World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:5, 2021

PointNetLK-OBB: A Point Cloud Registration Algorithm with High Accuracy

Wenhao Lan, Ning Li, Qiang Tong

***Abstract*—**To improve the registration accuracy of a source point cloud and template point cloud when the initial relative deflection angle is too large, a PointNetLK algorithm combined with an oriented bounding box (PointNetLK-OBB) is proposed. In this algorithm, the OBB of a 3D point cloud is used to represent the macro feature of source and template point clouds. Under the guidance of the iterative closest point algorithm, the OBB of the source and template point clouds is aligned, and a mirror symmetry effect is produced between them. According to the fitting degree of the source and template point clouds, the mirror symmetry plane is detected, and the optimal rotation and translation of the source point cloud is obtained to complete the 3D point cloud registration task. To verify the effectiveness of the proposed algorithm, a comparative experiment was performed using the publicly available ModelNet40 dataset. The experimental results demonstrate that, compared with PointNetLK, PointNetLK-OBB improves the registration accuracy of the source and template point clouds when the initial relative deflection angle is too large, and the sensitivity of the initial relative position between the source point cloud and template point cloud is reduced. The primary contribution of this paper is the use of PointNetLK to avoid the non-convex problem of traditional point cloud registration and leveraging the regularity of the OBB to avoid the local optimization problem in the PointNetLK context.

***Keywords*—**Mirror symmetry, oriented bounding box, point cloud registration, PointNetLK-OBB.

I. Introduction

In recent years, as depth sensors have become more affordable, equipment that can directly obtain 3D data is becoming increasingly popular. As simple and effective representations of 3D data, 3D point clouds are widely used in robotics, autonomous driving, and virtual and augmented reality applications. Recently, 3D point cloud registration algorithms, which obtain the transformation matrix between the point clouds to be registered by estimating the corresponding relationship between them, have received considerable research attention [1]-[4]. These research studies primarily focus on the accuracy, time efficiency, and robustness of 3D point cloud registration, and various methods have been proposed. Previously proposed point cloud registration methods can be classified as traditional methods based on deep learning and methods based on local descriptors.

Besl et al. proposed an iterative closest point (ICP) algorithm, a classic 3D point cloud registration algorithm. Their proposed algorithm uses the least square method and singular value decomposition to estimate the transformation parameter between two point clouds to be registered and update the corresponding relationship between point clouds using the obtained transformation parameters. These steps are iterated until 3D point cloud registration is complete [5]. The ICP algorithm is often used in the fine registration stage. In subsequent research, Segal et al. proposed Generalized-ICP, which combined ICP and Point-to-Plane ICP and improved the algorithm's robustness against noise and occlusion [6]. Bouaziz et al. redefined ICP by using sparse induced norm, and improved the robustness of ICP for external and incomplete point cloud data [7]. Biber et al. used normal distribution transformation and Newton's method to simplify the process of exploring the correspondence between point clouds [8]. Agamennoni et al. applied statistical inference technology to the full probability model, proposed a new probabilistic data association strategy, and improved the robustness of registration algorithm for external points and noises [9]. Yang et al. proposed the TEASER method, which eliminated the interference of external points by using truncated least squares and semi-definite relaxation optimization [10]. However, due to non-convex problems, registration accuracy is highly dependent on the initial relative positions of the source and template point clouds. If the initial relative deflection angle between the clouds to be registered is too large, the ICP will fall into local optimum and registration accuracy will be reduced. To find the global optimal solution, Yang et al. proposed Go-ICP, which used the branch-and-bound method to search the 3D transformation space [11]. Although Go-ICP can obtain the global optimal solution, the time complexity of the algorithm is several times that of ICP and its improved algorithm; thus, Go-ICP cannot meet real-time requirements. Other studies on global optimal solutions for point cloud registration, investigate mixed integer programming, Riemann optimization, and convex relaxation [12]-[14].

The traditional 3D point cloud registration method is both a non-convex and computationally inefficient problem. The traditional method needs to clearly estimate the correspondence between point clouds and computational efficiency will decrease as the number of points to be registered increases.

This study was supported by Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University.

Wenhao Lan is with the School of Automation, Beijing Information Science and Technology University, Beijing 100101, China (phone: +8618515280911; e-mail: cn_alex_2018@ outlook.com).

Ning Li was with University of Kent, Canterbury, England. He is now with the School of Computer, Beijing Information Science and Technology University, Beijing 100101, China (corresponding author, e-mail: ningli.ok@163.com).

Qiang Tong was with Department of computer science and technology, Tsinghua university, Beijing, China. He is now with the School of Computer, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: tongq85@bistu.edu.cn).

With the development of deep learning technology, solutions that address the problems with traditional 3D point cloud registration methods have been proposed [15]-[17].

Aoki et al. introduced deep learning into 3D point cloud registration. They proposed PointNetLK, which uses the PointNet deep neural network and the modified LK algorithm to complete 3D point cloud registration [18]-[20]. PointNetLK outperformed the traditional algorithm in terms of registration efficiency. Inspired by PointNetLK, PCRNet [21] used PointNet to extract 3D point clouds features and adopted a neural network with Siamese architecture to concatenate feature vectors among point clouds. The learning method was used to obtain point cloud transformation parameters non-iteratively, which improved the anti-interference ability of the algorithm to some extent.

To find the correspondence between source and template point clouds, some researchers have attempted to design local descriptors, such as scale-invariant curvature descriptors and FPFH [22], [23]. Due to the limitation of manual design features, some studies investigated using deep learning to extract 3D local descriptors. For example, 3DMatch employs self-supervised learning of local feature descriptors from RGBD 3D reconstruction data [24]. PPFNet combines the local and global characteristics of point clouds to train local descriptors with global information [25], and Ppf-foldnet uses unsupervised learning to obtain local descriptors with rotation invariance [26]. However, these methods are not generalizable and can only be applied to specific scenarios.

