Vol. 48 No. SIG 9(CVIM 18) IPSJ Transactions on Computer Vision and Image Media June 2007

Regular Paper

Acquisition and Rectification of Shape Data Obtained by a Moving Range Sensor

Atsuhiko Banno† and Katsushi Ikeuchi††

"Modeling from Reality" techniques are making great progress because of the availability of accurate geometric data from three-dimensional digitizers. These techniques contribute to numerous applications in many areas. Among them, one of the most important and comprehensive applications is modeling cultural heritage objects. For a large object, scanning from the air is one of the most efficient methods for obtaining 3D data. We developed a novel 3D measurement system, the Floating Laser Range Sensor (FLRS), in which a range sensor is suspended beneath a balloon. The obtained data, however, have some distortions due to movement of the system during the scanning process. We propose two novel methods to rectify the shape data obtained by the moving range sensor. One method rectifies the data by using image sequences; the other rectifies the data without images. To test these methods, we have conducted a digital archiving project of a large-scale heritage object, in which our algorithms are applied. The results show the effectiveness of our methods. Moreover, both methods are applicable not only to our FLRS, but also to moving range sensors in general.

1. Introduction

1.1 Background

Many research projects on real object modeling are making great progress because of the availability of accurate geometric data from three-dimensional digitizers. The techniques of real object modeling contribute to numerous applications in areas such as academic investigation, industrial management, and entertainment.

Among them, one of the most important and comprehensive applications is modeling cultural heritage objects. Modeling these objects is of significance. Models lead to digital archives of the objects' shapes. Utilizing these archives enables us to restore the original shapes of the objects, even if they have been destroyed due to natural weathering, fire, disasters and wars. In addition, we can provide images of these objects through the Internet to people in their homes or in their offices. Thus, the techniques of real object modeling are available for many applications.

Many research groups have been conducting projects to model large objects such as statues, historical buildings and suburban landscapes [8),16),23)]. Basically, to scan these large objects, laser range finders are usually used with tripods positioned on stable locations. In the case of scanning a large object, however, it often happens that some part of the object is not visible from the laser range finder on the ground. To overcome this difficulty, researchers have scanned large objects from scaffolds temporally constructed near the object. However, this scaffold method requires costly, tedious construction time. In addition, it may be impossible to scan some parts of the object due to the limitation of available space for scaffold-building.

We are now conducting a project [17)] to model the Bayon Temple [35)] in Cambodia. The temple's scale is about 150 × 150 square meters with over 40 meter height. Scanning such a huge object from several scaffolds is unrealistic. To overcome this problem, several methods have been proposed. For example, aerial 3D measurements can be obtained by using a laser range sensor installed on a helicopter platform [33)]. High frequency vibration of the platform, however, prevents the acquisition of highly accurate data. Also, to avoid irrevocable destruction, the use of heavy equipment such as a crane should be eschewed in the case of scanning a cultural heritage object.

Based upon the above considerations, we proposed a novel 3D measurement system, a Floating (or Flying) Laser Range Sensor (FLRS) [14),15)]. This system digitizes large objects from the air while suspended from the underside of a balloon platform (**Fig. 1**). The bal-

† Institute of Industrial Science, The University of Tokyo
†† Interfaculty Initiative in Information Studies, The University of Tokyo

Fig. 1 The FLRS and the Bayon Temple.

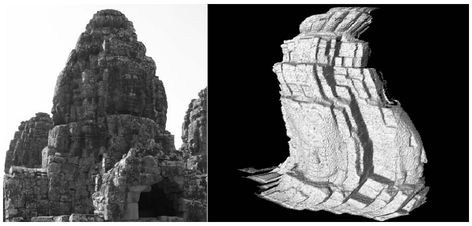


Fig. 2 An sample snap shot and the distorted range data obtained by the FLRS.

loon platform is certainly free from high frequency vibration such as that caused by a helicopter engine. The obtained range data are, however, distorted because the laser range sensor itself is moving during the scanning processes (**Fig. 2**).

1.2 Our Contributions

In this study, we propose two methods to rectify 3D range data obtained by a moving laser range sensor. Not only is this method limited to the case of our FLRS, but it is also applicable to a moving range sensors in general.

Several attempts have been made to rectify deformed range data by a moving range sensor. The following three strategies have been considered to solve this kind of problem:

- Window matching-based method [14)]
- 3D registration-based method [15),22)]
- Structure from motion-based method [1)]

In Ref. 14), under the assumption that translation of the sensor is very small and within a plane parallel to the image plane without any rotation, the shape is recovered by using a video sequence image. Then, supposing that the changes in sequential images are very small, the sensor's motion is estimated by a local window matching technique. This method is very fast, but it restricts the sensor to a simple and small motion.

In Refs. 15) and 22), the sensor's motion is parametrized by the specification of a velocity vector that maintains a uniform linear motion with a constant angular velocity. Then, an extended ICP algorithm is applied to align the deformed model obtained by the moving range sensor with the correct model obtained by another range sensors located on the ground. This method does not require image sequences, but it assume the simple motions.

In our study, we adopt two strategies for the rectification; one method is based on the third strategy and another is based on the second one.

In accordance with the third strategy, we proposed a method based on "Structure from Motion" using image sequences and distorted range data obtained by the FLRS. The image sequences are obtained by a video camera mounted on the FLRS and the range data are obtained by a moving range sensor. The motion of the FLRS is roughly estimated only by the obtained images. Then the more refined parameters are estimated based on an optimization imposing some constraints, which include information derived from the distorted range data itself. Finally, using the refined camera motion parameters, the distorted range data are rectified.

In accordance with the second strategy based on "3D registration," we adopt a method similar to that described in Refs. 15) and 22), but supposing smooth and more generalized balloon motion.

These methods are not limited to the case of our FLRS but also applicable to a general moving range sensor that has smooth motion. In this study, we do not utilize physical sensor such as gyros, INS and GPS for estimation of position and pose. We try to solve our problems only by range sensors and video cameras through the techniques of *Computer Vision*.

This article is organized as follows. We briefly explain our FLRS system in Section 2. In Section 3, we explain a full perspective factorization, which is utilized as the initial value for the camera motion. We use a weak perspective factorization iteratively for the perspective projection camera model. In Section 4, we describe our proposed algorithm for refinement of the parameters. Our method applies three constraints for the optimization, which are tacking, smoothness and range data constraint. Applying these constraints and optimizing the cost function, we can estimate more precise parameters. In Section 5, we describe another method for shape rectification which need not any image sequences. Instead of using images, this method requires range data obtained by another range sensor fixed on the ground. In Section 6, we evaluate our algorithms with known models. Constructing a virtual FLRS on a PC by using a CG model, we estimate the accuracy of our method. In Section 7, we show several experimental results of applying our algorithms for range data processing conducted in the Bayon Temple in Cambodia. Finally, we present our conclusions and summarize our possible future works in Section 8.

2. FLRS

The FLRS (Floating Laser Range Sensor) was developed to measure large objects from the air by using a balloon without constructing any scaffolds (**Fig. 3**).

We have two types of FLRSs. Each FLRS is composed of a scanner unit, a controller and a personal computer (PC). These three units are suspended beneath a balloon.

The scanner unit includes a laser range finder, especially designed to be suspended from a balloon. **Figure 4** shows the interior of the scanner unit. It consists of a spot laser radar unit and two mirrors. We chose the LARA25200 and LARA53500 supplied by Zoller+Fröhlich GmbH [37] as laser radar units because of their high sampling rate. Each laser radar unit is mounted each FLRS scanner unit. Two systems equipped with Lara25200 and LARA53500 are respectively referred to as "25 m sensor" and "50 m sensor". The specifications of two units are shown in **Table 1**.

Both sensors have the similar mirror configurations. There are two mirrors inside each unit to give direction to the laser beam. One is a polygon mirror with four reflection surfaces, which determines the azimuth of the beam. In normal use, the polygon mirror, which rotates rapidly, controls the horizontal direction of the laser beam. Another is a plane mirror (swing mirror) which determines the elevation of the beam. The plane mirror swings slowly to controls the vertical direction of the laser beam.

The laser beam emitted from the LARA is hit on a surface of the polygon mirror at first. Then the polygon mirror reflects the laser beam into the plane mirror. The plane mirror also reflects the beam into the outside of the unit (lower of Fig. 4).

Fig. 3 The FLRS (25 m sensor).

