Mobile Augmentation Based on Switching Multiple Tracking Method

Ayaka Miyagi, Daiki Yoshihara, Kei Kusui, Asako Kimura, and Fumihisa Shibata

Graduate School of Information Science and Engineering, Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga, 525-8577, Japan
miyagi@rm.is.ritsumei.ac.jp

Abstract. This paper presents a localization mechanism for mobile augmented reality systems in various places. Recently, variety of image-based tracking methods have been proposed: artificial marker based methods, and natural feature based methods. However, localization done with only one tracking methods is difficult in all situation. Therefore, we propose a system, which enables users to continually track in various situation by dynamically switching the multiple localization methods. Our proposed mechanism consists of clients, a switcher, and servers. The server estimates the camera pose of the client, and the switcher selects the outstanding localization method. Furthermore, we employed real-time mapping to continually estimate the position and orientation even if the camera is apart from the prior knowledge of the environment. After localization, the newly updated mapping result is stored in the server. Thus, we could continually track even if the environment has changed.

Keywords: mixed reality, localization, tracking.

1 Introduction

Recently, Mixed Reality (MR) has become increasingly important. In particular, mobile MR systems that could be used in wide area are attracting attentions. One of the most important issues for mobile MR systems is to estimate the device's position and orientation accurately in real-time. Therefore, a variety of tracking methods have been proposed: artificial marker based methods, natural feature based methods. However, it is next to impossible to use the only single method to cope with all the conceivable environments. To successfully accomplish localization in diverse environments, we propose a mechanism which dynamically switches multiple localization methods depending on the usage environment.

In this paper we describe a localization mechanism that switches multiple tracking methods to track in wide area environment using client-server model. The mobile device communicates with the switcher, which selects a suitable tracking method. The selected method changes dynamically according to the surroundings of the device. Additionally, we have applied real-time mapping to continue each tracking method without any prior knowledge.

R. Shumaker (Eds.): Human-Computer Interaction, Part I, HCII 2014, LNCS 8525, pp. 385–395, 2014.

2 Related Work

The simple way to estimate camera pose is a marker based method that place an artificial marker in the environment. For example, Wagner *et al*. has applied ARToolKit [1] by Kato *et al*. for PDA [2]. This method obtains camera position and orientation by detecting the four corners of the marker. In general marker based method has small computational cost. However, these kinds of methods need to always detect a marker from a captured image. Also, placing these markers in an environment is visually obtrusive because the markers usually have an obvious colors and shapes to improve their detection rate. We have proposed a less noticeable marker based method using a poster [3-4], but could not solve the former problem. On the other hand, hence performance of the mobile device has improved significantly in recent years, a variety of feature based methods have been proposed. The methods that apply planar constraint using features are [5, 6]. This method could achieve a robust localization using a plane, such as a poster. However, compared to the marker based method, it is believed that this method is suited for occlusion, but in general the processing speed is slow.

Meanwhile, Arth *et al.* has proposed a mobile based localization method that construct a 3D map of the feature points [7]. Thereafter, Arth *et al.* has also proposed a localization method using panoramic image [8]. This method could be done in wide field of view in outdoor scene by using a panoramic image generated from the images acquired from mobile device. However, the proposed method presupposes the rotation of the camera, although translation is not mentioned. As method capable of performing a translation are method created by Oe *et al*[9] and Ventura *et al*[10] that build a 3D map using image obtained from omnidirectional camera, and localize using 3D point and key frames. However these feature based methods could not be done in an environment where there are less features detected. Also, it is difficult to localize in environments where there are similar patterns due to the error when matching the feature points.

For other feature based localization method contains PTAM [11, 12] which construct the 3D map and estimate the position and orientation of the camera in parallel threads. This method construct the 3D map in real-time, so it could localize without the need to determine the 3D position of the feature points in advance. However, this method could not measure the absolute coordinate of the camera in the environment. As a result, it could not be applied directly to application such as navigation that uses the information depended on the absolute position. Furthermore, as the 3D map expands, the optimization of the detected feature points would become heavy for the mobile phone, and the accuracy of the position and orientation of the camera may lack due to the error contained in feature points as the feature points increases.

