ISSN: 2710-0723

Preliminary Point Cloud Data Analysis for Thin-Walled Part Defect Detection based on Point Cloud Data Processing

Yuan Cao[a], Yongxiang Jiang[b], Chaofan Hong[c], Yanxi Guo[d], Yuanlong Xu[e], Jiayi Li[f]

School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin, China

[a]3238330442@qq.com, [b]jiangyongxiang@tute.edu.cn, [c]1919742227@qq.com, [d]1244420211@qq.com, [e]3094943674@qq.com, [f]3343707068@qq.com

Abstract

This paper studies the processing of point cloud data for objects, focusing on improving the bilateral filtering algorithm to address poor denoising effects by removing edge noise while preserving features. It employs voxel grid downsampling to streamline large datasets. During the point cloud registration phase, a standard plane is first fitted before registration, introducing the FPFH and SAC coarse registration and ICP fine registration algorithms to enhance registration outcomes. In the defect detection task, defect data points are extracted by calculating Euclidean distances, and the greedy projection triangulation method is used to reconstruct surfaces for visualization and quantification. This approach provides data support for subsequent work, aiding in assessing the extent of object damage.

Keywords

Point Cloud Data Processing; Thin-Walled Parts; Defect Detection; Data Analysis; Computational Methods.

1. Introduction

This paper provides a brief overview of the specific implementation processes for point cloud denoising and simplification algorithms during the preprocessing stage. It emphasizes analyzing the obtained data through a series of steps, including denoising and registration, applied to the point cloud data of thin-walled cover parts. After processing the point cloud data through filtering and simplification, a standard point cloud model representing a damage-free surface is obtained through the registration operation. By comparing the standard point cloud with the damaged point cloud, defect points are identified.

During the comparison of the two point clouds, a coarse registration followed by a fine registration approach is adopted. The goal of coarse registration is to roughly align the two point clouds, serving as the initial condition for fine registration. Fine registration is then performed to achieve a more optimal alignment. The paper primarily discusses this data analysis process and related methodologies.

2. Research on Point Cloud Preprocessing Algorithms for Thin-Walled Cover Parts

2.1. Causes of Noise Generation and Types of Noise in Point Clouds of Thin-Walled Cover Parts

This section analyzes two common types of defects to illustrate the algorithms and related steps in the point cloud data processing. The point clouds of the two thin-walled cover parts obtained using the established stereo structured light system are shown in Fig. 1 and Fig. 2. It is evident

ISSN: 2710-0723

that there are significant noise points on the upper surface and the edges of the thin-walled cover parts, with some noise points already marked. This subsection studies the causes of noise generation and classifies the types of noise.

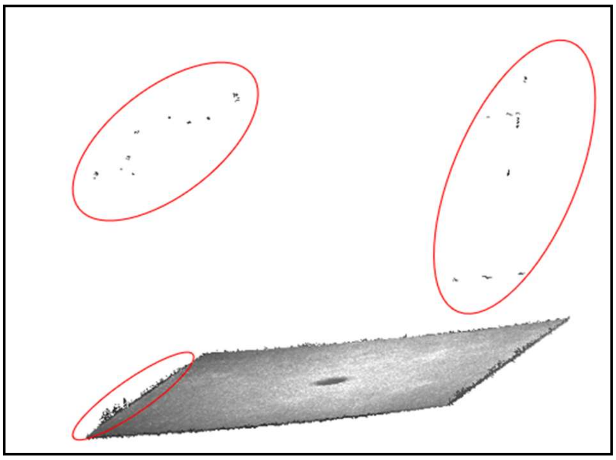


Fig. 1 Point Cloud of Thin-Walled Cover Parts (case 1)