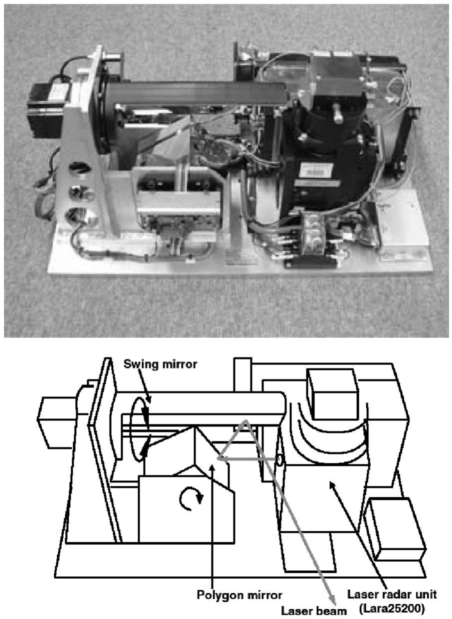


Fig. 4 The interior of scanner unit (25 m sensor).

Table 1 The specifications of the 25 m (LARA25200) and 50 m (LARA53500) Sensors; w.r.t. laser source.

	25 m Sensor	50 m Sensor
Ambiguity interval (m)	25.2	53.5
Minimum range (m)	1.0	1.0
Resolution (mm)	1.0	1.0
Sampling rate (pix/s)	$\leq$ 625,000	$\leq$ 500,000
Linearity error (mm)	$\leq$ 3	$\leq$ 5
Range noise at 10 m (mm)	$\geq$ 1.0	$\geq$ 1.5
Range noise at 25 m (mm)	$\geq$ 1.8	$\geq$ 2.7
output power (mW)	23	32
Laser wavelength (nm)	780	780

The combination of two mirrors demonstrates the specifications as shown in **Table 2**.

3. Full Perspective Factorization

Estimations of the shape of an object or of camera motion by using images are called "Shape from Motion" or "Structure from Mo-

Table 2 The specifications of the 25 m sensor and 50 m sensor; w.r.t. viewing field.

	25 m Sensor	50 m Sensor
Angle Resolution		
Horizontal (deg)	0.05	0.05
Vertical (deg)	0.02	0.02
Horizontal field (deg)	≤ 90	≤ 90
Vertical field (deg)	≤ 30	≤ 30
Scanning time/image (s)	≤ 15	≤ 1

tion", and have been one of the main research fields in computer vision.

The factorization method proposed in Ref. 34) is one of the most effective algorithms for simultaneously recovering the shape of the object and the motion of the camera by using the image sequence. The factorization was extended to several perspective approximations and applications [6),7),11),12),25),28)].

In Ref. 28), they presented perspective refinement by using the solution under the paraperspective factorization as the initial value. In Ref. 12) a factorization method with a perspective camera model was proposed. Using the weak-perspective projection model, they iteratively estimated the shape and the camera motion under the perspective model.

3.1 Weak-Perspective Factorization

Given a sequence of F images, in which we have tracked P interest points over all frames, each interest point p corresponds to a single point $\vec{S_p}$ on the object. In image coordinates, the trajectories of each interest point are denoted as $\{(u_{fp}, v_{fp}) | f = 1, \ldots, F, p = 1, \ldots, P,\ \ 2F \geq P\}$.

Using the horizontal coordinates u_{fp}, we can define an $F \times P$ matrix U. Each column of the matrix contains the horizontal coordinates of a single point in the frame order, while each row contains the horizontal coordinates for a single frame. Similarly, we can define an $F \times P$ matrix V from the vertical coordinates v_{fp}.

The combined matrix of $2F \times P$ becomes the measurement matrix as follow.

$$W = \begin{pmatrix} U \\ V \end{pmatrix}. \tag{1}$$

Each frame f is taken at camera position $\vec{T_f}$ in the world coordinates. The camera pose is described by the orthonormal unit vectors $\vec{i_f}$, $\vec{j_f}$ and $\vec{k_f}$. The vectors $\vec{i_f}$ and $\vec{j_f}$ correspond to the x and y axes of the camera coordinates, while the vector $\vec{k_f}$ corresponds to the z axis along the direction perpendicular to the image plane (**Fig. 5**).

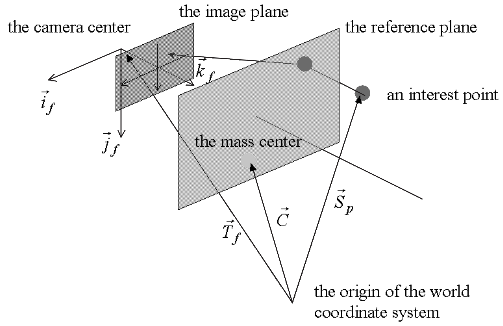


Fig. 5 The Coordinate System: $\vec{T_f}$ denotes the position of the camera at time of frame f. The camera pose is determined by three unit basis vectors.

Under the weak-perspective camera model, a single point in the world coordinates $\vec{S_p}$ is projected onto the image plane as (u_{fp}, v_{fp}).

$$u_{fp} = \frac{f_c}{z_f} \vec{i_f}^{\,t} \cdot (\vec{S_p} - \vec{T_f}), \tag{2}$$

$$v_{fp} = \frac{f_c}{z_f} \vec{j_f}^{\,t} \cdot (\vec{S_p} - \vec{T_f}), \tag{3}$$

$$\text{where } \ z_f = \vec{k_f}^{\,t} \cdot (\vec{C} - \vec{T_f}). \tag{4}$$

The vector $\vec{C}$ is the center of mass of all interesting points; f_c is the focal length. Without loss of generality, the origin of the world coordinates can be placed at the centroid, that is $\vec{C} = 0$. This means that $z_f = -\vec{k_f}^{\,t} \cdot \vec{T_f}$ to simplify the expansion of the following formulation. To summarize,

$$\begin{cases} u_{fp} = \vec{m_f}^{\,t} \cdot \vec{S_p} + \mathbf{x_f} \\ v_{fp} = \vec{n_f}^{\,t} \cdot \vec{S_p} + \mathbf{y_f}, \end{cases} \tag{5}$$

where

$$\vec{m_f} = \frac{f_c}{z_f} \vec{i_f}, \quad \mathbf{x_f} = -\frac{f_c}{z_f} \vec{i_f}^{\,t} \cdot \vec{T_f}^{\,t},$$

$$\vec{n_f} = \frac{f_c}{z_f} \vec{j_f}, \quad \mathbf{y_f} = -\frac{f_c}{z_f} \vec{j_f}^{\,t} \cdot \vec{T_f}^{\,t}.$$

Assuming that the center of all interest points is the origin,

$$\sum_{p=1}^{P} u_{fp} = \sum_{p=1}^{P} \vec{m_f}^{\,t} \cdot \vec{S_p} + \sum_{p=1}^{P} \mathbf{x_f} = P\mathbf{x_f}, \tag{6}$$

similarly,

$$\sum_{p=1}^{P} v_{fp} = P\mathbf{y_f}. \tag{7}$$

We obtain the registered measurement matrix $\tilde{W}$, after translation $\tilde{W} = W - (\mathbf{x_1}\ \mathbf{x_2}\ \ldots\ \mathbf{x_F}\ \mathbf{y_1}\ \ldots\ \mathbf{y_F})^t\ (1, \ldots 1)$ as a product of two matrices M and S,

$$\tilde{W} = M \cdot S, \tag{8}$$

where $M : 2F \times 3$ *matrix* $S : 3 \times P$ *matrix.*

The rows of the matrix M represent the orientation of the camera coordinates axes throughout the sequence, while the columns of the matrix S represent the coordinates of the interest points in the world coordinates. Both matrices are at most rank 3. Therefore, by using the Singular Value Decomposition (SVD), we can find the best approximation to $\tilde{W}$.

3.2 Extension to Full-Perspective Factorization

The above formulation is under the weak perspective projection model, which is a linear approximation of the perspective model. Next, using an iterative framework, we obtain approximate solutions under the non-linear, full perspective projection model.

Under the perspective projection model, the projective equations between the object point $\vec{S_p}$ in 3D world and the image coordinate (u_{fp}, v_{fp}) are written as

$$u_{fp} = f_c \frac{\vec{i_f}^t \cdot (\vec{S_p} - \vec{T_f})}{\vec{k_f}^t \cdot (\vec{S_p} - \vec{T_f})}, \tag{9}$$

$$v_{fp} = f_c \frac{\vec{j_f}^t \cdot (\vec{S_p} - \vec{T_f})}{\vec{k_f}^t \cdot (\vec{S_p} - \vec{T_f})}. \tag{10}$$

Replacing $z_f = -\vec{k_f}^t \cdot \vec{T_f}$, we obtain the following equations.

$$(\lambda_{fp} + 1) u_{fp} = \frac{f_c}{z_f} \vec{i_f}^t \cdot (\vec{S_p} - \vec{T_f}), \tag{11}$$

$$(\lambda_{fp} + 1) v_{fp} = \frac{f_c}{z_f} \vec{j_f}^t \cdot (\vec{S_p} - \vec{T_f}), \tag{12}$$

$$\lambda_{fp} = \frac{\vec{k_f}^t \cdot \vec{S_p}}{z_f}. \tag{13}$$

Note that the right hand sides of Eqs. (11) and (12) are the same form under the weak-perspective model (see Eqs. (2) and (3)). This means that multiplying an image coordinate (u_{fp}, v_{fp}) by a real number λ_{fp} maps the coordinate in the full perspective model space into the coordinate in the weak-perspective model space. Solving for the value of λ_{fp} iteratively, we can obtain motion parameters and coordinates of interest points under the full perspective projection model in the framework of weak-perspective factorization.

