Context-aware vocabulary tree for mobile landmark recognition

Tao Chen*, Shijian Lu, Jiayuan Fan

Institute for Infocomm Research, Agency for Science, Technology and Research, 1 Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632, Singapore

A R T I C L E   I N F O

Article history:
Received 15 October 2014
Accepted 8 May 2015
Available online 16 May 2015

Keywords:
Codeword discrimination learning
Location and direction
Inverted file structure
Vocabulary tree
GPS
Direction
Image scoring
Mobile landmark recognition

A B S T R A C T

This paper presents an effective approach that incorporates contextual information into vocabulary tree learning for mobile landmark recognition. For most existing mobile landmark recognition works, the context information (GPS or direction) is mainly used to reduce the search space in a heuristic and insufficient manner. Some recent work uses the context information for codebook learning but only the GPS information is explored. We propose an effective mobile landmark recognition approach which exploits both context (direction and location) and content information for vocabulary tree learning and image recognition. The proposed approach has two major contributions: (i) it proposes an information gain-based codeword discrimination learning method to evaluate the discriminative capability of each direction-aware codeword; as generated by a context-aware vocabulary tree, and (ii) it develops a context-aware image scoring technique based on an inverted file structure that speeds up the image matching process greatly. Experimental results on the NTU and San Francisco database show that the proposed method can achieve good recognition performance with fast speed.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

In recent years more and more mobile phones are equipped with high-resolution cameras, color displays, and hardware-accelerated graphics. Moreover, many of them are also equipped with GPS devices, and connected to wireless networks. Hence a new class of applications that uses the camera phones to search information about objects in visual proximity has become increasingly popular, e.g., mobile navigation, mobile shopping, mobile product, and landmark search, etc. Among these applications, mobile landmark recognition which uses the camera phone to capture a landmark and find out its related information, is receiving more and more users’ attention and becoming a hot research topic, such as mobile landmark identification [2–4] and recognition [4–10].

Several works that are related to mobile landmark recognition have appeared recently [2, 3, 4, 7–9, 10, 26, 28, 29, 32, 35–37]. In these works, a common scheme is adopted which first extracts local features (e.g., SIFT [16, 31]) from the images, then quantizes these features to visual words, and finally applies methods from text search for image retrieval or recognition [17, 30]. Different from conventional object or location recognition, mobile landmark recognition has its own requirements that need to be considered including: (1) mobile users’ fast response time requirement for the return of the recognition results [26]; and (2) availability of the context information such as GPS or direction data [10, 29] collected from the mobile device. The context information can be utilized to reduce the search space, and further integrated with content analysis (or image feature analysis) to increase the recognition accuracy.

Considering the mobile users’ fast response time requirement, Scalable Vocabulary Tree (SVT) that is proposed in [17] has been widely used for large-scale image retrieval or recognition in [5–7, 9, 11–14, 19, 29] recently. SVT is constructed by performing hierarchical k-means clustering algorithm on a set of local features that are extracted from the images. This method is also known as Tree Structured Vector Quantization (TSVQ). When performing image matching, the descriptors of the query image are quantized through the SVT and a histogram of the node (words) visits on the tree nodes is generated. Candidate images are then sorted according to the similarity between the histograms of these images and the histogram of the query image. The descriptors are treated as a visual “bag-of-words (BoW)” in the SVT matching process.

Considering the availability of the Geo-spatial-based context information collected from the mobile device, various methods have been proposed to incorporate the GPS information into the image recognition process in [6–8, 10, 24, 32, 36, 37]. In [6, 8, 10, 24], they mainly use the GPS data captured by the mobile device to reduce the search space for the query image. Content analysis techniques are then used to refine the search results. In [7], the GPS
data are not only used for image filtering, but also incorporated into the TF-IDF scheme [27] to weight various visual words.

As an observation from the above works, it can be seen that (i) the context information used in these works is limited to GPS alone [6–8,24,32,36,37], whereas GPS often contain large errors in the dense built-up areas. Even though some works [10,29] start to consider direction as a context for landmark recognition, the direction is only used to refine the image search space, which is preliminary. (ii) No context information is considered when using the hierarchical k-means algorithm to generate the codewords and construct an inverted file structure for fast image scoring. In [7,29], the codewords are generated using content analysis alone (hierarchical k-means algorithm), and their inverted file structures also do not consider context information.

