The Twentieth International Symposium on Artificial Life and Robotics 2015 (AROB 20th 2015),
B-Con Plaza, Beppu, Japan, January 21-23, 2015

Point Cloud Matching Using Singular Value Decomposition

Shinji Oomori[1], Takeshi Nishida[1], and Shuichi Kurogi[1]

[1]Kyushu Institute of Technology, Fukuoka, Japan
(Tel: +81-93-884-3190, Fax: +81-93-861-1159)

[1]nishida@cntl.kyutech.ac.jp

Abstract: We propose a simple, fast 3D (three-dimensional) matching method that determines the best rotation matrix between non-corresponding PCs (point clouds) with no iterations. An estimated rotation matrix can be derived by the following two steps. 1) The SVD (singular value decomposition) is applied to a measured data matrix, and a database matrix is constructed from the PC datasets. 2) The inner product of each left singular vectors is used to produce the estimated rotation. Through experimentation, we demonstrate that the proposed method executes 3D PC matching with less than 4% of the computational time of the ICP (iterative closest point) algorithm with nearly identical accuracy.

Keywords: 3D matching, point cloud, singular value decomposition

1 INTRODUCTION

Recently, LRFs (laser range finders) and RGB-D (red, green, blue depth) cameras have been widely used for 3D (three-dimensional) object recognition. Furthermore, 3D object recognition and registration using PC (point cloud) data captured by these sensors are important in multiple industries. The ICP (iterative closest point) algorithm is a common approach used for the 3D matching of measured partial datasets between a target object and a PC's database [2, 3]. In the ICP algorithm, the correspondence of 3D points and a rigid body coordinate transformation, which results in the best rotation matrix, is iteratively searched for between the two 3D PC sets by repeating the closest point search when the correspondence is unknown.

Recently, we proposed a 3D matching method that determines the best rotation matrix between two non-corresponding PCs without iteration [4]. This method is simple and fast. An estimated rotation matrix can be derived using two steps of the SVD (singular value decomposition). However, this method has restrictions in its application to actual PC conditions. Therefore, we constructed a new PC matching method by combining the ICP algorithm and our 3D matching method in order to reduce the computational cost and overcome these restrictions. Through several simple experiments using actual PCs, we demonstrate that the proposed method executes 3D PC matching using less than 4% of the computational time of the ICP algorithm and with almost the same accuracy.

2 ESTIMATE OF TRANSFORMATION

2.1 Three-dimensional pose estimation

The rigid body coordinate transformation of two 3D PC sets is a relation, which can be is expressed as follows [5]. Given two 3D point sets, $A = \{\boldsymbol{a}_i \in \mathbb{R}^3 | i = 1, \cdots, N\}$ and $B = \{\boldsymbol{b}_i \in \mathbb{R}^3 | i = 1, \cdots, N\}$ and an element $\boldsymbol{a}_i = [a_{ix} \;\; a_{iy} \;\; a_{iz}]^T$, which is a point in Euclidean space, then a corresponding pair of two points $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ is related with translation $\boldsymbol{t} \in \mathbb{R}^3$ and rotation $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ by:

$$\boldsymbol{b}_i = \boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t}. \quad (1)$$

The estimation problem of $(\boldsymbol{R}, \; \boldsymbol{t})$ results in the following minimization problem:

$$\min_{(\boldsymbol{R},\boldsymbol{t}) \in SE(3)} \|\boldsymbol{B} - (\boldsymbol{R}\boldsymbol{A} + \boldsymbol{t}\boldsymbol{1})^T\|_F, \quad (2)$$

where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are elements in the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. The set $SE(3)$ is a Euclidean exercise group in 3D space, $\boldsymbol{1} = [1, 1, \cdots, 1]^{\boldsymbol{T}}$, $\boldsymbol{t} \in \mathbb{R}^N$, and $\|\cdot\|_F$ is a Frobenius norm. In general, because PC data include size information, a parameter for size does not need to be included in the cost function. Translation component $\boldsymbol{t}$ can be calculated from the center of gravity of the PCs. Therefore, eq. (2) can be rewritten so that it depends only on $\boldsymbol{R}$:

