ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

3D registration process of ICP using linear constraint

Zihuan Ding

University of California, Davis, Davis, 95616, United States

Abstract. This paper implements an ICP registration algorithm that aligns mul- tiple 3D scans into a common coordinate system. To better visualize the result, we calculate the vertex valence using a triangle mesh and visualize the 3D object in color based on the vertex valence. The initial job of implementing the ICP algorithm is to evenly sub-sample the 3D scans to in- crease the calculation speed, compute the correspondence, and remove the bad corresponding pairs based on distance threshold and normal compat- ibility. Subsequently, perform the registration process with point-to-point and point-to-surface registration, which transforms the ICP equation by solving a matrix equation with a linear constraint. The result shows that both methods recreate a complete 3D object, but the point-to-surface method converges much faster.

Keywords: Iterative closest point (ICP),3D registration, linear transformation, sub- sampling, point-to-point constraint, point-to-plane constrai.

1. Introduction

The application of geometric processing in daily life is becoming increasingly widespread. Visual effects, augmented reality, computer animation, and other related fields are some of their applications. All of these applications involve the use of methods that are part of geometric modeling. This term refers to the methods and algorithms that are used to represent and process geometric objects. Reconstruction will be one of the most significant aspects of geometric processing, particularly for processing three-dimensional objects. In real-world circumstances, the reconstruction process typically involves scanning 3D ob- jects. To make better use of them, we need to do scan matching or point cloud registration to align those 3D scans into a full model. However, the conven- tional ICP technique employs a non-linear method, which requires a significant proportion of the resources available for computing. Consequently, the purpose of this work is to analyze a linear transformation of the ICP algorithm that incorporates point-to-point as well as point-to-plane constraints. The following sections will evaluate the related work in this area, elaborate the methods, and analyze the results.

2. Related work

One of the most famous algorithms in the 3D registration process is the ICP algorithm introduced in [1]. It is widely used in matching each data point on a 3D scan to the closest corresponding point on the other scan. The ICP algorithm is so important that many other algorithms are constructed on the original ICP algorithm, and Rusinkiewicz and Levoy [2] conducted a survey discussing those variants of the ICP algorithm. Similar algorithms also find the transformation between the different views, such as Chen and Medioni's algorithm [3].

The ICP algorithm and its variations have remained quite popular in recent years. For instance, the voxel lattice is used to re-sample the point cloud data, and the k-d tree is employed to improve the computation of the normal vector to address the issue that the traditional iterative closest point technique converges slowly [4]. [5] expands the NDT registration workflow by learning and predicting per-point semantic labels with PointNet, a deep neural network for segmenting and classifying point clouds. Pose interpreter networks for 6-DoF object pose estimation are introduced in [6]. An innovative method for model-based 6D posture refinement in color data is presented in [7]. The system provides 3D pose estimate accuracy that is above state-of-the-art and recovers dense 3D hand forms, according to experiments employing three RGB-based benchmarks. Each of the aforementioned technical elements significantly increases accuracy [8]. Deepest Closest Point

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

(DCP), a learning-based approach proposed by [9], draws inspiration from current developments in computer vision and natural language processing to handle local optima and other challenges in the ICP pipeline. The first quick and verifiable approach for registering two sets of 3D points in the presence of several outlier correspondences is proposed by [10].

The basic ICP algorithm proposed by Besl and McKay [1] includes the pro- cess of matching each data point to its closest point in the other data scans and constructing an error function of the corresponding pairs. The error function can be considered a point-to-point error metric that calculates the sum of the squared distance between the data pairs and tries to find the minimum of this value. The process is iterated until it converges and stops changing. Chen and Medioni [3], however, propose a method utilizing a point-to-plane error metric that minimizes the sum of the squared distance between a point and the corre- sponding tangent plane of its data pair. This error metric can be solved using a nonlinear least squares method such as the Levenberg-Marquardt method [11], but it is hard to calculate compared to a linear model. Although the original point-to-point ICP algorithm is well known, it is observed that a point-to-plane ICP has a much faster convergence rate [2].

