Construction of Feature Landmark Database Using Omnidirectional Videos and GPS Positions

Sei IKEDA[1], Tomokazu SATO[1], Koichiro YAMAGUCHI[1,2] and Naokazu YOKOYA[1]
[1]Nara Institute of Science and Technology
8916–5 Takayama-cho, Ikoma-shi, Nara-ken, 630-0101 Japan
{sei-i,tomoka-s,yokoya}@is.naist.jp
[2]Toyota Central R&D Labs., Inc.
41-1 Aza Yokomichi, Oaza Nagakute, Nagakute-cho, Aichi-gun, Aichi-ken, 480-1192, Japan
yamaguchi@mosk.tytlabs.co.jp

Abstract

This paper describes a method for constructing feature landmark database using omnidirectional videos and GPS positions acquired in outdoor environments. The feature landmark database is used to estimate camera positions and postures for various applications such as augmented reality systems and self-localization of robots and automobiles. We have already proposed a camera position and posture estimation method using landmark database that stores 3D positions of sparse feature points with their view-dependent image templates. For large environments, the cost for construction of landmark database is high because conventional 3-D reconstruction methods requires measuring some absolute positions of feature points manually to suppress accumulative estimation errors in structure-from-motion process. To achieve automatic construction of landmark database for large outdoor enviroments, we newly propose a method that constructs database without manual specification of features using omnidirectional videos and GPS positions.

1. Introduction

Estimation of camera position and posture is widely applicable to augmented reality systems equipped with wearable computers [1, 2] and self-localization of robots and automobiles. In these applications, there are many situations where any infrastructures such as markers and beacons cannot be newly installed. For this problem, a number of online camera position and posture estimation methods using artificial 3-D models [3–6] were proposed. Although these methods can estimate camera poses without markers and beacons, it is difficult to model complex scenes by hand.

To avoid such a problem, absolute camera position and posture estimation methods that do not require any markers and artificial 3-D models have already been developed [7–10]. These methods use a landmark database that stores 3-D positions of sparse feature points with their view-dependent visual features. Construction of the database is done semi-automatically by using a structure-from-motion technique which requires some absolute 3-D positions of feature points and their correspondences between the world and image coordinate systems. However, for large environments, much human cost is needed to construct the landmark database because conventional 3-D reconstruction methods requires measuring some absolute positions of feature points manually to suppress accumulative estimation errors in structure-from-motion process.

In this paper, we propose a method for constructing a landmark database without manual measurement of environments using omnidirectional videos and GPS positions. The structure-from-motion algorithm using GPS positions [11, 12] is extended to acqire feature landmarks using an omnidirectional multi-camera system (OMS), which consists of multiple cameras arrange radially. The proposed method enables us to obtain absolute positions of feature points and absolute camera poses in the geodesic coordinate system automatically. In the remainder of this paper, the camera position and posture estimation method using landmark database is first briefly described in Section 2. The generation of landmark database using an omnidirectional video and GPS positions is then described in Section 3. In Section 4, the validity of the method is demonstrated by experiments with a real outdoor scene. Finally, we give conclusion and future work in Section 5.

2. Camera Position and Posture Estimation Using Landmark Database

This section describes the camera position and posture estimation method using the feature landmark database. First, the elements of the feature landmark database are de-

fined. How to use that information to estimate poses of a camera is then described.

2.1 Elements of Feature Landmark Database

Feature landmark database consists of a number of landmarks as shown in Figure 1. Each landmark retains the 3-D position of itself, and multiple view-dependent image templates and their geometric information. The former is used with 2-D position of a feature point detected in an input image in order to estimate position and posture of the camera. The latter is used for a robust matching between the landmark image template and input image acquired from various positions. These database elements are generated from omnidirectional videos by using a 3-D reconstruction method described in the next section.

(1) 3-D position of landmark
3-D coordinate of each landmark is estimated by 3-D reconstruction of the environment and is represented in the world coordinate system.

(2) Information for view-dependent image template
This information is used to find correspondences between feature points in an input image and the landmarks.

(A) Position and posture of omnidirectional camera (OMS)
Position and posture of OMS are retained in the world coordinate system. They are used to select landmarks from the database to match the input image.

