ATLANTIS PRESS

Advances in Engineering Research, volume 138

2nd International Conference on Machinery, Electronics and Control Simulation (MECS 2017)

Rigid 3D Point Cloud Registration Based on Point Feature Histograms

Xi Wang [a], Xutang Zhang [b]

School of Mechatronics Engineering, Harbin Institue of Technology, Harbin 150001, China

[a]wang727803@qq.com, [b]3221884957@qq.com

Keywords: 3D point cloud, 3D registration, rigid transformation, Iterative Closest Point(ICP), Point Feature Histograms(PFH)

Abstract. Depending on the displacement and orientation between point clouds, the registration of scattered point clouds is offten divided into two steps: crude and fine alignment. An approach of point cloud classification based on point feature histogram was proposed in this paper. We propose a method of establishing the point feature histograms to match feature points in different clouds. To reject the outliers, Random Sample Consensus algorithm is used. The rigid transformation matrix in crude alignment is then computed by Singular Value Decomposition method. The golden standard for fine alignment is the Iterative Closest Point algorithm and its variants. In this paper we apply a dynamic constraint of distance to improve the traditional algorithm. The experiment shows that our process of registration works fine with higher accuracy and efficiency.

1. Introduction

During the last few years, the technology research of computer vision applications has shifted from 2D image towards surface or environment reconstruction based on 3D point cloud data. Registration algorithms can be widely used in many fields, such as 3D object scanning, 3D map reconstruction and so on. In many applications of reverse engineering, multiple measurements from different orientation is required to get the complete data of the target. Point clouds obtained in different coordinate system need to be aligned in one, as called global coordinate system, which is often refer to as registration.

Complex or simple, there are many methods for Point cloud registration. A simple Singular Value Decomposition(SVD) [1] or Principal Component Analysis(PCA) based registrations may work fine in some cases, but most of the state-of-art applications offten employ a more advanced iterative scheme based on the Iterative Closest Point(ICP)[2] algorithm. As any registration problem, point cloud registration consists of matching and estimation of the rigid transformation. ICP as well as its variants[3~5], however, requires a proper initial relative pose between source cloud and target cloud. Researchers often divide the point cloud align process into crude alignment and fine registration.Jiang[6] proposed an angular-invariant feature method. SHEN[7] create a lower dimension descriptor named LDFH to registrate clouds. While He Y propose a spin image based three-dimensional feature to registrate LiDAR point cloud models in [8]. And the concepts of balance factor is proposed in [9] to improve ICP algorithm.

In this paper, the feature points are extracted rapidly according to their normal information. To get the points correspodences, the point feature histograms are established for each feature point. Finally, a variant ICP is used to align the cloud accurately. Concretely, to solve the registration problem, (Part 2)the normals of the points in each cloud is computed according to [10]. Then, the feature points is extracted on each cloud according to the distribution of normals, non-maximum suppression(NMS) is used to filter those features. Then a point's descriptor is constructed by the relation between its normal and neighbor's. Combined with the distance constraint of rigid transformation, Random Sample Consensus(RANSAC) algorithm is used to obtain a set of reliable matches, thus, (Part 3)the transform matrix of crude alignment is computed with simple Singular Value Decomposition(SVD) algorithm. In the further registration, we improve the traditional ICP algorithm with a dynamic constraint of distance. (Part 4)The exprienment shows that our method to establish PFH is feasible

and effetive in crude alignment, and the whole process works fine with higher accuracy and efficiency.

2. Extraction and Matching of Feature Points

2.1 Feature Extraction

Feature points, also known as key points, are detected by the algorithm on the point cloud, which is stable, distinguishable and rich in information. The number of feature points of a point cloud is only a smal proportion, so the efficiency of searching and evaluating matches between clouds can be greatly improved by analyzing the feature point set.

