Fast and Accurate Camera Parameter Estimation Based on Feature Landmark Database for Augmented Reality

Takafumi Taketomi,[†1] Tomokazu Sato[†1]
and Naokazu Yokoya[†1]

In the field of augmented reality (AR), a number of vision-based extrinsic camera parameter estimation methods have been proposed to achieve geometric registration between real and virtual worlds. Previously, we proposed a feature landmark-based camera parameter estimation method for outdoor AR applications. Advantages of this method were that a feature landmark database can be automatically constructed by the structure-from-motion (SfM) and there is no necessity to arrange artificial markers in a target place. However, the previous method could not work in real-time because this method involves high computational matching cost between landmarks in a database and image features in an input image. Additionally, the accuracy of estimated camera parameters was insufficient for specific applications which need to overlay CG objects at the position close to the user's viewpoint. This is due to the difficulty in compensation of visual pattern change of close landmarks only from sparse 3-D information obtained by the SfM. In this report, we achieve fast and accurate feature landmark-based camera parameter estimation by employing the following approaches. (1) The number of matching candidates is reduced to achieve fast camera parameter estimation using tentative camera parameter and priority of landmark. (2) Image templates of landmarks are adequately compensated for by considering local 3-D structure of landmark using dense depth information obtained by a laser range sensor. To demonstrate the effectiveness of the proposed method, we have developed several AR applications.

1. Introduction

The technique of overlaying virtual worlds onto the real world is called augmented reality (AR). AR enables us to obtain location-based information intuitively. AR technologies are applicable to many fields such as human navigation[1),2)], assistance in education[3)], and landscape simulation[4)]. To realize these applications, the real and virtual world coordinate systems should be aligned to overlay virtual objects. Recently, video see-through AR is extensively investigated because it can achieve highly accurate geometric registration by vision-based camera parameter estimation.

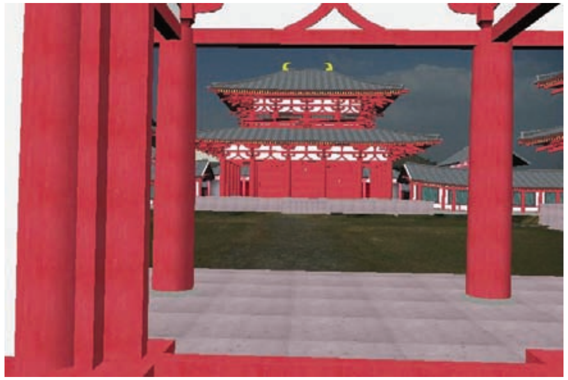


Fig. 1 Example of overlaid virtual objects close to the user's viewpoint.

Previously, Oe *et al.* proposed a feature landmark-based camera parameter estimation method[5)]. This method uses a feature landmark database that is automatically constructed by the structure-from-motion (SfM) to estimate extrinsic camera parameters. Although this method can easily be applied to large-scale environments, its computational cost in the matching process is expensive and it cannot work in real time. In addition, for applications that need to overlay virtual objects near the use's viewpoint as shown in Figure 1, the accuracy of estimated camera parameters is not sufficient. This is due to the difficulty of compensation for visual aspect change of landmarks only from sparse depth information obtained by SfM. In this study, we propose a method to achieve fast and accurate feature landmark-based camera parameter estimation by solving above problems by following ideas.

- In order to reduce the computational cost for camera parameter estimation, a small number of confident matching candidates is selected based on tentative camera parameters and priorities of landmarks.
- In order to improve the accuracy of camera parameters at the spot where

[†1] Nara Institute of Science and Technology

virtual objects must be aligned at the position close to the user, image templates of landmarks are compensated for by using dense depth information obtained by the omnidirectional laser range sensor.

Figure 2 shows a flow diagram of the proposed method. In the offline stage (A), the proposed method uses both SfM and the laser range sensor for specific site where virtual objects must be aligned close to the user. SfM is used to efficiently collect landmark information in a wide area ((A-1.1), (A-2.1)). On the other hand, the laser range sensor is used to collect landmark information at the spot where virtual objects are placed close to the user ((A-1.2), (A-2.2)). In the online stage (B), in order to reduce the matching cost, tentative camera parameters are estimated to limit matching candidates of natural features in an input image by landmark tracking in successive frames (B-2). In addition, priorities are associated with landmarks by using training videos taken in the target environment and these priorities are then used to preferentially select matching candidates of landmarks whose matching confidences are high (B-3).

The remainder of this report is organized as follows. Section 2 discusses related works. Section 3 reviews the previous feature landmark-based camera parameter estimation method[5)]. Then, the reduction in computational cost and the improvement in accuracy are described in Sections 4 and 5, respectively. To show the effectiveness of the proposed method, our method is applied to some AR applications in Section 6. Finally, Section 7 presents the conclusion and outlines the the future work.

2. Previous Work

In the research field of AR, vision-based camera parameter estimation methods are widely employed because they can achieve pixel-level alignment. Most of vision-based methods focus on estimating extrinsic camera parameters by assuming that the intrinsic camera parameters are known. These methods can be classified into two groups. One is a visual-SLAM based approach[6)–10)] that estimates camera parameters without preknowledge of target environments. The other is preknowledge-based approach[5),11)–16)] that uses 3-D information of target environments.

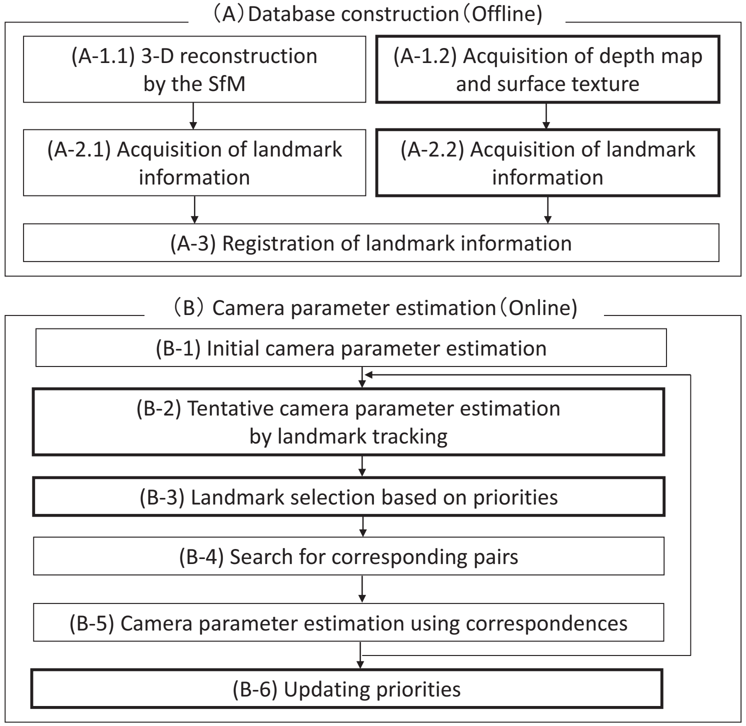

（A）Database construction（Offline）
(A-1.1) 3-D reconstruction by the SfM
(A-1.2) Acquisition of depth map and surface texture
(A-2.1) Acquisition of landmark information
(A-2.2) Acquisition of landmark information
(A-3) Registration of landmark information
（B）Camera parameter estimation（Online）
(B-1) Initial camera parameter estimation
(B-2) Tentative camera parameter estimation by landmark tracking
(B-3) Landmark selection based on priorities
(B-4) Search for corresponding pairs
(B-5) Camera parameter estimation using correspondences
(B-6) Updating priorities

Fig. 2 Flow of the proposed method. Thick squares indicate the new or modified processes.

