Finding Perfect Rendezvous On the Go: Accurate Mobile Visual Localization and Its Applications to Routing *

Heng Liu [1], Tao Mei [2], Jiebo Luo [3], Houqiang Li [1], Shipeng Li [2]
[1] University of Science and Technology of China, Hefei 230027, P. R. China
[2] Microsoft Research Asia, Beijing 100080, P. R. China
[3] University of Rochester, Rochester, NY 14627, USA
{sorcerer@mail., lihq@}ustc.edu.cn; {tmei, spli}@microsoft.com; jluo@cs.rochester.edu

ABSTRACT

While on the go, more and more people are using their phones to enjoy ubiquitous location-based services (LBS). One of the fundamental problems of LBS is localization. Researchers are now investigating ways to use a phone-captured image for localization as it contains more scene context information than the embedded sensors. In this paper, we present a novel approach to mobile visual localization that accurately senses geographic scene context according to the current image (typically associated with a rough GPS position). Unlike most existing visual localization methods, the proposed approach is capable of providing a complete set of more accurate parameters about the scene geo—including the actual locations of both the mobile user and perhaps more importantly the captured scene along with the viewing direction. Our approach takes advantage of advanced techniques for large-scale image retrieval and 3D model reconstruction from photos. Specifically, we first perform joint geo-visual clustering in the cloud to generate scene clusters, with each scene represented by a 3D model. The 3D scene models are then indexed using a visual vocabulary tree structure. The phone-captured image is used to retrieve the relevant scene models, then aligned with the models, and further registered to the real-world map. Our approach achieves an estimation accuracy of user location within 14 meters, viewing direction within 9 degrees, and scene location within 21 meters. Such a complete set of accurate geo-parameters can lead to various LBS applications for routing that cannot be achieved with most existing methods. In particular, we showcase three novel applications: 1) accurate self-localization, 2) collaborative localization for rendezvous routing, and 3) routing for photographing. The evaluations through user studies indicate these applications are effective for facilitating the perfect rendezvous for mobile users.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*; I.4.8 [**Computing Methodologies**]: Image Processing and Computer Vision—*scene analysis*

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Mobile Visual Localization, Geo-tagging, Scene Reconstruction, Location-based Services.

* This work was performed at Microsoft Research Asia.

MM'12, October 29–November 2, 2012, Nara, Japan.

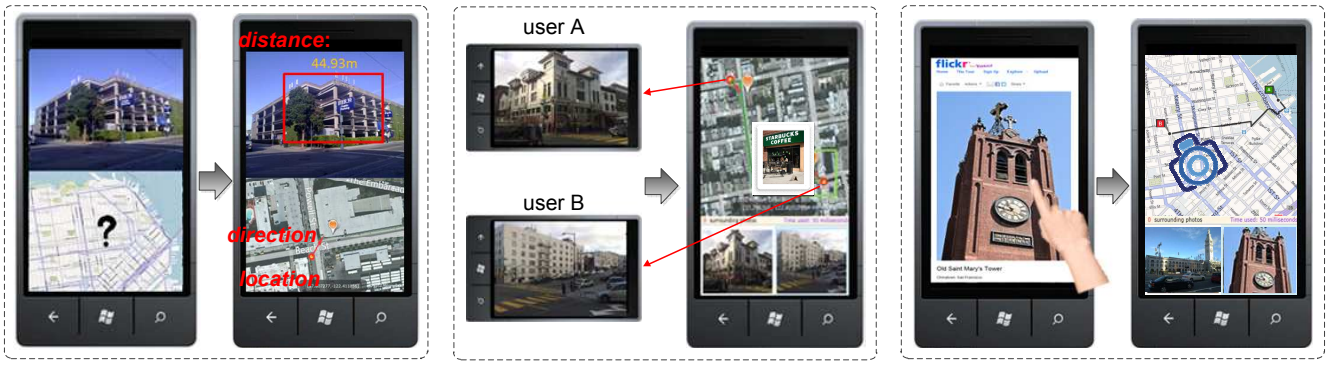
1. INTRODUCTION

Location-based services (LBS) are becoming ubiquitous. People are using their mobile phones to enjoy the many LBS applications on the move. For example, 18% of U.S. smartphone users accessed at least one LBS application per month in 2011, and 10% of mobile applications in the market are LBS-oriented. One of the fundamental problems in these applications is localization—How to automatically obtain the accurate location context of mobile users still remains a challenge.

Traditional approaches for mobile localization employ inertial hardware such as embedded GPS or cellular tower proximity to determine location in outdoor environments, or WiFi/GSM-based alternatives for indoor operation. Unfortunately, the embedded GPS modules rely on a satellite navigation system and need a clear view of at least four satellites to provide sufficient positioning accuracy. As a result, the estimated GPS location in a crowded urban scene or on a cloudy day is error prone, usually leading to an error of 50∼100 meters [5], [29]. Moreover, in a multimedia-oriented application, when users annotate their photos or videos using these records, they are actually labeling according to the camera location rather than the true scene location, which can cause significant deviations in many LBS applications. Similar problems exist in positioning with other sensors such as Cell-IDs or WiFi.

Precise geographic information about location plays a key role in LBS, especially in multimedia-oriented applications (e.g., augmented reality) that have a high expectation of location accuracy in meters or less. Researchers are now investigating ways to use the phone-captured images for localization as a complement or an alternative, as it contains more context information than embedded sensors, leading to a more precise location description. A typical approach is to match the captured image against a database with geo-referenced images, and use the geo-tag from the (near-) duplicate images to determine location [2], [32]. However, this approach relies heavily on the "ill-posed" geo-referenced image database, where images are associated with the geo-locations by the mobile devices (since the geo-locations are usually obtained from the embedded GPS devices) rather than the scenes themselves. This could result in incorrect geo-tagging because the distance between the mobile user and the scene is undetermined.

We are investigating in this paper if we can provide a complete



(a) accurate self-localization (b) collaborative localization for routing (c) routing for photographing suggestion

Figure 1: LBS applications based on accurate mobile visual location: (a) accurate self-localization: a mobile user wants to determine the accurate location of himself and the building, as well as his viewing direction, (b) collaborative localization for routing: two users are finding a perfect rendezvous point collaboratively by taking pictures, and (c) routing for photography suggestions: a tourist is finding the best place to photograph a landmark appearing in a shared picture.

set of more accurate parameters for the geo-context of a scene. The geo-context parameters include the actual location of a mobile user, the viewing direction, as well as the location of the scene. This is quite different from most existing approaches that only focus on one or part of these attributes [2], [5], [11], [25], [28], [32]. The comprehensive set of geo-context attributes can lead to a wide variety of LBS applications that cannot be achieved by any existing system. Figure 1 shows three novel applications of this kind:

- Accurate self-localization for a single user. A mobile user takes a picture of a landmark to determine his location, viewing direction, and the distance from the landmark.
- Collaborative localization for finding the perfect rendezvous. Two users or a group of users are looking for a place in a crowded area to gather. They all take pictures of nearby buildings through their phones independently. They can then determine each other's location, , as well as a suggested rendezvous point and its route.
- Routing for photography suggestions. A tourist is interested in a photo of a church shared by his friend. He then wants to know where to take a similar picture of this church (i.e., the route to this church, and the location and viewing direction for photographing this church).