The entire algorithm of the perspective factorization is as follows:

Input: An image sequence of F frames tracking P interest points.

Output: The 3D positions of P interest points $\vec{S_p}$. The camera position $\vec{T_f}$ and poses $\vec{i_f}$, $\vec{j_f}$, $\vec{k_f}$ at each frame f.

(1) Given $\lambda_{fp} = 0$

(2) Supposing Eqs. (11) and (12), solve for $\vec{S_p}$, $\vec{T_f}$, $\vec{i_f}$, $\vec{j_f}$, $\vec{k_f}$ and z_f through the weak perspective factorization.

(3) Calculate λ_{fp} by Eq. (13).

(4) Substitute λ_{fp} into step (2) and repeat the above procedure.

Until: λ_{fp}'s are close to ones at the previous iteration.

3.3 Tracking

As input material, we need P interest points at each frame, which are tracked identified points in the 3D world. There are several methods to derive interest points of images [24),32)]. Among them, we adopt *Harris operator* [13)] and *SIFT key* [21)] for derivation of interest points. SIFT key is robust for scale, rotation and affine transformation changes. The main reason why we adopt the method is its stability of points derivation and usefulness of the key, which has 128 dimensional elements and can be used for the identification for each point.

4. Refinement

Without noise in the input data, the factorization method leads to an excellent solution. As a result, the rectified 3D shape through the estimated camera parameters is valid. Real images, however, contain a bit of noise. Therefore, it is not sufficient to rectify range data obtained by the FLRS only through the factorization. For the sake of a more refined estimation of motion parameters, we impose three constraints: for tracking, movement, and range data. The refined camera motion parameters can be found through the minimization of a global functional. To minimize the function, the solution by the full perspective factorization is utilized as the initial value to avoid local minimums.

4.1 Tracking Constraint

As the most fundamental constraint, any interest point $\vec{S_p}$ must be projected at the coordinates (u_{fp}, v_{fp}) on each image plane. This constraint is well known as Bundle Adjustment [4)]. When the structure, motion and shape have been roughly obtained, this technique is utilized to refine them through an image sequence. In our case, the constraint conducts the following function:

$$F_A = \sum_{f=1}^{F}\sum_{p=1}^{P}\left(\left(u_{fp} - f_c\frac{\vec{i_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})}{\vec{k_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})}\right)^2 + \left(v_{fp} - f_c\frac{\vec{j_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})}{\vec{k_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})}\right)^2\right). \quad (14)$$

The minimization of F_A leads to the correct tracking of fixed interest points by a moving camera. However, we can see that the presence of parameters we are trying to estimate in the denominator makes this equation a difficult one. We have to seek the optimal solution via some non-linear minimization techniques. Therefore, suppose that instead we consider the following function:

$$F'_A = \sum_{f=1}^{F}\sum_{p=1}^{P}\Big(\big(\vec{k_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})u_{fp} - f_c\cdot\vec{i_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})\big)^2 + \big(\vec{k_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})v_{fp} - f_c\cdot\vec{j_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})\big)^2\Big). \quad (15)$$

The term $\vec{k_f}^{\,t}\cdot(\vec{S_p}-\vec{T_f})$ means the depth, the distance between the optical center of camera f and a plane, which is parallel to the image plane and include the point $\vec{S_p}$. The cost function F_A is the summation of squared distances on the image plane while the cost function F'_A is estimated on the plane of the point $\vec{S_p}$. It is true that we can only observe the image points on the image sequence, therefore the noise occurs on these images. However it is also true that the cost function F_A does not assure that the reconstructed points are close to the correct ones in the real 3D world.

Based on the above consideration, we choose to minimize the cost function F'_A for the facility of the differential calculation.

4.2 Smoothness Constraint

One of the most significant reasons for adopting a balloon platform is to be free from the high frequency that occurs with a helicopter platform [15]. A balloon platform is only under the influence of low frequency: the balloon of our FLRS is held with some wires swayed only by wind. This means that the movement of the balloon is expected to be smooth. Certainly, the movement of the balloon is free from rapid acceleration, rapid deceleration, or acute course changing. Taking this fact into account, we consider the following function:

$$F_B = \int\left(w_1\left|\frac{\partial^2\vec{T_f}}{\partial t^2}\right|^2 + w_2\left|\frac{\partial^2\mathbf{q}_f}{\partial t^2}\right|^2\right)dt. \quad (16)$$

Here, $\vec{T_f}$ denotes the position of the camera; t is time; w_1, w_2 are weighted coefficients; and $\mathbf{q}_f$ is a unit quaternion that represents the camera pose. The first term of the above integrand represents smoothness with respect to the camera's translation while the second one represents smoothness with respect to the camera's rotation. When the motion of the camera is smooth, the function F_B takes a small value.

We implement in practice the following discrete form:

$$F'_B = \sum_{f=2}^{F-1}\left(w_1\left|\vec{T}_{f-1} - 2\vec{T_f} + \vec{T}_{f+1}\right|^2 + w_2\left|\mathbf{q}_{f-1} - 2\mathbf{q}_f + \mathbf{q}_{f+1}\right|^2\right). \quad (17)$$

4.3 Range Data Constraint

Taking a broad view of range data obtained by the FLRS, the data are distorted by the swing of the sensor. We can find, however, that these data contain instantaneous precise information locally. That information is utilized for refinement of the camera motion.

Our FLRS re-radiates laser beams in raster scan order. This means that we can instantly obtain the time when each pixel in the range image is scanned because the camera and the range sensor are calibrated (**Fig. 6**). If the video camera is synchronized with the range sensor, we can find the frame among the sequence when the pixel is scanned. With the video camera calibrated with the range sensor, we can also obtain the image coordinate of each

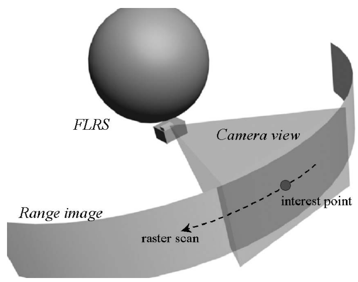


Fig. 6 Finding the time when each interest point in the sequence is scanned by the range sensor.

interest point in the 3D world with respect to the instantaneous local coordinate.

Considering this constraint, we can compensate the camera motion.

At time t, suppose that the sensor position is $\vec{T}(t)$ and the three bases $\vec{i_f}$, $\vec{j_f}$, $\vec{k_f}$ are described as $\vec{i}(t)$, $\vec{j}(t)$, $\vec{k}(t)$. At this moment, suppose that the range sensor output $\vec{x}(t)$ (in the local coordinate) as the measurement of the point $\vec{X}$, which is described in the world coordinate, the following equation is obtained.

$$\vec{X} = \left(\vec{i}\ \vec{j}\ \vec{k}\right)\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{T} = R\vec{x} + \vec{T}. \tag{18}$$

When the range sensor scans interest point $\vec{S_p}$, we can derive the third constraint to be minimized as follows:

$$F_C = \sum_{p=1}^{P}\left|\mathbf{x}_{fp} - R^t(\vec{S_p} - \vec{T_{fp}})\right|^2. \tag{19}$$

Here, the index fp denotes the frame number when the range sensor scans interest point $\vec{S_p}$. It is very significant to note that $\mathbf{x}_{fp}$ is the 3D coordinate values not described in the sensor-oriented coordinate system but in the camera-oriented one, which is rewritten based on the range data and camera-sensor calibration. In practice, we find sub-frame fp by using a linear interpolating technique for the motion of interest points between frames. The main purpose of the above constraint is to adjust the absolute scale.

As $\mathbf{x}_{fp} = (x_{fp}, y_{fp}, z_{fp})$, the above function can be rewritten as the stronger constraint:

$$\begin{aligned} F_C' = \sum_{p=1}^{P}\Big(&\big(x_{fp} - \vec{i_{fp}}^{\,t}\cdot(\vec{S_p} - \vec{T_{fp}})\big)^2 \\ &+ \big(y_{fp} - \vec{j_{fp}}^{\,t}\cdot(\vec{S_p} - \vec{T_{fp}})\big)^2 \\ &+ \big(z_{fp} - \vec{k_{fp}}^{\,t}\cdot(\vec{S_p} - \vec{T_{fp}})\big)^2\Big). \end{aligned} \tag{20}$$

4.4 The Global Cost Function

Based on the above considerations, it will be found that the next cost function should be minimized. Consequently, the weighted sum,

$$F = w_A F_A' + w_B F_B' + w_C F_C', \tag{21}$$

leads to the global cost function. The coefficients w_A, w_B and w_C are determined experimentally so that three terms take almost the same magnitude. We set three coefficients as $w_A : w_B : w_C = 1 : 10^7 : 10^6$ in simulation cases (Section 6) and $w_A : w_B : w_C = 1 : 10^5 : 10^2$ in real cases (Section 7).