The feature points candidate of a cloud is extracted according to its normal vectors, which can be computed using the algorithm proposed in [10]. Firgure 1 shows the distribution of normals in different regions, it's easily to found that region with larger fluctuation distributes normals with larger variation, while the smooth region has almost parallel normals. The regoin with large variation of the normals is suitable to be a feature, from this observation we evaluate a point's feature score by the variation between its normal vector and its neighbors'. For the point p, define the mean angle between its normal vector n and the normal vectors of the points in its k-neighborhood as its informative characteristics:

$$f(p)=\frac{1}{k}\sum_{i=1}^{k}<\vec{n},\overrightarrow{n_k}> \tag{1}$$

where $f(p)$ is the characteristics of p, n_k is the neighbor's normal vector. Under this definition, $f(p)$ represents the fluctuation of the region on p, so we can use this feature to extract the key points in the point cloud. Select threshold ε_1, p_i is considered suitable to be a feature point only when $f(p_i)>\varepsilon_1$. The non-maximum suppression(NMS) strategy is use to filter out the redundant or overlapping points. Concretely, p_i is add to the feature points set when:

$$\begin{cases} f(p_i)>\varepsilon_1 \\ f(p_i)=\max\{f(p_{ik})\} \end{cases} \tag{2}$$

where $f(p_{ik})$ is the characteristics of p_i's nerghbor p_{ik}. This means only the point with maximum characteristics value in the neighborhood can be choosen as a feature point. For the two input point clouds S (source cloud) and T (target cloud), we now get $S_f=\{s_1,s_2,s_3,...s_m\}$ and $T_f=\{t_1,t_2,t_3,...t_n\}$ as their feature points set, where m,n is the number of feature points in S and T extract by above method.

We then test the feature extraction on the test model stanford bunny, as show in figure 2. It can be seen that by apply NMS the redundant points after feature extraction(figure 2b) is significantly reduced(figure 2(c)). Besides, the feature points are much more uniformly distributed, which is very conducive to the subsequent process.

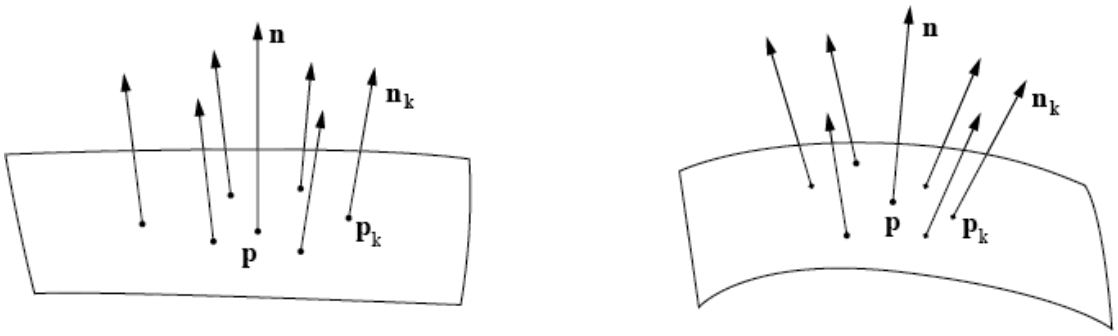
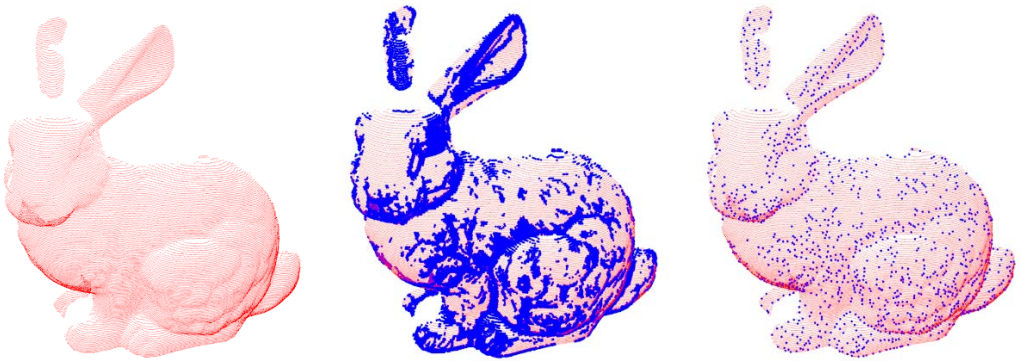


Fig. 1 Normal vectors distribution in different region



(a) (b) (c)