Our approach takes advantage of the advanced techniques for large-scale image retrieval and 3D model reconstruction from photos. Specifically, we first perform joint geo-visual clustering in the cloud to generate scene clusters, with each scene represented by a 3D model reconstructed from the images in this cluster. The 3D scene models are then indexed using a visual vocabulary tree structure. The phone-captured image is used to retrieve relevant scene models, aligned with the models, and further registered to the real-world map. As a result, we are able to achieve an estimation accuracy of user location within 14 meters, viewing direction within 9 degrees, and scene location within 21 meters.

In summary, our contributions are two-fold. First, we propose an approach for accurate image localization for mobile users, by taking advantage of the the geo-referenced images on the web and reconstructed 3D scene models. Our approach can provide a more comprehensive set of accurate parameters of a scene's context. Second, we present a wide variety of LBS applications based on accurate location context, including 1) accurate self-location, 2) collaborative localization for rendezvous routing, and 3) routing for photographing.

The remainder of this paper is organized as follows. Section 2 reviews the related work for image-based localization and mobile location recognition. Section 3 describes our proposed approach to accurate mobile visual localization. Section 4 showcases various applications derived from our system. Experiments are presented in Section 5, followed by the conclusions in Section 6.

2. RELATED WORK

2.1 Image-based Localization

With the growth of geo-referenced data, image-based location recognition has drawn extensive interest from multimedia and vision communities [20]. Zhang and Kosecka propose a framework that searches the closest reference views for an image and provide a coarse method for location recognition in an urban environment [33]. However, their scheme cannot scale up for large datasets. Schindler *et al.* use an inverted file indexed with a vocabulary tree for city-scale location recognition [28]. Their dataset contains 30,000 images. Zamir *et al.* propose a method to exploit repetitive features discovered in large urban environments to achieve higher accuracy for image search and GPS estimation [32]. The authors combine the features from similar images to form "scene maps" to compress the data and achieve considerable improvement for retrieval [2].

Some researchers have moved a step further—they are more concerned about how to estimate the view directions of photos. Kroepfl *et al.* locate photos by registering on street view panoramas [15]. Similarly, Park *et al.* utilize both Google Street View [9] and Google Earth Map [8] to estimate the view direction of a geo-referenced image [25]. In [21], given a query image, a set of matched photos of the underlying scene is found ind the database using the Scale-Invariant Feature Transform (SIFT) Flow [18]. The photos with view directions pointing to the user-indicated region are returned by the system. However, these systems cannot deal with the estimation of location and do not work without an initial GPS position.

In some recent works [12], [16], [17], [27], the structure-form-motion (SfM) technique is exploited to reconstruct a 3D scene from a collection of images. The images are then localized by matching the feature points to the 3D models. This could produce more precise localization results. However, the scene models are usually reconstructed from the datasets at the scale of 10^3 to 10^5 photos,

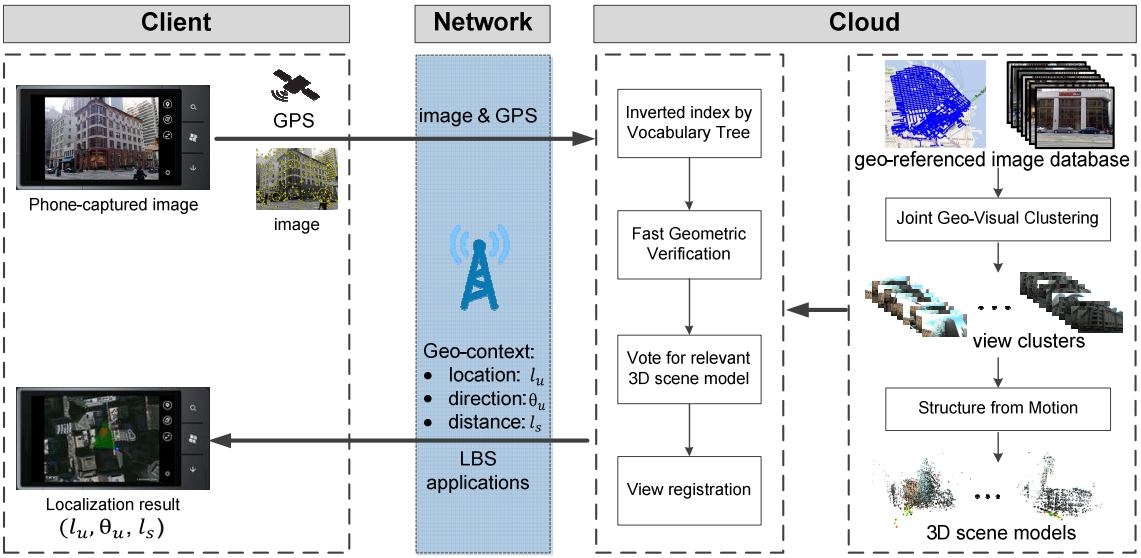


Figure 2: The proposed framework for accurate mobile visual localization and its applications.

acquired by text-based search from the Web or artificially captured. As a result, the models usually cover a small fraction of the whole city. Furthermore, in these works the reconstruction treats all the images as a whole while ignoring context such as geo-information, which could be used to guide the reconstruction for a better performance.

2.2 Mobile Location Recognition

Schroth *et al.* is concerned about the latency and the limited storage of the mobile device [29]. Chen *et al.* build a million-scale street view dataset and conducted comprehensive experiments to evaluate their retrieval scheme [5]. Ji *et al.* propose a discriminative vocabulary coding scheme to achieve a low latency query delivery and discriminative description [13]. In particular, the authors in [31] present a mobile application that can teach mobile users to capture pictures that can distinctively represent the surrounding scenes.

In contrast to the above systems, our proposed framework aims to provide the complete information of a scene's geo-context: location, view direction, and distance to the captured scene with a higher accuracy than the GPS-level. Accurately estimating geo-context can lead to better LBS services for mobile users.

3. ACCURATE MOBILE VISUAL LOCALIZATION

Figure 2 shows the proposed system architecture for accurate mobile localization, including: clients, network, and the cloud. The localization is considered a combination of retrieval and registration problems in a geo-referenced database. To better utilize geometric constraints and derive a more accurate estimation of query image, we build up the geo-reference database that not only has images but also 3D scene models. Given a photo captured by a user with a mobile device, an image (with a rough GPS location from the sensor) is transmitted through a wireless network to the cloud for more accurate localization. First, we extract the SIFT descriptors from the query image. Then, we retrieve the related 3D scene models in the database using the Bag-of-Word (BoW) representation indexed with a vocabulary tree structure [23]. If we have the GPS information associated with the query image, then we use this information to filter unrelated images. The query image will finally be aligned to 3D models on a real-world map. In this way, we can obtain the complete scene context $(\mathbf{l}_u, \theta_u, \mathbf{l}_s)$, with the location of the mobile user $\mathbf{l}_u$, the view direction θ_u, and location of the captured scene $\mathbf{l}_s$. In the following section we will describe the major components of our system in details.

