A Path Planning Algorithm of Spray Robot based on 3D Point Cloud

Jubo Yang[1,2], Minxiu Kong[1, 2],Rundong Wang[1, 2]
[1]School of Mechatronics Engineering, Harbin Institute of Technology
[2]State Key Laboratory of Robotics and System
Harbin, China
yjb960320@163.com

***Abstract*—The automatic path planning method of spray robot has always been a challenging problem, especially for the mixed-line production of various workpieces. This paper discusses the self-acquisition technology of spray path based on the point cloud model of furniture. The coordinate relationship between the coordinate system of the spraying system and the adaptive adjustment of the pose are discussed for the point cloud pose of the furniture. The furniture surface path planning based on point cloud slices and edge spraying based on edge extraction technology were studied to obtain the ordered path point sequence and posture of the spray gun. Through the robot spraying experiment on flat furniture and complex curved furniture, the reliability of the algorithm is verified from the aspects of algorithm efficiency and spraying effect.**

Keywords—furniture spraying; parameter optimization; point cloud filtering; point cloud slicing; automatic spraying

I. INTRODUCTION

In order to improve the level of industrial automation, it is necessary to use industrial robots for furniture spraying. Furniture spraying is a high-risk and harmful body industry, and it is urgent to reduce the participation rate of workers. The application of robots can greatly improve quality, increase efficiency, and increase repeatability. Among them, the automatic acquisition of spraying path information is an important research content to improve the robot's participation in work automation, because the path information is the basis for the robot to perform operations, and it determines various indicators in spraying. At present, the production line generally adopts the offline programming method based on the CAD model of the workpiece to plan the spraying path of the spraying robot.

Hansbo [1] et al. studied the thermal spraying spray path optimization algorithm for the workpiece with rotating geometry. The mathematical expression of the concise mathematical model was used to approximate the coating thickness, and the iterative method was used to optimize the spraying parameters. Since 2002, Chen [2] and others have carried out a series of research work on spray path planning. The 3D model of the workpiece is meshed, and the distance between adjacent straight paths is optimized according to the paint deposition model to obtain spray path points. In the same year, for the three-dimensional model of complex curved workpieces, the method of self-acquisition of the spray path was studied, and experiments proved that the method's efficiency and accuracy meet the requirements [3]. Later Chen [4] et al. continued to study the paint deposition model in depth, and used experiments to fit the paint thickness models of flat and freeform surfaces. Gyorfi [5] et al. aimed at the constraint conditions in the spray path parameter optimization problem, combined with the overall and local neighborhood features for synchronous optimization, and used graph optimization and biological models to solve the optimal problem to obtain the spray gun path parameters that meet the constraints.

In this article, we propose a method to obtain the spraying path information for unknown workpieces in real time to meet the requirements of flexible production lines. The innovation of this paper is to propose a method for spraying the edge of the workpiece, a method for adaptively selecting the slice direction, an optimized slice processing method and an optimized normal vector calculation method. This paper is organized as follows: Section II introduces the point cloud acquisition method. Section III introduces the path acquisition algorithm based on point cloud slicing in detail. And Section IV discusses the experimental results of the algorithm. Finally, Section V concludes this paper.

II. POINT CLOUD ACQUISITION METHOD

The line laser sensor selected in this paper intersects with the surface of the furniture as a line segment, and single-frame measurement data can only obtain the coordinate information of the x and z axes, so the y axis coordinate information needs to be supplemented. Through the relative displacement between the laser sensor and the furniture workpiece, multiple frames of data can be continuously obtained, thereby obtaining complete three-dimensional information on the surface of the furniture. In the overall project design of this subject, a chain plate conveyor line is used to transport furniture workpieces. At the same time, the photoelectric encoder is installed on the conveyor spool. The sensor is set to pulse trigger mode. The sensor will only turn on when the electrical signal from the encoder is received. One frame of data, so that the frame rate can be adjusted by adjusting the encoder pulse width and pulse number.

In the actual measurement system, the sensor is fixed, and the workpiece coordinate system will be driven by the chain conveyor to produce relative motion at a constant speed. In order to facilitate calculation and data processing, we selected the initial position of the furniture workpiece coordinate system as the base coordinate system.