The visual-SLAM based method simultaneously estimates relative camera motion and the 3-D structure of the target environment by tracking natural features in input images[6)–10)]. This method can easily construct an AR environment without premeasurement of the target environment. The disadvantage of visual-SLAM methods is that they cannot predetermine the coordinate system for arrangement of CG objects. This means the approach cannot be directly used for position-dependent AR applications such as navigation and landscape simulation by itself. To determine the coordinate system, Bleser *et al.* use a partially known 3-D model[10)] and Klein *et al.* employ interactive initialization[9)]. However, these approaches are impractical for a large-scale outdoor environment due to human effort of users. In addition, accumulative estimation error is introduced when camera parameters are estimated by visual-SLAM.

The other uses some kinds of preknowledge of target environments to estimate camera parameters in the global coordinate system[5),11)–16)]. In this approach, 3-D models[11)–14)] and feature landmarks[5),15),16)] are used as preknowledge of target environments. Three-dimensional model based methods are used only for small workspaces because they require large human costs to construct 3-D models for large-scale environments. To reduce the construction cost of 3-D models, Neubert *et al.* proposed a semiautomatic model construction method[17)] that involves the detection of planar regions in a video sequence. However, it is still difficult to construct 3-D models for complex environments using this method.

On the other hand, feature landmark based methods estimate extrinsic camera parameters from correspondences of landmarks in a database and image features in an input image. Skrypnyk *et al.*[15)] and Arth *et al.*[16)] use SfM to construct the feature landmark database. The feature landmark database can be automatically constructed using SfM only from image sequences captured in the target environment. However, SfM only from image sequences results in accumulative estimation error, which is the same drawback as that of the visual-SLAM approach. To avoid this problem, in our previous method[5)], the feature landmark database is efficiently constructed even in a large-scale and complex environment by using the accumulative error free SfM for an omnidirectional camera[18),19)]. However, this method has several problems. It cannot achieve real-time processing, which is necessary for AR, because of the computational cost of matching landmarks to image features. For the computational cost reduction, Skrypnyk *et al.* employ approximate nearest neighbor search in the matching process[15)]. In order to achieve fast retrieval of matching candidates of landmarks from the database, Arth *et al.*[16)] limit the number of visible landmarks using a potential visible set, and Irschara *et al.*[20)] employ the vocabulary tree[21)] to retrieve landmarks. Unlike other natural feature based camera parameter estimation methods, in our previous method, most of the computational time is spent on pattern compensation to handle the difference between the camera properties the omnidirectional camera used in the database construction process and that of the monocular camera used in the online camera parameter estimation process. In general, the camera used to acquire the prior knowledge and the user's camera are different. This pattern compensation cost could not be reduced by using the previously proposed computational cost reduction approaches[15),16),20)]. In addition, the accuracy of the estimated camera parameters is insufficient for AR applications that involve the placing of a virtual object near the user's viewpoint. This is due to the difficulty of matching landmarks that exist close to the user. Visual patterns of close landmarks easily change with viewpoint change. The sparse 3-D information obtained by the SfM process is insufficient for the successful compensation for the pattern change caused by the viewpoint change for close landmarks. To achieve matching that is robust to the viewpoint change, Wu *et al.*[22)] proposed robust pattern matching for viewpoint change by the extraction of a SIFT descriptor[23)] from a normalized patch generated by projecting an input image to a local plane around the landmark. However, it is still difficult for this method to determine the correspondences for close landmarks because the visual aspects of close landmarks are easily changed even for a small viewpoint change.

In this study, we focus on the feature landmark-based method[5)], which can be easily applied in large-scale and complex environments. By solving the problems of computational cost and accuracy, we develop a fast and accurate camera parameter estimation method for implementing AR applications.

3. Overview of Landmark-based Camera Parameter Estimation

In this section, the basic framework of the feature landmark-based camera parameter estimation method[5)] is briefly reviewed. The feature landmark-based method is composed of the offline stage, which comprises database construction, and the online stage, which comprises camera parameter estimation, as shown in Figure 2.

3.1 Database Construction

The feature landmark database must be constructed before starting the online camera parameter estimation. In this process, first, 3-D information of the target environment is acquired by SfM. Next, landmark information is generated from the SfM results and they are registered to the database.

3.1.1 Three-dimensional Reconstruction by SfM

Three-dimensional reconstruction of the target environment is achieved by SfM for an omnidirectional camera[18),19)], as shown in Figure 3. In this process, first, the target environment is captured in the form of omnidirectional video sequences. Next, natural features in the captured video are detected and tracked using the Harris corner detector[24]. Three-dimensional positions of natural features and extrinsic camera parameters of the omnidirectional camera are estimated by SfM. In this SfM process, several known 3-D points[18] or GPS measurements[19] can be used to suppress accumulative estimation error. Again, by using this additional information, we obtain the 3-D information in the global coordinate system.

3.1.2 Acquisition of Landmark Information

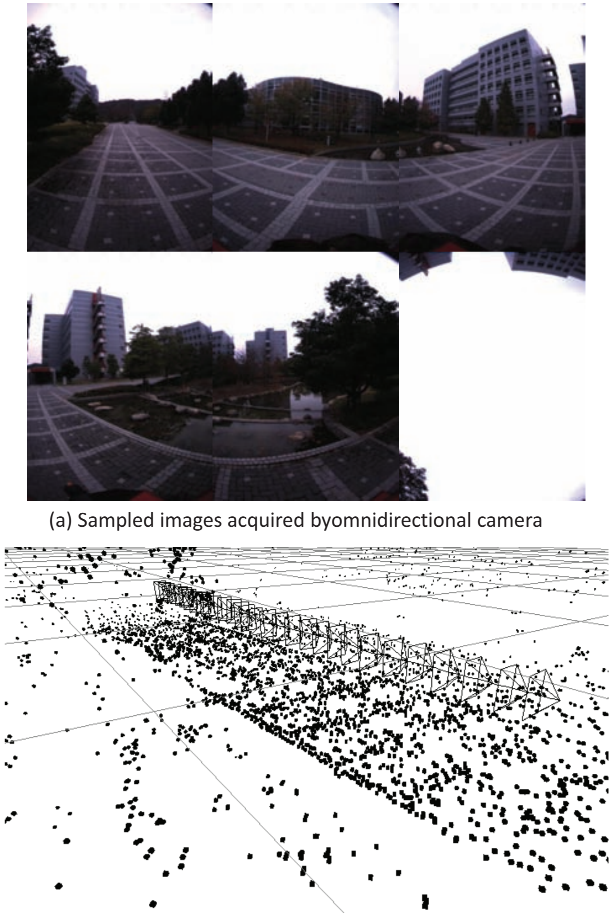
The feature landmark database consists of a number of landmarks. The 3-D coordinate and viewpoint dependent information associated with each landmark is stored in the database. Viewpoint dependent information consists of captured positions and image templates of the landmark.

Three-dimensional coordinate of landmark: Three-dimensional positions of natural features obtained using the SfM process are registered to the database as landmarks. In the online stage, extrinsic camera parameters are estimated from correspondences between the 3-D positions of landmarks in the database and the 2-D positions of natural features in the input image.