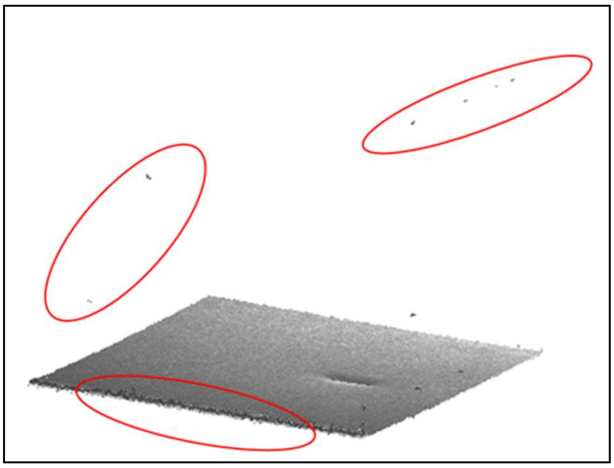


Fig. 2 Point Cloud of Thin-Walled Cover Parts (case 2)

The spatial distribution of noise points in the 3D point cloud data of a target object is irregular, making it difficult to distinguish them using an appropriate mathematical model. Based on the locations of the noise points, they can be categorized into three main types:

(1) Isolated Points: These are points located far from the target point cloud area and feature a high density. Such points have no connection to the target and can be defined as noise points.

(2) Redundant Points: These refer to excess data points collected during the acquisition process that extend beyond the predefined collection area.

(3) Mixed Points: These are noise points within the point cloud that are relatively close to the target point cloud but do not belong to it.

Among the three types of noise points, the first two categories of noise points are classified as outlier points due to their distance from the target point cloud and lower density. The third type of noise point, which is mixed with the main target point cloud and adheres to the surface of the measured object, is referred to as internal noise.

ISSN: 2710-0723

2.2. Traditional Point Cloud Filtering Algorithms

2.2.1. Pass-Through Filtering Algorithm

Pass-Through filtering, also known as a pass-through filter, is a straightforward filtering method compared to other filtering algorithms. This filter is commonly used in the processing of point cloud data. Its basic principle is to set a threshold range within a specified area to divide the point cloud data into two parts: those within the range and those outside the range. Depending on the requirements, points can be retained or filtered out.

Pass-Through filtering is particularly effective for processing point cloud data with specific spatial distribution features. For instance, when point cloud data is acquired using a stereo camera, it often leaves a significant amount of noise points along the z-axis, while the distribution along the x and y axes is relatively limited. In such cases, a pass-through filter can be used to identify the range of point clouds in the x or y direction and quickly remove outlier points, thereby achieving a noise reduction effect.

Fig. 3 shows the initial point cloud obtained, where it can be observed that the circled area contains noise points. Fig. 4 illustrates the result after applying the pass-through filtering algorithm. It is evident that this method effectively removes noise points that are far from the main point cloud; however, noise points that are close to the main point cloud still remain, and there are many noise points along the edges of the main point cloud.

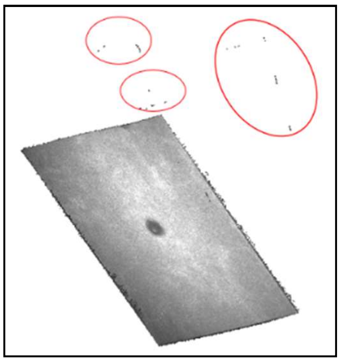


Fig. 3 Before Pass-Through Filtering

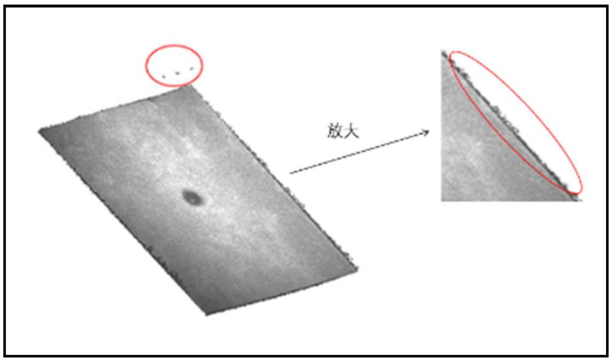


Fig. 4 After Pass-Through Filtering

2.2.2. Statistical Filtering

Statistical filtering is also a common filtering method that uses statistical analysis to remove measurement noise points. The statistical filtering algorithm traverses each point in the target point cloud data and performs statistical analysis on the neighborhood of each point, eliminating neighborhood points that do not meet certain criteria. Specifically, for each point, the statistical filter calculates the average distance to all neighboring points. If the resulting point cloud data distribution follows a Gaussian distribution, it is possible to compute a mean and a standard deviation. Points with distances greater than a certain range can be defined as outliers, with this range being related to the mean and standard deviation, which can be set based on the specific conditions of the point cloud.