978-1-7281-9761-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CISCE50729.2020.00078

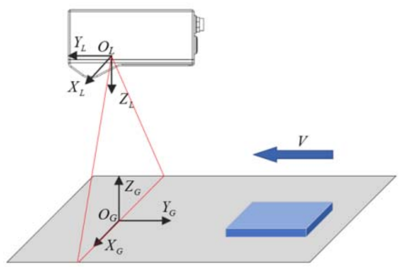


Fig. 1. The scanning system

The calibration of the sensor is also very important, because when measuring the workpiece of the furniture, if the plane calibration is not performed, the background other than the workpiece will be scanned as part of the workpiece. Therefore, this article uses the workpiece measurement plane as the calibration target to calibrate the sensor. If the distance from the origin of the sensor to the measurement plane is h, the transformation matrix between the sensor coordinate system and the furniture workpiece coordinate system is:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The Gocator laser sensor used in this subject has two scanning trigger forms, one is pulse type and the other is fixed frequency type. In this paper, a fixed frequency scanning mode is used to obtain point cloud data on the furniture surface frame by frame by setting a fixed scanning frequency. The y coordinate information of the scanning point can be obtained from the scanning frequency f and the fixed moving speed v of the chain conveyor line:

$$y = \frac{c \bullet v}{f} \tag{2}$$

According to this formula, we can calculate the 3D data of the scanned object and provide data foundation for future processing.

III. Point Cloud Slicing Algorithm

In general, the point cloud slice is to intersect the point cloud with several planes with a certain interval to calculate the intersection point of the scattered point and the plane. Since the point cloud has a limited density after being simplified, there are not many points obtained by slicing, and it is necessary to interpolate these points to obtain a smooth curve. After slicing the point cloud, we can not get the sliced data points immediately, and we need to perform a series of operations.

A. *Adaptive Adjustment of Point Cloud Position and Pose*

When planning the point cloud spraying path, we usually use the point cloud slicing algorithm. The principle of the algorithm is to use a set of planes to truncate and intersect the point cloud to obtain the data points on the cross section. Since we choose a raster-type travel route, this article selects a set of parallel planes for slicing. The direction of the cutting plane we choose directly determines the direction of movement of the subsequent robot end on the workpiece surface, so the following requirements must be met when selecting the cutting plane: the selection of the cutting plane should be based on the spraying process parameters such as the interval between adjacent paths; for special geometry The characteristic furniture workpiece surface, the direction when the cutting plane is selected follows its characteristics; for the isotropic furniture workpiece surface, in order to improve the spraying efficiency and make the slice direction better fit the furniture geometric features, usually choose the longest edge The vertical direction is the slice plane direction.

We can calculate the covariance matrix of the 3D data points of the point cloud, and then obtain the center of gravity, eigenvalues, and feature vectors of the point cloud, and then perform matrix transformation to adjust the pose of the point cloud so that the direction of the main feature vector is parallel to a certain coordinate axis, and The center of gravity coordinate moves to coincide with the origin. At this point, the point cloud is sliced, and the slice plane is perpendicular to a certain coordinate axis, so the form of the equation is simpler, and the amount of subsequent calculation is reduced. The algorithm flow is as follows:

Calculate the center of gravity of the point cloud:

$$\overline{p} = \frac{\sum_{i=1}^{k} p_i}{k} \tag{3}$$

Calculate the coordinate difference between the k-neighbor points and the center of gravity to form a new decentralized set:

$$Q = \left\{ p_i - \overline{p} \right\} \tag{4}$$

Calculate the covariance matrix F of set Q:

$$C = \begin{bmatrix} \operatorname{cov}(x,x) & \operatorname{cov}(x,y) & \operatorname{cov}(x,z) \\ \operatorname{cov}(y,x) & \operatorname{cov}(y,y) & \operatorname{cov}(y,z) \\ \operatorname{cov}(z,x) & \operatorname{cov}(z,y) & \operatorname{cov}(z,z) \end{bmatrix} \tag{5}$$

