2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

Image Based Localization with Sparse Database Using Panning Query Images

Tetsuo Inoshita[†], Akihiko Iketani[†], Shuji Senda[††] and Takashi Shibata[†]
[†]Information and Media Processing Labs., [††]C&C Innovation Initiative
NEC Corporation
l753 Shimonumabe, Nakahara-Ku, Kawasaki, 211-0011, Japan
{t-inoshita@ak, iketani@cp, s-senda@ap, t-shibata@hw}.jp.nec.com

***Abstract*—This paper presents a novel image based localization system which only requires a small database with sparsely-captured images. In the proposed system, a query is a set of consecutive images captured while the user pans the camera. In contrast to a one-shot query used in previous methods, this makes it easier to find a corresponding image in the sparsely-captured images. This, however, does not guarantee that an exact match is found, i.e. there exists a certain residual between the actual location of the query and that of the matched image. In order to compensate for the residual, offset in user's location that best describes the transformation between the two images is estimated, and used to correct the location. The proposed system was evaluated in a real shopping mall. First, a collection of images was captured at 30 different points, each of which is at least 10 meters apart from others. Then, a set of images were randomly selected for each point and registered to a database. When 9 images were registered for each point in the database, the system succeeded in estimating user's location with the average error of 2.6 meters in approximately 90% of the area. This result shows the system is capable of estimating accurate localization, even with a sparse database.**

Keyword—Image based localization; sparse database; panning images

I. INTRODUCTION

Image based localization has recently drawn broad attention as an alternative to conventional positioning technologies [1]. Previous methods search a database for an image that best matches the query image captured by a user, and report the location associated to the retrieved image. In order to achieve high accuracy in localization, it is crucial to construct a database with densely-captured images [2]. However, this comes to a great cost when there is a change in the environment, since images have to be densely recaptured.

A simple solution to this cost issue in database construction is to use a database with less numbers of images captured in sparse locations. This, however, results in less-accurate localization, since it becomes more difficult to find an image in the database that matches the query, and moreover, even if a match is found, there exists a certain residual between the actual location of the query and that of the matched image. In order to compensate for this residual, Werner et al. proposed a localization system with location correction based on scale change between the query and matched images [3]. The system first retrieves an image that best matches the query, and then the scale change between the two images is calculated, and is used to estimate the offset in location between the matched and query images. The system was evaluated in a corridor scene, and succeeded in estimating user's location with the average error of 1.3 meters. This system, however, assumes that users only move along a one-dimensional path in constant direction, thus only estimates the position along the path and does not estimate the user's viewing direction. Thus, the method cannot be applied to localization in two-dimensional area, e.g. a shopping mall, etc.

This paper proposes a two-dimensional localization system based on a small database with sparsely-captured images. In the proposed system, a query is a set of consecutive images captured while the user pans the camera. In contrast to a one-shot query used in previous methods, this makes it easier to retrieve a corresponding image in the sparsely-captured images. This, however, does not guarantee that an exact match is found. In order to compensate for the residual in location, the system first computes the perspective transformation between the two images. Then, the system estimates the residual in user's viewing direction from that associated to the retrieved image. Finally, the residual in the position along the viewing direction is estimated, and the system reports the corrected location of the user in two-dimensional space.

II. PROPOSED SYSTEM

A. System Overview

The proposed system is composed of an image database, client terminals with built-in cameras, and a server which is connected to the client over 3G wireless networks. The database is composed of a collection of images that were captured at sparse locations in the target environment. Each image is associated with location data, which consists of three elements: 1) three-dimensional location in longitude, latitude and height (in this case, floor number), 2) direction of the camera when the image was captured, and 3) the distance between the location and the dominant object in the image, e.g. walls, signs, etc. The user captures a set of consecutive images as a query by panning the client. For each query image, the client extracts *feature points,* which are the corners where multiple edges intersect, and transmits *feature vectors* to the