3.1 Building 3D Scene Models

To localize the query images, searching should be conducted in a geo-referenced dataset. The most commonly seen type of dataset is a collection of images with GPS information. Luckily, with the increasingly popularity and accessibility of online photo-sharing services, such as Flickr [6] and Panoramio [24], more and more people are sharing their photos, a portion of which are geo-tagged. This abundant resource of geo-referenced images has proven very useful in many multimedia and visualization applications. However, the contents is diverse and noisy. One way to build a better-structured database is harnessing street view images that are obtained by vehicles equipped with cameras and other sensors (like GPS) [5]. Existing visual location recognition systems are mainly based on these collected datasets. However, they can only estimate limited parameters as the search results from these datasets consist only of the images with GPS information. To predict the geographic information of a query image, we need more accurate information or models from these datasets, e.g., the epipolar geometry among the images [33].

To accurate localize image in a million-scale database, we take advantage of both geo-information and visual content to support 3D scene reconstruction and retrieval. Images which are geographically and visually close to each other should be clustered into the same group. Then, we can reconstruct 3D models using the structure-from-motion (SfM) technique in these groups. To achieve this, we present a joint geo-visual clustering strategyto do the preprocessing.

3.1.1 *Geo-clustering of database images*

First, *geo-clustering* is performed using the location information $(latitude, longitude)$ to ensure that only geographically-nearby image are able to depict the same scene. This will significantly reduce the computational costs of the succeeding visual-clustering. For simplicity, images far away from each other are disconnected

by leveraging an open-ball operation with the radius r_g, thus the geographical similarity matrix is computed as follows:

$$Sim_g(i,j) = \begin{cases} g_{i,j}, & d_{geo}(i,j) < r_g \\ 0, & d_{geo}(i,j) \geq r_g \end{cases}, \quad (1)$$

where $g_{i,j} = exp\{\frac{-d_{geo}(i,j)}{\sigma_g}\}$ denotes the geo-proximity of two images, $d_{geo}(i,j)$ is the distance from the location of image i to image j, and σ_g is the scale factor. Then, the affinity propagation (AP) algorithm is adopted to find the geo-clusters [7]. AP is widely used to group data into a number of groups according to the similarity between pairs of data points. The total number of groups is automatically inferred from the data.

3.1.2 Visual-clustering of database images

After we determine the geo-clusters, we further conduct clustering based on the visual similarity of images. Similar with the geo-clustering, we also need to compute a matrix Sim_v for each geo-cluster that represents the pairwise visual similarity between images. Unlike geo-clustering, computing pairwise similarity of images is the most computational procedure. Typically this is done by matching local visual features, followed by checking the consistency in the geometry using RANSAC and other forms of spatial matching [26]. To speed up this procedure, we use the inverted-indexing technique. For each image in the cluster, its features are quantized as visual words, as described in Sec 3.2. Then, each image is represented by a histogram of visual words. Matching this image to others in the cluster is followed by fast geometric verification. This matching is conducted in a small subset rather than the whole database, thus allowing one query per image in each cluster. Furthermore, we restrict the process to ensure that only the image pairs with the number of inlier matched features more than r_v are connected in the visual graph. The visual similarity matrix is:

$$Sim_v(i,j) = \begin{cases} v_{i,j}, & v_{i,j} \geq r_v \\ 0, & v_{i,j} < v_g \end{cases}, \quad (2)$$

where $v_{i,j}$ is the number of inlier matches between image i and j, representing their visual similarity.

Finally the visual-clusters are obtained using AP with the visual similarity matrix. The results of geo-clustering in a city and an example of visual-clusters of a geo-cluster are given in Figure 3, where each row in 3(b) depicts the images in the same visual cluster. The reconstructed 3D models are shown at the end of each row.

3.1.3 3D Reconstruction via SfM

After the geo-visual clustering, we have a set of N image clusters $\{\mathcal{C}_i\}_{i=1}^{N}$. Each cluster $\mathcal{C}_i = \{I_i^m\}_{m=1}^{M_i}$ consists of images depicting the same scene. Instead of representing the scenes with sets of images, a better choice is using 3D models that contain stronger geometric constraints. These geometric constraints can deliver the pose of images directly compared to the 2D image matches.

A 3D model can be created using multiple-view vision methods from image clusters $\mathcal{C} = \{I^m\}_{m=1}^{M}$. In our system, the reconstruction of 3D scenes from images in the same cluster is mainly based on the state of the art **SfM** algorithm [30]. We briefly summarize the related steps for 3D scene reconstruction as follows. First, for each pair of images, the key point descriptors are matched using the approximate nearest neighbors (ANN) KD-tree algorithm [1]. We next reconstruct the set of cameras and 3D points. We start the reconstruction from a initial selected image pair which has a large number of matches and a wide baseline. The five point algorithm is used to estimate camera parameters for this initial pair [22]. The matched image points are triangulated. Then, by adding a few images to the model at a time, we can build the scene model incrementally. The reconstructed model $\mathcal{S}(\mathcal{C}) \triangleq (\mathcal{X}, \mathcal{A}, \mathcal{W})$ of a cluster $\mathcal{C}$ is represented with the following structure: 1) a set of 3D points $\mathcal{X} = \{X^n\}_{n=1}^{N}$ in 3D Euclid coordinate (U, V, Z), 2) a set of camera $\mathcal{A} = \{A^m\}_{m=1}^{M}$, where each camera A^m consists of an image I^m, a rotation matrix $\mathbf{R}^m$, a translation $\mathbf{t}^m$, and a focal length l^m, and 3) a set of Binary mappings $\mathcal{W} = \{w_{n,m}\}_{n,m=1}^{n=N,m=M}$ between the points and cameras indicates whether X^n can be observed from camera A^m. If $X^n = (X_U^n, X_V^n, X_Z^n)$ is observable in A^m, then there exists an 2D feature point $x^{n,m} = (x_u^{n,m}, x_v^{n,m})$ represented in 2D image coordinate (u, v). At each iteration, the reconstructed model is further refined using a bundle adjustment. It is equivalent to the following optimization problem that minimize the reprojection errors:

$$\arg\min_{X,\mathbf{R},t,l} \sum_{X^n,\mathbf{R}^m,t^m,l^m} w_{n,m} \parallel x^{n,m} - \widehat{x}^{n,m} \parallel^2, \quad (3)$$

where $\widehat{x}^{n,m} = (\widehat{x}_u^{n,m}, \widehat{x}_v^{n,m})$ is the reprojection of X^n into image I^m, the reprojection is computed as:

$$\lambda \begin{pmatrix} \widehat{x}_u^{n,m} \\ \widehat{x}_v^{n,m} \\ 1 \end{pmatrix} = l^m(\mathbf{R}^m\mathbf{X}^n + \mathbf{t}^m), \quad (4)$$

where λ is an arbitrary homogeneous scaling factor. Thus bundle adjustment jointly refines the estimated camera parameters and 3D points. Finally, when the reconstruction has been completed, we convert the coordinate into a real world coordinate. With the labeled GPS information of the images, a similar transformation can register the 3D model into real world coordinate. Finally, the query image can be localized by searching for and registering the 3D scenes in the geo-referenced database.

3.2 Searching 3D Scene Models

To efficiently find the relevant images and scenes for the query image, a visual vocabulary tree structure is adopted for indexing. We will describe these steps in the following sections.