To minimize this function, we employ Fletcher–Reeves method or Polak–Ribiere method [18),27),31)], which are types of the conjugate gradient method. Then, we use the golden section search to determine the magnitude of gradient directions. For optimization, Levenberg-Marquardt method [20)] is generally employed to minimize a functional value. The Levenberg-Marquardt method is very effective in estimating function's parameters, especially to fit a certain function. However in our function, it is not a parameter fitting problem to minimize the value of F_B'. All we have to do is to decrease F_B' simply. Therefore we adopt the conjugate gradient method.

4.5 Shape Rectification

After the refinement, we possess the vector $\vec{T_f}$ and three bases $\vec{i_f}$, $\vec{j_f}$ and $\vec{k_f}$ at each frame. That means we know the position and pose of the camera at all frames. To rectify the deformed shape data by using these extrinsic parameters quantized with respect to time, these parameters have to be interpolated. To be more precise, we have to interpolate three components with respect to translation $\vec{T_f} = (T_{xf}, T_{yf}, T_{zf})$, and three components with respect to rotation $\mathbf{q}_f = ((s_f,)\, u_f, v_f, w_f)$. Each parameter's variation with respect to time is, therefore, approximated by a polynomials[☆].

5. Shape Rectification without Images

The method mentioned so far does not need another range data set; the distorted range data are rectified by using only a single range image and an image sequence.

In actual cases, however, there should be some available range data sets taken by another range sensor fixed on the ground. Our FLRS is originally devised to complement the measurement for the region that is invisible from the ground.

Some parts of a range image taken by the FLRS are also taken by another range sensor fixed on the ground. Based on these overlapping regions, we propose another algorithm which rectifies the distorted range data obtained from the FLRS. In this method, we do not use any image sequences.

☆ In this study, we adopt 7-order polynomials.

5.1 Basic Idea

Original ICP (Iterative Closest Point) algorithm[2),5),38)] was developed to align two shapes. In a range image, coordinates of 3D points are described in the sensor-oriented coordinate system. Two range images from different viewpoints, therefore, have different coordinate systems. To unify two shapes, two data sets have to be described in the unified coordinate system. In order to do that, we apply a coordinate conversion to one data set. When there are some overlapping regions in the two data sets, we apply a transformation of the coordinate system in order to coincide them.

To simplify the transform procedure, we assume that one shape is fixed and another can move. We call the fixed shape the *model shape* and the movable one the *data shape*. Rotating and translating the data shape aligns two shapes. In overlapping region, a point on the model shape has a corresponding point on the data shape. Which point is the corresponding point, however, is usually unknown. This correspondence problem is resolved by an iterative method. Initially a temporal corresponding point is assumed. A movement is determined so as to minimize an objective function, which is defined by the total distances between the corresponding points. The temporal correspondences are changed after the movement. Then a new movement is determined under the new temporal correspondence. This procedure is repeated until the total distance converges. The objective function, which should be minimized for the alignment, is defined as

$$f\left(R, \vec{T}\right) = \sum_i \left| R\vec{x_i} + \vec{T} - \vec{y_i} \right|^2 . \quad (22)$$

This objective function indicates the summation of distances between all pairs of corresponding points. If two shapes coincide, the function takes a low value.

There are many variations of ICP algorithms[30)]. For example, while we estimate the cost function as the total distances of point-to-point pairwise[2),38)], some methods adopt the distance between a point and its mate's tangent plane[5),26)].

There are several methods to determine corresponding points. Some methods search the corresponding point along the viewing ray[3)]. In this article, we adopt the nearest neighbor points as the corresponding points. We speed up searches for the nearest neighbor point by using KD-tree[9)].

We use a quaternion as rotational elements of the objective function f. By substituting quaternion $\mathbf{q}$ to rotate matrix R, motion vector $\vec{T}$ can be found as follows:

$$\{\mathbf{q}, \vec{T}\} = \min_{\mathbf{q},\vec{T}} f\left(\mathbf{q}, \vec{T}\right) = \sum_i \left| R(\mathbf{q})\vec{x_i} + \vec{T} - \vec{y_i} \right|^2 . \quad (23)$$

In the conventional ICP algorithm mentioned above, it is assumed that both shapes are obtained by fixed range sensors. On the other hand, in our situation, the model shape is obtained by a fixed range sensor while the data shape is measured by a moving sensor. Therefore we have to take account into the motion of the range sensor.

The motion of the sensor is expected to be smooth, as mentioned in the previous section. It is, therefore, proper that the traces of the motion parameters are approximated by some polynomials with respect to time. Consequently, we approximate six parameter: three translational elements and three elements of the quaternion, by following polynomials:

$$\vec{T}(t) = \vec{T_0} + t\vec{T_1} + t^2\vec{T_2} + \cdots = \sum_{n=0}^{N} t^n \vec{T_n}, \quad (24)$$

$$\mathbf{q}(t) = \mathbf{q_0} + t\mathbf{q_1} + t^2\mathbf{q_2} + \cdots = \sum_{n=0}^{N} t^n \mathbf{q_n}, \quad (25)$$

where $\{\vec{T_0}, \vec{T_1}, \cdots, \vec{T_N}, \mathbf{q_0}, \mathbf{q_1}, \cdots, \mathbf{q_N}\}$ are the parameters that describe the sensor motion. Then we formulate a new cost function including the above forms.

5.2 Extended ICP Algorithm

Instead of Eq. (23), we have to set up a new cost function. First, we will change the index of points of the data shape, $\vec{x_i}$. Our sensors measure the distance to a point in the raster scan order. Therefore, all points on the data shape, which are measured by the moving sensor, are distinguishable by time t. According to the time factor, the corresponding points on the model shape $\vec{y_i}$, which are obtained by the fixed sensor, are described as functions $\vec{y}\left(\vec{x}(t), t\right)$.

The cost function for the extended ICP algorithm is described as follows:

$$f\left(\vec{T_0}, \vec{T_1}, \cdots, \vec{T_N}, \mathbf{q_0}, \mathbf{q_1}, \cdots, \mathbf{q_N}\right)$$

$$= \sum_t \left| R\left(\mathbf{q}(t)\right)\vec{x}(t) + \vec{T}(t) - \vec{y}(\vec{x}(t), t) \right|^2 . \tag{26}$$

We take a summation form with respect to time t in spite of the continuity of time. Since it is only necessary to pick up the moments when the point on the data shape is actually scanned.

To minimize the above function, the parameters of the sensor's motions should be estimated.

$$\{\vec{T_0}, \vec{T_1}, \cdots, \vec{T_N}, \mathbf{q_0}, \mathbf{q_1}, \cdots, \mathbf{q_N}\} = \min f\left(\vec{T_0}, \vec{T_1}, \cdots, \vec{T_N}, \mathbf{q_0}, \mathbf{q_1}, \cdots, \mathbf{q_N}\right). \tag{27}$$

If we assume N-order polynomials, the number of unknown valuables is $6(N+1)$. We minimize the cost function through the steepest descent method and Golden section search. Furthermore we adopt a robut estimation, M-Estimator [10),29),36)] to decrease the influence of outliers; the cost function Eq. (26) is rewritten as follows:

$$\{\vec{T_0}, \vec{T_1}, \cdots, \vec{T_N}, \mathbf{q_0}, \mathbf{q_1}, \cdots, \mathbf{q_N}\} = \min \sum_t \log\left(1 + \frac{1}{2\sigma^2}|\mathbf{z_t}|^2\right), \tag{28}$$

where $\mathbf{z_t} = R(\mathbf{q}(t))\vec{x}(t) + \vec{T}(t) - \vec{y}(\vec{x}(t), t)$.

6. Evaluation

6.1 Benchmark Shapes

To evaluate our rectification algorithms quantitatively, the most efficient method is to check them for given models in advance.

In order to do that, we construct a virtual FLRS system on a PC and obtain the distorted range data and the image sequences for the known model. Motion parameters are also known completely. Then, we rectify the distorted range data through our two proposed methods.

The rectified shape data are, eventually, compared with the correct shape data, and the results are evaluated numerically.

We use CAD models as a benchmark for the evaluation (**Fig. 7**). The benchmark has great depth, which has a strong perspective effect. For reference, the height of the pyramid is 0.6, that of the side wall is 0.78 and the thickness of the side wall is 0.2. The equation of the back plane is $z = 0$ and that of the floor is $y = 0$.

Then, we map textured pictures onto the surfaces of the benchmark shapes to detect many interest points for tracking.