$$\min_{\boldsymbol{R} \in SO(3)} \|\boldsymbol{B}' - \boldsymbol{R}\boldsymbol{A}'\|_F, \quad (3)$$

in which $\boldsymbol{R} \in SO(3)$ is a 3D rotation group, and $\boldsymbol{A}'$ and $\boldsymbol{B}'$ are defined to be:

$$\boldsymbol{A}' = [\boldsymbol{a}'_1, \cdots, \boldsymbol{a}'_N] = \boldsymbol{A}\{\boldsymbol{I}_N - (1/N)\boldsymbol{1}\boldsymbol{1}^T\}, \quad (4)$$

$$\boldsymbol{B}' = [\boldsymbol{b}'_1, \cdots, \boldsymbol{b}'_N] = \boldsymbol{B}\{\boldsymbol{I}_N - (1/N)\boldsymbol{1}\boldsymbol{1}^T\}. \quad (5)$$

Here, $\boldsymbol{I}_N \in \mathbb{R}^{N\times N}$ is the unit matrix, and $\boldsymbol{a}'_i$ and $\boldsymbol{b}'_i$ are the 3D points that result when the translation components are subtracted from $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, respectively. In order to solve this estimation problem of a rigid body coordinate transformation, several methods have been proposed in the research field called Procrustes analysis and are widely used [6]. One solution to this problem uses SVD [7, 8]:

$$\boldsymbol{B}'\boldsymbol{A}'^{\boldsymbol{T}} \overset{\text{SVD}}{=} \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}} \in \mathbb{R}^{3\times3}, \quad (6)$$

in which $\boldsymbol{U} \in \mathbb{R}^{3\times3}$ is a left singular vectors, $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is a diagonal matrix containing singular values, and $\boldsymbol{V} \in \mathbb{R}^{3\times3}$ is a right singular vector matrix. Thus,

$$\boldsymbol{S} = \text{diag}(\boldsymbol{1} \;\; \boldsymbol{1} \;\; |\boldsymbol{V}\boldsymbol{U}^{\boldsymbol{T}}|) \in \mathbb{R}^{3\times3}. \quad (7)$$

Using eq. (6), we obtain $\boldsymbol{R}$ from eq. (3) as follows:

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^{\boldsymbol{T}}, \quad (8)$$

where $\boldsymbol{S}$ is a matrix used to avoid mirror image matching when measurement data include substantial measurement noise.

Because each element of $\boldsymbol{A}'$ and $\boldsymbol{B}'$ in eq. (6) needs to precisely correspond [4], the method is combined with the

ICP algorithm in order to determine a correspondence between each element in the 3D PCs.

2.2 Rigid body coordinate transformation estimation

Procrustes analysis provides another solution based on a different usage of the SVD [9, 10]. In previous research [4], it was shown that the proposed estimation method of rigid body coordinate transformation does not require an iterative procedure and that the algorithm is very simple and fast. The concrete procedure for this consists of two steps. The first step can be written as

$$\boldsymbol{A}' \overset{\text{SVD}}{=} \boldsymbol{U}_{\boldsymbol{A}'}\boldsymbol{\Sigma}_{\boldsymbol{A}'}\boldsymbol{V}_{\boldsymbol{A}'}^{T} \tag{9}$$

$$\boldsymbol{B}' \overset{\text{SVD}}{=} \boldsymbol{U}_{\boldsymbol{B}'}\boldsymbol{\Sigma}_{\boldsymbol{B}'}\boldsymbol{V}_{\boldsymbol{B}'}^{T}. \tag{10}$$

In the next step, the left singular vectors $\boldsymbol{U}_{\boldsymbol{A}'} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{U}_{\boldsymbol{B}'} \in \mathbb{R}^{3\times 3}$ are used to determine the rotation matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{R} = \boldsymbol{U}_{\boldsymbol{B}'}{\boldsymbol{U}_{\boldsymbol{A}'}}^{\boldsymbol{T}}, \tag{11}$$

where $\boldsymbol{A}' \in \mathbb{R}^{3\times N}$ is a database pattern matrix and $\boldsymbol{B}' \in \mathbb{R}^{3\times M}$ is a measurement pattern matrix.

