

REGISTRATION OF MULTIPLE RANGE VIEWS USING THE REVERSE CALIBRATION TECHNIQUE

Do Hyun Chung¹, Il Dong Yun² *, and Sang Uk Lee¹ †

¹: School of Electrical Engineering,

Seoul National University,

Seoul, 151-742, Korea

²:Department of Control and Instrumentation Engineering,

Hankuk University of F. S.,

Yongin, 449-791, Korea

May 20, 1997

⁰The authors are also affiliated with the Engineering Research Center for Advanced Control and Instrumentation, Seoul National University, Seoul 151-742, Korea.

*e-mail: yun@san.hufs.ac.kr

†To be addressed all the correspondence.

Abstract

In this paper we propose a new registration algorithm. The proposed algorithm consists of two steps. The first step is to estimate the transformation parameters among multiple range views, making use of the eigenvectors of the weighted covariance matrix of the 3-D coordinates of data points. The weighting factors are carefully selected to take into account the projection effect caused by different viewpoints. The next step is to register the views iteratively with the estimated transformation parameters as initial values. To solve the correspondence problem, the reverse calibration technique is used, which is adapted to the space encoding rangefinder. The object function, defined by means of the reverse calibration technique, is minimized iteratively. Experimental results show that the proposed algorithm is very fast and efficient.

Key words: rangefinder, space encoding technique, reverse calibration, initial estimation, registration

1 Introduction

The 3-D shape reconstruction is an important problem in computer vision. To reconstruct a complete 3-D shape of an object, whole range data of the surface should be acquired in advance. However, it is impossible, in fact, to obtain whole range data of the surface at once, because of the limitation in the field of view of the range-finder and the characteristics of the surface to be measured. Therefore, in order to acquire whole range data of the surface, it is a common practice to use several range images, acquired from different viewpoints [3]-[11]. In order to merge them into a complete model, the transformation parameters among the different views should be extracted. The extraction of the transformation parameters among multiple range views is called “*registration*” [4, 5, 9, 10, 11].

Besl and McKay [5] proposed the iterative closed point (ICP) algorithm that minimizes

the distance from points in a view to tangential planes at corresponding points in other views. They used an iterative line-surface intersection algorithm to find the corresponding points.

Dorai [10] enhanced the ICP algorithm to be optimal, using the minimum variance estimator technique, and also presented a method to obtain the initial estimates of the transformation parameters using the principal axes of the views.

Blais [11] proposed a registration algorithm, in which the correspondence problem is approached by reversing the calibration process of the laser range-finder. Blais used the simulated annealing technique to minimize the object function. However, the problem with [11] is that it requires a heavy computational complexity.

In this paper, we propose a new registration algorithm for the space encoding range-finder [3]. The proposed algorithm consists of two steps. The first step is to estimate the transformation parameters among multiple range views, by making use of the weighted principal axes which is formed by the eigenvectors of the weighted covariance matrix of the 3-D coordinates of data points. The weighting factors are carefully selected to take into account the projection effect, due to different viewpoints.

The next step is to update the transformation iteratively with initial values estimated in the first step. To solve the correspondence problem without requiring iteration, the reverse calibration technique is adapted to the space-encoding range-finder. Then, the object function, defined by means of the reverse calibration technique, is minimized iteratively.

The rest of this paper is organized as follows. Section 2 briefly describes the basic principles and introduces the reverse calibration technique of the space encoding range-finder. Section 3 presents the proposed registration algorithm. Section 4 provides several experimental results with discussion, and finally Section 5 gives the concluding remarks.

2 Space encoding range finder

In this paper, the space encoding range-finder is used to acquire the range data. For the sake of the complement, the basic principles and the reverse calibration technique for the space encoding range-finder are presented in this section with some useful tips.

2.1 Basic principles

Inokuchi [3] devised a range-finder using a liquid crystal display device to generate gray coded pattern masks. The basic principles, such as how to determine the camera and the projector parameters and how to calculate the 3-D coordinates of the data points, will be briefly discussed in this subsection.

The 3-D coordinates (x_w, y_w, z_w) of a point in the 3-D space and its 2-D image coordinates (x_c, y_c) are related by a composite transformation, including perspective, translational and rotational transformations [1, 3], given by

$$\begin{pmatrix} h_c x_c \\ h_c y_c \\ h_c \end{pmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}, \quad (1)$$

where 3×4 matrix, $\{c_{ij}\}$, is the camera parameter matrix.