In this paper, we propose an effective mobile landmark recognition approach that explores both direction and location information for codebook learning and image matching. The proposed mobile landmark recognition approach has two major contributions:

(i) It develops a context-aware codebook learning approach based on an SVT. A set of direction-aware leaf codewords are first generated by partitioning the leaf nodes of the original SVT, according to the direction information labeled in the features. Compared with the original leaf nodes with no direction information, the new direction-aware leaf nodes (words) are more accurate to describe the images that are captured from a particular direction, and thus produce less quantization errors. The discriminative capability of these direction-aware codewords is then learned through an information gain-based word discrimination learning method that takes the image’s location information into consideration. The discriminative direction-aware codewords are then selected to produce an adaptive codebook for each query image for quantization.

(ii) It proposes a context-aware fast image scoring approach based on an inverted file structure in the SVT. The proposed method first generates an adaptive compact codebook for each query image according to their labeled direction and location information. Tree Structured Vector Quantization using the generated codebook is then performed on this image to yield a histogram. This histogram is then matched with the database images through an inverted file structure in the SVT for recognition. The inverted files are constructed only for the codewords in the generated compact codebook, and also consider both location and direction information for image scoring.

The rest of the paper is organized as follows. A general overview of the proposed mobile landmark recognition framework is presented in Section 2. The context-aware vocabulary tree for mobile landmark recognition is discussed in Section 3, which includes the direction-aware codeword discrimination learning, and context-aware fast image scoring. Quantitative evaluation that includes the experimental results and discussions is given in Section 4. Section 5 concludes the paper with a summary of our findings.

2. Overview of the proposed mobile landmark recognition framework

The developed context-aware mobile landmark recognition framework is illustrated in Fig. 1, which consists of two stages: the offline direction-aware codewords discrimination learning and online context-aware image scoring. During the offline stage, the visual local features such as SIFT are first extracted from the captured image. The context information including the picture location and the camera direction (as captured by GPS and digital compass embedded in the mobile devices) are extracted from the image EXIF header [34]. Hierarchical k-means clustering algorithm is then performed on the local features to construct an SVT. The leaf nodes of the SVT are further partitioned into direction-aware leaf nodes (codewords), according to the images’ direction information. The discriminative capability of these direction-aware codewords is finally learned through an information gain-based method which takes the location information into consideration, for generating an adaptive compact codebook for each image. The inverted files are constructed for each selected direction-aware codeword in the SVT for fast image scoring. During the online stage, local features and context information are first extracted from the query image. The context information (direction and location) is then used to determine a compact codebook for the query image to quantize it into a compact histogram. The histogram is finally transmitted to the server, and further matched with the database images through an inverted file structure for fast recognition. The inverted files are constructed only for the codewords in the generated compact codebook, and also consider both location and direction for context-aware image scoring.

3. Context-aware vocabulary tree

This section gives a detailed description to the proposed mobile landmark recognition approach based on a context-aware vocabulary tree, which consists of two steps: (i) Direction-aware codeword discrimination learning and (ii) context-aware fast image scoring.

3.1. Direction-aware codeword discrimination learning

In this section the direction information is first used to generate a set of direction-aware leaf codewords from the original leaf nodes generated by the SVT. The discriminative capability of these direction-aware codewords is then learned for fast image scoring in the online stage. As shown in Fig. 2, the direction-aware codewords generation consists of two steps:

![Fig. 1. Offline and online phases of the proposed mobile landmark recognition framework.](image-url)
(i) Feature decomposition step, which constructs an SVT by performing the hierarchical \( k \)-means clustering algorithm on the set of local features extracted from the images. This ensures the features within the same cluster are visually as close to each other as possible, so that the generated words have good representative capability. The reason for adopting hierarchical \( k \)-means clustering rather than other methods is that the hierarchical \( k \)-means algorithm is not only used to generate the codebook for image quantization, but also produce the SVT to index the inverted files for each codeword (to be discussed in Section 3.2). Note that hierarchical \( k \)-means has been successfully used to construct the SVT for image recognition [1,3,7].

(ii) Direction decomposition step, which generates a set of direction-aware codewords by using direction information to further partition the local features in each leaf cluster of the original SVT. This step is detailed as: for the leaf cluster of the original SVT, each member feature in the cluster is labeled by a direction data according to which image the feature is extracted from. These features are then clustered into various direction clusters based on their labeled direction data. The mean vector of the local features in each direction cluster is used as the direction-aware codewords, e.g., the nodes in the bottom layer as shown in Fig. 2. Here the reason to adopt direction information to further decompose the original clusters is based on the observation that, images that belong to the same landmark and are captured from different directions usually have different visual appearances, as illustrated in Fig. 3. This causes the leaf clusters of the SVT that contain image features from different directions may undergo visual divergence. Generally, for an \( H \)-depth, \( B \)-branch SVT, it produces \( B^H \) codewords [7].