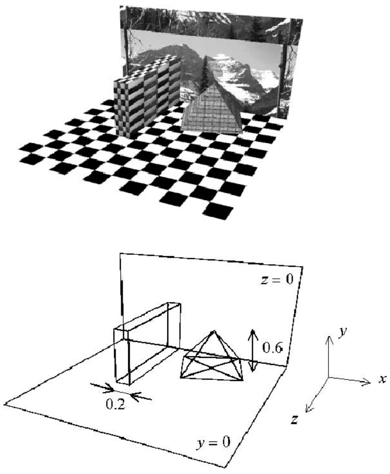


Fig. 7 The benchmark shape for the evaluation.

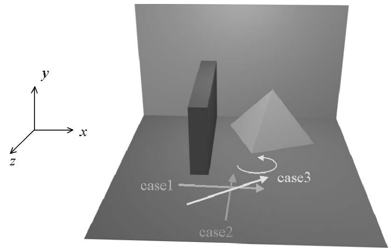


Fig. 8 The sensor paths for the evaluation.

After that, we provide three sensor motions for virtual measurements (**Fig. 8**).

- **Case1:** Pure translation along the x direction (parallel to the image plane).
- **Case2:** Pure translation along the $-z$ direction (perpendicular to the image plane).
- **Case3:** Translation (within the $x - z$ plane) and rotation (around the y axis).

6.2 Evaluation of the Algorithm with Images

Case 1:

In this case, the FLRS simply moves during the measurement process toward the horizontal direction with respect to the camera-oriented coordinate system. The motion path is parallel to the image plane and the back plane of the benchmark model.

Several example images of the sequence are shown in **Fig. 9**. These images look like pictures obtained by simple parallel stereo vision since there are no rotational elements in Case 1.

The distorted shape that is obtained by the

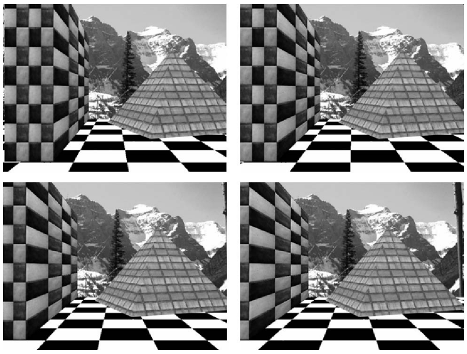


Fig. 9 Some sample images of the sequence Case 1. (top left → top right → bottom left → bottom right).

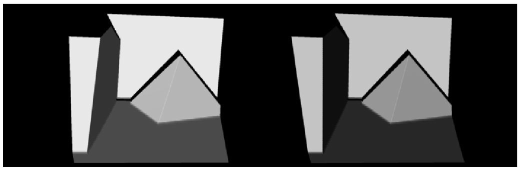


Fig. 10 The original and rectified model of Case 1.

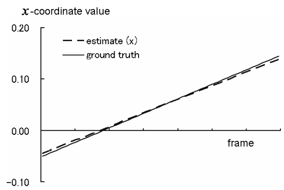


Fig. 11 The time transient parameter (x in translation) and the ground truth in Case 1.

virtual FLRS is shown on the left of **Fig. 10**. Especially, it is found that the top region of the side wall is skewed to the right side. On the other hand, in the right shape, which is the rectified shape by our algorithm, the side wall stands perpendicular to the ground. For the time being, the shape seems to be rectified properly by our method. The numerical evaluation for the rectified shape is show at the end of this subsection.

Figure 11 indicates the estimated x position and the ground truth. In Case 1, we set a uniform straightly-line motion and the result shows it. The difference between the estimated velocity and the ground truth is only 6.4%.

All parameters, three components of translation and three components of camera pose, through the scanning period are shown in **Fig. 12**. The left side of Fig. 12 shows that the FLRS moved only along the x direction, which corresponds to the ground truth. In addition,

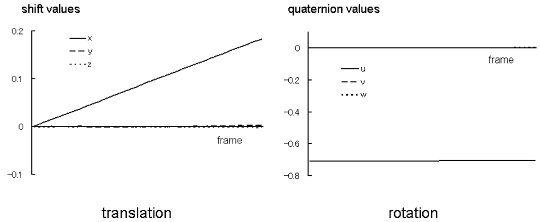


Fig. 12 The all motion parameters in Case 1.

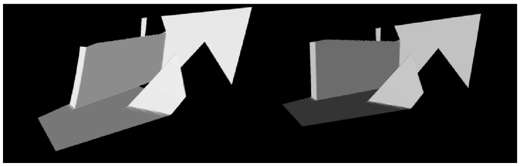


Fig. 13 The original and rectified model of Case 2.

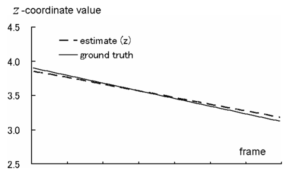


Fig. 14 The time transient parameter (z in translation) and the ground truth in Case 2.

the right side Fig. 12 shows that the motion did not have any rotational component, which also corresponds to the ground truth.

Case 2:

In this case, the FLRS moves along the optical axis, which is perpendicular to the image plane.

The distorted shape, which is obtained by the virtual FLRS, is shown on the left side of **Fig. 13**. When the virtual FLRS scans the top region of the scene it is located far from the scene. Then the closer the FLRS moves, the lower region it scans. Therefore, the obtained shape seems as though it is skewed backward. As with Case 1, the right side of the figure shows the rectified shape, which looks like the proper shape.

Figure 14 indicates the estimated z position and the ground truth. The difference between the estimated velocity and the ground truth is 13.4%. While the estimated error is larger than that of Case 1, the motion of Case 2 is wider than that of Case 1. The virtual FLRS's speed in Case 2 corresponds to about $3.0\,m/s$ in terms of the real FLRS scale. It is thought that the our algorithm can rectified the distorted shape in spite of the wide motion.

All motion parameters are shown in **Fig. 15**.

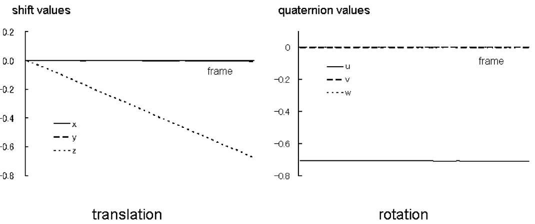


Fig. 15 The all motion parameters in Case 2.

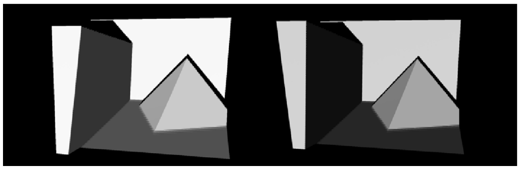


Fig. 16 The original and rectified model of Case 3.

The left side of the figure, which shows the translational components, shows that the FLRS moved only along the z direction. And the right side of the figure shows that the FLRS was keeping the same pose during the scanning process. These figures indicate that the parameters are estimated properly.

Case 3:

In this case, the virtual FLRS motion has two translational components, x and z. In addition, the FLRS rotates around the y axis during the scanning process.

The distorted shape obtained by the virtual FLRS is shown in the left side of **Fig. 16**. As in Case 1, it is found that the top region of the side wall is skewed to the right side. The right side of the figure shows the rectified shape, which looks like proper shape.

Figure 17 indicates the estimated parameters and the ground truths. In Fig. 17, three parameters, x position (a), z position (b) and rotational component around y axis (c) are shown. The difference between the estimated velocity and the ground truth is 13.8% with respect to x and 15.0% with respect to z. The difference with respect to the rotational angle is within 5.6%.

All motion parameters are shown in **Fig. 18**. These figures show that our algorithm works well on a case with several motion components.

Finally, **Table 3** shows the errors in all cases. These values are mean errors by point-to-patch distance. The errors in the "Before Rectification" row are the mean errors between the distorted shapes and the ground truth, which are aligned by ICP algorithm [2),5)]. On the other hand, the values in the "After Rectification" row are the mean errors between the rectified shapes and the ground truth. It is found that our method could decrease the errors in all cases. In the case of the real 25 m FLRS, the maximum distance for scan is at most 25 meters, while the distance to the backplane in the benchmark shapes is about 3.5☆ in the CAD model scale. Therefore, multiplying the values of Table 3 by at most 7 gives the estimated errors in the practical measurement scale. In most data sets in the Bayon Temple project, we measure objects at a distance of 15 ∼ 18 meters. For example, the estimated accuracy in Case 2 will be about 3 cm in practice.