Viewpoint dependent information: Viewpoint dependent image templates of landmarks are generated and then registered to the database to deal with visual aspect change of landmarks. To generate these image templates, first, a local plane that is perpendicular to the line connecting the 3-D position of a landmark with the projection center of the omnidirectional camera is defined, as shown in Figure 4. Next, pixel values of the image templates are determined by projecting the captured image to the local plane. The generated image templates are then registered to the database. Positions of the omnidirectional camera, from which image templates are generated, are also registered to the database as the index for landmark selection in the online stage.

3.2 Camera Parameter Estimation

In this process, first, camera parameters for the first frame are estimated using the landmark-based camera parameter estimation method for a still image input[25]. Next, the landmark selection, corresponding pair search, and camera parameter estimation processes are repeated.



(a) Sampled images acquired byomnidirectional camera

(b) SfM result

Fig. 3 Sampled images and SfM result used for database construction.

3.2.1 Landmark Selection from Database

Observable landmarks from the user's viewpoint are selected from the database as matching candidates for natural features in the input image. To select observable landmarks, the following criteria are used.

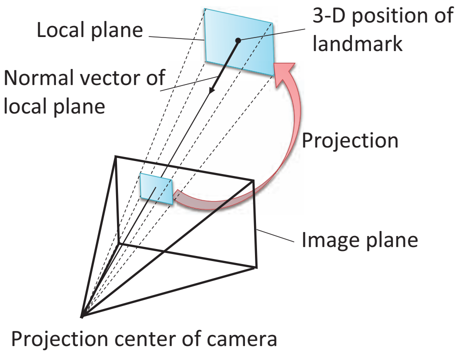


Fig. 4 Generation of viewpoint dependent image template.

- Landmarks must be projected onto the input image by using camera parameters of the previous frame $\boldsymbol{M}_{t-1}$.
- The angle between the normal vector of the image template of the landmark and the vector from the camera position of the previous frame to the landmark must be under the threshold θ.

Landmarks that satisfy the above requirements are selected from the database, and then, observable landmarks are narrowed down to a certain number N and arranged in the ascending order of the distance between the camera position of the previous frame and the captured position of the landmark. In order for the landmarks to be evenly distributed over the input image, we divide the input image into a grid, and only one landmark is selected for each grid.

3.2.2 Search for Corresponding Pairs and Camera Parameter Estimation

Camera parameters are estimated from correspondences between landmarks and image features. In this process, first, landmarks selected from the database are projected onto the input image using the camera parameters $\boldsymbol{M}_{t-1}$ as follows:

$$\begin{bmatrix} a_iu_i & a_iv_i & a_i \end{bmatrix}^T = \boldsymbol{M}_{t-1} \begin{bmatrix} x_i & y_i & z_i & 1 \end{bmatrix}^T \quad (1)$$

where, (x_i, y_i, z_i) and (u_i, v_i) represent the 3-D position and 2-D position of landmark i, respectively. a_i represents the depth of landmark i in the camera coordinate system.

Next, natural features within a fixed window W whose center is located at (u_i, v_i) are selected as matching candidates. Image patterns of natural features are then compensated for in the same manner as that in the database construction process. Corresponding pairs of landmarks and image features are searched using normalized cross-correlation (NCC).

After determining the corresponding pairs, extrinsic camera parameters $\boldsymbol{M}_t$ are estimated by solving the PnP problem[26]. To remove outliers, the LMedS estimator[27] is applied to this process.

The computational cost for this matching process C_{prev} is as follows:

$$C_{prev} = NFA \quad (2)$$

where, N represents the number of selected landmarks, F represents the average number of natural features in the window W, and A represents the testing cost for each feature, including compensation of visual pattern and calculation of similarity measures.

4. Reduction of Computational Cost by Priority-based Landmark Selection and Landmark Tracking

This section describes the method to reduce the computational cost of the corresponding pair search process described in 3.2.2. The computational cost of the matching process is given by Eq. (2). In this equation, the testing cost A cannot be easily reduced for successful matching because the computation of a similarity measure requires that for the difference between the cameras used in the offline stage and the camera used in the online stage be compensated. Thus, in our method, the numbers of candidates matching landmarks N and natural features F are reduced by the tentative camera parameter estimation (B-2) and landmark selection based on priorities of landmarks (B-3). Details of the proposed method are described in the following sections.

4.1 Tentative Camera Parameter Estimation by Landmark Tracking

In order to reduce the number of candidates matching natural features F in Eq. (2), we carried out the new process of estimating tentative camera parameters of the current frame is newly estimated by landmark tracking in successive frames. In this process, first, the landmarks that are used to estimate camera parameters in the previous frame are selected and then tracked to the current frame. In the successive frames, the visual patterns of landmarks hardly change and compensation for patterns is not necessary. Thus, in this tracking process, visual patterns around the projected positions of the landmarks in the previous frame are used as image templates of landmarks in the current frame, and the sum of squared differences (SSD) is simply used as the similarity measure. It should be noted that matching candidates in the current frame are limited to natural features within the window W whose center is located at the position of the matched landmark in the previous frame. After finishing the landmark tracking process, outliers are rejected by the LMedS estimator, and then, tentative camera parameters in the current frame $\hat{\boldsymbol{M}}_t$ are estimated by solving the PnP problem using the tracked landmarks. It should be noted that estimation of tentative camera parameters fails when the number of tracked landmarks is less than six or the rate of outliers is over 50% in our system. In this case, camera tracking is terminated in current our implementation. In order to recover the camera parameter estimation process after failure, re-initialization techniques suggested in Reitmayr *et al.*[28], and Williams *et al.*[29] can be employed.

The computational cost for tentative camera parameter estimation C_{track} is as follows:

$$C_{track} = N_{track}FB + E_{LMedS} \tag{3}$$

where, N_{track} represents the number of tracked landmarks, B represents the calculation cost of the SSD for each pair, and E_{LMedS} represents the cost for outlier rejection and camera parameter estimation. Tentative camera parameter estimation can be achieved with the lower computational cost because the matching cost B is much smaller than that of A in Eq. (2).

4.2 Landmark Selection Based on Priorities

In this process, the number of landmarks N in Eq. (2) is reduced using a geometric constraint and assigning priorities of landmarks. The approach of assigning priorities to landmarks is newly considered in the proposed method. The priority P_i of the landmark i is defined as the probability that landmark i is used in the online camera parameter estimation, and it is given as follows:

$$P_i = \frac{E_i}{D_i} \tag{4}$$

where, E_i represents the frequency that the landmark i is used as the inlier in the camera parameter estimation process (B-5), and D_i represents the frequency that the landmark i is selected from the database in the landmark selection process (B-3). In this paper, we assume that in order to set priorities, the system administrator trains the system with several videos captured in the target environment before the system is used by users.

In this landmark selection process, first, observable landmarks except for the landmarks tracked in process (B-2) are selected from the database using a geometric constraint that is almost the same as the one used in the previous method. Next, top N_{prior} confident landmarks are selected from the observable landmarks. It should be noted that several landmarks (N'_{track}) have already been matched in the tentative camera parameter estimation process (B-2) before starting the landmark selection process. Therefore, in this process, the maximum number of N_{prior} is fixed as N_{max}, and N_{prior} is determined by subtracting the number of tracked landmarks N'_{track} from N_{max}. Using priorities of landmarks, we can efficiently select the small number of landmarks to be used in the next process (B-4).