Considering the complexity of solving both point-to-point ICP and point- to-plane ICP in a nonlinear least squares method, this report will focus on proposing a way to convert the nonlinear error metric in solving the trigono- metric function into a rotational matrix. The central idea is proposed in [2]: if the rotation between two tangent planes is small, we can approximate the

nonlinear least-squares optimization problem by changing sin θ to θ and cos θ to

1 in the rotation matrix, thus getting rid of the nonlinear trigonometric func- tion, and rewrite the equation into a general form of Ax = b where A shows the overall transformation by combining translation vector and rotational matrix.

3. Method

3.1 Basic ICP algorithm

The typical ICP algorithm for 3D object data scans follows a procedure:

Step1: Select some random points on a 3D data scan.

Step2: Match each data point to the closest point on the other scan, using a data structure such as a k-d tree.

Step3: Reject corresponding data pairs with a distance greater than a large threshold or the normal compatibility does not satisfy.

Step4: Construct error function: $E(R, T) = \Sigma_{n\ i=1}|(|Rp_i + T - q_i|)|^2$ and minimize the equation where p and q represent the data points of two different sources, and R and t represent the rotational and translation matrix[1].

The translation matrix T and rotational matrix R assuming matrix T takes in parameters (tx,ty,tz) and matrix R takes in parameters (α,β,γ) is shown in equation 1 and 2, rij with row number i and column number j is calculated by

a range of trigonometric functions illustrated in Eq.3:

$$T(t_x, t_y, t_z) = \begin{matrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{matrix} \quad (1)$$

$$R(\alpha, \beta, \gamma) = R_z(\gamma) * R_y(\beta) * R_x \qquad (\alpha) = \begin{matrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (2)$$

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

$$r_{11} = \cos\gamma\cos\beta$$
$$r_{12} = -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha\, r_{13} = \sin\gamma$$
$$\sin\alpha + \cos\gamma\sin\beta\cos\alpha\, r_{21} = \sin\gamma\cos\beta$$
$$r_{22} = \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha\, r_{23} = -\cos$$
$$\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha\, r_{31} = -\sin\beta \quad (3)$$
$$r_{32} = \cos\beta\sin\alpha\, r_{33} =$$
$$\cos\beta\cos\alpha$$

As shown in Eq.3, we need to figure out parameters α, β, γ, which are related to trigonometric functions, so it can not directly apply linear least-square cal- culation, which is generally more efficient than non-linear calculation [12]. As a result, our proposed method is designed to modify the procedure to utilize a linear transformation to eliminate the nonlinearity of the sum-of-square cal- culation for the error function, simplifying the calculation. The modified ICP algorithm now consists of the following:

Step1: Selecting data points from 3D source object

Step2: Matching data points to the target point or tangent plane

Step3: Weighting the correspondences

Step4: Rejecting bad pairs

Step5: Compute error metric

Step6: Minimize error metric

The detailed explanation of each step will be expanded in the following sections 4.2, 4.3, and 4.4.

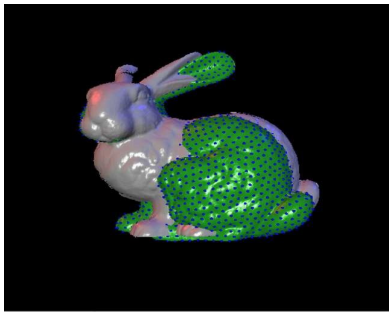

Figure 1: Result of Uniform sub-sampling

3.2 Uniform Sub-sampling

We choose to apply uniform subsampling to efficiently choose the data points in order to decrease the quantity of work and hasten the registration procedure. ”Uniform subsampling” refers to sampling data points at uniform intervals. The work is completed by repeatedly going through all the data points and discarding any candidate data points that are larger than a predetermined radius or threshold of all other points that have already been sampled. We use a k-d tree data structure that is easy to find using the ANN library. Then we used ANN to locate the closest neighbor, establish communication, and ultimately remove the border correspondence. Figure 1 displays the sampling results in graphic form. The blue dots represent the data points after uniform sampling, while the green portion represents the newly arriving data scan.

However, they might be meaningless if two data pairs are extremely distant from one another or come from two tangent planes with sharp angles. Exam- ining those faulty data pairs would be pointless because it would prolong the registration process or potentially lead to inaccurate findings. We should prune correspondences based on these two scenarios—significant separation and poor normal compatibility—to consider both. We set up a threshold, and for each data pair, if the distance exceeds the threshold, we exclude the data pairs from the correspondence list to prune correspondence based on the distance. The threshold in our experiment is set at five times the typical distance between the vertices of the triangular meshes of the data points.