In order to quantize an image into a BoW histogram, a nearest direction cluster is first determined based on the direction data of the image. The codewords in the nearest cluster are then used for image quantization and the rest from other clusters are abandoned. A BoW histogram is thus generated which records the number of each codeword appearing in the image. Through this, less quantization errors are produced, as the codewords in the same direction cluster with the image have better representative capability to describe the image compared with the codewords from other clusters. Furthermore, when matching two images in the same direction cluster, only the codewords in that direction cluster are considered, whereas the codewords from other clusters are ignored. This can greatly reduce the computational cost and fasten the recognition time.

However, one problem still exists for the above image quantization. In particular, it is difficult to determine which direction cluster to use when the image feature's direction lies at or near the boundary of two direction clusters. In such case, it will be prone to errors to use only one direction cluster for image quantization. Considering this, we propose a soft quantization strategy to resolve this problem. The proposed method considers both the direction and visual feature similarity for image quantization. The method first determines several nearest direction clusters by comparing the feature direction with each cluster center, and then assigns soft scores to the direction-aware codeword \( c_i \) in these clusters based on the direction and visual similarity, which is denoted as below,

\[
 h(c_i) = \frac{1}{Q} \left\{ \cos(d_f, i) \cdot \exp \left( -\frac{d_f^2}{\sigma^2} \right) \right\}
\]

where

Fig. 2. The direction-aware codewords generation process.

Fig. 3. Illustration of various landmark images captured from different directions, each row denotes a landmark category, and each column denotes a direction cluster.
are defined as the direction distance and norm-2 visual distance between the \( i \)-th closest codeword \( c_i \) and the feature descriptor \( f \), respectively. \( \theta(f) \) and \( \theta(c_i) \) denote the direction of \( f \) and \( c_i \), respectively. \( v_i(c) \) is the vector of \( c_i \). \( h(c_i) \) is the final soft score assigned to \( c_i \). \( K \) is the number of selected direction clusters for soft quantization, and its value will be investigated in the experiment part. Here Cosine and Gaussian functions are used to calculate the direction and visual similarity between \( f \) and \( c_i \). \( \sigma \) is a scaling factor to smooth the Gaussian normalization function. \( Q \) is used to normalize the final soft score within \([0, 1]\).

Next through performing the proposed soft quantization strategy on the training images, the occurrence number of each direction-aware codeword in each landmark category can be obtained. We then learn the direction-aware codeword’s discriminative capability for each landmark category, to generate an active compact codebook for each query image. The reason to learn the direction-aware codeword’s discriminative capability is that (i) some codewords may not be discriminative enough in their direction clusters as stated in [7,10,18,20,21], and these confusing codewords should be removed, and (ii) the total number of codewords in each direction cluster may be very large, and raises the image’s BoW transmission cost from the mobile client to the server. In this work, the information-gain-based method in [19,23] which effectively determines the dependence between two variables is explored to score each codeword’s discriminative capability for their landmark categories. This is formulated as,

\[
I(l; c) = E(l) - E(l|c)
\]

where \( I(l; c) \) denotes the information gain with respect to the target landmark \( l \) and the target codeword \( c \), \( E(l) \), \( E(l|c) \) correspond to the entropy and conditional entropy of distributions \( P(l), P(l|c) \), respectively. According to the definition of information gain [19], the visual words that yield larger \( I(l; c) \) for landmark \( l \) will be more relevant to this landmark and thus have stronger discriminative capability to describe this category. Here \( E(l) \) is constant across all direction-aware codewords for landmark \( l \), and we only need to consider \( E(l|c) \), which can be calculated as,

\[
E(l|c) = -P(c)(P(l|c) \log(P(l|c)) + P(l) \log(P(l|c))) - P(c)
\]

\[
\times (P(l|c) \log(P(l|c)) + P(l) \log(P(l|c)))
\]

S.t.

\[
I = \{ y \in L : ||y - l||_{\text{geod}} \leq R \},
\]