4.3 Search for Corresponding Pairs and Camera Parameter Estimation

This process is basically the same as that used in the previous method except for the range of search window. First, selected landmarks are projected onto the input image using tentative camera parameters $\hat{\boldsymbol{M}}_t$. Next, corresponding pairs of landmarks and natural features are searched within the fixed window W', whose center is located at the projected position in the input image. Using tentative camera parameters, the window size of W' can be made smaller than that of the process (B-2). Finally, extrinsic camera parameters are estimated by using corresponding pairs and tracked landmarks.

The computational cost of new matching process (B-4) is as follows:

$$C_{proj} = N_{prior}F\frac{S'}{S}A \tag{5}$$

where, S and S' represent the sizes of windows W and W', respectively. By estimating the tentative camera parameters, we reduce the number of matching candidates of natural features by S'/S.

4.4 Updating Priorities

After finishing the camera parameter estimation, we update the priorities of landmarks using frequency information obtained from the result of the camera parameter estimation as follows:

$$P_i = \frac{E_{iold} + E_{inew}}{D_{iold} + D_{inew}} \qquad (6)$$

where, E and D are frequency information described in Section 4.2. Subscripts *iold* and *inew* denote the past and current video sequences, respectively.

4.5 Comparison of Computational Cost

The ideal effect of the computational cost reduction in matching process (B-4) can be computed from Eqs. (2), (3), and (5) as follows:

$$\frac{C_{new}}{C_{prev}} = \frac{C_{track} + C_{proj}}{C_{prev}} \qquad (7)$$

$$= \frac{C_{track}}{C_{prev}} + \frac{N_{max} - N'_{track}}{N} \frac{S'}{S} \qquad (8)$$

where, C_{new} is the matching cost in the proposed method. In this equation, the first term and the second term represent the overhead for tentative camera parameter estimation in the process (B-2) and the effect of computational cost reduction in the process (B-5), respectively. In fact, the effect of computational cost reduction does not perfectly conform with this equation because of the cost of the overhead in the iteration process. The actual effect of the cost reduction will be demonstrated in the experiment.

4.6 Experiment

The computational cost is compared with that of the previous method[5]. We take an omnidirectional sequence in the target environment and then the feature landmark database is constructed using SfM[18] and an omnidirectional camera (Point Grey Research, Inc.; Ladybug) in the outdoor environment. Figure 3 shows the sampled images used for database construction and the SfM result. In this experiment, about 12400 landmarks are registered to the database and each landmark has 8 image templates on average. For the proposed method, we captured three training videos of the target environment to determine the priorities of landmarks. Camera paths of these training sequences are almost the same as the test sequence. To evaluate the proposed and previous methods, we also capture another video image sequence (720 × 480 pixels, progressive scan, 15 fps, 1,000 frames). For a quantitative evaluation, we generated the ground truth by the estimating camera parameters with manually specified correspondences of landmarks. It should be noted that we have removed several frames in which the reprojection error of the obtained ground truth is over 1.5 pixels. Table 1 shows the parameters for this experiment.

To verify the effectiveness of the proposed method, the following four methods are compared.

Method A: Previous method[5]
Method B: Proposed method without landmark selection based on priorities
Method C: Proposed method without tentative camera parameter estimation
Method D: Proposed method

In this experiment, first, in order to determine the number of landmarks to be selected, we compared the rate of estimation failure. Next, the computational cost of these methods is compared.

Figure 5 shows the number of failure frames for various number of selected landmarks in process (B-3). In this experiment, we deemed the result to be a failure when the number of corresponding pairs is less than 6. Methods A and B, which did not use priorities of landmarks, failed to estimate the camera parameter for several frames when the number of landmarks was 70 or less. Methods C and D, which use priorities of landmarks, did not fail when the number of landmarks

Table 1 Parameters in experiment.

	Previous method[5]	Proposed method
Image template size in process (B-2) (pixel)	-	15
Window size W (pixel)	-	120 × 60
Window size W' (pixel)	120 × 60	20 × 20
Angle threshold θ (degree)	15	
Number of grids in input image	74 × 48	
Training data	-	Three sequences
Initial value of priority	-	1/2

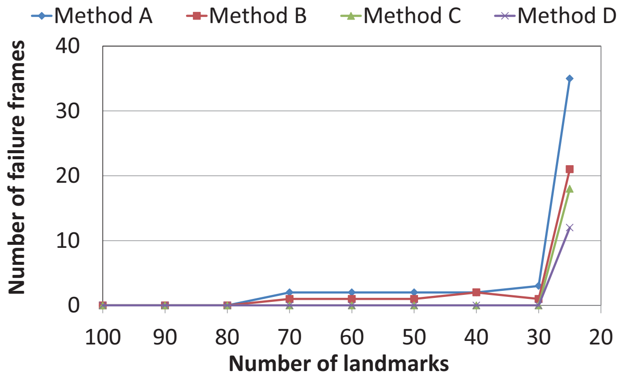


Fig. 5 Relation between number of landmarks and failure frames.

was more than 30. From these results, we determine the number of landmarks as required 80 for the methods A and B and 30 for the methods C and D. Table 2 shows the processing time for each method when we used a laptop PC (CPU: Core2 Extreme 2.93 GHz, Memory: 2 GB). For method D, which involved the estimation of tentative camera parameters and selection of landmarks with high priorities, the total computational cost was about six times lower than that of the method A. As a result, the proposed method can work at video rate. The computational cost of matching process (B-4) was 21 times lower than that of the method A. However, Eq. (8) indicates that ideally, the effect of the computational cost reduction would make method D over 48 times cheaper than that of the method A ($N = 80, N_{prior} - N'_{track} \leq 30, S'/S = 1/18$). This difference between the ideal and real outcomes is caused by the cost of overhead. Table 3 shows the accuracy of each method. From this result, methods B, C, and D improved the accuracy of estimated camera parameters. We think this improvement is obtained by selecting small number of confident matching candidates.

Table 2 Comparison of processing time for one frame (ms).

Method	A	B	C	D
Process (B-2)	-	26	-	21
Process (B-3)	12	3	2	1
Process (B-4)	316	51	131	15
Process (B-5)	61	16	16	17
Overhead	4	4	4	5
Total cost	393	100	153	59

Table 3 Comparison of accuracy.

Method	A	B	C	D
Avg. position error (mm)	360	257	231	256
Std. dev. position error (mm)	528	137	204	181
Avg. posture error (degree)	0.84	0.95	1.13	0.91
Std. dev. posture error (degree)	0.71	1.20	1.16	0.91
Avg. reprojection error (pixel)	2.5	2.3	2.1	1.8

4.7 Summary

In this section, we achieved fast feature landmark based camera parameter estimation by reducing the matching candidates. The number of feature points are reduced by estimating tentative camera parameters. The number of landmarks are reduced by using priorities of landmarks. In the experiment, our method can achieve online camera parameter estimation in video rate. In addition, the accuracy of online camera parameter estimation is improved by selecting small number of confident matching candidates using above strategies.

5. Accuracy Improvement using Laser Range Sensor

This section describes the method for database construction using the dense depth map to improve the accuracy of online camera parameter estimation. In the offline stage, the dense depth map is obtained by using the laser range sensor at the spot where user will come close to CG objects. In this place, landmark information is also collected by using the dense depth map in addition to landmark information collected by the SfM. In the online stage, these collected landmarks are used by considering the difference of 3-D measuring method.

