Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
17th Conference on Applied Mathematics **APLIMAT 2018**

Proceedings

NOTES ON ITERATIVE CLOSEST POINT ALGORITHM

PROCHAZKOVA Jana (CZ), MARTISEK Dalibor (CZ)

Abstract. Point clouds are commonly used in many areas of technical practice. Nowadays, the common applications are in area of autonomous cars and 3D terrain or object modelling. During the scanning process, the scanning device or object is moving. So, the problem of precise point cloud registration arises. In this article, we describe Iterative Closest Point (ICP) algorithm that is suitable for fine registration. We introduce some of the improvements that leads to higher precision, speed computation and efficiency. We also tested the ICP algorithm on the noisy point clouds to process the registration.

Keywords: Iterative closest point, point cloud, least square method

Mathematics subject classification: Primary 65D18; Secondary 68U05

1 Introduction

We currently see substantia arise of point clouds applications in a large number of technical disciplines. The most significant applications include the use of 3D scanning technology to describe the environment (roads, city parts, forests, etc.). Nowadays, we can see the progress of scanning and processing technologies in area of autonomous vehicles [20]. For this purpose is often used LiDAR technology (Light Detection and Ranging, [3] [12]). It is a surveying method that measures distance to a target by illuminating that target with a pulsed laser light, and measuring the reflected pulse with a sensor. In engineering applications, the optical 3D scanners are widely employed. The result of both scanning principles is a point cloud that can be described as a set of data points in some coordinate system, usually defined by x, y, z coordinates.

When a sensing device or a scanned object moves, the resulting cloud of points is partially overlaid. The main task is to assemble these partial scans together to form the final object. This process is usually called point registration.

Due to [4], we can divide the registration process into two main parts: coarse and fine matching. In the coarse matching part, we find the corresponding points with detectors and (or) descriptors. There is a huge number of detector and descriptor methods, e.g. Maximally Stable Volumes [14], MeshDoG, Instrinsic Shape Signatures [22], Point Signature descriptor [9] and Spin Image [10]. This part determines the coarse matching of the scans. In our article, we deal with to second part – fine

matching that align these two point clouds precisely. The most powerful algorithm Iterative Closest Points is presented in Sec. 2 and the results are described in Sec. 3 in details.

Fig. 1. Point cloud registration, source [4].

In our article, we introduce Iterative Closest Point (ICP) algorithm that is one of the common used algorithms in practice. The algorithm was firstly described in [8], [1]. The basis of the algorithm is to find an optimal mapping consisting of a translation and a rotation that minimizes the distance between the corresponding points in the least squares sense.

The authors in [1] formulated the problem as:

"Given 3-D data in a sensor coordinate system, which describes a data shape that may correspond to a model shape, and given a model shape in a model coordinate system in a different geometric shape representation, estimate the optimal rotation and translation that aligns, or registers, the model shape and the data shape minimizing the distance between the shapes and thereby allowing determination of the equivalence of the shapes via a mean-square distance metric."

2 ICP algorithm

2.1 Theoretical background

In this section, we briefly describe the mathematical background of the ICP algorithm. For more details, see e.g. [2]. This algorithm is very popular because of its simplicity and clarity. However, the basic algorithm works only in ideal cases, leading to its further modifications and improvements, which will be described in 2.3.

Generally, let $\mathcal{M}, \mathcal{S}$ are two finite sets in finite dimensional real vector space R^d that contains M, resp. N points. We want to determine the rigid or non-rigid transformation $T : R^d \longrightarrow R^d$ that minimize the distance between given sets:

$$\mathtt{dist}_g(T(\mathcal{M}), \mathcal{S}) = \sum_{m \in T(\mathcal{M})} \sum_{s \in \mathcal{S}} g((m-s)^2) \tag{1}$$

Also Root Mean Square Distance (RMSD) can be used to comparison of $T(\mathcal{M})$ and $\mathcal{S}$:

$$RMSD(T(\mathcal{M}), \mathcal{S}) = \frac{\sum_{m \in T(\mathcal{M})} \sum_{s \in \mathcal{S}} |m-s|^2}{n} \tag{2}$$

In the case that the sets $\mathcal{M}, \mathcal{S}$ contain the outliers, i.e. points that are not in both sets, we can compute partial overlapping only. In these cases, it is suitable to set the threshold parameter thr that eliminates the mistakes. For a set of pairs of points $C = (m_i, s_j)$, where $m_i \in \mathcal{M}$, $s_j \in \mathcal{S}$ must hold:

$$\forall s_k \in \mathcal{S} : dist(m_i, s_k) \geq dist(m_i, s_j), dist(m_i, s_j) < thr \tag{3}$$

The Eq. (2) can be written as:

$$RMSD(T(\mathcal{M}), \mu(\mathcal{S})) = \frac{\sum_C dist(m_i, T(s_j)^2}{|C|}, m_i, s_j \in C \tag{4}$$

where distance $T(m_j)$ and s_i is minimal and $|C|$ is cardinality of the set C.

