

A Point Cloud Contour Extraction Method based on Plane Segmentation

Lei Lu¹, Ran Gao¹, Wei Pan^{2,*} and Wenming Tang³

¹ College of Information Science and Engineering, Henan University of Technology, Zhengzhou Henan, 450001, China

² Research and development department, OPT Machine Vision Tech Co., Ltd, Dongguan Guangdong, 523850, China

³ College of Intelligent Manufacturing and Equipment, Shenzhen Institute of Information Technology, Shenzhen, Guangdong, 518172, China

* Corresponding author: Wei Pan

Abstract: A method based on plane segmentation and dimensionality reduction for extracting incomplete and slow contour features of object point clouds is proposed. The method consists of two main steps: plane segmentation and contour extraction. In plane segmentation, the random sample consensus (Random Sample Consensus, RANSAC) algorithm is optimized based on principal component analysis (Principal Component Analysis, PCA); the optimized planar point cloud is then subjected to dimensionality reduction, and the contour features are extracted using gradients. Experimental results show that the method can effectively segment point clouds and extract the contours of target surfaces, and has great potential for application in industrial inspection and other fields.

Keywords: Three-dimensional Image Processing; Object Detection; Random Sample Consensus; Contour Extraction; Point Cloud Processing.

1. Introduction

Point cloud contour feature extraction is a research hotspot in fields such as machine vision and computer graphics, widely used in industrial inspection, robot operations, and other fields [1-2]. Due to factors such as equipment or environment, the obtained point cloud contains noise and lacks topological relationships, which poses challenges for contour detection. In order to expand the application scope of point cloud data in engineering projects, efficiently and accurately analyzing object point cloud data and rapidly extracting contour features [3-4] have become urgent issues that need to be resolved.

The Hough transform is a traditional method for predicting parameter models. Dorit Borrmann et al. evaluated the applicability of different Hough transforms and proposed the design of the accumulator ball, which proved the high performance of the stochastic Hough transform prediction plane [5]. Anne-Laure Chauve et al. combined a priori urban and architectural scenarios to use iterative area growth algorithms for plane prediction and post-processing for sharp locations [6]. Liangliang Nan et al. employed the iterative RANSAC to predict the plane [7] and it works well for the light weight point cloud. For boundary extraction, based on the triangular mesh constructed by the original point cloud,

Menci Robert transferred the point cloud data to the triangular mesh firstly and achieved boundary feature extraction by restricting the angle between the edges [8]. Xijiang Chen et al. proposed to use the point cloud breakline and regularization method to sort the points according to the vector deflection angle and distance; then, the improved the cubic b-spline curve fitting algorithm is employed to complete the boundary extraction [9]. The alpha shape algorithm proposed by Herbert Edelsbrunner et al. is a scattered contour extraction algorithm, which can be applied to the extraction of polygon inner and outer contour points, but for point clouds with large data volume, the time complexity of the algorithm is high [10].

This article proposes an effective method for extracting contour features of various surfaces of object point clouds. The method involves fusing the random sample consensus (RANSAC) algorithm and principal component analysis (PCA) for plane segmentation, thereby enhancing the robustness and accuracy of point cloud plane segmentation. Additionally, the method includes dimensionality reduction of the point cloud data to transform 3D point clouds into 2D images, thereby improving boundary extraction speed and achieving contour feature extraction of object point clouds.