The mathematical calculation process is as follows:

Perform statistical analysis on the neighborhoods of each point and calculate the average distance from each point to its nearest k (specified) neighbors. Assuming that the distances between all points in the point cloud follow a Gaussian distribution, and based on the mean μ and standard deviation σ. Defining the coordinates of the n-th point in the point cloud be $P_n\left(X_n, Y_n, Z_n\right)$, and the distance from this point to a certain point $P_m\left(X_m, Y_m, Z_m\right)$ in space is given by:

ISSN: 2710-0723

$$S_i = \sqrt{(X_n - X_m)^2 + (Y_n - Y_m)^2 + (Z_n - Z_m)^2} \quad (1)$$

The formula for calculating the average distance between each point and any other point is:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} S_i \quad (2)$$

The standard deviation σ :

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(S_i - \mu)^2} \quad (3)$$

Using std of the standard deviation, two thresholds k and std are pre-defined. When the average distance of a point from its k neighbors is within, that point is retained; if it exceeds this range, it is considered an outlier and is deleted. The flowchart is shown in Fig. 5.

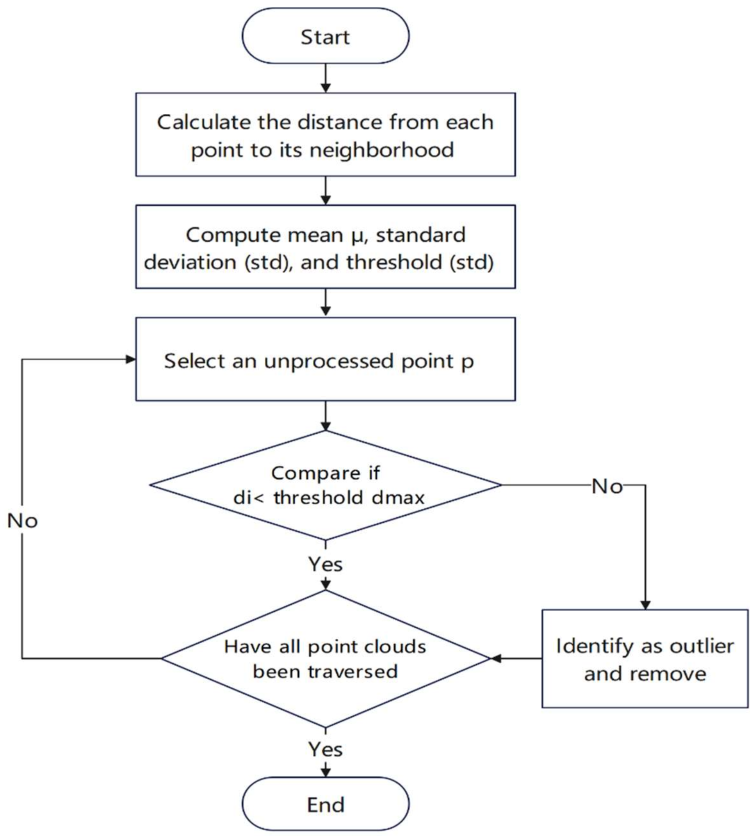


Fig. 5 Statistical Filtering Flowchart

2.3. Point Cloud Simplification for Thin-Walled Cover Parts

When acquiring 3D point clouds using a stereo camera, the volume of data is substantial and includes noise points. Factors such as the accuracy of the acquisition equipment and the surrounding ambient light contribute to noise generation. Different sized objects being measured result in variations in the number of points in the point cloud; generally, larger objects yield point cloud data in the millions or even more. The sheer volume of point cloud

ISSN: 2710-0723

data entails significant computational requirements, which can affect the efficiency of subsequent point cloud data processing. Therefore, it is necessary to perform point cloud simplification as a critical step.