5.1 Acquisition of Depth Map and Surface Texture

In this process, 3-D information of the target environment is acquired by using the omnidirectional camera and the omnidirectional laser range sensor as shown in Figure 6. In this scanning process, geometric relationship $\boldsymbol{M}_{rc}$ between these sensors are fixed and calibrated in advance. The depth map corresponding to an omnidirectional image is generated from the range data obtained by the range sensor using geometric relationship $\boldsymbol{M}_{rc}$. It should be noted that the laser range sensor has a limit of measuring range. The obtained depth map involves lack areas including the sky area. If we simply mask and ignore them, the aperture problem will be caused against landmarks which exist at the boundary between landscape and the sky. However, these landmarks are important to determine the camera posture. To avoid the aperture problem, in this study, infinite depth values are set for the sky areas. Concretely, the largest region without depth values in the omnidirectional image is determined as the sky area.

5.2 Acquisition of Landmark Information

Landmark information is generated from the acquired depth map and the omnidirectional image. First, natural feature points are detected from the omnidirectional image by using Harris corner detector[24] and then 3-D positions of these features are determined from the depth map. These natural features are registered to the database as landmarks. In order to deal with view dependent information in the same as the previous method, the ground plane is divided into the grid whose center is the sensor position and then image templates of landmarks are generated for every grid point as shown in Figure 7. Concretely, first, virtual camera is set at the grid point. Next, visible landmarks at the position of the virtual camera are determined by using the range data and then depth value for each pixel on the image template of the landmark is obtained from range data. Finally, pixel values on the image template are determined by projecting the omnidirectional image using these depth values. In this pattern generation, occluded areas in the image template are set as masked areas in order to ignore them in the pattern matching process (B-4).

5.3 Merging of Landmarks from SfM and Laser Range Sensor

In the proposed method, two kinds of landmarks obtained by the SfM (LDB_{SfM}) and the range sensor (LDB_{Laser}) can be merged seamlessly. However, in the online process (B-5), LMedS estimator tends to select landmarks only from one side. This one-sided selection causes a jitter problem at the position where both landmarks obtained by the SfM and those by the range sensor can be observed. To avoid this problem, we newly add the constraint to the random sampling process (constrained random sampling). Concretely, two different random sampling processes are executed depending on the situation.

- If all of the temporal pairs for the LMedS are selected from either LDB_{SfM} or LDB_{Laser}, general random sampling is used.
- Otherwise, the samples which do not contain either LDB_{SfM} or LDB_{Laser}t are immediately rejected in the repeating process of LMedS.

By this strategy, one sided selection of landmarks is avoided. Effect of constrained

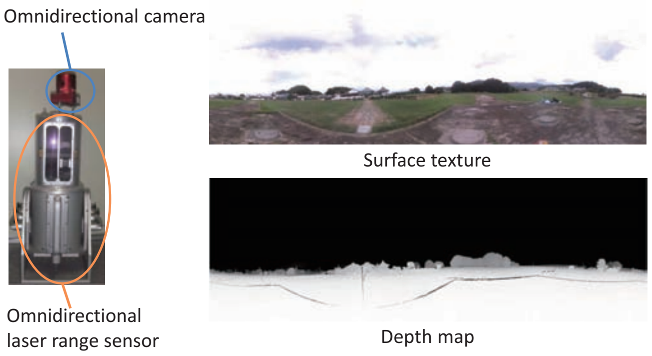


Fig. 6 Sensor setup and Omnidirectional data.

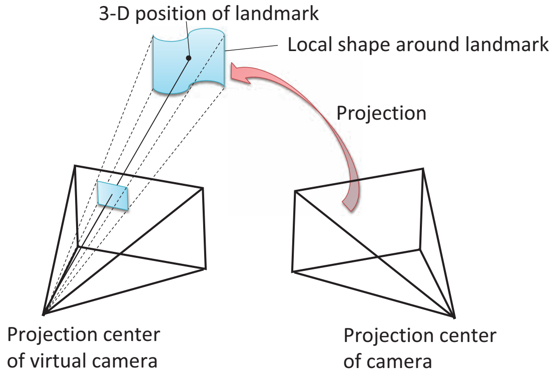

3-D position of landmark
Local shape around landmark
Projection
Projection center of virtual camera
Projection center of camera

Fig. 7 Generation of image template by considering local 3-D structure around the landmark.

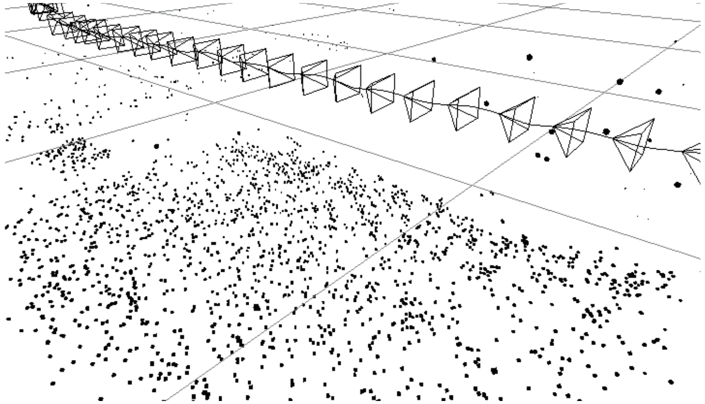


Fig. 8 SfM result.

random sampling will also demonstrated in the experiment.

5.4 Experiment

In this experiment, to demonstrate the effectiveness of combination of the landmark database constructed using the SfM and that by the laser range sensor, the effectiveness of pattern compensation by considering local 3-D structure of the landmark is evaluated and then the accuracy of estimated camera parameters is compared to the method which uses only SfM-based database construction. The range data is obtained using the omnidirectional laser range sensor (Reigl Inc. LMS-Z360) and one omnidirectional sequence is captured in the target environment. Specifications of this sensor are shown in Table 4. Figure 6 and Figure 8 show acquired surface texture as well as corresponding depth map and a result of SfM, respectively. The ground plane of the target environment is divided into 10 × 10 grid points at 1 meter intervals for range sensor based landmark acquisition. Constructed feature landmark database consists of about 8800 landmarks (LDB_{SfM}) and about 3500 landmarks (LDB_{Laser}). The video image sequence (720 × 480 pixels, progressive scan, 15fps, 450 frames) captured in the target environment is used as the input video for the evaluation of online camera parameter estimation. The parameters in online camera parameter estimation are same as the section 4.6. For the quantitative evaluation, we made the ground truth by the same way as the section 4.6. In this experiment, the maximum distance between the omnidirectional camera path and the monocular camera path was about 3 meters.

Table 4 Specifications of laser range sensor

Measurable range	1m∼100m
Measurement accuracy	±12mm
Measurable angle	Horizontal: 360° Vertical: -50° ∼ 40°
Step angle	0.08°

First, to verify the effectiveness of the pattern compensation using the dense depth map, generated image templates of landmarks by using the process (A-2.1) which uses sparse depth information and the process (A-2.2) which uses dense depth information are quantitatively evaluated by comparing them with ground truth. In this experiment, viewpoints for pattern compensation are given by estimating camera parameters with manually specified correspondences of landmarks in input images. Table 5 shows average and standard deviation of NCC values between compensated image templates and image patterns of landmarks in input images for 30 image templates of landmarks. By using the dense depth information, the average NCC value (0.63) is higher than that of the method which does not consider the local 3-D structure around the landmark (0.47). Figure 9 shows the generated image patterns. It is confirmed that the image templates of landmarks are adequately compensated for by considering the local 3-D structure around the landmark.