Similarly, the 3-D coordinate and the active projector coordinate x_p are defined as

$$\begin{pmatrix} h_p x_p \\ h_p \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \end{bmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}, \quad (2)$$

where 2×4 matrix, $\{p_{ij}\}$, is the projector parameter matrix.

The camera parameter matrix and the projector parameter matrix should be determined before range data acquisition. The procedure to determine these parameters is

called “*calibration*” [3].

Once the camera and projector parameters are determined by the calibration, the 3-D world coordinates of a data point \mathbf{x} can be calculated from its 2-D image coordinates (x_c, y_c) and its space code x_p by using the following relation:

$$\mathbf{x} = \mathbf{Q}^{-1} \mathbf{f}, \quad (3)$$

where $\mathbf{x} = (x_w \ y_w \ z_w)^t$, $\mathbf{f} = (c_{34}x_c - c_{14} \ c_{34}y_c - c_{24} \ p_{24}x_p - p_{14})^t$, and

$$\mathbf{Q} = \begin{bmatrix} c_{11} - c_{31}x_c & c_{12} - c_{32}x_c & c_{13} - c_{33}x_c \\ c_{21} - c_{31}y_c & c_{22} - c_{32}y_c & c_{23} - c_{33}y_c \\ p_{11} - p_{21}x_p & p_{12} - p_{22}x_p & p_{13} - p_{23}x_p \end{bmatrix}.$$

2.2 Reverse calibration for the space encoding range-finder

Blais [11] proposed the reverse calibration technique for a laser range-finder. In [11], the reverse of the range-finder calibration leads to a set of equations, which can be used to directly compute the location of a point in a range image, corresponding to an arbitrary point in 3-D space.

In this paper, the reverse calibration technique is adapted to the space encoding range-finder. The reverse calibration of the space encoding range-finder can be derived very easily from the definition of the camera and projector parameter matrices, *i. e.*, (1) and (2). Using those relations, the 2-D image coordinates or the space code of a data point can be reversely calculated from the 3-D world coordinates. However, if h_c is zero in (1), then the x_c and y_c cannot be calculated. It is also the case for h_p and x_p in (2). Thus, it is important to ensure that h_c and h_p never become zero in real range data.

It can be easily shown that the condition, $h_c = 0$, in (1) corresponds to the focal point of the imaging system [1]. Because a range view is a set of sampled points of the surface of an object which is placed far from the range-finder, this implies that h_c never becomes zero in real range data, neither does h_p for the same reason.

From these results, we can also find some useful relations. The equation of the lens plane can be written in the world coordinate system as

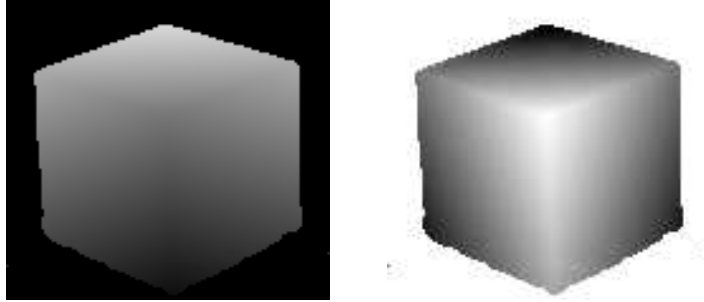
$$c_{31}x_w + c_{32}y_w + c_{33}z_w + c_{34} = 0. \quad (4)$$

From (4), the distance from this plane to a data point (x'_w, y'_w, z'_w) can be calculated from

$$d = \frac{|c_{31}x'_w + c_{32}y'_w + c_{33}z'_w + c_{34}|}{\sqrt{c_{31}^2 + c_{32}^2 + c_{33}^2}}. \quad (5)$$

Thus, using (5), the range data can be represented as a depth map. Figure 1 shows an example of the depth map representation. And from (4), we can also calculate the unit viewpoint vector of the range-finder, which will be utilized in the initial estimation step of the proposed registration algorithm, as

$$\mathbf{v} = \frac{1}{\sqrt{c_{31}^2 + c_{32}^2 + c_{33}^2}} \begin{pmatrix} c_{31} & c_{32} & c_{33} \end{pmatrix}^t. \quad (6)$$



(a) space code image

(b) depth map

Figure 1: An example of depth map representation

3 Registration of multiple range views

In this section, we present the proposed registration algorithm, which consists of the initial estimation and the iterative registration algorithms.

