Available online at www.sciencedirect.com
ScienceDirect
Computers and Graphics 00 (2011) 1–13
Computers and Graphics

Real-time and Accurate Extrinsic Camera Parameter Estimation using Feature Landmark Database for Augmented Reality

Takafumi Taketomi, Tomokazu Sato, Naokazu Yokoya

Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{takafumi-t,tomoka-s,yokoya}@is.naist.jp

Abstract

In the field of augmented reality (AR), many kinds of vision-based extrinsic camera parameter estimation methods have been proposed to achieve geometric registration between real and virtual worlds. Previously, a feature landmark-based camera parameter estimation method was proposed. This is an effective method for implementing outdoor AR applications because a feature landmark database can be automatically constructed using the structure-from-motion (SfM) technique. However, the previous method cannot work in real time because it entails a high computational cost or matching landmarks in a database with image features in an input image. In addition, the accuracy of estimated camera parameters is insufficient for applications that need to overlay CG objects at a position close to the user's viewpoint. This is because it is difficult to compensate for visual pattern change of close landmarks when only the sparse depth information obtained by the SfM is available. In this paper, we achieve fast and accurate feature landmark-based camera parameter estimation by adopting the following approaches. First, the number of matching candidates is reduced to achieve fast camera parameter estimation by tentative camera parameter estimation and by assigning priorities to landmarks. Second, image templates of landmarks are adequately compensated for by considering the local 3-D structure of a landmark using the dense depth information obtained by a laser range sensor. To demonstrate the effectiveness of the proposed method, we developed some AR applications using the proposed method.

Keywords: extrinsic camera parameter estimation, natural features, landmark database, augmented reality

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. In recent years, AR has become very important in the market growth of the mobile devices, including smartphones and portable game devices. Unfortunately, at present, the AR experience using these devices is does not practically applicable because of the large registration error between real and virtual environments. Generally, the position and posture of the camera embedded in a mobile device should be correctly estimated to achieve geometric registration between real and virtual environments. Current AR applications rely on embedded sensors such as GPS, magnetic compass, and gyroscope; however, the errors introduced by the use of such devices are large and are directly translated into registration errors in the AR image. Vision-based methods are known as an effective solution for reducing registration errors because they can directly compute and reduce alignment errors on the image. In fact, for mobile devices in small workspaces, the vision-based camera parameter estimation method is often employed with fiducial markers. However, as the employment of fiducial markers is impractical for implementing outdoor AR applications, many kinds of natural feature-based

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

methods have been developed in the past decade.

Previously, we proposed a feature landmark-based camera parameter estimation method as a geometric registration method for implementing outdoor AR applications [1]. This method uses the structure-from-motion (SfM) technique for omnidirectional camera system to automatically and efficiently estimate the 3-D positions of natural feature points in large-scale outdoor environments; these positions are then used as landmarks of known 3-D positions for camera parameter estimation. However, there remained several problems for implementing a practical AR system in an outdoor environment using this method. One is the high computational cost of calculating similarity measures between a large number of pairs of landmarks and feature points. The other is the poor accuracy of geometric registration in specific conditions. In this study, we focused on these two problems and solved them by adopting the following ideas.

(1) Priorities are assigned to landmarks for effectively omitting unnecessary computation of similarity measures between landmarks and feature points.

(2) Dense depth information for specific sites acquired by an omnidirectional laser range sensor is integrated in the feature landmark database to improve the accuracy of geometric registration.

It should be noted that SfM-based methods for landmark database construction are sufficient for most parts of the target environment. However, if both the following conditions are satisfied, registration error easily becomes large in an AR image.

(a) Landmarks that exist near the viewpoint (close landmarks) are not detected.