Solve the eigenvalues and eigenvectors of the covariance matrix to obtain the eigenvalue matrix *Za* composed of three eigenvalues and the corresponding eigenvector matrix *Ze*. The three eigenvectors represent the most significant direction of the point cloud feature. The covariance matrix can be decomposed Is as follows:

$$C \bullet Z_e = Z_a \bullet Z_e \tag{6}$$

It can be seen from the definition of the covariance matrix that the matrix is a symmetric positive semi-definite matrix, the three eigenvectors are perpendicular to each other, and the eigenvalue is not less than 0. Therefore, when adjusting the position and posture of the point cloud, the feature vector is used as the basis, so that it rotates parallel to the three coordinate axes of the robot base coordinate system, and the calculated center of gravity point is translated to the origin of the base coordinate system. The rotation and translation transformation are as follows formula:

The transformation matrix is as follows:

$$T = \begin{bmatrix} & V_e^T & & \begin{matrix} x \\ y \\ z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$P_c' = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = -V_e^T \bullet P_c \tag{8}$$

Where V_e^T is the transposed matrix of the eigenvector matrix V_e, P_c represents the centroid coordinates of the point cloud.

After the rotation and translation transformation of the point cloud, the point cloud pose can be adaptively adjusted to be in a position more conducive to the slicing algorithm. The result of the adjustment is shown in Fig. 2.

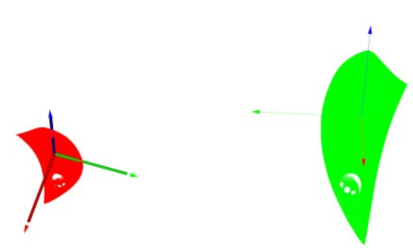


Fig. 2. Point cloud before and after transformation

In the figure, the green point cloud is the initial point cloud, and the red point cloud is after the coordinate translation and rotation. The coordinate system with thick coordinate axes and no arrows is the robot base coordinate system. The coordinate system with arrows is the coordinate system composed of the point cloud feature vector and the center of gravity. The green z-axis is the direction of the feature vector corresponding to the largest eigenvalue, that is, the most obvious direction of the point cloud feature. It can also be seen from the figure that the z-axis is the direction of the longest edge of the point cloud. Therefore, the tangent plane is selected to be perpendicular to the z axis, and after the point cloud slicing operation is performed, the obtained slice data points are inversely transformed back to the base coordinate system.

B. Surface Path Point Acquisition and Sorting Based on Point Cloud Slices

Point cloud slicing is to intersect a point cloud with several planes with a certain interval, and calculate the intersection point of the scattered point and the plane. Because the density of the point cloud is actually limited after the streamlining, there are not many points obtained by slicing. After the point cloud is sliced, we cannot get the slice data points immediately, and a series of operations are required. In order to reduce the amount of calculation of the algorithm, this paper limits the processing range, that is, the calculation is only performed in a point cloud band with equal left and right distances in the normal direction of the slice plane. This point cloud band is called the slice thickness, as shown in Fig. 3.

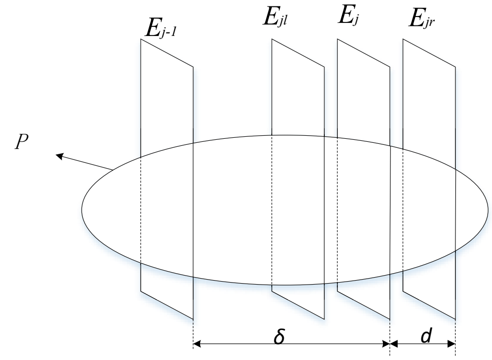


Fig. 3. The schematic diagram of point cloud slice

In Fig. 3, the thickness of the slice plane is the offset of the tangent plane in the normal direction by the left and right distance *d*. In this paper, d is selected as half of the path distance *a*, and the left and right offset planes *Enl* and *Enr* form a thickness of *a*. The data band and the tangent plane divide the data band to be processed into left and right parts. In this paper, the left and right parts of the cloud band are used to obtain slice points.