PointNetLK, which is based on deep learning, uses the PointNet network to reduce the dimension of the point cloud so that the time complexity of the algorithm does not increase as the number of points increases. PointNetLK can be extended to a wide range of application scenarios. PointNetLK considers PointNet as the "imaging" function of a 3D point cloud and uses the 2D optical flow tracking algorithm LK to align the obtained "image" and avoids the non-convex problem of the traditional 3D point cloud registration algorithm. However, PointNet does not have a convolutional structure; thus, the extracted point cloud features lack local information on the point cloud surface and will fall into another form of the non-convex problem. The direct result is that when the deviation angle of two point clouds is too large, registration accuracy will be reduced.

This paper attempts to improve the PointNetLK algorithm to solve this problem.

II. Original PointNetLK Algorithm

To reduce the computational cost of point cloud registration, PointNetLK uses deep learning technology to extract point cloud features and reduce the dimension of the point cloud. This process can be understood as "imaging" a point cloud. After obtaining the "image" of the source and template point clouds, PointNetLK uses LK, a 2D optical flow tracking algorithm, to obtain the transformation parameters between the two "images"; thus deriving the twist parameters between the 3D point clouds and completing the registration. However, the PointNet network, which is used to extract point cloud features, does not have a convolution structure, and only the macroscopic point cloud features with coordinate information can be obtained. Consequently, PointNet becomes a multi-peak "imaging" function, as shown in Fig. 3, and, as a result, PointNetLK falls into a non-convex optimization problem.

In the following, the matrix and 3D point set are represented by upper case bold and italicized letters (e.g., $\boldsymbol{P}$), column vectors and n-dimensional points are represented in lowercase bold italics (e.g., $\boldsymbol{t}$), constants are represented in uppercase italics (e.g., M), and scalars are represented in lowercase italics (e.g., j).

A. General Target of 3D Point Cloud Registration

We use $\boldsymbol{P}=\{\boldsymbol{p}_i \mid \boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^{N}$ and $\boldsymbol{Q}=\{\boldsymbol{q}_j \mid \boldsymbol{q}_j \in \mathbb{R}^3\}_{j=1}^{M}$ to represent 3D point clouds, where M and N represent the number of 3D points. For 3D point clouds, the registration task is to find rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and translation vector $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ so as to minimize the sum of the distance between corresponding points, as shown in:

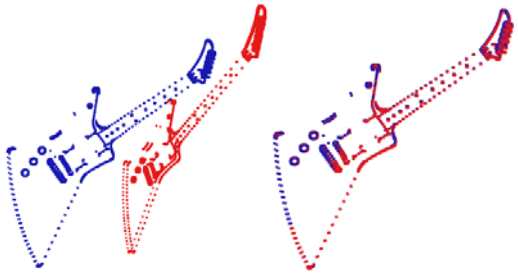
$$\arg\min_{\boldsymbol{R},\boldsymbol{t}} \sum_{k=1}^{L} \left\| p_k - (\boldsymbol{R}q_k + \boldsymbol{t}) \right\|_2 \tag{1}$$

where $L=\min\{M, N\}$.

The rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ are usually expressed in homogeneous form as:

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}_{4\times4} \tag{2}$$

The expected effect of 3D point cloud registration is shown in Fig. 1. The traditional registration algorithm requires all points in the point cloud to participate in the registration process; thus, the run time of the algorithm increases relative to the scale of the point cloud. By leveraging deep learning technology, features suitable for the registration target can be extracted from the point cloud. With this technology, the dimension of the features is far smaller than the scale of the original point cloud, which is expected to reduce the computational cost and solve the multi-extremum optimization problem that occurs in the traditional iterative registration algorithm, i.e., the non-convex problem.



(a) Before registration (b) After registration

Fig. 1 Registration process of guitar point cloud

B. Point Cloud Feature Extraction in PointNetLK

As described in Section II *A*, 3D point cloud $\boldsymbol{P}$ and $\boldsymbol{Q}$ are represented in matrix form, i.e., $\boldsymbol{P} \in \mathbb{R}^{3\times N}$ and $\boldsymbol{Q} \in \mathbb{R}^{3\times M}$.

PointNet is considered an imaging function of a 3D point cloud: $\Psi : \mathbb{R}^{3\times N} \to \mathbb{R}^{K}$. Function Ψ applies a multi-layer perceptron (MLP) to each point in the 3D point cloud, which transforms N 3D points into N k-dimensional points, i.e., $\mathbb{R}^{3\times N} \to \mathbb{R}^{K\times N}$, and then performs the max pooling operation. From the N k-dimensional points, the most significant point is selected as the k-dimensional global feature vector of the 3D point cloud, i.e., $\mathbb{R}^{K\times N} \to \mathbb{R}^{K}$, the "image" of the 3D point cloud.

Although Ψ plays a role in the dimensional reduction of the 3D point cloud, thereby reducing computational costs, it is a MLP deep neural network that lacks convolutional structure and cannot extract local microscopic point cloud surface features. This will result in $\Psi(\boldsymbol{P})$ and $\Psi(\boldsymbol{Q})$ reflecting only the macroscopic contour features of point clouds $\boldsymbol{P}$ and $\boldsymbol{Q}$. The LK algorithm cannot capture the difference in surface detail coordinate information between the point clouds to be registered. Consequently, registration tends to fall into non-convex problems and remain in the locally optimal state.