Fig. 2 Feature points extraction. (a) input point cloud, (b) feature points extracted according to normals, (c) feature points after filtered with non-maximum suppression strategy

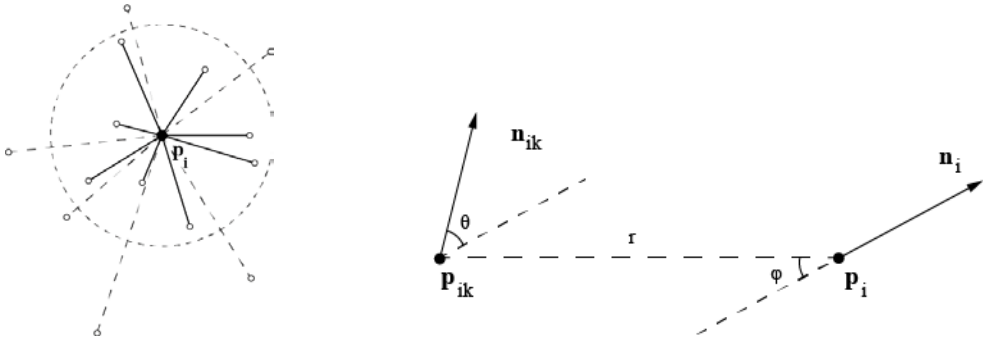
2.2 Point Feature Histograms

The normal vectors describe the local geometric features, is an important feature of the surface. However, as point feature representations go, surface normals and curvature estimates are somewhat basic in their representations of the geometry around a specific point. Though extremely fast and easy to compute, they cannot capture too much details. In mose case, the surface point cloud will contain many points with the same or very similar feature values, we cannot use these eigenvalues to matching points directly.

Point Feature Histograms(PFH) construct a highly dimensional represents on point's geometry feature, it is widely use in robotics object recognition and image registration[11]. In this paper we introduce a feature desciptor based on PFH. A PFH representation is based on the relationships between the points in the k-neighborhood and their estimated surface normals. Simply put, it attempts to capture as best as possible the sampled surface variations by taking into account all the interactions between the directions of the estimated normals. Thus the PFH works much better matching feature points[12].

Figure 3(a) presents an influence region of the PFH computation of a query point p_i, all its k neighbors in its R-domain is computed so as to got the final PFH descriptor. To compute the relative difference between p_i and the point p_{ik} in its nerghborhood, their associated normals n_i and n_{ik} are specified(Figure 3(b)), a set of features are expressed as follows:

$$\begin{cases} \theta = < n_i, n_{ik} > \\ r = \| p_i - p_{ik} \| \\ \varphi = < n_i, (p_i - p_{ik}) > \end{cases} \tag{3}$$



(a) (b)

Fig. 3 (a) infulence region, (b) relations between two points

where θ represents the angle between the normals, r is the Euclidean distance between the two points and the angle of n_i and the vector from p_{ik} to p_i is computed as φ. All of them have invariance of displacement and rotation. θ and φ is in $[0, \pi]$. In general, θ can get a larger value only when the point is in a region of great fluctuation such as the model edges and corners, so the value range is divided into four subintervals: $[0, \pi/12)$, $[\pi/12, \pi/6)$, $[\pi/6, \pi/4)$ and $[\pi/4, \pi]$;

while φ represents the curvature of the surface, the vlaue will be distributed near $\pi/2$ since the model can be most of smooth region, the value range is divided from the middle; r hits in $(0, R]$, consider the average distance γ between points in the whole cloud, the range is divided with step γ, that is, $N = floor(R/\gamma)+1$ subintervals. In this way, the description of the relation between p_i and p_{ik} falls in specified $L = 4\times 2\times N$ subdivisions.

To create the PFH representation of the query point, all k nerghbors is binned into a histogram, thus we obtain a L-dimensional vector, divided by the number of nerghbors is the descriptor of p_i. Figure 4 presents a example of PFH representation at different regions, in which the search radius $R = 4\times\gamma$.