3.2.1 Image Representation

A SIFT descriptor is exploited for image representation because of its strong capacity for description and discrimination [19]. For each image, a set of interest points are detected, each with a 128-dimensional descriptor, the orientation, and the scale of the key point. We use the BoW scheme to quantize the local descriptors. Each descriptor is mapped to an integer visual word ID. The visual words are typically trained from a sampled SIFT descriptor set by clustering, while the clustering centroids are defined as visual words. For efficient clustering and quantization, a hierarchical tree based structure is adopted and the resulting leaf nodes are considered visual words [23]. The hierarchical tree has two basic parameters: branching factor B and hierarchical layer H, where B controls the number of clusters in each layer of the tree and H determines the depth of the tree. In general, the number of visual words is approximately B^H. The total number of comparison operations is $B \times H$. In our implementation, we set the parameters as $B = 10$, $H = 6$, yielding a codebook with one million visual words. Finally, the inverted file system is exploited to index the images with the trained vocabulary tree.

3.2.2 Searching with GPS

Our system can deal with a situation where no initial GPS information is available in addition to the image. However, a rough GPS initiation can be very beneficial for guiding a search that is limited

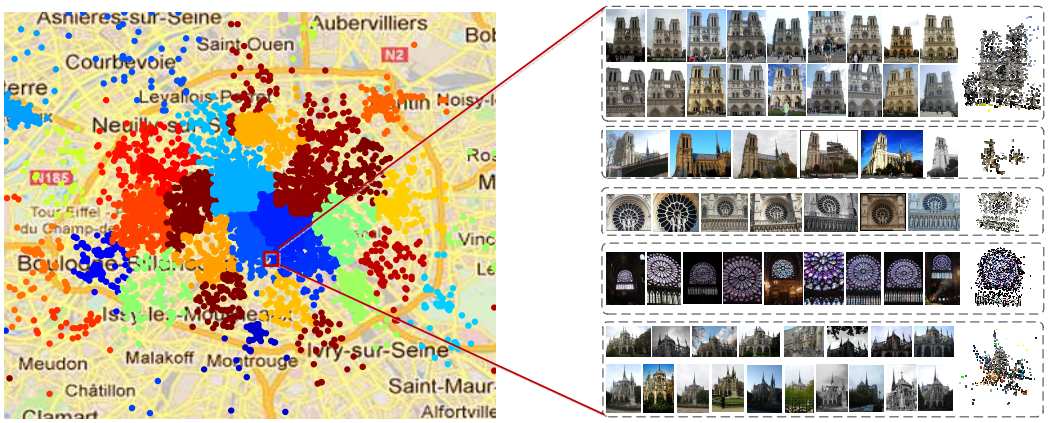


Figure 3: An example of geo-visual clustering in the city of Paris. (a) The geographical regions are generated by employing Affinity Propagation clustering. Different clusters are denoted by different colors. (b) Part of the visual clusters in one geo-cluster near Norte Dame de Paris. Each row represents images in the same visual cluster along with the reconstructed 3D scene model.

in scope compared to the whole database. This could be helpful both in accelerating the retrieval speed and enhancing the retrieval precision. When a GPS context is available for the query image, only those images in the vicinity of the current GPS record are scored in the inverted index matching. Given $\mathbf{G} = (G_{lat}, G_{long})$ as the GPS coordinate for a query image, we define the area within R meters as the vicinity. We choose $R = 300$ in our experiments according to the report in [5] that it would be appropriate in most cases.

3.2.3 *Fast Geometric Verification*

A subsequent geometric verification (GV) considering the spatial consistency of local features is needed to reject the false-positive matches. Especially, this verification is useful and indispensable in landmark or street view image retrieval where repetitive structures are common and local patches like windows are similar so that multiple descriptors might be mapped to the same visual word. Usually, RANSAC-based methods with an affine or epipolar geometry model are applied. But the full verification with RANSAC is time consuming thus can only be performed on a small set in the top results. Moreover, images of landmarks or street views are usually captured with known gravity orientation. This gravity-aligned property has been analyzed and utilized for more precise feature matching in previous works [5]. In our work we also noticed this characteristic of the landmark images. So we choose to use a relaxed but efficient GV strategy proposed in [34]. First, each descriptor is quantized in both descriptor and orientation spaces to enhance the matching accuracy. A soft quantization strategy is employed to mitigate the quantization error. Then, the concise binary matrices named spatial maps that encode the relative feature positions in each image are generated. By comparing the spatial maps we can find and remove the false matches effectively. This achieves performance comparable to RANSAC and the computational time is greatly reduced, thus enabling retrieval in a large-scale database. If the number of inliers after the geometric verification is higher than threshold T_{GV}, the retrieved image is accepted as a true match, otherwise the image is rejected.

3.3 Localization by View Registration

The retrieval process returns a short list of candidate images, along with the filtered feature matches between the query image and these candidates. At the same time, all the 3D scene models in the database that have related candidate images ($\{\mathcal{S}(\mathcal{C}_i)|q \cap \mathcal{C}_i \neq \emptyset\}$) can be evaluated using the image retrieval results. We use a majority voting scheme to select the 3D scene model with which the query image q has the most overlap. For each 3D scene model $\mathcal{S}(\mathcal{C}_i)$, a voting score $V(\mathcal{S}(\mathcal{C}_i), q)$ is computed by

$$V(\mathcal{S}(\mathcal{C}_i), q) = \sum\nolimits_{f^q \in q} \delta(f^q, \mathcal{S}(\mathcal{C}_i)), \tag{5}$$

where $\delta(f^q, \mathcal{S}(\mathcal{C}_i)) \rightarrow \{0, 1\}$ indicates whether a feature point f^q in the query image q can find a match in the candidate images belonging to the scene model $\mathcal{C}_i$. Thus, $V(\mathcal{S}(\mathcal{C}_i), q)$ denotes the approximate strength of the link between the query image and $\mathcal{S}(\mathcal{C}_i)$ by measuring how many feature points in the query image are covered by the scene model. The scene model with the highest number of hits is chosen to further align the query image for an accurate localization.

From the previous voting procedure, the most related scene model can be determined. Figure 4 illustrates the registration procedure. The camera pose can be estimated by registering the query image to 3D points. For each 3D model, a KD-tree is established. Features extracted from the query image can search the nearest neighbor in this tree to find 2D-to-3D matches. When there is sufficient correspondences between image features x and 3D points X, the 3×4 projection matrix $\mathbf{P}$ can be estimated.

$$\lambda \begin{bmatrix} x_u \\ x_v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_U \\ X_V \\ X_Z \\ 1 \end{bmatrix}, \tag{6}$$

where λ is an arbitrary homogeneous scaling factor that is the same as in section 3.1.3, There are 11 degrees of freedom (DOF) in $\mathbf{P}$ and the 6-point DLT algorithm is usually adopted [10]. $\mathbf{P}$ can be decomposed into a 3×3 intrinsic matrix $\mathbf{K}$, a 3×3 rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{t}$. where

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}], \tag{7}$$

$$\mathbf{K} = \begin{bmatrix} l & \gamma & u_0 \\ 0 & l & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where l is the focal length of the camera, γ is the skew of the camera which is often 0, and u_0 and v_0 define the principal point of the

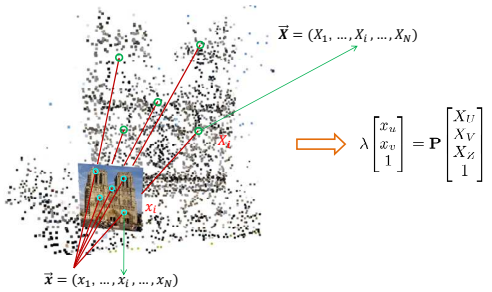


Figure 4: Localization by view registration, the feature points in the image are matched to the 3D points to estimate the camera pose.

image, usually in the center. With these assumptions, only l need to be considered in $\mathbf{K}$. Notice that $\mathbf{R}$ is actually determined by three constituent Euler angles and t is a 3-vector, so that the total DOF is reduced to 7 and only 4-points are needed.