C. Exponential Mapping: The Link between Optical Flow Tracing and Point Cloud Transformation

In Section II *A*, the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ are used to represent the coordinate transformation $\boldsymbol{T}$ in 3D space. To easily obtain the updated parameters of the coordinate transformation matrix using the LK algorithm, the exponential mapping form is used to represent the coordinate transformation in 3D space. The spatial coordinate transformation matrix $\boldsymbol{T}$ is expressed as:

$$\boldsymbol{T} = \exp\left(\sum_{i=1}^{6} \delta_i \boldsymbol{G}_i\right) \quad \boldsymbol{\delta} = (\delta_1, \delta_2, ..., \delta_6) \tag{3}$$

Here, $\boldsymbol{G}_i$ is as follows:

$$\boldsymbol{G}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \boldsymbol{G}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \boldsymbol{G}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{G}_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \boldsymbol{G}_5 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \boldsymbol{G}_6 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\boldsymbol{G}_i$ is the generating operator of exponential mapping, which corresponds to the rotation parameter δ_i one by one. If $\boldsymbol{P}$ is considered a template point cloud and $\boldsymbol{Q}$ is considered a source point cloud, the target of 3D point cloud registration can be transformed into finding the best 3D space coordinate transformation $\boldsymbol{T}$, i.e.,

$$\Psi(\boldsymbol{P}) = \Psi(\boldsymbol{T} \bullet \boldsymbol{Q}) \tag{4}$$

Here, $\bullet$represents the coordinate transformation of source point cloud $\boldsymbol{Q}$ in 3D space.

D. Combination of Registration Target and Optical Flow Tracking

During the registration process, the source point cloud's coordinates change dynamically. With reference to the inverse combinational LK algorithm [27], the iterative process of the algorithm can be completed by calculating the 3D coordinate change rate of the template point cloud $\boldsymbol{P}$ with respect to the infinitesimal twist parameter $\boldsymbol{\delta}$ only once, thus reducing the computational cost. According to this idea, the form of the registration target is changed to:

$$\Psi(\boldsymbol{Q}) = \Psi\left(\boldsymbol{T}^{-1} \bullet \boldsymbol{P}\right) \tag{5}$$

The right side of (5) can be expanded as:

$$\Psi(\boldsymbol{Q}) = \Psi(\boldsymbol{P}) + \frac{\partial}{\partial \boldsymbol{\delta}}\left[\Psi\left(\boldsymbol{T}^{-1} \bullet \boldsymbol{P}\right)\right]\boldsymbol{\delta} \tag{6}$$

where $\boldsymbol{T}^{-1}$ is as follows:

$$\boldsymbol{T}^{-1} = \exp\left(-\sum_{i=1}^{6} \delta_i \boldsymbol{G}_i\right) \tag{7}$$

For the convenience of computing, the finite difference gradient is used to approximate the gradient of Ψ for the twist parameter $\boldsymbol{\delta}$. Jacobian matrix is represented by $\boldsymbol{J}$:

$$\boldsymbol{J} = \frac{\partial}{\partial \boldsymbol{\delta}}\left[\Psi\left(\boldsymbol{T}^{-1} \bullet \boldsymbol{P}\right)\right] \tag{8}$$

Then, the calculation process of column $\boldsymbol{J}_i$ in $\boldsymbol{J}$ is as:

$$\boldsymbol{J}_i = \frac{\Psi\left(\exp(-\delta_i \boldsymbol{G}_i) \bullet \boldsymbol{P}\right) - \Psi(\boldsymbol{P})}{\delta_i} \tag{9}$$

where δ_i takes a very small, fixed value rather than an infinitesimal value. The twist parameter $\boldsymbol{\delta}$ can be calculated as:

$$\boldsymbol{\delta} = \boldsymbol{J}^{+}\left[\Psi(\boldsymbol{Q}) - \Psi(\boldsymbol{P})\right] \tag{10}$$

where $\boldsymbol{J}^{+}$ is the generalized inverse of $\boldsymbol{J}$.

In the iterative registration process, the twist parameter $\boldsymbol{\delta}$ is used to update the source point cloud $\boldsymbol{Q}$ in each iteration, as:

$$\Delta\boldsymbol{T} \bullet \boldsymbol{Q} \to \boldsymbol{Q} \quad \Delta\boldsymbol{T} = \exp\left(\sum_{i=1}^{6} \delta_i \boldsymbol{G}_i\right) \tag{11}$$

After the PointNetLK iteration phase, the 3D coordinate

transformation matrix of the source point cloud can be expressed as:

$$\boldsymbol{T}_{\text{PointNetLK}} = {}_{\triangle}\boldsymbol{T}_n \bullet \ldots \bullet {}_{\triangle}\boldsymbol{T}_1 \bullet {}_{\triangle}\boldsymbol{T}_0 \tag{12}$$

PointNetLK avoids the non-convex problem of traditional iterative registration. However, because PointNet does not have a convolutional structure, only the macroscopic features of the point cloud containing coordinate information can be extracted. When the point cloud deflection angle is too large, the registration algorithm will fall into a new local optimal problem. The intuitive registration result is shown in Fig. 2.

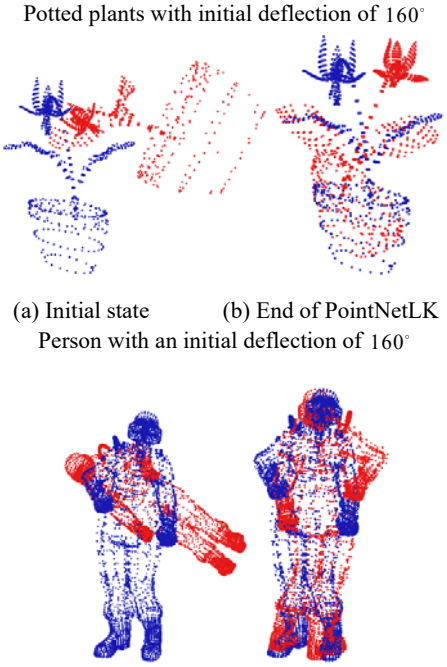


Fig. 2 Registration results of point clouds with larger deflection angle in PointNetLK phase

We consider PointNet as a function $\Psi(\boldsymbol{\delta})$ of the twist parameter $\boldsymbol{\delta}$. When the initial deflection angle of the source and template point clouds is too large, using the LK algorithm to iteratively update $\boldsymbol{\delta}$ will result in at least two different twist parameters $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$, making $\Psi(\boldsymbol{\delta}_1) = \Psi(\boldsymbol{\delta}_2)$. In this case, the right side of (8) is zero, and $\boldsymbol{\delta}$ stops updating. The registration algorithm falls into a local optimum, and registration accuracy is lost, as shown in Fig. 3.