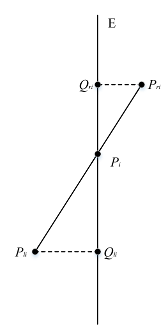


Fig. 4. The projection method and intersection method

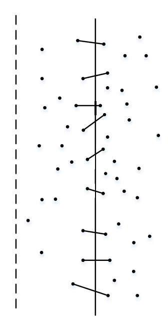


Fig. 5. The intersection between point cloud and plane

There are usually two methods for calculating slice data points through point cloud bands, the projection method and the intersection method, as shown in Fig. 4 and Fig. 5. The basic principle of the projection method is to project the points in the point cloud bands on both sides directly to the tangent plane. As a slicing point, the disadvantage of this method is very obvious. It is only suitable for slicing the surface of a planar point cloud. If the point cloud curvature is large at the slicing point Then the effect is very poor. The intersection method is to search for the closest point pair in the point cloud bands on both sides of the slice, and then calculate the intersection point of their line and the tangent plane as the slice point. This paper chooses the intersection method to calculate the slice point.

The flow of the algorithm is as follows:

First traverse all the points in the left half point cloud band, and use k-nearest neighbor search to find the point closest to the query point in the right half point cloud band. The calculation amount of traversing the data points on both sides is too large. In order to improve the calculation efficiency, this paper uses voxel division to reduce the search range of the right half point cloud belt. First, a cube with a side length of 0.5a is created centering on the query point. The search cube belongs to the right half point For the cloud point, if the search result is 0, the side length of the cube is gradually increased in steps of 0.5a until the search result is not 0. Search the nearest point in the limited cube.

Similarly, we need to find the closest point pair, and we must repeat the above steps in the right half point cloud band, and compare the closest point to the query point in the left half point cloud band. If they are closest to each other, they are stored in the corresponding linked list structures list1 and list2, and the point is marked as queried and removed from the right half of the point cloud band; if they are not closest to each other, they are directly marked as queried And removed from the point cloud belt.

Repeat the above two steps until all points on the left and right sides are queried.

Connect all the matching point pairs found in the linked list, calculate the intersection between the connecting line and the cutting plane, and store it in the slice point linked list list3.

The straight line equation formed by the matching point pairs is as follows:

$$\frac{x-x_{ri}}{x_{li}-x_{ri}}=\frac{y-y_{ri}}{y_{li}-y_{ri}}=\frac{z-z_{ri}}{z_{li}-z_{ri}}=t \tag{9}$$

If the direction of the slice is parallel to the Z-axis, the intersection $P_i\left(x_i, y_i, z_i\right)$ can be calculated as follows:

$$\begin{cases} x_i=t\left(x_{li}-x_{ri}\right)+x_{ri} \\ y_i=t\left(y_{li}-y_{ri}\right)+y_{ri} \\ z_i=z_{\min}+\left(0.5+j\right)\delta \end{cases} \tag{10}$$

Repeat the fourth step for all cutting planes to calculate the slice point, you can get the projection of the spray gun's path point on the workpiece surface.

After finding all the path points, because the point cloud data points are spatially disordered, and the spray path planning requires a curve composed of ordered points, so we have to search and sort.

Since the point cloud slicing algorithm is based on the point cloud adaptive adjustment of the pose, the slices are all perpendicular to a certain coordinate axis. The previously discussed convenient slice processing is reflected here. The slice points on the same cutting plane The z coordinates of are equal, so that the sorting of point cloud data is transformed into the sorting of data on the plane. Sort the slice points corresponding to each slice according to the increasing order of the y coordinate to obtain a list_Li of several slice point sorting linked lists. The sequence of single slice points is shown in Fig. 6.