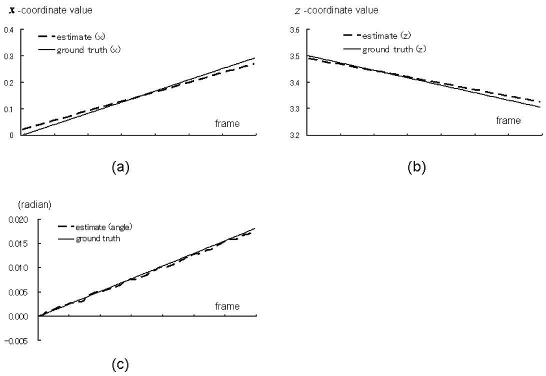


Fig. 17 The time transient parameters and the ground truths in Case 3. (a) x and (b) z in translation; (c) Rotational angle (radian) around y axis.

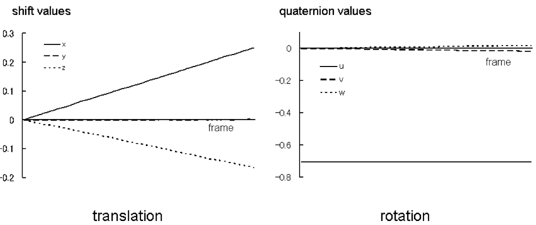


Fig. 18 The all motion parameters in Case 3.

Table 3 The mean errors of the method with images.

	Case1	Case2	Case3
Before Rectification	0.01342	0.06632	0.03103
After Rectification	0.004990	0.006379	0.004268

6.3 Evaluation of the Algorithm without Images

Next, we evaluate the method mentioned in Section 5, which uses correct shapes obtained by other fixed laser sensors without any image sequences. In this section, the data sets are

☆ In the CAD model, there is no unit with respect to length.

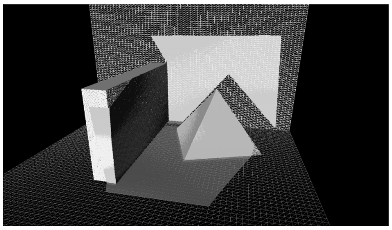


Fig. 19 The ground truth and rectified model of Case 1.

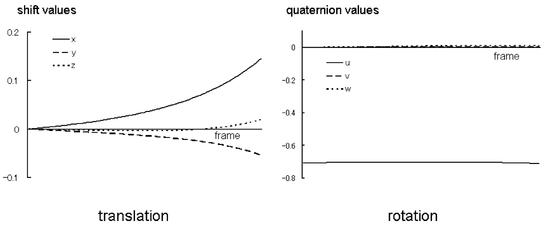


Fig. 20 The all motion parameters in Case 1.

the same as in the previous sub section. Besides these, Case 4 is added, in which the motion of the sensor contains only rotation without any translational components. In fact, a motion without translational components is a critical motion for the method described in Section 3 and 4 since no disparities could be detected in images. In order to rectify distorted range datasets by using image sequences, different methods should be applied.

Case 1:

In Case1, the sensor simply moves toward the horizontal direction.

Figure 19 shows the rectified model and the ground truth.

The following figure, **Fig. 20**, shows all motion parameters. All translational parameters change in time although the ground truth setting moves the sensor only along the x axis. In addition, the estimated velocity is not constant. Comparing it to Fig. 12, it is found that the graphs, especially on the left side of the figure, differ from those using the method with images. In spite of these graphs, we can safely state that our method is effective. This method places more emphasis on the minimization of the geometrical error and less on the proper estimation of sensor's motion. For example, when the FLRS scans a simple plane, many patterns of motion can be proper. Therefore, we consider that our method could rectify the deformed shape properly.

The table of errors in all cases is also shown at the end of this sub section.

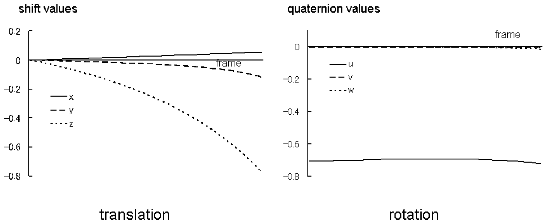


Fig. 21 The all motion parameters in Case 2.

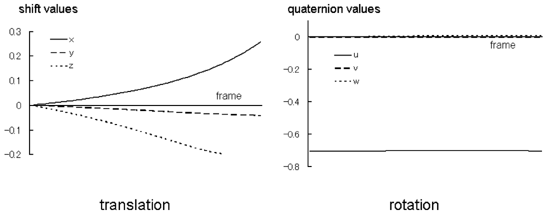


Fig. 22 The all motion parameters in Case 3.

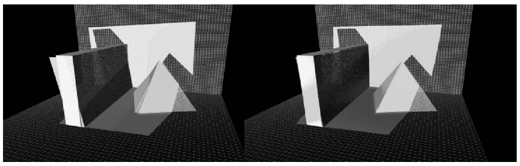


Fig. 23 The ground truth and rectified model of Case 4.

Case 2:

In this case, the sensor moves along the optical axis at a fast speed.

Figure 21 shows the all motion parameters. Under the ground truth configuration, only the x translational parameter is supposed to change. In Fig. 21, it is easily found that almost all parameters fluctuate.

Case 3:

In this case, the sensor moves within a plane parallel to $y = 0$ and rotates around the y axis.

Figure 22 shows the all motion parameters. Comparing it to Fig. 18, the graphs in Fig. 22 have similar properties. The translational graphs are, however, curved and the y component, which is supposed to be fixed, is moving.

Additional Case (Case 4):

In this case, while the position of the sensor does not change, it rotates around the y axis. As previously noted, the method with images can not rectify the distorted model because it is impossible to reconstruct the 3D model from images without disparity.

The left side of the figure in **Fig. 23** is a comparison between the ground truth and the original distorted model while the right side of the figure is a comparison between the ground truth and the rectified model. It is found that the method without images can properly rectify distorted models that are obtained from a sensor only with rotation. Thus, this is the strong advantage for this method.

Figure 24 indicates the estimated rotational angle and the ground truth. The difference between the estimated angular speed and the ground truth is 15.4%.

Figure 25 shows the all parameters. It is found that the estimated position is moving, especially with respect to the x component, although all parameters are not supposed to change.

Table 4 shows the errors by the method without images in all cases. These values are also mean errors by point-to-patch distance. Overall, the method with images is superior to the method without images in accuracy. This table shows the worst result is obtained in Case 2, which has a rapid sensor motion, and the accuracy in the practical case is about 10 cm. On the other hand, the accuracy of other test case results, especially in Case 1 and 4, are the same in number as those when the method with images is used. This means that the method without images is effective in the case of the sensor motion only with rotation.

We have used the complete model as the ground truth in this section. On the other hand,

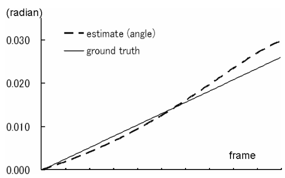


Fig. 24 The time transient parameter (rotational angle) and the ground truth in Case 4.

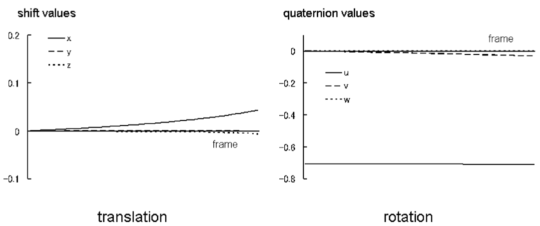


Fig. 25 The all motion parameters in Case 4.

Table 4 The mean errors of the method without images.

	Case1	Case2	Case3	Case4
Before	0.01342	0.06632	0.03103	0.04583
After	0.005561	0.01428	0.008894	0.005084

in practical cases, it is expected that a correct shape will have many missing parts and that we have to rectify the distorted shape based on an incomplete reference. We are going to demonstrate such cases in the following section by using real data sets.

7. Experiments

We have been conducting the "Digital Bayon Project", in which the geometric and photometric information related to the Bayon Temple is preserved in digital form. With respect to the acquisition of the geometric data, large parts of the temple visible from the ground are scanned by range sensors placed on the ground. On the other hand, some parts invisible from the ground, for example, roofs and tops of towers, are scanned by our FLRS system.

7.1 Shape Rectification with Images

Figure 26 shows a sample image of the sequence obtained by the FLRS.

Figure 27 shows a photo picture of the scanned area. On the right side of Fig. 27, the dense fine model is the correct shape obtained by a range sensor, the Cyrax-2500 [19] fixed on the ground.

The result of the rectifications is shown in **Fig. 28**. The upper shape in Fig. 28 is the original one obtained from the FLRS. It is found that the shape is widely deformed. In the middle of Fig. 28, the rectified shape by full-perspective factorization is shown. With respect to motion parameters, the ambiguity in scale is removed manually. At a glance, the factorization seems to rectify the shape properly.

Fig. 26 A sample shot of the image sequence.

Fig. 27 A scene for this experiment.