2.2 ICP overview

Let $\mathcal{S}$ and $\mathcal{M}$ are input point clouds where source point cloud is $\mathcal{S}$ and $\mathcal{M}$ is model point cloud. We are looking for the rigid transformation which satisfies the best correspondence between given point clouds. The following steps describes the ICP algorithm:

1. compute the nearest point in the set $\mathcal{S}$ for every point (part of the points) in the set $\mathcal{M}$ using e.g. Euclidean distance:

$$d_i = \min\{\sqrt{m_i^2 - s_j^2}\} \quad j = 1, \ldots, N_S \tag{5}$$

2. if the distance $d_i > treshold$, remove a given pair of points

3. add the weights to pairs of points: implicitly $w_{ij} = 1$, we can set the weights in the dependence of the direction of normal vectors as their dot product:

$$w_{ij} = n_i n_j \tag{6}$$

4. compute the rotation matrix R and translation vector $\mathbf{t}$ using the least square method for distance minimization that is described in Eq. (9)-(13)

5. compute the transformation of the set $\mathcal{S}$ using computed values – $Rs_j + \mathbf{t}$,

6. compute the error $E(R, \mathbf{t})$ using Eq.(7), iterate until we reach to required accuracy.

The pseudocode of the algorithm ICP:

$$\begin{aligned}
&AlgorithmICP(\mathcal{M}, \mathcal{S}) \\
&\qquad \theta := \theta_0 \\
&\text{while not registered:} \\
&\qquad X := \emptyset \\
&\qquad \text{for } \; m_i \in T(\mathcal{M}, \theta) \\
&\qquad\quad \hat{s}_j := \text{closest point in } \mathcal{S} \text{ to } m_i \\
&\qquad\quad X := X + \langle m_i, \hat{s}_j \rangle \\
&\qquad\quad \theta = \; \text{least squares}(X) \text{ using SVD} \\
&\text{return } \theta
\end{aligned}$$

In the pseudocode the symbol θ means the rotation matrix R and translation vector $\mathbf{t}$. The initial values θ_0 are: identity matrix R and zero translation vector $\mathbf{t}$. The set X contains all the pairs that cover the closest points in set $\mathcal{S}$ to points in the set $\mathcal{M}$.

In the next step, we calculate the new transformation $\theta = (R, \mathbf{t})$ using SVD and Eq.(9)-(13). Then, the transfromation θ is applied to the model points $\mathcal{M}$ and the error is computed. We iterate this setup until required accuracy.

2.2.1 Mathematical description of the ICP

Let $\mathcal{M}, \mathcal{S}$ are the finite sets with N_M and N_S points. Firstly, we compute the nearest point in the set $\mathcal{S}$ for every points in the set $\mathcal{M}$. It is possible to use all of the points or we can choose the points randomly. The ICP algorithm suppose that corresponding points are the nearest ones. For these pairs, we compute the rotation matrix and translation. We try to minimize the error:

$$E(R,t) = \sum_{i=1}^{N_M} \sum_{i=1}^{N_S} w_{ij} \|m_i - (Rs_j + t)\|^2 \tag{7}$$

where w_{ij} are the weights for corresponding points. If point m_i is the nearest point to d_j, we set the weights as $w_{ij} = 1$, otherwise the weight is equal to zero.

The Eq. 7 can be rewritten as:

$$E(R,t) \propto \frac{1}{N} \sum_{i=1}^{N} \|m_i - (Rs_i + t)\|^2 \tag{8}$$

where

$$N = \sum_{i=1}^{N_M} \sum_{i=1}^{N_S} w_{ij}$$

and corresponding matrix of w_{ij} is substituted by the sum of corresponding points.