3.1 Initial estimation of the transformation parameters

Given a range view, the principal axes [10] are defined by three eigenvectors of the covariance matrix of three dimensional coordinates of the data points as shown in Figure 2.

The covariance matrix \mathbf{K} for given N data points $\mathbf{x}_i = (x_i \ y_i \ z_i)^t$, $i = 1, \dots, N$ is defined

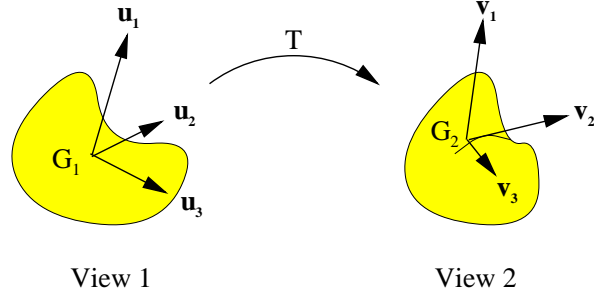


Figure 2: Initial estimation of the transformation parameters using principal axes

as

$$\mathbf{K} = \frac{1}{N} \sum_{i=1}^N \{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^t\}, \quad (7)$$

where the center of mass $\bar{\mathbf{x}} = (\mu_x \ \mu_y \ \mu_z)^t = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$.

Let V_1 and V_2 be two range views and \mathbf{p}_1 and \mathbf{p}_2 be a pair of corresponding points of V_1 and V_2 , respectively. Then, the relation between \mathbf{p}_1 and \mathbf{p}_2 is given by

$$\mathbf{p}_2 = \mathbf{R}\mathbf{p}_1 + \mathbf{t}, \quad (8)$$

where \mathbf{R} is a (3×3) matrix that represents the rotation, and \mathbf{t} is a three dimensional column vector that represents the translation, respectively. Using the eigenvectors of each range view, \mathbf{R} and \mathbf{t} can be estimated from

$$\hat{\mathbf{R}} = \mathbf{H}_2 \mathbf{H}_1^{-1}, \quad (9)$$

where \mathbf{H}_1 and \mathbf{H}_2 are 3×3 matrices made up of the normalized eigenvectors of the covariance matrices of V_1 and V_2 , respectively, and

$$\hat{\mathbf{t}} = \mathbf{g}_2 - \mathbf{R}\mathbf{g}_1, \quad (10)$$

where \mathbf{g}_1 and \mathbf{g}_2 are the center of mass of each view.

As shown in Figure 3, however, it is noted that the projected area of a surface segment is proportional to the original area, scaled by the cosine of the angle between the surface normal \mathbf{n} and the viewpoint \mathbf{v} . However, notice that this effect was not taken into account in [10]. As a result, if the axis of rotation is not corresponding to the viewpoint, the technique in [10] does not seem to work properly.

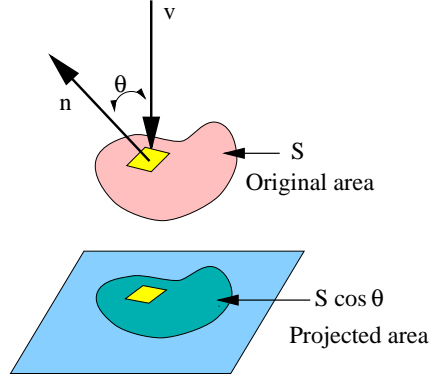


Figure 3: The projection effect

Therefore, in this paper, we attempt to modify the initial estimation algorithm to take into account the change in the projected area, due to different viewpoints, by including a weighting factor in the calculation of the covariance matrix. In our approach, the weighting factor for the i^{th} data point is

$$\begin{aligned}\omega_i &= \frac{1}{\cos \theta_i}, \\ &= \frac{\|\mathbf{v}\| \|\mathbf{n}_i\|}{\mathbf{v} \mathbf{n}_i^t},\end{aligned}\tag{11}$$

where \mathbf{v} represents the viewpoint vector of the range-finder, which can be obtained from (6), and \mathbf{n}_i represents the surface normal vector at the point. The surface normal vectors can be computed by the least squares (LS) method. However, the surface normal vectors of the points near step or crease edges cannot be computed correctly by the LS method. Note that erroneous surface normals yield incorrect weighting factors, degrading the overall

performance. However, it is well known that the edges are faithfully preserved by the least median of squares (LMedS) method. Thus, in our approach, for the points near edges, we compute the surface normal vectors by the LMedS method [2], instead of the LS method.