The final result can be viewed in Figure 1.

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

3.3 Point-to-point optimization

We must resolve the issue of translating the error function indicated in sec- tion 4.1 into solving Ax = b in linear constraints after making the necessary preparations to change the data for accurate and quick registration. The most crucial step in such a change is using a linear transformation. To eliminate the trigonometry routines and linearize the rotating matrix parameters. We can approach sin x = x and cos x = 1 if we assume that minimal angles result in a

minimal rotation. The assumption must be accurate because we already reject undesirable pairs during the sub-sampling procedure to increase accuracy and prune correspondence. A new rotation matrix R is acquired following the linear transformation, as illustrated in Eq.4.

Multiplying the T translation matrix with the new R matrix creates a new matrix M that represents the complete transformation:

$$R = \begin{matrix} 1 & -\gamma & \beta & 0 \\ \gamma & 1 & -\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \tag{4}$$

$$M = \begin{matrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{matrix} \tag{5}$$

This M matrix shows how a specific point on the source scan would transform

to the target scan, so we find $M_{opt} = \arg\min_M \sum_n (M * p_i - q_i)^2$. Afterthe matrix multiplication and transforming it into the form Ax = b with x = (α β γ t_x t_y t_z)T , we would obtain a n ∗ 18 matrix where n represents the size of the input source points. The parameters of matrix A are shown in Eq.6:

The corresponding b matrix is illustrated in Eq.7:

$$\begin{aligned} A[i][1] &= 0 \\ A[i][2] &= p_{iz} \\ A[i][3] &= -p_{iy} \\ A[i][4] &= 1 \\ A[i][5] &= 0 \\ A[i][6] &= 0 \\ A[i][7] &= -p_{iz} \\ A[i][8] &= 0 \\ A[i][9] &= p_{ix} \\ A[i][10] &= 0 \\ A[i][11] &= 1 \\ A[i][12] &= 0 \\ A[i][13] &= p_{iy} \\ A[i][14] &= -p_{ix} \\ A[i][15] &= 0 \\ A[i][16] &= 0 \\ A[i][17] &= 0 \\ A[i][18] &= 1 \end{aligned} \tag{6}$$

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

$$\begin{aligned} b[i][1] &= p_{ix} - q_{ix} \\ b[i][2] &= p_{iy} - q_{iy} \\ b[i][3] &= p_{iz} - q_{iz} \end{aligned} \tag{7}$$

With a linear equation Ax = b given the parameters above, we can solve it by applying SVD (singular value decomposition) [11]. However, we can likewise use Cholesky decomposition instead since A is a positive-definite matrix and implementing Cholesky decomposition is easier. Moreover, Cholesky only takes O(n3), which is faster than SVD(O(mn2)) when m > n.

3.4 Point-to-plane optimization

Point-to-point optimization generally converges much slower since it does not account for the motion along the surface. Therefore, a second optimization strategy of optimizing the distance between each source point to the tangential plane of the corresponding target point should be evaluated. The corresponding tangential surface of a vertex pi is defined by the correspondence qi and the normal ni. We then need to solve the equation $E(R, T) = \Sigma_{n\ i=1} |(|Rp_i+T-q_i|)|^2$ instead. Similar to point-to-point optimization, we need to solve an equation Ax = b in linear constraint, and the general theory of converting the rotation matrix and translation matrix is the same. After the linear approximation, we should obtain matrix A:

$$\begin{aligned} A[i][1] &= n_{iz}p_{iy} - n_{iy}p_{iz} \\ A[i][2] &= n_{ix}p_{iz} - n_{iz}p_{ix} \\ A[i][3] &= n_{iy}p_{ix} - n_{ix}p_{iy} \\ A[i][4] &= n_{ix} \\ A[i][5] &= n_{iy} \\ A[i][6] &= n_{iz} \end{aligned} \tag{8}$$

And the corresponding vector b is shown in Eq.9:

$$b[i] = n_{ix}q_{ix} + n_{iy}q_{iy} + n_{iz}q_{iz} - n_{ix}p_{ix} - n_{iy}p_{iy} - n_{iz}p_{iz} \tag{9}$$

We still use Cholesky decomposition to solve the equation since A is still a positive-definite matrix.

