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

Regular Paper

Registration and Deformation of 3D Shape Data through Parameterized Formulation

Tomohito Masuda† and Katsushi Ikeuchi†

In this paper, we investigate conventional registration implementation, consisting of rotation and translation, to design the most precise registration so as to accurately restore the 3D shape of an object. To achieve the most accurate registration, our registration implementation needs robustness against data noise, or initial pose and position of data. To verify the accuracy of our implemented registration, we compare the registration behavior with the registration behavior of conventional methods, and evaluate the numerical accuracy of transformation parameter obtained by our registration. However, registration by rigid-body transformation is not enough for modeling and shape comparison: registration with deformation is needed. In this paper, we extend our robust registration to simultaneously estimate the shape parameter as well as the rigid-body transformation parameter. This extension method assumes that the deformation is formulated strictly from the deformation mechanism. We additionally introduce the applications of our extension method.

1. Introduction

Recently, progress has been made in restoring the accurate 3D shapes of objects in the real world using computer graphics. In this research, a laser range sensor is usually used to capture the 3D shape data of an object. However, the shape data is just partial because of the view limitation of the sensor at one scanning. In order to reconstruct the whole shape of the object, therefore, it is necessary to restore the neighboring status of partial data that can compose the whole shape of the object. This restoration process involves registration among 3D data.

Registration among 3D data is usually achieved by rigid-body transformation consisting of translation and rotation. This is implemented by the iterative minimization framework of the squared sum of the distance between closest points among overlapping 3D data of point cloud (Iterative Closest Point, ICP). There are various kinds of implementation according to the purpose of the procedure.

In this paper, we investigate conventional registration implementation to design the most precise registration so as to accurately restore the 3D shape of an object. In our design of registration implementation, the top priority is its accuracy, even if its computation cost could be expensive as far as the computation complexity is within the limit of the current computer platform. To achieve this, our registration implementation needs robustness against data noise, or initial pose and position of data. To verify the accuracy of our implemented registration, we compare the registration behavior with the behavior of conventional registrations, and evaluate the numerical accuracy of transformation parameter obtained by our registration.

However, registration by rigid-body transformation is not enough for modeling and shape comparison: registration when deformation is needed. In this paper, we extend our robust registration to simultaneously estimate the shape parameter as well as the rigid-body transformation parameter. This extension method assumes that the deformation is formulated strictly from the deformation mechanism. Using this extension framework, we implement a deformation registration to estimate the shape parameter from the shape measurement data of a mathematical plaster model made at the end of the 19th century.

The proposed deformation registration pays attention to the significance of estimated parameter as well as the convergent registration result. To remove the distortion of data obtained by the sensor suspended beneath the balloon (Floating Laser Range Sensor, FLRS), we exploit our deformation registration for the distortion rectification, regarding the movement of FLRS during scanning as shape parameter. In each implementation, we evaluate the accuracy of the estimation of the shape parameter.

† The University of Tokyo

2. Related Works

2.1 Iterative Closest Point Algorithm

Automatic registration we consider here needs to give the initial pose and position resulting in the optimal registration. This acquisition of initial pose and position can be achieved by a user through Graphic User Interface (GUI).

However, initial registration through GUI is, at most, the result that the user subjectively and visually regards as the optimal one, so the closest points in this stage might not be the closest points in the optimal registration result. In the ICP algorithm framework, therefore, the point correspondence in between neighbor data sets is taken as the closest point temporally in the current registration status, and then the registration is gradually improved. These two steps, the point correspondence and registration improvement, are iteratively repeated until the optimal registration is reached[1].

Assuming that multiple neighbor data sets are considered in the registration, the straightforward quantative function, which we call "objective function" here, is defined as follows:

$$f(\mathbf{t}_i, \mathbf{R}_i) = \sum_{j \neq i} \sum_{k} ||\mathbf{R}_i \mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}||^2, \quad (1)$$

where $\mathbf{t}_i$ translation vector,
$\mathbf{R}_i$ rotation matrix,
$\mathbf{x}_k$ kth point in the transformed data set,
$\mathbf{y}_{jk}$ the corresponding point (closest point) of $\mathbf{x}_k$ in the j th neighbor data set.

The registration problem is to find the parameter vector $\mathbf{t}_i$ and $\mathbf{R}_i$ in this function.

After obtaining their parameter set at each iterative step, $\mathbf{x}_k$ can be updated to $\mathbf{x}'_k$ as follows:

$$\mathbf{x}'_k = \mathbf{R}_i \mathbf{x}_k + \mathbf{t}_i. \quad (2)$$

2.2 Registration Strategies

The above ICP algorithm was proposed by Besl and McKay[1], and became the most fundamental framework for 3D data registration. This algorithm framework reduces registration to the minimization problem of the distance sum between the corresponding data by the iterative calculation. The function minimization with respect to the transformation parameter leads the optimal one which represents the plausible transformation between the aligning data sets, for example, three translation and three rotation parameters in the case of the rigid-body transformation. This framework assumed that two data sets were roughly aligned, and that the shape of a transformed data set was the partial one of the neighbor data set. Currently it is extended in various way in order to handle multiple data sets and to pursue the robustness and the speed of convergence.

We can classify them from the viewpoint of the registration ordering, matching unit, point correspondence, error metric, and outlier elimination.

2.2.1 Registration Ordering

In the registration of multiple sets of 3D data, the ordering affects the convergence of the final result. The sequential ordering chooses a corresponding pair of data sets at each iteration for the registration, and repeats this process until all the data sets are aligned[2)~6)]. Its computation cost is lower because only two data sets are handled at each registration. However, it is susceptible to registration failure since the registration errors are locally accumulated and this causes the local discrepancy of the registration result.

In contrast, the simultaneous ordering aligns all the data together at each iteration. Although its computation cost is higher, it enables more accurate registration because the registration error is distributed globally. Consequently, we adopt the simultaneous ordering.

2.2.2 Matching Unit

Matching unit determines the way for selecting the points used in the registration. The matching unit of the ICP algorithm has two kinds: All-points matching uses all points of a data set. Feature-points matching uses only points satisfied with some condition, for example, only high-curvature points.

Assuming that one-to-one correspondence exists among all the feature points, the feature-points matching usually does not change their correspondence at any iteration[7),8)]. So it cannot achieve the accurate registration in the case in which the correspondence cannot be taken precisely. Even if it changes their correspondence, the feature points are unreliable when the range data has considerable noise, because the feature points are derived by some differential operation.

The all-points matching updates the correspondence so that it can be more plausible as the iteration proceeds[1),9)], and therefore can achieve more accurate registration. Hence, our registration uses all-points matching.

2.2.3 Point Correspondence

Point correspondence determines how the corresponding $\mathbf{x}_k, \mathbf{y}_{jk}$ is chosen in Eq. (1). There are many implementations in finding corresponding pairs.

As described in Section 2.1, the typical ones are nearest neighbor correspondence 1),9) (**Fig. 1**-(1)) and normal direction correspondence 10) (Fig. 1-(2)). Nearest neighbor correspondence is taken as the nearest pair in Euclidean space.

Normal direction correspondence is taken as the nearest pair in the normal direction of a point, and they are time-consuming. In contrast, laser ray direction correspondence can reduce the computational cost drastically 5),11),12) (Fig. 1-(3)). This correspondence is taken in the direction of a laser ray emitted from the sensor in 3D point measurement. In Ref. 12), its search computation mainly depends on the graphics hardware. In the case of normal and laser ray direction correspondence, the correspondence is taken between the point $\mathbf{x}_k$ and the point ($\mathbf{y}_{jk}$) on the plane hit in the laser direction of the point $\mathbf{x}_k$. Since the plane is calculated by the differential operation, so lots of wrong correspondences are caused because of the data noise.

Registration accuracy and convergence speed change greatly according to their point correspondence, and Rusinkiewicz et. al. quantatively evaluate this in Ref. 13). Paying attention to the difference of these convergence characteristics, Ref. 4) adopts the hybrid correspondence of nearest point-to-point and point-to-plane. The top priority in our implementation is a registration accuracy, so we employ the nearest neighbor correspondence because the accuracy is guaranteed for the registration of various classes of shape in this correspondence.

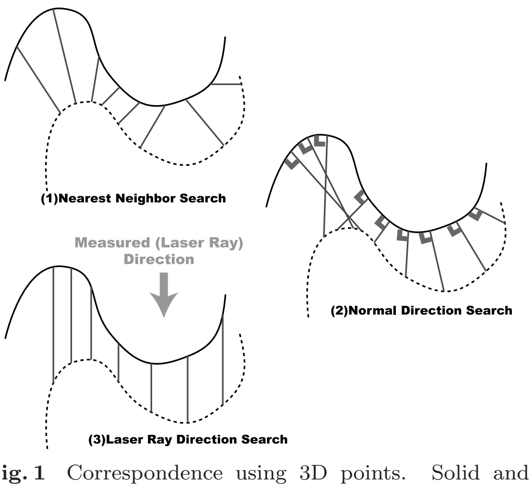

(1)Nearest Neighbor Search
(2)Normal Direction Search
Measured (Laser Ray) Direction
(3)Laser Ray Direction Search

Fig. 1 Correspondence using 3D points. Solid and dotted lines depict the 3D data sets. The red line shows the correspondence between the 3D data on solid and dotted lines. The solid line is then aligned to the dotted line according to the correspondence between the data.

2.2.4 Error Metric

The error metric depends on what kind of value $\mathbf{x}_k$ is. Namely, $\mathbf{x}_k$ may represents a position vector, sometimes including a color (red, blue, green: RGB) vector associated with the point.

In most implementations, the Euclidean distance of the matching point is mainly used 10),11). Some other algorithms adopt such additional information as the surface normal and curvature 14), the reflectance (the reflection ratio of the laser ray)15) and color of the captured point as the error metric 16) in order to make up for the inaccuracy of point coordinate. In the latter algorithms, the Eq. (1) have to be changed as the following:

$$\begin{aligned} &f(\mathbf{t}_i, \mathbf{R}_i) \\ &= \sum_{j \neq i} \sum_{k} \left|\left| \begin{bmatrix} \mathbf{R}_i \mathbf{x}_k \\ \alpha \mathbf{e}_{\mathbf{x}_k} \end{bmatrix} + \begin{bmatrix} \mathbf{t}_i \\ \mathbf{0} \end{bmatrix} \right.\right. \\ &\quad \left.\left. - \begin{bmatrix} \mathbf{y}_{jk} \\ \alpha \mathbf{e}_{\mathbf{y}_{jk}} \end{bmatrix} \right|\right|^2. \end{aligned} \tag{3}$$

Here $\mathbf{e}_{\mathbf{x}_k}$ and $\mathbf{e}_{\mathbf{y}_{jk}}$ represent such properties as color, normal, or curvature (possibly their combination), of $\mathbf{x}_k$ and $\mathbf{y}_{jk}$, respectively. And α means weight against the squared norm of Euclidean distance.