There are methods that can estimate the focal length and other parameters such as the 4-point-solver recently proposed in [14]. When focal length is given, e.g., in the EXIF, the 3-point perspective pose estimation method can be used. According to the report in [16] and [27], using the focal length recorded in the EXIF can reduce the estimation errors and is beneficial for real-time estimation. Thus, in our system we assume the focal length is always available. For a query image in which the EXIF is not available, we transfer the focal lengths from the top candidate images. So the query image is aligned using the 3-point method. With the estimated parameters R and t, the camera position of the image is given by $-\mathbf{R}'\mathbf{t}$ and view direction is $-\mathbf{R}'\begin{bmatrix}0 & 0 & 1\end{bmatrix}'$. The scene position is estimated using the 3D points that can be observed from the query image and we use their mean as the scene location of the image. Finally we register the image to the real world map. We estimate a similar transform from 3D model's horizontal axis to the GPS coordinate. Then we can get $(\mathbf{l}_u, \theta_u, \mathbf{l}_s)$ denoting the camera location, the camera view direction, and the scene location respectively from this similar transform.

3.4 Collaborative Localization with Multiple Images

We are also concerned about the collaboration in the localization using a group of K images $\mathcal{Q} = \{q_k\}_{k=1}^K$, with the assumption that the images are taken in nearby areas. First, features are extracted from the query images. A set of candidate images are returned for each query image. Then we check whether there are overlaps in the retrieval results. If the images have any overlap, then we combine the feature vector of the images to search again in the database. The voting score for each scene model $\mathcal{S}(\mathcal{C}_i)$ in this situation is:

$$V_m(\mathcal{S}(\mathcal{C}_i), \mathcal{Q}) = \sum_k \frac{V(\mathcal{S}(\mathcal{C}_i), q_k)}{K}, \quad (9)$$

- If these images are capturing the same scene, they will share features. Thus, by combining the feature vectors the shared features will have a higher weight in the search.

- Using this combined feature vector, the scenes in the database that cover more query images will have higher match score and thus are more likely to be retrieved. So, we can localize these images in the same scene to get their relative distance and directions.

If the images have no overlap, each image is localized individually.

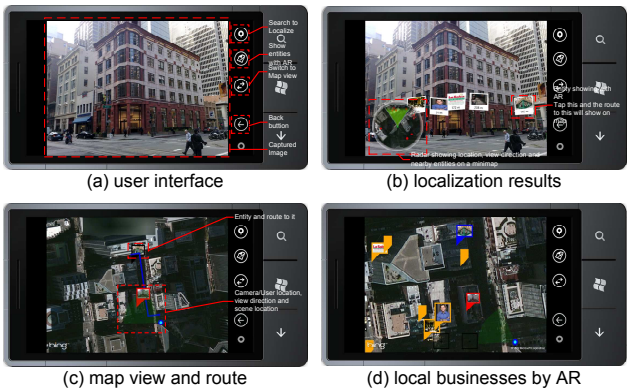


Figure 5: The user interface for mobile visual localization. (a) shows that the user captures the scene with the phone camera. (b) Then the location and view direction are estimated by the proposed framework. Nearby businesses are shown on the screen to augment reality. (c) The map view of the localization results and the route to the selected destination. (d) The map view with nearby local businesses.

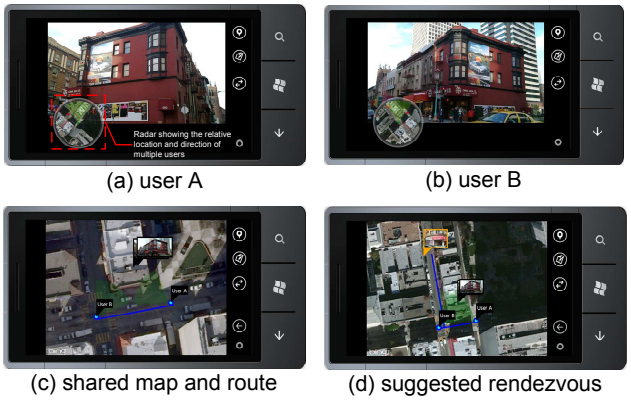


Figure 6: User interface for collaborative user location. (a)(b) Two users are located in collaboration. (c) Their relative locations, view directions and suggested rendezvous points shared on the same map. (d) The suggested rendezvous and route.

4. APPLICATIONS WITH ACCURATE GEO-CONTEXT

In this section, we will show how the proposed accurate mobile visual localization framework can be integrated into a variety of practical applications for mobile users.

Application I: Accurate self-localization As its core functionality, our framework can supply accurate localization and mapping service for a mobile user. Consequently, the user can obtain accurate geographic information and other information such as local business. This can help the user better explore his/her surroundings. The user interface for this application is shown in Figure 5. First, the user uses the camera to capture a photo as a fingerprint of the scene. Then by pressing the "localization" button the image and rough GPS will be transmitted to the cloud. Our service will return to the user the location $\mathbf{l}_u$, view direction θ_u, and the location of user captured scene $\mathbf{l}_s$. With the rich and accurate information, we can supply better mapping services to help the user better explore the area. In addition, we can recommend nearby businesses to the user and show related entities along the view direction θ_u using augmented reality techniques. In Figure 5 (b), the radar shows the surrounding area of the user within one kilometer.

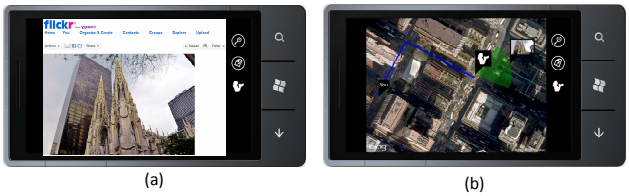


Figure 7: When a user is browsing a photo online, we can calculate the location and view direction so that the user can take a similar photo and suggest route from the user's current location to photography spot.

Application II: Collaborative localization for finding rendezvous points. The proposed framework can also be used to serve multiple users. A useful application for multiple users is guiding them to a rendezvous point—the perfect place for gathering and meeting. Another scenario is that friends in a crowded area may get separated from each other, especially in an unfamiliar location. To locate themselves and quickly find their way back together, each of them can take a photo of a landmark at his/her location and upload the photos to the cloud. With the multiple image localization service, they will receive accurate geographic information for themselves and their friends. Let $(\mathbf{l}_{u1}, \theta_{u1}, \mathbf{l}_{s1})$ and $(\mathbf{l}_{u2}, \theta_{u2}, \mathbf{l}_{s2})$ denote their localized context respectively. This information is shown with a radar on the screen as a guideline for the users—whether they have been to the same location ($\mathbf{l}_{u1}$ close to $\mathbf{l}_{u2}$), and which direction they should turn ($\theta_{u1} \leftrightarrow \theta_{u2}$). Moreover, we will recommend the rendezvous point for them according to their intended activity (e.g., dining together) and the geo-context, and suggest a route.