Next, the accuracy of estimated camera parameters using the database constructed by the SfM and the range sensor (SfM+Range method) is compared to the method which uses only SfM-based database construction (SfM method). Figure 10 shows corresponded landmarks used to estimate camera parameters. As can be seen in this figure, although the SfM method finds small number of corresponding landmarks at the ground part of the images, a lot of correspond-

Table 5 Comparison of normalized cross-correlation value.

	Using dense depth information	Using sparse depth information
Avg.	0.63	0.47
Std. dev.	0.039	0.052

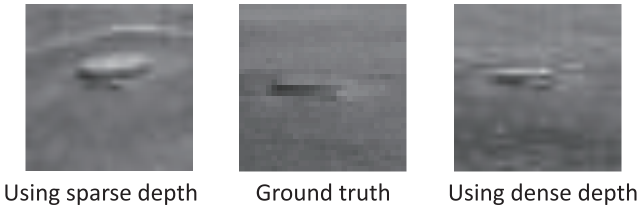


Fig. 9 Generated image templates of the landmark.

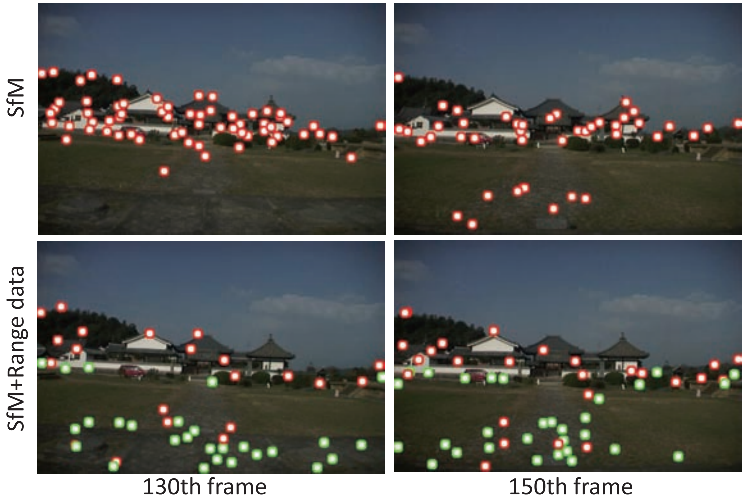


Fig. 10 Corresponded landmarks. Red circles indicate landmarks measured by SfM. Green circles indicate landmarks measured by range sensor.

ing pairs of landmarks and feature points were found for the ground part in the SfM+Range method. This is regarded as the effect of the pattern compensation using dense 3-D information. Figure 11 shows error in position for each frame. The accuracy of estimated camera parameters which only uses the database constructed by the range sensor (Range method) is also shown in this figure. It should be noted that the range method cannot estimate camera parameters in the entire sequence. The effect of the SfM+Range can be confirmed because the accuracy of the SfM+Range is improved to the same level of that by the Range method for the places where range data are available. Average position errors for the SfM+Range, Range, and SfM methods are 282mm, 229mm, and 543mm, respectively. In this experiment, the SfM+Range method has used landmarks measured by the range sensor during frame number 82 to 301. From this result, it is confirmed that the SfM+Range method can improve the accuracy of estimated camera parameters for most of the frames. This improvement is gained from corresponded landmarks close to the user.

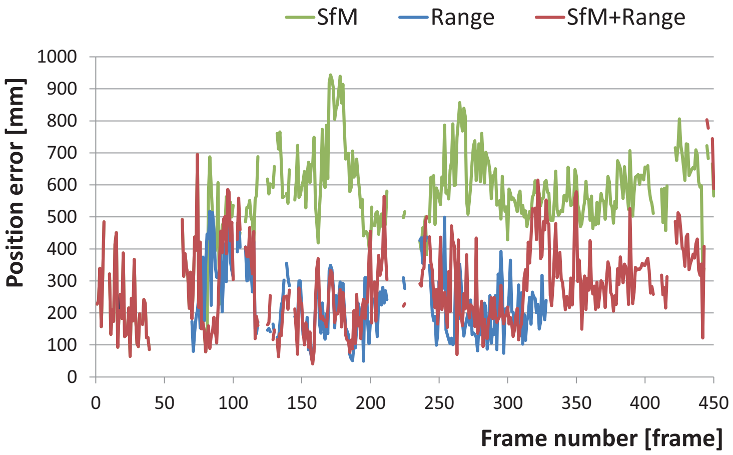


Fig. 11 Error in position for each frame. Red line indicates result of proposed method, which uses both SfM and range data. Blue line indicates the result of the method that uses only Range. Green line indicates the result of the method that uses only SfM.

5.5 Summary

In this section, we proposed the accuracy improvement of online camera parameter estimation at the spot where CG objects must be placed near the user's viewpoint by using the laser range sensor. Unlike other methods, the landmarks close to the user's viewpoint that effect the accuracy of geometric registration are aggressively used by compensating its visual patterns based on dense depth information acquired by using omni-directional range finder. Importance of close landmarks are validated quantitatively through the experiment.

6. Applications

To show the usefulness of the proposed method, the proposed method is applied to three different kinds of applications.

- Outdoor Navigation
- MR-PreViz
- Virtual Historical Experience

Database construction methods for each application are shown in Table 6. In virtual historical experience, we use both SfM and the laser range sensor to construct the database because users will come close to the CG objects in this application. Details of applications are described in the following sections.

6.1 Outdoor Navigation

In this application, the feature landmark database is constructed by the SfM in the campus shown in Figure 12. Navigation information is manually aligned in advance. In the scenario of user navigation, low cost development of the database for wide-range area is more important than alignment errors of several pixel levels. Thus, we didn't use the laser range sensor. In this experiment, we used the video camera (Sony DSR-PD150) and the laptop PC (CPU: Core 2 Quad 3.0 GHz, Memory: 4 GB). Figure 13 shows the result of AR navigation. It is confirmed that annotation information is overlaid at geometrically correct positions. By using AR for navigation, navigation information is intuitively given to the user. However, in the initialization process in the online process, the camera must be fixed during the initial camera parameter estimation because this initialization process needs about 30 seconds. In order to realize practical application, speed-up of the initialization process or estimation of user's movement is needed.

6.2 MR-PreViz

In the pre-production of filmmaking, pre-visualization techniques are employed in order to test a camera work and acting. Conventionally, a pre-visualization movie has been created by computer graphics. On the other hand, a MR pre-visualization (MR-PreViz) technique, which creates the pre-visualization movie using both real images and CG-rendered actors on site, has been proposed. From the MR-PreViz, directors and actors can easily grasp the camera work and acting. Currently, most of MR-PreViz methods[30),31)] are designed for indoor environments. In order to realize MR-PreViz in outdoor environments, we applied our method to MR-PreViz. In this experiment, we used the SfM[18)] to construct the database. For the online process, we used the video camera (Sony DSR-PD150) and the laptop PC (CPU: Core 2 Extreme 2.93 GHz, Memory: 2 GB) Figure 14(a) and Figure 14(b) show the detected landmarks in the input image and MR-PreViz images, respectively. Our method has successfully worked in such a natural environment. PreViz images are generated at 15 frames per second. Although MR-PreViz images include a minimum of 1/15 seconds delay, it was little problem for the actual application.