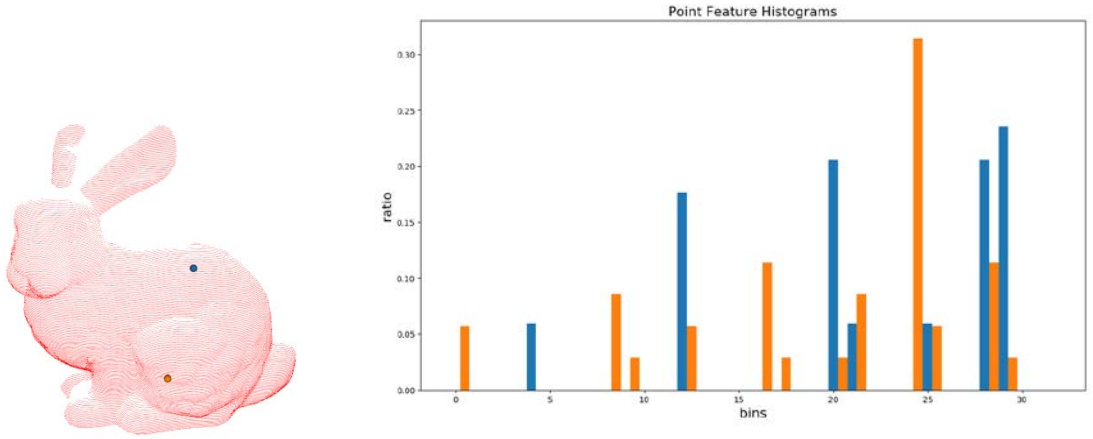


Fig. 4 Sample of point feature histograms

2.3 Matching Features

Now we have the feature points set S_f and T_f as well as their PFH descriptors for each point. The next thing is to search matches between them. The initial set of matches M is constructed according to the feature points' PFH descriptors: for each point in S_f, seaching all points in T_f, the corresponding point is the point with nearest descriptor (that is, minimum variance of histograms). Meanwhile, threshold ε_2 is used to filter the matches, matches with PFH difference lower than ε_2 can be pushed into M:

$$M = \{m_i = (s_i, t_i) \mid s_i \in S_f, t_i \in T_f\} \tag{4}$$

In face, a single point cloud may contains many similar points, it is inevitable that the initial matching set contains many wrong matches, further screening is needed to obtain precise matches.

In this paper, the inital set is streamlined by the distance constraint of rigid transformation combine with the similar Gaussian curvature constraint. Then The Random Sample Consensus(RANSAC) algorithm is used to select the accurate matches.

For two pairs in M: $m_i = (s_i, t_i)$, $m_j = (s_j, t_j)$, if both of them are the right matches between the input clouds. then:

1) distance constraint

Rigid transformation is invariant on distance: $dist(s_i, s_j) = dist(t_i, t_j)$

2) Gaussian curvature constraint

Gaussian curvature is the product of the principal curvatures, it is an intrinsic measure of curvature. We take Gaussian curvature into evaluation, that the correct corresponding points shall have the same value: $K(s_i) = K(t_i)$

However, it's diffcult to find the perfect corresponding points in the actual situation. So the upper two constraints are taken into satisfaction approximately. In particular:

$$K = \left| \frac{K(s_i) - K(t_i)}{K(s_i) + K(t_i)} \right| < \varepsilon_k \tag{5}$$

$$D = \left| \frac{dist(s_i, s_j) - dist(t_i, t_j)}{dist(s_i, s_j) + dist(t_i, t_j)} \right| < \varepsilon_d \tag{6}$$

where ε_k and ε_d is the threshold value of error tolerance.

During simplification, the streamlined matching set M_k is obtained by apply the Gaussian curvature constraint. Then for each m_i in M_k, test all m_j in the rest of M_k, counting the number of m_j which satisfied the expression (6) with m_i to C_i. Note that correct matches m_i result in larger C_i, since there has been a certain number of correct matching pairs in the filtered M_k, we select N matches with the N maximum C, marked as M_N.

It is inevitable that the streamlined set M_N still contains some wrong correspondences, which could bring serious error in matching and estimation of the rigid transformaion. Here we use RANSAC algorithm to filter out those wrong matching pairs: Each time three pairs of points are randomly selected from M_N to estimate the rigid transformation matrix T. The source points in M_N is then transformed with T to check if they are correct in this sampling, that is, for $m_i = (s_i, t_i)$, m_i is classified as the correct pair if only $\| T(s_i) - t_i \| < \varepsilon_3$. Repeat this process until the number of random samplings reaches the setted iteration times, the exact matches set M' is the correct matches set of the sampling with the maximum correct points in all iterations.