Common methods to achieve point cloud simplification and reduce the volume of point cloud data include the following two approaches:

(1) Random Resampling Method

The random resampling method[1] is a preprocessing technique for point cloud data that reduces the number of points by randomly discarding data points. The advantage of this method is its speed, allowing for the setting of a desired target number of points to retain. However, due to its random nature, the random resampling method may result in uneven point cloud data distribution, potentially leading to the loss of important features.

(2) Voxel Grid Downsampling Method

Voxel grid downsampling[2] is a fast processing method for point cloud simplification. The main idea is to enclose all the three-dimensional point cloud data to be processed within a bounding box and set the side length of the bounding box according to specific requirements. The bounding box is then divided into uniform cubic grids. Subsequently, all the three-dimensional point cloud data is categorized into the corresponding cubic grids. Within each cubic grid, the centroid can be used to replace all the data points in that grid, thereby achieving simplification of the point cloud data.

By representing each voxel grid with its centroid, the shape of the point cloud can be preserved while balancing the distribution density of the point cloud data. This method is highly efficient. Therefore, this paper chooses the voxel grid downsampling method to reduce the quantity of three-dimensional point cloud data.

The specific steps of the algorithm are as follows:

Assuming there are a total of N data points in the point cloud data to be processed, and the number of points in each small cubic grid (voxel) is denoted as n, then the following relationships can be established:

$$n = \frac{N}{V} \tag{4}$$

In the equation, V represents the volume of the bounding box, which can also be expressed using the following formula:

$$V = L_x L_y L_z \tag{5}$$

Where Lx is the maximum distance of the point cloud along the X-axis; Ly represents the maximum distance along the Y-axis; and Lz indicates the maximum distance along the Z-axis. The side length L of the cubic voxel grid can be expressed as:

$$L = \partial \sqrt[3]{\frac{s L_x L_y L_z}{N}} \tag{6}$$

Where ∂ is the scaling factor. To change the side length of the cubic voxel grid, you can adjust the value of ∂; s is scaling coefficient.

ISSN: 2710-0723

After setting up the three-dimensional voxel grid, each point in the point cloud data needs to be assigned to the corresponding voxel. To achieve this, each point in the three-dimensional point cloud data is encoded to determine its location within the voxel grid. Then, the normal vector for each point is computed.

Next, the normal vectors of the points within the same voxel are calculated to determine the angles between them. If the angle between the normal vectors within the same voxel exceeds a predetermined threshold, it is not appropriate to replace all the data points in that voxel with a single centroid. In this case, the scaling factor ∂ must be adjusted, and the preceding calculations for centroids should be repeated until the predetermined threshold is met.

Finally, for each cubic voxel, the centroid can be expressed as:

$$\begin{cases} X_{bp} = \dfrac{\sum_{i-1}^{g} x_i}{g} \\ Y_{bp} = \dfrac{\sum_{i-1}^{g} Y_i}{g} \\ Z_{bp} = \dfrac{\sum_{i-1}^{g} Z_i}{g} \end{cases} \tag{7}$$

3. Research on Defect Detection Methods for Thin-Walled Parts

After the preprocessing of the point cloud data, in order to identify defects, it is necessary to fit the defective point cloud to obtain a standard point cloud model that is free of defects. Subsequently, point cloud registration and other processing can be carried out. This paper employs the RANSAC (Random Sample Consensus) 6519 algorithm to obtain the standard point cloud model, and uses a combination of coarse registration and fine registration methods to align the defective point cloud with the fitted point cloud model, where point cloud feature descriptors PFH[3] and FPFH[4] are used for coarse registration, and the ICP[5] (Iterative Closest Point) algorithm is used for fine registration based on the coarse registration.