III. Improvement of PointNetLK Algorithm

As shown in Fig. 3, PointNetLK falls into a local optimum when the initial relative deflection angle of the source and template point clouds is too large. This paper proposes the PointNetLK-OBB algorithm to address this problem. Based on PointNetLK, PointNetLK-OBB uses a 3D geometry tool, i.e., an OBB, that can describe the global regularity of point clouds [28]. Under the guidance of ICP, it avoids the non-convex problem associated with PointNetLK. Therefore, the registration accuracy between point clouds with excessively large deflection angles is improved.

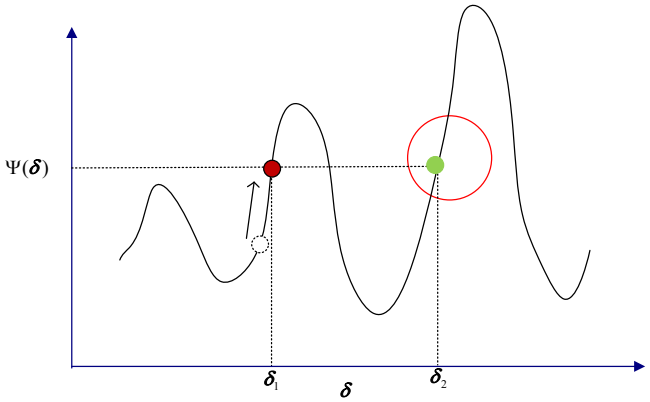


Fig. 3 Schematic diagram of PointNetLK falling into local optimum when updating $\boldsymbol{\delta}$ with the LK algorithm

A. OBB Construction

The OBB refers to the smallest volume that can enclose a 3D point cloud. It is usually represented by a center point $\boldsymbol{c} \in \mathbb{R}^3$, three orthogonal unit vectors $\{\boldsymbol{u} \in \mathbb{R}^3, \boldsymbol{v} \in \mathbb{R}^3, \boldsymbol{w} \in \mathbb{R}^3\}$ representing directions, and three scalar variables $\{l, w, h\}$ representing the length, width, and height extension radius, respectively.

Similar to an OBB, the Axis Aligned Bounding Box (AABB) of a 3D point cloud is a geometric description tool in 3D space. We consider the OBB and AABB of the general point cloud $\boldsymbol{X} \in \mathbb{R}^{3\times N}$, denoted OBB_X and AABB_X respectively. OBB_X can be considered the AABB of the point cloud $\boldsymbol{X}$ after a certain space rotation $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$:

$$\text{OBB}_X = \text{AABB}_{X'}, \qquad \boldsymbol{X}' = \boldsymbol{R} \bullet \boldsymbol{X} \tag{13}$$

Therefore, the problem of solving the OBB is transformed into finding a suitable 3D rotation $\boldsymbol{R} \in SO(3,\mathbb{R})$ from the special orthogonal group such that the AABB volume is the smallest:

$$\min_{\boldsymbol{R}\in SO(3,\mathbb{R})} (l \times w \times h) \tag{14}$$
$$\text{s.t.} \begin{bmatrix} -l \\ -w \\ -h \end{bmatrix} \le \boldsymbol{R} \bullet \boldsymbol{x}_i - \boldsymbol{c} \le \begin{bmatrix} l \\ w \\ h \end{bmatrix}, \boldsymbol{x}_i \subset \boldsymbol{X}, \forall i \in \{1,\ldots,N\}$$

The OBB solution uses the hybrid bounding box rotation identification method [29].

B. Combination of 3D Point Cloud Registration Target and OBB

Equation (12) gives the spatial transformation $\boldsymbol{T}_{\text{PointNetLK}}$ that can align $\boldsymbol{P}$ and $\boldsymbol{Q}$ subject features obtained in the

PointNetLK stage. Here, the source point cloud updated with $T_{\text{PointNetLK}}$ is denoted Q':

$$Q' = T_{\text{PointNetLK}} \bullet Q \tag{15}$$

The OBBs of Q' and P are denoted $\text{OBB}_{Q'}$ and OBB_P, respectively. To use the regularity of OBB to complete the 3D point cloud registration task, it is necessary to align $\text{OBB}_{Q'}$ and OBB_P. To align $\text{OBB}_{Q'}$ and OBB_P, ICP is used to align the vertices of $\text{OBB}_{Q'}$ and OBB_P. Here, let the vertices of $\text{OBB}_{Q'}$ be $Q'_{\text{OBB}} \in \mathbb{R}^{3\times 8}$, and the vertices of OBB_P be $P_{\text{OBB}} \in \mathbb{R}^{3\times 8}$.

We consider general point cloud X. The vertices of OBB_X can be considered vertices of $\text{AABB}_{X'}$ in (10), which has undergone spatial rotation R; thus, the eight vertices of OBB_X are given as:

$$R^{-1} \bullet \left(\begin{bmatrix} (-1)^{(i/2)} \times l \\ (-1)^{((i+3)/2)} \times w \\ (-1)^{((i-1)/4)} \times h \end{bmatrix} + c \right), \; i = 1,...,8 \tag{16}$$

where / is the divisor sign, i.e., the integer part of the quotient is taken, and the decimal part is erased. In addition, $\begin{bmatrix} l \\ w \\ h \end{bmatrix}$ is determined by the right-hand rule of the 3D Cartesian coordinate system.

