A Trajectory Planning Method for Robot Scanning System Using Mask R-CNN for Scanning Objects with Unknown Model ⋆

Zhan Li [a], Zhaoting Li [a], Yipeng Yang [a]

[a] *the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin, Heilongjiang, China*

Abstract

Laser scanning has played an important role in many fields of industrial production, such as product defect detection, reverse engineering and so on. We believe that the key to the realization of machine replacement is to automatically plan the scanning trajectory. This paper presents a method for planning laser scanning trajectory, especially for the objects with unknown model. This method is also suitable for painting and welding operations. Mask R-CNN is used to segment RGB images. Combined with the well-registered depth map, rough point clouds of processed objects can be extracted, which can be used as the basis of trajectory planning. In the part of trajectory planning, on the basis of slicing method, the least square method is used to smooth the trajectory. In the actual scanning process, a method of real-time adjusting sensor position and posture by using PID controller is proposed to optimize the collected data. Finally, we construct an actual scanning system to verify the effectiveness of our method.

Key words: Deep learning;Point cloud segmentation; Trajectory planning.

1 Introduction

Automatic laser scanning systems have demonstrated their potential in removing human errors from many fields of industrial production, such as product defect detection, reverse engineering and so on. The trajectory planner plays

⋆ This paper was not presented at any IFAC meeting. Corresponding author: H. Gao.

Email addresses: `zhanli@hit.edu.cn` (Zhan Li), `lizhaoting@hit.edu.cn` (Zhaoting Li), `yangyipeng@hit.edu.cn` (Yipeng Yang).

an important role in an automatic laser scanning system since it determines the exact position and orientation of the laser scanner, thus determining the quality of scanning data. The common goal of a trajectory planning part is to find a trajectory that optimizes the scanning results, i.e., the constrained position method and the Next-Best-View (NBV) method.

However, the constrained position method only can be applied in the situation where the model of sacnned objects is unknown. The NBV method tends to have long computation time and need a 3D surface laser scanner, which is usually more expensive than a line laser scanner. We are motivated to address these issues with a novel automatic laser scanning system consisting of a line laser scanner, a Kinect camera and a UR10 robot. Our main contributions lie in these two novel techniques:

- Automatic pointcloud segmentation with a Mask R-CNN network.
- Trajectory planning with least square fitting method and online correction.

2 Related work

2.1 Trajectory planning

According to whether the object to be scanned has known information, we divide the path planning into three situations: the object has a CAD model, the object has a model but needs to consider uncertainty, and the object has no model.

For scanning the object with a known CAD model, the methods of generating scanning path mainly focus on generating an optimal scanning path that satisfies some major constraints of the laser scanner. These constraints include the view point, field of view, scanning angle and so on [9]. Because the CAD model of an object is available, the surface normal vectors of this model can determine the scanning directions. Then the sequence of points with known scanning directions can generate the scanning path. The details of this overall procedure are introduced in [9]. Although good results can be obtained by this method when the object has a known model, the larger or more complex model will lead to too long planning time. When the object model is unknown, the information of surface normal vector is difficult to obtain accurately. Thus this method is difficult to apply directly to scan objects without a known model. Nguyen [5] proposes a novel path planning approach by using a conformal map ,which stretches 3D surface on a 2D plane, to control the overlap between two adjacent scanning paths. However, this methods may cause a sudden change of scanner orientation between two paths.

When objects have known models, many studies investigate the uncertainty in the scanning process in order to obtain better scan results. Mahmud [7] proposes a method which guarantee the measurement accuracy of scanning. Every geometrical entity is attributed to a maximum measurement uncertainty which depands on the angle of incidence between the surface of the object and the laser plane. Then an optimal trajectory is found by analysis of the visibility map. However, the whole process is also time-consuming and is not fully automated.

When the model of the object is unknown, the trajectory planning of the sensor is more challenging. [5] presents a non-model based Next Best Views planning method, which is an incremental approach based on the scanned part 3D data. First, the scanning data are classified into Well Visible and Barely Visible area according to some constraints. Second, the potential view points are determined by clustering the Barely Visible paths. However, the assumption made in this method that the object shape is totally in the field of view of the ranging device makes it only applied to the scanning of small objects using a 3D surface laser scanner. For an automatic scanning of ship blades, Eirik B et al. [8] presents an approach which focuses on estimating the axis of objects to be scanned. And an uninformed grid based search algorithm is proposed, whose initial search position is generated form the estimated axis. However, the ship blades need to vertically placed, which means this method can not be applied to scan other objects.

2.2 Pointcloud extraction

It is a feasible method to obtain point clouds with low accuracy by using inexpensive depth cameras such as Kinect as the basis of scanning trajectory planning. The segmentation of interested parts from a frame including background information and noisy point clouds is called point cloud segmentation.

The related work of point cloud segmentation can be seen from two aspects: on the one hand, point cloud segmentation directly or from depth map; on the other hand, point cloud extraction is indirectly realized by segmentation RGB images and then generating point cloud combined with depth map.

In [3], E. Grilli reviewed the application of traditional methods such as edge-based, region growing, model fitting in point cloud segmentation. These methods have limitations for point cloud extraction in our system. When the shape of the object is flat and close to the processing plane, the attributes of the depth map depended on by edge-based method do not change significantly, and it is affected by the accuracy of the sensor, so it can not achieve good segmentation. The region growing method needs appropriate seed points to achieve good segmentation results, but we can not select them in advance in our system. Therefore, this method is not applicable, and the model fitting method needs a primitive model. Our system mainly divides the object without model, so this method is not applicable. For the method of RANSAC (Random Sample Consensus) fitting plane and extracting body, we proposed to use this method in previous papers, but after a lot of experimental verification, this method still needs to manually adjust parameters in order to separate object from plane, which needs to be improved.

With the successful application of deep learning in image processing, many scholars are studying the application of neural networks in point clouds to achieve semantic segmentation or instance segmentation, and have achieved good results. In [1], Boulch proposed a method to label unstructured point cloud. The method uses multiple camera viewpoints to project point clouds into RGB images and depth images, then uses deep Convolutional Neural Networks (CNN) to achieve semantic segmentation.Finally, the segmented 2D images are projected back to point clouds.He tested different architectures to achieve a profitable fusion. In [6], Landrieu proposed a new structure called superpoint graph (SPG), which can represent the contextual relationships between object parts in a in a compact yet rich way and capture geometrically homogeneous elements from point cloud. A graph convolutional network was used to exploit the representation. In [8], Qi proposed a 3D graph neural network (3DGNN) that combines 2D image features extracted by CNN and geometric information of point clouds projected from RGBD images. In [11], Wang proposed Similarity Group Proposal Network (SGPN) to realize instance segmentation on point clouds. He used a novel form of a similarity matrix to formulate group proposals to directly recover accurate instance segmentation results.

Another method of point cloud segmentation is to segment 2D images and generate point clouds by combining depth maps with them. The key is to segment 2D images accurately. In the case segmentation research, Mask R-CNN [4] is a very popular case segmentation network in recent years, and many scholars have made further research on it. In [2], based on Mask R-CNN, Danielczuk verified the effectiveness of its method of using virtual data to train network and migrate to real environment. In [10], Vuola ombined Mask R-CNN with U-NET to generate a combination network for nuclei segmentation. The method we propose also uses Mask R-CNN to segment RGB images and extract the pixel-wise coordinates of the objects to be scanned.