Assuming that there exist point clouds P and Q, the source point cloud P can be represented as:

$$P = \left\{ p_i \in R^3, i = 1,2,3,\ldots,N_p \right\} \tag{8}$$

The target point cloud Q can be represented as:

$$Q = \left\{ q_i \in R^3, i = 1,2,3,\ldots,N_q \right\} \tag{9}$$

The ICP (Iterative Closest Point) algorithm is as follows:

(1) For each point in point cloud P, find the nearest point in point cloud Q as its corresponding point;

(2) Construct the objective function based on all corresponding point pairs, as follows:

$$f(R,t) = \sum_{i=1}^{N} \left\| Q_i - (RP_i + t) \right\|^2 \tag{10}$$

ISSN: 2710-0723

Where $Q_i \in Q$, $P_i \in P$, and P is the corresponding point in point Qi; N is the number of corresponding points between the two point clouds

Calculating the rotation matrix R and the translation vector t that minimize the objective function, and record the error value εk at this stage.

(3) Performing a rigid transformation on the point cloud using the calculated rotation matrix R and translation vector t to obtain the transformed point cloud P’:

$$P' = RP + t \tag{11}$$

(4) Replacing point cloud P with the transformed point cloud P’ and repeat the previous steps until the difference in error values $|\varepsilon_k - \varepsilon_{k-1}|$ between consecutive iterations is less than a predetermined threshold ε or until the number of iterations K exceeds a preset maximum iteration count.

Finally, the resulting registered point clouds are illustrated as follows:

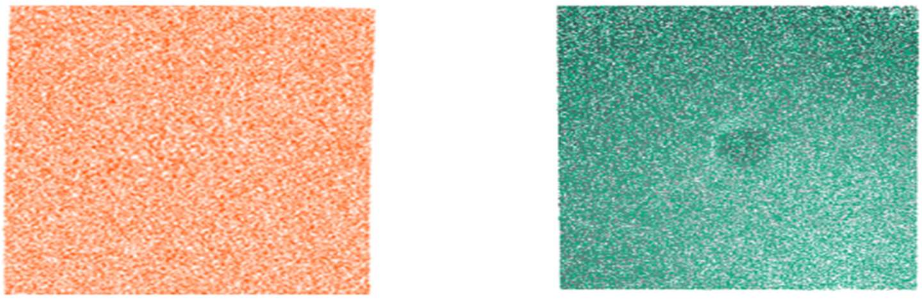


Fig. 6 The resulting registered point clouds

The commonly used metric for point cloud registration is error analysis, and this paper uses the Root Mean Square Error (RMSE) to analyze the errors. The RMSE calculation formula for the registration of the source point cloud P and the target point cloud Q is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - q_i)^2} \tag{12}$$

Where n is the number of points in the source point cloud P, and after registration is completed, pi and qi represent a set of corresponding point pairs. RMSE, as a discriminative metric, represents the root mean square error of the distances between the corresponding point pairs in the source point cloud P and the target point cloud Q. A smaller RMSE value indicates a smaller discrepancy, reflecting better surface registration quality.

The registration metrics are shown in Table 1.

Table 1. Evaluation index results of different registration algorithms

Name	Method	Registration Error(μm)	Registration Time(s)	Iterations
Thin-Walled 1	PFH+ICP	3.481	8.32	11
	PHFH+ICP	0.514	3.75	7