Another method created by Castle *et al*. is PTAMM [13]. This method creates multiple maps simultaneously to widen the area of localization. However, since each map is independent, it could not create MR applications that extend over multiple maps.

These proposed methods limit its place to be used. Therefore, wide range localization could not be done with single method.

3 System Architecture

3.1 Overview

Figure 1 shows the conceptual image of our switching mechanism. The proposed mechanism switches each method to realize both wide and various locations in tracking.

The system is composed of client-server model. The Server estimates the initial position and orientation with high processing loads. Subsequently, to ensure estimation in real-time, the light processing camera tracking is been done on the client side. Additionally, in order to ensure the extensibility and load balance executed in the server for the newly proposed localization, we provide each localization method with each server. By analyzing the calculated result of each server in the switcher, the overall throughput does not depend on the number of localization method employed. To improve the stability of the overall system, we also applied sensor based localization and real-time mapping. For the sensor based method, we used geomagnetic sensor, acceleration sensor, and a GPS built in the mobile device. The real-time mapping could perform the localization to continue even after when each localization method fail in tracking.

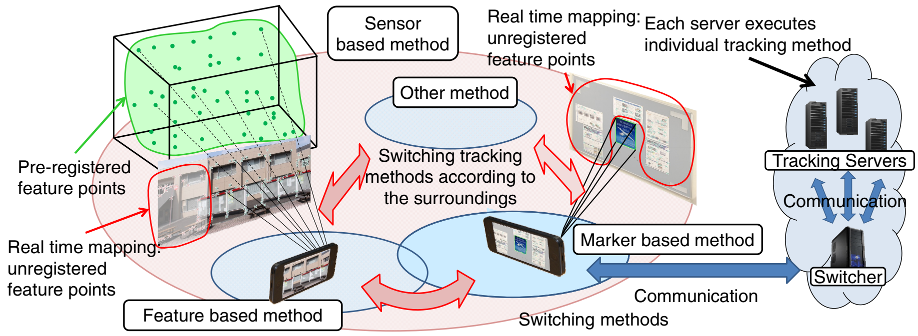


Fig. 1. The concept of our proposed mechanism. The system switches the tracking method according to the surroundings of the device and executes real-time mapping to continue the pose estimation.

3.2 Flow of the System

The proposed method contains three components: client, server, and switcher. The server manages each localization method to estimate the client's position. The switcher selects the superior tracking method to be used by the client. The client achieves the real time localization by camera tracking.

1. Requests to choose a tracking method (Client)
   The client gets a captured image and GPS information, and sends them to the server.
2. Request to estimate position and orientation (Switcher)
   The switcher sends the captured camera image and GPS information to each server.
3. Estimate position and orientation of the camera (Server)

Each server receives the captured camera image and GPS information and estimates the position and orientation.
4. Sends back the position and orientation to the switcher (Server)
   Each server sends back the estimated position and orientation to the switcher.
5. Selects the superior tracking method (Switcher)
   From the estimated position and orientation provided by the each server, the switcher selects the highly accurate tracking method.
6. Notify the tracking method to the client (Switcher)
   The switcher notifies the tracking method and sends the estimated initial position and orientation to the client.
7. Real-time tracking (Client)

From the received information, the client estimates its position and orientation in real-time. In this moment, the client periodically sends a captured camera image, position and orientation of the camera, and GPS information to the server. The system continually repeat (1) thorough (6) and switch into the superior tracking method for client.

4 Evaluation Function for Switching the Localization Method

4.1 Policy of Switching

In order to achieve the localization in both wide and various locations, we proposed a mechanism that selects the most outstanding localization method from several localization methods. In order to achieve this, an evaluation function is necessary. We examined to consider an evaluation function that switches between marker based method and feature based method. The system switches the tracking method when the tracking fails in the current method, and when there is superior accuracy in other tracking method.

We have decided to set the evaluation formula for evaluating the adaptability of the localization method. The localization that has the highest value from the formula will be considered as the worthiest localization method to be used.