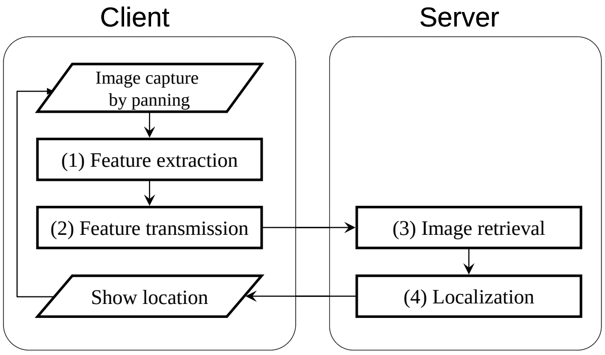


Figure 1. Flow of proposed localization algorithm

server, which describe the appearances of those points and their positions in the image. Then, the server retrieves a so-called *reference image*, an image that best matches the query image, by comparing the feature vectors in the query image with those for the images in the database. Then, the server estimates the user's location based on the location data associated to the retrieved reference image, and reports the estimated location to the client. Note that for each image in the database, its feature vectors are precomputed offline in order to save computational cost in the image retrieval process.

In the following subsection, the localization algorithm is described in detail.

B. *Localization Algorithm*

The proposed localization algorithm is composed of four steps, as shown in Figure 1. Each step will be described below.

1) FeatureExtraction

The client first extracts feature vectors from the query images captured by panning, which will be used for user location estimation in the server. Considering the limited bandwidth of 3G network, it is neither feasible nor practical to transmit all the feature vectors for the whole query images to the server. Thus, in the proposed system, instead of transmitting the feature vectors for every frame, they are transmitted only when sufficient numbers of new feature points appeared in the image. In addition, assuming that the appearance of the feature points remains the same throughout the whole query sequence, full-length feature vectors will be transmitted only once when the points first appeared in the image. For the rest of the frames, only the positions of the points are transmitted to the server. In the proposed algorithm, feature points are extracted by FAST corner detection [4], and their feature vectors are computed using BRIEF descriptor [5]. Whether a feature point is a new one that first appeared in the query images or not is determined by a simple feature tracking combined with outlier detection based on RANSAC algorithm [6]. If the feature point has no corresponding point in the previous frame, the point is treated as a new one.

2) Feature Transmission

For new feature points that first appeared in the frame, full-length feature vectors composed of both the appearance and position information are transmitted. On the other hand, for those points that have been tracked from the previous frame, only the position information is transmitted, since the appearance information has already been transmitted to the server in the past frames. Note that for the latter points, the full-length feature vectors are reconstructed in the server by coupling the position information with the appearance information stored in the server.

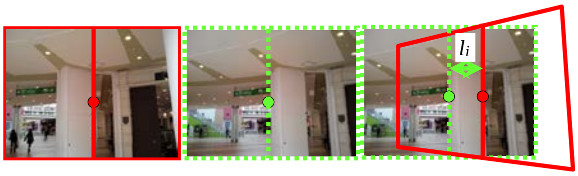


Figure 2. Viewing direction estimation. The left shows the query image. The center shows the reference image in the database. The right shows the projection of the query image to the reference image using the homography parameters (red rectangle).

3) Image Retrieval

The reference frame that best matches the query image is retrieved from the database. Here, a simple voting scheme is employed. For each feature points in the query image, the distance in the feature vector space is computed against all the feature points in the images stored in the database. The image in the database gets voted every time its feature point is selected as the closest point to the query point. This process is repeated for all the points in the query, and the image that acquired the highest votes is retrieved as the reference image.