In our implementation, we use only the Euclidean distance.

2.2.5 Outlier Elimination

To cope with outlier, such as data noise and wrong point correspondence in the initial registration, we need to reconsider the objective function. The straightforward function is represented as follows:

$$E(\mathbf{p}_i) = \sum_{j \neq i, k} z_{jk}(\mathbf{p}_i), \tag{4}$$

where

$$\mathbf{p}_i = (\mathbf{t}_i, \mathbf{R}_i), \tag{5}$$

$$z_{jk}(\mathbf{p}_i) = ||\mathbf{R}_i \mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}||^2. \tag{6}$$

In this straightforward least-square (LS) objective function, noise leads to an imprecise registration of 3D data, because the exact correspondences between the noisy data are unknown. Any erroneous correspondences must be eliminated before registration, and a thresholding is often used to eliminate such false correspondences[3)~5)]. The threshold value can be determined as a fraction of the standard deviation, σ, to the errors in the data[17)]. Typically, it is set to greater than or equal to 3σ. This is the simplest method, but it is unreliable method because elimination is affected by the binary classification of the threshold value.

Better outlier elimination can be provided by M-estimation[15),18)], since probability distribution of the error is considered. M-estimation maximizes the probability by minimizing a function of the form

$$E(\mathbf{p}) = \sum_k \rho(z_k(\mathbf{p})), \tag{7}$$

where $\rho(z)$ is an arbitrary function of the errors z_k in the data set. The M-estimator is the maximum-likelihood estimator such that the probability distribution P is equivalent to $E(z_k)$.

A Lorentz function is used as the M-estimator; a Lorentz function can be represented as:

$$\rho(z_k(\mathbf{p})) = \log\left(1 + \frac{1}{2\sigma^2} z_k(\mathbf{p})\right). \tag{8}$$

Wheeler summarized the registration behavior according to the probability distribution in M-estimator in Ref. 19).

2.3 Deformation Registration

In this paper, we propose the extended framework of the conventional registration algorithm to allow the shape deformation during registration process. This kind of registration, namely, deformation registration, has been researched in such field as the medical imaging, and the target object for the registration is mainly soft tissues. They adopt similarity[20)], affine[14)], a kind of spline (octree-spline)[21)], geometric hashing[22)], quadric/superquadric[23)], and displacement-field-based transformation[24)] so that their deformation works well for any kind of target shape. For another example, deformable motion parameter are estimated using deformable net model that enables a linear transformation of deformable objects[25)].

These methods can be generally adopted in shape modeling and fitting. However, if the deformation is strictly defined by some parameterized formulation derived form the deformation mechanism, the deformation is much more accurate when using its formulation than when derived from their methods. The parameters obtained from our strict formulation carry with them the essential information about the cause and origination of the deformation. So our framework pays as much attention to the obtained parameters as to the appearance resulting from the deformation. In this point, our aim is different from theirs. So in our assumption that the shape changes are strictly represented with a mathematical formula including some variable parameters and its formula is known a priori, we formulate the generally extended registration which allows the 3D data to be deformed and determines both the deformation and the translation and rotation parameters.

3. Robust Determination of Translation and Rotation Parameters

3.1 Robust Simultaneous Registration Algorithm

Based on the previous section, here we explain the details of our designed registration.

In the ICP based registration algorithm, the acquisition of the valid initial parameter is important for the optimal registration result. As a preprocess in our implementation, all the initial transformation parameters are set manually, using GUI, with accuracy good enough to reach a true optimum. In iterative process, the followings are done:

(1) Constructing kd-trees of data sets.

(2) Searching nearest neighbors using kd-trees.

(3) Minimizing the objective function (squared sum of nearest neighbor distance) to find the better (optimal) registration parameter.

(4) Updating data sets according to the obtained registration parameter.

The above process is repeated until the optimal registration is reached.

This kind of algorithm is usually time-consuming, and most of the computation cost depends on searching the corresponding (closest) point. Therefore, we use kd-tree search. Kd-tree is a tree structure from data set to effectively search closest points. We have already proposed more effective kd-tree algorithm for nearest neighbor search than the basic one[26)], and we adopt it in our designed registration.

Pseudo-code for this technique is shown as Algorithm 1 in Section 4.1 together with the difference from that of our extended registration. Ignore the gray boxes in Section 4.1 if you understand just the simultaneous registration without deformation.

3.2 Minimization of Objective Function for Parameter Estimation

Our registration algorithm aligns all data sets simultaneously so as to minimize the squared sum of nearest neighbor point-to-point distances. The objective function is represented as follows:

$$E(\mathbf{p}_i) = \sum_{j \neq i} \sum_{k} \rho(z_{jk}(\mathbf{p}_i)), \tag{9}$$

where

$$\mathbf{p}_i = (\mathbf{t}_i, \mathbf{q}_i), \tag{10}$$

$$z_{jk}(\mathbf{p}_i) = ||\mathbf{R}(\mathbf{q}_i)\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}||^2, \tag{11}$$

$$\rho(z_{jk}(\mathbf{p}_i)) = \log(1 + \frac{1}{2\sigma^2} z_{jk}(\mathbf{p}_i)), \tag{12}$$

$\mathbf{t}_i$: translation vector,
$\mathbf{R}(\mathbf{q}_i)$: rotation matrix corresponding to quaternion q_i,
$\mathbf{x}_k$: kth point in the data set of interest,
$\mathbf{y}_{jk}$: the corresponding point of x_k in the jth measured data set.

As for its rotation matrix, we use a quaternion representation of 3 Degrees Of Freedom (DOF).

Using error metric $E(\mathbf{p}_i)$, we compute the parameters $\mathbf{p}_i$ which fulfill the following equation:

$$\mathbf{p}_{i_{opt}} = \arg\min_{\mathbf{p}_i} E(\mathbf{p}_i). \tag{13}$$

For the gradient-based solution of our nonlinear optimization, the descent gradient is:

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{p}_i} &= \sum_{j \neq i} \sum_{k} \frac{\partial \rho(z_{jk})}{\partial z_{jk}} \cdot \frac{\partial z_{jk}}{\partial \mathbf{p}_i} \\ &= \sum_{j \neq i} \sum_{k} \frac{1}{2\sigma^2 + z_{jk}} \cdot \frac{\partial z_{jk}}{\partial \mathbf{p}_i} \end{aligned} \tag{14}$$

If we evaluate $\partial z_{jk} / \partial \mathbf{p}_i$ by an identity quaternion $\mathbf{q}_I$, we can represent $\partial z_{ij} / \partial \mathbf{p}_i$ as

$$\begin{aligned} &\frac{\partial z_{jk}(\mathbf{p}_i)}{\partial \mathbf{p}_i} \\ &= 2(\mathbf{R}(\mathbf{q}_i)\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}) \\ &\quad \left. \frac{\partial(\mathbf{R}(\mathbf{q}_i)\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk})}{\partial \mathbf{p}_i} \right|_{\mathbf{q}_I} \\ &= \begin{bmatrix} 2(\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}) \\ 4\mathbf{C}(\mathbf{x}_k)^T(\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}) \end{bmatrix} \\ &= \begin{bmatrix} 2(\mathbf{x}_k + \mathbf{t}_i - \mathbf{y}_{jk}) \\ -4\mathbf{x}_k \times (\mathbf{t}_i - \mathbf{y}_{jk}) \end{bmatrix}, \end{aligned} \tag{15}$$

because the (negative) gradient of quaternion at an identity quaternion $\mathbf{q}_I$ is obtained by Eq. (16).

$$\left. \frac{\partial(\mathbf{R}(\mathbf{q}_i)\mathbf{x}_k)}{\partial \mathbf{q}_i} \right|_{\mathbf{q}_I} = 2\mathbf{C}(\mathbf{x}_k)^T. \tag{16}$$

Here, $\mathbf{C}(\mathbf{x}_k)$ is the skew-symmetric matrix to represent a cross product of two vectors as follows:

$$\begin{aligned} \mathbf{a} \times \mathbf{b} &= \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{bmatrix} \\ &= \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \\ &= \mathbf{C}(\mathbf{a})\mathbf{b}, \end{aligned} \tag{17}$$

$$\text{where} \quad \mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix},$$

and it has such property as $\mathbf{C}(\mathbf{a})^T = -\mathbf{C}(\mathbf{a})$.

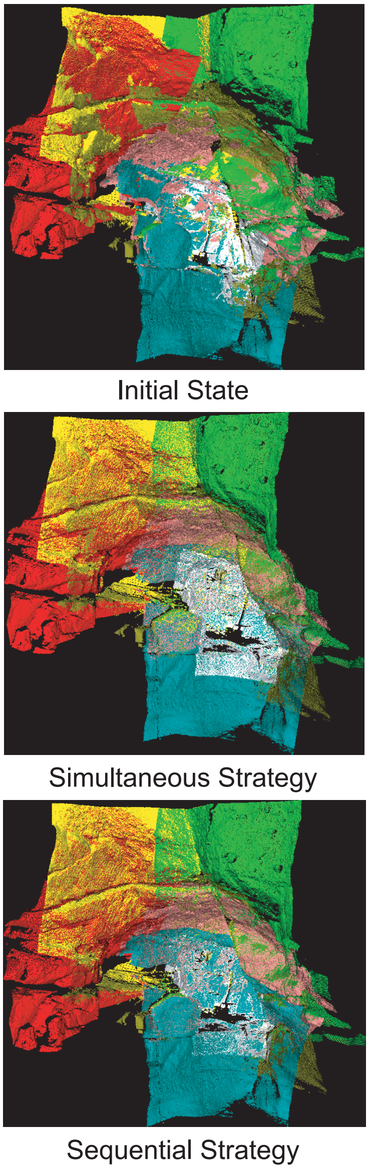
From the obtained descent gradient, the conjugate gradient is calculated so that all the obtained gradient is guaranteed to be orthogonal. Transformation vector $\mathbf{p}_i$ is acquired using the conjugate gradient [27)~29)] and line minimization method with a combination of golden ratio bracketing (golden section search) and parabolic fits.

3.3 Evaluation

In this section, we quantify the effectiveness of our robust registration on the basis of four issues by comparing previous registration methods. First, we discuss the merits of adopting a simultaneous strategy. Second, we observe the effectiveness of using stochastic outlier elimination to increase the robustness of the technique. Third, to evaluate the effectiveness of these two steps, we evaluate the overall estimation accuracy of our registration.