4.2 Evaluation Function

To switch into the superior localization method, various elements could be considered such as re-projection error and positional relationship of the marker.

Thus, we have parameterized individual element of each localization method. We have gradually added the parameters and select the highest score. Currently, the localization method is switched using the following formula (1).

$$S_i = M_i + (-E_i) \tag{1}$$

The number of localization method is i, evaluation value for each localization method is S_i. M_i is a parameter for each method. E_i is the average re-projection error. For M_i, it is possible to select the method with priority by setting the values to appropriate natural number.

4.3 Fail-Safe

Evaluation formula described in the previous section selects localization method when estimation of the position and orientation succeeds. When all the initial localization fails in all methods or when the tracking could not achieve stable estimation, we roughly estimate the localization using GPS and direction sensor. In particular, the position of the camera is determined by the value of the GPS. The yaw is determined by magnetic sensor. Also, the estimation of roll and pitch is determined by gyro sensor. However, even when we use the sensor based localization, the switcher continuously receive localization methods from the server. When switcher select a localization other than sensor based method, we process the newly selected localization method to start.

5 Real-Time Mapping

5.1 Overview of Real-Time Mapping

In the following, we describe the real-time mapping process which has been employed in the proposed mechanism.

While it is possible to dynamically switch into localization method according to the location used for tracking, the camera path and the movement will be limited with the conventional method. Therefore, we estimate the 3D position of the feature points from camera image in real-time parallel with the tracking. However, real-time mapping process could not estimate the camera position in world coordinate. For this reason, we have employed real-time mapping process, which we designed to suit each localization method.

5.2 Real-Time Mapping for Marker Based Method

Processing Flow of Real-time Mapping for Marker Based Method. With the localization method with the marker-based method, we estimate the 3D position of the feature point around the marker and continually track, even when the camera is distant from the marker.

We used PTAM proposed by Klein et al. [11] for mapping process. Mapping the feature points of a large amount in high speed, PTAM has achieves stable process by asynchronously updating the environment model in parallel threads. When performing the marker based method, we use both the conventional localization method and PTAM in parallel thread to estimate the 3D position of the feature points.

Since PTAM could not define the coordinate, we have converted the scale from the amount of movement from the camera and with the transformation matrix; we convert 3D positional points that were mapped with PTAM to absolute coordinate.

Initial Map. In the following, we include the process of building the initial map.

1. Select the first frame that succeeded in position and orientation with the marker-based method.
2. Select the second frame that moved parallel to the direction of optical axis.
3. We estimate the 3D position using the correspondence of selected keyframes.

In order to realize the process (2), it is necessary to determine if the localization method has moved parallel other than optical axis. To determine the movement of the mobile phone, we use gyroscope that is built into a device.

Switching Mechanism. The accuracy of camera tracking in PTAM depends on the number of feature points detected from the camera image. Therefore, when the marker is captured within the camera image, the marker is superiorly selected. The tracking is switched into PTAM when the camera is distant from the marker. In addition, since two approaches could not run simultaneously due to the low computational power in mobile device, marker based method and PTAM run in separate thread to reduce the processing speed.

5.3 Real-Time Mapping for Feature Based Method

Flow of Real-Time Mapping for Feature Based Method. In the proposed mechanism, we assume the feature points to be extracted and stored from prior knowledge. We will describe a method for introducing the mapping process. We will describe in detail with Landmark Database (LMDB) as example.

In LMDB method, the feature point is registered in prior. With the 3D-2D matching correspondence, the LMDB are constructed. Therefore, it is difficult to estimate the camera position and orientation when the camera is distant from the constructed area. Unlike the marker based method, the feature points will gradually decrease as the camera moves apart from the constructed environment. Without using the previously stated marker based real-time mapping method, we proposed a method to add and update a new feature points as landmark in real-time. By using the position and orientation estimated by the priory registered landmarks and added landmarks, we realized to continually track even when camera is apart from the constructed LMDB. We also update the database, so the newly added landmarks could be used next time.

