AMIGO: Accurate Mobile Image Geotagging *

Xiaobin Xu [1], Tao Mei [2], Wenjun Zeng [3], Nenghai Yu [1], Jiebo Luo [4]
[1] University of Science and Technology of China, China [2] Microsoft Research Asia, China
[3] University of Missouri, USA [4] University of Rochester, USA
xxb263@mail.ustc.edu.cn, tmei@microsoft.com, zengw@missouri.edu
ynh@ustc.edu.cn, jluo@cs.rochester.edu

ABSTRACT

With location-based services gaining popularity among mobile users, researchers are exploring the way using the phone-captured image for localization as it contains more context information than the embedded sensory GPS coordinates. We present in this paper a novel mobile image geotagging approach to accurately sense the actual geo-context of a mobile user. The proposed approach, named *AMIGO* (Accurate Mobile Image GeOtagging), is able to provide a comprehensive set of accurate geo-context based on the current image and its associated scene in the database. The geo-context includes the real locations of a mobile user and the scene, the viewing angle, and the distance between the user and the scene. Specifically, we first perform partial duplicate image retrieval to select crowdsourced images capturing the same scene as the query image. We then employ the structure-from-motion technique to reconstruct a sparse 3D point cloud of the scene. Finally, by projecting the reconstructed scene onto the horizontal plane, we can derive user's location, viewing angle, and distance. The effectiveness of AMIGO has been validated by experimental results.

Categories and Subject Descriptors

I.4.8 [**Computing Methodologies**]: Image Processing and Computer Vision—*Scene Analysis*

General Terms

Algorithms, Performance, Experimentation.

Keywords

Geotagged image, mobile image localization, scene reconstruction, location-based services.

1. INTRODUCTION

With the development of mobile communication technologies and the popularity of mobile Internet, people have increasingly tended to use portable mobile devices such as mobile phones, digital cameras, personal digital assistants (PDAs) for entertainment and communication. The recent development of location-based services (LBS) is an important manifestation of this trend. Examples include searching for nearby social events and friends, location-based advertising, mobile recommendation for nearby foods and restaurants, and so on [14]. Localization is the key problem of LBS. Only when the accurate location of a mobile device is found, all other LBS applications can be applicable.

However, the traditional methods for localization, such as Cell-ID based positioning and GPS positioning, etc., are inadequate in terms of accuracy and reliability. Reports have shown that the average localization errors of mobile phone GPS sensors are in the range of 50-100 meters [9]. The errors are caused by the requirement of at least four satellites to be visible at the same time, as well as the multipath effects which are severe in urban areas with crowded tall buildings. High-end smart phones may also carry a digital compass to detect user's viewing angle, but it is sensitive to motion and magnetic disturbances. On the other hand, with the popularity of photo-sharing communities on the Internet in recent years, a huge number of geo-tagged images are available on the Internet (e.g., Flickr). In particular, we are interested in utilizing these crowdsourced geo-tagged images to facilitate the inference of accurate knowledge about geo-context (such as position, viewing angle, distance, etc.) for LBS applications.

We develop an approach called AMIGO for accurate mobile image localization. AMIGO only requires a query image taken by the mobile user as input. The challenges of AMIGO originate from the following aspects: 1) query images taken by mobile users may be subject to occlusions, overlaps, various lighting conditions, and motion blur, which makes it difficult to match the query image against the images in the database; 2) GPS tags of the reference images collected from photo-sharing community may be inaccurate and inconsistent; and 3) dynamic objects and weather changes require the selection of robust features for both query image and reference images. AMIGO can provide the accurate estimation of a comprehensive set of geo-context, such as user location, viewing angle, and scene location, which is different from most existing approaches to mobile image localization capable to predict a subset of them [2, 3, 5, 7, 8].