2.3 Features and constraints

The elements of these matrices do not need to be in corresponding order. In addition, the number of the elements that comprise these matrices may differ, i.e. the condition $N \neq M$ is acceptable. Moreover, all the SVD results of the database patterns, $\boldsymbol{U}_{\boldsymbol{A}'}$, can be calculated in advance. Therefore, this method can be quickly executed using an online procedure. However, it also has the following constraints.

1. *Two PCs should express the same 3D region.*
2. *Elements of the 3D PCs should be uniformly distributed.*

This method is applicable to a set of PCs that express different regions. However, when the occupying space volumes of two PCs are unequal, the matching cost function cannot be minimized. Moreover, when a PC is missing some elements because of a lack of measurement, for example, the pattern matching decision cannot be appropriately performed. Thus, a pre-processing procedure must be employed to uniformly extract the 3D PC in order to apply this method. Therefore, we combine this method and the ICP algorithm.

2.4 Types and accuracy of SVD algorithms

In this research, we have adopted a program provided by the GSL (GNU science library) [11] as an actual SVD algorithm. The appropriate choice of an SVD algorithm or program is important; some SVD algorithms possess different conditions or calculation accuracy. For example, the Golub-Reinsch algorithm, the modified Golub-Reinsch algorithm, and the one-side Jacobi orthogonalization method have been implemented in the GSL. We used the modified Golub-Reinsch algorithm as an SVD algorithm in our experiment. This algorithm is suitable for the execution of our method because its computational time is substantially faster when the number of columns differs from the number of rows in the target matrix, i.e. when $N, M \gg 3$.

3 PC MATCHING ALGORITHM USING SVD

3.1 Summary of ICP algorithm

The ICP algorithm is a method that matches partial 3D shape data to a larger model's shape when the points' correspondence is unknown [3]. The following processes (Tab. 1) are repeated in order to check the matching ratio between the test data $C = \{\boldsymbol{c}_i\}_{i=1}^{N} \in \mathbb{R}^3$ and a given database $D = \{\boldsymbol{d}_i\}_{i=1}^{M} \in \mathbb{R}^3$. The point correspondences are not given in the initial condition. In practice, the ICP algorithm needs several other procedures, such as the $\boldsymbol{d}_i$ sampling method, the closest point searching method, and the pre-processing method for non-correspondence datasets. It is important to note that the procedure [ICP step 1] and [ICP step 2] used to estimate $\boldsymbol{R}$ can be replaced with the previously mentioned method, that of eq. (9) to eq. (11). In the ICP algorithm, the value of ϵ converges with innumerable repetitions.

Table 1. ICP algorithm

[ICP given]
Construct a test pattern $\boldsymbol{C} = [\boldsymbol{c}_1 \ \cdots \ \boldsymbol{c}_N]$ from the measurements.

[ICP step 1]
Construct $\boldsymbol{D} = [\boldsymbol{d}_1 \ \cdots \ \boldsymbol{d}_N]$ by extracting the points $\boldsymbol{d}_i$ closest to $\boldsymbol{c}_i$ $(\forall i)$ from D.

[ICP step 2]
Estimate rotation matrix $\boldsymbol{R}$ from eq. (3) to eq. (8).