4) Localization

In general, the location where the retrieved reference image was captured does not exactly coincide with the location where the query image is captured. Thus, a process for correcting the location associated to the reference image is required. Here, we propose a two-step correction algorithm; first, the difference in the viewing direction is corrected, and then the user's location along the corrected direction is estimated. The detail in each step is described below.

a) Viewing direction estimation

First, the perspective transformation between the query and the reference images, which is represented by the homography parameters, is computed using the correspondence of the feature points obtained in image retrieval process. Then, as depicted in Figure 2, the image center of the query image is projected to the reference frame using the homography parameters, and the horizontal distance l_i between the image center of the reference frame and the projected center is computed. Finally, assuming that the residual in the viewing direction is proportional to l_i , the residual θ_d is obtained by $\theta_d \approx k_i l_i$ .Here, the coefficient k_i was empirically determined to $k_i = 0.087$.

b) Location estimation

Next, the user's location along the estimated viewing direction is estimated. This process makes use of the fact that the further the user moves from the object in the scene, the smaller it appears in the image, and vice versa. Let us consider Z axis along the viewing direction with its origin at the dominant object in the image, as illustrated in Figure 3.

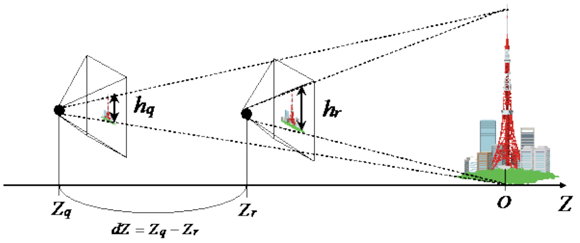


Figure 3. Location estimation

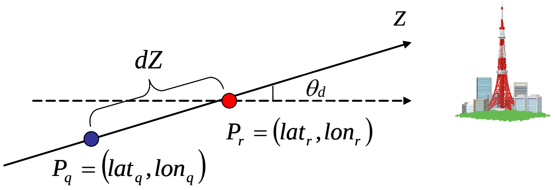


Figure 4. Relationship between estimated direction, position and spherical coordinates (longitude, latitude).

Here, the position of the optical centers of the cameras when they captured the reference and query images are depicted as Z_r and Z_q. The relationship between the offset dZ and the heights of the object in both images h_r and h_q are given as follows:

$$dZ = Z_q - Z_r \approx Z_r\left(h_q / h_r - 1\right).$$

Note that Z_r is obtained from the location data registered in the database in advance, as described in section II *A*. The scale ratio h_q / h_r is calculated from the homography parameters between the two images.

Finally, the user's two-dimensional location in spherical coordinates, i.e. longitude and latitude, is calculated. The geometric relationship among θ_d , dZ and the spherical coordinates $P_r = \left(lat_r, lon_r\right)$ and $P_q = \left(lat_q, lon_q\right)$, for the reference and the query, respectively, is illustrated in Figure 4. Based on this relationship, P_q , the coordinates for the query, or in other words, the user's current location, is given by the following equations;

$$lat_q = \alpha \cos\left(\theta_d\right) \cdot dZ + lat_r \text{ ,}$$

$$lon_q = \beta \sin\left(\theta_d\right) \cdot dZ + lon_r \text{ .}$$

Note that α and β are the coefficients for the unit conversion from meter to longitude/latitude. In our experiments, they are set to $\alpha = 0.91 \times 10^{-5}$, $\beta = 1.11 \times 10^{-5}$, respectively.

Since multiple queries are transmitted to the server while the user pans the client, multiple locations are obtained for the same points. In this system, the location which had the largest and sufficient number of feature correspondence in the image retrieval process was chosen, otherwise the system reports the localization has failed.

III. Experiments

In order to evaluate the performance of the proposed system, experiments were performed in a real shopping mall, Mitsui Shopping mall LaLaport KASHIWANOHA, Japan [7]. This section first describes the data acquisition for the experiments, and then presents the results.