The optimization step (find $(R, \mathbf{t})$) is based on Singular Value Decomposition (SVD). For the computation of SVD, we need centroid alignment. We compute the centroids of both point sets as:

$$C_M = \frac{1}{N} \sum_{i=1}^{N} m_i \tag{9}$$

$$C_S = \frac{1}{N} \sum_{i=1}^{N} s_i \tag{10}$$

Subsequently, we align all of the points of $\mathcal{M}, \mathcal{S}$ using corresponding centroid and we get:

$$\mathcal{M}' = \{m_i' = m_i - C_M\}_{1,\dots,N}$$

$$\mathcal{S}' = \{s_i' = s_i - C_S\}_{1,\dots,N}$$

For these values, it is easy to compute covariance matrix $H = \mathcal{M}'\mathcal{S}'^T$ and make the SVD as:

$$H = U\Lambda V^T \tag{11}$$

The matrix of rotation is given by:

$$R = VU^T \tag{12}$$

and vector of translation can be computed by:

$$\mathbf{t} = C_S - RC_M \tag{13}$$

First iteration consists of the transformation of the cloud points using rotation matrix R and translation vector $\mathbf{t}$. Subsequently, we compute the error and if the error is higher then given threshold, the iteration continues.

2.3 The improvement methods of ICP

The previous section outlines the basic steps and theoretical background of the ICP algorithm. Nevertheless, there is the amount of the improvements that try to satisfy better convergence and improve the computation time. The overview can be find in e.g. [17] in details. In this part, we show some of the possible way how to optimize the ICP algorithm.

k-d tree K-d trees are a special case of binary space partitioning trees that are often used as suitable structure in many applications, such as searches involving a multidimensional search key (e.g. range searches and nearest neighbor searches). ICP algorithm computation can be efficiently accelerated with its usage. Authors [11] declare that the usage of k-d tree can improve the speed of registration 10 times.

Corresponding points The original work [1] works with all the points in both cloud points. Certain enhancements have been made to the use of uniform sub-sampling of given points and random sampling [13]. The problem occurs in the case of noise or high curvature shapes, where these methods do not select enough reference points (see Figure 2, left). The solution is to use the normals for sampling [6]. The detection of corresponding points using normals is on Figure 2, right.

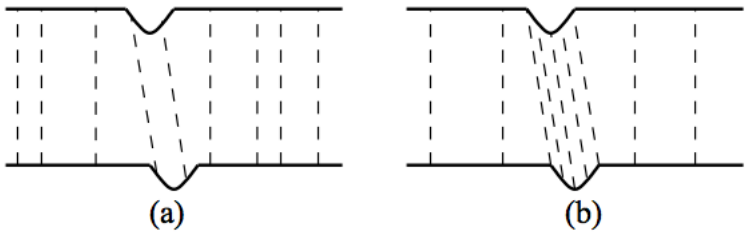

Fig. 2. Random sampling (left) and sampling with normals (right), source [17].

Point matching The algorithm [1] is based on the calculation of the Euclidean distance between points. However, this leads to significant computational demands, in the worst case, the algorithm has order $O(N_M N_S)$ of time complexity and average time complexity is $O(N_M\ logN_S)$. The use of point sorting in *kd*-tree leads to time complexity equal to $O(\ logN)$.

There are a number of methods to speed up the distance computation. The work [18] covers the most of possibilities of point matching methods. The normal method [8] is based on minimizing the distance not only of two points, but the approximate distance of the point from the plane. This computation reduces the number of iteration but the method is not so robust.

Another group of methods is based on projection of points from the original cloud (model) to the second cloud. The center of the projection can be given as the position of the camera that scans or generally. Then you can select a point of intersection or work with to points depending on the distance or intensity [19] or the color [16]. It is also possible to use the Hausdorff distance as in [5] that studies some theoretical results using Hausdorff distance in ICP.

It is also possible to use the SIFT feature point is used as corresponding point in the process of the improved ICP algorithm in order to reduce the error corresponding points. The reducing of the closest

point search improves the accuracy and efficiency of the ICP algorithm as described the authors in [21].