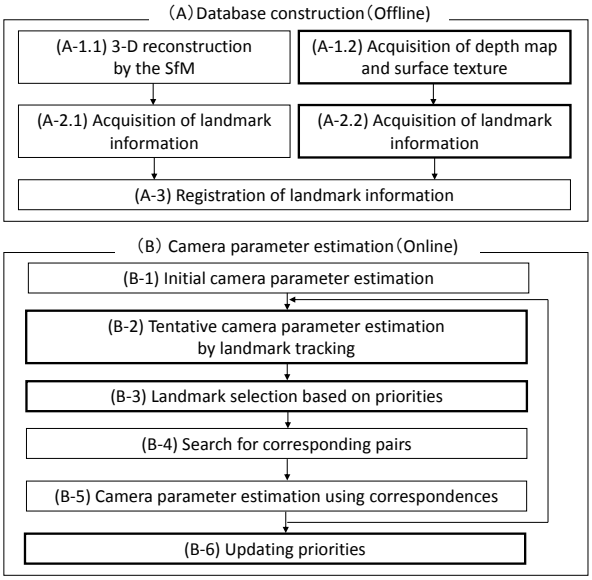

Figure 2: Flow of proposed method. The thick squares indicate the new processes employed in this study.

(b) CG objects are drawn near the viewpoint, as in the case shown in Figure 1.

Condition (a) is caused when the camera goes far away from the original viewpoints of the SfM or when feature points do not exist around the user's viewpoint. For these places, we measure dense depth information using the laser range sensor, and the information from the laser range sensor is integrated with the data from SfM. Although several methods that use either the SfM or the laser range sensor for constructing the database have already been reported [1–4], as far as we know, the combination of the SfM and the laser range sensor for balancing the database construction cost and the accuracy has not been discussed in the AR community.

Figure 2 shows a flow diagram of the proposed method. Our method is composed of two stages. In offline stage (A), a landmark database, which contains 3-D positions of landmarks and associated visual information, is constructed by using SfM and the laser range sensor. In the online stage (B), both landmark tracking and priorities of the landmarks are used, resulting in a noticeable decrement in the computational cost. It should be noted that the laser range sensor is not used in the online stage (B). Contributions of this article are the following.

- Suggestion of priority based landmark selection and landmark tracking for reducing computational cost in the online stage.

- Suggestion of the combined use of the SfM and the laser range sensor for balancing the database construction cost and the accuracy of geometric registration.
- Verifications of the importance of close landmarks and the effectiveness of above two suggestions in real outdoor environments.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 reviews the basic framework of the feature landmark-based camera parameter estimation method [1]. Then, the reduction in computational cost and the improvement in accuracy are described in Sections 4 and 5, respectively. The effectiveness of the proposed method is quantitatively evaluated in Section 6. Finally, Section 7 presents the conclusion and outlines the future work.

2. Related Works

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 [5–9] that estimates camera parameters without preknowledge of target environments. The other is a preknowledge-based approach [1–3, 10–13] that uses 3-D information of target environments.

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 [5–9]. This method can easily construct an AR environment without premeasurement of the target environment. The disadvantage of visual-SLAM methods is that they cannot determine the absolute position for arrangement of CG objects. This implies that, by itself, the approach cannot be directly used for position-dependent AR applications such as navigation and landscape simulation. Another problem of visual-SLAM is that estimation errors are accumulated, if we use only visual information. It causes the drift of overlaid CGs for large-scale outdoor environments. To determine the coordinate system, Bleser *et al*. use a partially known 3-D model [9] and Klein *et al*. employ interactive initialization [8]. However, these approaches are impractical for a large-scale outdoor environment because they require manual arrangement of CGs and coordinate system by users themselves.

The other uses some kinds of preknowledge of target environments to estimate camera parameters in the global coordinate system [1–3, 10–13]. In this approach, 3-D models [10–13] and feature landmarks [1–3] are used as preknowledge of target environments. Three-dimensional model based methods are used only for small workspaces because they require large human effort 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 [14] 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 the correspondences of landmarks in a database with image features in an input image. Skrypnyk *et al*. [2] and Arth *et al*. [3] 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 [1], we used accumulative error free SfM for an omnidirectional camera, and the feature landmark database is efficiently constructed even in a large-scale and complex environment [15, 16]. 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. To reduce the computational cost, Skrypnyk *et al*. employ approximate nearest neighbor search in the matching process [2]. In order to achieve fast retrieval of matching candidates of landmarks from the database, Arth *et al*. [3] limit the number of visible landmarks using a potential visible set, and Irschara *et al*. [17] employ the vocabulary tree [18] to retrieve landmarks. In our previous method, most of the computational time is spent on pattern compensation to handle the difference between the camera properties of the omnidirectional camera used in the database construction process and that of the monocular camera used in the online camera parameter estimation process. This cost could not be reduced by using the previously proposed computational cost reduction approaches [2, 3, 17]. 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*. [4] proposed robust pattern matching for viewpoint change by the extraction of a SIFT descriptor [19] 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 [1], 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. Basic Framework of Feature Landmark-based Camera Parameter Estimation

In this section, the basic framework of the feature landmark-based camera parameter estimation method [1] 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 registered to the database.