2. Proposed Method

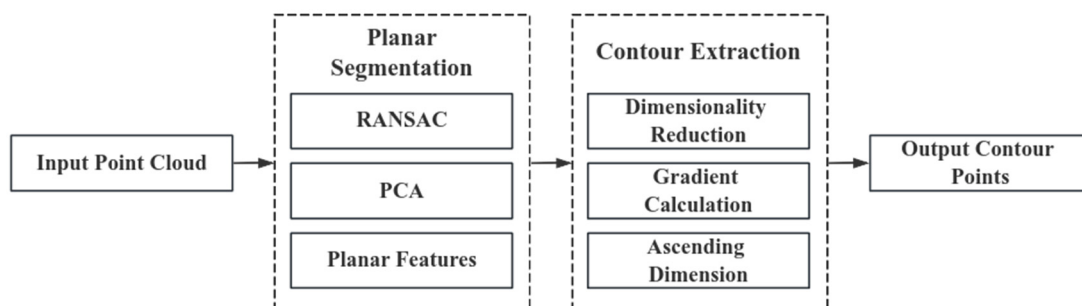


Figure 1. Schematic diagram of the algorithm flow

The proposed method in this paper is mainly divided into two stages, and the algorithm flow is shown in Figure 1. The first stage is the plane segmentation stage, mainly through the fusion of RANSAC and PCA algorithm based on plane features to complete the plane point cloud segmentation. The second stage is contour extraction, which mainly converts the point cloud into a two-dimensional image through dimensionality reduction processing, calculates the two-dimensional boundary points through gradient, and then converts the corresponding positions in three-dimensional space to complete the contour extraction of the point cloud.

2.1. The Planar Segmentation based on PCA and RANSAC

With the obtained point cloud data, the initial planar set is extracted based on RANSAC firstly. Assumes that P is an initial point cloud containing n points and each point is represented as $p_i(x_i, y_i, z_i)$, $i \in [0, 1, 2, \dots, n-1]$. Three points that are far away from each other are randomly selected as the sample point set S . A plane is fitted based on S and is described in Eq. (1):

$$Ax + By + Cz + D = 0, \quad (1)$$

where A, B, C, D are the coefficients of the plane. Rewrite Eq. (1) to Eq. (2):

$$z = \left(-\frac{A}{C}\right)x + \left(-\frac{B}{C}\right)y + \left(-\frac{D}{C}\right), \quad (2)$$

Suppose $a_0 = \left(-\frac{A}{C}\right)$, $a_1 = \left(-\frac{B}{C}\right)$, $a_2 = \left(-\frac{D}{C}\right)$ and the variance of the distance from all points to the plane is F :

$$F = \sum_{i=0}^{n-1} (a_0x_i + a_1y_i + a_2 - z_i)^2, \quad (3)$$

For each point in S minimizing F , we have $\frac{\partial F}{\partial a_0} = 0$,

$\frac{\partial F}{\partial a_1} = 0$, $\frac{\partial F}{\partial a_2} = 0$. The parameters a_0, a_1, a_2 can be obtained by solving Eq. (4) and the initial plane fitting is completed.

$$\begin{cases} \frac{\partial F}{\partial a_0} = 0 \\ \frac{\partial F}{\partial a_1} = 0, \\ \frac{\partial F}{\partial a_2} = 0 \end{cases} \quad (4)$$

Then the outliers in the point cloud P is removed based on the initial fitting plane. First, the distance d_i from each point in P to the initial fitting plane is obtained by Eq. (5). A threshold d_{th} is set and the point is added to S if $d_i < d_{th}$, otherwise, the point is discarded.

$$d_i = \frac{|Ax_i + By_i + Cz_i + D|}{\sqrt{A^2 + B^2 + C^2}}, \quad (5)$$

Implement the above steps iteratively by $N_{iterate}$ times and select the point set S including the largest number of points as the current plane prediction result. Remove the current plane point set from the initial point cloud P and select a new plane S to start a new round plane prediction until the number of the points is less than a threshold N_{minum} . A series of point cloud sets $P_{array} = \{P_1, P_2, \dots, P_{N_{plane}}\}$ can be obtained and N_{plane} is the number of extracted planes.