3.3.1 Simultaneous vs. Sequential Ordering

In this evaluation, we align seven partial data sets of the Fugoppe Cave in simultaneous and sequential strategies. The data sets were acquired by Cyrax 2500 (Leica), and the standard deviation of this sensor is 2 [mm]. The average length of neighbor edges are 5 [mm]. The used data sets are shown in **Fig. 2**, and the height in this figure is 6 [m]. The upper figure in Fig. 2 shows the initial state of these data sets. They are slightly shifted among the overlapping data. The middle and lower figures in Fig. 2 respec-



Initial State

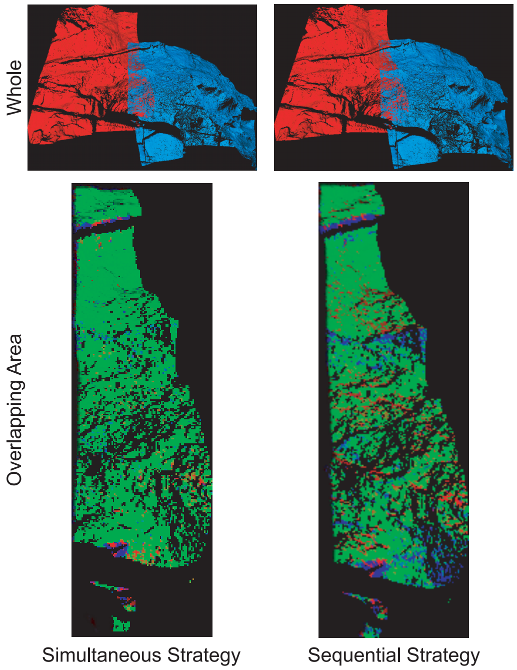
Simultaneous Strategy

Sequential Strategy

Fig. 2 Registration results in simultaneous and sequential strategies. The upper figure shows the initial state of partial 3D data. The middle and lower figures respectively show the registration results in simultaneous and sequential strategies.

tively show the registration result in simultaneous vs. sequential ordering. The sequential registration we used is basically the implementation proposed by Ref. 2). In order to observe only the effect of simultaneous and sequential strategies, however, this sequential registration uses M-estimator for outlier elimination.

In sequential registration, we must determine the data pairing of alignment targets such that they are exactly overlapping each other. Sequential registration considers pairing only two data sets at a time, and assumes that the registration works well among each pair of data sets. So if the transformation is determined in one data set, it is transformed together with the rest of the data.



Simultaneous Strategy

Sequential Strategy

Fig. 3 Detailed observation of registration results of a pair of aligned data sets in simultaneous and sequential strategies. The upper figures show their whole appearance of registration results in simultaneous (left) and sequential (right) strategies. The lower figures show the detail of their overlapping area between them. The green shows no difference (less than 1 [cm]), while the red and blue show larger differences (more than 1 [cm]).

Good registration is visible as evenly mottled pattern in overlapping area because of slight differences in sampling and because of the random noise, even if the area has an identical shape.

Comparing simultaneous registration methods, sequential registration introduces local discrepancies (between the yellow and cyan-blue data sets, for example). The detailed observation of a pair of aligned data sets is shown in **Fig. 3**. In the lower figures, the green color indicates areas of little difference (less than 1 [cm]), while the red and blue colors indicate areas of larger difference (more than 1 [cm]). Simultaneous registration results in almost no dif-

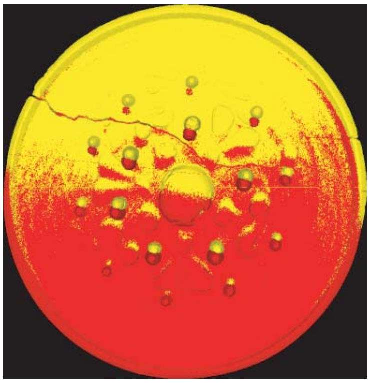


Fig. 4 Initial pose and position between two mirrors. In this figure, The yellow mirror is slightly translated and rotated against the red. (Data Informant: Kashihara Institute of Archaeology and Tokyo National Museum.)

ference, by comparison; simultaneous registration is clearly better.

3.3.2 Straightforward Least Square Registration vs. Robust Registration

In this investigation, we align the data sets of two ancient mirrors that were cast from the same mold. They have local differences in their shapes.

The initial pose and position between them is shown in **Fig. 4**. They are aligned using straightforward LS registration as well as robust M-estimator registration. The former is the registration proposed by Ref. 17), and the latter is our implemented registration. The registration result is shown in **Fig. 5**. Figures in the first row show the convergence result.

Figures in the second and third row, respectively, show the convex and concave areas of one mirror vs. the other when the length between each corresponding point is exceeded by the setting threshold. This threshold is respectively set to 0.5 and 0.25 [mm] in the second and third row. In the second row, the upper circular area has more concave area when using an M-estimator, but the lower area has more convex area otherwise. Similarly in the third row, the left area has more convex area in the M-estimator result, but the right area has more convex area otherwise. As shown in the numerical results, the green area, regarded as an area of no difference in shape, is 51.7 and 49.6 percent of the total in the middle, and is 77.0 and 77.3 percent in the lowest, respectively, in the case for which the thresholds are 0.25 and 0.5. This result shows that the outlier area is automatically recognized and ignored in the registration process in order to align as much area as possible.

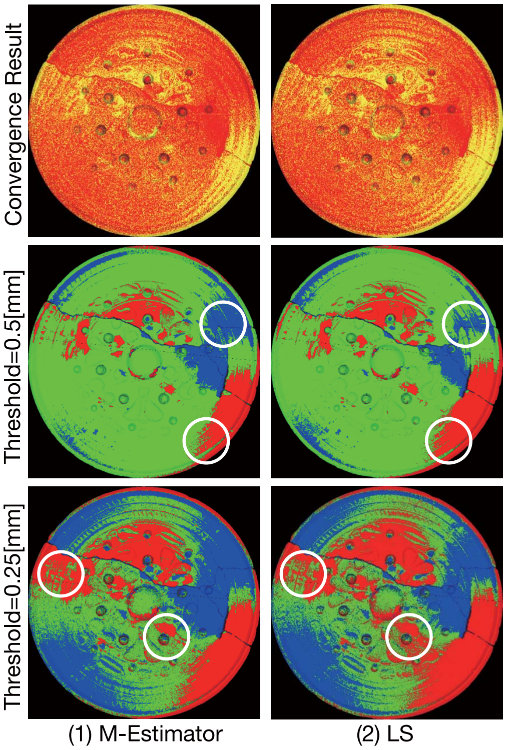


Fig. 5 Convergence results of two mirrors. Figures in the first row show the convergence result, and figures in the second and third row show the convex and concave areas of one mirror against the other when the length between each corresponding point is exceeded by the setting threshold, regarded as shape difference. (Data Informant: Kashihara Institute of Archaeology and Tokyo National Museum.)

3.3.3 Estimation Accuracy of Translation and Rotation parameters

In this investigation, we align two data sets capturing the face of a tower at Bayon ruin in Cambodia (**Fig. 6**-(a)). To consider the registration of the actual measurement data, we created two data sets from the same measurement data by sub-sampling the different points. Figure 6-(b) shows the appearance of two superimposed data, which is regarded as the correct registration between them.

To create the initial position states of two data sets — original and transformed data sets — the transformed data set is translated and rotated, then it is realigned to the original data. The estimation accuracy of the registration parameter is regarded as the difference between the amount of the translation and rotation of

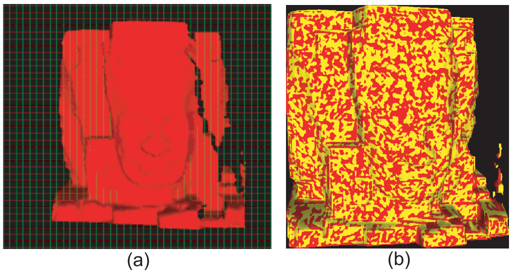


Fig. 6 Data used in this evaluation. In (a) in this figure, the size of each unit square is 0.1 by 0.1 [m]. Here, positive axes of x-axis and y-axis are respectively set to the right and upper direction, and the positive direction of z-axis is set to the front direction, perpendicular to this figure.

the transformed data set in its initial state and that of the registration result.

The initial position of transformed data set is set to three steps in translation and rotation respectively. It is translated to ± 0.5 [m] in each axis, and is rotated to ± 30 [deg] around all the direction that can be represented as the combination of −1, 0, and 1 in each axis. As a result, the number of translation settings is 26 (27 (3^3) minus 1 (to remove the trivial translation (0, 0, 0) because the translation distance of the different sampling data sets is approximately 10^{-3} [m] at the convergence if they are aligned without changing their initial pose and position)). In rotation, the number of rotation axes is 26 (27 (3^3) minus 1 (likewise, to remove the "rotation" axis (0, 0, 0))), but half of these axes are symmetrical with respect to the coordinate origin (For example, (−1, 1, 1) and (1, −1, −1)), so the actual number of rotation axes is 13. In each axis, transformed data set is rotated 30 [deg] in clockwise and counter-clockwise directions, so the number of rotation settings is 26. The number of combining case of translations and rotations is 676 (26 × 26). Then, the number of our test cases is 728 (26 cases of only translation, 26 cases of only rotation, and 676 cases of translation and rotation caused together).

Figure 7 shows the registration process. In this case, transformed data set is translated 0.5 [m] and rotated 30 [deg] around the z axis, and they are accurately aligned after the registration.

As a numerical result, we show twelve parameter sets which are considered typical of all the estimation results, in **Fig. 9**. In this figure, "x-t 0.5" means 0.5 [m] translation along x axis, and "x-r 30" means 30 [deg] rotation around x axis.

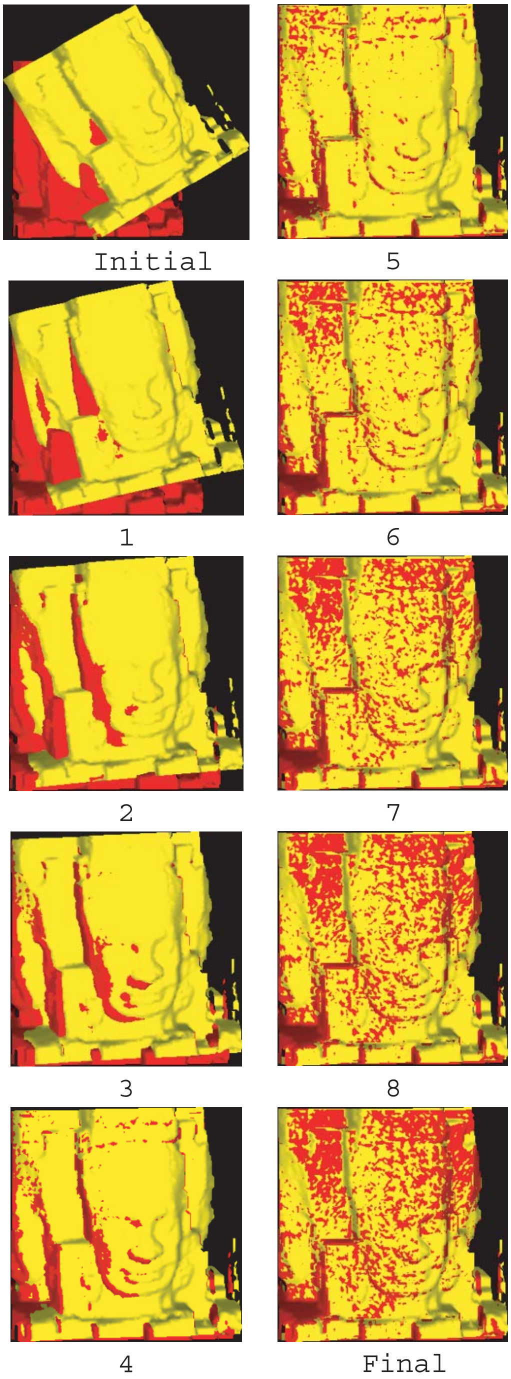


Fig. 7 Registration process.