where \( L \) is the set of all the landmarks located in the direction cluster, \( c \) is the direction-aware target codeword whose discriminative capability is to be estimated and \( c \) is the union of other codewords in the direction cluster. \( P(c) \) refers to the prior probability of \( c \), and can be approximated using the occurrence ratio of \( c \) to all the codewords for the category. \( l \) denotes the target landmark, and \( P(l|c) \) and \( P(l) \) denote the conditional probability of determining landmark \( l \) given \( c \) and \( c \), respectively. They can be approximated using the occurrence ratio of \( c \) (or \( c \)) between the positive and negative categories. \( l \) corresponds to the union of the negative landmark categories whose geographical distance from \( l \) is within a distance of \( R \) in the cluster. \( ||y - l||_{\text{geod}} \) is the geographical distance which is calculated as in [10,22]. Unlike the work [19] which randomly selects several landmarks as the negative landmarks, we only select the negative landmarks that are within a radius of \( R \) from the target landmark for codeword discrimination estimation. This is due to that landmarks often appear to be very small or cluttered by occlusions in the mobile phone when captured from very far away distance. Hence mobile users tend to capture a landmark within a certain distance away from him/her. As a result, the negative landmarks that are geographically close to the target are more likely to be treated as the correct landmark and yield recognition errors. Therefore, these nearby negative landmarks should be given more consideration when learning the codeword discriminative capability.

After obtaining all the direction-aware codewords’ information gain \( I(l; c_m) \), \( m = 1, 2, \ldots, M \) for landmark \( l \), where \( M \) is the number of codewords in the direction cluster, Gaussian normalization function is used to normalize \( I(l; c_m) \) into discriminative scores \( w^l \) for the landmark \( l \):

\[
w^l(m) = \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2} F(l; c_m) \right), \text{ where } m = 1, 2, \ldots, M
\]

which are then used to generate a compact codebook for the query image.

3.2. Context-aware image scoring

In this section, the context-aware image scoring approach is presented. Given a query image tagged with location and direction data, the method first uses the direction and location data to determine a compact codebook for the query image, which is used to encode the local features of the image into a compact histogram. The inverted file structure is then constructed for the codewords in the compact codebook. Context information is finally integrated into the inverted file structure to score the database images fastly. From this process it can be seen that, the most important step is how to adaptively select the discriminative codewords for each query image to encode it and perform context-aware image scoring based on an inverted file structure as elaborated in the ensuing subsections.

3.2.1. Discriminative codeword selection for the query image

The discriminative codewords are adaptively selected based on the direction and location information of the query image. Considering the different kinds of context information in the mobile devices, two situations are considered during the codeword selection process: (i) direction information alone is provided in the query image and (ii) both direction and location are provided in the query image.

In the first case where only direction information is provided, the direction cluster that the image belongs to is first determined using its direction data. The discriminative codewords are selected from this cluster to form a compact codebook \( \Omega \) as follows,

\[
\Omega = \{c_{i} | i = 1, 2, \ldots, M \text{ and satisfy } \mu(w^l(i)) > \xi \}
\]

where \( i \) is the word index, \( M \) denotes the number of direction-aware codewords in the direction cluster, which is defined the same as (8), \( \mu(\cdot) \) is the mean operation to obtain the average word discriminative score and is defined as,

\[
\mu(w^l(i)) = \frac{1}{L} \sum_{l=1}^{L} w^l(i)
\]

where \( L \) is the total number of the landmarks that are located in that direction cluster. From (9) it is seen that the codebook is formed by selecting only the words whose average discriminative scores in that cluster are larger than a threshold \( \xi \). This is based on the fact that larger \( \mu(w^l(i)) \) indicates that the word \( c_i \) has high discriminative scores for most landmark categories in the cluster and the word is thus selected into the compact codebook. The number of selected
words to form the codebook will be investigated in the experimental part.

In the second case where both direction and location data are provided in the query image, the codebook \( \Omega \) is formed for the image as below,

\[
\Omega = \{ c_i | i = 1, 2, \ldots, M \text{ and satisfy } \mu_{\theta}(w^i) > \xi \}
\]

where \( \mu_{\theta}(\cdot) \) is the mean operation that is performed on a set of landmarks which is defined as,

\[
\mu_{\theta}(w^i) = \frac{1}{L_\theta} \sum_{l=1}^{L_\theta} w^i(l)
\]

where \( L_\theta \) is the number of landmarks whose geographical distance from the query image is within a radius of \( \theta \). The reason to use \( \theta \) for selecting candidate landmarks is the same as in Section 3.1.

3.2.2. Context-aware image scoring

After selecting the discriminative codewords for the query image, the local features extracted from the image can be quantized into a compact histogram using these codewords. The server then uses the compact histogram to perform context-aware image scoring for recognition. The inverted files are constructed only for the words in the generated codebook to fastly score the database images.