(B) Multi-scale image template of landmark
Image template is created by rectifying the omnidirectional image so as to be perpendicular to the line passing through 3-D position of the landmarkand and the position of the OMS, as shown in Figure 2.

(C) Normal vector of image template
As shown in Figure 2, the normal vector of image template is the normal vector of the plane which is perpendicular to the line passing through 3-D position of the landmark and position of the OMS. This is used to select an image template for matching from multi-directional image templates taken by different camera positions.

(D) Base scale of image template
As shown in Figure 2, the scale of image template is the size of the plane used to create the image template. The scale size is retained in the world coordinate system, and the base scale is determined so that the resolution of the omnidirectional image and the image template becomes nearly equal.

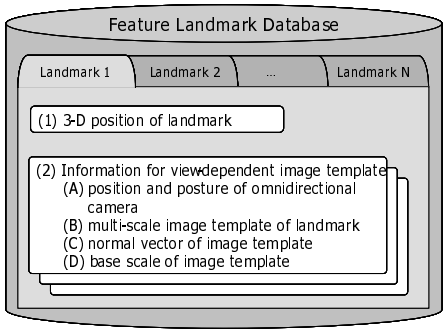


Figure 1. Elements of feature landmark database.

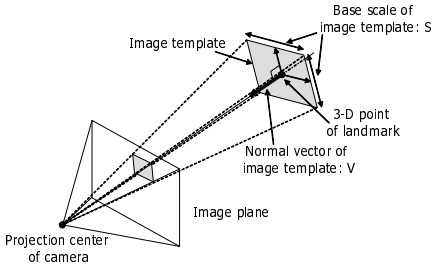


Figure 2. Landmark and its image template.

2.2 Algorithm of Camera Position and Posture Estimation

This section describes a camera position and posture estimation algorithm using feature landmarks described above. This algorithm assumes that the initial camera position and posture are estimated by some other methods. In the subsequent frames, first, landmarks are selected from the landmark database by using the previous camera position and posture. Detecting features from the input image and matching them with the landmark image templates, the correspondence between landmark and input image is then established. Lastly, camera position and posture are estimated from the correspondences between landmarks and input image. The following sections describe these steps.

2.2.1 Selecting Landmark from Landmark Database

To find a correspondence with a feature point in the input image, several landmarks are selected from numerous landmarks in the landmark database. Furthermore, to handle

partial occlusions and aspect changes, an image template with the nearest appearance to the input image should be chosen from a number of view-dependent image templates. Considering the appearance, it is ideal if the image template and input image are taken in the same position. However, the camera position and posture of the current frame image are not estimated yet. We use the camera position and posture of the previous frame as a replacement. Landmarks satisfying the following requirements [10] are selected to make a correspondence with the input image.

(requirement 1) Landmark has to be contained in the image plane when projecting its 3-D coordinate using the previous camera position and posture.

(requirement 2) Distance between the OMS position where the landmark was taken and the camera position where the input image was taken should be small.

(requirement 3) Angle between the normal vector of the image template and the vector from landmark to camera position when the input image was taken should be smallest of all the image templates of the landmark.

(requirement 4) Landmark must not be adjacent to already selected landmarks.

Landmarks satisfying the requirement 1 are selected first. Then, the selected landmarks are narrowed down to a fixed number of landmarks by the ascending order of the distance mentioned in the requirement 2. From the list of landmarks, landmarks with smaller angles in the requirement 3 are picked up one by one, and are repeated until a fixed number of landmarks that satisfy the requirement 4 are chosen.

2.2.2 Determining Correspondence between Landmark and Input Image Feature

The next step is to find the correspondences between selected landmarks and features in an input image. Features are detected from the input image, and are corresponded with the selected landmarks.

Detecting Features from Input Image To find the correspondence between landmarks and input image, feature points are detected from the input image by using Harris operator [13]. We adopt the operator which can be computed faster than more distinctive operators such as SIFT [14]. In this step, a landmark is projected to the input image, using the previous camera position and posture. On the assumption that the corresponding point for the landmark exists near the projected point, feature points are detected within a fixed window surrounding the projected point. The detected feature points are listed as correspondence candidates of the landmark.