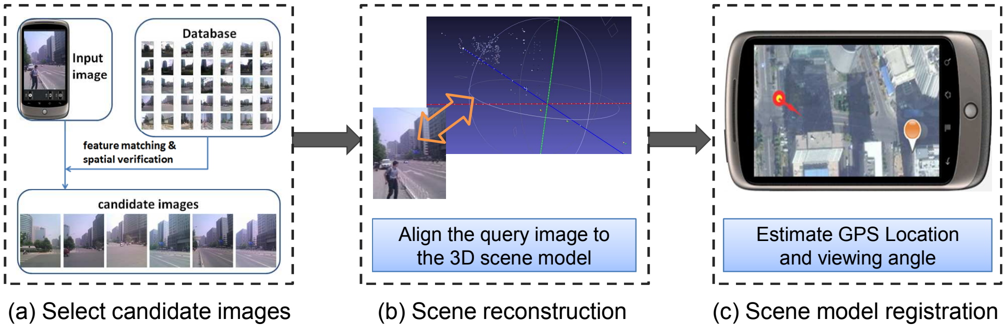
The remainder of this paper is organized as follows. Section 2 provides a brief review of the related work. Our proposed AMIGO is described in Section 3. Section 4 presents experiments, followed by conclusions in Section 6.

2. RELATED WORK

More and more researchers are trying to utilize the massive geotagged images shared on the Internet to infer location related properties for new images. In this section we present some representative works.

* This work was performed at Microsoft Research Asia.

ICIMCS'12, September 9–11, Wuhan, Hubei, China.
Copyright 2012 ACM 978-1-4503-1600-2/12/09 ...$15.00.


Input image
Database
feature matching & spatial verification
candidate images
Align the query image to the 3D scene model
Estimate GPS Location and viewing angle

(a) Select candidate images (b) Scene reconstruction (c) Scene model registration

Figure 1: Overview of the AMIGO system: a) a photo taken by the user is used to retrieve partial duplicate images from the database; b) the candidate images are used to reconstruct a sparse 3D point cloud of the scene, and the query image is then aligned to the scene model; c) the estimated locations and viewing angle are projected onto the horizontal plane and mapped to real-world coordinate through a similarity transform, after which the results are displayed on a bird's eye view 3D map.

Image location estimation: Hays *et al.* estimated a probability distribution of the image location by mean-shift clustering of visual descriptors [3]. They made it to locate 25% of the images to within 750 km. Gallagher *et al.* further improved the prediction accuracy by taking into account the textual tags of images [2]. Schindler *et al.* studied the repetitive patterns of textured building facades in a city and built a GPS-tagged database to facilitate repeated pattern matching and recovery of camera orientation and location [8]. Their method relies on a pre-existing database of geolocated planar facades covering the whole area which is hard to build. Moreover, their method needs input images containing multiple buildings that exhibit highly repetitive structure.

Viewing angle estimation: Luo *et al.* proposed a method to find photos with viewing angles pointing to the user-indicated region [5]. They retrieved photos within a range of certain radius to the given geo-location, and clustered the retrieved photos into a set of subsets representing different scenes. Then they used normalized 8-point algorithm to estimate the camera pose of the scenes expressed by those subsets. A creative solution was proposed by Park *et al.* in [7], where they took advantage of Google Street View and Google Earth satellite images to estimate viewing angle for potentially all images above the earth. However, all this work of viewing angle estimation requires a given geo-location as input and has not dealt with the estimation of GPS locations.

In contrast, our proposed method is designed to accurately locate the phone-captured photos in urban areas and does not need any initial geo-locations as input. The contribution of our work is two-fold: 1) we propose a novel geotagging approach called AMIGO which can accurately sense the real geo-context of a mobile user; 2) we propose a practical method for registering the sparse 3D point cloud of the scene on the real map by robustly calculating a similarity matrix. We perform experiments to show the effectiveness of the proposed approach.

3. AMIGO

As shown in Fig.1, our proposed AMIGO consists of three stages: candidate images retrieval, scene reconstruction and query image alignment, and mapping to real geo-context. First, a photo taken by the user is used to retrieve partial duplicate images from the database. Then the candidate images are used to reconstruct a sparse 3D point cloud of the scene and the query image is aligned to the scene model. Finally, the estimated location and viewing angle are projected onto the horizontal plane and mapped to real world GPS location through a similarity transform, after which the results are displayed on a bird's eye view 3D map.