Table 6 Database construction method

	Method
Outdoor navigation	SfM[18)]
MR-PreViz	SfM[18)]
Virtual historical experience	SfM[18)] and laser range sensor

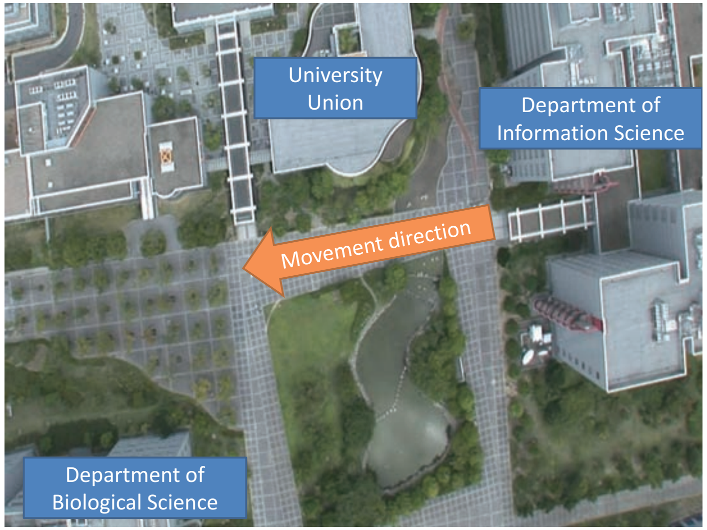

University Union
Department of Information Science
Movement direction
Department of Biological Science

Fig. 12 Overhead view of the target environment.

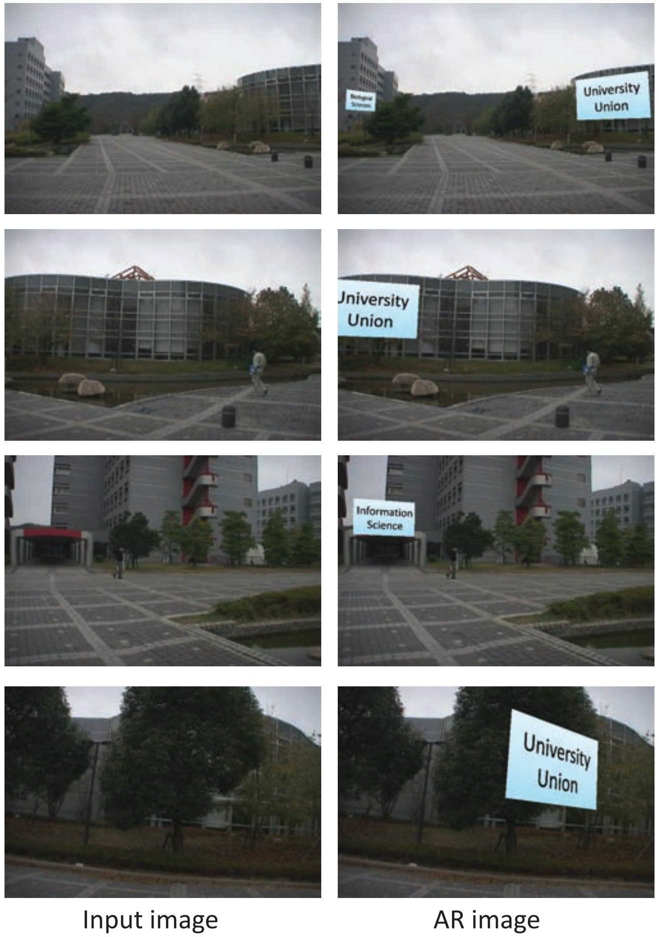
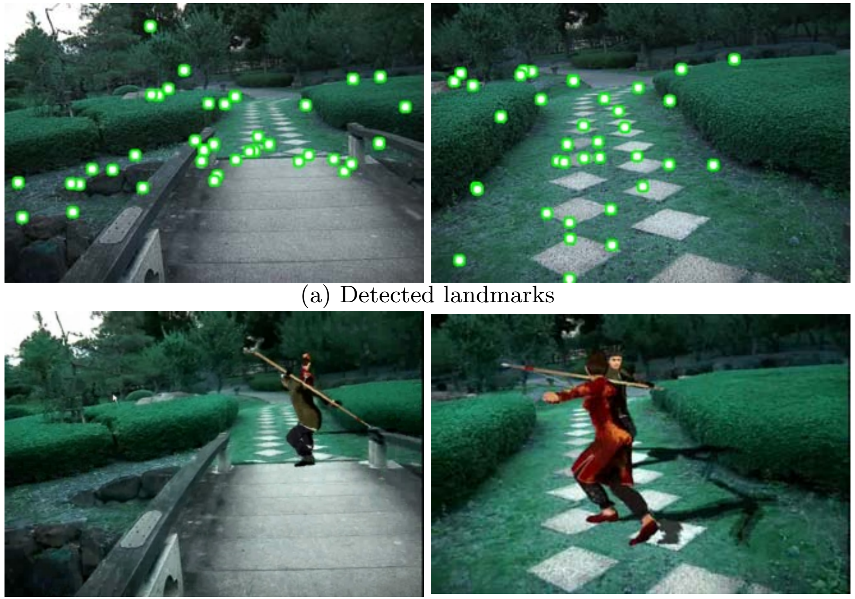


Fig. 13 AR navigation.

6.3 Virtual Historical Experience

There are many dilapidated historical sites worldwide. In these places, AR applications could enable visitors to visualize their original appearance of the cultural heritage. In this experiment, temple ruins in the ancient Japanese capital city of Asuka are virtually reconstructed at its original site. In this scenario, users would come close to virtual objects. Therefore, a feature landmark database is constructed using SfM[18] and the laser range sensor. In this experiment, we used the same equipments of the AR navigation in the online stage. Figure 15 shows the result of the AR sightseeing. Virtual objects are overlaid on the site of the old temple. We have confirmed that CG objects placed at the position close to the user's viewpoint are correctly registered. The AR sightseeing can realize virtual historical experience.



(a) Detected landmarks

(b) Overlaid CG actors

Fig. 14 MR-PreViz. Green circles indicate detected landmarks in input images.