As illustrated in Fig. 4, for each direction-aware codeword \( c_m \), its inverted file maintains two lists: (i) a sorted array of image IDs \( I_1, I_2, \ldots, I_n \) indicating the database images that have visited that direction-aware codeword, and simultaneously (ii) a corresponding array of counts \( h_{c_m1}, h_{c_m2}, \ldots, h_{c_mk} \) indicating the visit number. When performing a query for image \( X \), a database of \( N \) total images can be quickly scored by traversing only the codewords that are visited by the query descriptors. In this work since location and direction information are provided in the query image, there is no need to match the query with all the database images, which is time-consuming. Instead, a context-aware image scoring and matching approach is proposed, which consists of four steps including:

(i) Firstly, the context analysis employs the image’s GPS information to estimate the user’s roughly standing position, and direction information to select several nearest direction clusters. The two types of information then form a viewcone to shortlist several landmark candidates as shown in Fig. 5. One point that needs to mention here is that the GPS errors may cause the user’s standing position to be incorrectly estimated and this incorrect position drifts a distance away from the true position. As a result, the viewcone generated using the incorrect position will fail to include the correct landmark in the candidate set. In this work to solve this problem, the image’s GPS indicating the user’s standing position is extended backwards by 50 m along the direction where the image is captured. Based on the extended user’s position, a new viewcone is generated to include the correct landmark in the candidate set. The value of 50 m here is selected according to the observed GPS error range in the NTU landmark database.

(ii) Secondly, the preliminary candidates whose positions are within a distance of \( R \) from the captured image are selected as the further candidates (Fig. 5). These candidate landmarks’ images \( I \in \psi \) in the database are then assigned likelihood scores \( \omega \) based on the location and direction analysis,

\[
\omega(I) = \cos(\delta_{X,Y}) \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{1}{2\sigma^2} ||X - Y||_{\text{geodis}}^2 \right)
\]

where \( Y \) is the shortlisted landmark that the image \( I \) belongs to, \( \delta_{X,Y} \) is defined as the angle between the landmark \( Y \) and the direction of the captured image \( X \) which is the same as in \( \delta \). Intuitively, the landmark that is closer to the user and nearer to the captured direction of the query image will have higher degree of likelihood. The Gaussian and cosine functions are used to incorporate the distance and direction information, respectively.

(iii) Thirdly, the image candidates that are selected from the database as described in step (i)-(ii) are assigned soft scores based on the inverted file structure as shown in Fig. 4. For each selected image \( I_m \) in the database, all the codewords \( c_m \in \Omega \) that are visited by both the query image \( X \) and \( I_m \) (in other words, the BoW histograms of \( X \) and \( I_m \) both have non-zero counts at codeword \( c_m \)) are first found. A similarity score \( z \) between \( I \) and \( X \) is then computed by summing their weighted intersection similarity over all the visited codewords together:

\[
z(I_m) = \sum_{c_m \in \Omega} w(m) I_m(I, X) = \sum_{c_m \in \Omega} \mu_{\theta}(w^m) \min(h_{c_mI}, v_X(m))
\]

where \( w(m) \) is the discriminative score of codeword \( c_m \) which is used to emphasize the results calculated for more discriminative words, and can be calculated by \( \mu_{\theta}(w^m) \) as in (12). \( I_m(I, X) \) is the intersection similarity between images \( I_m \) and \( I \) at codeword \( c_m \), \( v_X(m) \), \( h_{c_mI} \) denote the count number of word \( c_m \) appearing in the images \( X \) and \( I_m \), respectively.

![Fig. 4. Illustration of inverted files for the direction-aware codewords in the SVT.](image-url)
(iv) Lastly, the context-aware score $\omega$ and content-aware score $z$ are integrated to give the matching score $s$ for the image $I$ with the query image $X$ as:

$$S(I) = \omega(I)z(I)$$ (15)

The image attaining the highest score is treated as the most similar image to the $X$ and its category is recognized as the category of the query image.

4. Quantitative evaluation

4.1. Experimental setup

To the best of our knowledge, little direction information is available in most public landmark datasets, such as the world-wide geo-tagged dataset [7,24], the San Francisco geo-tagged dataset [3], and the city-scale dataset with GPS tag [19]. We therefore perform evaluation by using the NTU50Landmark database which consists of 4156 images (50 categories x about 80 images/category) that were collected from 50 landmarks within the campus of Nanyang Technological University (NTU) [10]. Among these images, 3622 are used for training, and the rest 534 are used for testing. The images of the database are tagged with both location and direction information, and captured using camera phones with different built-in camera settings. The selected landmarks are well-recognized buildings, structures, and places-of-interests in NTU such as Chinese Heritage Center, and Yunnan Monument and Pavilion, that cover an area of more than 2 km$^2$. Sample images of these landmarks are given in Fig. 6(a) and their geospatial distribution is given in Fig. 6(b). From the figure, it can be seen that the landmarks spread across the whole campus of NTU, with certain areas having a higher concentration of landmarks. The database covers a large variety/complexity of the real world conditions that can be used in the training of the SVT. All images are resized to $320 \times 240$ or $240 \times 320$ pixels.