3.1 Candidate Images Retrieval

To effectively retrieve partial duplicate geo-tagged images from the massive database, we need an efficient and robust indexing and retrieving scheme. A commonly adopted scheme extracts local image features, quantizes their descriptors into visual words, and applies methods from text search for image retrieval. In our experiments, we use the vocabulary tree proposed by Nister *et al.* to quantize and index scale-invariant feature transform (SIFT) features extracted from the database images [4, 6]. SIFT feature is the state-of-the-art local descriptor, it is widely used in content-based image retrieval (CBIR) systems due to its distinctiveness, robustness, being abundant and computationally efficient.

Typically, an image can have thousands of features. It is a huge amount when searching from a large database. Vocabulary tree can significantly reduce the searching time by adopting a hierarchical quantization structure to quantize each feature into a visual word. After quantizing and representing the images by visual words, we can utilize the inverted file indexing technology and TF-IDF model in text retrieval to evaluate the similarity and retrieve visually similar images efficiently. In practice, we use the method recommended in [6] to implement this process.

The retrieved images from the above process are visually similar images, but what we specifically want is partial duplicate images. Due to the sampling and quantization which are lossy processes during the construction of the vocabulary tree, the retrieved images sometimes can vary significantly. To single out those true partial duplicate images, other features of the images can be imposed on the retrieved images. In particular, repetitive patterns of textured building facades are common in street view images which makes it indispensable to enforce spatial constraints between the input image and the retrieved images, as multiple descriptors of similar repetitive patterns may be quantized into the same visual word. Spatial constraints are taken into account in our system through spatial coding method proposed by Zhou *et al.* [13].

By encoding spatial relationships among local features, we can reject false matches between images and re-rank the retrieved similar images. After re-ranking, we examine the top 20 results. These results are then clustered according to their GPS metadata and a distance-based similarity metric: any two images located within 500 meters to each other belong to the same cluster. If there are more than five reference images in the largest cluster, they are accepted as the candidate images.

3.2 Scene Reconstruction

After the candidate images are found, we utilize them along with their geo-tags to estimate the location of our input image. It would be good if we can recover the relative position of the cameras which photographed these images because the geo-tags related to the candidate images is actually the position of the cameras. Then we can utilize the relative position to obtain the actual geo-location of the input image.

The task of accurately calculating the pose of cameras and scene structure is referred to as the structure from motion (SfM) problem. Although the theory of SfM problem has been mature for a long time, the reconstruction of scene structure from unordered image set is not feasible in practice until recently. Moreover, Snavely *et al*. demonstrated the power of SfM techniques by reconstruction from a large set of Internet images [10]. We use the bundler package developed by Snavely to obtain sparse 3D points of the scene and camera pose from the candidate images retrieved in Section 3.1.

The reconstructed scene may not be positioned “upright” as we expected. So we calculate the upright vector of the scene according to the method proposed by Szeliski [11]. Observing that people usually take photographs with the horizontal edge of their cameras parallel to the ground plane, we can assume that all cameras’ horizontal axis is perpendicular to the scene’s vertical axis along which the upright vector lies. By enforcing this constraint on every candidate image we can formulate the estimation of upright vector as a least squares problem. After the upright vector is calculated, we rotate the reconstructed scene such that the upright vector is positioned along the vertical direction. Then we align the input image to the 3D scene model by adding it into the reconstruction.

3.3 Location and Viewing Angle Estimation

Calculating the Similarity Matrix. The relationship of the reconstructed scene model and the real world scene can be approximately described by a similarity transform. Since we have already obtained the upright vector of the scene, only the horizontal projection of the scene needs to be considered. If we denote the horizontal projection of camera location in the model as (x, y), and the corresponding GPS coordinates of the camera (i.e. the geo-tag of the image that the camera captures) as (G^{lon}, G^{lat}), then the transformation is uniquely determined by a similarity matrix:

$$\begin{pmatrix} G^{lon} \\ G^{lat} \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

where the similarity transform matrix is composed of the scale factor s, the rotation angle θ and the translation vector (t_x, t_y). But we do not need to calculate them, so we rewrite the equation as:

$$\begin{pmatrix} G^{lon} \\ G^{lat} \\ 1 \end{pmatrix} = \begin{bmatrix} u & -v & t_x \\ v & u & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2}$$

This equation will generate two linear constraints on the similarity matrix. Since the matrix has four degrees of freedom, we need at least two similarity correspondences to determine it. That means ideally two candidate images from Section 3.1 are required. And the solution is simply:

$$\begin{pmatrix} u \\ v \\ t_x \\ t_y \end{pmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} G_1^{lon} \\ G_1^{lat} \\ G_2^{lon} \\ G_2^{lat} \end{pmatrix} \tag{3}$$

When there exists more than two candidate images, we can use linear least-squares technique to compute the similarity matrix.