3.1.1. Three-dimensional reconstruction by the SfM

Three-dimensional reconstruction of the target environment is achieved by SfM for an omnidirectional camera [15, 16], 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 [20]. Three-dimensional positions of natural features and extrinsic camera parameters of the omnidirectional camera are estimated by the SfM. In this SfM process, several known 3-D points [15] or GPS measurements [16] 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.

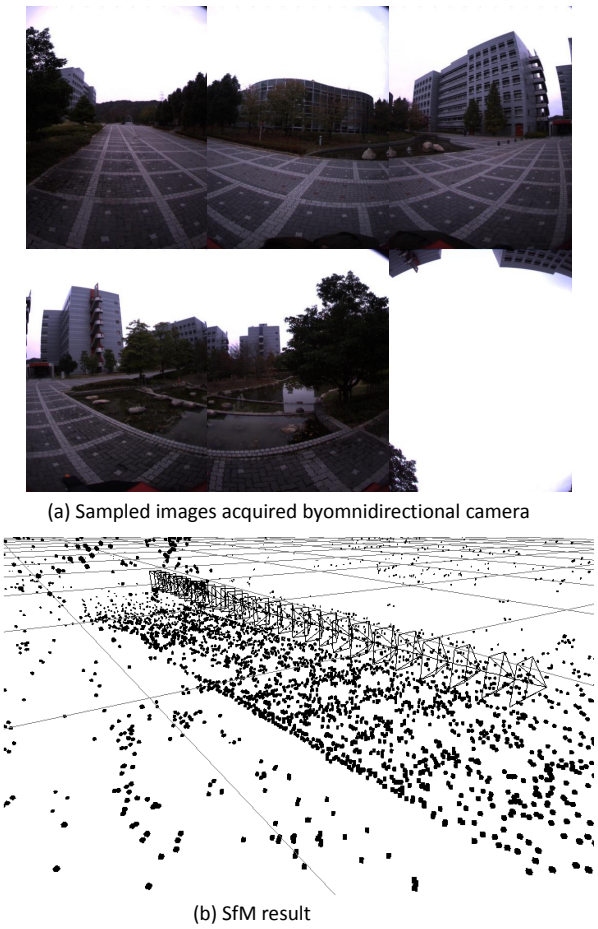

Figure 3: Sampled images and SfM result used for database construction.

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

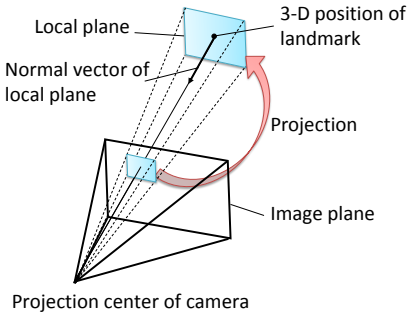

Figure 4: Generation of viewpoint dependent image template.

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, we assume camera parameters for the first frame have already been given using the landmark-based camera parameter estimation method for a still image input [21] or other methods. Next, the landmark selection, corresponding pair search, and camera parameter estimation processes are repeated.

3.2.1. Landmark selection from the 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.

- 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:

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

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 [22]. To remove outliers, the LMedS estimator [23] 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 for Fast Camera Parameter Estimation

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

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*. [24], and Williams *et al*. [25] 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 cost of calculating 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 a computational cost lower than that of conventional camera parameter estimation because the matching cost B in Eq. (3) is much lower than the testing cost 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:

$$P_i = \frac{E_{iold} + E_{inew}}{D_{iold} + D_{inew}} \tag{6}$$

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:

$$\frac{C_{new}}{C_{prev}} = \frac{C_{track} + C_{proj}}{C_{prev}} \tag{7}$$

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

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.

5. Improvement of the Accuracy of Camera Parameter Estimation

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 the user comes close to CG objects. In this place, we fuse landmark information from SfM and the laser range sensor.