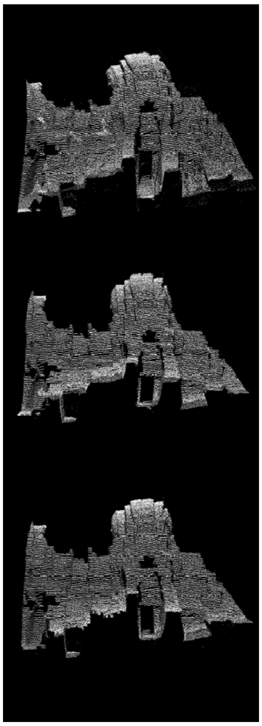


Fig. 28 The upper figure shows the original distorted shape obtained by the FLRS. The middle one shows the rectified shape by the full-perspective factorization without ambiguity in scale. The lower shows the rectified shape by our method.

In detail, however, the distortion in S shape still remains. Especially, the shape of the entrance is skewed. On the other hand, the lower shape is rectified correctly by our method. It is clear that the distortion in S shape is removed and the shape of the entrance is correctly recovered into a rectangle.

To evaluate the accuracy of our shape rectification algorithm, we compare the rectified shape with other data, which are obtained by the Cyrax-2500, positioned on the ground. Aligning two data sets by using the conventional ICP algorithm [2),5)], we analyze the overlapping area.

Figure 29 indicates the point-to-point distances in the ICP algorithm. The region where the distances between them are less than 6.0 cm is colored light gray ☆. The area where the distances are further than 6.0 cm is displayed in dark gray. The upper figure shows the comparison between the correct shape and the original distorted shape obtained by the FLRS. The middle figure shows the rectified shape by the full-perspective factorization without ambiguity in scale. The lower figure shows the rectified shape by our method.

☆ In the previous section, we have approximated the accuracy in the practical case as 3.0 cm. Therefore, we set the threshold as 6.0 cm, twice of the estimated error.

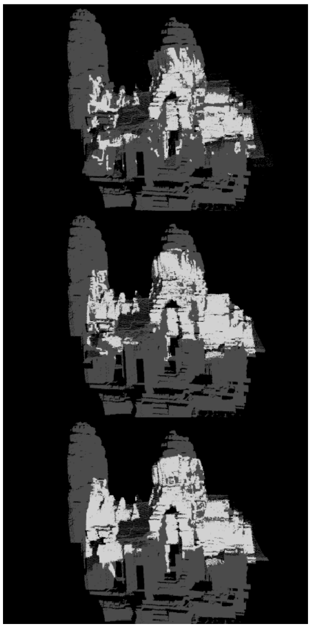


Fig. 29 The upper figure shows the comparison between the correct shape and the original distorted one obtained by the FLRS. The light gray region indicates where the distance of two shapes is less than 6.0 cm. The middle one shows the rectified shape by the full-perspective factorization without ambiguity in scale. The lower shows the rectified shape by our method.

At a glance, the light gray region is clearly expanded by our rectification algorithm. Some parts of the rectified shape are colored dark gray because of the lack of corresponding points. Taking account of the fact the correct shape could not measure the parts invisible from the ground, the proposed method could rectify the 3D shape correctly.

Figure 30 shows several samples of the method with images.

7.2 Shape Rectification without Images

We also applied the method without images to the real data set. As the reference shape, we also utilize the shape obtained by the Cyrax-2500. There are some blank parts in the reference shape because there are no data set on the part that is invisible from the ground. **Figure 31** shows the sample snap in this experi-

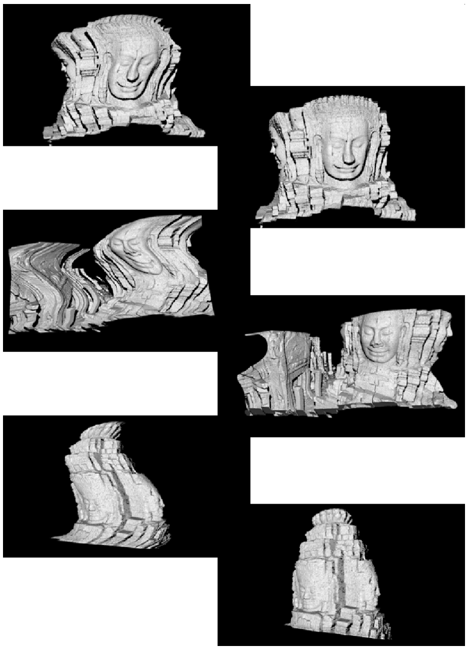


Fig. 30 The original distorted data sets (left) and the rectified sets (right).

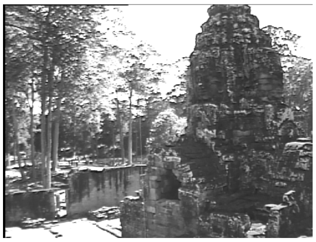


Fig. 31 A sample shot in this case.

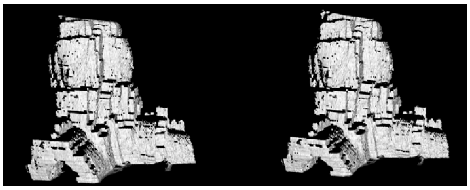


Fig. 32 The original distorted shape (left) and the rectified shape (right).

mental case.

In **Fig. 32**, the left figure shows the original shape obtained by the FLRS while the right one shows the rectified shape by our method.

The upper figure of **Fig. 33** shows the original distorted shape by the FLRS (sparse model) and the reference shape (dense model). The lower figure of Fig. 33 shows the rectified shape and the reference one. It is found that the rectified 3D shape is well-fitted onto the reference one, particularly the area of ellipses in the upper figure, in spite of the blanks on the reference shape.

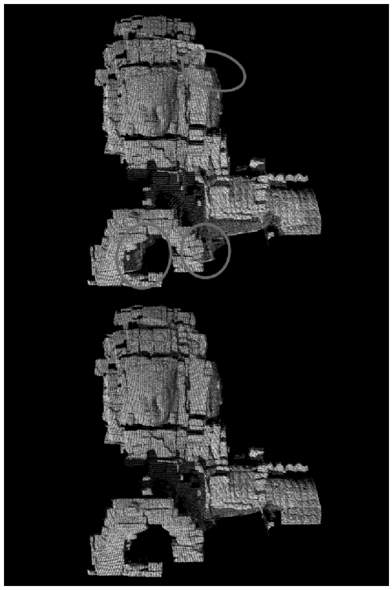


Fig. 33 Range data before and after the rectification method without images: the upper figure shows the original distorted shape by the FLRS (sparse) and the reference shape obtained by the Cyrax-2500 fixed on the ground (dense). The lower figure shows the recovered shape fitted onto the correct one.

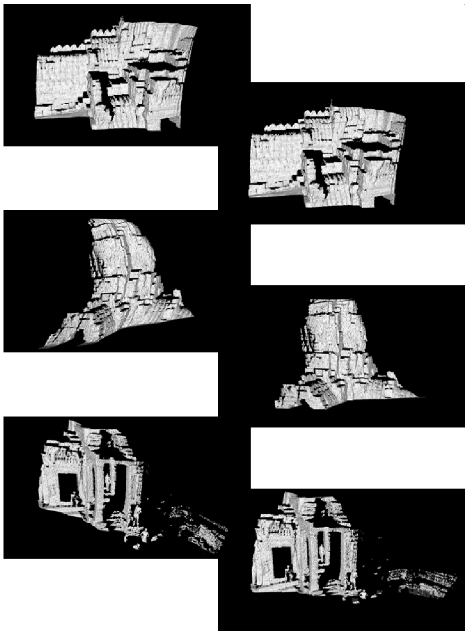


Fig. 34 The original distorted data sets (left) and the rectified sets (right).

Finally, **Fig. 34** shows several results of the method without images.

8. Conclusions

In this article, we have described FLRS system and two proposed methods to rectify 3D range data obtained by a moving laser range sensor.

We described how an outstanding measurement system FLRS was built to scan large objects from the air. This system allowed us to measure large cultural heritage objects by using a balloon. To rectify the distorted shapes obtained from the FLRS, we proposed two methods

- The rectification method based on the "Structure from Motion" techniques by using image sequences
- The rectification method based on the extended ICP algorithm by using another range data sets

In the first method, we described a method based on "Structure from Motion". We utilized distorted range data obtained by a moving range sensor and image sequences obtained by a video camera mounted on the FLRS. First, the motion of the FLRS was estimated through full perspective factorization only by the obtained image sequences. Then the more refined parameters were estimated based on an optimization imposing three constraints: the tracking, smoothness and range data constraints. Finally, by using refined camera motion parameters, the distorted range data are rectified.

In the second method, we proposed an extended ICP algorithm without using any images. Assuming that the motions of the sensor are smooth, we applied them to polynomials. Then, we rectified the distorted range data based on the correct model obtained by other range sensors fixed on the ground.

The results by both methods have shown proper performance and practical utilities. These methods can be generally applied to a framework in which a range sensor moves during the scanning process, and is not limited to our FLRS.