If the angle between the viewpoint vector and the surface normal vector approaches to 90° , then the ω_i becomes too large, which may result in a singular value of the weighting factor. Thus, a maximum value of the weighting factor is limited to 20 experimentally, in order to avoid such an occasion.

Then, the weighted covariance matrix $\hat{\mathbf{K}}$ is defined as

$$\hat{\mathbf{K}} = \frac{1}{W} \sum_{i=1}^N \left\{ \omega_i (\mathbf{x}_i - \tilde{\mathbf{x}})(\mathbf{x}_i - \tilde{\mathbf{x}})^t \right\}, \quad (12)$$

where $W = \sum_{i=1}^N \omega_i$, and $\tilde{\mathbf{x}} = (\tilde{\mu}_x \ \tilde{\mu}_y \ \tilde{\mu}_z)^t = \frac{1}{W} \sum_{i=1}^N \{\omega_i \mathbf{x}_i\}$. The block diagram for the proposed initial estimation is presented in Figure 4. The result of experiments demon-

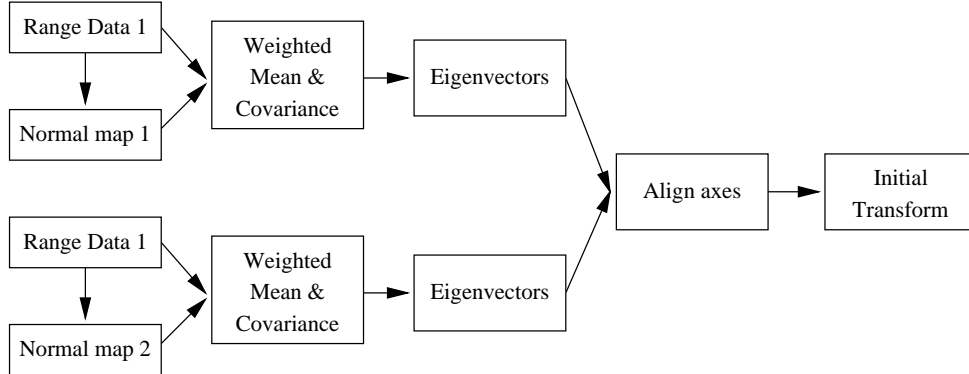


Figure 4: Block diagram of the initial estimation process

strates that the performance of the proposed algorithm is much improved, by such a simple modification. Some experimental results are presented in Section 4.

3.2 The iterative registration algorithm using the reverse calibration technique

The proposed registration algorithm may be considered a kind of iterative minimization technique. The registration error of each step of the iteration is modeled as

$$E_k = \frac{1}{N_k} \sum_{i=1}^{N_k} d(\mathbf{T}_k \mathbf{p}_i, C_k(\mathbf{p}_i)), \quad (13)$$

where \mathbf{p}_i is a point of one view, N_k is the number of \mathbf{p}_i 's in the k^{th} step, \mathbf{T}_k is the transformation matrix in the k^{th} step, $C_k(\cdot)$ is the correspondence function which maps \mathbf{p}_i to the corresponding point of the other view, and $d(\mathbf{x}, \mathbf{y})$ is the 3-D Euclidean distance between \mathbf{x} and \mathbf{y} , respectively. Indeed, the registration error of k^{th} step, E_k , is the average 3-D distance between the corresponding pair of points. The correspondence function C_k is defined by the reverse calibration technique. As depicted in Figure 5, the corresponding points can be found by the following two steps: first, transform a point with the transformation between the two views, namely \mathbf{T} , and then apply the reverse calibration technique, as described in the Section 2, to the transformed point.