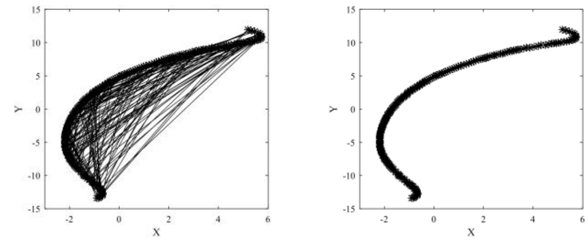


Fig. 6. The unsorted slice data(left) and the sorted slice data(right)

Since the path of the point cloud surface spray gun is a raster type, after sorting the data points on each slice according to the y-axis coordinates, it is necessary to connect different paths. Considering that the two adjoining paths of the raster-type advancing method have opposite advancing directions, when connecting each slice point linked list, it is necessary to reverse the direction of each two paths, so define the parameter, where i is the number of the slice path. When DIR is negative, we input the nodes into the total slice path point list list_L starting from the last node of list_Li; when DIR is positive, we enter list_L from the head node to obtain the complete path of the spray gun on the surface of the furniture workpiece Point projection.

C. Research on Edge Extraction Algorithm of Point Cloud

In the previous section, we studied the grating spraying path planning on the furniture surface, which can obtain better spraying effect for different furniture workpieces. If only grating type spraying is applied, the uniformity of the paint thickness at the edges of the furniture is poor, and the quality of the spraying cannot be guaranteed. Therefore, a separate patching operation needs to be performed on the edges of the furniture. In this section, we study furniture edge path point acquisition algorithm based on edge extraction.

The edge extraction of point cloud is usually used in the edge detection algorithm of building point cloud. In this paper, the detection algorithm based on the neighborhood angle threshold is used to calculate and analyze the neighborhood of each point of the point cloud using the maximum angle criterion to determine whether it is an edge point. The algorithm utilizes the regional differences of point cloud neighborhoods for edge detection. Fig. 7 shows the difference between the general data points of the point cloud and the edge data points in the spatial neighborhood. It can be seen that the topological distribution of the neighborhood of the general data point is relatively uniform, and the structure of the neighborhood extension at the boundary is only distributed in The side of the point.

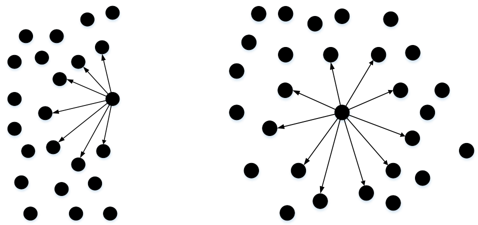


Fig. 7. Comparison of neighborhood topologies

It can be seen from the figure above that the biggest difference between the two lies in the uniformity of the topology, so we study its quantitative expression for this characteristic. Since a point cloud is a messy, three-dimensional data point distributed in three-dimensional space, if it is not reasonable to directly compare the k-neighbor distribution of each point, this article will make a tangent plane at each data point and take the points in the k-neighborhood. Project onto this tangent plane, as shown in Fig. 8.

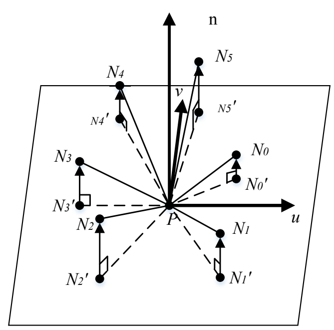


Fig. 8. k-nearest tangent plane projection

The purpose of this algorithm is to calculate the projection angle on the tangent plane of the angle formed by the k-nearest neighboring neighboring point and the target point, respectively, that is, if the method of calculating the angle according to direct projection is too complicated, so this paper A new coordinate system is introduced, with the target point as the origin, the normal direction of the point as the n-axis, and the orthogonal vector as the u-axis. Calculate the angle between the projection of the u-axis and the neighboring point of the target point and the line of the target point as follows:

$$a_j = \operatorname{atan}\left(\boldsymbol{u}\bullet\overline{PP_j}, \boldsymbol{v}\bullet\overline{PP_j}\right) \tag{11}$$

According to the above formula, traverse all points in the nearest neighbor of the target point k, and calculate its angle with the u axis. Set the calculated angle value as a set, and use bubble sorting to arrange the angles in the set in increasing order to get a new set. According to the new set of angles, we calculate the change value of two adjacent angle values, and also form a set of angle change values A:

$$A = \left\{A_m = a_m^{'} - a_{m-1}^{'}\right\}, m \in \left[1, 2, 3..., k-1\right] \tag{12}$$

