ICP Algorithm for 3D Surface Registration

Zhipeng Hu

Beijing institute of technology, Beijing, 100081, China

Abstract. With the development of 3D model applications, we need to deal with more and more mesh data, such as 3D surface registration. In this paper, based on the ICP algorithm, we use approximate processing in the calculation, which greatly simplifies the process and improves the efficiency of the operation. Experiments show that the method can achieve good registration of 3d surfaces.

Keywords: Computer Vision; Surface Registration; ICP.

1. Introduction

With the development of computer technology, the technique of generating 3D digital models from physical models has been more and more widely used. In this application process, we often need to collect point cloud data from different angles of the model. During the acquisition process, the data obtained from each change of angle measurement is generally partially overlapping and is likely to have rotational misalignment and translational misalignment problems. In order to build a complete 3D model, these point cloud data need to be integrated and aligned, for example, by rotating and translating a point cloud data to make it fit with the duplicate part of another point cloud data, so that a more complete 3D model can be obtained.

For such an alignment problem, the classical ICP algorithm is generally used. This algorithm can be divided into 6 steps.

1.Pick the initial points from the two initial data.

2.Match the points with the target mesh.

3.Weighing the matched point pairs.

4.Reject the bad point pairs.

5.Construct the error function, i.e., the Euclidean distance between the point pairs.

6. Optimize the solution.

In simple terms, the algorithm optimally solves the minimum Euclidean distance between the corresponding points of the two frames to obtain the transformation relationship between the two-point clouds. However, in solving the rotation and translation matrices of the two-point clouds, the least squares method with SVD and polar decomposition is generally used for extraction, and the computational process of this method is very complicated. In this paper, this step is simplified by using an approximation method. And it is improved from point-to-point optimization to point-to-surface optimization, so that the speed of alignment can be greatly improved.

2. Improved ICP Algorithm

2.1 Approximate Simplification of Rotation Matrix

For any rotation matrix R, we can decompose it into the product of three basic rotation matrices$R_x(\alpha)$, $R_y(\beta)$, $R_z(\gamma)$ representing the rotations on the x, y, and z axes, respectively. The three basic rotation matrices and their products yield the following rotation matrix

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix},$$

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$R_z(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathrm{R} = \mathrm{R_x}(\alpha)\mathrm{R_y}(\beta)\mathrm{R_z}(\gamma)$$
$$= \begin{pmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\gamma\cos\beta \end{pmatrix}$$

Assume that there is only a very small rotation angle and therefore there is $\sin x = x, \cos x = 1$. From this, R can be obtained as a simplification and the result is as follows.

$$R = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}$$

2.2 Point-to-point Optimization

After the first four steps of the ICP algorithm, we can obtain two groups of points, P and Q. And each point found in group P can be found in Q corresponding to the matching point that needs to be approached. Thus, we have k sets of point pairs (pi, qi). We need to find a transformation (including rotation matrix R and translation matrix t) to minimize the Euclidean distance between the point pairs. This error function is as follows:

$$E = \sum_{i=1}^{k} \|Rp_i + t - q_i\|_2^2$$

To obtain the minimum value of this function, we derive it and make it equal to zero, obtaining

$$\mathrm{Rp_i + t - q_i = 0}$$

By simplifying rotation matrix in 2.1, we can further transform the above equation to obtain

$$\begin{pmatrix} 0 & p_{i_z} & -p_{i_y} & 1 & 0 & 0 \\ -p_{i_z} & 0 & p_{i_x} & 0 & 1 & 0 \\ p_{i_y} & -p_{i_x} & 0 & 0 & 0 & 1 \end{pmatrix} (\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z)^T$$
$$= (q_{i_x} - p_{i_x} \quad q_y - p_{i_{yx}} \quad q_{i_{zx}} - p_{i_{zx}})^T$$

In which, $x = (\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z)^T$ contains all the unknowns in the rotation and translation matrices that we need, and just by finding it, we can get the bitwise transformation of the whole point cloud. We get it by solving.

2.3 Point-to-Surface Optimization

The convergence speed of point-to-point optimization is slow because it is only a matching of point pairs and cannot be shifted along the faces. If we have the normal vector data of the mesh face where each point of the target mesh is located, the convergence speed can be greatly improved. The point-to-point optimization error function is as follows.