When the initial translation and rotation is set as shown in the translation and rotation axes (e.g., x-t 0.5, x-t −0.5, ..., z-t −0.5, x-r 30, x-r −30, ..., z-r −30), the difference between each true initial parameter and the corresponding estimated parameter with respect to each axis

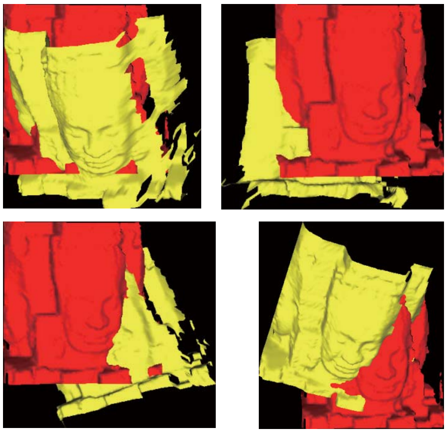


Fig. 8 Pairs of data sets which result in registration failure.

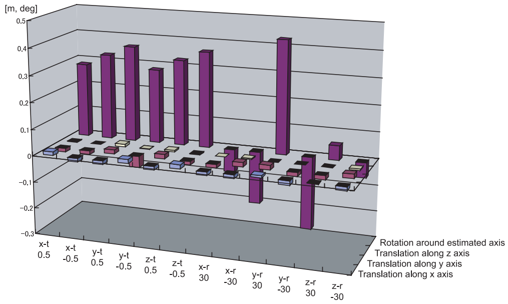


Fig. 9 Estimation errors in translation and rotation for each initial position.

(Translation along x axis, Translation along y axis, Translation z axis, Rotation around estimated axis) is shown on the vertical axis (e.g., 0, ± 0.1, ...). This figure shows the translation and rotation estimation errors of our registration is respectively within 0.05 [m] and 0.5 [deg].

Of our 728 cases, 694 result in good registration. **Figure 8** shows four pairs of initially positioned data sets which result in registration failure. Because we can easily observe large position differences in these 34 cases as shown in Fig. 8, our registration seldom fails if the initial position estimate is manually improved.

In addition, we investigate the result of two implementations proposed by Ref. 12) (laser ray direction, point-to-plane correspondence, thresholding) and Ref. 17) (nearest neighbor, point-to-point, thresholding) by aligning 728 pairs of data set in the same condition as the above. In the first registration method [12), 483 result in good registration. Observing the registration process, the convergence of this registration looks slow until the optimal registration is acquired. And in the second registration method [17), 715 result in good registration. These pairs are completely superimposed in all area each other, so it looks preferable not to employ the operation for outlier elimination. To verify this, we create and align the partial shape data sets as shown in **Fig. 10**. The initial setting of translation and rotation is the same as the above. Then in our registration, 343 result in goodregistration, while 335 result in good registration in the registration of Ref. 17). Though our initial setting is rough in this evaluation, our implementation can prove to be more robust than Ref. 17), if the more detail evaluation is done than in this investigation.

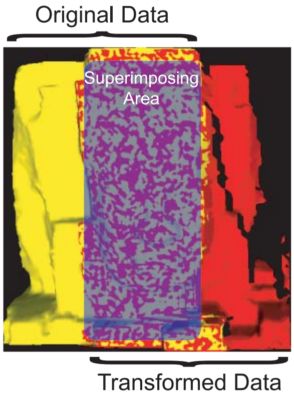


Fig. 10 Partial data used to compare the registration result of ours and Ref. 17)

4. Extension of Rigid-body Transformation

In this section, we first generally extend the rigid-body transformation to allow deformation during a registration. Therefore, estimated parameters include those which affect their shape in addition to six parameters of the pose and position in a conventional registration. In later sections, we adopt this extended framework to solve each problem.

4.1 Simultaneous Determination of Registration and Deformation Parameters

Our proposal assumes that the deformation can be represented by a parameterized mathematical formula whose form is known a priori, but whose parameters are unknown.

Our goal is to simultaneously determine these deformation, translation, and rotation parameters by comparing the target data to transform with its corresponding data. We do this using the ICP framework: translation and rotation parameters are determined in a minimization paradigm. If we fix the translation and rotation parameters, determination of the deformation parameter becomes an iterative shape matching problem. Thus, we can handle all parameter determinations in a unified minimization framework.

We extend the parameter estimation of the registration formulation to add the shape parameter by extending the objective function in Eq. (11). Therefore, $z_{jk}(\mathbf{p})$ in Eq. (11) is transformed into:

$$z_{jk}(\mathbf{p}_i) = ||\mathbf{R}(\mathbf{q}_i)\mathbf{g}(\mathbf{x}_k, \mathbf{s}_i) + \mathbf{t}_i - \mathbf{y}_{jk}||^2, \tag{18}$$

where $\mathbf{p}_i =$ $(\mathbf{t}_i, \mathbf{q}_i, \mathbf{s}_i)$,
$\mathbf{s}_i$: shape parameter,
$\mathbf{g}(\mathbf{x}_k, \mathbf{s}_i)$: deformation function of point $\mathbf{x}_k$
with respect to parameter $\mathbf{s}_i$.

Our rigid-body registration is designed to be robust, and here we adopt the same strategy as in Section 3. In this extended framework, we consider the registration of multiple data sets. Pseudo-code is as follows.

```
Algorithm 1 SimultaneousRegistrationAndDeformation
input: InitialDataSets $\{D_i|i = 0, 1, ..., n-1\}$ {initially aligned data sets}
output: AlignedDataSets $\{D_i|i = 0, 1, ..., n - 1\}$ {updated and aligned data sets}
local: Kdtree $\{K_i|i = 0, 1, ..., n - 1\}$
local: RegistrationAndDeformationParameter $\{P_i|i = 0, 1, ..., n - 1\}$
local: PreviousDescentGradient $V$
local: CurrentDescentGradient $S$
local: PreviousConjugateGradient $U$
local: CurrentConjugateGradient $C$
local: TemporalDataSet $T$
const: GainControl $\lambda$

  repeat
    for all $i = 0, 1, ..., n - 1$ do
      $K_i \leftarrow$ MakeKdtree($D_i$)
      $P_i \leftarrow identityparameter$ {initializing registration and deformation parameter}
    end for
    for all $i = 1, 2, ..., n - 1$ do
      for all $j = 0, 1, ..., m - 1$ do
        $T \leftarrow UpdateDataSet(P_i, D_i)$ {temporally updating data set according to current $P_i$}
        $S \leftarrow identityparameter$ {initializing descent gradient}
        for all $k = 0, 1, ..., n - 1$ do
          if $i \neq k$ then
            $S \leftarrow S+DescentGradient(K_k, T)$
          end if
        end for{process for conjugate gradient method begins next}
        if $j = 0$ then
          $C \leftarrow S$
          $U \leftarrow S$
        else
          $C \leftarrow ConjugateGradient(S, U, V)$ {converting descent gradient to conjugate gradient}
          $U \leftarrow C$
        end if
        $V \leftarrow S$ {process for conjugate gradient method ends here}
        $C \leftarrow LineMinimization(\{K_l|l = 0, 1, ..., n-1\}, P_i, C)$ {1-dimensional search in the direction of conjugate gradient $C$}
        $P_i \leftarrow P_i + C$
      end for
    end for
    for all $i = 1, 2, ..., n - 1$ do
      $P_i \leftarrow \lambda P_i$ {decelerating transformation to converge stably}
      $D_i \leftarrow UpdateDataSet(P_i, D_i)$
    end for
  until Registration AndDeformation Error Converged
end
```

Notice that n and m respectively represent the number of data sets and parameters.

4.2 Detail Explanation of Our Algorithm

We denote D_i, K_i, and P_i $(i = 0, 1, ..., n - 1)$ respectively by a data set of 3D points in the ith range image, kd-tree of a data set D_i, the transformation parameter of D_i in each iteration. For every i, all points in every data set D_i are converted to a kd-tree structure in function *MakeKdtree* (**Fig. 11**). These kd-trees are used in function *DescentGradient*. Function *ConjugateGradient* converts the descent gradient of a transformed data set into the conjugate gradient in registration parameter space. As a final procedure, Function *LineMinimization* performs 1-

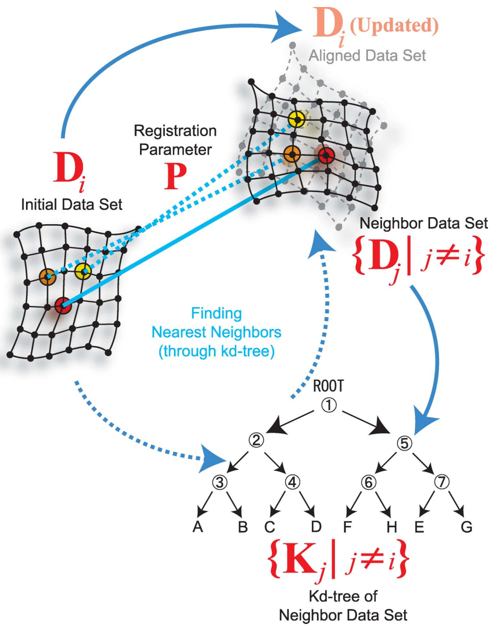


Fig. 11 Illustration of alignment process.

dimensional search in the direction of conjugate gradient. This procedure is described in Section 3.2. After obtaining every transformation parameters, Function *UpdateDataSet* updates the data sets according to the obtained transformation parameter as shown in Eq. (2).

In our algorithm, two ideas are implemented to align data sets stably, and to avoid the wrong convergence and the vibration. First, one data set (D_0 in the pseudo-code, for example) is not transformed. Second, the transformation is decelerated without using the obtained transformation parameter itself. Our implementation solves the optimal parameters of both translation and rotation together, then the deceleration parameter (GainControl λ in the pseudo-code) is determined so as not to exceed the maximum permissible levels of both translation and rotation at each iteration. In our implementation, the maximum permissible translation and rotation are empirically determined as a fraction of the scale of a transformed data set, and as a constant, respectively.