Matching Between Landmark Image Template and Input Image In this step, each landmark is compared with its correspondence candidates. First, an image pattern is created for each feature point listed as a correspondence candidate. Next, the landmark image template is compared with each image pattern by normalized cross correlation. Then, the feature point with the most correlative image pattern is selected, and its neighboring pixels are also compared with the landmark as correspondence candidates. Lastly, the most correlative feature point is corresponded with the landmark.

2.2.3 Camera Position and Posture Estimation Based on Established Correspondences

Camera position and posture are estimated from the list of 2-D and 3-D correspondences acquired from the matching between landmarks and input image. First of all, outliers are eliminated by RANSAC [15]. Then, camera position and posture are determined, using only the correspondences that are supposed to be correct. As a result, camera position and posture with the minimum re-projection error becomes the answer. Here it should be noted that more than five correspondences are required in order to determine camera position and posture uniquely.

3. Construction of Landmark Database from Omnidirectional Video and GPS Positions

To obtain the elements of the landmark database described in Section 2.1, poses of OMS and 3-D positions of feature points are required. This section describes a 3-D reconstruction method which enables us to estimate these parameters. In the proposed method, the general structure-from-motion algorithm is enhanced to treat multiple videos acquired with OMS and GPS position information. In the general structure-from-motion algorithm, re-projection error is minimized to obtain camera parameters and 3-D positions of feature points. In the proposed method, a new error function combining the re-projection error and the error concerning GPS is minimized. First, in this section, we introduce an omnidirectional multi-camera system. The new error function combining the re-projection error and the error function concerning GPS is then described. Finally, the algorithm to minimize the function is described.

Note that the following conditions are assumed in our method: (i) OMS and GPS are correctly synchronized; (ii) the geometrical relation among all the cameras and the GPS receiver is always fixed and known. In this paper, it is also assumed that OMS has been calibrated in advance [16] and the intrinsic camera parameters (including lens distortion, focal length and aspect ratio) of each element camera of OMS are known.

Figure 3. A sampled frame of an acquired omnidirectional video. Right bottom is an image of vertical element camera. Others are horizontal ones.

3.1 Omnidirectional Multi-camera System

Omnidirectional multi-camera system is constructed of a set of element cameras which can obtain omnidirectional videos as shown in Figure 3. As mentioned above, we assume that position and posture relations among element cameras are known and fixed in this paper. The poses of all the cameras can be relatively expressed as a pair of position and posture of a representative camera. In the i-th frame, the transformation from the world coordinate system to the camera coordinate system of each element camera c can be expressed by the following matrix N_{ic} by using the transformation M_c from the world coordinate system of a calibration process to the camera coordinate system of the camera $c\ (= 0, 1, 2, 3...n)$.

$$\mathsf{N}_{ic} = \mathsf{M}_c(\mathsf{M}_0)^{-1}\mathsf{N}_{i0} = \begin{bmatrix} \mathsf{R}_{ic} & \mathbf{t}_{ic} \\ 0 & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{t}_{ic}$ and R_{ic} represent the translation and the rotation from the world coordinate system to the camera coordinate system of the camera c for the i-th frame. This problem is treated as estimation of position ($\mathbf{t}_i = \mathbf{t}_{i0}$) and posture ($\mathsf{R}_i = \mathsf{R}_{i0}$) of the representative camera (c=0).

3.2 Error Function for Optimization Process

Re-projection Error Re-projection error is generally used for extrinsic camera parameter recovery based on feature tracking. The method for minimizing the sum of squared re-projection errors is usually referred to as bundle adjustment. The re-projection error Φ_{ijc} for the feature j in the i-th frame of the camera c is defined as follow.

$$\Phi_{ijc} = |\mathbf{q}_{ijc} - \hat{\mathbf{q}}_{ijc}|, \tag{2}$$

where $\hat{\mathbf{q}}$ represents the 2D projected position of the feature's 3D position and $\mathbf{q}$ represents the detected position of the feature in the image. The 2D projected position $\hat{\mathbf{q}}$ of the 3-D position $\mathbf{p}_j$ of the feature j whose depth is z is calculated by the following equation.