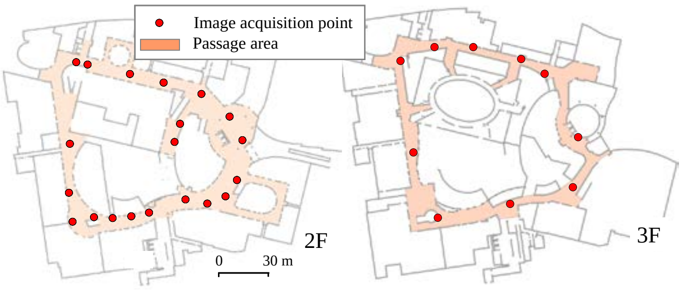
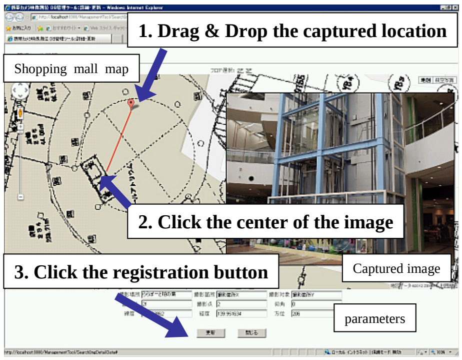


Figure 5. Points where reference images were captured


1. Drag & Drop the captured location

Shopping mall map

2. Click the center of the image

3. Click the registration button

Captured image

parameters

Figure 6. Database management tool

A. Database Construction

Images in various directions at sparse locations were captured. Here, in order to capture images with equal angular intervals, a camera and a compass attached to a tripod was employed. As shown in Figure 5, images were captured at 30 different points, each of which is at least 10 meters apart from others. At each point, the camera was rotated on the tripod, and for every 10 degree interval, an image was captured.

In order to register the location data efficiently, a web-based database management tool was developed (Figure 6). When the operator specifies on the map the location where an image was captured and the dominant object in image, the tool automatically calculates the distance to the object and the viewing direction, and registers them to the database.

B. Experimental Results

Query images were captured by the client, and the error in the user's location estimated by the server was evaluated. The server is a PC with XEON 2.5GHz CPU and 16GB memory. The client is a mobile phone NEC MEDIAS N-01D with 1.4GHz CPU, 1GB memory and a built-in camera, and was handheld and panned by the user to capture query images at 25 different points, each of which is at least 2 or 3 meters apart from the others, as shown in Figure 7. These points were chosen on a typical shopping route.

Since the accuracy of the localization is expected to depend on the size of the database, experiments were performed on databases with different configurations; the number of images registered for each data acquisition point varied from 1 to 17, and for each configuration, the specified number of images

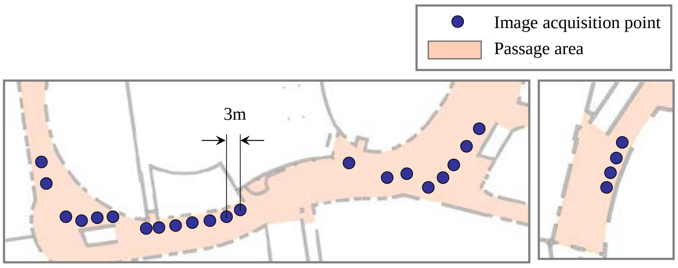


Figure 7. Points where query images were captured

were randomly sampled and registered to the database. Note that for each configuration, we generated 10 different databases and the errors for all the databases were averaged for each configuration.

Figure 8 shows the error in the location estimated by the proposed system, along with those by the systems without the location correction, panning query, or both. As for the proposed system, regardless of the change in the number of images registered for every point in the database, the error is constantly within 3 meters, and is less than those by the others. It should be noted that the error slightly decreases as the number of registered images increases.

Figure 9 shows the success rate in localization, i.e. the percentage of the estimated location with sufficient number of feature correspondences in the image retrieval process. Even with a sparse database where only 9 images were registered for each data acquisition point, the success rate higher than 90% was achieved, i.e. the system was able to estimate the locations for more than 90% of the query points in the shopping mall. This figure also shows that the success rate increases as the number of registered images increases.