In addition, the minimization of objective function is usually involved with respect to all registration parameters $(P_0, P_1, ..., P_{n-1})$ (letting the dimensional number of $P_k (k = 0, 1, ..., n-1)$ be m, then the parameter dimension is $m \times n$.), but our registration is implemented by minimizing the objective function with respect to each registration parameter $(P_i) (i = 0, 1, ..., n-1)$ (where the parameter dimension is m) because our registration assumes to begin with a good initial state, and to be gradually improved because of the transformation deceleration. As a consequence, both implementations provide the same convergence result. Moreover, it is easier to implement the minimization with respect to each parameter, and to extend this rigid-body transformation to allow the deformation during a registration step.

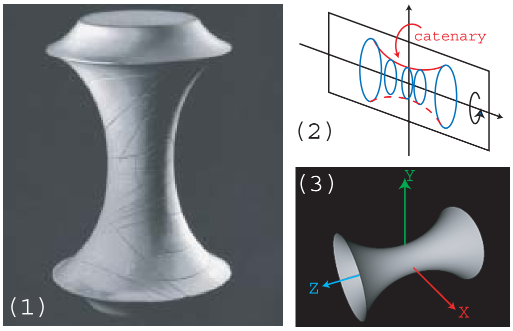


Fig. 12 (1) A Mathematical model and (2) its ideal representation used in our experiment. It has the constant negative curvature on all points of their surface. [Data Informant: Prof. Toshitake Kohno (Graduate School of Mathematical Sciences, The University of Tokyo)]

5. Shape Parameter Estimation of Mathematical Model

5.1 Mathematical Model : Revolution Surface of Catenary

As a main topic in this section, we estimate the shape parameter of certain mathematical model made of plaster in order to examine its manufacturing accuracy (**Fig. 12**). This model is a cultural asset; it was manufactured in Germany at the end of the 19th century for educational purposes. It has been displayed in The University Museum, The University of Tokyo.

This object has no documentation, and we are interested in identifying the shape parameters the makers used in manufacturing it. We wish to estimate deformation parameters by applying our extended registration framework algorithm to both measured data sets and the data set computed by mathematical formula, in order to evaluate the manufacturing accuracy of the plaster model.

Using our estimated parameters, we also wish to remake more accurate model for comparison, because both historians and the mathematicians are interested in the level of manufacturing skill extant in those days. Our target is the model that is called "revolution surface of catenary".

5.2 Mathematical Formula and Experimental Result

The surface generated by rotating a 2D catenary is shown in Fig. 12-(1). Such a surface always has azimuthal symmetry. Besides scale parameter (l), there are two parameters (a, b) involved in the generation of such surfaces. This model does not depend on measurement points, so $\mathbf{g}(\mathbf{x}_k, \mathbf{s}_i)$ in Eq. (18) is actually $\mathbf{g}(\mathbf{s}_i)$. The numerical formula is as follows:

$$\textit{shape parameter} \quad \mathbf{s}_i = (a, b, l) \quad (0 < b \leq a),$$

$$\mathbf{g}(\mathbf{s}_i) = (l\phi(v)\cos u, l\phi(v)\sin u, l\psi(v)), \tag{19}$$

where

$$0 \leq u \leq 2\pi,$$

$$-a \cdot \sinh^{-1}\left(\frac{a}{b}\right) \leq v \leq a \cdot \sinh^{-1}\left(\frac{a}{b}\right),$$

$$\phi(v) = b\cosh\left(\frac{v}{a}\right), \tag{20}$$

$$\psi(v) = \int_0^v \sqrt{1 - \frac{b^2}{a^2}\sinh^2\left(\frac{t}{a}\right)}\,dt. \tag{21}$$

Then, the descent gradient is calculated as

$$\frac{\partial z_{jk}(\mathbf{p}_i)}{\partial \mathbf{p}_i} = \begin{bmatrix} 2(\mathbf{R}_i\mathbf{g}(\mathbf{s}_i) + \mathbf{t}_i - \mathbf{y}_{jk}) \\ -4\mathbf{g}(\mathbf{s}_i) \times (\mathbf{t}_i - \mathbf{y}_{jk}) \\ 2(\mathbf{R}_i\mathbf{g}(\mathbf{s}_i) + \mathbf{t}_i - \mathbf{y}_{jk})\mathbf{R}_i\frac{\partial(\mathbf{g}(\mathbf{s}_i))}{\partial \mathbf{s}_i} \end{bmatrix}, \tag{22}$$

where

$$\mathbf{p}_i = (\mathbf{t}_i, \mathbf{q}_i, \mathbf{s}_i).$$

The 3D shape of the plaster model was captured using a VIVID 900 (KonicaMinolta) range finder. The data sets were initially aligned using a manual process via Graphic User Interface (GUI). Initial shape parameter was also manually estimated. **Figure 13** shows the registration process. The result was well-behaved and convergent. The shape parameters were estimated as follows:

$$a = 0.0568, \quad b = 0.0237, \quad l = 0.996.$$

5.3 Evaluation

Our estimation is affected by various kinds of errors: range data measurement errors; initial registration errors; and the errors in the manually input initial shape parameter. We have already reported how the accuracy of our estimated parameter depended on such errors by using synthesized data computed using known parameters and adding Gaussian noise[30].

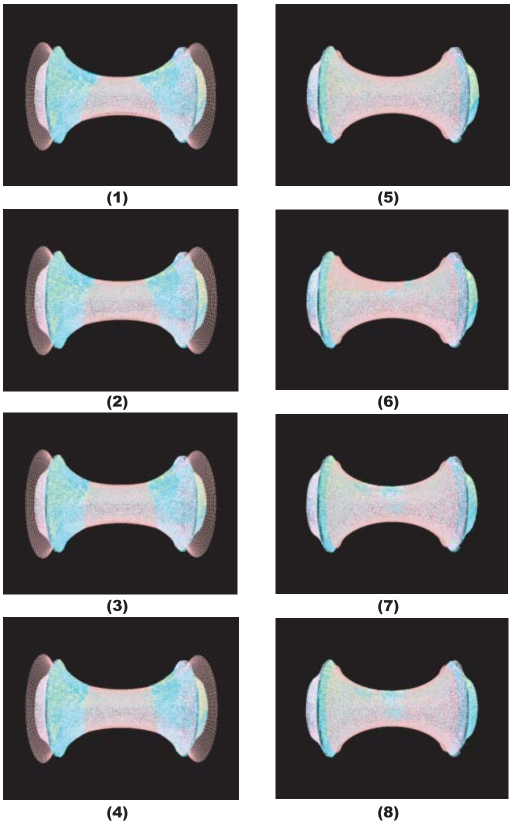


Fig. 13 The convergent characteristic of the parametric data. Views (1)-(8) show gradual convergence to the data of the revolution surface of a 2D catenary.

Additionally, here we investigated the combined effects of an initial translation, rotation, and specified shape parameter. The initial shape parameter (a, b, l) of the calculated data was set to five steps around each truth value. In particular, a, b, and l were set to 0.03, 0.04, 0.05, 0.06, 0.07, to 0.01, 0.015, 0.02, 0.025, 0.03, and to 0.7, 0.85, 1.0, 1.15, 1.3. Initial translation and rotation were exclusively set to three steps as follows: translation to 0.01, 0.02, and 0.03 [m] along x and z axes, and rotations of 10, 20, and 30 [deg] around the x axis. This results in 124 deformation cases (125 (5^3) minus 1 (0.05, 0.02, 1.0: the truth value)). There are 9 translation and rotation cases, so there are 1,116 (124 × 9) cases to investigate. Altogether, therefore, we investigated 1,249 (124 + 9 + 1,116) cases.

Of these 1,249 cases, 991 result in the correct registration. Judging from these results, a registration tends to fail if there is too much dif-

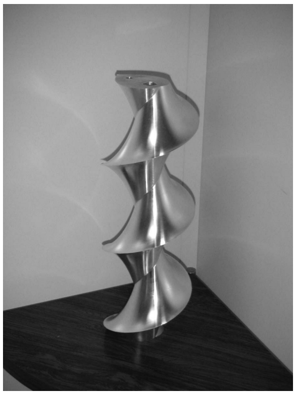


Fig. 14 Reproduced metallic mathematical model.

ference between the initial value and the truth. These data sets are obviously different in their shape and position; these differences might be easy to cancel because the user can immediately recognize a deficiency and re-run the algorithm after improving the initial shape parameter and position estimates.

5.4 Reproduction of Mathematical Model

Using our algorithm of shape parameter estimation, another mathematical model of Dini's Surface was reproduced in metal by Yamada Seiki Co., ltd.[31] under the supervision of an artist, Mr. Hiroshi Sugimoto[32]. Yamada Seiki Co.,ltd successfully generated the 3D Shape of the original model with high accuracy (**Fig. 14**), and Mr. Sugimoto held an exhibition of the work at the Mori Art Museum at Roppongi Hills.

In this way, our algorithm can create CAD (computer-aided design) primitives and compressed 3D shape data faithful to the original shape, and as a result, we can refine or alter the shape as desired.

6. Registration for Range Data Obtained by Floating Laser Range Sensor

6.1 Floating Sensing System

To obtain 3D measurement data of large objects, a laser range sensor (LRS) mounted on a tripod is often used. Unfortunately, it often happens that some part of a large object is invisible from the ground. In order to scan these invisible faces, a scaffold might be built nearby. However, this involves time and expense, and moreover, some surfaces might still not be visible due to space limitations for this scaffolding, lack of a viable superstructure, and so forth.

In order to remedy this problem, several aerially-based measurement systems have been proposed. For example, aerial 3D measurements could be taken with a laser range sensor installed on a helicopter platform[33)~36)]. High frequency vibration of the platform, however, would need to be considered to obtain accurate results. Another technique is aerial stereo photography with a digital camera attached to a balloon[37]; however, this method cannot achieve a high level of measurement precision. For another example, the 3D data acquisition method of freely moving objects can be developed by using Lissajous pattern for tracking objects[38]. But this system does not use raster scan unit, so this rectification algorithm cannot be adopted for general range sensors.

We have developed a novel 3D measurement system[39]. Our system digitizes objects from the air while being suspended beneath a balloon. Although our system is free from high frequency vibration like that caused by helicopter engines, there still remains low frequency movement due to the floating balloon which distorts the data. However, this movement can be modeled as simple trajectory consisting of constant velocity translation and constant angular velocity rotation.

Our system consists of two main processes: scanning and registration. For the 3D scanning of visible surfaces from the ground, we use an LRS mounted on a tripod on the ground, as usual. To scan facets invisible from the ground, such as the rooftop of a building, we have developed and tested a Floating Laser Range Sensor (FLRS). The FLRS data contains distortion caused by the swing motion of the balloon during scanning, but our extended registration framework can be applied to remove this to rectify the data.