5.1. Acquisition of Depth Map and Surface Texture

In this process, 3-D information of the target environment is acquired using the omnidirectional camera and the omnidirectional laser range sensor, as shown in Figure 5. In this scanning process, the geometric relationship $\boldsymbol{M}_{rc}$ between these sensors is 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 limited measuring range. The obtained depth map lack information about the sky area. If we simply mask and ignore these areas, the aperture problem will arise for 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 method, we set infinite depth values for the sky areas. In practice, 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 using the Harris corner detector [20], and then, 3-D positions of these features are determined from the depth map. These natural features are registered to the database as landmarks. The viewpoint dependent information is processed in the same manner as that in the previous method: the ground plane is divided into a grid whose center coincides with the sensor position, and then, image templates of landmarks are generated for every grid point, as shown in Figure 6. In practice, first, a 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, the depth value for each pixel on the image template of the landmark is obtained from the range data. Finally, pixel values on the image template are determined by projecting the omnidirectional image using these depth values. In the pattern generation process, occluded areas in the image template are set as masked areas in order to exclude them from consideration during the pattern matching process (B-4).

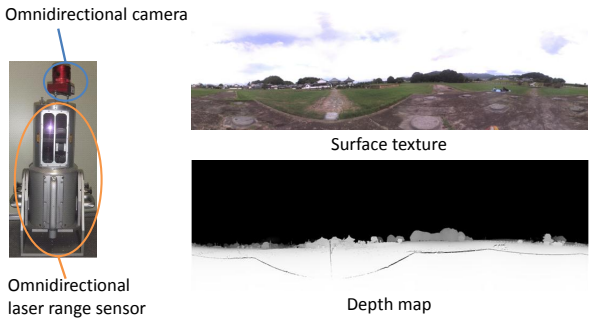

Figure 5: Sensor setup and omnidirectional data.

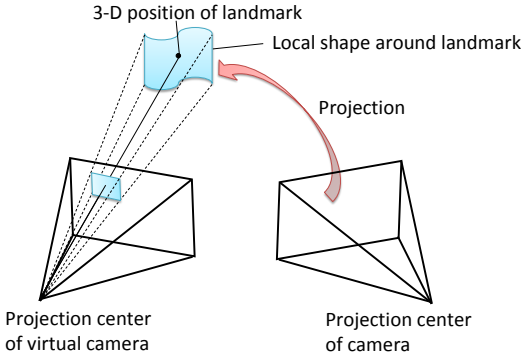

Figure 6: Generation of image template by considering local 3-D structure around landmark.

5.3. Merging of Landmarks from SfM and Laser Range Sensor

In the proposed method, the 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), the LMedS estimator tends to select landmarks only from one side. This one-sided selection causes a jitter problem at positions where the landmarks obtained by both SfM and the range sensor can be observed. To avoid this problem, we add a new constraint to the random sampling process (constrained random sampling). In practice, 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 that do not contain either LDB_{SfM} or LDB_{Laser} are immediately rejected in the repeating process of LMedS.

By using this strategy, we avoid one-sided selection of landmarks. The effect of constrained random sampling will also be demonstrated in the experiment.

6. Experiments

In this experiment, to demonstrate the effectiveness of the proposed method, first, we compare the computational cost of this method with that of the previous method [1]. Next, in order to validate the effectiveness of combining the landmark database constructed using SfM and that constructed using the laser range sensor, the accuracy of the estimated camera parameters is compared to that of the method that uses only SfM-based database construction. The usefulness of the proposed method is also demonstrated by applying the proposed method to some AR applications. In these experiments, intrinsic camera parameters of the monocular camera used in the online stage are calibrated in advance using the Tsai method [26].

6.1. Effectiveness of Computational Cost Reduction

The computational cost is compared with that of the previous method [1]. We take an omnidirectional sequence in the target environment and then the feature landmark database is constructed using SfM [15] 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 [1]

Method B: Proposed method without landmark selection based on priorities

Table 1: Parameters in experiment.