Based on the results shown in Figure 8 and 9, it can be confirmed that for approximately 90% of the area in the shopping mall, the user's location was successfully estimated with the average error of 2.6 meters. The average computation time for estimating the user's location was 0.4 seconds on the server. These results show the advantage of the location correction and the panning query in the proposed system.

IV. Conclusions

This paper presented a novel image based localization system which uses a database with sparsely-captured images and the panning query images. Experimental results show the proposed system succeeded in estimating the user's location with the average error of 2.6 meters in approximately 90% of the area in a real shopping mall. The proposed system is based on image matching. Therefore, different points having similar appearances may cause error in location estimation. One promising solution for this is to combine the system with other technologies, e.g. Wi-Fi based positioning, etc. Also, evaluating the performance in other environments is our future work.

Acknowledgment

Parts of this research were supported by Ministry of Internal Affairs and Communications, Japan.

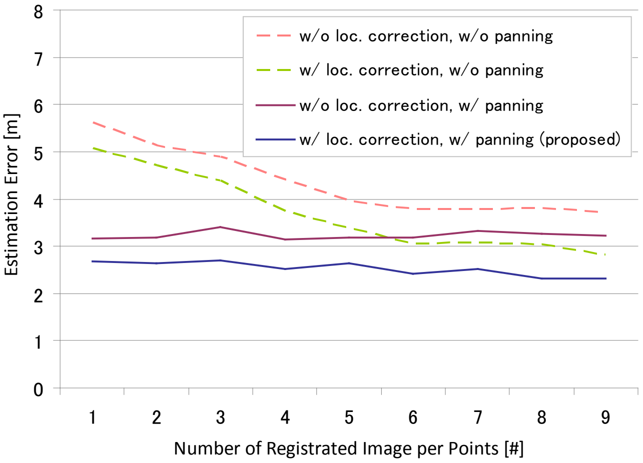


Figure 8. Estimation error of the registration number of images per points

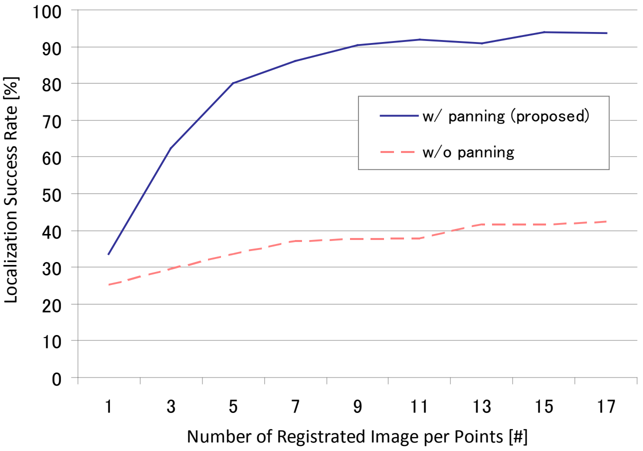


Figure 9. Localization success rates

References

[1] R.Mautz and S.Tilch, "Survey of Optical Indoor Positioning Systems", International Conference on Indoor Positioning and Indoor Navigation (IPIN), 21-23 September 2011.

[2] H. Kang, A. Efros, T. Kanade and M. Hebert, "Image matching in large scale indoor environment", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop on Egocentric Vision, pp.33-40, 2009.

[3] M.Werner, M.Kessel and C.Marouane, "Indoor Positioning Using Smartphone Camera", Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp.1-6. 2011

[4] E.Rosten and R.Drummond, "Machine learning for high-speed corner detection" , Proc. ECCV, Vol.1, pp.430-443, 2006

[5] M.Calonder, V.Lepetit, C.Strecha and P.Fua "BRIEF : Binary Robust Independent Elementary Features", Proc ECCV, pp.778-792, 2010.

[6] M. A. Fischler and R. C. Bolles. "Random Sample Consensus:Paradigm for Model Fitting with Applications to Image analysis and Automated Cartography", Comm. of the ACM, Vol.24, p 381-395, 1981.

[7] http://kashiwa.lalaport.jp