7. Conclusion

In this study, we have proposed the real-time and accurate camera parameter estimation method using the feature landmark database for outdoor AR. To achieve real-time processing, matching candidates of landmarks and natural

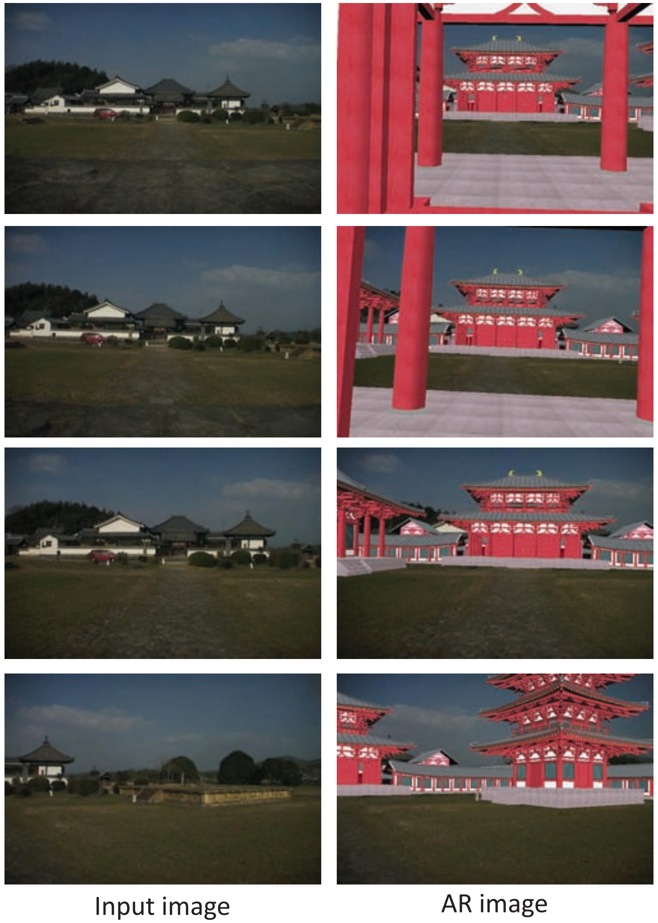


Fig. 15 AR sightseeing.

features are efficiently reduced by using tentative camera parameter estimation and priority-based landmark selection and confident matching candidate selection affected the improvement of the accuracy of camera parameter estimation. The accuracy of estimated camera parameters is improved by using the dense depth map obtained by the laser range sensor at the spot where virtual objects are placed near the user's viewpoint. Importance of close landmarks is quantitatively validated through the experiment. The feasibility of the proposed method was demonstrated by applying the proposed method to some AR applications.

Currently, the feature landmark database must be rebuilt when appearance of the scene is partially or completely changed *e.g.* by construction of new buildings and season change. Our next challenge in this project is to develop a strategy for updating the landmark database using images captured by a user's camera. This will reduce the cost of constructing and maintaining the landmark database.

Acknowledgments This research is supported in part by the "Ambient Intelligence" project granted by the Ministry of Education, Culture, Sports, Science and Technology.

References

1) Wagner, D. and Schmalstieg, D.: First steps towards handheld augmented reality, *Proc. Int. Symp. on Wearable Computers*, pp.21–23 (2003).

2) Miyashita, T., Meier, P., Tachikawa, T., Orlic, S., Eble, T., Scholz, V., Gapel, A., Gerl, O., Arnaudov, S. and Lieberknecht, S.: An augmented reality museum guide, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.103–106 (2008).

3) Kaufmann, H. and Dunser, A.: Summary of usability evaluations of an educational augmented reality application, *Proc. Int. Conf. on Virtual Reality*, No.10, pp.660–669 (2007).

4) Dähne, P. and Karigiannis, J.: Archeoguide: System architecture of a mobile outdoor augmented reality system, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.263–264 (2002).

5) Oe, M., Sato, T. and Yokoya, N.: Estimating camera position and posture by using feature landmark database, *Proc. Scandinavian Conf. on Image Analysis*, pp. 171–181 (2005).

6) Davison, A., Mayol, W. and Murray, D.: Real-time localization and mapping with wearable active vision, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.18–27 (2003).

7) Eade, E. and Drummond, T.: Scalable monocular SLAM, *Proc. Conf. on Computer Vision and Pattern Recognition*, pp.469–476 (2006).

8) Chekhlov, D., Gee, A.P., Calway, A. and Mayol-Cuevas, W.: Ninja on a Plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual SLAM, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.153–156 (2007).

9) Klein, G. and Murray, D.: Parallel tracking and mapping for small AR workspaces, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.225–234 (2007).

10) Bleser, G., Wuest, H. and Stricker, D.: Online camera pose estimation in partially known and dynamic scenes, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp. 56–65 (2006).

11) Drummond, T. and Cipolla, R.: Real-time visual tracking of complex structure, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.27, No.7, pp.932–946 (2002).

12) Comport, A., Marchand, E., Pressigout, M. and Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework, *IEEE Trans. on Visualization and Computer Graphics*, Vol.12, No.4, pp.615–628 (2006).

13) Lepetit, V., Vacchetti, L., Thalmann, D. and Fua, P.: Stable real-time 3D tracking using online and offline information, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.26, No.10, pp.1391–1402 (2004).

14) Vacchetti, L., Lepetit, V. and Fua, P.: Combining edge and texture information for real-time accurate 3D camera tracking, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.48–57 (2004).

15) Skrypnyk, I. and Lowe, D. G.: Scene modelling, recognition and tracking with invariant image features, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp. 110–119 (2004).

16) Arth, C., Wagner, D., Klopschitz, M., Irschara, A. and Schmalstieg, D.: Wide Area Localization on Mobile Phones, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.73–82 (2009).

17) Neubert, J., Pretlove, J. and Drummond, T.: Semi-autonomous generation of appearance-based edge models from image sequences, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.79–89 (2007).

18) Sato, T., Ikeda, S. and Yokoya, N.: Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-Directional Multi-Camera System, *Proc. European Conf. on Computer Vision*, Vol.Vol. 2, pp.326–340 (2004).

19) Ikeda, S., Sato, T., Yamaguchi, K. and Yokoya, N.: Construction of feature landmark database using omnidirectional videos and GPS positions, *Proc. Int. Conf. on 3-D Digital Imaging and Modeling*, pp.249–256 (2007).

20) Irschara, A., Zach, C., Frahm, J. and Horst, B.: From structure-from-motion point clouds to fast location recognition, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.2599–2606 (2009).

21) Nistér, D. and Stewenius, H.: Scalable Recognition with a Vocabulary Tree, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.2161–2168 (2006).

22) Wu, C., Clipp, B., Li, X., Frahm, J. and Pollefeys, M.: 3D model matching with viewpoint-invariant patches (VIP), *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.1–8 (2008).

23) Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints, *Int. J. of Computer Vision*, Vol.60, No.2, pp.91–100 (2004).

24) Harris, C. and Stephens, M.: A Combined Corner and Edge Detector, *Proc. Alvey Vision Conf.*, pp.147–151 (1988).

25) Susuki, M., Nakagawa, T., Sato, T. and Yokoya, N.: Extrinsic camera parameter estimation from a still Image based on feature landmark database, *Proc. ACCV'07 Satellite Workshop on Multi-dimensional and Multi-view Image Processing*, pp.124–129 (2007).

26) Klette, R., Schluns, K., Koschan, A. and editors: *Computer Vision: Three-dimensional Data from Image*, Springer (1998).

27) Rousseeuw, P.J.: Least Median of Squares Regression, *J. of the American Statistical Association*, Vol.79, pp.871–880 (1984).

28) Reitmayr, G. and Drummond, T.: Going out: robust model-based tracking for outdoor augmented reality, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp. 109–118 (2006).

29) Williams, B., Klein, G. and Reid, I.: Real-time SLAM relocalisation, *Proc. Int. Conf. on Computer Vision* (2007).

30) Shin, M., s. Kim, B. and Park, J.: AR storyboard: An augmented reality based Interactive storyboard authoring tool, *Proc. Int. Symp. on Mixed and Augmented Reality*, pp.198 – 199 (2005).

31) Thomas, G.: Mixed Reality Techniques for TV and their Application for On-Set and Pre-Visualization in Film Production, *Proc. Int. Workshop on Mixed Reality Technology for Filmmaking*, pp.31–36 (2005).