This angle change value has its special geometric meaning. First, the angles are sorted in increasing order. In fact, the k neighborhood points are sorted counterclockwise around the target point based on the projection. Then the difference between the adjacent angle values is actually the neighbors. The angle formed by the connection between the points and the origin is the goal required by this algorithm. Since only the difference between adjacent angles is calculated and a closed loop is not formed in space, the angle between the nearest neighbor and the origin corresponding to the maximum and minimum angles is also required as follows:

$$A_{m+1} = \theta_1^{'} - \theta_m^{'} + 2\pi \tag{13}$$

Adding to set A, we can get the complete set of angles formed by the three neighboring points and the target point that we initially requested. According to the angle threshold set in advance, it is compared with the maximum value in set A as a basis for determining whether the target point is an edge point.

Therefore, the steps of the edge acquisition algorithm of the furniture point cloud are as follows:

Traverse all data points in the point cloud, use kd-tree to establish the point cloud topology, and search for k nearest neighbors of each point. According to the maximum angle judgment criterion, the maximum angle formed by the adjacent points in the k-neighborhood of the target point is calculated by the above method, and compared with the angle threshold, if it is greater than the threshold, it is judged as an edge point. Back to step 1, repeat the above process until all the point cloud data points are traversed.

D. *Clustering and Sorting of Edge Points*

As with surface path planning, the detected edge data points are a set of spatially disorderly distributed points, which cannot be used as the path information of the robot, and when the furniture geometrical features are complex and there are internal contours such as holes, several edges will be mixed together. Therefore, before path planning, the edge point set must be separated and sorted. Generally, the region growing method or clustering algorithm can be used to distinguish different edges of the edge points. The area growth method is based on the two important characteristic normal vectors of the point cloud and the curvature. The principle is to grow points with similar directions and similar curvature to the surroundings and classify them. Because the density of data points obtained by edge detection in this paper is not enough to use this algorithm to obtain good results, this paper chooses the European clustering algorithm with low data density requirements for edge separation. The flow of the algorithm is as follows:

Select any point from the obtained set of edge data points as the initial point, and calculate the neighboring points in the calculation space whose distance is less than the preset threshold.

The selected target points are compared with the points in the search neighborhood and the edge set, and the coincident points are eliminated from the edge points. Then, the remaining points in the search neighborhood are continuously calculated as the search neighborhood that is smaller than the threshold, and still compared with the edge set to eliminate coincident points.

Repeat step 2 above until all points in the search neighborhood of the target point do not coincide with the original edge set, indicating that a complete edge has been separated.

Repeat all the above steps until all points in the edge set are eliminated, indicating that all edge data points have been clustered and separated to their respective edges.

The original edge data set is clustered to obtain multiple independent edges. Since the points in each edge are disordered, the topological relationship and order between the points are unestablished. If you want to be a waypoint for a painting robot, you must first sort the edge points to form an ordered point cloud sequence. This subject ranks the edges based on the nearest point search algorithm. Find the closest point of the target point using k nearest neighbor search with k value of 1. The flow of the algorithm is as follows:

Create a linked list L, randomly select a point from the edge points as the table header s, and record the point as searched.

Using k nearest neighbor search, set k to 1, find the edge point closest to the head of the table s as the tail point e of the linked list, and mark the point as searched.

Search for the point closest to the table head in the set of edge points, and calculate the distance between the point and the table head and footer respectively, and record them as ls and le respectively and compare the sizes. If $l_s<l_e$, it means that the point is in front of the table head; If $l_s>l_e$, it means that the point is behind the end of the table.

If $l_s<l_e$, the point is defined as a new table header, otherwise it is used as a new table footer, and the point is marked as searched.

Repeat steps 3 and 4 above until all points are marked as searched.

E. *Calculation of Waypoint Spray Gun Attitude*

According to the influencing factors of the spraying process analyzed before, the direction of the spray gun should be kept in a direction parallel to the direction of the normal vector of the path point, so as to ensure the best uniformity of spraying.