4.2. Determination of the parameters

In this section, several groups of experiments are conducted to determine the model parameters including the number of selected landmarks.
direction clusters for image soft quantization (Section 3.1) and image candidate pruning (Section 3.2) and the radius of $R$ for compact codebook generation and image candidate pruning (Section 3.2). As shown in [25,31], dense SIFT descriptors that are shown to be effective in describing landmark images are extracted from the image, and then used to generate an SVT. The $\xi$ in (11) is tuned to select the top 30% highly scored direction-aware codewords in each direction cluster (the number “30%” will be discussed later) into the compact codebook. The SVT depth is set as $H = 4$, and branch number $B = 10$.

First, the experimental results of varying the number of selected direction clusters for landmark recognition are given in Table 1, where no location information is used. Several evaluation criteria including the averagely generated codebook size for the query, averagely selected image number for similarity matching, recognition rate, and average recognition time are given. From the table it can be seen that when the number of selected direction clusters is increased, (i) the averagely generated codebook size and candidate image number both increase. This is due to that each direction cluster has its own discriminative codewords and associated images, when more clusters are included, more codewords and images will be selected for recognition, (ii) the recognition rate is first increased to a highest value of 90.3% before the direction cluster number reaches 2, and then decreases after that. This is because that when the number of clusters goes beyond 2, more confusing codewords will be selected into the codebook to describe the image and meanwhile more other categories’ images are included for image matching, which greatly increases the error match chances for the query, and (iii) the recognition time increases monotonically as the candidate image number and codebook size rise. Considering both the recognition accuracy and time, the number of direction clusters is selected as 2.

Second, we set the direction cluster number as 2, and investigate the influence of varying $R$ for landmark recognition. The experimental results are given in Table 2, where the evaluation criterion is the same as in Table 1. It can be seen that the averagely generated codebook size, candidate image number, and average recognition time all increased gradually when the radius $R$ is rising. For the recognition rate, it first increases to a highest value of 93.3% when the radius $R$ reaches 200 m, and then decreases. This is due to that users tend to capture a landmark within 200 m away, and larger radius of beyond 200 m tends to introduce more irrelevant visual words and images which may reduce the recognition accuracy. Further it is noted that when location information is combined with direction information to further refine the search space, the averagely generated codebook size and candidate image number are both reduced greatly (e.g., the codebook size is reduced from 4300 to 2900 and candidate image number is reduced from 700 to 310), with higher recognition rate and faster recognition time (the recognition rate is increased from 90.3% to 93.3% and the recognition time is reduced from 0.06 s to 0.02 s). Considering this, $R$ is set as 200 m.

Thirdly, after determining the direction cluster number and GPS radius, the number of codewords to describe each query image in the database, in conjunction with the number of selected candidate images for image matching are given in Fig. 7(a) and (b), respectively. It is worth noting that in Fig. 7(a) the zero elements in the BoW histogram are removed and thus the number of the rest code-words is smaller than the original codebook size (around 2900). From the figure it can be seen that for query images tagged with different GPS and directions data, the final number of the meaningful codewords to describe the image (averaged size is around 800) and selected candidate images (averaged size is around 300) are also different. Furthermore, it is seen that for the query images with indexes between 210 and 230, the number of selected codewords and candidate images is smaller than that for other images. The reason is that there are few landmarks around the position of these query images, which yield a sparse distribution for both selected codewords and candidate images in these areas. For the query images with indexes between 1 and 200, the number of selected codewords and candidates images is larger than that for other images. The reason is that in the areas corresponding to these query images’ positions, there are quite a few landmarks which yield a dense distribution for both selected words and candidate images. This shows that the proposed mobile landmark recognition is an adaptive process, where the loaded vocabulary and selected candidate images mainly depend on the location and direction data of the specified query image.

