Jhuo, and S. F. Chang. 2012. Robust late fusion with rank minimization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 3021–3028.

T. Yeh and T. Darrell. 2005. Doubleshot: An interactive user-aided segmentation tool. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (ACM IUI)*. 287–289.

W. Zhang, K. Gao, Y. D. Zhang, and J. T. Li. 2011. Efficient approximate nearest neighbor search with integrated binary codes. In *Proceedings of the ACM International Conference on Multimedia*. 1189–1192.

S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Tian Q. 2010. Building contextual visual vocabulary for large-scale image applications. *ACM Multimedia* 501–510

W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. 2010. Spatial coding for large scale partial-duplicate web image search. *ACM Multimedia* 511–520.

Received July 2014; revised May 2015; accepted June 2015