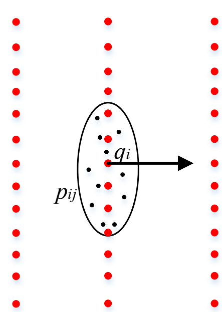


Fig. 9. Slice point normal vector calculation

As shown in the figure above, the red dots are the slice data points obtained from the aforementioned point cloud slices, and are also the projection of the spray gun on the surface of the furniture workpiece, because the normal vector calculation in the above is based on the normal direction of the neighborhood fitting plane, but The distribution of slice points is very sparse. If the normal calculation based on the surface fitting algorithm is used directly, a very serious deviation will occur. Therefore, for the calculation of slice point normal vectors, this paper proposes an improved principal component analysis method based on plane fitting, which transfers the area of k-nearest neighbor search during normal calculation from the slice point cloud to the original point cloud. Is correct and effective, the process of the pose acquisition algorithm is as follows:

Traverse the set of slice points, select a point to retrieve the k-nearest neighbors of the slice point in the original point cloud, and form the new set of neighbor points.

In k-nearest neighbors, the eigenvalues and eigenvectors of the covariance matrix of the neighbor point set are calculated according to the PCA method, and the eigenvector corresponding to the smallest eigenvalue is taken as the normal vector of the slice point.

Repeat the above two steps to calculate the normal vector at all slice points, and then optimize the normal vector. The optimization algorithm is also improved with the same idea, that is, performing k-nearest neighbor search on the original point cloud.

IV. RESULTS AND DISCUSSION

In order to verify the above algorithm, we use a laptop with an i7-4700HQ CPU and 8G RAM for testing. The experimental algorithm implementation is based on Point Cloud Library (PCL), an open source library that provides basic functions for point cloud processing tasks and 3D geometry processing.

The flat furniture selected for this subject is the door panel of the cabinet door. The surface of the door panel is rectangular, 60 cm long and 50 cm wide. According to the overall scheme of spraying path planning, firstly pre-process the acquired original point cloud of door panel.

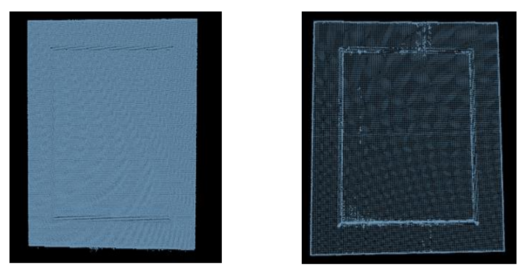
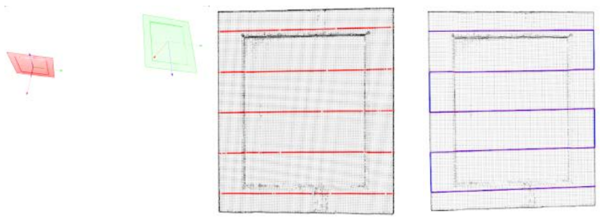


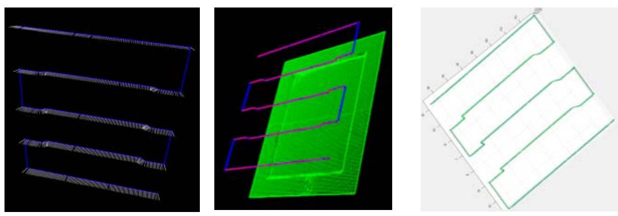
Fig. 10. The point cloud without pre-processing (left) and after pre-processing (right)

The preprocessing process of the furniture point cloud takes no longer than 3.25 seconds, and the remaining points are enough to provide good feature point cloud data for the subsequent path planning. Next, the spray path planning experiment of the flat point cloud is carried out, which is divided into two steps, the first is the planning of the surface of the furniture, and then the planning of the edge of the furniture.

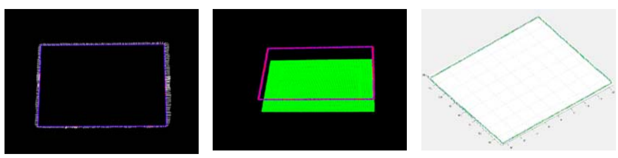
The overall planning process is shown in Fig. 11, and the parameters and time consumption of each link in the path planning are shown in Table 1.