In this paper, we are motivated to propose a trajectory planning method for our robot scanning system. It is composed of multiple planning sections with two novel techniques. The remaining part of this article is organized as follows. In section 3, the main proposed planning approach is explained in detail. In section 4, the experiment of this approach is presented. We finally conclude the paper in section 5.

3 Technical approach

The overall planning scheme consists of four steps:

(1) Extract the pointcloud based on a Mask R-CNN network to automatically segment the scanned object.

(2) Slice the pointcloud by finding nearest neighbour point pairs to generate a preliminary trajectory.

(3) Smooth the preliminary trajectory using least square method to fit the real profile of the scanned object.

(4) Generate the sensor trajectory and correct this trajectory during the scanning process to optimize the scanning results.

The first step is based on a Mask R-CNN network trained by our own training samples. It is the backbone of this automatic trajectory planning because it can extract the scanned object from the environment automatically. The last three planning steps makes use of least square method with a novel online correction scheme. This formulation of our planning method makes it possibile to perform automatic scanning with objects with unknown models.

3.1 Definition of some laser scanner constraints

The quality of the laser scan results is determined by factors such as the pose of the laser sensor and the relative position of the object being measured. Unlike a 3D surface laser scanner, a line laser scanner can only measure information on one plane at a time. When the attitudes of the two are deviated from the optimal attitude to the same extent, the scanning results of the line laser scanner has a worse quality, so the requirement of the line laser scanner itself relative to the posture of the measured object should be more strict. These factors are defined in detail below.

The distance of scanning Because of the measurement principle of the laser scanner is laser triangulation, too large or too small scanning distances will result in a large measurement error. The point cloud data will not be acquired when the effective measurement range is exceeded. Define the minimum effective scanning distance as d_{min}, which is also called as clearance

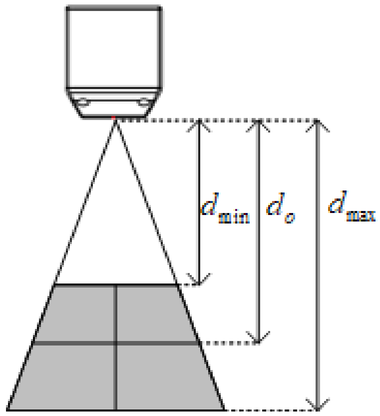

Fig. 1. Automatic Scan System

distance. The effective measurement range is r, and the maximum scanning distance $d_{\max} = d_{\min} + r$. In the actual measurement process, the scanning distance should be within the effective measurement range. In order to ensure the quality of the data, the scanning distance should be $d = d_0 = d_{\min} + \frac{1}{2}r$

The angle of scanning Define the angle between the Z axis of the laser scanner and the normal vector of the scanned area as the angle of the scanning.The scanner can get best scanning effect when these two vectors are in the same direction. When the scanning angle θ are larger than the allowed maximum deviation angle $\theta_{\max}$, the scanner fail to get any scanning data. During the process of path planning, we want the Z axis unit vector of the laser scaner to be parallel to the normal vector of the object, thus ensuring the best scanning results.

3.2 Extraction of point cloud

Through our long-term experiments, it is found that point cloud segmentation takes a long time to process point cloud directly, and there are the following problems:

(1) Fitting plane takes a long time and is affected by the accuracy of sensor. The result of plane fitting affects the result of point cloud segmentation, and then affects the result of trajectory planning.

(2) The method of removing background and limiting the scope of workspace through through through-pass filtering does not have the ability to cope with environmental changes, such as the position of the manipulator moving, and the relevant parameters need to be reset.

(3) When the volume of the object is small, it is easy to be mistaken as a part of the plane, or be treated as noise and then filtered, resulting in the

subsequent failure to plan the trajectory.

(4) If the shape of the object is flat and the difference between the plane height is not obvious, the segmentation effect is very poor, and it is easy to extract incomplete or mistaken as belonging to the plane. At this time, many parameters need to be adjusted manually to segment the object.

For sensors such as Kinect and Realsense, they are equipped with depth cameras and color cameras. By calibration, the images of the two cameras can be registered to generate color point clouds. Based on this, RGB images can be processed by convolution neural network, and scanned objects can be segmented for instance to achieve pixel-level extraction, and then combined with the registered depth map, the point cloud of the object can be extracted.

Based on Mask R-CNN, [2] presented a synthetic data set for instance segmentation network of depth images. However, when the shape of the object is flat or its platform is uneven, segmentation based on depth image is prone to problems, such as mistaking the object as belonging to the platform or a part of the platform as belonging to the object.

Therefore, we decided to use Mask R-CNN network to process RGB images, to achieve the segmentation of the target object instance, and to obtain all the pixel coordinates belonging to the target object. The depth map is registered with the color map, so that the pixels belonging to the target object can be obtained in the depth map. Then, the point cloud belonging to the target object can be extracted with high precision by combining the formula of generating point cloud from the depth map.

However, most of the existing data sets are in natural scenes, and the background is natural environment, such as street, field and so on, which is far from the working environment of the manipulator. Therefore, we annotated our own data set, which is suitable for the scene of manipulator work, and can be used for tasks such as manipulator grasping. In this paper, data sets are used to train mask R-CNN networks for extracting point clouds of specific objects from scenarios. ResNet101, the backbone network of the Mast R-CNN network, can be initialized using the weights trained on the CoCo data set. In this way, we only need very small training samples to achieve good results, and our data set does not need to be large to achieve the shape of similar objects recognition. The experiment has achieved good results. It can extract point clouds accurately, and identify objects faster and more accurately. Parameters do not even need to be modified, so this method has a certain robustness to environmental changes.

We used our own data set to train on the basis of the model that had been pre-trained by the CoCo data set. Similar to the case segmentation of category-agnostic mentioned in [2], we only care about the location of the object on the processing platform, but the type is not our concern.

3.2.1 Registration of color and depth cameras

A pinhole camera model is used to model the depth camera and the color camera. The mathematical expression is shown as follows

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where s is an arbitrary scale factor, $(u, v, 1)^T$ and $(X, Y, Z, 1)^T$ are the homogeneous expressions of the pixel coordinates of an object point and the coordinates in a camera reference coordinate system, respectively, and $\mathbf{A}$ and $[\mathbf{R}\ \mathbf{t}]$ are the camera intrinsic and extrinsic parameters, respectively.

By using Zhang's calibration method [13], the camera intrinsic parameters and the extrinsic parameters matrix relative to the calibration plate coordinate system can be calibrated. It is easy to find the transformation matrix between the color camera coordinate system and the depth camera coordinate system. The conversion relationship between each pixel in the depth image and the corresponding pixel in the RGB image is shown below.

$$s_{color} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{color} = \mathbf{A}_{color} \mathbf{R}_{depth}^{color} \mathbf{A}_{depth}^{-1} s_{depth} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{depth} + \mathbf{A}_{color} \mathbf{t}_{depth}^{color} \tag{2}$$

After identifying all the pixels belonging to the object on the color image, point clouds can be generated according to equation (1) (2), and then point clouds of the object can be extracted.