$$\begin{bmatrix} z\hat{\mathbf{q}}_{ijc} \\ z \\ 1 \end{bmatrix} = \mathsf{N}_{ic}\mathbf{p}_j, \tag{3}$$

Error of GPS positions Generally, if GPS positions and estimated camera parameters do not contain any errors, the following equation is satisfied in the i-th frame among the camera parameters (position $\mathbf{t}_i$, posture R_i), GPS position $\mathbf{g}_i$ and the position of GPS receiver $\mathbf{d}$ in the camera coordinate system.

$$\mathsf{R}_i\mathbf{g}_i + \mathbf{t}_i = \mathbf{d} \quad (i \in \mathcal{F}), \tag{4}$$

where $\mathcal{F}$ denotes a set of frames in the frames where GPS positions are obtained. However, unfortunately GPS position $\mathbf{g}_i$ and the parameters $\mathbf{t}_i$ and R_i usually contain some errors. We introduce the following error function Ψ_i as an error of measured GPS position, which means the distance between the measured position of the GPS receiver and the predicted one.

$$\Psi_i = |\mathsf{R}_i\mathbf{g}_i + \mathbf{t}_i - \mathbf{d}|. \tag{5}$$

Error Function Concerning Feature and GPS The new error function E is defined as follows:

$$E = \frac{\omega}{|\mathcal{F}|}\sum_{i\in\mathcal{F}}\Psi_i^2 + \frac{1}{\sum_{i,c}|\mathcal{S}_{ic}|}\sum_{i,c}\mu_i\sum_{j\in\mathcal{S}_{ic}}w_j\Phi_{ijc}^2, \tag{6}$$

where ω means a weight for Ψ_i, and $\mathcal{S}_{ic}$ denotes a set of feature points detected in the i-th frame of the camera c. The coefficients μ_i and w_j mean the confidences for frame and feature, respectively. The coefficient w_j is computed as an inverse variance of re-projection error Φ_{ij}. The coefficient μ_i is a ratio of the frame rate for the rate of GPS. Two terms in the right-hand side in Eq. (6) is normalized by $|\mathcal{F}|$ and $\sum_i\sum_c|\mathcal{S}_{ic}|$, respectively, so as to set ω as a constant value independent of the number of features and GPS positioning points.

3.3 Algorithm of 3-D Reconstruction

The proposed method basically consists of feature tracking and optimization of camera parameters as shown in Figure 4. First, two processes of (A) feature tracking and (B)

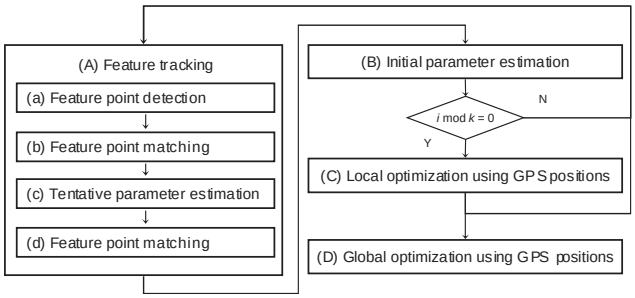


Figure 4. Overview of the proposed algorithm.

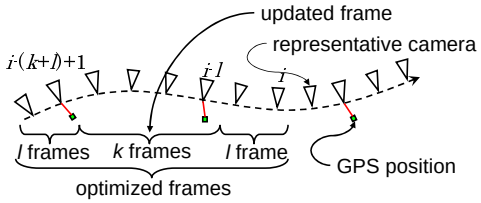


Figure 5. Optimized frames in the process (C).

initial parameter estimation are performed in order. At constant frame intervals, the local optimization process (C) is then carried out to reduce accumulative errors. Finally, estimated parameters are refined using many tracked feature points in the global optimization process (D). In the processes (C) and (D), a common optimization is performed. The difference in both processes is the range of optimized frames. In the process (C), the range of optimization is a small part of the input frames because future data cannot be treated in sequential process. On the other hand, in the process (D), all the frames are optimized and updated.

(A) Feature tracking : The purpose of this step is to determine corresponding points between the current frame i and the previous frame $(i-1)$. The main strategy to avoid mismatching in this process is that feature points are detected at corners of edges by Harris operator [13] and detected feature points are tracked robustly with a RANSAC [15] approach. Note that feature point tracking is carried out in intra- and inter-camera images of OMS.