6.2 Floating Laser Range Sensor

Our FLRS system consists of a scanner unit, a controller and a personal computer (PC). These three units are suspended below a balloon.

Our scanner unit includes a laser range finder especially designed to be hung from a balloon. Our design requirements were that the unit be compact and lightweight enough to be carried by a balloon, and that it be fast enough to minimize the influence of the balloon's normal swing.

The scanner unit includes a spot laser radar unit and two mirrors. We chose to use the

LARA25200 supplied by Z+F Inc. as a laser radar unit because of its high sampling rate (maximum 625,000 [points/sec]).

6.3 Inter- and Intra-Scanning Registration

6.3.1 Assumption and Formulation of FLRS Motion

In order to align data sets from the FLRS, we distinguish between two different types of movement, "inter-scanning" and "intra-scanning". Inter-scanning movements provide different views of a scene, and are equivalent to a series of rigid-body transformations. But the FLRS moves during the acquisition of each range data set; this intra-scanning movement of the sensor distorts the measurement data. Our extended registration framework enables the rectification of this distortion; we can represent this motion as a deformation parameter.

The motion of FLRS during scanning depends on the following:

(1) Its initial velocity
(2) Its initial angular velocity
(3) Any acceleration generated by external force
(4) Any angular acceleration generated by external moments

We can ignore the influence of translation and angular accelerations because our FLRS needs only one second to scan a frame. Therefore, we consider FLRS movement to have constant velocity in translation and rotation, without changing its rotation axis during a frame. Under this assumption we set up the deformation equation in Eq. (18).

Figure 15 shows positional relationship in intra-scanning registration. Here, $\mathbf{O}_u$ means the origin of the camera coordinate system for the case in which FLRS does not move during a scan, and $\mathbf{O}_d$ means the origin of the camera coordinate system at the time τ_k. (Note that $0 \le \tau_k \le 1$; one measurement can require up to a second.)

Assuming that the FLRS moves during scanning, τ_k is the time elapsed since the first point was captured. Then FLRS acquires a 3D point $\mathbf{a}_u$ in the camera coordinate system u (i.e., $\mathbf{a}_d$ in the camera coordinate system d).

If the corresponding point of $\mathbf{a}_u$ is $\mathbf{b}_u$ in the world coordinate system w, then the error in this registration can be represented as

$$z_i = ||\mathbf{R}_{w\leftarrow u}\mathbf{a}_u + \mathbf{T}_u - \mathbf{b}_u||^2. \tag{23}$$

If the FLRS moves during acquisition, the measurement point is captured from the origin of the camera coordinate $\mathbf{O}_d$ at the measurement time τ_k. Letting the translation vector and rotation matrix from the coordinate system u to the coordinate system d at time τ_k be $\mathbf{v}_u\tau_k$ and $\mathbf{R}_{d\leftarrow u}(\tau_k)$ respectively, then in the coordinate system u,

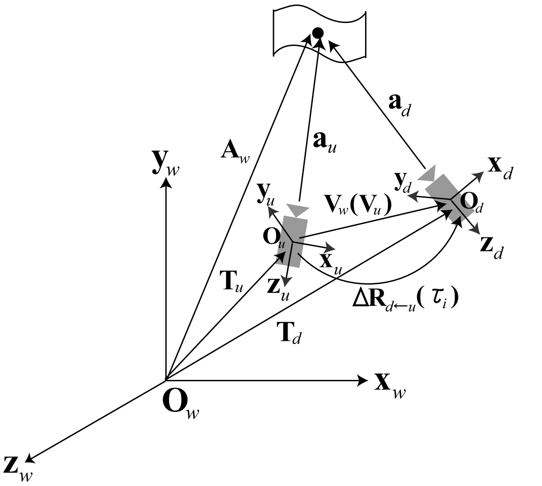


Fig. 15 Positional relationships in the case of intra-scanning registration. O_w, is the origin of the world coordinate system, O_u the camera coordinate system origin when the measurement starts, and O_d is the camera coordinate at time τ_k elapsed since its start. Capital and small letters are respectively concerned with the world coordinate and each camera coordinate shown by their subscript.

$$\mathbf{a}_u = \Delta\mathbf{R}^{-1}_{d\leftarrow u}(\tau_k)\mathbf{a}_d + \mathbf{v}_u\tau_k. \tag{24}$$

Notice that $\mathbf{R}^{-1}_{d\leftarrow u}(\tau_k)$ is equal to $\mathbf{R}_{u\leftarrow d}(\tau_k)$.

Substituting Eq. (24) for Eq. (23),

$$z_i = ||\mathbf{R}_{w\leftarrow u}(\Delta\mathbf{R}^{-1}_{d\leftarrow u}(\tau_k)\mathbf{a}_d + \mathbf{v}_u\tau_k) + \mathbf{T}_u - \mathbf{b}_u||^2. \tag{25}$$

In this case, the geometric function $\mathbf{g}(\mathbf{x}_k, \mathbf{k})$ is represented as follows:

$$\mathbf{g}(\mathbf{x}_k, \mathbf{p}_{intra}) = \Delta\mathbf{R}^{-1}_{d\leftarrow u}(\tau_k)\mathbf{x}_k + \mathbf{v}\tau_k, \tag{26}$$

where $\mathbf{p}_{intra}$ include the state of the intra-rotation axis, its angular velocity, and the intra-translation ($\mathbf{v}$).

In addition to the parameters of the rigid-body transformation $\mathbf{R}_{w\leftarrow u}$, $\mathbf{T}_u$, we have to estimate the deformation parameter of $\Delta\mathbf{R}^{-1}_{d\leftarrow u}(\tau_k)$, $\mathbf{v}_u$.

Intra-rotation is represented by the description of the rotation axis and angular velocity, but these parameters cannot be obtained in the same way as the rigid-body rotation solution which involved a quaternion derivative. In case of rigid-body rotation, the rotation axis description is first calculated, and then the amount of rotation around this calculated axis can be determined by the quaternion normalization. This rigid-body rotation is common to the whole data. But intra-rotation does change with the time τ_k at each i-th point, namely, it must be represented as a function with respect to τ_k.

To remedy this problem, we represent $\Delta\mathbf{R}_{d\leftarrow u}^{-1}(\tau_k)$, by allowing $\mathbf{m}$ and ω be the rotation axis and angular velocity respectively, as follows:

$$\begin{aligned}&\Delta\mathbf{R}_{d\leftarrow u}^{-1}(\tau_k)\\ &=\Delta\mathbf{R}_{d\leftarrow u}^{-1}(\mathbf{m},\omega\tau_k)=\begin{pmatrix} r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{pmatrix}\end{aligned}$$

where

$$\begin{cases} r_{11}=(1-\cos\omega\tau_k){m_x}^2+\cos\omega\tau_k\\ r_{12}=(1-\cos\omega\tau_k)m_xm_y-(\sin\omega\tau_k)m_z\\ r_{13}=(1-\cos\omega\tau_k)m_zm_x+(\sin\omega\tau_k)m_y\\ r_{21}=(1-\cos\omega\tau_k)m_xm_y+(\sin\omega\tau_k)m_z\\ r_{22}=(1-\cos\omega\tau_k){m_y}^2+\cos\omega\tau_k\\ r_{23}=(1-\cos\omega\tau_k)m_ym_z-(\sin\omega\tau_k)m_x\\ r_{31}=(1-\cos\omega\tau_k)m_zm_x-(\sin\omega\tau_k)m_y\\ r_{32}=(1-\cos\omega\tau_k)m_ym_z+(\sin\omega\tau_k)m_x\\ r_{33}=(1-\cos\omega\tau_k){m_z}^2+\cos\omega\tau_k\end{cases}\tag{27}$$

$$\mathbf{m}=(m_x,m_y,m_z)\quad \textit{and}\quad ||\mathbf{m}||=1.\tag{28}$$

6.3.2 Parameter Gradient of Objective Function

To derive the descent gradient in this non-rigid case, we replacing $\mathbf{R}_{w\leftarrow u}$, $\Delta\mathbf{R}_{d\leftarrow u}^{-1}(\tau_k)$, $\mathbf{a}_d$, $\mathbf{v}_u$, τ_k, $\mathbf{T}_u$, and $\mathbf{b}_u$ with $\mathbf{R}_i$, $\Delta\mathbf{R}_i(\tau_k)$, $\mathbf{x}_k$, $\mathbf{v}$, τ_k, $\mathbf{T}_i$, and $\mathbf{y}_{jk}$ in Eqs. (25) and (26) to obtain the following:

$$z_{jk}=||\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk}||^2,\tag{29}$$

$$\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})=\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)\mathbf{x}_k+\mathbf{v}_i\tau_k,\tag{30}$$

where

$$\mathbf{p}_{intra}=(\mathbf{v},\mathbf{m},\omega).\tag{31}$$

The descent gradient is therefore:

$$\begin{aligned}&\frac{\partial z_{jk}}{\partial\mathbf{p}_i}\\ &=2(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})\\ &\quad\frac{\partial(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})}{\partial\mathbf{p}_i}\end{aligned}\tag{32}$$

$$=\begin{bmatrix} 2(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})\\ -4\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})\times(\mathbf{T}_i-\mathbf{y}_{jk})\\ 2(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})\mathbf{R}_i\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\mathbf{v}}\\ 2(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})\mathbf{R}_i\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\mathbf{m}}\\ 2(\mathbf{R}_i\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})+\mathbf{T}_i-\mathbf{y}_{jk})\mathbf{R}_i\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\omega}\end{bmatrix},\tag{33}$$

where

$$\mathbf{p}=(\mathbf{T}_i,\mathbf{q}_i,\mathbf{v},\mathbf{m},\omega),\tag{34}$$

$$\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\mathbf{v}}=\begin{bmatrix}(\tau_k & 0 & 0)^T\\ (0 & \tau_k & 0)^T\\ (0 & 0 & \tau_k)^T\end{bmatrix},\tag{35}$$

$$\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\mathbf{m}}=\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial\mathbf{m}}\mathbf{x}_k,\tag{36}$$

$$\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\omega}=\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial\omega}\mathbf{x}_k.\tag{37}$$

Again, notice that only the rigid-body rotation is evaluated at q_I.