There are a lot of works to do in the future. First, we have to improve the accuracy of rectified shapes by our algorithm. The burning issue is the improvement of the accuracy of the method without images. We want to boost it to the same level as that of using the method with images.

Besides accuracy, there are a few challenging problems in the rectification algorithm without images. Currently, we use a single distorted shape and a single correct shape. As the next step, we are trying to rectify several distorted shapes at the same time by using a single correct shape. Moreover, we plan to rectify and register multi-distorted shapes simultaneously without any correct shapes. We envision a rectification method that utilizes both images and the correct models.

Acknowledgments This work was supported in part by Ministry of Education, Culture, Sports, Science and Technology, under the program, "Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets" and Research Fellowships of Japan Society for the Promotion of Science (JSPS).

References

1) Banno, A. and Ikeuchi, K.: Shape Recovery of 3D Data Obtained from a Moving Range Sensor by using Image Sequences, *Proc. International Conference on Computer Vision* (*ICCV2005*), Vol.1, pp.792–799 (2005).

2) Besl, P.J. and McKay, N.D.: A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14, pp.239–256 (1992).

3) Blais, G. and Levine, M.D.: Registering multiview range data to crate 3D computer objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.8, pp.820–824 (1995).

4) Brown, D.: The bundle adjustment — progress and prospect, *XIII Congress of the ISPRS*, Helsinki (1976).

5) Chen, Y. and Medioni, G.: Object modeling by registration of multiple range images, *Image and Vision Computing*, Vol.10, No.3, pp.145–155 (1992).

6) Christy, S. and Horaud, R.: Euclidean shape and motion from multiple perspective views by affine iterations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.18, No.11, pp.1098–1104 (1996).

7) Costeira, J. and Kanade, T.: A multi-body factorization method for motion analysis, *Proc. International Conference on Computer Vision* (*ICCV1995*), pp.1071–1076 (1995).

8) The Digital Michelangelo Project: http://graphics.stanford.edu/projects/mich/.

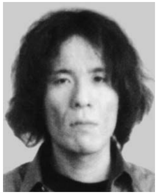
9) Friedman, J.H., Bentley, J.L. and Finkel, R.A.: An algorithm for finding best-matches in logarithmic time, *ACM Transactions on Mathematical Software*, Vol.3, No.3, pp.209–226 (1977).

10) Gill, P., Murray, W. and Wright, M.: *Practical Optimization*, Academic Press, London (1981).
11) Gruber, A. and Weiss, Y.: Multibody factorization with uncertainty and missing data using the EM algorithm, *Proc. Conference on Computer Vision and Pattern Recognition* (*CVPR2004*), Vol.1, pp.707–714 (2004).
12) Han, M. and Kanade, T.: Perspective factorization methods for euclidean reconstruction, *Technical Report: CMU–RI–TR–99–22*, Robotics Institute, Carnegie Mellon University (1999).
13) Harris, C. and Stephens, M.: A combined corner and edge detector, *Proc. Alvey Vision Conference*, pp.147–152 (1988).
14) Hasegawa, K., Hirota, Y., Ogawara, K., Kurazume, R. and Ikeuchi, K.: Laser range sensor suspended beneath balloon: FLRS (flying laser range sensor), *The IEICE Transactions on Information and Systems, PT.2* (*Japanese Edition*), Vol.J88-D-II, No.8, pp.1499–1507 (2005). (in Japanese)
15) Hirota, Y., Masuda, T., Kurazume, R., Ogawara, K., Hasegawa, K. and Ikeuchi, K.: Designing a laser range finder which is suspended beneath a balloon, *Proc. 6th Asian conference on Computer Vision* (*ACCV2004*), Vol.2, pp.658–663 (2004).
16) Ikeuchi, K., Nakazawa A., Hasegawa, K. and Ohishi, T.: The Great Buddha Project: Modeling Cultural Heritage for VR Systems through Observation, *Proc. 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality* (*ISMAR2003*) (2003).
17) Ikeuchi, K., Hasegawa, K., Nakazawa, A., Takamatsu, J., Oishi, T. and Masuda, T.: Bayon digital archival project, *Proc. International Conference on Visual Systems and Multimedia* (*VSMM2004*), pp.334–343 (2004).
18) Jacobs, D.A.: *The State of the Art in Numerical Analysis*, Academic Press, London (1977).
19) Leica Geosystems.: http://www.leica-geosystems.com/.
20) Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, Vol.11, pp.431–441 (1963).
21) Lowe, D.G.: Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol.60, No.2, pp.91–110 (2004).
22) Masuda, T., Hirota, Y., Nishino, K. and Ikeuchi, K.: Simultaneous determination of registration and deformation parameters among 3D range images, *Proc. 5th International Conference on 3-D Digital Imaging and Modeling* (*3DIM2005*), pp.369–376 (2005).
23) Miyazaki, D., Oishi, T., Nishikawa, T., Sagawa, R., Nishino, K., Tomomatsu, T., Yakase, Y. and Ikeuchi, K.: The great buddha project: Modelling cultural heritage through observation, *Proc. 6th International Conference on Virtual Systems and Multimedia* (*VSMM2000*), pp.138–145 (2000).
24) Moravec, H.P.: Towards automatic visual obstacle avoidance, *Proc. 5th International Joint Conference on Artificial Intelligence*, p.584 (1977).
25) Morita, T. and Kanade, T.: A sequential factorization method for recovering shape and motion from image streams, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.8, pp.858–867 (1997).
26) Neugebauer, P.: Geometrical cloning of 3D objects via simultaneous registration of multiple range images, *Proc. International Conference on Shape Modeling and Application*, pp.130–139 (1997).
27) Polak, E.: *Computational Methods in Optimization*, Academic Press, New York (1971).
28) Poelmann, C. and Kanade, T.: A paraperspective factorization method for shape and motion recovery, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.3, pp.206–218 (1997).
29) Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T.: *Numerical Recipes in C*, Cambridge University Press (1988).
30) Rusinkiewicz, S. and Levoy, M.: Efficient variant of the ICP algorithm, *Proc. 3rd International Conference on 3-D Digital Imaging and Modeling* (*3DIM2001*), pp.145–152 (2001).
31) Stoer, J. and Bulirsh, R.: *Introduction to Numerical Analysis*, Springer-Verlag, New York (1980).
32) Smith, S.M and Brady, M.: SUSAN — A new approach to low level image processing, *International Journal of Computer Vision*, Vol.23, No.1, pp.45–78 (1997).
33) Thrun, S., Diel, M. and Haehnel, D.: Scan alignment and 3-D surface modeling with a helicopter platform, *Proc. 4th International Conference on Field and Service Robotics* (2003).
34) Tomasi, C. and Kanade, T.: Shape and motion from image streams under orthography: A factorization method, *International Journal of Computer Vision*, Vol.9, No.2, pp.137–154 (1992).
35) Visnovcova, J., Zhang, L. and Gruen, A.: Generating a 3D model of a bayon tower using non-metric imagery, *Proc. International Workshop Recreating the Past —Visualization and Animation of Cultural Heritage* (2001).
36) Walter, E. and Prontazo, L.: *Identification of*

Parametric Models from Experimental Data, Springer (1997).
37) Zoller+Fröhlich GmbH.: http://www.zf-lase.com/.
38) Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision*, Vol.13, pp.119–152 (1994).

(Received September 10, 2006)
(Accepted March 20, 2007)

(Editor in Charge: *Jun Sato*)



Atsuhiko Banno received his B.Eng. degree from the University of Tokyo in 1994 and M.Eng. degree from the University of Tokyo in 1996. He had worked in National Research Institute of Police Science. He received his Ph.D. degree in information science and technology from the University of Tokyo in 2006, and now is in Institute of Industrial Science, the University of Tokyo as a project research associate. His current research interests are 3D reconstruction and Structure from Motion. He received the IPSJ Yamashita SIG Research Award in 2005. He is a member of IEEE and IPSJ.

Katsushi Ikeuchi received his B.Eng. degree in mechanical engineering from Kyoto University in 1973 and Ph.D. degree in information engineering from the University of Tokyo in 1978. He is a professor at the University of Tokyo. After working at the Artificial Intelligence Laboratory at Massachusetts Institute of Technology, the Electrotechnical Laboratory of the Ministry of International Trade and Industries, and the School of Computer Science at Carnegie Mellon University, he joined the University of Tokyo in 1996. He has received various research awards, including the David Marr Prize in ICCV1990, IEEE R&A K–S Fu Memorial Best Transaction Paper Award in 1998, and best paper awards in CVPR1991, VSMM2000, VSMM2004 and VSMM2005. In addition, his 1992 paper, "Numerical Shape from Shading and Occluding Boundaries" was selected as one of the most influential papers to have appeared in the Artificial Intelligence Journal within the past 10 years. He is a fellow of IEEE.