In the first step (a) in Figure 4, feature points are automatically detected by using the Harris operator for limiting feature position candidates in the images. In the next step (b), every feature in the $(i-1)$-th frame is tentatively matched with a candidate feature point in the i-th frame by using a standard template matching. In the third step (c), tentative parameters are then estimated by selecting correct matches using a RANSAC approach [15]. In the final step (d), every feature is re-tracked within a limited searching area in image frames of all the element cameras, which can be computed by the tentative parameters and 3D positions of the features.

(B) Initial parameter estimation : This process computes 3D positions of feature points and position and posture parameters of cameras which minimize the sum of squared re-projection errors. In this process, the parameters of the current frame i are computed by using the tracked feature points. The error function E_{init} defined by Eq. (7) is minimized to optimize both the parameters $\mathbf{t}_i$ and R_i of all the frames and 3D positions of all the feature points.

$$E_{init} = \sum_{j \in \mathcal{S}_{ic}} w_j \Phi_{ijc}^2. \tag{7}$$

(C) Local optimization : In this process, the frames from the $(i-(k+l)+1)$-th to the current frame are used to refine the camera parameters from the $(i-(k+l)+1)$ to the $(i-l)$-th frames, as illustrated in Figure 5. This process is designed to use feature points and GPS positions obtained in the frames around the updated frames. The camera parameters of $(k+2l)$ frames are optimized by minimizing the new error function given in Eq. (6) by Levenberg-Marquardt method. To reduce computational cost, this process is performed every k frames. Note that the estimation result is insensitive to the value of l if it is large enough. The constant l is set as tens of frames to use a sufficient number of feature points reconstructed in the process (B). The constant k is set as several frames, which is empirically given so as not to accumulate errors in the initial parameters estimated in the process (B). The weight μ_i $(\in \mathcal{F})$ in which GPS positions are obtained is set as larger number than other frames.

(D) Global optimization : The optimization in the process (C) dose not provide sufficient accuracy for a final output because it is performed for a part of frames and GPS positions. The purpose of this process is to refine parameters using tracked features and GPS positions in all the frames. The algorithm of this process is the same as the narrow optimization process (C) when l and k are set as several hundred frames except that divided ranges are independent of each other.

4. Experiments

To construct the landmark database, we used Ladybug (resolution of element camera 768x1024, 15fps) and a GPS