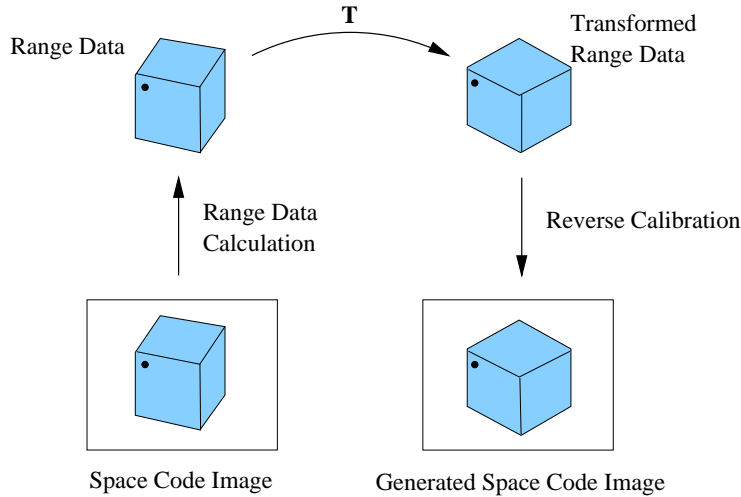


Figure 5: Reverse calibration in the registration process

However, it is believed that E_k is not appropriate to be used as the object function to minimize, because the transformation parameters could not be exact, until the iteration

converges. Thus, if we attempt to minimize E_k iteratively, the convergence may not be guaranteed. Therefore, we remodeled the object function as

$$\eta_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \min_{\mathbf{q}_j \in H\{C_k(\mathbf{p}_i)\}} d(\mathbf{T}_k \mathbf{p}_i, \mathbf{q}_j), \quad (14)$$

where $H\{\mathbf{x}\}$ implies the neighborhood of a point \mathbf{x} . In (14), \mathbf{q}_j is the point that makes the 3-D distance to $\mathbf{T}_k \mathbf{p}_i$ be minimized in the neighborhood of $C_k(\mathbf{p}_i)$. It is interesting to note that the size of neighborhood is to change, according to the shape of the object, since $C_k(\mathbf{p}_i)$ uses the reverse calibration, *i.e.*, the regenerated space code image. For example, the space code of smooth surface changes smoothly and its neighborhood is relatively smaller than that of rapidly changing surface.

Now, it is worthy to explain the physical meanings of (13) and (14) using Figure 6. The E_k implies the average distance between \mathbf{a} and \mathbf{b} , whereas η_k implies the distance between \mathbf{a} and \mathbf{c} . Note that \mathbf{c} is the nearest point to \mathbf{a} in the neighborhood of \mathbf{b} . Therefore, the object function (14) would accelerate the convergence speed, although the convergence to the optimal registration is not theoretically proved. Note that the ICP algorithm also has not guaranteed yet the theoretical optimality, although the convergence behavior of the ICP could be proved.

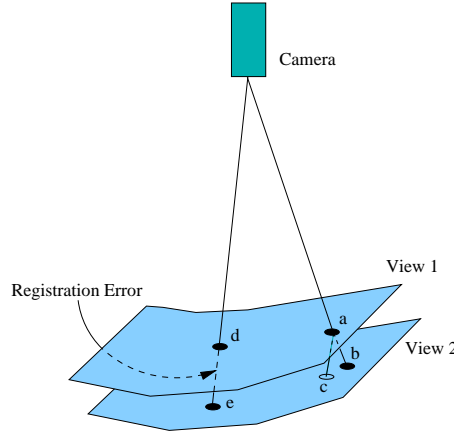


Figure 6: The registration error

In this context, the proposed algorithm may be considered as a modification of the

iterated closed point (ICP) algorithm [4]. However, it should be noted that the proposed algorithm is very fast, compared to the ICP algorithm, because it does not require any iteration to search the corresponding point pairs. Of course, there is a trade-off between the size of neighborhood and the speed, but with a carefully selected window size, we found that the proposed algorithm is still fast and efficient.

In each step of the iteration, the registration error η_k is evaluated and tested, if it satisfies the required accuracy, which is usually determined, by the limitation in the acquisition error of the range-finder. If η_k satisfies the required accuracy, the iteration is terminated. Otherwise, the object function η_k is evaluated and the transformation matrix \mathbf{T}_k is updated to minimize η_k . In homogeneous coordinates, the transformation \mathbf{T}_k can be represented as

$$\mathbf{T}_k = \begin{bmatrix} C_z C_y & C_z S_y S_x - S_z C_x & C_z S_y C_x + S_z S_x & t_x \\ S_z C_y & S_z S_y S_x + C_z C_x & S_z S_y C_x - C_z S_x & t_y \\ -S_y & C_y S_x & C_y C_x & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

where $C_i = \cos \theta_i$, and $S_i = \sin \theta_i$. Therefore, \mathbf{T}_k has 6 degrees of freedom, and has nonlinear terms that cause difficulties in obtaining an adequate solution. However, if the rotational angles are sufficiently small, then \mathbf{T}_k in (15) can be approximated by linear terms [5], that is, $C_i = 1$ and $S_i = \theta_i$, providing a solution to minimize η_k .