$\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial\mathbf{m}}$ and $\frac{\partial\mathbf{g}(\mathbf{x}_k,\mathbf{p}_{intra})}{\partial\omega}$ in Eqs. (36) and (37) can be derived from Eq. (27) as follows:

$$\begin{aligned}&\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial m_x}\\ &=\begin{bmatrix} 2(1-\cos\omega\tau_k)m_x & (1-\cos\omega\tau_k)m_y & (1-\cos\omega\tau_k)m_z\\ (1-\cos\omega\tau_k)m_y & 0 & -\sin\omega\tau_k\\ (1-\cos\omega\tau_k)m_z & \sin\omega\tau_k & 0\end{bmatrix},\end{aligned}\tag{38}$$

$$\begin{aligned}&\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial m_y}\\ &=\begin{bmatrix} 0 & (1-\cos\omega\tau_k)m_x & \sin\omega\tau_k\\ (1-\cos\omega\tau_k)m_x & 2(1-\cos\omega\tau_k)m_y & (1-\cos\omega\tau_k)m_z\\ -\sin\omega\tau_k & (1-\cos\omega\tau_k)m_z & 0\end{bmatrix},\end{aligned}\tag{39}$$

$$\begin{aligned}&\frac{\partial\Delta\mathbf{R}_i(\mathbf{m},\omega\tau_k)}{\partial m_z}\\ &=\begin{bmatrix} 0 & -\sin\omega\tau_k & (1-\cos\omega\tau_k)m_x\\ \sin\omega\tau_k & 0 & (1-\cos\omega\tau_k)m_y\\ (1-\cos\omega\tau_k)m_x & (1-\cos\omega\tau_k)m_y & 2(1-\cos\omega\tau_k)m_z\end{bmatrix},\end{aligned}\tag{40}$$

and

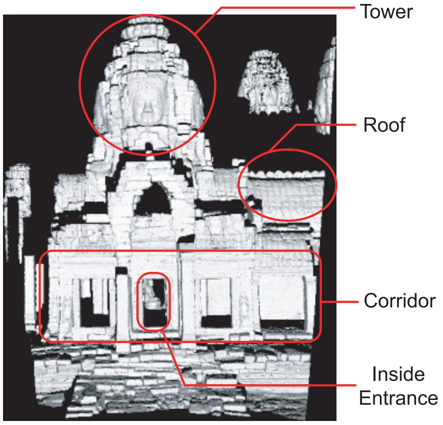


Fig. 16 Detail of Bayon Data. This range data is the partial shape of the Bayon temple in Cambodia.

$$\frac{\partial \Delta \mathbf{R}_i(\mathbf{m}, \omega\tau_k)}{\partial \omega} = \tau_k \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

where

$$\begin{cases} f_{11} = ({m_x}^2 - 1)\sin\omega\tau_k \\ f_{12} = m_x m_y \sin\omega\tau_k - m_z \cos\omega\tau_k \\ f_{13} = m_z m_x \sin\omega\tau_k + m_y \cos\omega\tau_k \\ f_{21} = m_x m_y \sin\omega\tau_k + m_z \cos\omega\tau_k \\ f_{22} = ({m_y}^2 - 1)\sin\omega\tau_k \\ f_{23} = m_y m_z \sin\omega\tau_k - m_x \cos\omega\tau_k \\ f_{31} = m_z m_x \sin\omega\tau_k - m_y \cos\omega\tau_k \\ f_{32} = m_y m_z \sin\omega\tau_k + m_x \cos\omega\tau_k \\ f_{33} = ({m_z}^2 - 1)\sin\omega\tau_k \end{cases} \qquad (41)$$

6.4 Experiment

As an experiment on an actual case, we executed our algorithm on the data of the Bayon temple. In this experiment, we aligned the corresponding data captured by our FLRS and Cyrax 2500. The latter data set was scanned from the stable ground so that there was no movement during scanning, and we assume that it is sufficiently reliable. The details of the data set and the registration process are respectively shown in **Figs. 16** and **17**. The overlapping area between data sets of FLRS and Cyrax 2500 is illustrated in the initial state (top figure) in Fig. 17. You can see the big gap in shape between two data sets in the initial state, but the gap is guradually improved, and as a result, they are aligned and fitted in the final state.

6.5 Evaluation

To evaluate the accuracy of the algorithm, we aligned the two data sets used in the accuracy evaluation of our rigid-body transformation pa-

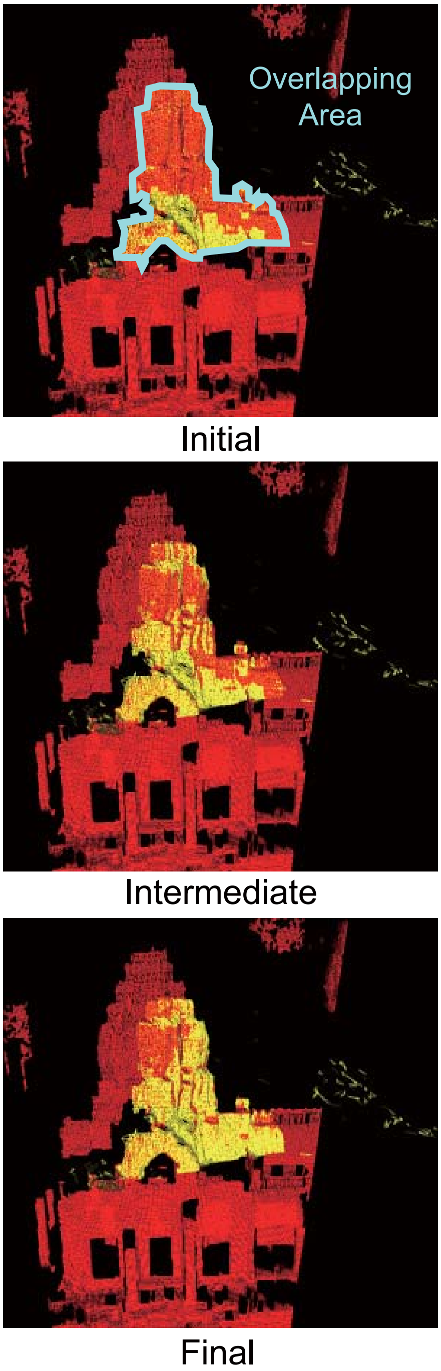


Fig. 17 Range images in our registration process: A range image of FLRS (yellow) is aligned and fitted onto the corresponding range image of Cyrax 2500 (red) simultaneously. There is initially a big gap in shape between two data sets, but the gap is gradually reduced as our extended registration proceeds.

rameter (See Section 3.3.3). One data set is regarded as a data set without distortion, and the other as being deformed according to Eq. (26). This latter data set is equivalent to the quality of a data set obtained by FLRS. We considered the position difference (inter-translation and inter-rotation: rigid-body translation and rotation) of our two data sets.

The initial intra-transformation (intra-translation and intra-rotation) of pseudo-FLRS data was manipulated in five steps: exclusively setting translation to ±0.5, ±0.25, and 0 [m/s] in each axis, and exclusively setting rotation to ±20, ±10, and 0 [deg/s] around each axis. In this scenario we assume the rotation axis is known. The number of actual intra-transformation cases are therefore 168 because the number of only intra-translation, only intra-rotation, and combined intra-transformation is respectively 12 (4 (±0.5 and ±0.25) × 3 (each axis)), 12 (4 (±20 and ±10) × 3 (each axis)), 144 (12 × 12). (The effect of a varying initial rotation axis is investigated later.)

Next, the effects of inter-transformation (inter-translation and inter-rotation) are considered. The initial inter-transformation was manipulated in three steps: translation to ±0.1 and 0 [m], and rotation to ±5 and 0 [deg]. In this case, the number of inter-transformation is 12 because the number of only inter-translation and only inter-rotation is 6 (2 (±0.5) × 3 (each axis)) and 6 (2 (±5) × 3 (each axis)). Thus, the total number of evaluation pattern is 2169 (144 (only intra-transformations) + 12 (only inter-transformations) + 144 × 12 (combination of intra- and inter-transformation)).

As a numerical result, we show 24 evaluation results on behalf of all the estimation results in **Figs. 18** and **19**. Figure 18 and 19 respectively investigate the effects of only inter-transformation as well as only intra-transformation on parameter estimation. In these figures, “x-t” means translation along x axis, and “x-r” means rotation around x axis. Judging from these figures, the resulting errors of inter-translation, inter-rotation, intra-translation, and intra-rotation are respectively within 0.15 [m], 2.2 [deg], 0.11 [m/s], and 1.9 [deg/s]. It can be seen that inter-transformation errors tend to have more effect on parameter estimation than that of intra-transformation.

For estimation evaluation of an intra-rotation axis, a data set in which intra-translation is set to −0.25 [m] along z axis (= (0, 0, 1))and intra-rotation is set to −10 [deg] is aligned to the data set without distortion. Then, we set the initial intra-rotation axis to (0.00, 0.50, 0.87 $(= \frac{\sqrt{3}}{2})$). After registration, estimated axis of intra-rotation is (0.00, 0.48, 0.88). Therefore, the intra-rotation axis can be reasonably estimated by our method, with its resulting appearance visibly close to the truth (**Fig. 20**). However, the initial estimate must be somewhat close to the truth or wrong convergence may result.

Of 2,196 cases, we considered 1,860 cases resulted in good registration. Generally speaking, as long as the initial error of inter-translation, inter-rotation, intra-translation, and intra-rotation is respectively within ±0.5 [m], ±5 [deg], ±0.25 [m/s], and ±20 [deg/s], the registration result will be accurate, even if these effects are combined.

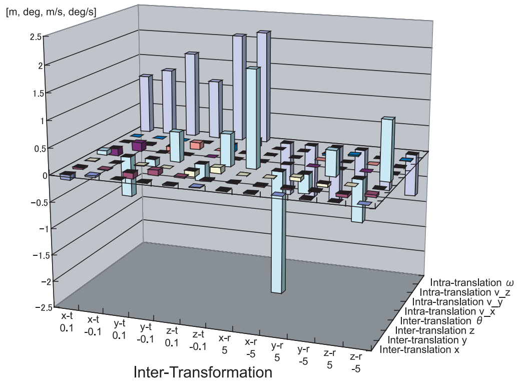


Fig. 18 Estimation errors of parameters for each initial inter-transformation.

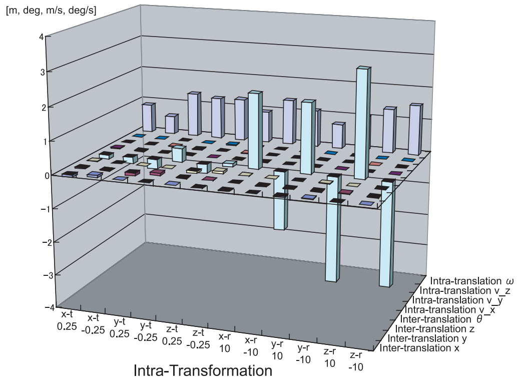


Fig. 19 Estimation errors of parameters for each initial intra-transformation.

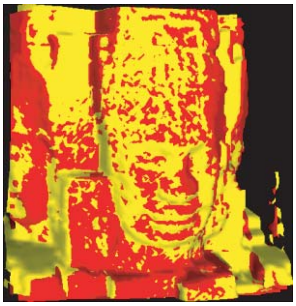


Fig. 20 Registration result in the case which the different parameter of intra-rotation axis is initially set.