3.3 Generation of preliminary trajectory Based on the Point Cloud Slicing

The purpose of the point cloud slice is to extract the outline of the object on slicing sections , which can be used to calculate the pose of the laser sensor. The paper [12] put forward a CAD-based slicing approach, which generates object outlines by projecting curves of the surface to the slicing plane. When the CAD model is unavailable, we can approximate object outlines by using

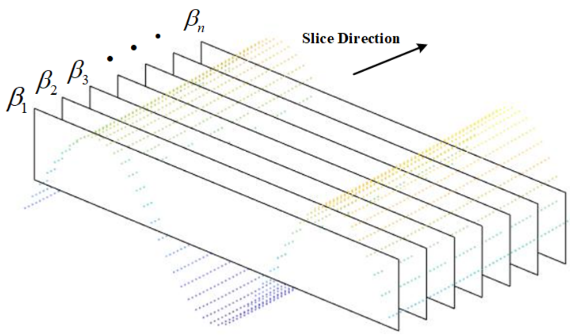

Fig. 2. Slicing of point cloud

raw point cloud information. However, this point cloud information tends to have a low precision, and the projecting method fails to approximate the real outlines of the object because of overlooking the sparse degree of point cloud information. We choose to find all the neighbouring point pairs, which are defined as two points whose nearest neighbor point is each other, by nearest neighbor search method.

Define the raw point cloud data set of the object from Kinect as $\Omega\left(x_k, y_k, z_k\right)$, $k = 1, 2, \ldots, N$,where N is the number of points in this data set. To illustrate the process of slicing and trajectory generation, we select a surface ψ composed of several sinusoidal functions.

$$\psi_z = 0.25\sin\left(4.5\psi_x\right) + 0.1\sin\left(2\psi_x\right) + 0.15\sin(5\psi_x + 0.5)] \tag{3}$$

$$\psi_y = 0.5\psi_x \tag{4}$$

To facilitate data processing, we select the X axis direction as the slice direction and the Y axis as the scanning direction of a single layer. If the object is placed on the workbench at random, it can still be transformed into a space by coordinate transformation in which we can use the X axis direction as the slice direction. Define the width of slicing as σ , we can get the number of slicing $n = \frac{\Omega_{x\max} - \Omega_{x\min}}{\sigma}$,and the point cloud set can be divided into $n+1$ subset $\Omega_i\left(i = 1, 2, \ldots, n+1\right)$ by n slicing section $\beta_i\left(i = 1, 2, \ldots, n\right)$,as shown in Fig. 2.

After getting the slicing sections, we need to determine the outline of the object on every slicing section. Considering the ith slicing section , the two point cloud subsets are Ω_i and Ω_{i+1}. By finding all the neighbouring point pairs in this two subsets and getting the intersection of the slicing section and the line passing this neighbouring point pairs, we can use all the intersection points to approximate the outline of the object on this slicing section.[2] Let A be a point in Ω_i, the closest point B in Ω_{i+1} to A can be found by k-nearest neighbors method. If point A is also the closest point in Ω_i to point B, the two points A and B make a pair of neighbouring points. The parametric equation of a straight line passing through A and B is:

$$x = x_A + t\,(x_A - x_B) \tag{5}$$

$$y = y_A + t\,(y_A - y_B) \tag{6}$$

$$z = z_A + t\,(z_A - z_B) \tag{7}$$

Define intersection point of the line AB and slicing section β_i as C . Because the slicing direction is along the X axis, we can get the coordinate of point C, where

$$y_c = y_A + \frac{\Omega_{ix} - x_A}{x_A - x_B}\,(y_A - y_B) \tag{8}$$

$$z_c = z_A + \frac{\Omega_{ix} - x_A}{x_A - x_B}\,(z_A - z_B) \tag{9}$$

3.4 preliminary trajectory filtering and smoothing

Because of the influence of noise and the low accuracy of the Kinect, the path obtained from the slicing of point cloud is composed of broken line segments with possible outliers. The appearance of these outliers has two possible reasons: the actual feature of the object and the noise during the measurement process. These outliers cause the scanning path unsmooth, which can lead into a poor scanning result. However, considering the first possibility, treating the outliers as noisy points is unreasonable. Otherwise, the laser scanner scanning along this path may not meet the requirement of the distance and angle of scanning. Overlooking these outliers may also result in the collision between the laser scanner and the object. In order to resolve this contradiction, we adapt the method involving filtering and fitting .

Filtering the path requires grasping the difference between a normal point cloud set and an abnormal point cloud set. We stipulate that normal point clouds derived from the surface of simple objects usually have the following characteristics: First, the point clouds are continuous, that is, any point cloud is always able to find other multiple point clouds in a certain area. Second, the point clouds should be relatively smooth and there is no surface with a continuous slope change.According to the above constraints, the filtering can be performed by the method of finding nearest neighour and comparing the rate of change of the centroid distance ratio.

After filtering, the initial trajectory is still rough. For example, there are still a few outliers that have not been filtered out. Moreover, the extracted surface of the originally smooth object becomes unsmooth because of the low measurement accuracy. However, rough trajectories reduce the accuracy of laser scanning and cannot be used directly for scanning. To address this problem, we need to replan a path $\xi(\mathrm{T}_0, \Gamma)$ that is as smooth as possible by least squares method, where T_0 is the path of single layer generated by slicing and filtering, Γ is the constraints during the scanning process. Define $p_i(i = 1, 2, 3, \ldots, k)$ as all the points on the path T_0, then $\xi(\mathrm{T}_0, \Gamma)$ should minimize the following cost function:

$$\min_{\xi} J = \sum_{i=1}^{k} [\xi(i) - p_i]^2 \tag{10}$$

Select the three or five times curve to fit the trajectory, which is assumed

$$\xi(i){=}ai^3 + bi^2 + ci + d \tag{11}$$

or

$$\xi(i){=}ai^5 + bi^4 + ci^3 + di^2 + ei + f \tag{12}$$

In the process of fitting the initial trajectory, first assume that the curve can be fitted with the cubic curve, and then judge whether the value of cost J is within the allowable range. If not, select the five times curve and make the judgment again. If the minimum value of cost still can't be satisfied, we need to split T_0 into several segements and perform local fitting, in which we use the line and parabola to smoothly join the fitted paths.

3.5 Acquisition of Sensor Trajectory

The path we obtained through the above process is actually the sequence of the ideal scanning points, which are the intersection points of laser scanner Z axis and the object when the scanning distance is optimal. We need to derive the laser sensor path and attitude based on the constraints on the scanning point. According to the constraints of laser scanner, it needs to maintain a certain distance from the object , so the path of the laser scanner is the equidistant line of the scan point path.

Considering the situation of a single scanning layer, define the path of laser scanner as $\mu_0(x_i, y_i, z_i)$, the path of scanning point as $\xi_0(x_i, y_i, z_i), i \in \{0, 1, \ldots, k\}$, the one point on this path as $\xi_0(x_i, y_i, z_i)(i = 0, 1, \ldots, k)$, the normal vector of ξ_0 with outward direction as η. We only discuss the case where the fitting result is a cubic curve, and the case where the fitting result is a five times curve is the same. The calculation process of the path of laser scanner $\mu_0(x_i, y_i, z_i)$ is shown as follows.