3. Registration

3.1 Crude Registration

Since the set of reliable matches M' has been obtained, the rigid transfromation between S and T can be evaluated by Singular Value Decomposition(SVD) algorithm. Transform the source cloud with the estimated transform matrix T_c to $S_c = T_c(S)$, here comes to accurate registration with S_c and T.

3.2 Accurate Registration

Consider S_c as the initial source cloud for registration. The traditional ICP algorithm requires that: (1) the initial position of the two clouds cannot differ too much; (2) there is an inclusion relation between them. Though the clouds already have a fine inital position after crude registration, the traditional ICP cannot be used directly since the input clouds are only partially overlapping. We propose a variant of traditional ICP algorithm:

In each iteration, the point on the source cloud search for closest point from the target cloud, to reject the wrong correspondences(which often produce large distances between two clouds) a threshold ε is used to filter out the points with too much corresponding distance. Simply put, in the n th iteration, the point s_i is removed from the search list if:

$$\frac{dist(s_i, t_i)}{\max\{dist(s,t)\}} > \varepsilon(n) \tag{7}$$

where $\max\{dist(s,t)\}$ is the maximum corresponding distance in this searching process. Since the point clouds are partially overlapping, threshold $\varepsilon(0)$ can be smaller to filter out the non-overlapping region of the clouds. As the iteration proceeds, the screening should be loosened since the two clouds are aligning better, which means the $\varepsilon(n)$ should increase along n. So a dynamic threshold cutting method is proposed:

$$\varepsilon(n) = \begin{cases} \varepsilon_0 & n = 0 \\ 1 - e^{-\alpha(n+\beta)} & n > 0 \end{cases} \tag{8}$$

Through this improvement, the non-overlapping region between two clouds is greatly removed, this ensures the convergence of ICP algorithm. Besides, the elimination of corresponding points is still running in each iteration, which speed up the algorithm by reducing the searching points. The threshold ε increases over interations and finnally approaches to 1(Figure 5), that means the elimination is deactived after certain number of times, therefore, guaranteed the accuracy of alignment and ensure the effectiveness of the registration.

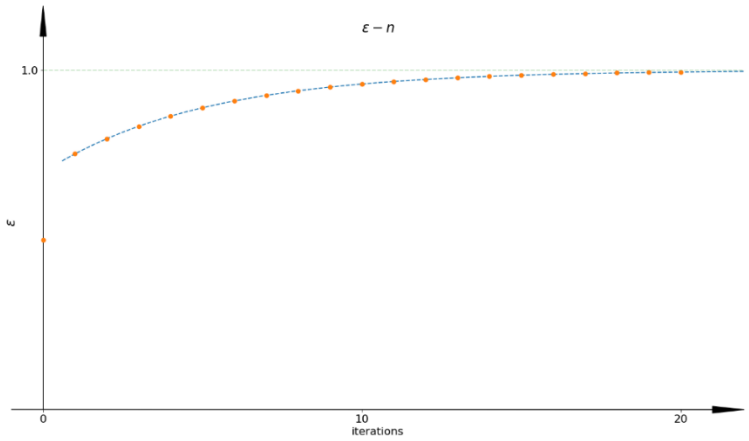
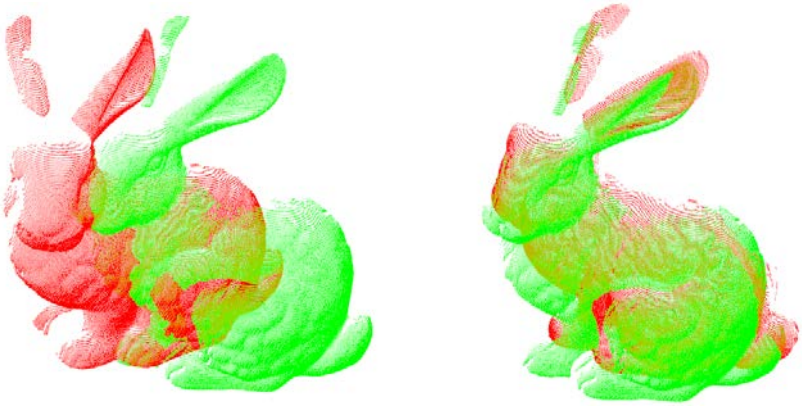


Fig. 5 Dynamic threshold ε

4. Experimental Results

We now present the experimental results of our registration algorithm. The data source is obtained from Stanford University Computer Graphics Laboratory.