4 Experimental results

To evaluate the performance of the initial estimation algorithm of the transformation parameters and the iterative registration algorithm, many experiments are performed with real range views. Figure 7 shows three range views of a plastic cube, and Figure 8 shows two range views of a ceramic pot, respectively. In these images, the objects are rotated by 15 degrees.

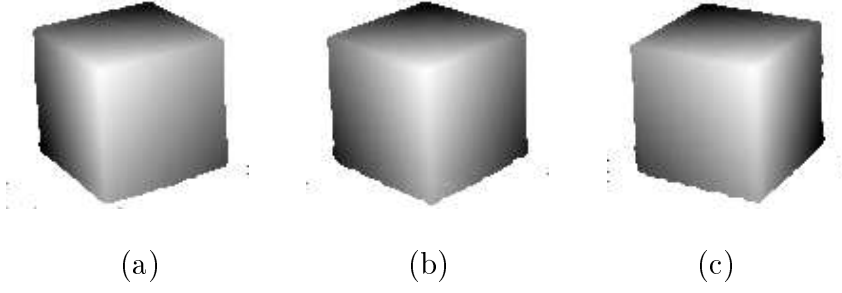


Figure 7: Range Images of a Cubic



Figure 8: Range Images of a Pot

In order to measure the error in the estimation of the rotation parameters, we use a measurement for the errors in rotation, which is independent of the actual rotation parameters. The relative error of the rotation matrix \mathbf{R} , E_R [10] is defined as

$$E_R = \frac{\|\hat{\mathbf{R}} - \mathbf{R}\|}{\|\mathbf{R}\|}, \quad (16)$$

where $\hat{\mathbf{R}}$ is the estimated \mathbf{R} .

The error in translation E_t [10] is defined as

$$E_t = \sqrt{(\hat{t}_x - t_x)^2 + (\hat{t}_y - t_y)^2 + (\hat{t}_z - t_z)^2}, \quad (17)$$

where \hat{t}_x , \hat{t}_y , and \hat{t}_z are the estimated t_x , t_y , and t_z , respectively.

The performance of the proposed initial estimation algorithm is compared with that of the algorithm in [10] in Table I and II, respectively. As shown in Table I and II, the proposed algorithm is found to be more accurate than the algorithm in [10]. The initial

values are very important in many cases, because they affect both the error performance and the speed of convergence. Note that E_R of the proposed algorithm is quite less than that in [10].

Table I: Results of initial estimation: Cube

Parameters	True value	(b) \rightarrow (a)		(c) \rightarrow (b)	
		[10]	Proposed	[10]	proposed
$\theta_x(deg)$	0	13.276	-0.858	12.792	3.946
$\theta_y(deg)$	0	13.498	-2.278	12.792	3.517
$\theta_z(deg)$	-15	-12.882	-9.781	-11.296	-11.212
$t_x(mm)$	-16.856	-25.647	-7.682	-22.794	-16.182
$t_y(mm)$	21.967	34.574	10.585	35.205	23.793
$t_z(mm)$	0	5.249	-0.662	4.434	0.103
E_R		0.267	0.082	0.265	0.092
E_t		16.241	14.634	15.171	1.949

Figure 9 provides an example of the iterative registration algorithm. In Figure 9, the range image of Figure 7 (c) is registered to of Figure 7 (b), by the proposed algorithm. As shown in Figure 9, the range image (b), which is the result of the estimated initial transformation, looks very close to the final result, demonstrating the validity of the proposed estimation algorithm.