(a) Posture adaptive adjustment | (b) Point cloud slice | (c) Slice ordering



(d) Slicing point spray gun gesture | (e) Airbrush surface path acquisition | (f) Surface path curve fitting



(g) Edge point spray gun gesture | (h) Edge point sorting | (i) Edge path fitting

Fig. 11. Flat furniture surface and edge path planning

From the above picture, we can see the path information acquisition process of the surface and edge of the point cloud of the flat furniture, and obtain the path point information available for the robot controller to run.

TABLE I. FLAT FURNITURE PATH PLANNING INFORMATION

Links of path planning	*parameters*	*Time-consuming (s)*
Pose adjustment	-	0.004
Point cloud slice	Path interval 133.125mm	0.858
Sorting of data	points 850	-
Slice point normal	-	0.299
Spray gun surface path	offset height 110mm	0.002
Surface curve fitting	Cubic B-spline	0.985
Edge point normal	-	0.784
Edge point sorting	-	0.001
Edge curve fitting	Cubic B-spline	0.986
Total	-	3.919

According to the results of the actual robot spray gun operation, it can be seen that the spraying planning algorithm for furniture surfaces and edges proposed in this paper can complete the actual spraying operation more quickly and accurately.

In terms of the efficiency of the algorithm, when the target workpiece is a flat door furniture, the point cloud preprocessing time is 3.25s, and the spray path planning time is 3.919s, so the total time is 7.169s.The requirement of this topic for the generation time of the spray path is less than 20 seconds. It can be seen that for the furniture workpiece, the total time for the path acquisition does not exceed half of the limited time, and the algorithm of this paper has a good effect in terms of operating efficiency.

V. CONCLUSIONS

The advantage of robotic spray trajectory planning based on point cloud slicing is that it can be operated directly on the point cloud, giving the path and the posture along the path directly. Combined with external measuring equipment to realize automatic planning, production preparation time can be greatly reduced. However, when it comes to considering the actual application in flexible manufacturing system, some special requirements need taking into account to make point cloud slicing algorithm suitable for robotic spray path planning.

Based on analyzing and summarizing previous research results, some significant improvement of traditional point cloud slicing algorithm are discussed in this paper to meet the requirements of robotic flexible manufacturing system. An azimuth prediction and transformation algorithm of the work piece is proposed to adapt to the uncertainty of work piece orientation on the assembly line. Then, the traditional point cloud intersection algorithm is improved to increase the operation speed. What is more, a new method is proposed to calculate the normal vector of the data points on slices. Finally, in order to meet the requirements of robot motion path, the spatial ordering of data points on the slice is carried out.

The results of the algorithm operation show that the algorithm in this paper can quickly plan a reasonable spraying path for work pieces with different surfaces. Therefore, this method based on vision sensors is a feasible method to solve the trajectory planning problem of spraying robot.

ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China (Grant No. 2017YFB130482).

REFERENCES

[1] Hansbo A , Per Nylén. Models for the simulation of spray deposition and robot motion optimization in thermal spraying of rotating objects[J]. Surface & Coatings Technology, 1999, 122(2-3):191-201.

[2] Chen H , Sheng W , Xi N , et al. Automated robot trajectory planning for spray painting of free-form surfaces in automotive manufacturing[C]// Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on. IEEE, 2002..

[3] Chen H.P, Xi N, Sheng W.H. Cad-based automated robot trajectory planning for spray painting of free-form surfaces. Industrial Robot, 2002, 29(5) : 426-433.

[4] Chen H.P, Xi N, Sheng W.H, Chen Y.F. Optimal spray gun trajectory planning with variational distribution for forming process. In : Proc of IEEE International Conference on Robotics and Automation. New Orleans, 2004 : 51-56.

[5] Gyorfi J.S, Gamota D.R, Mok S.M. Evolutionary path planning with sub-path constraints. IEEE Transactions on Electronics Packaging Manufacturing,2010, 33(2) : 143-151.