7. Conclusion and Future Work

In this thesis, we proposed the robust simultaneous registration of 3D shape data. This registration reduces the solution of a nonlinear equation to iteratively minimize the distance between a pair of corresponding 3D data sets. As a preparation for designing the registration algorithm, we analyzed the merits and demerits of conventional methods, and to design the most accurate registration algorithm, we adopted the simultaneous ordering, all point matching, closest point-to-point distance, and M-estimator for outlier elimination.

To verify the robustness of our registration against the initial position and the measurement noise, we evaluated the estimation accuracy of registration parameter, comparing the registration result between our method and conventional registration. To summarize our implemented registration, it can align only the identical part of superposing 3D shape data robustly because of the use of M-estimator, and can restore even complex shape since simultaneous strategy is employed.

Moreover, we extended our registration, namely, rigid-body transformation, to enable registration among 3D data that can deform each other through some known mathematical formula. This extended method requires determining more parameters concerned with shape than just the six translation and rotation parameters. It can solve the rigid-body transformation and shape parameter in a unified manner. Here we assumed that the shape changes are strictly defined by some parameterized formulation derived from the deformation mechanisms.

We employed our extended registration to estimate the shape parameter from the shape measurement data of mathematical plaster models. We successfully estimated their parameters and reproduced a metallic replica model of the original Dini's Surface with high accuracy. We verified the estimation accuracy through a simulation-based experiment.

We also applied our extended registration to registration among range images obtained by the laser range sensor suspended beneath a balloon. In contrast with a conventional 3D sensing system, this registration needs to rectify the distortion due to the movement during measurement. We also evaluated the estimation accuracy of FLRS movement, and the applicable limitation.

In our future work, we have some goals for improving our system. The first is to automatically determine the initial pose and position among aligned data sets. This automatic determination would enable totally automatic registration among 3D data sets. The problem is how to lead the initial state for our system to work well.

As the next goal, the capability of our deformable data should be investigated in more detail. In this paper, we do not pay attention to the relation of the registration correctness to the coverage ratio of identity surface between undeformable and deformable data. And it is interesting to consider the deformable data registration without rigid (undeformable) data. In that case, some additional constraints might be needed. In this thesis, we always assumed the registration between rigid data and deformable data. This problem here may also be how to determine the initial pose, position, and shape parameter as described above.

The applications we proposed here are only a few of the possible applications, and we are trying to develop an application to generate the CAD primitives under the shape parameter estimated from the range image. This application will convert the range images into the properly approximated CAD data sets. The benefit of this application is to be able to compress the range images which usually consist of numerous 3D points and polygons. We intend to apply our framework widely to various classes of problem.

Acknowledgments This work was, in part, supported 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". We appreciate the cooperation of Yoichi Town Board of Education in Hokkaido Prefecture, Kashihara Institute of Archaeology, Tokyo National Museum, Japanese Government Team for Safeguarding Angkor (JSA), Professor Toshitake Kohno, Professor Yoshiaki Nishino (The University of Tokyo), Mr. Hiroshi Sugimoto (Artist), and Mr. Yasuhiro Yamada (Yamada Seiki Co., Ltd.).

References

1) Besl, P. and McKay, N.: A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp.239–256 (1992).

2) Turk, G. and Levoy, M.: Zipped polygon meshes from range images, *ACM SIGGRAPH Proceedings*, pp.311–318 (1994).

3) Masuda, T. and Yokota, N.: A Robust Method for Registration and Segmentation of Multiple Range Images, *Computer Vision and Image Understanding*, Vol.61, No.3, pp.295–307 (1995).

4) Pulli, K.: Multiview Registration for Large Data Sets, *Proc. 2nd International Conference on 3D Digital Imaging and Modeling*, pp.160–168 (Oct.).

5) Blais, G. and Levine, M.: Registering Multiview Range Data to Create 3D Computer Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.8, pp.820–824 (1995).

6) Masuda, T.: Registration and Integration of Multiple Range Images by Matching Signed Distance Fields for Object Shape Modeling, *Computer Vision and Image Understanding*, Vol.87, pp.51–65 (2002).

7) Higuchi, K., Herbert, M. and Ikeuchi, K.: Building 3-D Models from Unregistered Range Images, *Graphical Models and Image Processing*, Vol.57, pp.315–333 (1995).

8) Johnson, A. and Herbert, M.: Surface Matching for Object Recognition in Complex 3-Dimensional Scenes, *Image and Vision Computing*, Vol.16, No.9/10, pp.635–651 (1998).

9) Simon, D.: Fast and Accurate Shape-Based Registration, PhD Thesis, School of Computer Science, Carnegie Mellon University (1996).

10) Chen, Y. and Medioni, G.: Object Modeling by Registration of Multiple Range Images, *Image and Vision Computing*, Vol.10(3), No.3, pp.145–155 (1992).

11) 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).

12) Oishi, T., Nakazawa, A., Kurazume, R. and Ikeuchi, K.: Fast Simultaneous Alignment of Multiple Range Images Using Index Images, *Proc. 5th International Conference on 3-D Digital Imaging and Modeling* (*3DIM 2005*), pp.476–483 (2005).

13) Rusinkiewicz, S. and Levoy, M.: Efficient Variants of the ICP Algorithm, *Proc. 3rd International Conference on 3D Digital Imaging and Modeling*, pp.145–152 (2001).

14) Feldmar, J. and Ayache, N.: Rigid and Affine Registration of Smooth Surfaces using Differential Properties, *Proc. Third European Conference on Computer Vision* (*ECCV'94*), pp.397–406 (1994).

15) Nishino, K. and Ikeuchi, K.: Robust Simultaneous Registration of Multiple Range Images, *Proc. 5th Asian Conference on Computer Vision*, Vol.2, pp.455–461 (2002).

16) Johnson, A. and Kang, S.: Registration and Integration of Textured 3-D Data, *Proc. International Conference on 3D Digital Imaging and Modeling*, pp.234–241 (1997).

17) Zhang, Z.: Iterative Point Matching for Registration of Free Form Curves and Surfaces, *International Journal of Computer Vision*, Vol.13, No.2, pp.119–152 (1994).

18) Wheeler, M.D. and Ikeuchi, K.: Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.3, pp.252–265 (1995).

19) Wheeler, M. D.: Automatic Modeling and Localization for Object Recognition, PhD Thesis, School of Computer Science, Carnegie Mellon University (1996).

20) Stewart, C.V., Tsai, C.L. and Perera, A.: A View-Based Approach to Registration: Theory and Application to Vascular Image Regisrtaion, *Proc. International Conference on Information Processing in Medical Imaging (IPMI)*, pp.475–486 (2003).

21) Szeliski, R. and Lavallee, S.: Matching 3-D Anatomical Surfaces with Non-Rigid Deformations using Octree-Splines, *International Journal of Computer Vision*, Vol.18, No.2, pp.171–186 (1996).

22) Guéziec, A., Pennec, X. and Ayache, N.: Medical Image Registration using Geometric Hashing, *IEEE Computational Science and Engineering, special issue on Geometric Hashing*, Vol.4, No.4, pp.29–41 (1997).

23) Bardinet, E., Cohen, L. D. and Ayache, N.: A parametric deformable model to fit unstructured 3D data, *Computer Vision and Image*

Understanding, Vol.71, No.1, pp.39–54 (1998).

24) Andresen, P. R. and Nielsen, M.: Non-rigid registration by geometry constrained diffusion, *Proc. Medical Image Computing and Computer-Assisted Intervention* (*MICCAI'99*), pp.533–543 (1999).

25) Yamamoto, M. and Boulanger, P.: Direct Estimation of Range Flow on Deformable Shape from a Video Rate Range Camera, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.15, No.1, pp.82–89 (1993).

26) Sagawa, R., Masuda, T. and Ikeuchi, K.: Effective Nearest Neighbor Search for Aligning and Merging Range Images, *Proc. 4th International Conference on 3-D Digital Imaging and Modeling* (*3DIM 2003*) (2003).

27) Polak, E.: *Computational Methods in Optimization*, New York, Academic Press (1971).

28) Jacobs, D.A.: *The State of the Art in Numerical Analysis*, London, Academic Press (1977).

29) Stoer, J. and Bulirsch, R.: *Introduction to Numerical Analysis*, New York, Springer-Verlag (1980).

30) Masuda, T., Hirota, Y., Ikeuchi, K. and Nishino, K.: Simultaneous Determination of Registration and Deformation Parameters among 3D Range Image, *Proc. 5th International Conference on 3-D Digital Imaging and Modeling* (*3DIM 2005*), pp.369–376 (2005).

31) Yamada Seiki Co., Ltd.
http://www.yamadaseiki.co.jp

32) Koyanagi., G.
http://www.gallerykoyanagi.com/

33) Thrun, S., Daniel, M. and Hahnel, D.: Scan Alignment and 3-D Surface Modelling with a Helicopter Platform, *4th International Conference on Field and Service Robotics*, pp.14–16 (2003).

34) Miller, R. and Amidi, O.: 3-D Site Mapping with the CMU Autonomous Helicopter, *5th International Conference on Intelligent Autonomous System. IAS–5* (1998).

35) Chen, T. and Shibasaki, R.: Ground Truth Measurement System Using RC Helicopter, *Proc. Asian Conference on Remote Sensing*, Chiba University (1999).

36) Gruenl, A., Lil, Z. and Wang, X.: 3D City Modelling with TLS (Three-Line Scanner) Data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol.XXXIV-5/W10 (2003).

37) 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).

38) Blais, F., Picard, M. and Godin, G.: Recursive Model Optimization Using ICP and Free Moving 3D Data Acquisition, *Proc. 4th International Conference on 3-D Digital Imaging and Modeling* (*3DIM 2003*), pp.251–258 (2003).

39) 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.*

(Received September 4, 2006)
(Accepted March 20, 2007)

(Editor in Charge: *Yasuhiro Mukaigawa*)

Tomohito Masuda received the Ph.D. degree in Information Science and Technology from The University of Tokyo in 2006. His research interests include the 3D shape reconstruction of real objects into virtual graphics space, and its application to virtual reality/computer graphics contents.

Katsushi Ikeuchi received the Ph.D. degree in Information Engineering from The University of Tokyo in 1978. He is currently a professor at the Institute of Industrial Science, The University of Tokyo. He has served (or will serve) as the program/general chairman of several international conferences, including 1995 IEEE-IROS (General), 1996 IEEE-CVPR (Program), 1999 IEEE-ITSC (General) and 2003 ICCV (Program). He is an Editor-in-Chief of the International Journal of Computer Vision. He has been a fellow of IEEE since 1998. He was selected as a distinguished lecture of IEEE SP society for the period of 2000–2001.