Table II: Result of initial estimation: Pot

Parameters	True value	(b)→(a)	
		[10]	Proposed
$\theta_x(deg)$	0	-1.974	-2.637
$\theta_y(deg)$	0	6.637	2.421
$\theta_z(deg)$	-15	-18.475	-16.479
$t_x(mm)$	-16.856	-21.641	-18.451
$t_y(mm)$	21.967	20.767	19.026
$t_z(mm)$	0	7.021	3.456
E_R		0.110	0.055
E_t		8.581	4.810

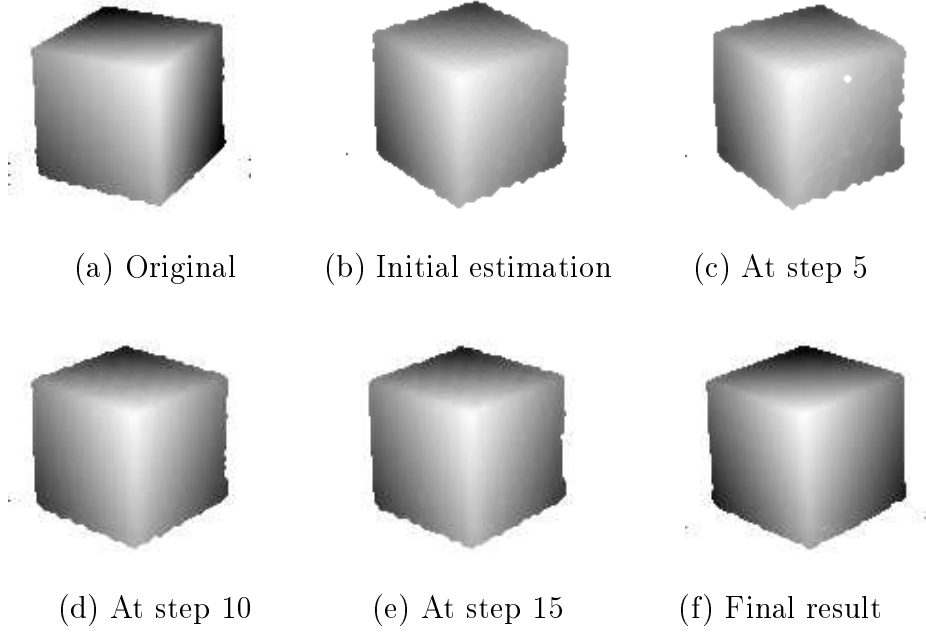


Figure 9: Results of the iteration process: (b) → (c) of Figure 7

The proposed registration algorithm is also tested with other views, and we find that the proposed algorithm successfully converges in 20 steps in every case. Thus, it is believed

that the initial estimation algorithm seems to accelerate the convergence. The iteration is terminated when the registration error is less than the limit of the acquisition error of the range-finder.

To compare the performance of the proposed algorithm quantitatively, we present the final registration error of the Cubic in Table III. The result shows that the proposed algorithm outperforms the Dorai’s algorithm and the converged parameters are closer to the true values than the initial estimation, provided in Table I.

Table III: Result of registration: Cubic (b)–(c)

Parameters	True value	(b)→(a)	(c)→(b)
$\theta_x(deg)$	0	-0.395	0.194
$\theta_y(deg)$	0	-0.010	0.517
$\theta_z(deg)$	-15	-15.013	-14.912
$t_x(mm)$	-16.856	-15.422	-16.182
$t_y(mm)$	21.967	23.382	20.598
$t_z(mm)$	0	-1.218	-0.103
E_R		0.006	0.008
E_t		2.354	1.529

Finally, Figure 10 presents an example of the input range data and the reconstructed shape of a toy bear. Nine range images, acquired from different viewpoints, are used to reconstruct the shape. The propagation of registration error, however, is not yet considered here. It is left for the future research on the integration of multiple range views.

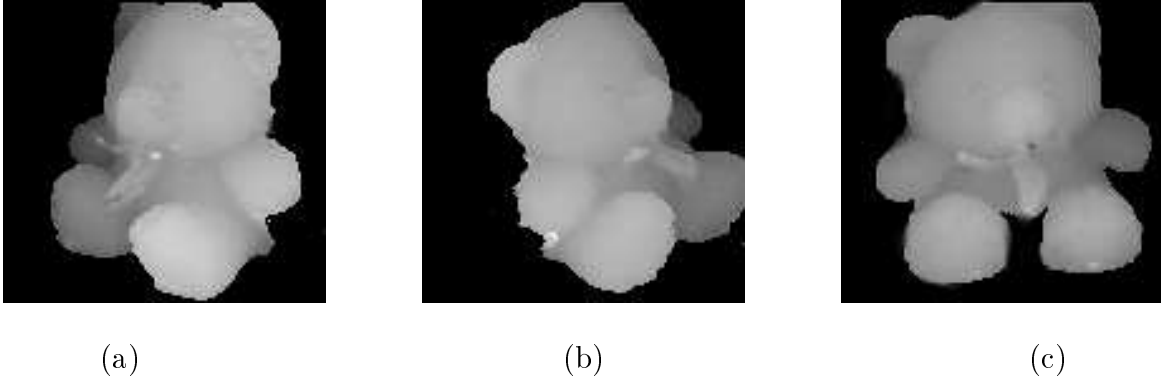


Figure 10: The registration of a toy bear: range data of (a) left side view, (b) right side view, (c) and the center view of the merged range data.