Figure 6 shows the crude alignment of stanford bunny data, the parameters choosen in this experiment are: $K = 10$, $\varepsilon_1 = 0.1$, $R = 0.024$, $\varepsilon_2 = 0.02$, $\varepsilon_k = 0.1$, $\varepsilon_d = 0.1$ In figure 6a, source cloud S is load from bunny000 rendered in red, while the target cloud T from bunny045 in green. The result of crude alignment is show in figure 6b. It can be seen that the clouds have been aligned much better, with Mean Squared Error(MSE) 0.0147941. The validity of the proposed PFH based feature extraction and matching method is proved.



(a) (b)

Fig. 6 Crude Registration (a) initial position (b) aligned with algorithm in this paper

The clouds after crude registration are set as inputs of the fine registration process. The proposed improved ICP algorithm is used to estimate accurate transformation, and the closest point screening method in this paper is compared with the traditional ICP iterative method. The result of accurate registration is shown in figure 7. Figure 7b shows the final aligning result of two clouds. Table 1 lists the number of iterations, MSE and the computing time of two algorithms. The convergence of the algorithm is shown in figure 8. The parameter selected in this process is: $\alpha = 4$, $\beta = 6$. Note that in the implementation of the algorithms, we refered to [13] and construct KD tree to speed up the closest point searching process in iteration, and the time consumed constructing KD tree is not included

when comparing computing time. It is shown in the experimental results that the proposed algorihtm can complete the task of accurate registration and, the improved ICP algorithm based on dynamic threshold filter works much better than the traditional ICP algorithm in the convergence speed as well as the accuracy. The validity of the proposed accurate registion algorithm is verified.

(a) (b)

Fig.7 Registration results (a) tranditional ICP algorithm (b) improved ICP in this paper

Table 1 Comparison of traditional ICP and our algorithm

	Iteration	MSE	Time
Tranditional ICP	6	0.003091	5.541
Our algorithm	5	0.001171	4.809