Adding the Landmark in Real-Time. To add the landmark in real-time, it is essential to estimate the 3D position of the feature in real-time. In PTAM, they use a triangulation from two frames to estimate the camera position. This method has a simple calculation, so the 3D position of the features could be estimated immediately, but depending on the selected frame, the camera position and orientation would include a major error. PTAM uses large quantities of feature and repeatedly optimize the mapping to decrease this error. However feature based methods like LMDB, would fail in tracking when 3D reconstructed features and new detected features are simultaneously tracked.

To achieve highly accuracy in estimation, we used the flow listed below.

1. Determine whether the 3D position is estimated in current frame
   Without using the 3D position estimated by all the frames, we use frames that include a great quantity of feature with small re-projection error. We also observe certain parallax compared to the previous frame. The tracking thread verifies if the position and orientation estimation is satisfied in each frame.

2. Correspondence using additional keyframe
   We add the keyframe that meets these listed conditions in the mapping thread. At this time, we use Lucas-Kanade [15] to correspond each feature from the previous keyframe.
3. Estimate 3D position of the feature
   When there are parallax compared to the previous frame, and when keyframe are obtained and stored, we estimate the 3D position of the feature using the multiple keyframes. This newly detected landmark will be stored. We also remove the feature that contains major errors. Specifically, we re-project 3D position for all frames, and calculate the average value of the re-projection error for each frame. We delete these feature that has re-projection error in certain threshold.

Update the LMDB Obtained from Real-Time Mapping. Using the real-time mapping, we update the registered information of Landmark in database. With the real-time mapping, we could perform camera tracking in location outside the constructed landmark. However, the initial position and orientation could not be estimated to the place where landmark does not exist. Therefore, we store the information of the feature points which was detected in real-time mapping in the server, and register it in the database as a new landmark to update the database. Due to the changes in environment, we exclude landmarks from database that could not correspond to the feature detected while tracking. With this approach, even when there are changes in the environment, we could easily update the database.

This approach is done with online process and offline process. The online process stores the 3D position of the feature detected in real-time mapping, image information of each keyframes, and position and orientation of the camera. Furthermore, in order to prevent decrease in accuracy and capacity in data-base, we register feature with high utilization rate. We calculate the utilization rate by the following equation (4). U_i is the utilization rate, I_i is the number of times that are considered as inlier when using RANSAC. F_i is the total number of frames that contain the added landmarks. The utilization rate is stored in the server and used in the offline process.

$$U_i = \frac{I_i}{F_i} \tag{4}$$

The offline process builds additional landmarks and eliminates other landmark that has low utilization rates. The landmarks with low utilization rate are deleted and others are registered into the database.

6 Experiment

We have qualitatively evaluated our switching system in both outdoor and indoor environment. For the server we have employed ARToolKit[1], SFINCS-PM[3-4], and LMDB[9]. The equipment used for client, switcher and server is listed in Table 1. The resolution is 640 x 480.

Table 1. Specification of the client, switcher and server

	Client	Switcher & Server
Device	4th iPad	Notebook PC
OS	iOS 6	Windows7 x64
CPU	Apple A6X 1.4GHz	Intel Core i7 2.8GHz
RAM	1GB	8.0GB

6.1 Camera Tracking

In this experiment, we check whether our proposed mechanism selects the superior tracking method by comparing the true value. The true value is measured with robotic arm. Fig.4 shows the environment of the experiments. The coordinate system is set to x-axis to be parallel to the camera path, and y-axis to be opposite direction of gravity, and z-axis in the front direction.

Fig. 2 shows the camera position of each method. From the estimated camera position, the LMDB is switched into ARToolKit from 185 frames; ARToolkit is changed into LMDB from 305 frames. Lastly LMDB is switched into SFINCS-PM method from 441frame. We could see that the estimation in camera position changes according to the motion of the robotic arm. Fig. 3 shows the localization done inside the room.