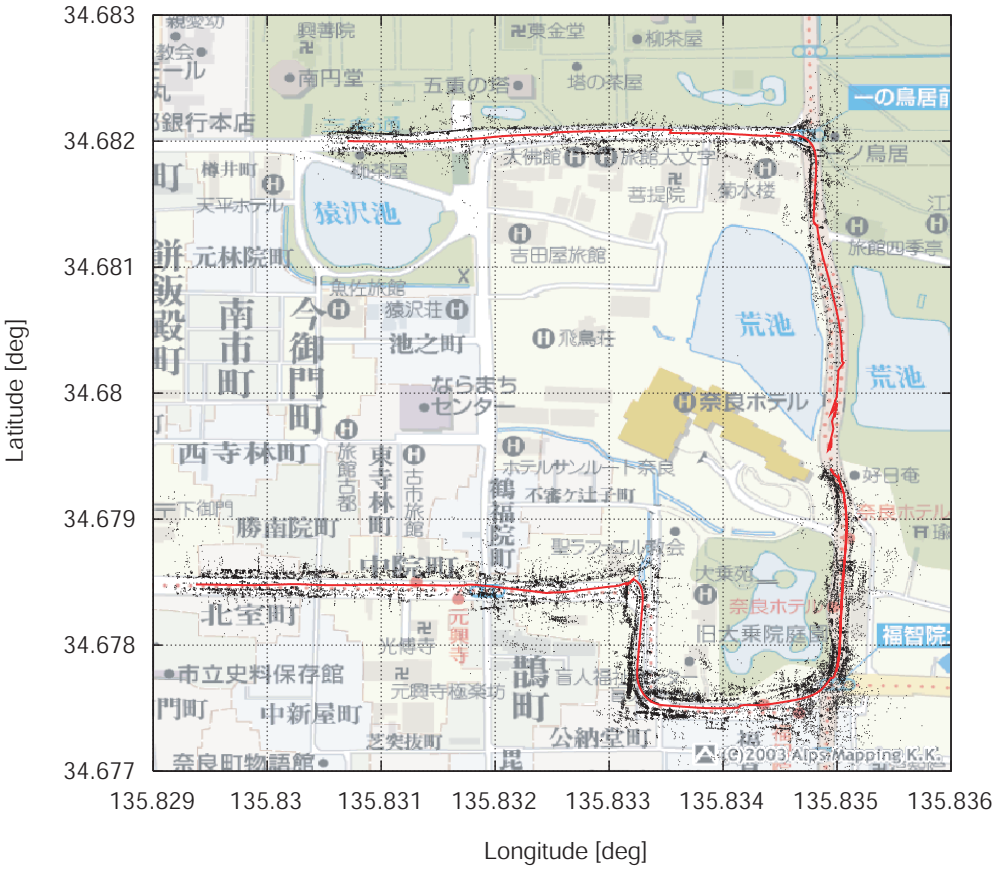


Figure 7. Estimated camera path of the OMS and 3-D positions of feature landmarks.

(a) Data acquisition vehicle.

(b) GPS receiver (left) and OMS (right).

Figure 6. Equipments for acquisition of images and GPS positions.

receiver (Nikon LogPakII, horizontal accuracy ± 3.0 cm, vertical accuracy ± 4.0 cm) mounted on a car as shown in Figure 6. Captured image sequence consists of 100 frames long with 6 images per each frame (totally 600 images). The distance between the first frame and the last frame is about 1.5km. Then, the landmark database is created by estimating camera path and 3-D coordinates of features. The weight coefficient ω is set as 10^{-9}, the local optimization range parameters (l, k) are (22, 30), the global ones are (22, 100). For every landmark, multi-scale image template with three different scales of 15×15 pixels each is created per each camera position. The number of landmarks created in this experiment is about 20,000, and the number of image templates created per each landmark is 8 on average. In Figure 7, black dots show the estimated 3-D positions of landmarks projected on a map. This figure indicates that the estimated 3-D positions of landmarks dose not include large errors. However, the places where reconstructed landmarks do not exist are found as shown in Figure 8. The polygonal lines show estimated path of the OMS and large dots indicate acquired GPS positions. In these places, GPS positioning errors more than 10m are observed. To solve this problem, we must investigate landmark database construction methods using confidence coefficients from GPS.

Next, we performed match move of virtual CG objects to confirmed that the constructed landmark database is applicable to estimation of positions and posuture of a normal

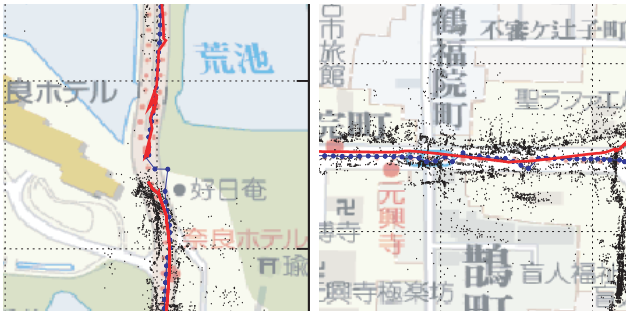
camera. The input video was captured with a handy camera while walking near the place A shown in Figure 7. We have obtained a 300-frame-long monocular video image sequence (720 $\times$ 480 pixels, progressive scan, 15fps) with a video camera (SONY DSR-PD-150) and camera position and posture are sequentially estimated using the landmark constructed earlier. To give the initial position and posture of the camera, image coordinates of six landmarks are manually specified in the first frame of the input sequence. The maximum number of landmarks selected from the database to correspond with input image is 100 per frame, the window size for detecting features from input image is 120 $\times$ 60 pixels, and the number of RANSAC iterations is 500. As a result, computation time per frame was about 1.4 seconds with a PC (Intel Pentium4 3GHz CPU $\times$2, 1.5GB RAM). Figure 9 shows the result of match move; matching virtual 3-D objects to the camera movements using the estimated camera position and posture as shown in Figure 10. It can be observed that the CG person and tank are drawn in geometrically correct positions throughout the sequence. These results indicate that the landmark database constructed by the proposed method is applicable to augmented reality system except for computation time problem.