Point weighting The trivial weighting is that every point has same weight. However, it is more appropriate to use the distance of points as a measure of suitability. Depending on the distance of the points, we can either totally eliminate the pair if it exceeds a predetermined threshold or add a weight to the pair according to the formula [7]:

$$Weight = 1 - \frac{\text{dist}(P_1 P_2)}{\text{dist}_{max}} \tag{14}$$

We can also assign the weights in dependence to the direction of normal vectors. If they are parallel at both points, the scalar product is large number. With increasing size of angle-normal we get the scalar product smaller (the perpendicular vectors have a scalar product equal to zero). The weight assignment can be performed as:

$$Weight = \mathbf{n}_1 \mathbf{n}_2 \tag{15}$$

Correspondence between points The point clouds often contain the noise points and outliers. To avoid misalling of the scans, criteria are added to eliminate or at least decrease this problem. The usual recommendation is to remove pairs of points that contain points at the borders if one of the skins exceeds the other (Fig. 3).

Another option is to remove points that have a greater distance than a given threshold, to remove a certain percentage of points depending on the metric, or to remove points whose distance is greater than a given multiple of the standard deviation of the distance.

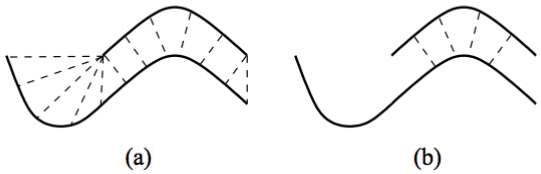

Fig. 3. The elimination of overlapping pairs, source [17].

3 Results

We tested the algorithm on the model point cloud in software Matlab, version 2017b [15]. We generated point clouds with random sets and we also added a noise. Tested point clouds have 3 sizes: 1000, 10 000 and 25 000 points. Then, we applied our algorithm ICP in Matlab. In Fig. 4, we can see the initial position of input point clouds and the registered position after application of ICP algorithm. The computation time and number of iterations is in Table 3.

Number of points	1000	10 000	25 000
Number of iteration	22	32	35
Time (s)	1.552	6.348	22.502

Tab. 1. Point cloud registration on tested point clouds.

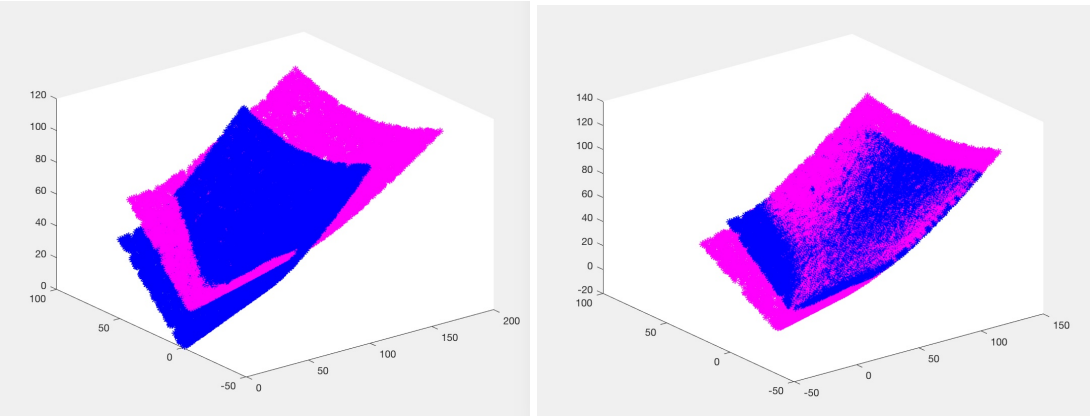

Fig. 4. Testing point cloud before (left) and after registration (right).

The implementation in Matlab consists of the steps described in 2.2. Matlab is suitable tool for ICP programming because of the easy use of matrices. We can also use the implemented functions as *svd*, *pdist* (Euclidean distance), etc. In implementation, we can choose the distance metrics (Euclidean, normal), threshold for ejecting noise points, type of weighting. The comparison of different parameters is difficult. It depends on the type of point clouds – their size, noise, overlapping region and shape. We also have different requirements on the result, we can prefer computation speed, precision or robustness. For example, the usage of normal distance computing increase the precision but it also decrease the robustness. Only the usage of k-d tree storage is strongly recommended. It can speed up the computation more than 10 times. Other parameters are optional and it depends on the input point clouds and future use of results.

4 Conclusion

Iterative closest point algorithm is the essential method of the point cloud registration. It is used for fine matching to ensure the best results. We present the overview of this method and also described the possible improvements of the algorithm. We tested the basic algorithm on the data. In future work, we would like to implement the possible improvements and compare the results.

Acknowledgment

This work was supported by Project LO1202 by financial means from the Ministry of Education, Youth and Sports under the National Sustainability Programme I.