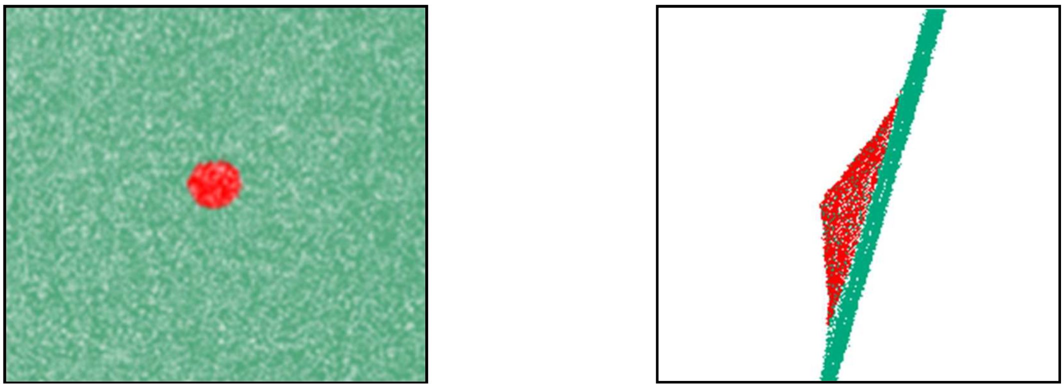
By analyzing the visualization effects of point cloud data registration, the registration root mean square error, and the time required, it was found that when using the PFH+ICP algorithm for registration, the defective point cloud and the fitted point cloud did not completely overlap after registration. The registration results at the edge details of the point cloud data were not ideal, with a registration root mean square error magnitude reaching 3.481 μm.

In contrast, the proposed PHFH+ICP registration method not only has a slightly lower time complexity but also provides outstanding registration results. Observations from the point cloud model indicate almost complete overlap, with the root mean square error reduced to 0.514 μm. This level of accuracy meets the high precision requirements for point cloud registration.

After registering the defective point cloud with the standard point cloud, the differences in the registered points represent the defects that need to be identified. This paper identifies foreign impact damage defects by calculating the Euclidean distance between the point clouds. Assuming there are two data points qi and qj, and their corresponding normal vectors are $n_i = \left(a_i, b_{i,} c_i\right)^T$ and $n_j = \left(a_j, b_j, c_j\right)^T$. The Euclidean distance between these two points can be calculated using the formula:

$$d_{ij} = \sqrt{\left(a_i - a_j\right)^2 + \left(b_i - b_j\right)^2 + \left(c_i - c_j\right)^2} \tag{13}$$

The specific approach involves calculating the Euclidean distance d from each point on the defective point cloud to all points in the standard point cloud and comparing d with a predefined threshold D. Points with a distance less than the threshold D are classified as non-defective points, while points greater than the threshold are deemed defective points. These defective points are then stored in the defective point cloud collection and marked in red, completing the visualization of the defective point cloud.



(a) Front View (b) Defect Magnified View

Fig. 7 Defect Display Image

4. Conclusion

Due to the prolonged exposure of thin-walled parts to external environments, they may exhibit a certain degree of dent damage. To address this, a detection algorithm based on three-dimensional point clouds for identifying defects is proposed. First, the stereo equipment is calibrated to obtain point cloud data with defects. Next, filtering operations are applied to these defective point clouds to remove outlier points. Following this, point cloud simplification is performed to improve computational efficiency. Subsequently, the RANSAC algorithm is utilized to obtain a fitted point cloud that is free of defects. The fitted point cloud is then subjected to both coarse and fine registration with the defective point cloud. Finally, the

ISSN: 2710-0723

Euclidean distance between the two point clouds is calculated and compared with a predetermined threshold. If the conditions are met, the defects are highlighted in red, thus achieving defect visualization and providing a foundation for subsequent measurements of defect sizes.

Acknowledgement

Project Support: College Student Innovation and Entrepreneurship Project, supported by the design of a welding inspection kit system based on machine vision. Number 202410066009.

References

[1] Wu Lushen, Li Ze, Chen Huawai, et al. Research on Improved Resampling Algorithm. Journal of Mechanical Design and Manufacturing, 2015(04): 244-247.

[2] Yuan Hua, Pang Jiankang, Mo Jianwen. Research on Point Cloud Simplification Algorithm Based on Voxel Grid Downsampling. Television Technology, 2015, 39(17): 43-47.

[3] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms[C]//2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2008: 3384-3391.

[4] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.

[5] Huang Y, Da F, Tao H. An automatic registration algorithm for point cloud based on feature extraction[J]. Chin. J. Lasers, 2015, 42(3): 242-248.