An Approach to Edge Extraction Based on 3D Point Cloud for Robotic Chamfering

To cite this article: Yongzhi Wang et al 2019 J. Phys.: Conf. Ser. 1267 012015

View the article online for updates and enhancements.
An Approach to Edge Extraction Based on 3D Point Cloud for Robotic Chamfering

Yongzhi Wang\textsuperscript{a}, Zhijiang Du, Yongzhuo Gao, Mingyang Li and Wei Dong\textsuperscript{b,*}

Harbin Institute of Technology

E-mail:\textsuperscript{a}17S008047@stu.hit.edu.cn; \textsuperscript{b,*}dongwei@hit.edu.cn

Abstract. Inconsistency of workpieces always results in low level automation of subsequent finishing process. As decreasing number of skilled workers, the demands of flexible manufacturing with robots is growing. In this paper, a kind of workpiece with complex free-form surface will be chamfered by industrial robots. Due to the intolerant differences among workpieces, traditional teach-playback method for robot is not appropriate for this application. The edge spline of each workpiece needs to be detected so that trajectories are generated to chamfer. Thus, an approach to edge extraction based on a 3D point cloud obtained by the 3D industrial camera is introduced to solve this problem. A time optimizing method is proposed to accelerate the extraction process. Finally, we decrease computing time from 55 minutes to 300 seconds, and the precision is 0.4180 mm.

1. Introduction

In many industrial area, there are many situations that the workpieces are shapes of differences and complex, and with the increasing of labor cost, flexible manufacturing is paid more and more attention to now. The workpiece with complex free-form surface such as which we will chamfer in this paper shown as figure 1 is hard to process. The traditional teach-playback is not useful for three reasons. First, they are in different shapes and sizes. Second, the first process, cost, will cause much error. At last, there is not very nice positioning reference of the workpiece with the free surface. Therefore, the visual system of industrial robot needs to be used.

Figure 1. the workpieces of different shapes and sizes

In the feature and edge identification from point cloud field, there are a lot of researches now. Their method are about contour getting straightly or the feature of surface. In the field of contour getting straightly, M J Milroy[1] et al. use an energy-minimizing active contour interactively to find the edge points and they use this method on segmenting a phone model successfully. In the field on curvature, M. Yang[2] et al, propose an automatic edge-based approach based on the computation of the surface.
curvatures and the principal directions, and the edge points are regarded as the curvature extremes. They also supposed an forming boundary curves algorithm based on edge-neighborhood. In the field on the feature of surface, Vanna Sam[3] et al. propose an robust algorithm for extracting a centered curve skeleton from point cloud. A lined segment sequencing algorithm proposed by Kai Wah Lee[4] is to segment the different surfaces in the point cloud of complex geometry surrounded by a developable surface.

Although there are many researches about point cloud, there is less application on the industrial robot processing. Because of the high precision in industry, it is hard to get the accurate curve to process which satisfies the requirement. In this paper, we get the workpiece from point cloud, and extract edge based on Micro-cut plane. To solve the precision problem, we propose a new iterative algorithm and get a well consequence finally.

In this paper, the program of recognizing an edge to chamfer of a workpiece from a point cloud is shown.

The rest of the paper is organized as follows: First, the related work on point cloud is introduced in Section 2. The problem is expressed in Section 3. The flow of identifying the edge from the point cloud is shown in Section 4. The result and analysis are shown in Section 5, and Section 6 concludes the paper.

2. Problem Formulation
In our works, we will chamfer a set of workpiece with different shapes and sizes, and the precision requirement of chamfering is 0.3mmx0.3mm~0.5mmx0.5mm. For the end effector are designed to be compliant, it can compensate for errors caused by the visual system and robot. Considering the compensating ability of the end tool, the requirement of the visual system is set as 0.3mmx0.3mm~0.5mmx0.5mm.

The industrial camera is used to obtain the point cloud of the workpiece, but there will be some other things like fixture in the point cloud. So, we must segment the workpiece from the point cloud first and then recognize the edge to be processed. Finally, a fitting curve of the edge will be got and a set of sequence points will be calculated and transferred to robot to execute.

In the paper, we set the evaluation indicators of the fitting curve as follow:

Let the fitting curve have a total of i sequence points (xs, ys, zs). One of (xs, ys, zs) nearest neighbor point in the point cloud is (x, y, z). The distance between the two points, (xs, ys, zs) and its (x, y, z), is D, and the precision index is that the average value of all D.