5 Conclusion

In this paper, we have proposed an algorithm to estimate the transformation parameters among multiple range views, by making use of the principal axes formed by the three eigenvectors of the weighted covariance matrix of the 3-D coordinates of data points. The proposed algorithm was carefully devised to take into account the projection effect, due to different viewpoints.

An iterative registration algorithm was also presented in this paper. To solve the correspondence problem, the reverse calibration technique was employed to the space encoding range-finder. The average 3-D distance between corresponding pair of points of two views was used for the error measure of the registration during the iterative minimization process.

One advantage of the proposed algorithm is that we can begin the iteration with a good estimation of the transformation parameters. The initial estimation algorithm provides a good starting point for the iteration, which accelerates the convergence speed.

Another advantage is that no iteration is required to obtain the corresponding point pairs during the iteration thanks to the reverse calibration technique which is adapted to

the space encoding range-finder.

References

- [1] M. D. Altschuler and K. Bae, “Robot Vision by Encoded Light Beams,” *Three-Dimensional Machine Vision*, edited by Takeo Kanade, pp. 97-149, Kluwer Academic Publishers, 1987.
- [2] P. J. Rousseeuw, “Least Median of Squares Regression,” *Journal of American Statistical Association*, vol. 79, no. 388, pp. 871-879, Dec. 1984.
- [3] S. Inokuchi, K. Sato, and F. Matsuda, “Range Imaging System Utilizing Nematic Liquid Crystal Mask,” *Proceedings of IEEE International Conference on Computer Vision*, pp. 657-661, London, June 1987.
- [4] P. J. Besl and N. D. McKay, “A Method of Registration of 3-D Shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239 - 255, Feb. 1992.
- [5] Y. Chen and G. Medioni, “Object Modeling by Registration of Multiple Range Images,” *Proceedings of IEEE Conference on Robotics and Automation*, pp. 2724-2729, Sacramento, Apr. 1991.
- [6] M. Soucy and D. Laurendeau, “Multi-resolution Surface Modeling from Multiple Range Views,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 348-353, Champaign, June 1992.
- [7] G. Truk and M. Levoy, “Zippered Polygon from Range Images,” *Proceedings of SIGGRAPH’94*, pp. 311-318, 1994.
- [8] R. Bergevin, D. Laurendeau, and D. Poussart, “Estimating the 3D Rigid Transformation between two Range Views of a Complex Object”, *Proceedings of IEEE*

- International Conference on Pattern Recognition*, vol. 1, pp. 478-482, Hague, Aug. 1992.
- [9] H. Gagnon, M. Soucy, R. Bergevin, D. Laurendeau, “Registration of Multiple Range Views for Automatic 3-D Model Building”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 581-586, Seattle, June 1994.
- [10] C. Dorai, J. Weng, and A. K. Jain, “Optimal Registration of Multiple Range Views,” *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 569-571, Los Alamitos, Oct. 1994.
- [11] G. Blais and M. D. Levine, “Registering Multiview Range Data to Create 3D Computer Objects,” *Technical Report of Centre for Intelligent Machines, McGill Univ.*, pp. 1-41, Nov. 1993.

List of Tables

I	Results of initial estimation: Cube	13
II	Result of initial estimation: Pot	14
III	Result of registration: Cubic (b)–(c)	15

List of Figures

1	An example of depth map representation	5
2	Initial estimation of the transformation parameters using principal axes . .	6
3	The projection effect	7
4	Block diagram of the initial estimation process	8
5	Reverse calibration in the registration process	9
6	The registration error	10
7	Range Images of a Cubic	12
8	Range Images of a Pot	12
9	Results of the iteration process: (b) \rightarrow (c) of Figure 7	14
10	The registration of a toy bear: range data of (a) left side view, (b) right side view, (c) and the center view of the merged range data.	16