Finally, the time complexity for feature extraction including dense SIFT extraction from landmark images and SIFT quantization via context-aware SVT is analyzed as follows. For dense SIFT extraction, the desiror is a collection of information in a $2x \times 2x$ sized image patch. The image patch is divided into $4 \times 4$ regions. In each region the vectors are binned into 8 buckets covering 360 degrees. This requires $O(x^2)$ time for the extraction of SIFT in each image patch. Then for a $M \times N$ dimensional whole image, it requires $O(x^2 \cdot \frac{MN}{4}) = O(MN) \approx O(MN)$ time for the extraction of all dense SIFTS in the image. For SIFT quantization, each SIFT desircctor needs to traverse through the SVT to find a match with the leaf codeword. Suppose the depth and branch number of the SVT is $H$ and $B$, then each SIFT requires $O(B \cdot H)$ time to find its matched codeword in the codebook. For all the SIFTS in an image, they require $O(B \cdot H \cdot \frac{MN}{4}) = O(MN)$ time for matching with the codebook in the SVT. Therefore, the total time complexity for the feature extraction in a $M \times N$ dimensional landmark image is $O(MN) + O(MN)$, where $x$ is half of the image patch size. It can be seen that when $x$ is smaller, more SIFT descriptors are generated from the image, and more time are required for these descriptors’ quantization via SVT.

<table>
<thead>
<tr>
<th>Number of selected direction clusters</th>
<th>Averagely generated codebook size</th>
<th>Average candidate image number</th>
<th>Recognition rate</th>
<th>Recognition time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2200</td>
<td>400</td>
<td>0.869</td>
<td>0.03</td>
</tr>
<tr>
<td>2</td>
<td>4300</td>
<td>700</td>
<td>0.903</td>
<td>0.06</td>
</tr>
<tr>
<td>3</td>
<td>5800</td>
<td>930</td>
<td>0.901</td>
<td>0.09</td>
</tr>
<tr>
<td>4</td>
<td>7600</td>
<td>1100</td>
<td>0.899</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>9500</td>
<td>1300</td>
<td>0.895</td>
<td>0.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPS radius (m)</th>
<th>Average codebook size</th>
<th>Average No. of candidate images to be matched</th>
<th>Recognition rate</th>
<th>Recognition time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>300</td>
<td>8</td>
<td>0.480</td>
<td>0.001</td>
</tr>
<tr>
<td>50</td>
<td>600</td>
<td>40</td>
<td>0.730</td>
<td>0.007</td>
</tr>
<tr>
<td>100</td>
<td>1300</td>
<td>130</td>
<td>0.893</td>
<td>0.01</td>
</tr>
<tr>
<td>200</td>
<td>2900</td>
<td>310</td>
<td>0.933</td>
<td>0.02</td>
</tr>
<tr>
<td>300</td>
<td>3500</td>
<td>510</td>
<td>0.921</td>
<td>0.04</td>
</tr>
<tr>
<td>400</td>
<td>4300</td>
<td>700</td>
<td>0.903</td>
<td>0.06</td>
</tr>
</tbody>
</table>

4.3. Performance comparison of proposed approach vs. other approaches

After determining the parameters, the proposed approach is compared with several typical methods including city-scale
location recognition (CLR) in [19], the descriptive visual word (DVW) in [20] and the location discriminative codebook (LDV) in [7] for mobile landmark recognition. To make a fair comparison, dense SIFT is used as the image feature representations for all the methods. Hierarchical $k$-means algorithm is then used to generate the SVT for feature quantization. The SVT depth ($H$) ranges from 3 to 6 with the branch $B = 10$, which corresponds a codebook with size ranging from $10^3$ to $10^6$. Fig. 8 gives the recognition rates of these methods for different SVT depths. From the figure, it is seen that the proposed method consistently outperforms the other methods for codebook size ranging from $10^3$ to $10^6$. The overall SVT-based mobile landmark recognition method achieves a highest recognition rate of 93.3% for a codebook size of $10^4$. The reasons for the proposed approach superior to other methods are that: (i) context information is considered to generate a set of direction-aware codewords in this work while in other works such as [7,19,20], no context information is considered to generate the codewords. The direction-aware codewords are more representative to describe a landmark image captured from a specific

![Fig. 7. Illustration of the number of (a) visual words to describe each query image (upper row), and (b) candidate images selected for each query image (lower row).](image)

![Fig. 8. Comparison of the proposed overall approach vs. other typical approaches for mobile landmark recognition.](image)
direction, (ii) the proposed method uses an integration of context and content analysis for fast image scoring, which greatly reduces the search space for the query image and increases the recognition accuracy. Further, we have conducted a few statistical hypothesis tests [38] to evaluate these comparisons. It is found that the proposed approach outperforms the LDV method at 90% level of confidence, and outperforms both the DVW and CLR methods at 95% level of confidence on the NTU database.