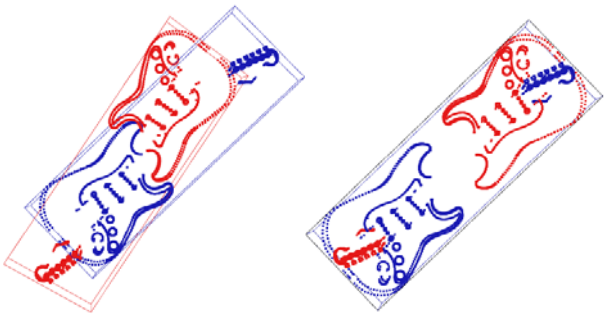
Through experiments, it was found that the registration task of Q'_{OBB} and P_{OBB} can be completed quickly and accurately using ICP because Q'_{OBB} and P_{OBB} only contain eight points, and their shapes are regular. The maximum point-to-point distance of the ICP should be set to one-half the length of the longest side of the OBB, which will improve the registration result. The space transformation matrix aligning Q'_{OBB} and P_{OBB} is denoted T_{ICP}:

$$P_{\text{OBB}} = T_{\text{ICP}} \bullet Q'_{\text{OBB}} \tag{17}$$

The OBB of the point cloud depicts the contour of the point cloud; thus, the spatial transformation applied to the eight vertices of the OBB can also be applied to each point in the point cloud. Based on (15), spatial transformation T_{ICP} is applied continually to source point cloud Q' at the end of the PointNetLK stage. Here, the updated Q' is denoted Q'':

$$Q'' = T_{ICP} \bullet Q' \tag{18}$$

When the initial relative deflection angles of P and Q are small, T_{ICP} can achieve fine registration of P and Q. Determining whether P and Q complete fine registration must be assessed according to the fitting degree of P and Q. When the initial relative deflection angles between P and Q are large, the effect achieved is shown in Fig. 4.



(a) Before aligning OBB (b) After aligning OBB

Fig. 4 OBB alignment process of guitar point cloud

The OBB of source point cloud Q'' updated by spatial transformation T_{ICP} is denoted $\text{OBB}_{Q''}$. As shown in Fig. 4 (a), the main features of the guitar point cloud have been aligned at the end of the PointNetLK registration stage; however, they are not sufficiently regular and are in a local optimal state in the non-convex optimization problem. Using a regular OBB that can describe the main features of the point cloud can help the point cloud registration algorithm avoid the local optimum and improve registration accuracy.

As shown in Fig. 4 (b), after aligning $\text{OBB}_{Q''}$ and OBB_P, Q'' and P are in mirror symmetry. Fig. 4 (b) shows the source point cloud Q'' and target point cloud P in a mirror symmetry state, and their symmetry plane passes through the center point of the OBB and is parallel to the side surface of the OBB. To restore the point cloud Q'' and P from the mirror state to the fully fitted state, Q'' must be centered on the center $c \in \mathbb{R}^3$ of the OBB, where the unit vector $u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \in \mathbb{R}^3$ represents the OBB direction and is parallel to the long side of the OBB as the axis of rotation (rotating by π radians) This space transformation matrix is denoted T_{mirror}:

$$T_{\text{mirror}} = T_2 \bullet T_1 \bullet T_0 \tag{19}$$

$$Q''' = T_{\text{mirror}} \bullet Q'' \tag{20}$$

$$Q''' \approx P \tag{21}$$

where

$$T_0 = \begin{pmatrix} 0_{3\times 3} & -c_{3\times 1} \\ 0_{1\times 3} & 1 \end{pmatrix},$$

$$\boldsymbol{T}_1 = \begin{pmatrix} \left(\exp\left(\theta \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \right) \right)_{3\times3} & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{pmatrix},$$

$$\boldsymbol{T}_2 = \begin{pmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{c}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{pmatrix}.$$

The approximately equal relationship in (189 is due to the presence of noise and sensor error.

The final spatial transformation matrix required to complete the registration task can be obtained as:

$$\boldsymbol{T}_{\text{final}} = \boldsymbol{T}_{\text{mirror}} \bullet \boldsymbol{T}_{\text{ICP}} \bullet \boldsymbol{T}_{\text{PointNetLK}} \tag{22}$$

In the process of obtaining $\boldsymbol{T}_{\text{mirror}}$, mirror symmetry planes between the template point cloud and source point cloud may have three situations in which the unit orthogonal vectors $\{\boldsymbol{u}\in\mathbb{R}^3, \boldsymbol{v}\in\mathbb{R}^3, \boldsymbol{w}\in\mathbb{R}^3\}$ representing the OBB direction are used as the normal plane. Therefore, there are three calculation results for $\boldsymbol{T}_{\text{mirror}}$, from which the best T is selected such that $\boldsymbol{P}$ and $\boldsymbol{T}_{\text{mirror}} \bullet \boldsymbol{Q}''$ have the highest degree of fitting.

OBB can express the macroscopic feature of the 3D point cloud more regularly, as shown in Fig. 5, thereby providing global point cloud information for the 3D point cloud registration task. PointNetLK-OBB uses OBB to avoid local optima in the PointNetLK context, increase registration speed, and improve registration accuracy.

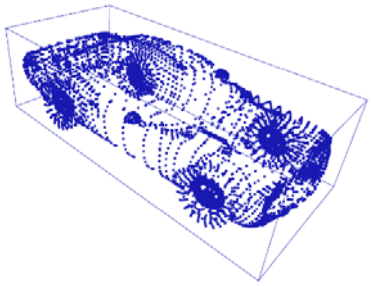


Fig. 5 Orientation bounding box of car point cloud

C. Training PointNetLK-OBB

To ensure that PointNet can extract features from point cloud data, it is necessary to train the PointNet classification network. However, by training only PointNet, the features extracted by the network are not suitable for point cloud registration tasks. To ensure that PointNet can extract global features from the point cloud data that are suitable for the registration task and lay the foundation for subsequent high-accuracy registration, it is necessary to train PointNetLK based on training the PointNet classification network to fine-tune the PointNet network parameters.