The problem can be described by the mathematical formula as:

\[ E = \sum_{i}^{D} \sqrt{(x_{s} - x)^{2} + (y_{s} - y)^{2} + (z_{s} - z)^{2}} \] (1)

3. Method
The flow of getting boundary is made up of four steps as follow: get workpiece from point cloud, get the boundary point, select the edge point and fit the curve of the edge. The strategy will be introduced in the following:

3.1 Extraction of Workpiece From Point Cloud
The point cloud is gotten by the industrial camera, so there is the workpiece and some other things in the point cloud which contain fixture plane mainly, and we must segment them.

In this paper, the algorithm of RANSAC[5] is used to recognize the fixture plane. RANSAC is the abbreviation of Random Sample Consensus. It can calculate the parameter of a mathematical model iteratively from a set of observation data sets containing “outliers”. In this paper, the threshold t is set as 15mm, k is 6000. The point clouds after Section 4.1 is shown as Figure 2. The red presents “inliers”, and the green presents “outliers”.
In the paper, we transform the pointcloud to make the fixture plane on XOY plane. And then, we cut the point whose absolute value of Z coordinate value is smaller than the threshold.

3.2 Extraction of Boundary Points

3.2.1 Denoise The Pointcloud. There will be some noise around the edge of the workpiece, for the error of the camera. In this paper, we use the Statistical Outlier Removal filter[6] to denoise the point cloud. This kind of filter has two parameters,"NumNeighbors” and “Threshold”. The noise points can be caused by two reasons. One is the error of the camera and environment. These points are shown as cluster and farther from the boundary. And the other reason is that the thickness of the workpiece. These points are near the boundary. It has been seen from our practice that the larger “NumNeighbors” can delete the former, and the lower “NumNeighbors” can delete the latter. More small the value of “threshold” is, more better the result is. Therefore, we filter the point cloud twice by (10, 0.1), (50, 0.3).

3.2.2 Simplifications of the points. The number of points is in millions, and if they are all used in the next step, the time is long about 1 hour. Therefore, we must simplify the point cloud. The algorithm is described as follows, in which Workpiece_d, Bon, Boundary_range, workpiece_in, workpiece_in_d are point cloud, and Sphere_search(Co, grid) is a kind of algorithm that searches the nearest points in a sphere whose center is the point to be deal with, and radium is R=Co×Grid:

\[
\text{Algorithm 1: simplification to point cloud} \\
\text{Input:3d point cloud Workpiece, length of box side grid, coefficient co} \\
\text{Output:3d point cloud Workpiece\_simplification} \\
\text{Workpiece\_d}\leftarrow\text{DOWNSAMPLE(Workpiece, grid);} // as in equation (2). \\
\text{Bon}\leftarrow\text{BOUNDARY(Workpiece\_d);} // as in equation (3). \\
\text{Calculate the neighbor points in Workspiece of each point in Bon by Sphere\_search;} \\
\text{Merge all neighbor points into one point cloud Boundary\_range;} \\
\text{Delete Boundary\_range from Workpiece, extract point cloud workpiece\_in;} \\
\text{Workpiece\_in\_d}\leftarrow\text{DOWNSAMPLE(Workpiece\_in, grid);} \\
\text{Workpiece\_simplification}\leftarrow\text{Merge(Boundary\_range, workpiece\_in\_d);} \\
\]

Therein, DOWNSAMPLE is a kind of algorithm to decrease the number of points by voxelization[7], which use the center of a cube to replace the points in the cube. The width of the side of the cube is called Grid. We describe the algorithm as:

\[
\text{pointcloud\_new}=\text{DOWNSAMPLE(pointcloud\_old, grid)} \quad (2)
\]

Grid will influence the number of points in Workpiece\_simplification, which will eventually influence the time of Section 3.2.2–3.2.3, the mainly time in the whole flow. We change the value of Grid, and obtain the difference of the number and time, which will be shown in Section 4.
3.2.3 Extraction of The Boundary Point. After denoising the point cloud, we can get the boundary point from the workpiece. In this paper, a method named micro-cut plane projection[8] is used, as figure 3. For each point, we search its k neighbors, and make the micro-cut plane by these k neighbors. Then we project the point and its k neighbors onto the micro-cut plane, and call the projection P and N_i(i=1,…,K), spectively. We connect the P and N_i to make k vectors, and calculate the angle between adjacent two vectors. If the maximum in the set of angles is larger than the threshold \( t \), we regard the point as the boundary point.

![Figure 3: The program of getting the boundary point](image)

We describe the algorithm as equation (3):

\[
\text{point} \_\text{boundary} = \text{BOUNDARY}(\text{pointcloud})
\] (3)

3.3 Select The Edge Point
In the paper, the workpiece has 4 sides, and they will be all got from the point cloud through the above steps. But the edge to be process is not all the sides. Therefore, we must get the edge we want from the boundary points. The edge we want is a space curve and has two corners. If we get the two corners, it can be selected.