The x and y coordinates of one point $\xi_0(x_i, y_i, z_i)$ in this path point set have the following mathematics Relationship

$$\xi_0(z_i){=}a\xi_0(y_i)^3 + b\xi_0(y_i)^2 + c\xi_0(y_i) + d \tag{13}$$

The slope of the scanning path can be obtained by differentiating equation

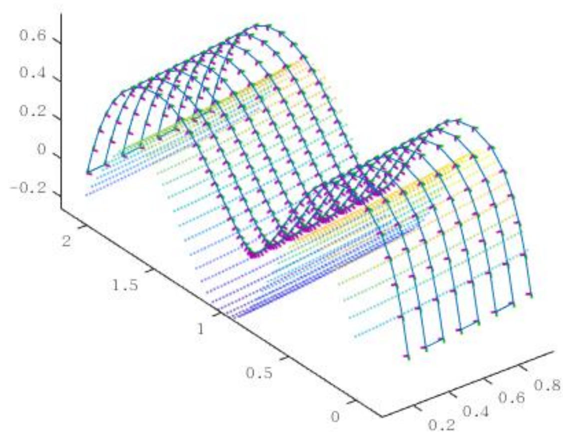

Fig. 3. The initial scanning path of laser scanner

(13):

$$k = 3a\xi_0(y_i)^2 + 2b\xi_0(y_i) + c \tag{14}$$

Then we can get the unit normal vector η_0, where

$$\eta_{0x} = 0 \tag{15}$$

$$\eta_{0y} = \frac{1}{\sqrt{1 + \left(-\frac{1}{k}\right)^2}} \tag{16}$$

$$\eta_{0z} = \frac{-\frac{1}{k}}{\sqrt{1 + \left(-\frac{1}{k}\right)^2}} \tag{17}$$

At this time, the direction of η_0 of ξ_0 is not all facing outward. The sign of the slope k is discussed, and the unit normal vector η whose direction is outward can be obtained.

$$\eta = \begin{cases} \eta_0, k < 0 \\ -\eta_0, k > 0 \end{cases} \tag{18}$$

The scanning path μ_0 of laser scanner is:

$$\mu_0(x_i, y_i, z_i) = \xi_0(x_i, y_i, z_i) + d \cdot \eta \tag{19}$$

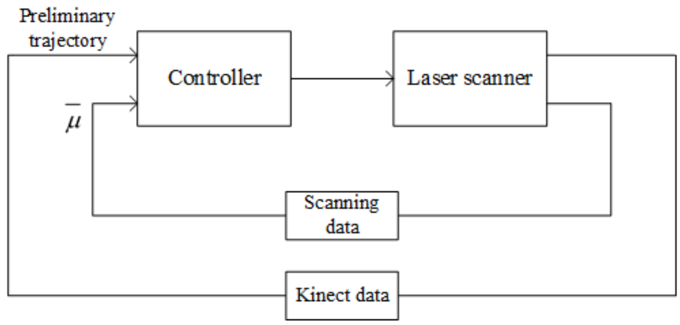

Fig. 4. The structure diagram of online correction

3.6 online correction

The trajectory of the preliminary planning only satisfies the requirements of smoothness, and there are still some slight errors with the actual surface of the object. To improve accuracy, we need to use feedback to suppress scanning errors. In the process of scanning, the path of laser scanner is adjusted in real time according to the data stored during the scanning process, so that the measured data has higher precision and higher reliability. Real-time adjustment of the scanning path only needs to use the scan data whose number is little, so the calculation amount is small, which can ensure better real-time performance.

During the actual scanning process, record the data of the first n laser scanning data ϕ_j $(j = 1, 2, \ldots, n)$. By analyzing the local data, the actual average normal vector can be obtained, which can be used for controlling the attitude of the laser sensor. ϕ_j is a line segment including many points, define the middle point of ϕ_j as α_j. These middle points make up the actual path of scanning points $\bar{\xi}(x_j, y_j, z_j)$, and the normal vector $\mu(x_j, y_j, z_j)$ of $\bar{\xi}(x_j, y_j, z_j)$ can be computed. Averaging the n normal vectors $\mu(x_j, y_j, z_j)$, we can get the average normal vector $\overline{\mu}$ of this local area. It is used as feedback to compare with the actual pose of the laser sensor. Then the laser sensor can be adjusted to maintain a better posture.

4 Experiment

In this section, we verify the effectiveness of our method through a complete experimental process and a number of experimental data.Our experimental platform, as shown in Fig. 6, includes a Universal Robot UR10 manipulator, a Gocator 2130C line laser scanner, a Kinect V2 depth camera and two customized brackets Used to fix the camera and the manipulator.

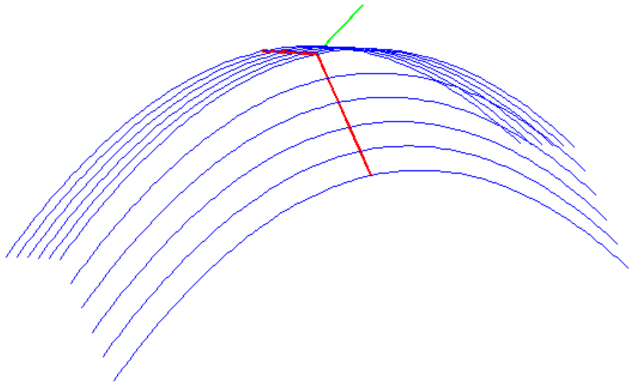

Fig. 5. The caculation of feedback

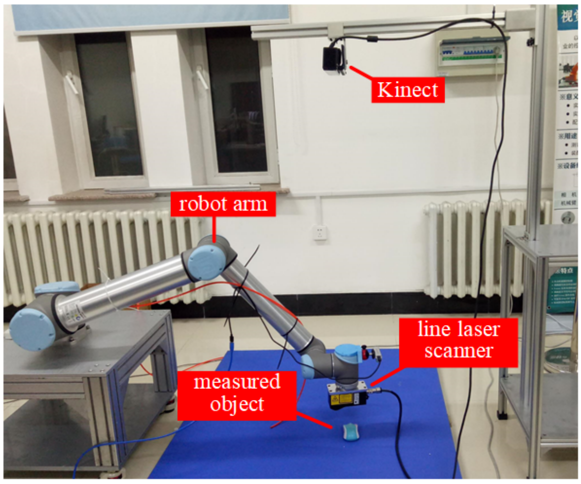

Fig. 6. Automatic Scan System

4.0.1 Extraction of point cloud using Kinect

The Kinect V2 camera is used in our experiment. When it leaves the factory, its register stores the transformation matrix parameters of RGB camera and depth camera, but they are not accurate enough and often needs to be re-calibrated. We re-calibrated the intrinsic parameters of color camera and depth camera, and the transformation matrix between them. The comparison of point clouds generated before and after re-calibration is shown in Fig. **??**. Depth maps and RGB maps before re-calibration are misaligned. Depth maps and color maps after re-calibration can be well matched.