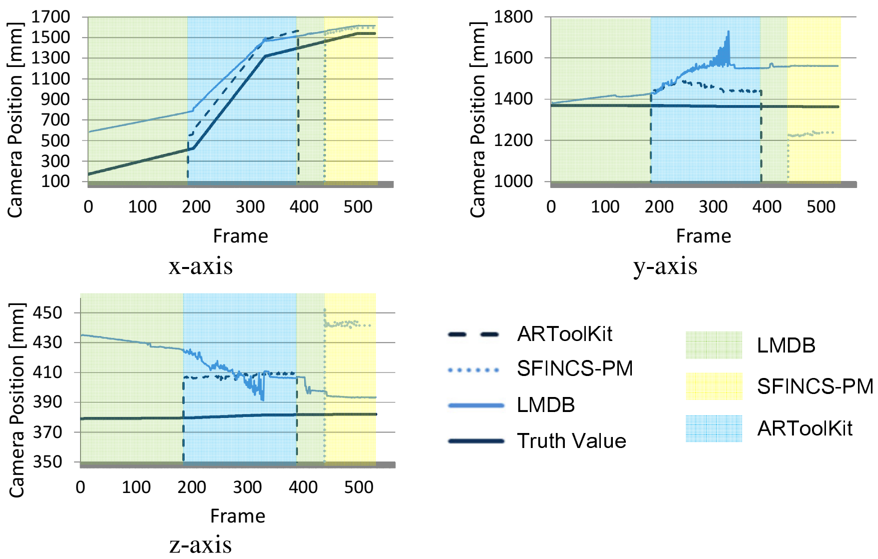


Fig. 2. Comparison of localization result

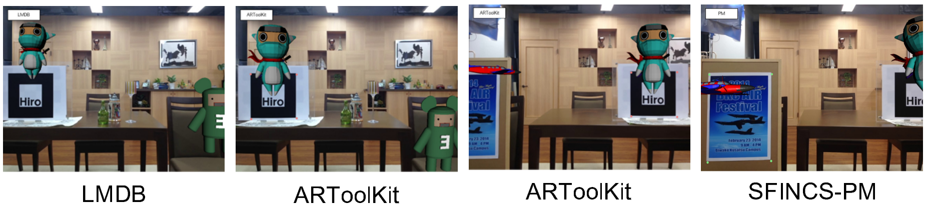


Fig. 3. Result of tracking inside the room

6.2 Tracking Using Real-Time Mapping

We checked the operation of the updated LMDB using real-time mapping described in Ch.5.3.3 with feature based LMDB method. The experiment is done in indoor scene. The LMDB is constructed in scene listed in Left Top of Fig 4. The black and white ARToolKit plus markers are used for setting the world coordinate. We perform the localization using real-time mapping in each database after changing the position of the object after constructing LMDB (Fig 4). After updating the LMDB, we execute the camera tracking using LMDB and compare with the camera position with pre-updated LMDB (Fig. 5). Further, in order to compare the initial camera position, we compare the number of times the initial camera position is estimated (Fig. 6).

With the pre-updated LMDB, the camera tracking failed from 180 frames. However with the updated LMDB, we could continually track even when the object is moved after the constructed LMDB.

Fig. 6 shows the result of estimating the initial camera position started form 10 location. In the pre-updated LMDB, we could only estimate five locations. With the updated LMDB, it has become possible to estimate 9 locations.

Fig. 4. Indoor Scene. Left Top: Constructed LMDB. Left Bottom: Object is moved after constructed. Right: Tracking using updated LMDB.

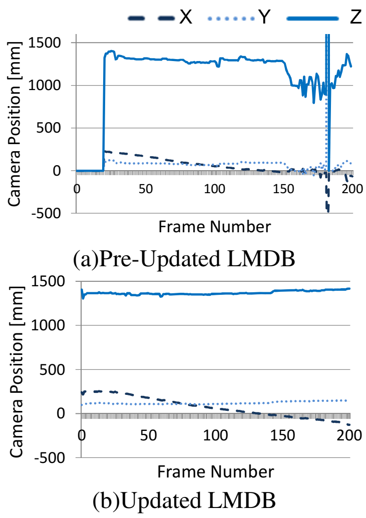
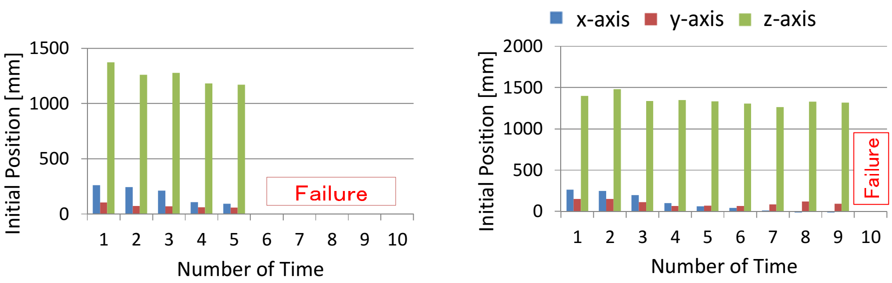