Application III: Rendezvous point for photographing. This is an application useful for tourists. When visiting a city, people may turn to websites such as Flickr [6] or Panoramio [24] to browse for scenic spots, since there is substantial information shared by other tourists (e.g., photos, ratings, and comments). While browsing these photos, people may be attracted to some of them. They will think about where these pictures were taken and plan to visit the same spot to take a similar photo. Our system could help the user to find a the specific "rendezvous" point based on the scenery in a photograph. When the user is attracted by an online photo (e.g., when the user is browsing Flickr), he or she could use this mobile application to search for the location where this photo was taken. As described in Section 3.3, we can return the $\mathbf{R}, \mathbf{t}, \mathbf{K}$ of the photo to the user, which can be further translated to $(\mathbf{l}_p, \theta_p, \mathbf{l}_s)$. Other parameters can also be suggested to the user for photographing. Through this application, we can suggest the user: where to go ($\mathbf{l}_p$), what to photograph ($\mathbf{l}_s$), and the best camera pose (θ_p could further be decomposed to heading, tilt, and roll of the camera) and parameters (e.g, focal length) to capture a satisfying photo [4].

5. EXPERIMENTS

In this section, we first introduce the datasets and queries used for evaluation, and then show a series of experimental results of geo-visual clustering, scene reconstruction, single image localization, and multiple localization. Finally, we conduct user studies to evaluate the usability of our proposed accurate mobile visual localization system (AMVL).

5.1 Datasets

To evaluate the performance of AMVL, we conducted experiments on several challenging real-world datasets. We collected two types of datasets, which consist of images with geo-context information such as GPS, but have some differences in their modalities.

- San Francisco street view dataset (**SF street view**). This is a public dataset provided by Stanford and NAVTEQ[1]. It consists of street view photos in San Francisco captured by a vehicle with adequate cameras and other equipment like GPS [5]. There are about 1.06 million images in this database. A set of 803 queries collected by mobile phones with GPS is also provided. There are no other ground truth data, such as view direction, location of the scene, available for the queries. However these are essential to evaluate the performance of our AMVL. We have therefore selected a subset of 100 images from the given queries and manually labeled these images with the help of Bing Street Side [3] and Google Street Views [9].

- Images of five cities crawled from Flickr (**Flickr five cities**): This dataset consists of images crawled from the most popular photo-sharing site Flickr [6], which covers the urban area of five cities: Beijing, New York, Paris, San Francisco, and Seattle (Figure 8). Images in **Flickr five cities** contain diverse visual content. They are dense near some popular landmarks. Moreover, the GPS tags are much noisier than those in the **SF street view** dataset. To collect queries, we select the images from the 10 densest regions in the city, so that there are 50 queries in total. Similar to **SF street view**, these selected queries are also manually labeled with camera location, view direction, and scene location.

5.2 Objective Evaluation

Evaluation of joint geo-visual clustering. As mentioned in Section 3.1, we should cluster the images in the database into small clusters using geo-visual clustering. (1) For **SF street view** dataset, image locations are uniformly distributed so that it is much easier to get image clusters. For simplicity we quantize locations into small rectangular tiles with a radius of 50 meters leading to about 150 images per geo-cluster. Then, visual clustering is performed to each small geo-cluster to generate 5∼10 visual clusters with 15∼40 images in each. (2) For **Flickr five cities** dataset, as the images are more diverse in both geographical and visual space, in geo-clustering, we choose a radius $r_g = 700m$ to get larger geo-clusters. While for fast clustering a spatial tile with size $50m \times 50m$ is used instead of single image locations as the basic unit for geo-clustering. We build inverted index files in every geo-clusters so the visual similarity matrix can be computed quickly. Even with large geo-clusters(up to thousands of images in the largest geo-cluster), we can achieve very fast visual clustering. Finally we can get visual clusters, but there are many clusters that are single images due to the diversity of Flickr images. Once the clustering is finished, we can get the index of images to the cluster.

Evaluation of 3D scene model reconstruction. Images in the same visual cluster represent the same scene or similar views of the same scene. We build a 3D scene model for each visual cluster from these images. The scene models are 3D point clouds and camera parameters reconstructed using SfM. In our experiments, the bundler toolbox [30] is adopted to compute the reconstruction. As we have clustered the images into small groups, the reconstruction are conducted with around dozens of images, which is small compared to those used in the Photo Tourism system [30]. Thus the reconstruction requires low computational power and can be performed on a single PC. The average time for constructing a scene

[1] http://www.stanford.edu/~dmchen/mvs.html

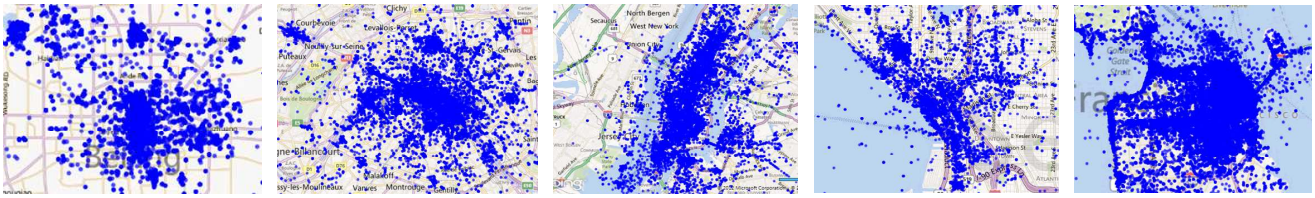
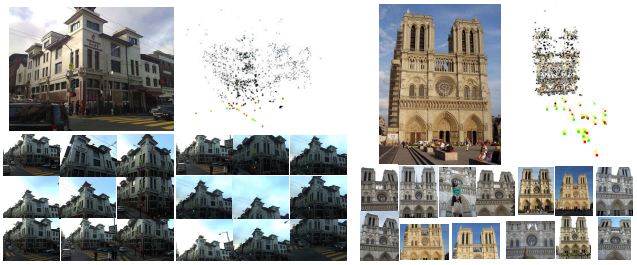


Figure 8: Geographical distribution of images of five cities crawled from Flickr are shown on the map. The five cities from left to right are: Beijing, Paris, New York, Seattle and San Francisco.



(a) A query image and it's retrieved 3D scene built from one geo-visual cluster in "SF street view" dataset.

(b) Query image and retrieved 3D scene of a landmark(Norte Dame de Paris) in "Flickr 5 cities" dataset.

Figure 9: The query image and retrieved 3D scene models built from images in geo-visual cluster.

is 2∼10 min since the clusters' size are usually 15∼40. We can reconstruct several scenes in parallel to further speed up the reconstruction. Figure 9 shows two 3D scenes for example.