References

[1] BESL, P. J., MCKAY, N.D.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 14 (2) (1992), 239–256.

[2] JIAN, B., VEMURI, B. C.: Robust Point Set Registration Using Gaussian Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 33(8), (2011), 1633–1645.

[3] CRACKNELL, A. P.: HAYES, L.:*Introduction to Remote Sensing* (2 ed.) London: Taylor and Francis. (2007).

[4] DIEZ, Y., ROURE, F., LLADO, X., SALVI J.: A Qualitative Review on 3D Coarse Registration Methods. ACM Comput. Surv. 47, 3, (2015), 54:1-45:36.

[5] EZRA, E., SHARIR, M., EFRAT, A.: On the performance of the ICP algorithm, Computational Geometry, vol. 41, 1–2, (2008), 77-93

[6] GELFAND, N., IKEMOTO, L., RUSINKIEWICZ, S. LEVOY, M.: Geometrically stable sampling for the ICP algorithm, in Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling, Banff, Canada, (2003) 260–67.

[7] GODIN, G. RIOUX, M. BARIBEAU, R.: Three-dimensional Registration Using Range and Intensity Information. Proc. SPIE: Videometrics III, vol. 2350, (1994), 279-290.

[8] CHEN, Y., MEDIONI, G.: Object modelling by registration of multiple range images. Image Vision Comput. (1992), 145–155.

[9] CHUA, C.S., JARVIS, R.: Point Signatures: A New Representation for 3D Object Recognition. International J. Computer Vision 25, 1, (1997), 63–85

[10] JOHNSON, A.E.: Spin-Images: A Representation for 3-D Surface Matching, (1997), Carnegie Mellon University Pittsburgh, Pennsylvania, Ph.D. thesis.

[11] LI, S., WANG, J., LIANG, Z., SU, L.: Tree point clouds registration using an improved ICP algorithm based on kd-tree. In Proceedings of 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 4545-4548.

[12] National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. Lidar 101: An Introduction to Lidar Technology, Data, and Applications. Revised. Charleston, SC: NOAA Coastal Services Center, (2012)

[13] TAKESHI, M.: Generation of geometric model by registration and integration of multiple range images. In Proceedings of the IEEE International Conference on 3D Digital Imaging and Modelling. (2001), 254–261.

[14] MATAS, J., CHUM, O., URBAN, M., PAJDLA, T. : Robust wide baseline stereo from maximally stable extremal regions. Proc. of British Machine Vision Conference, (2002), 384-396.

[15] Matlab software, Math Works, https://www.mathworks.com/products/matlab. html

[16] PULLI, K.: Surface Reconstruction and Display from Range and Color Data, (1997), University of Washington, Ph.D. thesis.

[17] RUSINKIEWICZ, S. LEVOY, M.: Efficient variants of the ICP algorithm. In Proceedings of the IEEE International Conference on 3D Digital Imaging and Modeling, (2001), 145–152.

[18] WANG, F., ZHAO, Z.: A survey of iterative closest point algorithm, In Procceding of Chinese Automation Congress (CAC), Jinan, China, (2017), 4395-4399.

[19] WEIK, S.: Registration of 3-D Partial Surface Models Using Luminance and Depth Information, In Proceeding of International Conference on Recent Advances in 3D Digital Imaging and Modeling, (1997), 93-100.

[20] ZHENG, Z., LI, Y.: LIDAR Data Registration for Unmanned Ground Vehicle Based on Improved ICP Algorithm. In Proceedings of 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Vol. 2 (2015), 554-558.

[21] ZHENG, Z., LI, Y., JUN, W.: LiDAR point cloud registration based on improved ICP method and SIFT feature, In Proceeding o IEEE International Conference on Progress in Informatics and Computing (PIC), Nanjing, (2015), 588-592.

[22] ZHONG, Y.: Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition, IEEE, 12th International Conference on Computer Vision Workshops, ICCV Workshops, (2009), 689-696.

Current address

Procházková Jana, Mgr. Ph.D.
Institute of Materials Science and Engineering
NETME centre
Brno University of Technology
Technická 2896/2 616 69 Brno, Czech Republic
E-mail: prochazkova.j@fme.vutbr.cz

Martišek, Dalibor, doc., PaedDr.
Institute of Materials Science and Engineering
NETME centre
Brno University of Technology
Technická 2896/2 616 69 Brno, Czech Republic
E-mail: martisek@fme.vutbr.cz