	Previous method [1]	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

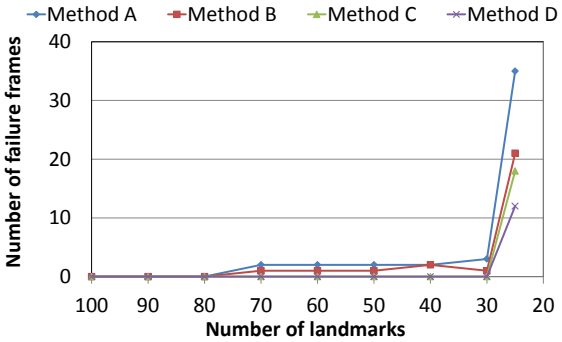

Figure 7: Relation between number of landmarks and failure frames.

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 7 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 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, we conclude that methods B, C, and D can reduce computational cost without increasing estimation error.

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

6.2. Effectiveness of Accuracy Improvement

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 (Riegl Inc.; LMS-Z360) and one omnidirectional sequence is captured in the target environment. Specifications of this sensor are shown

Table 4: Specifications of laser range sensor

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

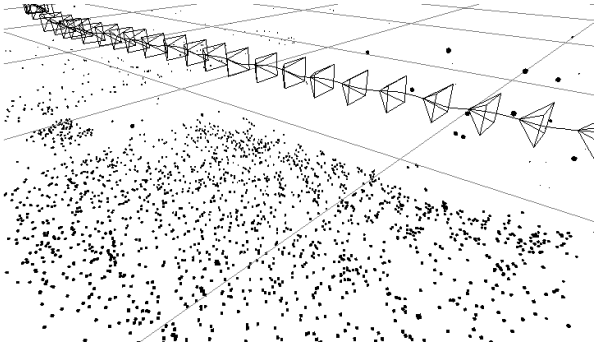

Figure 8: SfM result.

in Table 4. Figure 5 shows the acquired surface texture as well as the corresponding depth map, Figure 8 shows a SfM result. The ground plane of the target environment is divided into 10 × 10 grid points at 1-m 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, 15 fps, 450 frames) captured in the target environment is used as the input video for the evaluation of online camera parameter estimation. The parameters used in online camera parameter estimation are same as Section 6.1. For the quantitative evaluation, we generated the ground truth in the same manner as that described in Section 6.1. In this experiment, the maximum distance between the omnidirectional camera path and the monocular camera path was about 3-m.

First, to verify the effectiveness of the pattern compensation using the dense depth map, we quantitatively evaluate the generated image templates of landmarks using process (A-1.2), which uses sparse depth information, and process (A-2.2), which uses dense depth information by comparing them with ground truth. In this experiment, the viewpoints for pattern compensation are provided by estimating camera parameters with manually specified correspondences of landmarks in input images. Table 5 shows the 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. The average NCC value obtained using the dense depth information (0.63) is higher than that obtained using 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.

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

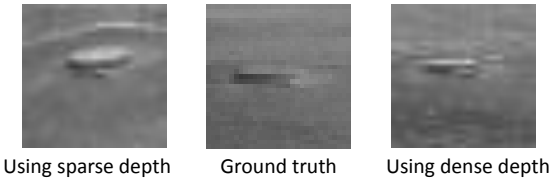

Figure 9: Generated image templates of landmark.

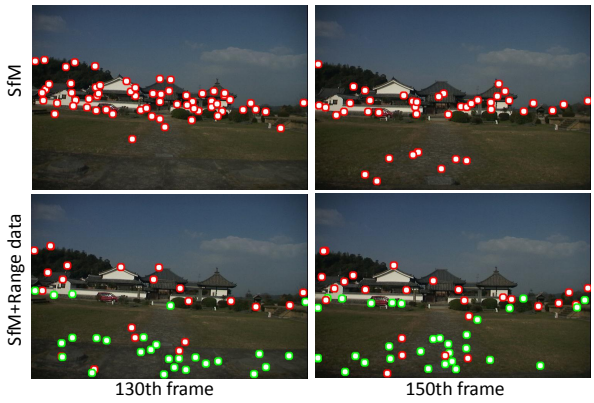

Figure 10: Corresponded landmarks. Red circles indicate landmarks measured by SfM. Green circles indicate landmarks measured by range sensor.