The algorithm after PointNetLK does not use the point cloud features extracted by the PointNet imaging functions Ψ. Instead, it directly uses the point cloud itself to complete the final high-accuracy registration; therefore, when training the algorithm model, only the PointNetLK component is trained.

The loss function used to train PointNetLK is the two-norm of the feature difference between the source point cloud $\boldsymbol{Q}$ and the target point cloud $\boldsymbol{P}$ extracted by the PointNet imaging function Ψ, and the difference between the transformation matrix estimated by PointNetLK and the real transformation matrix F Norm is given as:

$$\left\|\Psi(\boldsymbol{P}) - \Psi(\boldsymbol{Q})\right\|_2 + \left\|\left(\boldsymbol{T}_{\text{PointNetLK}}\right)^{-1} \bullet \boldsymbol{T}_{\text{groundtruth}} - \boldsymbol{I}_4\right\|_{\text{F}} \tag{23}$$

IV. Experiment

A. Dataset

ModelNet40 was used as the experimental dataset [30]. ModelNet40 is an open dataset containing a total of 40 categories of CAD models, and each category comprises training data and test data. PointNetLK uses ModelNet40 to test and analyze algorithm performance; thus, we employed this dataset to facilitate comparison with PointNetLK.

B. Registration Error under Different Initial Deflection Angles

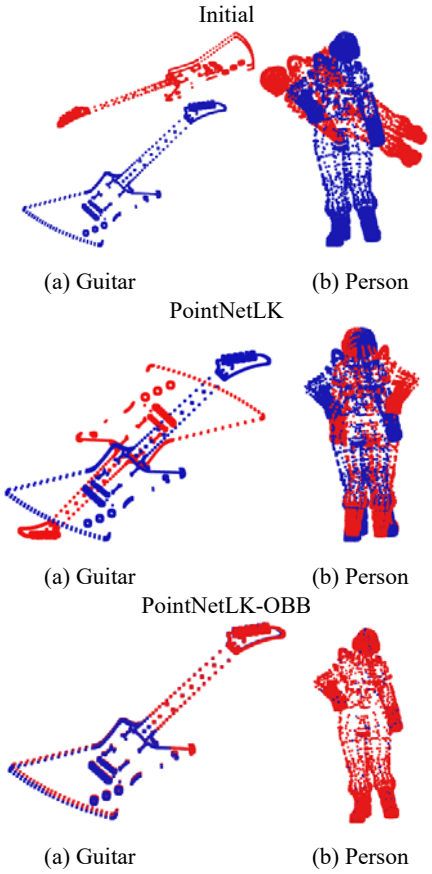


Fig. 6 Comparison of registration results between PointNetLK and PointNetLK-OBB

To verify the effectiveness of PointNetLK-OBB, which is improved PointNetLK by combining the oriented bounding box, the proposed was compared to the PointNetLK algorithm

on the Modelnet40 dataset, and the intuitive comparison effect is shown in Fig. 6. The results demonstrate that the improved PointNetLK combined with the oriented bounding box can effectively reduce registration error when the deflection angle of the point clouds to be registered is large.

We used PointNetLK and PointNetLK-OBB to experiment on the ModelNet40 test set. Here, to observe the registration performance of the algorithm in different initial states, the initial deflection angles of the source point cloud and template point cloud used in the experiment were spaced at 10° intervals from 0° to 180° uniformly distributed, and the displacement was selected randomly in the range $[0, 0.3]$. The product $\boldsymbol{T}_{\text{error}}=\boldsymbol{T}_{\text{final}}\bullet(\boldsymbol{T}_{\text{groundtruth}})^{-1}$ of the estimated transformation matrix $\boldsymbol{T}_{\text{final}}$ and inverse of the real transformation matrix $(\boldsymbol{T}_{\text{groundtruth}})^{-1}$ was used as the algorithm evaluation criterion. To verify the overall performance of the algorithm on the dataset, we used the mean error of all test data under the same deflection angle to represent the final performance of the algorithm.

The first set of observation data was the mean position error of the algorithm. Here, we used the norm of $\boldsymbol{t}$ in $\boldsymbol{T}_{\text{error}}$ (2) to represent the position error, and the final position error was averaged from the position errors of all test data under the same deflection angle. The final position error under different deflection angles is shown in Fig. 7.

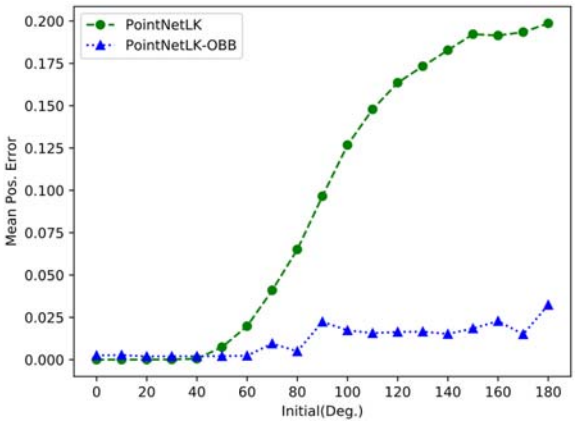


Fig. 7 Mean position error of PointNetLK-OBB and PointNetLK with different initial deflection angles

As shown in Fig. 7, the initial deviation angle between the point clouds to be registered increased, the mean position error of PointNetLK continued to increase, and PointNetLK-OBB always maintained a low error level.