Besides, the GPS information of images may be noisy. To avoid the influence of images with potentially incorrect GPS locations, we apply the classical RANSAC method to estimate the similarity transform such that the outliers would be rejected to obtain more accurate results. After the RANSAC procedure, the transform is re-estimated using all the inliers.

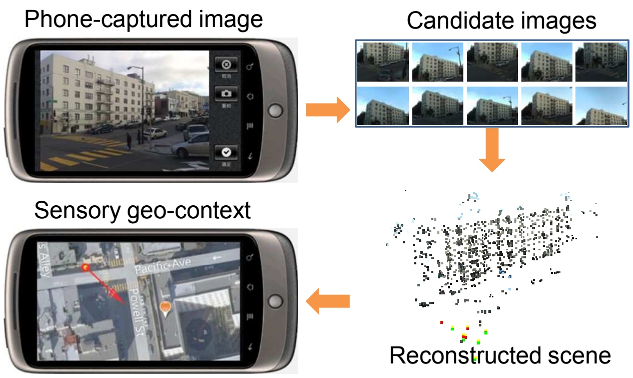


Figure 2: Visual results for each step in AMIGO.

Registering Images on the Map. The image location and viewing angle can be obtained by simply transforming the camera location and viewing angle in the model using the similarity matrix we just calculated.

For camera locations, by simply using Eq.(1) we can get the location of input image and refine the location of candidate images in the database.

Under the pin hole camera model, the viewing angle of each camera under the reconstructed coordinate can be obtained using the following formula:

$$V_p = R' \times \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}' \tag{4}$$

where V_p is the viewing angle of a camera, R is the rotation matrix of this camera estimated by the bundler package. As the similar transform would not change the angles between the cameras, we can also map the viewing angle of the camera to the real world coordinate using the same similarity matrix. It should be noted that we first project this 3D vector of the viewing angle to the 2D horizontal plane that is perpendicular to the upright vector.

Furthermore, we find that under our AMIGO framework, we can also get some useful information about the scene. By analyzing the relation of 3D points of reconstructed scene and the cameras, the location of the scene captured by each camera and the distance from the location where the user stood to the scene of the photo could be estimated. In our experiment, we calculate the scene location as the center of all feature matches’ locations.

4. EXPERIMENTS

4.1 Dataset

We used the San Francisco dataset provided by Chen *et al.* [1] in our experiment. This is a publicly available city-scale dataset with two types of images, *perspective central images* (PCIs) and *perspective frontal images* (PFIs). There are $1.06M$ PCIs and $638K$ PFIs in total. We only used the PCIs considering that the PFIs contain distortion that may cause errors in the reconstruction stage. The San Francisco dataset also provides a set of queries (803 in total) with ground truth or simulated GPS locations. These query images are taken from a pedestrian’s perspective at street level using several different camera phones by various people, so they are suitable for our purpose. However, the GPS locations provided for the query images are noisy and thus cannot be used directly as the

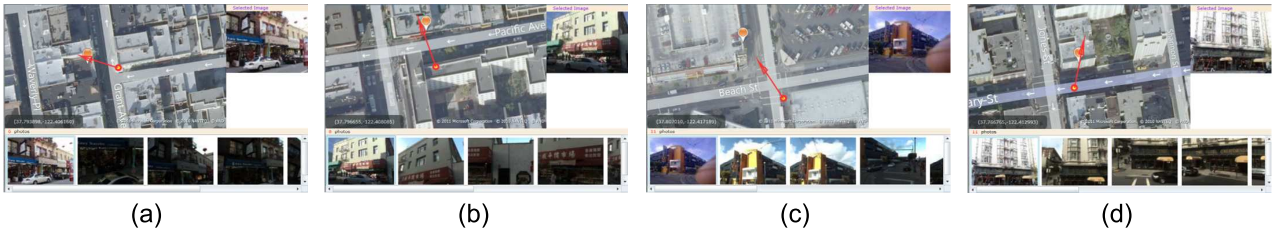


Figure 3: Example results of our system on the San Francisco dataset. The arrow represents the estimated viewing angle, and the circle at the starting point of the arrow indicates the estimated user location. The pushpin shows the location of the related scene. The bottom row shows the query images and their associated reference images used in the scene reconstruction.