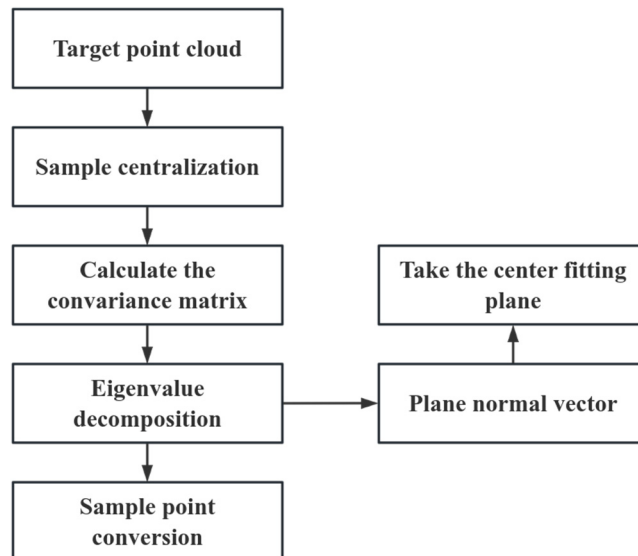


Figure 2. PCA plane fitting algorithm process.

RANSAC can achieve good performance when the point cloud has obvious noise. However, when the noise is close to the plane (such as adjacent planes), errors will be introduced. This paper addresses the above issue by PCA based RANSAC plane fitting method and optimization strategy. The PCA plane fitting process is shown in Figure 2.

At first, PCA plane fitting is applied to all the planes in point cloud sets P_{array} . Suppose that there are n_1 points in P_1 and the corresponding coordinates is (x_i, y_i, z_i) , $i \in [1, 2, \dots, n_1]$. Then the centroid (mx_1, my_1, mz_1) of the point cloud can be obtained by Eq. (6):

$$(mx_1, my_1, mz_1) = \frac{1}{n_1} \sum_{i=1}^{n_1} (x_i, y_i, z_i), \quad (6)$$

Apply Eq. (7) to centralize the point cloud and the sample matrix X_I is obtained:

$$X_I = \begin{bmatrix} x_1 - mx_1 & y_1 - my_1 & z_1 - mz_1 \\ x_2 - mx_1 & y_2 - my_1 & z_2 - mz_1 \\ \dots & \dots & \dots \\ x_{n_1} - mx_1 & y_{n_1} - my_1 & z_{n_1} - mz_1 \end{bmatrix}, \quad (7)$$

Calculate the corresponding covariance matrix E_I :

$$E_I = \frac{1}{n_1 - 1} (X_I^T X_I), \quad (8)$$

where X_I^T is the transposition of X_I . The eigenvalue decomposition of E_I is carried out to obtain its eigenvector and eigenvalue, select the eigenvector with the minimum eigenvalue as the plane normal vector $\mathbf{v}_{\text{normal1}}$, which is combined with the centroid (mx_1, my_1, mz_1) to fit the plane.

Repeat the above process for each planar in P_{array} and N_{plane} planes are obtained. That planes are merged or removed by compare the angles and distances between each other. The optimization strategy is as follows: assume the normal vectors of plane 1 and plane 2 are $\mathbf{v}_{\text{normal1}}$ and $\mathbf{v}_{\text{normal2}}$ respectively, and their angles are defined as θ_{angle12} ; the number of points for the two planes are n_{p1} and n_{p2} . When $n_{p1} + n_{p2} \geq N_{\text{minimum}}$ and $\theta_{\text{angle12}} < \theta_{\text{th}}$ (θ_{th} is the plane angle threshold, which is set to 15° in this paper), calculate the distance from one point to the plane between plane 1 and plane 2, if the proportion of the points whose distance is less than d_{th} is greater than 90%, the point sets of plane 1 and plane 2 are merged. In the other hand, if $n_{p1} + n_{p2} < N_{\text{minimum}}$, the point sets for planes 1 and 2 are removed. Apply the above optimization strategy to all the

planes until no planar can be merged or removed.

2.2. Boundary Extraction based on Point Cloud Dimensionality Reduction

In order to increase the speed of the boundary extraction, the dimensionality of the point cloud data is reduced by transform the 3D data to 2D image. Firstly, the posture of the planar point cloud is adjusted to parallel to the XOY plane based on Eq. (9).