Next, the accuracy of estimated camera parameters obtained using the database constructed by SfM and the range sensor (SfM+Range method) is compared to the accuracy of those obtained by the method that uses only SfM-based database construction (SfM method). Figure 10 shows the corresponded landmarks used to estimate camera parameters. As can be seen in this figure, although the SfM method finds a small number of corresponding pairs of landmarks and feature points for

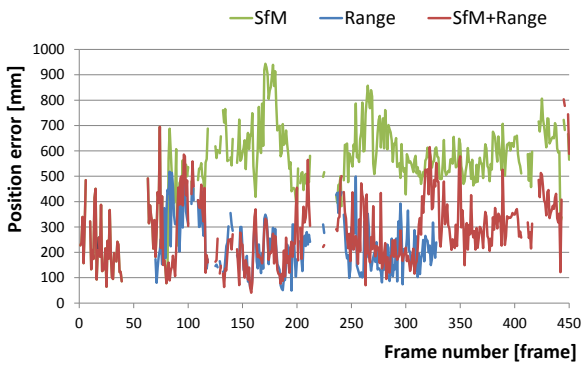

Figure 11: Error in position for each frame. Red line indicates result of proposed method, which uses both SfM and range data. Blue line indicates the result of the method that uses only Range. Green line indicates the result of the method that uses only SfM.

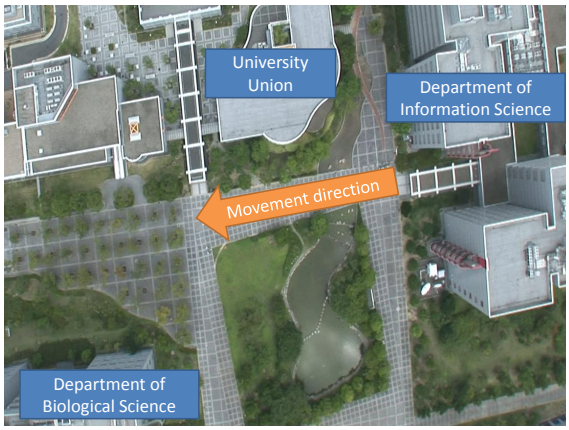

Figure 12: Overhead view of the target environment.

the ground part of the images, the SfM+Range method finds many more such pairs. This is considered to be due to 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. The average position errors for the SfM+Range, Range, and SfM methods are 282 mm, 229 mm, and 543 mm, respectively. In this experiment, the SfM+Range method has used landmarks measured by the range sensor during frame number 82 to 301. This result confirms that the SfM+Range method can improve the accuracy of estimated camera parameters for most of the frames. This improvement is due to the accurate matching of corresponded landmarks close to the user. In addition, the effect of accuracy improvement of estimated camera parameters has been confirmed in the generated AR video[1]. It is observed that when both kinds of landmarks are used, jitter in the generated video is suppressed as compared with that in the SfM method.

6.3. Applications

To show the usefulness of the proposed method, we apply the proposed method to two applications: AR outdoor navigation and AR sightseeing.

AR Navigation

In this application, the feature landmark database for the campus shown in Figure 12 is constructed by SfM. Navigation information is manually created and aligned in advance. In this scenario, the users do not come close to virtual objects. Thus, we do not use the laser range sensor. In this experiment, we used a video camera (Sony DSR-PD150) and a 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[2]. By using AR for navigation, the system intuitively provides navigation information to the user.

AR Sightseeing

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 and the laser range sensor. In the online stage of this experiment, we used the same equipment as that used for the AR navigation application. Figure 14 shows the result of the AR sightseeing application. Virtual objects are overlaid on the site of the old temple. We have confirmed that CG objects placed at a position close to the user's viewpoint are correctly registered[3]. The AR sightseeing

[1] http://yokoya.naist.jp/research2/LandmarkVideo/JitterComparison.wmv

[2] http://yokoya.naist.jp/research2/LandmarkVideo/result-navi.wmv

[3] http://yokoya.naist.jp/research2/LandmarkVideo/result-ARSightseeing.wmv

Figure 13: AR navigation result.

Figure 14: AR sightseeing result.

application can realize a virtual historical experience.