Points in our point cloud are unordered, but the work to get the corner needs a well-ordered point cloud. So the first work to do is to sort the boundary point. In this paper, we choose a point randomly as a starting point, and find its nearest point as the next point. And then, we regard the next point as the new starting point, and find the nearest point in the rest of the points as the next point. Finally, we get a set of well-ordered boundary point cloud.

We use a method named corner detection based on sliding windows[9], which judges if a point is a coner point by curvature. In our paper, the width of the window is constant and set as 17. After this step, there are some points got around the corner. Therefore, we must find the real coner point.

3.4 Fit The Curve of The Edge
After obtaining the coner of the edge to be process, we can regard the boundary points between the two corners as the point of the edge to be process. In this paper, a means named Smoothing spline is used to fit the curve.

The method, “smoothing spline”[10], is based on “least square method”, but the error term is

\[
\text{RSS}(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt
\] (4)

4. Result and evaluation
The combination of Matlab and PCL by C++ on i5-4460, 4G is used to practice the experiment. In Section 3.2.3, the \( k \) is set as 40 and the threshold \( \lambda \) is set as 90°. The boundary point we get is shown as figure 4.
In section 3.4, we fit the curve by $x$, $y$ and $x$, $z$, using equation (4) respectively. We regard $x$ as the independent variable, and regard $y$, $z$ as the dependent variable. Finally, we will get a space curve which fits the edge of the workpiece as figure 5. In this model, $E$, as equation (1) is 0.4180 finally.

We change the value of Grid in Section 3.2.3, and obtain the difference of the number and mean time which is shown in table 1. The point number of initial point cloud is 1310698, and the time is 55 min.

| Grid(Co=1.3) | 2     | 3     | 4     | 5     | 6     | 7     |
|--------------|-------|-------|-------|-------|-------|-------|
| Point Number | 72899 | 68555 | 76350 | 83013 | 85309 | 86998 |
| Time(s)      | 335   | 305   | 333   | 366   | 367   | 371   |

It can be seen from the table 1, that the number and time first decrease and then increase. Therefore, we set the value of Grid from 2 to 4, in order to find the best value.
It can be seen for figure 6 that time is influenced by point number totally, and we can find that the best value of Grid is 2.6, whose number is 65441, time is 300s.

5. Conclusion
In this paper, an edge to be processed needs to be extracted from 3D point cloud. At first, we get workpiece from point cloud by the separation of the fixture. Secondly, the boundary point is got. In the end, the edge point we need is selected and the curve fitting of the edge is done. A time optimizing method is proposed to accelerate the extraction process. Finally, we decrease computing time from 55 minutes to 300 seconds, and the precision is 0.4180 mm.

Acknowledgement
Authors gratefully acknowledge the financial supports by National Science and Technology Major Project of China (No. 2017ZX04005001).

References
[1] Milroy, M. J. , Bradley, C. , & Vickers, G. W. . 1997 . Segmentation of a wrap-around model using an active contour[J]. Computer Aided Design 29(4) 299-320
[2] Yang, M. , & Lee, E. . 1999 . Segmentation of measured point data using a parametric quadric surface approximation.[J].Computer Aided Design 31(7) 449-457
[3] Sam, V. , Kawata, H. , & Kanai, T. . 2012 . A robust and centered curve skeleton extraction from 3d point cloud[J]. Computer-Aided Design and Applications 9(6) 869-879
[4] Lee, K. W. , & Bo, P. . 2015 . Feature curve extraction from point clouds via developable strip intersection[J]. Journal of Computational Design and Engineering, S2288430015000639
[5] Fischler, M. A. , & Bolles, R. C. . 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 24(6), 381-395.
[6] Wolff, K. , Kim, C. , Zimmer, H. , Schroers, C. , Botsch, M. , & Sorkine-Hornung, O. , et al. 2016 . Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction[C]. 2016 Fourth International Conference on 3D Vision (3DV). IEEE Computer Society
[7] C. Connolly. 1984. Cumulative generation of octree models from rangedata[J]. Robotics and Automation. Proceedings. IEEE International Conference on, vol. 1, mar 1984, pp. 25 – 32.
[8] Wu lushen, Yan Haiping, Chen Huawei, Gao Xiangqing 2014 . A Boundary Extraction Algorithm Based On Scattered Point Cloud[J]. Computer Applications and Software(11) 264-268
[9] SUN Xiang-yang, YAO Li, WAN Yan, 2009: Adaptive Corner Detection Algorithm Based on Linear Fitting[J]. Journal of Shanghai University of Engineering Science, 23(1) 46-50
[10] Qiu, & Chunfu. 1995 . Nonparametric regression and generalized linear models: a roughness penalty approach[J]. Technometrics 37(3) 342-343c