Translated with www.DeepL.com/Translator (free version)

$$E = \sum_{i=1}^{k} \left\|n_i^T(Rp_i + t - q_i)\right\|_2^2$$

Similar to the calculation steps in 2.2, the above equation can be transformed to obtain

$$(-p_{i_z}n_{i_y} + p_{i_y}n_{i_z} \quad p_{i_z}n_{i_x} - p_{i_x}n_{i_z} \quad -p_{i_y}n_{i_x} + p_{i_y}n_{i_x} \quad n_{i_x} \quad n_{i_y} \quad n_{i_z})(\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z)^T = \left(n_{i_x}(q_{i_x} - p_{i_x}) \quad n_{i_y}\left(q_y - p_{i_{yx}}\right) \quad n_{i_z}(q_{i_{zx}} - p_{i_{zx}})\right)^T$$

Again, use$Ax = b$ to get results.

3. Experimental Procedure and Analysis of Results

3.1 Data Sampling

This experiment was validated using the stanford bunny dataset. figure 1 shows the 3D model of the first two point cloud data. the first step to be done in the ICP algorithm is to sample the data. here the average sampling method was used. the sampling results are shown in figure 2.

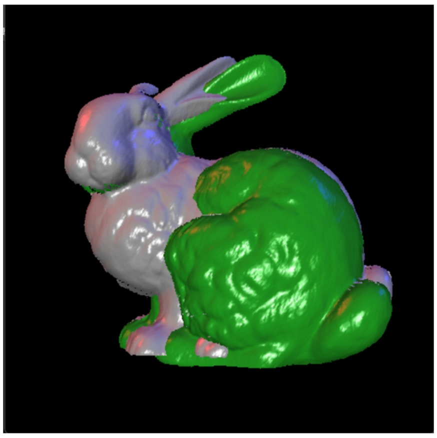

Figure 1. The 3D model of the first two point cloud data

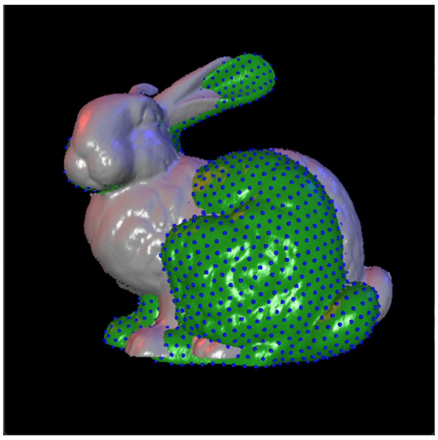

Figure 2. The results of the average sampling method

3.2 Matching Point Pairs

The ICP algorithm usually looks for the closest point pair when matching point pairs, which has a complexity of O(n) if searching directly and takes too much time if the data volume is huge. To improve the speed, this experiment uses the binary search tree method. Each time the space is divided into two parts by iterating with planes, as shown in Figure 3, planes A, B, and C divide the space into four parts, constructing a binary search tree as shown in Figure 4. In this way, only the points in a region are searched for in each search match, take Figure 3 as an example, the red point is the target point, it is in the F region, we only need to search all the points in the F region to get a minimum distance. And compare the minimum distance obtained from the search with the distance between the target point and the plane. If it is smaller than the distance from the plane, then the point obtained by the search is the matching result. If the distance to the plane is smaller, then the possibility that points in other regions have smaller distances cannot be excluded, and it is necessary to return to the upper level of the tree to search again. Overall, such a search algorithm greatly improves the efficiency and reduces the complexity to O(nlogn).

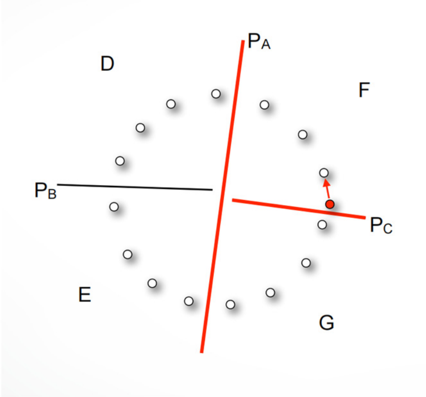

Figure 3. The space is divided into two parts by iterating with planes

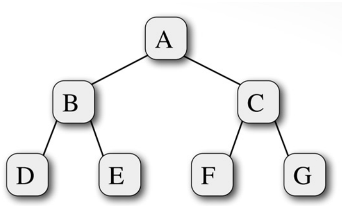

Figure 4. Constructing a binary search tree

3.3 Optimization Iteration

Both point-to-point and point-to-counter optimization converge well after a number of iterations. The final results are shown in Figure 5. However, the point-to-point optimization method is very much faster than the point-to-point method. As shown in Figure 6, the left side is the result of the initial two point clouds after 5 iterations of the point-to-point optimization, while the right side is the result after 5 iterations of the point-to-point optimization, which can be seen that after 5 iterations, the point-to-point optimization can already be well fitted, while the point-to-point optimization still needs many iterations. The specific results are shown in Figure 7. It can be seen that the convergence effect of the point-to-point optimization is significantly better than that of the point-to-point.