5. Conclusion

This paper has described a landmark database construction method using omnidirectional videos and GPS positions. Using the landmark database enables us to obtain absolute poses of a normal camera. In the construction of landmark database, any manual measurement processes are not required. Experiments indicate that the landmark database constructed by the proposed method is applicable to augmented reality system except for calculation time problem. As our future works, we must investigate landmark database construction methods using confidence coefficients from GPS.

Acknowledgement

This research is partially supported by CoreResearch for Evolutional Science and Technology (CREST) Program "Foundation of Technology Supporting the Creation of Digital Media Contents" of Japan Science and Technology Agency (JST).



(a) Place A. (b) Place B.

Figure 8. The places where positions of landmarks are not calculated correctly.

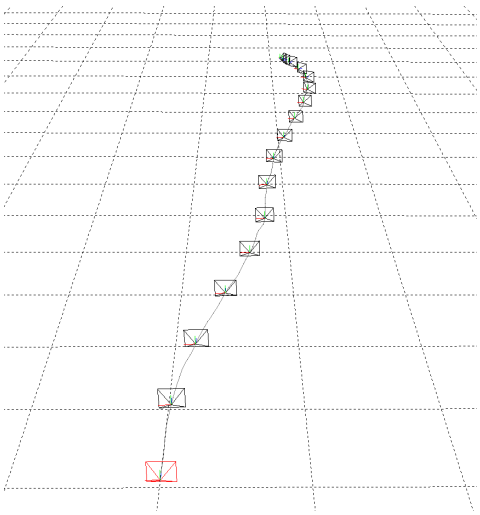


Figure 10. Estimated poses of a handy camera.

References

[1] S. Feiner, B. MacIntyre, T. Höller, and A.Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *Proc. 1st Int. Symp. on Wearable Computers*, pages 74–81, 1997.

[2] P. Daehne and J. Karigiannis. ARCHEOGUIDE: System architecture of a mobile outdoor augmented reality system. In *Proc. 1st Int. Symp. on Mixed and Augmented Reality*, pages 263–264, 2002.

[3] A. I. Comport, É. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *Proc. 2nd IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pages 36–45, 2003.

[4] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Proc. 10th IEEE Int. Conf. on Computer Vision*, volume 2, pages 1508–1515, 2005.

[5] H. Wuest, F. Vial, and D. Stricker. Adaptive line tracking with multiple hypotheses for augmented reality. In *Proc. 4th IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pages 62–69, 2005.

[6] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera track-

Figure 9. CG person and object superimposed onto captured frames.

ing. In *Proc. 3rd IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pages 48–57, 2004.

[7] R. Sim and G. Dudek. Learning and evaluating visual feautres for pose estimation. In *Proc. 1999 IEEE Int. Conf. on Computer Vision*, volume 2, pages 1217–1222, 1999.

[8] D. Burschka and G. D. Hager. V-GPS (SLAM): Vision-based inertial system for mobile robots. In *Proc. 2004 IEEE Int. Conf. on Robotics and Automation*, pages 409–415, 2004.

[9] I. Gordon and D. G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *Proc. 3rd IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pages 110–119, 2004.

[10] M. Oe, T. Sato, and N. Yokoya. Estimating camera position and posture by using feature landmark database. In *Proc. 14th Scandinavian Conf. on Image Analysis*, pages 171–181, 2005.

[11] Y. Yokochi, S. Ikeda, T. Sato, and N. Yokoya. Extrinsic camera parameter estimation based on feature tracking and GPS data. In *Proc. 7th Asian Conf. on Computer Vision*, volume 1, pages 369–378, 2006.

[12] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 964–971, 2004.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, pages 147–151, 1988.

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

[15] M. A. Fischler and R. C. Bolles. Random sample consensus: A pradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[16] S. Ikeda, T. Sato, and N. Yokoya. High-resolution panoramic movie generation from video streams acquired by an omnidirectional multi-camera system. In *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent System*, pages 155–160, 2003.