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M_{\text{pose}} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (9)$$

where M_{pose} is the rotation matrix. Secondly, Eq. (10) is used to transform the 3D point cloud data into 2D image.

$$\begin{cases} u = \frac{y'}{f_x} + c_x \\ v = \frac{x'}{f_y} + c_y \end{cases}, \quad (10)$$

where u and v are the row and column coordinates of the 2D image; f_x and f_y are focal lengths; c_x , c_y are offsets. For the convenience of contour extraction, the intensity value of the valid points on the object is set to 0 and the points outside the object is set to 1.

In order to extract the boundary of the object based on 2D image, the Gaussian function is applied to smooth the image. The Gaussian function and denoising filter are shown in Eq. (11) and (12), respectively.

$$H(u, v, \sigma) = \frac{\exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)}{2\pi\sigma^2}, \quad (11)$$

$$S(u, v) = H(u, v, \sigma) * I(u, v), \quad (12)$$

where $H(u, v, \sigma)$ is the Gaussian function, $I(u, v)$ is the original image, $S(u, v)$ is the filtered and denoised image, $*$ represents convolution, σ represents smoothness, if σ is larger, the noise can be better removed, but the edge will be smoothed out; on the contrary, if σ is small, the edge information can be better preserved, but the denoising ability is low.

Then, the smoothed image is processed to obtain the gradient of the horizontal and vertical directions and the direction of the gradient by Eq. (13-16):

$$G_u(u, v) = \frac{S(u+1, v) - S(u-1, v)}{2} \quad (13)$$

$$G_v(u, v) = \frac{S(u, v+1) - S(u, v-1)}{2} \quad (14)$$

$$G_\theta = \sqrt{G_u(u,v)^2 + G_v(u,v)^2} \quad (15)$$

$$\theta(u,v) = \arctan\left(\frac{G_v(u,v)}{G_u(u,v)}\right) \quad (16)$$

where $G_v(u,v)$ and $G_u(u,v)$ represent the gradient images in the row direction and column direction, respectively, G_θ represents the gradient value and $\theta(u,v)$ is the gradient direction.

the preliminary edge points can be obtained according to the gradient. For each boundary points, the intensity values of the adjacent points are compared along the gradient direction.

If the intensity value of the boundary point is the largest one, then keep the point as boundary point. Repeat the comparison so as to obtain all the boundary points of the 2D image. Finally, the corresponding contour points in the three-dimensional space are obtained by inversely transforming the boundary point sets according to Eq. (10) and Eq. (9).

3. Experimental Results and Analysis

In order to verify the effectiveness of the proposed algorithm, the contour features of the chip were extracted by taking the circuit board as the experimental object. The algorithm running on the platform with 12th generation Intel i7-12700 CPU and the running memory is 16G. The experimental objects and point clouds are shown in Figure 3.