We constructed our own data set and trained it on Intel i7 8700K and Nvidia 1080Ti for about 2 hours. The recognition time of a single RGB image of 1080*1920 is about 2 seconds, and it takes about 130 ms to generate point cloud with depth map registration. The recognition results of Mask R-CNN are shown in Fig. 8. Results of the extraction of point clouds are shown in Fig. 9 − Fig. 14 .

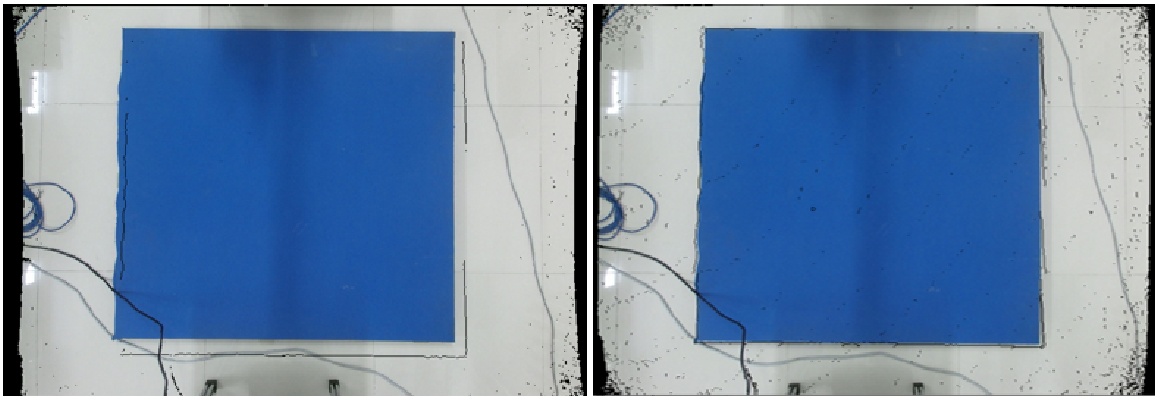

Fig. 7. Comparison of point clouds generated before and after re-calibration

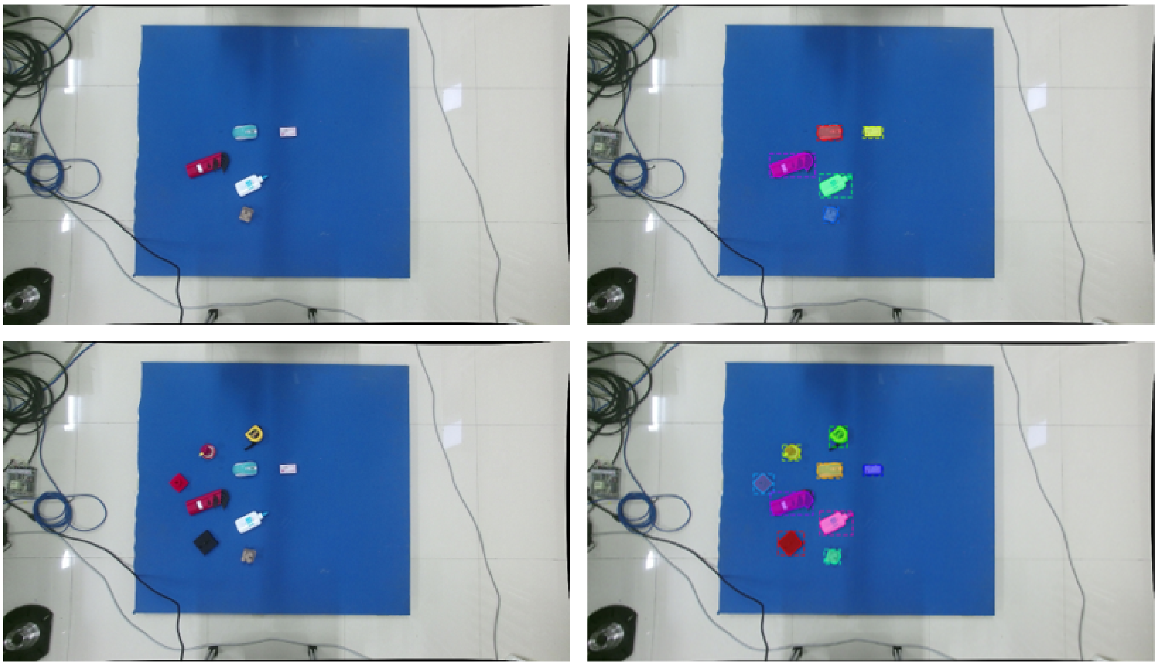

Fig. 8. Comparison of point clouds generated before and after re-calibration

4.0.2 Trajectory planned on point clouds and filtered trajectory

Taking the scanning path planning process of the mouse as an example, the path planning process is elaborated. At the end, the path planning diagrams of the other two unknown objects are given.

Choose the X axis direction as the slicing direction, the width of slicing σ is $0.01m$, the length of the mouse $h = \Omega_{x\,\max} - \Omega_{x\,\min}$, then the number of slicing $n = \frac{h}{\sigma}$. The slicing result is shown in fig.15. By finding all the neighbouring point pairs and computing the intersection of these pairs and slicing sections, we can get the raw outline of the mouse, which is shown in the middle of fig.15. However, the raw outline is very rough and contains many outliers. As discussed in the slicing part in this paper, on one hand, these outliers may

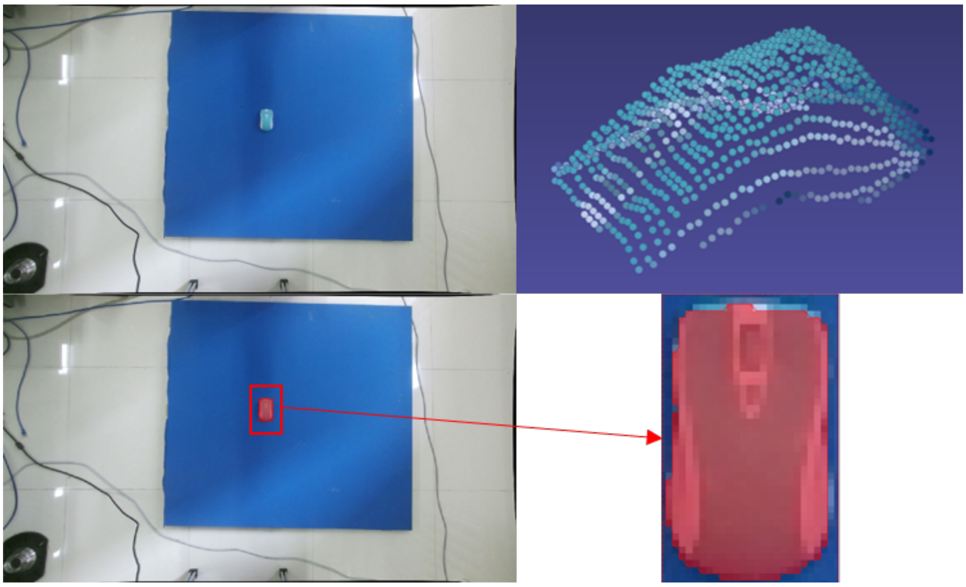

Fig. 9. result of extraction of point cloud (a)