Table 1: Average running time analysis

Num of correspondence	point-to-point	point-to-plane	traditional ICP
656	1.039	1.025	1.267

Table 2: Average steps to converge between scans

Steps for point-to-point	Steps for point-to-plane
23	9

a bunny are shown in Figure 2 and Figure 3. The front view and the back view come together wonderfully.

Figure 2: front view

Figure 3: back view

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023

3.5 running time

The point-to-point and point-to-plane methods outperform the original ICP al- gorithm, as shown in Table 1. Even though the difference between our method and traditional ICP is only about 0.25 seconds, this slight difference can be explained by the fact that we only have a few data points. Only 635 corre- spondences are present, according to the table. However, it already represents a significant improvement when expressed in percentage. The point-to-plane method also outperforms the point-to-point method since it takes less average running time in each step. Additionally, according to Table 2, we verified the hypothesis that point-to-surface registration takes fewer steps to converge than other types of registration during the experiment, mainly when the initial posi- tion of the new scans is far from the present scan.

4. Conclusion

In conclusion, this article successfully changes the nonlinear classical trigono- metric calculation into a linear matrix calculation. We proposed two differ- ent methods: point-to-point transformation and point-to-plane transformation.

The results show that point-to-plane transformation converges substantially faster than point-to-point transformation and takes less time to execute each step on average. The shorter running time makes point-to-plane transformation an appealing alternative for situations where speed and efficiency are critical. However, due to hardware and dataset limitations, we only conduct tests on data scans of roughly 70,000 data points. Although our data is insufficient, a distinct difference in results between our approach and the classic ICP algo- rithm is convincing enough to demonstrate that our approach is faster since the difference will become more apparent as the number of data points increases. Furthermore, because our method assumes that the corresponding points of the two data scans have a modest angle difference, we might investigate how to retain linearity even with huge-angle difference data scans in the future.

References

[1] Paul J. Besl, and Neil D. McKay. A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 14(2), pp. 239– 256, 1992.

[2] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algo- rithm. Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM), pp. 145–152, 2001.

[3] Yang Chen, and Gerard Medioni. Object Modeling by Registration of Mul- tiple Range Images. International Journal of Image and Vision Computing, 10(3), pp. 145–155, 1992.

[4] Wei Guan; WenTao Li; Yan Ren; ”Point Cloud Registration Based on Improved ICP Algorithm”, 2018 CHINESE CONTROL AND DECISION CONFERENCE (CCDC), 2018.

[5] Anestis Zaganidis; Li Sun; Tom Duckett; Grzegorz Cielniak; ”Integrating Deep Semantic Segmentation Into 3-D Point Cloud Registration”, IEEE ROBOTICS AND AUTOMATION LETTERS, 2018.

[6] Jimmy Wu; Bolei Zhou; Rebecca Russell; Vincent Kee; Syler Wagner; Mitchell Hebert; Antonio Torralba; David M. S. Johnson; ”Real-Time Ob- ject Pose Estimation With Pose Interpreter Networks”, ARXIV, 2018.

[7] Fabian Manhardt; Wadim Kehl; Nassir Navab; Federico Tombari; ”Deep Model-Based 6D Pose Refinement In RGB”, ARXIV, 2018.

[8] Seungryul Baek; Kwang In Kim; Tae-Kyun Kim; ”Pushing The Envelope For RGB-based Dense 3D Hand Pose Estimation Via Neural Rendering”, ARXIV, 2019.

[9] Yue Wang; Justin M. Solomon; ”Deep Closest Point: Learning Represen- tations for Point Cloud Registration”, ICCV, 2019

[10] Heng Yang; Jingnan Shi; Luca Carlone; ”TEASER: Fast And Certifiable Point Cloud Registration”, ARXIV, 2020.

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.542.2023
[11] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing, Second Edition, Cambridge University Press, 1992.

[12] Arun, K. Somani, Thomas S. Huang, and Steven D. Blostein. ”Least- squares fitting of two 3-D point sets.” IEEE Transactions on pattern anal- ysis and machine intelligence 5 (1987): 698-700.