[ICP step 3]
Construct rotated pattern $\bar{C} = \boldsymbol{R}^T\boldsymbol{C} \triangleq [\bar{\boldsymbol{c}}_1 \ \cdots \ \bar{\boldsymbol{c}}_i \ \cdots \ \bar{\boldsymbol{c}}_N]$, and arrange the points in corresponding order. Namely, construct $\bar{\boldsymbol{D}} = [\bar{\boldsymbol{d}}_1 \ \cdots \ \bar{\boldsymbol{d}}_N]$ using the extracted closest points $\bar{\boldsymbol{d}}_i$ obtained by searching $\bar{\boldsymbol{c}}_i$ $(\forall i)$.

[ICP step 4]
If $\epsilon = \|\bar{\boldsymbol{D}} - \bar{\boldsymbol{C}}\|_F$ is greater than the threshold value, return to [ICP step 1]. If ϵ is less than threshold value, exit the process.

3.2 Construction of database

In this section, the database construction using the proposed method is described. First, database patterns are produced as follows. The j th database pattern matrix $\boldsymbol{M}^{(j)} = [\boldsymbol{m}_1^{(j)} \ \cdots \ \boldsymbol{m}_L^{(j)}]$ is constructed by arranging the target object's 3D points that are measured from a single viewpoint. Here, $j = 1, \cdots, J$ is the database pattern's index, and the number of elements L must exceed N. Moreover, their centers of gravity are then matched to the origin. Next, the SVD is applied to the database pattern matrices:

$$\boldsymbol{M}^{(j)} \overset{\text{SVD}}{=} \boldsymbol{U}_{\boldsymbol{M}^{(j)}} \boldsymbol{\Sigma}_{\boldsymbol{M}^{(j)}} \boldsymbol{V}_{\boldsymbol{M}^{(j)}}^{\boldsymbol{T}}, \tag{12}$$

and $\boldsymbol{U}_{\boldsymbol{M}^{(j)}}$ is stored.

3.3 Proposed PC matching algorithm

We propose the PC matching algorithm shown in Tab. 2.

Table 2. Proposed PC matching algorithm

[Prop. given]
Construct the measurement pattern matrix

$$\boldsymbol{C} = [\boldsymbol{c}_1 \ \cdots \ \boldsymbol{c}_N]$$

from the measurement of the PC. At this time, the center of gravity is matched to the origin by translation.

[Prop. step 1]
Apply the SVD to the measurement pattern matrix,

$$\boldsymbol{C} \overset{\text{SVD}}{=} \boldsymbol{U}_{\boldsymbol{C}} \boldsymbol{\Sigma}_{\boldsymbol{C}} \boldsymbol{V}_{\boldsymbol{C}}^{T},$$

and estimate the rotation matrix as follows:

$$\boldsymbol{R}^{(j)} = \boldsymbol{U}_{\boldsymbol{C}} \boldsymbol{U}_{\boldsymbol{M}^{(j)}}^{T}.$$

[Prop. step 2]
Construct a rotated pattern,

$$\overline{\boldsymbol{C}}^{(j)} = \boldsymbol{R}^{(j)T} \boldsymbol{C} \triangleq [\bar{\boldsymbol{c}}_1 \ \cdots \ \bar{\boldsymbol{c}}_i \ \cdots \bar{\boldsymbol{c}}_N],$$

and construct $\overline{\boldsymbol{M}}^{(j)} = \left[\bar{m}_1^{(j)} \ \cdots \ \bar{m}_N^{(j)}\right]$ in order according to its closest point $\bar{\boldsymbol{c}}_i$ that was extracted from $\boldsymbol{M}^{(j)}$.

[Prop. step 3]
Calculate the matching error as follows:

$$\epsilon^{(j)} = \left\| \overline{\boldsymbol{M}}^{(j)} - \overline{\boldsymbol{C}}^{(j)} \right\|_F \ (\forall j). \tag{13}$$

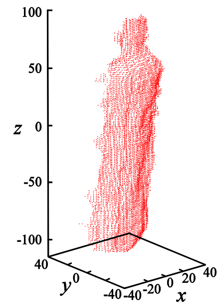
The index of the minimum $\epsilon^{(j)}$ is defined to be the matching pattern number.