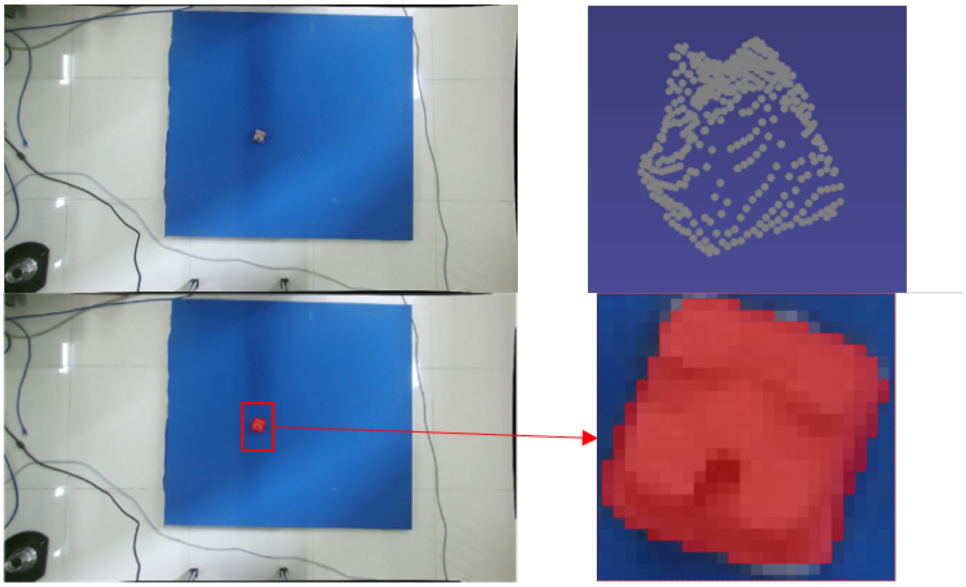

Fig. 10. result of extraction of point cloud (b)

be some nosiy data points generated by Kinect camera whose precision is low. On the other hand, these outliers can also be the actual feature of the object. Therefore, filtering all this outliers by sone filtering algorithms is not feasible. However, the presence of these discontinuities affects the measurement accuracy of the laser scanning and the quality of the scanning. The scanning path of the line laser sensor cannot be generated directly through this raw outline of object, so further processing is required.

The raw outline of the mouse is fitted using the least squares method, and we a cubic polynomial as the fitting function. A smooth scanning path can

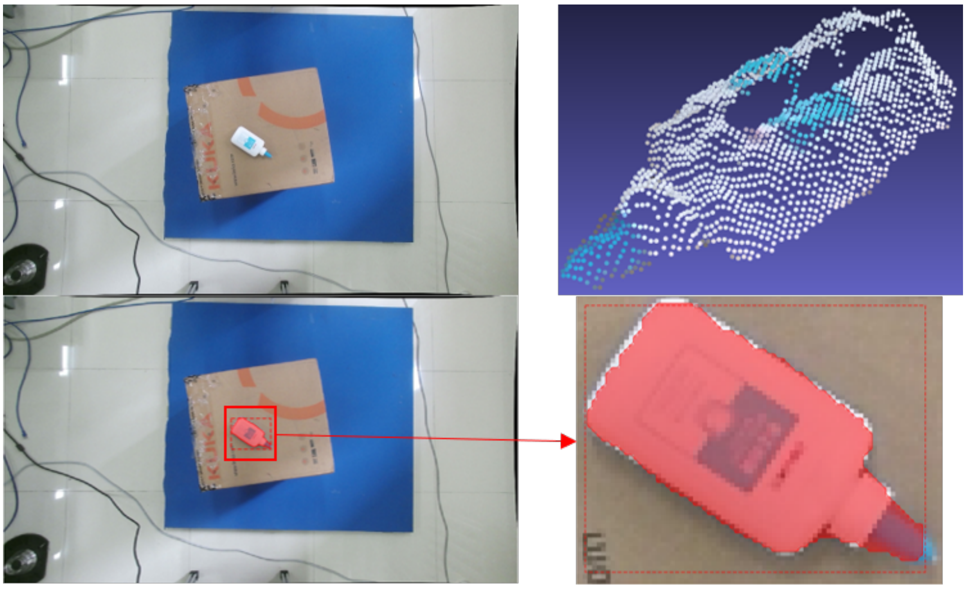

Fig. 11. result of extraction of point cloud (c)

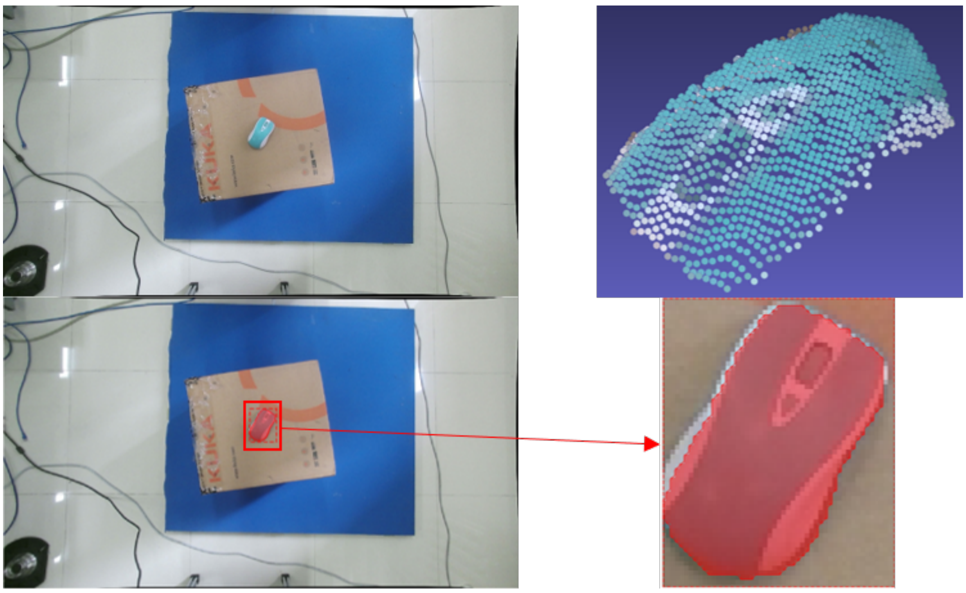

Fig. 12. result of extraction of point cloud (d)

be obtained by determining the coefficient of the fittinh function through minimizing the deviation of the fitted curve from the outline of the original object. This path can be used to further calculate the path of the laser scanner because of its smooth feature, which is shown in the right picture in fig.15.