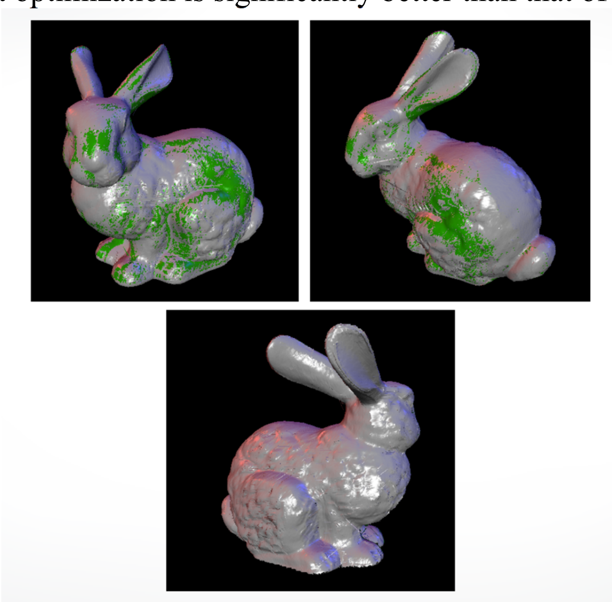

Figure 5. The end result of multiple iterations

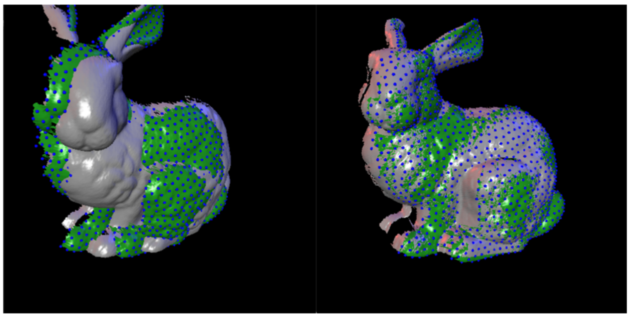

Figure 6. The results of iterations are different in different ways

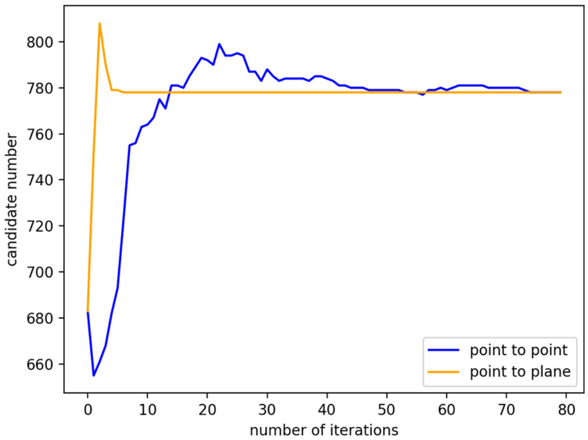

Figure 7. Simulation comparison of two different optimization methods

References

[1] Rusinkiewicz, Szymon M. and Marc Levoy. “Efficient variants of the ICP algorithm.” Proceedings Third International Conference on 3-D Digital Imaging and Modeling (2001): 145-152.

[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," Proceedings. 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2724-2729 vol.3, doi: 10.1109/ ROBOT. 1991.132043.

[3] Besl, Paul J. and Neil D. McKay. “A Method for Registration of 3-D Shapes.” IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992): 239-256.

[4] Horn, Berthold K. P.. “Closed-form solution of absolute orientation using unit quaternions.” Journal of The Optical Society of America A-optics Image Science and Vision 4 (1987): 629-642.

[5] Gelfand, Natasha et al. “Geometrically stable sampling for the ICP algorithm.” Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. (2003): 260-267.

[6] Pulli, Kari. “Multiview registration for large data sets.” Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062) (1999): 160-168.