These procedures, **[ICP step 1]** and **[ICP step 2]**, correspond to **[Prop. step 1]** for the estimation of the best rotation matrix, but the latter does not require multiple iterations in order to converge to the evaluation value. The procedures **[Prop. step 1]** and **[Prop. step 2]** can be quickly executed, and the matching procedures for all of the database patterns can be simultaneously performed. Moreover, **[ICP step 1]** corresponds to **[Prop. step 3]**, but the former requires that calculation ingenuity be applied to the point correspondence search if the distance between the PCs is large in initial iteration. However, **[Prop. step 3]** immediately estimates the best rotation matrix because it is executed on the simple closest point search. In addition, by providing a threshold of $\epsilon^{(j)}$, the proposed method can determine a reject pattern.

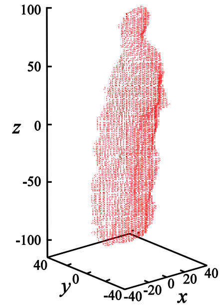
4 EXPERIMENTS

4.1 Conditions

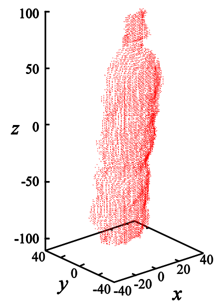
The target objects' images and the database PCs constructed using 3D measurement are shown in Fig. 1.



(a) Image of type A (b) PC of type A



(c) Image of type B (d) PC of type B



(e) Image of type C (f) PC of type C

Fig. 1 Images and PCs of plastic bottles for databases

These patterns are constructed using the measurement of three types of plastic bottles and an Xtion pro live sensor. The computer is configured as follows: the CPU is an Intel core i7 (3.4GHz), with 8 GB of RAM and an Ubuntu 12.04 (32bit) operating system.

A measurement pattern constructed to be a recognition target is shown in Fig. 2. This pattern was constructed by rotating the type A plastic bottle and measuring the distances that were farther than those of the database patterns. It was then rotated 90 degrees. The numbers of points used for these patterns are shown in Table 3. It is clear that the number used for the test pattern is significantly less than those used for the database patterns.

4.2 Results

Given these conditions, the pattern matching experiments are executed using both the proposed method and the ICP algorithm in order to create a comparison. The results are shown in Figs. 3 and 4. In both cases, the number of points for the test pattern was $n_s = 2613$. In

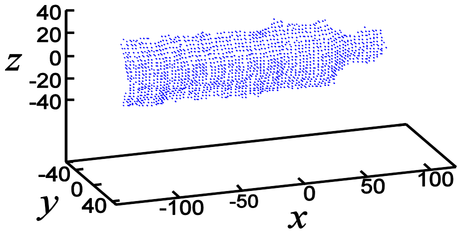


Fig. 2 PC of test pattern (type A, 90-degree rotation)

Table 3 Number of points in each dataset

Name of 3D dataset	Number of points
Plastic bottle A (database)	10,189
Plastic bottle B (database)	9,476
Plastic bottle C (database)	9,896
Plastic bottle A (test pattern)	2,613

these experiments, an appropriate estimation of the best rotation matrix was not determined using the ICP algorithm because the angle of the best rotation matrix, which associates the database and test patterns, was too large. Therefore, we applied the ICP algorithm to the test pattern after a 90-degree rotation around the y-axis. The results shown in Figs. 3 and 4 were obtained under these conditions. The ICP algorithm was repeated until the evaluation value converged, and the calculation time was measured for the value $\epsilon^{(j)}/3n$ to fall below 0.700 or for the value to converge. These results show that the matching errors produced by the proposed method and the ICP algorithm were minimal in the case of type A's pattern, meaning the pattern matching had been correctly executed. Moreover, despite the fact that the ICP algorithm was applied after adjusting the initial position, the calculation times achieved by the proposed method were less than 4% of those attained by the ICP algorithm.