Table 1: Error statistics of geo-context sensed by AMIGO

	User location	Scene location	Viewing angle	GPS sensor location
Error	10.536 m	14.050 m	21.442°	38.632 m
Std.	14.806 m	19.883 m	34.381°	49.554 m

ground truth. Moreover, the scene location and viewing angle are not provided. In order to evaluate AMIGO, we manually labeled all the query images using Google Street View [1].

4.2 Examples

For each query image, the candidate images are first retrieved by the CBIR system with fast spatial verification. Then, we cluster the top 20 retrieved images to gather those images located within 500 meters with each other into a cluster. If there are more than five reference images in the largest cluster, the bundler package is called to reconstruct the scene and camera parameters. The similarity transform is estimated using RANSAC. Finally the user location, viewing angle and location of the related scene are shown on a bird's eye view map. We show a real-world example in Fig.2 illustrating each step in our AMIGO system. We also present several typical results in Fig.3. It can be easily seen that our system is robust under occlusion, overlap, and various lighting conditions.

4.3 Evaluations

We find that among the 803 query images, there are 457 queries that could find more than five images in the largest cluster. We test these 457 images for the subsequent pipeline of the proposed algorithm and find that 282 of them are not aligned to the reconstructed scene. The error distribution of the 175 query images that can successfully sense the geo-context is shown in Fig.4. The result shows that 93.7% of the user locations can be accurately sensed with an error less than 20 meters. Table 1 presents the prediction accuracy of each geo-context sensed by AMIGO. The error statistics of corresponding locations obtained from the phone GPS sensors are also shown. For these query images, our AMIGO system outperforms the GPS sensors built in the phones.

5. CONCLUSIONS

We have proposed a novel approach named AMIGO for accurately sensing the geo-context of mobile users by utilizing the massive unordered geo-tagged pictures from crowd sources. Although our experiments have shown promising results, we observe that for most places in the world, the geo-tagged pictures cannot form a dense coverage. Our future work includes more practical solutions for speeding up the 3D scene reconstruction in the cloud and suggesting the best view for scene recognition like [12].

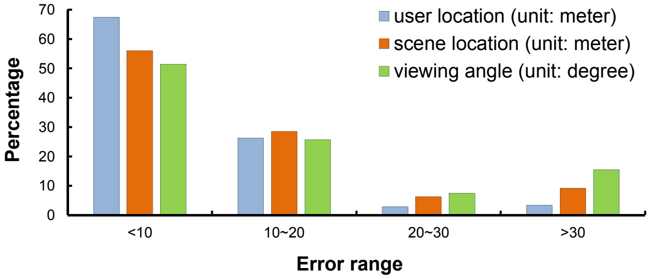


Figure 4: Error distribution of estimated geo-context.

6. ACKNOWLEDGEMENT

This work is partly supported by National Science and Technology Major Project (No. 2010ZX03004-003).

7. REFERENCES

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

[2] A. Gallagher, D. Joshi, J. Yu, and J. Luo. Geo-location inference from image content and user tags. *CVPR Workshops*, pages 55–62, 2009.

[3] J. Hays and A. Efros. IM2GPS: estimating geographic information from a single image. In *CVPR*, 2008.

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

[5] Z. Luo, H. Li, J. Tang, R. Hong, and T. Chua. Estimating poses of world's photos with geographic metadata. *Advances in Multimedia Modeling*, pages 695–700, 2010.

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

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

[8] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008.

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

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

[11] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

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

[13] 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.

[14] J. Zhuang, T. Mei, S. C. H. Hoi, Y.-Q. Xu, and S. Li. When recommendation meets mobile: contextual and personalized recommendation on the go. In *Proceedings of ACM UBICOMP*, pages 153–162, 2011.

[1] http://www.google.com/streetview