Then, we investigate the experimental results of varying the compact codebook size for landmark recognition. Two situations as in Section 3.2 (A) are considered, one is to use direction information alone to select the codewords for the codebook generation, and the other is to use both direction and location information to select the codewords for codebook generation. The depth of the SVT is chosen as 4. The overall experiment results are given in Fig. 9. It is seen that (i) the proposed approach using both location and direction information for codebook generation achieves the best performance of around 93%, using a smallest codebook size which takes up only 30% of the original number of codewords. (ii) Using direction alone gives inferior performance of around 90% with more codewords are selected to represent an image than using both direction and location information, which causes higher computational cost. The reason is that the combination of location and direction information can give relatively finer search space than using direction alone, and yield a smaller codebook size. (iii) When the size of the codebook goes beyond a certain threshold, the performance of all the methods decreases. This is due to that some codewords such as the background or noisy words have poor discriminative capabilities in discriminating different landmark categories, if these words are selected into the codebook, they will reduce the recognition accuracy. It is also noted that when no codewords selection process is involved (in this case, the original BoW is used), the performance of all the methods are the same which is around 87%.

In addition, we also compare the performance of using different local descriptors including sparse SIFT [16], dense SIFT [25], SURF and Upright-SURF (U-SURF) [15] in the proposed method. To guarantee fair comparison, all the experimental settings remain as the same as above (codebook size and radius $R$ are set as $10^4$ and 200 m respectively). The results are given in Table 3 below. It can be observed that the dense SIFT descriptor offers the best performance (93.3%), followed by U-SURF, SURF, and sparse SIFT. The reasons why dense SIFT has outperformed the other local descriptors is because the dense SIFT descriptors are obtained from overlapping regular patches in the image without rotation. As the landmark images are usually acquired in upright position with little rotation, the dense SIFT descriptors that do not consider orientation information turn out to be more effective than sparse SIFT and SURF that consider orientation information in landmark recognition.

![Fig. 9. Comparison of selecting various number of codewords for codebook generation between the proposed and other methods.](image)

Finally, the evaluation of the recognition time (the image matching time in the server end) for the proposed approach is performed. The results are given in Table 4. The recognition rate for each condition is also given based on the results provided in Fig. 9 above. From the table it is seen that when more context information (direction, locations) are used to reduce the dimension of the BoW histogram, the average recognition time is decreased along with increasing recognition rate. Specifically, when both direction and location information are used to generate the codebook, an average recognition time of 0.02 s is achieved with a highest recognition rate of 93.3%.

### 4.4. Evaluation of the proposed approach on San Francisco dataset

To demonstrate the effect of the proposed method on a large-scale dataset, the San Francisco landmark dataset from [28] is used for experiments. The 1.06M PCI training images in the dataset are used to build the vocabulary. A set of 803 cell phone images taken with a variety of different camera phones by various people over several months are used for query. The CLR method in [19], LDV method in [7] and DVW method in [20] are selected for comparison as discussed before. The experiment results of using original BoW which involves no codeword selection process is also given. As only GPS information is provided and there is no direction information in the dataset, we use the conventional SVT instead of the CSVT for vocabulary learning. The proposed content–context integrated mobile landmark recognition approach is used for image scoring. Fig. 10 gives the recognition rates of these methods for different SVT depth $H$ ranging from 3 to 6, and $B = 10$. From the figure it can be seen that when the SVT depth reaches 6, which means a million leaf codewords are generated, the proposed SVT-based mobile landmark recognition approach achieves the highest recognition rate among all the methods (around 85%). Further, the statistical hypothesis tests are also conducted on this dataset. It is found that the proposed approach outperforms the LDV method at 85% level of confidence when SVT depth reaches 5, and outperforms both the DVW and CLR methods at 95% level of confidence on the San Francisco database.
5. Conclusion

This paper presents a context-aware vocabulary tree approach for mobile landmark recognition by incorporating the context information (location and direction) into the codebook learning and image matching process. First, the image’s direction information is utilized to construct a context-aware scalable vocabulary tree by decomposing the leaf nodes of the original SVT into finer direction-aware codewords via hierarchical k-means clustering algorithm. The discriminative capability of these codewords is then learned through a proposed information gain based method which integrates the image’s direction and location information. Second, a compact vocabulary is generated for each query image based on the learned codewords’ discriminative information. The vocabulary quantizes the image into a compact BoW histogram, which is then feed to the SVT as indexed by inverted files for fast image scoring. The context information is utilized to shortlist the landmark candidates to speed up the image matching process. Experimental results on two challenging database show that the proposed method can achieve both high recognition rate and short recognition time in mobile landmark recognition.

In the future, we will further investigate how to perform fast image search without the context information. This is very important as when the dataset is scaled up to a larger world wide dataset, many images in the dataset may not have the direction and location information. In addition, we will also study some relevant computing techniques such as distributed inverse indexing to handle the scalability of the web scale search.

References