7. Conclusion

In this paper, we proposed a real-time and accurate camera parameter estimation method using the feature landmark database for implementing outdoor AR applications. To achieve real-time processing, we efficiently reduced the number of matching candidates of landmarks and natural features 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 the estimated camera parameters is improved using the dense depth map obtained by a 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 usefulness 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] M. Oe, T. Sato, N. Yokoya, Estimating camera position and posture by using feature landmark database, Proc. Scandinavian

Conf. on Image Analysis (2005) 171–181.
[2] I. Skrypnyk, D. G. Lowe, Scene modelling, recognition and tracking with invariant image features, Proc. Int. Symp. on Mixed and Augmented Reality (2004) 110–119.
[3] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, D. Schmalstieg, Wide area localization on mobile phones, Proc. Int. Symp. on Mixed and Augmented Reality (2009) 73–82.
[4] C. Wu, B. Clipp, X. Li, J. Frahm, M. Pollefeys, 3D model matching with viewpoint-invariant patches (VIP), Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008) 1–8.
[5] A. Davison, W. Mayol, D. Murray, Real-time localization and mapping with wearable active vision, Proc. Int. Symp. on Mixed and Augmented Reality (2003) 18–27.
[6] E. Eade, T. Drummond, Scalable monocular SLAM, Proc. Conf. on Computer Vision and Pattern Recognition (2006) 469–476.
[7] D. Chekhlov, A. P. Gee, A. Calway, W. Mayol-Cuevas, Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam, Proc. Int. Symp. on Mixed and Augmented Reality (2007) 153–156.
[8] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, Proc. Int. Symp. on Mixed and Augmented Reality (2007) 225–234.
[9] G. Bleser, H. Wuest, D. Stricker, Online camera pose estimation in partially known and dynamic scenes, Proc. Int. Symp. on Mixed and Augmented Reality (2006) 56–65.
[10] T. Drummond, R. Cipolla, Real-time visual tracking of complex structure, IEEE Trans. on Pattern Analysis and Machine Intelligence 27 (7) (2002) 932–946.
[11] A. Comport, E. Marchand, M. Pressigout, F. Chaumette, Real-time markerless tracking for augmented reality: the virtual visual servoing framework, IEEE Trans. on Visualization and Computer Graphics 12 (4) (2006) 615–628.
[12] V. Lepetit, L. Vacchetti, D. Thalmann, P. Fua, Stable real-time 3d tracking using online and offline information, IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (10) (2004) 1391–1402.
[13] L. Vacchetti, V. Lepetit, P. Fua, Combining edge and texture information for real-time accurate 3D camera tracking, Proc. Int. Symp. on Mixed and Augmented Reality (2004) 48–57.
[14] J. Neubert, J. Pretlove, T. Drummond, Semi-autonomous generation of appearance-based edge models from image sequences, Proc. Int. Symp. on Mixed and Augmented Reality (2007) 79–89.
[15] T. Sato, S. Ikeda, N. Yokoya, Extrinsic camera parameter recovery from multiple image sequences captured by an omnidirectional multi-camera system, Proc. European Conf. on Computer Vision Vol. 2 (2004) 326–340.
[16] S. Ikeda, T. Sato, K. Yamaguchi, N. Yokoya, Construction of feature landmark database using omnidirectional videos and GPS positions, Proc. Int. Conf. on 3-D Digital Imaging and Modeling (2007) 249–256.
[17] A. Irschara, C. Zach, J. Frahm, B. Horst, From structure-from-motion point clouds to fast location recognition, Proc. IEEE Conf. Computer Vision and Pattern Recognition (2009) 2599–2606.
[18] D. Nistér, H. Stewenius, Scalable recognition with a vocabulary tree, Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006) 2161–2168.
[19] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. of Computer Vision 60 (2) (2004) 91–100.
[20] C. Harris, M. Stephens, A combined corner and edge detector, Proc. Alvey Vision Conf. (1988) 147–151.
[21] M. Susuki, T. Nakagawa, T. Sato, N. Yokoya, 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 (2007) 124–129.
[22] R. Klette, K. Schluns, A. Koschan, editors, Computer Vision: Three-dimensional Data from Image, Springer, 1998.
[23] P. J. Rousseeuw, Least median of squares regression, J. of the American Statistical Association 79 (1984) 871–880.
[24] G. Reitmayr, T. Drummond, Going out: robust model-based tracking for outdoor augmented reality, Proc. Int. Symp. on Mixed and Augmented Reality (2006) 109–118.
[25] B. Williams, G. Klein, I. Reid, Real-time SLAM relocalisation, Proc. Int. Conf. on Computer Vision.
[26] R. Y. Tsai, An efficient and accurate camera calibration technique for 3D machine vision, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1986) 364–374.