5 CONCLUSION

We proposed a method for improving the PC matching algorithm using an estimation method based on SVD [4]. Upon comparison of our method with the ICP algorithm, it was shown that our method's calculation time was less than 4% of the proposed method's time because the proposed method requires the extraction of target objects from the environmental 3D PC in practical applications. Moreover, our method could prove even faster if it is applied after PC feature extraction.

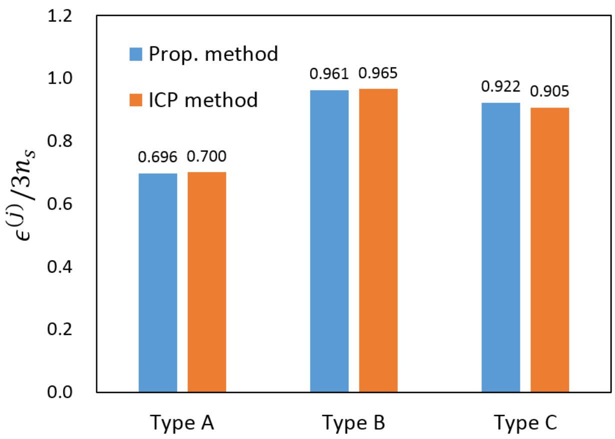


Fig. 3 Evaluation comparison for matching

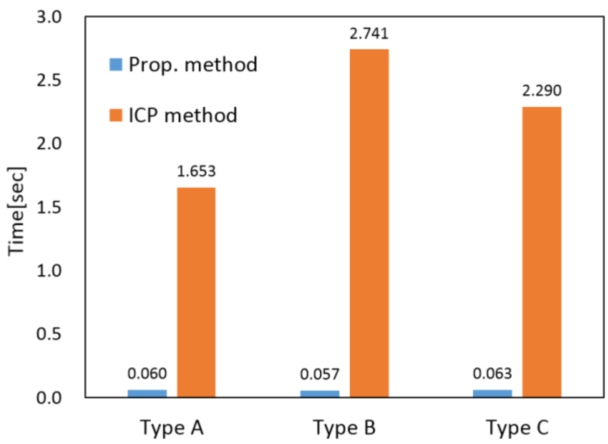


Fig. 4 Calculation time comparison for matching

REFERENCES

[1] Yongjin, J., Kurazume, Iwashita, Y., Hasegawa, T., (2013), Global Localization for Mobile Robot using Large-scale 3D Environmental Map and RGB-D Camera, JRSJ, 31(9):896–906.

[2] Besl, P. J., Mckay, N. D., (1992), A Method for Registration of 3-D Shapes, IEEE Trans. Pattern Analysis and Machine Intelligence, 14(2):239–256.

[3] Takeshi, M., (2009), ICP Algorithm, CVIM, 186(23):1–8.

[4] Nishida, T., Kurogi, S., (2013), Rotation Matrix Estimation Between Non-Corresponding 3-D Point Sets, JRSJ, 31(6):624–627.

[5] Tamaki, T., (2009), Pose Estimation and Rotation Matrices, SIS, 109(203):59–64.

[6] Akca, M. D., (2003), Generalized Procrustes Analysis and Its Applications in Photogrammetry, Technical report, ETH, Swiss Federal Institute of Technology Zurich, Institute of Geodesy and Photogrammetry.

[7] Kanatani, K., (1994), Analysis of 3-D rotation fitting, IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(5):1–25.

[8] Ueyama, S., (1991), Least-Squares Estimation of Transformation Parameter Between Two Point Patterns, IEEE Trans. on Pattern Analysis and Machine Intelligence, 13(4):376–380.

[9] Schöneman, P. H., (1966), A Generalized Solution of the Orthogonal Procrusters Problem, Psychometrika, 31(1):1–10.

[10] Higham, N., (1993), Matrix Procrustes problems, The University of Manchester.

[11] GNU Science Library, http://www.gnu.org/software/gsl/.