Fig. 5. Tracking result using the conventional database and updated database using real-time mapping



(a)Pre-updated LMDB (b)Updated LMDB

Fig. 6. Initial position

7 Conclusion

In this paper, we have proposed a mechanism to achieve mobile augmentation in various places by dynamically switching the multiple localization techniques. The proposed mechanism is executed on the server. The localization is achieved in various environments by selecting the superior method. Furthermore, when the tracking fails, the tracking is switched to real-time mapping. With the feature based method, the mapping result obtained from real-time mapping is updated in the database. Therefore, we could continually track in environments that have changed.

For future works, our evaluation formula for switching uses the weight defined by user and re-projection error. However, we should also consider the processing time and positional relationship of the camera and the marker. For future works, we will investigate the events that occur while tracking and make an advanced evaluation formula.

For the sensor based method, when the direction of the camera has been greatly changed, the jitter and error increases. To solve this problem, we should consider low-path filter which could reduce the amplitude obtained from the sensor.

References

1. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proc. Int'l Workshop on Augmented Reality (IWAR), pp. 85–94 (1999)
2. Wagner, D., Schmalsting, D.: First steps towards handheld augmented reality. In: Proc. Int'l Conf. Wearable Computers (ISWC), pp. 127–135 (2003)
3. Tenmoku, R., Yoshida, Y., Shibata, F., Kimura, A., Tamura, H.: Visually elegant and robust semi-fiducials for geometric registration in mixed reality. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 261–262 (2007)
4. Kubo, Y., Komurasaki, S., Shibata, F., Kimura, A., Tamura, H.: Trial evaluation of a marker-based geometric registration method using a TrakMark data set. In: Proc. Int'l Workshop on Benchmark Test Scheme for AR/MR Geometric Registration and Tracking Method (TrakMark), pp. 6–9 (2012)

5. Pirchheim, C., Reitmayr, G.: Homography-based planar mapping and tracking for mobile phones. In: Mixed and Augmented Reality (ISMAR), pp. 27–36 (2011)
6. Kurz, D., Olszamowski, T., Benhimane, S.: Representative feature descriptor sets for robust handheld camera localization. In: Proc. Mixed and Augmented Reality (ISMAR), pp. 65–70 (2012)
7. Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D.: Wide area localization on mobile phones. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 73–92 (2009)
8. Arth, C., Klopschitz, M.: Real-time self-localization from panoramic images on mobile devices. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 37–46 (2011)
9. Oe, M., Sato, T., Yokoya, N.: Estimating camera position and posture by using feature landmark database. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 171–181. Springer, Heidelberg (2005)
10. Ventura, J., Höllerer, T.: Wide-area scene mapping for mobile visual tracking. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 3–12 (2012)
11. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 1–10 (2007)
12. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: Proc. Int'l Symp. on Mixed and Augmented Reality (ISMAR), pp. 83–86 (2009)
13. Castle, R., Klein, G., Murray, D.W.: Video-rate localization in multiple maps for wearable augmented reality. In: Proc. Int'l Symp. on Wearable Computers (ISWC), pp. 15–22 (2008)
14. Chum, O., Matas, J.: Matching with PROSAC - progressive sample consensus. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 220–226 (2005)
15. Lucas, B.D., Kanade, T.: An interactive image registration technique with an application to stereo vision. In: Proc. Image Understanding Workshop, pp. 121–130 (1981)