Set the optimal distance of slicing $d_0 = 0.17\text{m}$ The pose of the laser sensor is obtained from the path of the scanning point. Because the path points are not evenly distributed, in order to smooth the robot trajectory, some path points are still need to be inserted between the planned scanning path points when controlling the robot arm moves, so that the scanning data is evenly

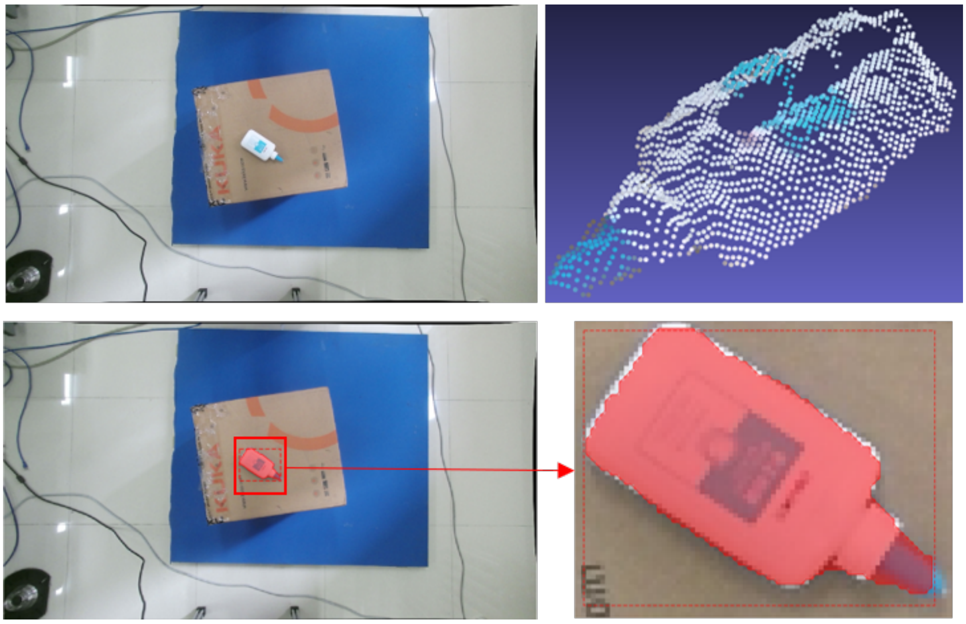

Fig. 13. result of extraction of point cloud (e)

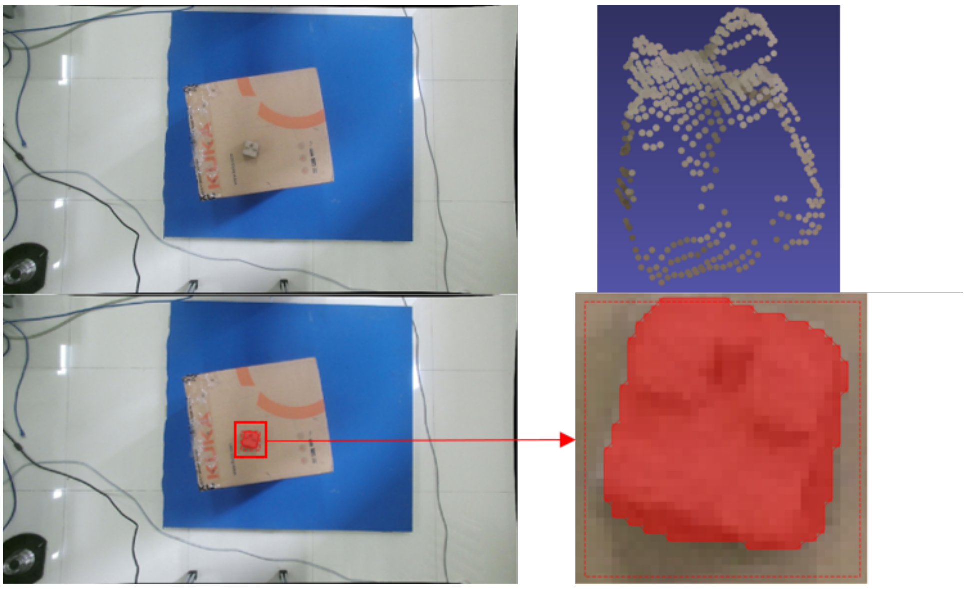

Fig. 14. result of extraction of point cloud (f)

distributed.

4.0.3 Scanning results

By scanning the planned trajectory, the point cloud data of the object can be obtained as shown in Fig. 17. Comparing it with the real object, we can find that although a few parts of the side of the object have not been scanned, the vast majority of the parts can get excellent scanning results.