The second set of observation data was the mean angle error of the algorithm. Here, we used $\boldsymbol{\delta}_{\text{error}}$ rather than $\boldsymbol{T}_{\text{error}}$ (3), and the first three components $\delta_1,\delta_2,\delta_3$ of $\boldsymbol{\delta}_{\text{error}}$ represented the rotation angle error, and the final angle error was averaged from the rotation angle errors of all test data under the same deflection angle. The final angle error at different deflection angles is shown in Fig. 8.

As shown in Fig. 8, the initial deflection angle between the point clouds to be registered increased, the mean rotation angle error of PointNetLK increased gradually, and PointNetLK-OBB always maintained a low error level. Due to the accumulation of errors, the angle error of PointNetLK-OBB increased when the initial deflection angle between the point clouds to be registered was 180°. However, this error was still at a low level.

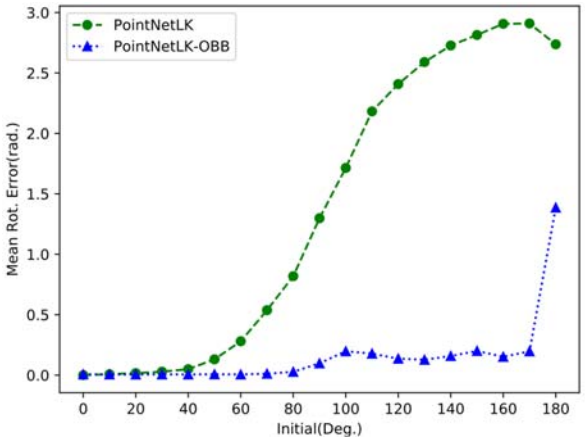


Fig. 8 Average rotation angle error of PointNetLK-OBB and PointNetLK with different initial deflection angles

PointNetLK uses the fully-connected PointNet network to extract only the point cloud macro contour features; thus, when the initial deflection angle of the point cloud to be registered is large, there is at least one point cloud feature in the non-target pose that is the same as the target pose feature, resulting in LK. When the algorithm is in the non-target state, it stops updating the rotation parameters (10), and the registration algorithm falls into a new local optimum (Fig. 3). Therefore, the greater the deflection angle of the point cloud, the more non-target poses with the same feature as the target pose point cloud, which results in greater PointNetLK registration error.

With PointNetLK-OBB, the non-convex problem of the traditional registration algorithm is eliminated, the main component in the point cloud is aligned, and the regularity of the OBB is used to align the OBB using the ICP algorithm. The frame uses the mirror symmetry effect to flip the point cloud, thereby avoiding the new local optimization problem caused by PointNet and (10) and effectively reducing registration error when the point cloud deflection angle is too large.

C. Registration Time Performance under Different Initial Deflection Angles

To test the time performance of the PointNetLK-OBB algorithm, we compared it to the original PointNetLK algorithm on an Intel Xeon 2.1 GHz CPU. The results demonstrate that the run time of the proposed algorithm is increased slightly compared to the original algorithm.

We used PointNetLK and PointNetLK-OBB to experiment on the ModelNet40 test set. The deflection angles of the source point cloud and template point cloud in this experiment ranged from 0° to 180° in 10° intervals uniformly distributed, and the displacement was selected randomly in the range $[0, 0.3]$. To test the overall time performance of the algorithm on this

dataset, we used the average run time of all test data under the same deflection angle to represent the final algorithm time performance, as shown in Fig. 9.

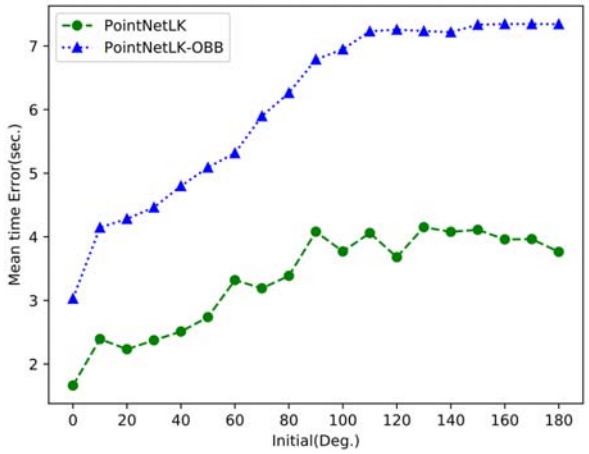


Fig. 9 Average run time of PointNetLK-OBB and PointNetLK with different initial deflection angles

As shown in Fig. 9, the run time of PointNetLK-OBB and PointNetLK increased as the initial deflection angle increased. The run time of PointNetLK-OBB was twice that of PointNetLK. Although the run time of PointNetLK-OBB increased relative to PointNetLK, by comparing Figs. 8 and 9, we find that the run time of PointNetLK-OBB is acceptable relative to completing the registration task at high accuracy.

To ensure sufficient registration accuracy, PointNetLK-OBB uses the regularity of OBB to complete the registration based on PointNetLK, which increases time costs. PointNetLK-OBB uses ICP to align the eight vertices of the OBB and creates a mirror symmetry effect between the point clouds to be registered. The point cloud to be registered in the mirror symmetry state only needs to rely on the rotation of the point cloud itself to complete high-accuracy point cloud registration. Here, while improving accuracy, the time cost has not increased significantly.

V. Conclusion

To address the problem where PointNetLK falls into local optimum and loses accuracy when the deviation angle of the point cloud to be registered is too large, this paper has proposed the PointNetLK-OBB point cloud registration method, which is improved by combining the OBB. The proposed method uses PointNetLK to escape the non-convex problem of traditional point cloud registration, completes the alignment of the principal components of the point cloud, uses the regularity of the 3D point cloud OBB, and, under guidance of ICP, and produces mirror symmetry between the point clouds. In conjunction with the rotation of the source point cloud itself, the effect avoids the local optimization problem caused by PointNet lacking a convolution structure.