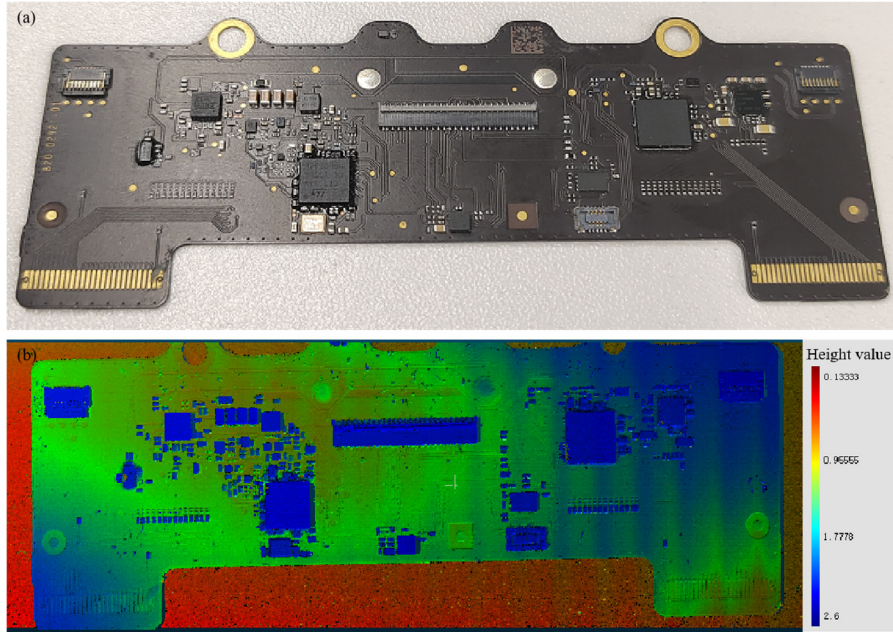


Figure 3. Experimental object circuit board. (a) Physical drawing of the circuit board; (b) Schematic board point cloud

Based on the planar segmentation method mentioned in this article, the initial point cloud is extracted and optimized (as shown in Figure 4). Through the optimization processing, the

noise data is effectively removed, which provides support for the subsequent boundary extraction.

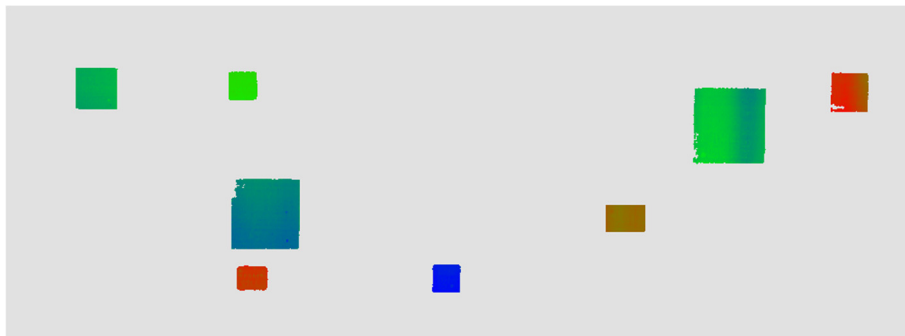


Figure 4. Planar perdition for all target chips by the proposed method.

To verify the effectiveness of the algorithm, the traditional RANSAC method and region growth algorithm are employed to process the same data for comparison. The chip on the right side in Figure 5 is selected and the comparison results are shown in Figure 6. The plane obtained by the traditional RANSAC, the region growth algorithm and the proposed method is shown in Figure 5. It can be clearly seen that the proposed method results in this paper with minimal noise impact.

Then, the point cloud rotation matrix is employed to

transform the plane point cloud data to the position paralleling to the XOY plane, then the dimensionality reduction process is carried out to convert the 3D point cloud data to 2D image. Finally, the boundary points are extracted according to the gradient change and turn back to the corresponding position in the space. In this experiment, the alpha-shapes algorithm [10] is employed to conduct a comparative experiment for boundary extraction and the results are shown in Figure 6. The detail of the boundary extraction is given in Table 1. It is apparent that the proposed method achieves accuracy result

with high efficiency.

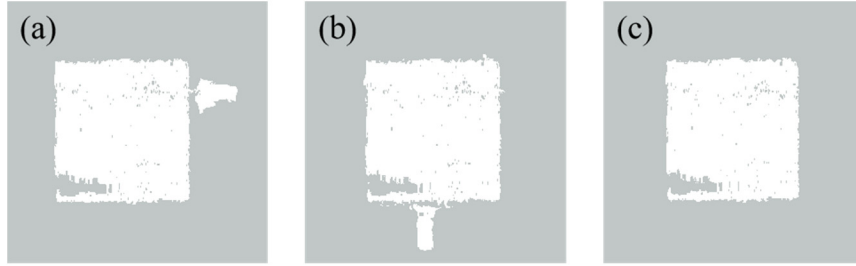


Figure 5. Comparison of planar prediction results. (a) Region Growing; (b) RANSAC; (c) PCA-RANSAC.