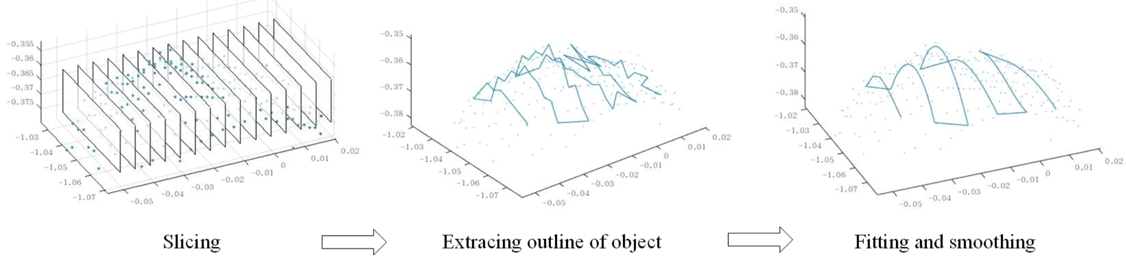

Fig. 15. The procedure of generating path of scanning points

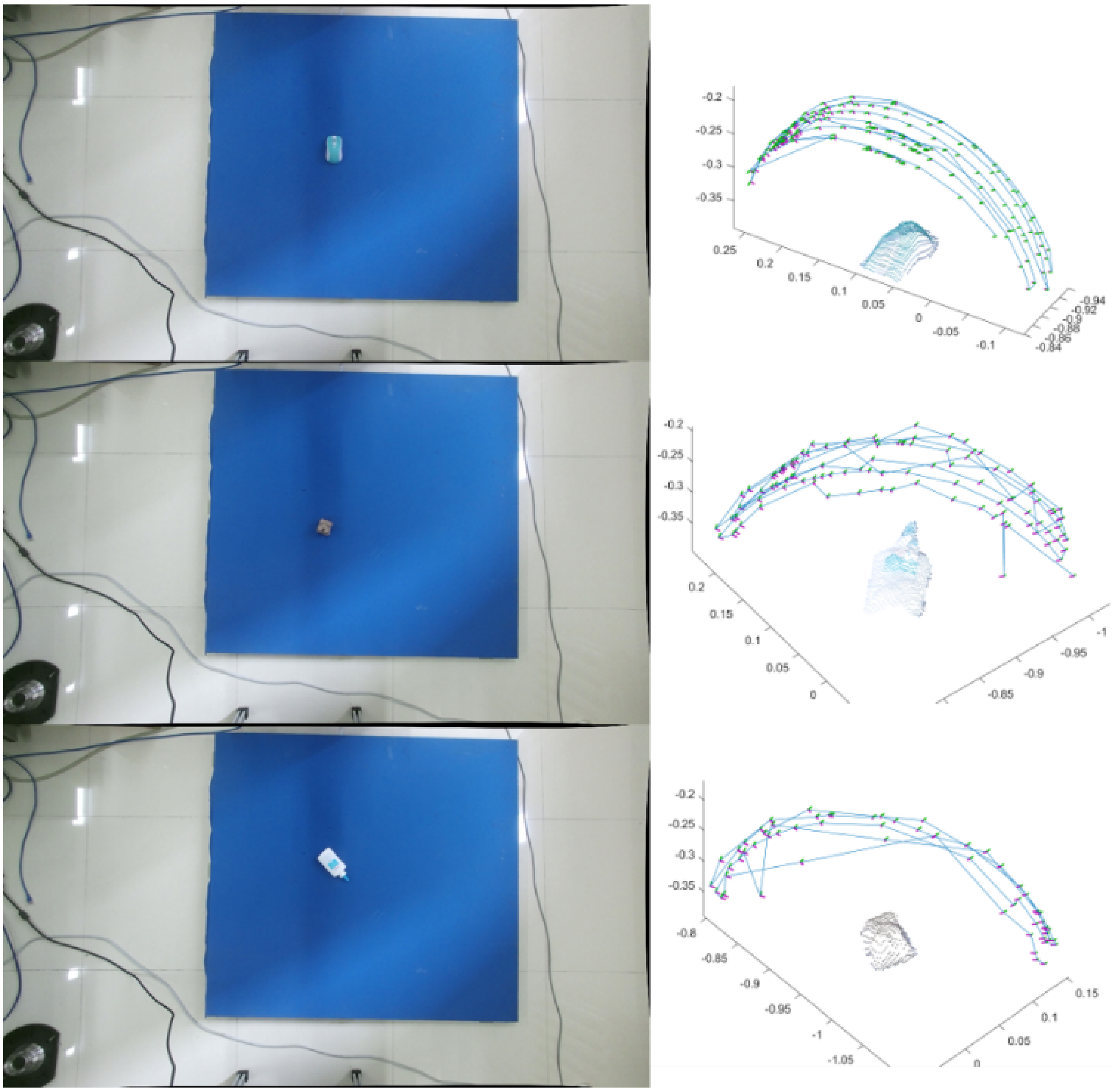

Fig. 16. The scanning path of laser scanner for different objects

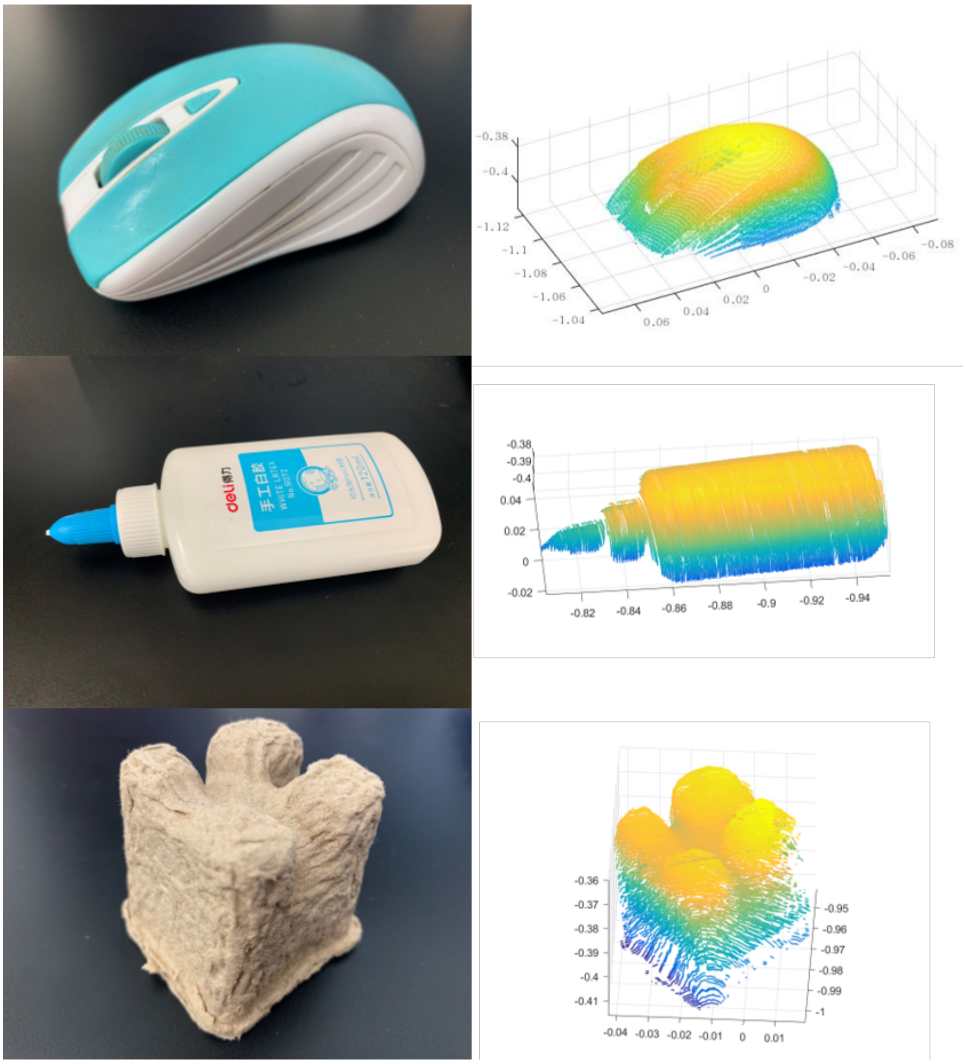

Fig. 17. result of scanning

5 Conclusion

A trajectory planning method for automatic laser scanning system is presented in this paper. Using Mask R-CNN network to process RGB images, the pixel level segmentation of small objects on different working platforms can be realized , and the point cloud of objects can be extracted by combining well-registered depth images. Then the preliminary trajectory is obtained by slicing method, and the smooth and effective scanning trajectory is obtained by combining the trajectory fitting and special processing method. Through experiments, the results of point cloud extraction, trajectory planning and scanning of various objects are listed, which verifies the effectiveness and feasibility of the method. The next step will be focus to optimize the trajectory planning method to scan objects with complex surfaces.

References

[1] A. Boulch, J. Guerry, B. L. Saux, and N. Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers and Graphics*, 71:S0097849317301942, 2017.

[2] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg. Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. 2018.

[3] E. Grilli, F. Menna, and F. Remondino. A review of point clouds segmentation and classification algorithms. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3:339–344, 2017.

[4] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[5] S. Khalfaoui, R. Seulin, Y. Fougerolle, and D. Fofi. An efficient method for fully automatic 3d digitization of unknown objects. *Computers in Industry*, 64(9):1152–1160, 2013.

[6] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. 2017.

[7] M. Mahmud, D. Joannic, M. Roy, A. Isheil, and J.-F. Fontaine. 3d part inspection path planning of a laser scanner with control on the uncertainty. *Computer-Aided Design*, 43(4):345–355, 2011.

[8] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. 2017.

[9] S. Son, H. Park, and K. H. Lee. Automated laser scanning system for reverse engineering and inspection. *International Journal of Machine Tools and Manufacture*, 42(8):889–897, 2002.

[10] A. O. Vuola, S. U. Akram, and J. Kannala. Mask-rcnn and u-net ensembled for nuclei segmentation. 2019.

[11] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. 2017.

[12] F. Xi and C. Shu. Cad-based path planning for 3-d line laser scanning. *Computer-Aided Design*, 31(7):473 – 479, 1999.

[13] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1330 – 1334, 12 2000.