The improved registration PointNetLK-OBB algorithm was compared to the original algorithm PointNetLK on the public 3D ModelNet40 dataset. The experimental results demonstrate that PointNetLK-OBB can avoid the local optimum in the context of PointNetLK when the deviation angle of the point cloud to be registered is large, which ensures time efficiency, and completes the 3D point cloud registration at high accuracy.

Some readers may argue that PointNetLK can be replaced with OBB and ICP, which can produce a mirror symmetry effect. We consider that the point cloud registration cannot be completed using only the OBB and ICP. The method of the OBB series requires the support of ICP; however, ICP is highly dependent on the initial position of the point clouds to be registered, and it is easy to fall into the traditional non-convex problem. The proposed method solves this problem well. Here, it uses PointNetLK to break out of the traditional non-convex problem and uses the regularity of the direction bounding box to avoid the local optimization problem in the context of PointNetLK. However, we found that PointNetLK-OBB can only register point clouds of the same scale. Thus, future improvements can include expanding the OBB to complete the point cloud registration with inconsistent scales.

The proposed algorithm can currently be applied to AR medical care, and, in future, can also be extended to other applications, e.g., robot path planning, autonomous driving, and 3D reconstruction.

References

[1] Choy C, Park J, Koltun V. Fully convolutional geometric features (C)//Proceedings of the IEEE International Conference on Computer Vision. 2019: 8958-8966.
[2] Eckart B, Kim K, Kautz J. Fast and accurate point cloud registration using trees of gaussian mixtures (J). arXiv preprint arXiv:1807.02587, 2018.
[3] Gojcic Z, Zhou C, Wegner J D, et al. The perfect match: 3d point cloud matching with smoothed densities (C)//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 5545-5554.
[4] Wang Y, Solomon J M. Deep closest point: Learning representations for point cloud registration (C)//Proceedings of the IEEE International Conference on Computer Vision. 2019: 3523-3532.
[5] Besl P J, McKay N D. Method for registration of 3-D shapes (C)//Sensor fusion IV: control paradigms and data structures. International Society for Optics and Photonics, 1992, 1611: 586-606.
[6] Segal A, Haehnel D, Thrun S. Generalized-icp (C)//Robotics: science and systems. 2009, 2(4): 435.
[7] Bouaziz S, Tagliasacchi A, Pauly M. Sparse iterative closest point (C)//Computer graphics forum. Oxford, UK: Blackwell Publishing Ltd, 2013, 32(5): 113-123.
[8] Biber P, Straßer W. The normal distributions transform: A new approach to laser scan matching (C)//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453). IEEE, 2003, 3: 2743-2748.
[9] Agamennoni G, Fontana S, Siegwart R Y, et al. Point clouds registration with probabilistic data association (C)//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 4092-4098.
[10] Yang H, Shi J, Carlone L. Teaser: Fast and certifiable point cloud registration (J). arXiv preprint arXiv:2001.07715, 2020.
[11] Yang J, Li H, Campbell D, et al. Go-ICP: A globally optimal solution to 3D ICP point-set registration (J). IEEE transactions on pattern analysis and machine intelligence, 2015, 38(11): 2241-2254.
[12] Rosen D M, Carlone L, Bandeira A S, et al. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group (J). The International Journal of Robotics Research, 2019, 38(2-3): 95-125.
[13] Maron H, Dym N, Kezurer I, et al. Point registration via efficient convex relaxation (J). ACM Transactions on Graphics (TOG), 2016, 35(4): 1-12.
[14] Izatt G, Dai H, Tedrake R. Globally optimal object pose estimation in point clouds with mixed-integer programming (M)// Robotics Research. Springer, Cham, 2020: 695-710.

[15] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space (C)//Advances in neural information processing systems. 2017: 5099-5108.
[16] Phan A V, Le Nguyen M, Nguyen Y L H, et al. DGCNN: A convolutional neural network over large-scale labeled graphs (J). Neural Networks, 2018, 108: 533-543.
[17] Wu W, Qi Z, Fuxin L. Pointconv: Deep convolutional networks on 3d point clouds (C)//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 9621-9630.
[18] Aoki Y, Goforth H, Srivatsan R A, et al. Pointnetlk: Robust & efficient point cloud registration using pointnet (C)//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 7163-7172.
[19] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation (C)//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
[20] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision (J). 1981.
[21] Sarode V, Li X, Goforth H, et al. Pcrnet: point cloud registration network using pointnet encoding (J). arXiv preprint arXiv:1908.07906, 2019.
[22] Gelfand N, Mitra N J, Guibas L J, et al. Robust global registration (C)//Symposium on geometry processing. 2005, 2(3): 5.
[23] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration (C)//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.
[24] Zeng A, Song S, Nießner M, et al. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions (C)//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1802-1811.
[25] Deng H, Birdal T, Ilic S. Ppfnet: Global context aware local features for robust 3d point matching (C)//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 195-205.
[26] Deng H, Birdal T, Ilic S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors (C)//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 602-618.
[27] Baker S, Matthews I. Lucas-kanade 20 years on: A unifying framework (J). International journal of computer vision, 2004, 56(3): 221-255.
[28] Schneider P, Eberly D H. Geometric tools for computer graphics (M). Elsevier, 2002.
[29] Chang C T, Gorissen B, Melchior S. Fast oriented bounding box optimization on the rotation group SO (3, $\mathbb{R}$) (J). ACM Transactions on Graphics (TOG), 2011, 30(5): 1-16.
[30] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes (C)//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.

Wenhao Lan was born in China and is a student member of the China Computer Federation. He is currently studying for a master's degree at Beijing Information Science and Technology University, majoring in pattern recognition and intelligent systems.