Evaluation of self-localization using single image. We follow the procedure introduced in Section 3.2 to estimate the accurate geographic context of an image. First, SIFT features are extracted from the images. Then the features are quantized to visual words using the vocabulary tree method. As $B = 10, H = 6$ in our experiments, we use a tree with $1M$ leaf nodes. Inverted file and fast geometric verification are used to retrieve and filter candidate images. The filtered top 20 candidate images are used to vote for the most related scene. Then the query image is aligned to the 3D scene. We have evaluated the image retrieval and localization performance. We use mean Average Precision (mAP) as the evaluation criterion of our image retrieval system. Given $\mathcal{Q} = \{q_i\}$ denoting the query image set, the mAP at rank N is defined by:

$$mAP@N = \frac{1}{|\mathcal{Q}|}\sum_{i=1}^{|\mathcal{Q}|}\Big(\frac{\sum_{r=1}^{N} P_i(r)}{N}\Big), \qquad (10)$$

where $|\mathcal{Q}|$ is the number of queries and $P_i(r)$ is the precision of i-th query image at rank r. We have evaluate the precision of several image retrieval strategies. 1) *BoW*: state of the art Bag-of-Words method with TF-IDF weighting. 2) BoW+GPS: excluding faraway ($>$ 300m) candidates by coarse GPS of the query image. 3) BoW+GV: BoW with feature orientation quantization and spatial verification. Orientation quantization also embeds geometric information, so we relegate it into GV. 4) BoW+GV+GPS. The retrieval performance can be seen in Figure 10. It can be observed from the results that the BoW method, which merely considering the matched feature weight leads to the worst performance. Feature orientation and geometric verification are important in image retrieval, because the structure information provides additional knowledge. This is especially useful in the dataset of street views and landmark images, where the image content is highly structured and mostly gravity-aligned. GPS coordinates associated with the query image to constrain the search range could further enhance the retrieval precision.

After image retrieval, we can use the obtained candidate scenes

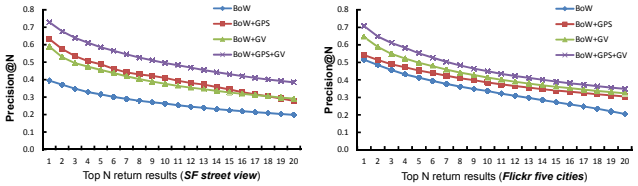


Figure 10: The retrieval performance on SF street view and Flickr five cities datasets with several retrieve strategies.

Table 1: The error statistics of estimated geographical information of queries in SF street view. AMVL is the proposed system in this paper. Comparison results implementing two other localization approaches are also shown in this table.

Errors		Camera location	View direction	Scene location	Phone GPS
AMVL	Mean	13.787m	8.963°	20.675m	30.344m
	Std.	25.844m	11.708°	26.089m	44.330m
VPS [27]	Mean	15.218m	9.531°	22.758m	-
	Std.	25.793m	12.967°	26.647m	-
BGPS-I [25]	Mean	-	19.864°	-	-
	Std.	-	22.836°	-	-
BGPS-II [25]	Mean	-	12.774°	-	-
	Std.	-	12.179°	-	-

models to perform view registration with the query images. The estimation errors of camera location, view direction and scene location are listed in Table 1 for **SF street view** and Table 2 for **Flickr five cities**. As there are images that cannot find the proper scenes or fail to align to the scene models, these results are calculated using the subset that successfully aligned to a scene model. In **SF street view**, 60 out of the total 100 query images are successfully aligned to a scene model. From the statistics we can see the proposed mobile visual localization system could reduce the mean localization error to 13.787 meters compared to the GPS data collected by mobile phone sensors, and the view direction estimation has an average mean error of 8.963 degrees. It can be seen from Figure 11 that most images achieve a high accuracy of localization. Notice that the estimated scene locations have a larger mean error than camera locations. That is probably because we estimate the scene location using the matched 3D points, while there may be 3D points from other buildings making the estimated scene align to further locations. The efficiency of the proposed system is evaluated and shown in Table 3. The total time cost for successfully localizing a 640×480 photo is 3.5∼6 seconds. Similar results of **Flickr five cities** can be seen in Table 2, where 37 out of 50 query images are successfully localized, but with greater mean errors and deviations. This was caused by the noisy GPS labels of Flickr images, which affected the estimation accuracy of similar transforms that map the camera to real-world coordinates.

We compared the localization performance with two other methods on the **SF street view** dataset. The first is the Vocabulary-

Table 2: The error statistics of estimated geographical information of queries in Flickr five cities

Errors	Camera location	View direction	Scene location	Labeled GPS
Mean	25.457m	14.547°	32.654m	50.487m
Std.	28.766m	13.876°	37.205m	66.325m

Table 3: Time complexity of the system (sec).

Operation	Uploading	Extract SIFT	Vocabulary tree search
Time	<0.2	0.7∼1.2	<0.3
Operation	Vote for 3D	Align to 3D	Register to map
Time	<0.02	2∼4	<0.1

based Prioritized Search (VPS) [27] which performs direct 2D-to-3D matching, while the second is Beyond GPS (BGPS) [25] where an approximation algorithm estimates the view direction of a geo-tagged photo using nearby street view images as references. For VPS, since it needs time to load descriptors, only the models within a certain radius (300m) of each query image are considered to conduct the 2D-to-3D matching. For BGPS, it requires the input image with given camera GPS and only estimates the view directions. We have evaluated the performance of BGPS for two settings: 1) with the noisy phone GPS as the input camera location (BGPS-I), 2) with groundtruth camera location (BGPS-II). Street view images within a certain peripheral area are used as references (the radius is 100m in our implementation). The experimental results are also shown in Table 1. From the results we can observe that the BGPS performance is good with the accurate camera location. However, it degenerates quickly when using noisy GPS coordinates. Our performance (13.787m) is comparable to VPS (15.218m), with even smaller mean error. This is probably because the accuracy of VPS is sensitive to some badly localized 3D points. However, we should point out that our system requires lower memory cost compared to VPS. The main memory cost of our system is the inverted table of quantized image features. Then a small 3D scene model voted on by image retrieval results is loaded into the memory for localization by view registration. For a database with about one million images the memory cost of our system is about 4 Gigabytes (Inverted Index) plus 10 megabytes (voted 3D model). Meanwhile, VPS needs to keep the 128-bytes descriptors for all 3D points in the memory. For a database with about 16K cameras, VPS requires more than 2 gigabytes. Thus, our system has better scalability than VPS.

Evaluation of collaborative localization. To examine the performance of localization of multiple images in collaboration, we use images in groups for evaluation. Images in the same group are taken within the same area of radius 300m. As discussed in Section 3.4, we use the combined feature vector to search whether the images have any overlap. We found this achieves a better localization for a group of images than and individual image. Figure 12 shows some group localization examples:

5.3 Usability Evaluation

We invited 12 subjects to use our AMVL system on Windows Phone 7 devices. The subjects include three female and nine male college students, with the ages ranging from 22 to 28. Half of them were undergraduates and the rest were graduate. Most of them had 1∼3 years of experience of using smart mobile phones and localization applications such as Google Maps [8], while two had a longer experience of 3∼5 years and two had never used such applications on mobile phones. After a 5-min orientation and demonstration, all of them knew how to use this application. Some of

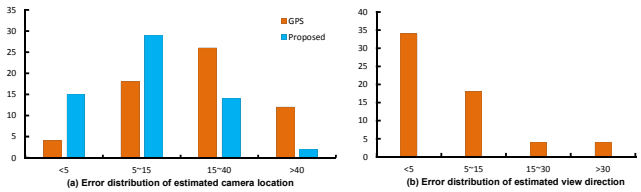


Figure 11: The error distributions of SF street view queries. (a) the error of estimated camera locations using propose framework compared to data collected from mobile phone sensors, (b) the estimated view direction errors.