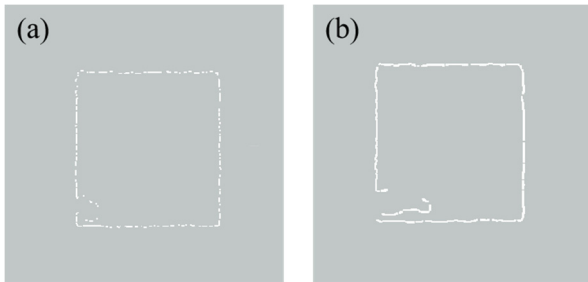


Figure 6. Boundary extraction comparison. (a) result by alpha shapes; (b) result by the proposed method

Table 1. Comparison of contour extraction data.

Methods	Number of Points in Point Cloud	Number of Boundary Points	Cost Time/ms
Alpha shapes	62668	514	745.253
Our method	62668	919	64.614

All the contour point extraction results of the target point clouds are shown in Figure 7.

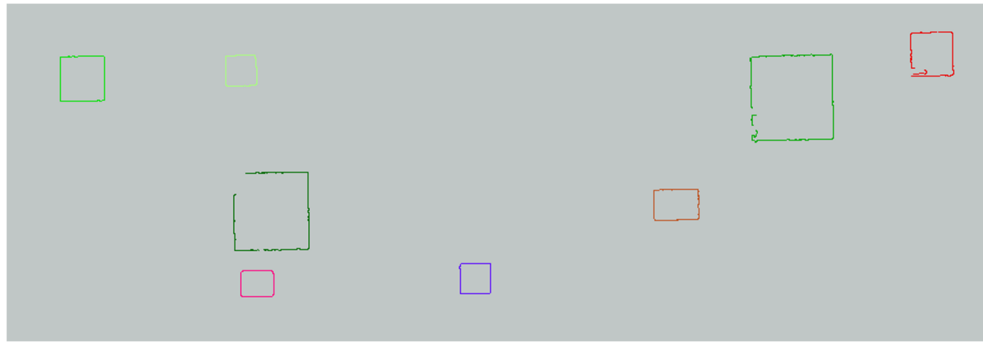


Figure 7. Contour extraction result

4. Conclusion

In this paper, a highly robust point cloud contour extraction method based on plane segmentation is proposed. Firstly, the PCA algorithm is employed to optimize the planar predication result obtaining by RANSAC. Secondly, in order to improve the boundary extraction efficiency, the dimensionality reduction is applied to transform the 3D point cloud to 2D image; Finally, the gradient information is used to extract the boundary point set and turn back to 3D space. Experimental results show that the proposed method can effectively extract the contour features from the point cloud and has a wide application prospect in the field of industrial measurement.

References

- [1] Emelyanov, A.; Skala, V. Surface Reconstruction from Problem Point Clouds. International Conference on Planarization Technology 2002, 5, 6-75.
- [2] Kim, J.; Jin, M.; Park, S.H.; Chung, S.Y.; Hwang, M.J. Task Space Trajectory Planning for Robot Manipulators to Follow 3-D Curved Contours. Electronics 2020, 9, 1424.
- [3] Hackel, T.; Wegner, J.D.; Schindler, K. Joint classification and contour extraction of large 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 2017, 130, 231-245.
- [4] Venkatesh, Y.V.; Raja, S.K.; Ramya, N. Multiple contour extraction from graylevel images using an artificial neural network. IEEE Transactions on Image Processing 2006, 15, 892-899.
- [5] Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. 3D Research 2011, 2, 1-13.
- [6] Chauve, A.L.; Labatut, P.; Pons, J.P. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, USA, 13-18 June 2010.
- [7] Nan, L. L.; Wonka, P. PolyFit: Polygonal Surface Reconstruction from Point Clouds. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22-29 October 2017.
- [8] Mencl, R. A Graph-Based Approach to Surface Reconstruction. Computer Graphics Forum 1995, 14, 445-456.
- [9] Chen, X.J.; Yu, K. Feature Line Generation and Regularization from Point Clouds. IEEE Transactions on Geoscience and Remote Sensing 2019, 57, 9779-9790.
- [10] Edelsbrunner, H.; Mücke, E. Three-dimensional alpha shapes. 3D Research 1994, 2, 1-13.