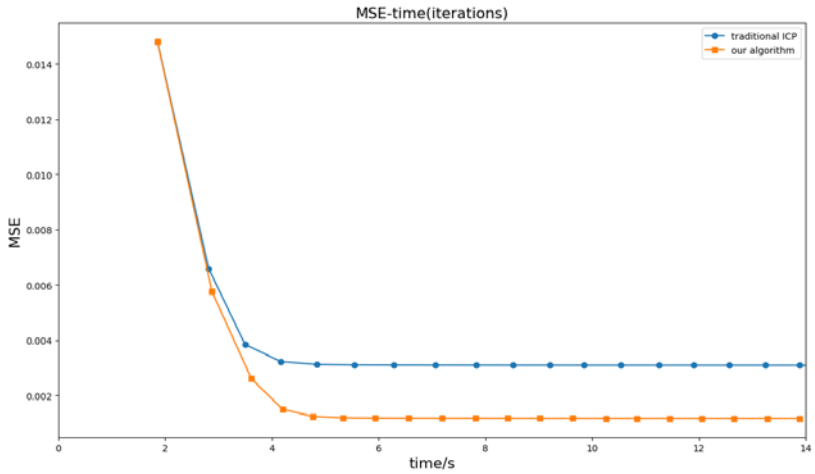


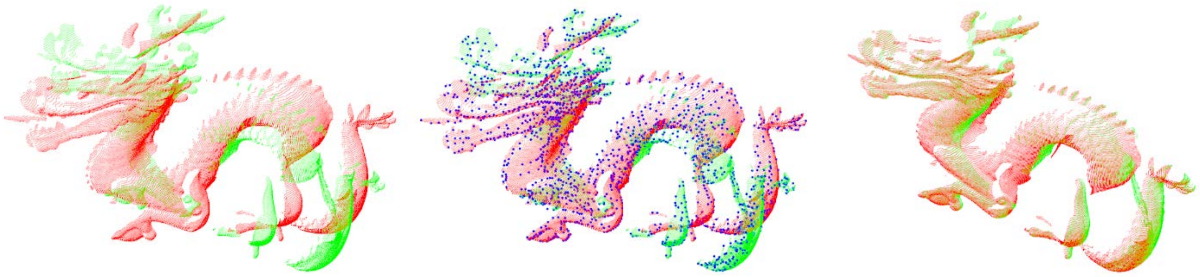
Fig. 8 Convergence of tranditional ICP and our algorithm

To verify the utilities and commonalities of our algorithm, the proposed process was also tested on the Dragon model and got the satisfactory result(with MSE only 0.003035), which is inllustrated by Figure 9.

5. Conclusions

In this paper we studied the point cloud regitration algorithm. In the crude alignment, the feature points candidate set is rapidly obtained with analysis of normals, we proposed a PFH descriptor based matching method combine with RANSAC algorithm to search the reliable corresponding points between source cloud and target cloud, the crude aligning matrix is then computed by SVD algorithm. In accurate registration, we apply a dynamic constraint of distance to improve traditional ICP algorithm, KD tree searching method is applyed in our experiment. The experimental results showed that the proposed feature matching based crude aligning algorithm is feasible and effective, and our

variant of ICP algorithm has the advantages in accuracy and effciency. The whole process is effective and feasible on point cloud registration.



(a) (b) (c)

Fig 9 Registration of Dragon model. (a) the initial position (b) feature points extracted (c) result

6. References

[1]. S.Marden and J.Guivant, Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search, 2012, pp. 3-5.

[2]. P.J. Besl and N.D. McKay, Method for registration of 3D Shapes, P.S. Schenker, Ed., Apr. 1992, pp. 586-606

[3]. Hacene A, Mekki A. Bio-CAD reverse engineering of free-form surfaces by planar contours [J]. Computer-Aid Design & Applications, 2011, 8(1):37-42

[4]. Senin N, Colosimo BM, Pacella M. Point set augmentation through fitting for enhanced ICP registration of point clouds in multisensory coordinate metrology [J]. Robotics and Computer-Intergrated Manufacturing, 2013, 29 (1): 39-52

[5]. ZHONG Ying, ZHANG Meng. Registration of scattered cloud data [J]. Control Engineering of China, 2014, 21(1): 37-40(in Chinese).

[6]. J Jiang, J Cheng, X L Chen. Registration for 3D point clouds using angular-invariant feature[j]. Neuro Computing, 2009, 72(16-18): 3839-3844

[7]. SHEN Ying-hua, LI Zhuo-jia, YANG Cheng, LI Hao-yong. Point cloud registration with normal feature histogram[J]. Optics and Precision Engineering, 2015, 23(10z): 591-598

[8]. He Y, Mei Y. An efficient registration algorithm based on spin image for LiDAR 3D point cloud models ☆[J]. Neurocomputing, 2015, 151(1):354-363.

[9]. YANG Xiao-qing, YANG Qiu-xiang, YANG Jian. Improved ICP algorithm based on normal vector[J]. Computer Engineering and Design, 2016, 37(1): 169-173

[10]. Hoppe H, Derose T, Duchamp T, et al. Surface reconstruction from unorganized points[C]. Conference on Computer Graphics and Interactive Techniques. ACM, 1992, 26(2):71-78.

[11]. Zhang Aiwu, LiWennig, Duan Yihao, et al. Point Cloud Classification Based on Point Feature Histogram[J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(5): 795-801

[12]. R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz, Aligning Point Cloud Views using Persistent Feature Histograms[C], International Conference on Intelligent Robots and Systems, September 22-26, 2008: 3384-3391

[13]. [13] Greenspan M, Yurick M. Approximate K-D Tree Search for Efficient ICP[C], International Conference on 3-D Digital Imaging and Modeling, 2003. 3dim 2003. Proceedings. DBLP, 2003:442-448.