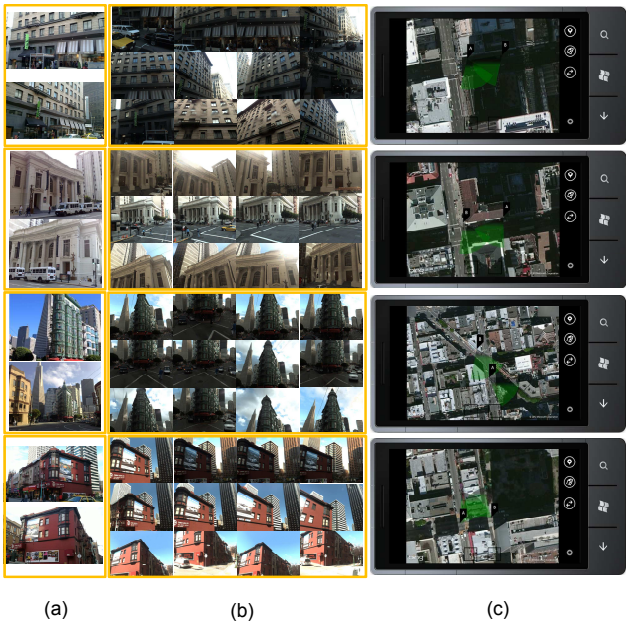


Figure 12: Examples of multi-image co-localization: (a) pair of query images; (b) images of the retrieved scene; (c) collaborative localization results, on a shared map.

them liked the augmented reality of showing local business along their viewing direction on the screen. Compared with having only their location and nearby local business shown on the map, they agreed that their intent can be expressed more clearly with our system.

After learning to use the system, the users were asked to use the application to accomplish the following tasks:

- The user is in the street and lost his/her way. Given a street view photo captured by the phone and the GPS data from the sensor, the user should try to obtain his/her location and orientation. Then the user is encouraged to find a way to the restaurant/café/bar in front of him/her.
- The user and his/her friends each capture a photo in front of them then log in to our system, share their locations, directions, and pictures with their friends on the same map. They can find a way to meet in the crowded area.

As our system has not been released to the public yet, the subjects were unable to conduct the tasks under actual circumstances. Thus we simulated 100 locations in San Francisco.. Meanwhile, the street view images queried with the estimated geo-context were shown to the user to check the localization accuracy. The users felt it was easy to fulfill these tasks and found AMVL successfully facilitates their intent. A questionnaire was also filled out by each

Table 4: A summary of the user study. Each metric is rated from 1 to 5 indicating the worst to the best level.

ID	Questions	Satisfaction
1	Attractiveness&Naturalness	3.7
2	Efficiency	3.3
3	Clarity of intent	3.8
4	Practicality	3.8
5	Ease to use	4.2
6	Finding Rendezvous point with friends	4.4
7	Preference/Recommendation	4.3

participant to evaluate the usability, user friendliness, and experience. The quantitative evaluation of users' satisfaction with the mobile visual localization system is shown in Table 4. As can be seen from the results, the users found our application attractive and easy to use with a friendly user interface. Some subjects were satisfied with the relevant local businesses shown on the screen. They agreed that this can assist them to sense their surroundings better. 91.7% of the subjects thought the multi-user system where one can share photo and geo-context with friends was quite useful, especially when they were going to meet in an unfamiliar place or were separated by a crowd. Most of them gave a positive response when asked whether they will install this application and recommend it to their friends. Moreover, they were asked about new features from our system. Some of them were concerned about the latency in the 3G wireless network environment, as it takes $3.5\sim6$ sec in our WiFi-based system. Some of them noticed that when the direction estimation was incorrect, it could mislead the user. They thought this could be used together with GPS and other sensors for better results.

6. DISCUSSION AND CONCLUSIONS

This paper presents a mobile visual localization system that achieve a comprehensive set of geo-context information. Compared to existing visual localization approaches, the proposed system combines advanced techniques of large-scale image retrieval, 3D model reconstruction, and localization by 2D-to-3D matching. It can achieve high localization accuracy and good scalability. The experiments show that the proposed system can provide more accurate and complete geo-context to the user. A variety of applications show the improved user experience in location-based services. However, it may fail when the query image is not able to find accurate candidate images, or the related images are unable to reconstruct a 3D scene model with enough points.

Our future works include: 1) helping the user to find distinctive views, which will enhance the accuracy of visual-based retrieval to make our system more robust and reliable, and 2) leveraging compressed representation of images to reduce the system latency to enable real-time localization.

7. ACKNOWLEDGEMENT

This work is partly supported by NSFC general project fund "Intelligent Video Processing and Coding Based on Cloud Computing."

8. REFERENCES

[1] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *JACM*, 1998.

[2] Y. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *ACM Multimedia*, 2010.

[3] Bing street side. `http://www.microsoft.com/maps/streetside.aspx`.

[4] S. Bourke, K. McCarthy, and B. Smyth. The social camera: a case-study in contextual image recommendation. In *IUI*, 2011.

[5] D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR*, 2011.

[6] Flickr. `http://www.flickr.com/`.

[7] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[8] Google Earth. `http://earth.google.com/`.

[9] Google street view. `http://www.google.com/streetview`.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004.

[11] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[12] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009.

[13] R. Ji, L. Duan, J. Chen, H. Yao, Y. Rui, S. Chang, and W. Gao. Towards low bit rate mobile visual search with multiple-channel coding. In *ACM Multimedia*, 2011.

[14] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *CVPR*, 2009.

[15] M. Kroepfl, Y. Wexler, and E. Ofek. Efficiently locating photographs in many panoramas. In *GIS*, 2010.

[16] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008.

[17] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010.

[18] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, 2008.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[20] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision: a survey. *MTA*, 2011.

[21] Z. Luo, H. Li, J. Tang, R. Hong, and T. Chua. Viewfocus: explore places of interests on google maps using photos with view direction filtering. In *ACM Multimedia*, 2009.

[22] D. Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 2004.

[23] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[24] Panoramio. `http://www.panoramio.com/`.

[25] M. Park, J. Luo, R. Collins, and Y. Liu. Beyond gps: determining the camera viewing direction of a geotagged image. In *ACM Multimedia*, 2010.

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[27] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011.

[28] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007.

[29] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach. Mobile visual location recognition. *Signal Processing Magazine*, 2011.

[30] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *TOG*, 2006.

[31] F. Yu, R. Ji, and S. Chang. Active query sensing for mobile location search. In *ACM Multimedia*, 2011.

[32] A. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010.

[33] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.

[34] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM Multimedia*, 2010.