remote sensing

MDPI

Article

Feature Surface Extraction and Reconstruction from Industrial Components Using Multistep Segmentation and Optimization

Yuan Wang [1], Jiajing Wang [2,*], Xiuwan Chen [1], Tianxing Chu [3], Maolin Liu [1] and Ting Yang [1]

[1] Institute of Remote Sensing and GIS, Peking University, No. 5 Yiheyuan Road, Haidian District, Beijing 100871, China; pkuwangy@pku.edu.cn (Y.W.); xwchen@pku.edu.cn (X.C.); maolin@pku.edu.cn (M.L.); hello_yangting@126.com (T.Y.)

[2] Tencent Technology (Beijing) Company Limited, China Technology Trade Center, No.66 North 4th Ring West Road, Hai Dian District, Beijing 100080, China

[3] Conrad Blucher Institute for Surveying and Science, Texas A&M University-Corpus Christi, 6300 Ocean Drive, Corpus Christi, TX 78412, USA; tianxing.chu@tamucc.edu

* Correspondence: whu_wangjj@163.com; Tel.: +86-10-6274-5183; Fax: +86-10-6276-7119

Received: 17 June 2018; Accepted: 4 July 2018; Published: 5 July 2018

Abstract: The structure of industrial components is diversified, and extensive efforts have been exerted to improve automation, accuracy, and completeness of feature surfaces extracted from such components. This paper presents a novel method called multistep segmentation and optimization for extracting feature surfaces from industrial components. The method analyzes the normal vector distribution matrix to segment feature points from a 3D point cloud. The point cloud is then divided into different patches by applying the region growing method on the basis of the distance constraint and according to the initial results. Subsequently, each patch is fitted with an implicit expression equation, and the proposed method is combined with the random sample consensus (RANSAC) algorithm and parameter fitting to extract and optimize the feature surface. The proposed method is experimentally validated on three industrial components. The threshold setting in the algorithm is discussed in terms of algorithm principles and model features. Comparisons with state-of-the-art methods indicate that the proposed method for feature surface extraction is feasible and capable of achieving favorable performance and facilitating automation of industrial components.

Keywords: 3D point cloud; feature surface extraction; RANSAC; region growing; segmentation and optimization; industrial components

1. Introduction

Industrial components are essential parts of machinery manufacturing processes, and their structural information exerts an important influence on industrial production. Benefiting from the advances in sensor technology for laser scanning, dense 3D point clouds have become increasingly common [1]. Industrial components can be digitized into 3D point cloud models that represent objects' external surfaces with a large number of points, and can be freely accessed in computers [2]. There are several 3D data acquisition techniques, which are classified into two groups: passive methods and active methods. In passive methods, the problems are illumination, shadow, and complicated image analysis, etc. So, generally, for industrial quality inspection, passive methods are not used. Active methods include time of flight, phase difference, conoscopic holography, and laser triangulation, etc. The first two methods are mostly suitable for large objects such as rooms or buildings because they cannot easily achieve high precision. Conoscopic holography and laser triangulation are suitable for small objects with simple construction and high measurement speed. For a demanding industrial purpose like a distance not more than 1 or 2 m, laser triangulation is used very commonly [3]. Previous studies that used 3D point cloud models to obtain detailed features from industrial components focused on high-redundancy points without examining point features [4]. Point features reveal spatial relations between neighboring points and essential topologic information for surface extraction. However, automatic feature extraction is usually difficult due to the limited point density of models [5], relatively high measurement noise [6], and uneven distribution of cloud points [7]. Many feature extraction routines, such as sampling [8], georeferencing [9], and smoothing [10], still require human intervention and manual operations.

Recent studies have proposed effective feature extraction methods based on curvature [11–15], normal vectors [16,17], local neighborhood distribution [18,19], and so forth. Misclassification, over-segmentation, and mixed models often occur during feature extraction because a large number of discrete points exist in models with noise and outliers [20]. Qin et al. [19] used the principal component analysis (PCA) method based on point coordinate information to extract building feature lines from a point cloud. This method relies on the distribution of the local neighboring points; therefore, it may cause misclassification when the distribution of discrete points is inconsistent with the actual surface. Several studies have proven that a surface normal provides useful data for reconstruction methods [21,22]. Ni et al. [5] proved the usefulness of the local normal vector and overcame the dependence on the distribution of local neighborhood points.

Clustering algorithms are also frequently employed in feature surface extraction [23,24]. A clustering algorithm manually determines the number of final aggregated collections and adjusts fine-grained patches in the results. Region growing, another commonly used algorithm for feature surface extraction, selects seeds and gathers the nearest neighbors in the seed area on the basis of certain constraints [16,25]. Minimal human intervention is required in region growing, but seed selection and the method's limiting conditions remain crucial issues. Vo et al. [26] proved the effectiveness of region growing in surface construction. In the meantime, they found that the method requires a considerable amount of time for parameter tuning. Wang et al. [27] confirmed that region growing can automatically extract features with minimal manual intervention.

The topology of object surfaces that vary in shape and size is supposed to be comprehensively considered to extract accurate surfaces. Random sample consensus (RANSAC) has been widely used to extract optimized surfaces robustly with local and global spatial constraints [28–30]. Wu [31] proved the effectiveness of RANSAC for feature extraction and shape recognition. The surface obtained after feature surface extraction needs to be represented and reconstructed, and implicit field is an effective method for this task [32,33]. The implicit description method can generate tightly closed 3D models from discrete data points [34], repair missing data points, fill holes, and filter noise that is introduced during the acquisition of target objects [25]. However, the generated 3D dataset may contain numerous points. Hence, a reduced-sized dataset should be established for obtaining accurate implicit equation. Combined fragment implicit description divides a 3D dataset into patches according to certain rules, and each patch is fitted by different implicit functions before the fitting results are combined [35]. An advantage of this method is that it enables low computational load and a small amount of data in addition to local smoothing [36].

Parameter expression is a primary tool used for describing the geometry of an object [37], especially in industrial manufacturing where parametric surfaces are widely applied [14]. Nonuniform rational B-spline (NURBS) is supported by much advanced 3D modeling software [38,39], and it can parameterize a surface model, thus facilitating the display and reuse of each platform. Therefore, the advantages of implicit and parameter expressions in surface reconstruction are supposed to be comprehensively utilized to obtain improved models.

In this study, we establish a novel method to extract feature surfaces from industrial components. The method adopts PCA based on the normal vector distribution matrix to analyze local features in the 3D point cloud. Patch classification is then performed based on the preliminary extraction results and shape features of the patches. Two forms of implicit and parameter expressions are used for surface reconstruction after combing the RANSAC algorithm and parameter fitting to extract and optimize feature surfaces. This method is designed to improve the automation and reliability of feature surface extraction, rapidly obtain high-level surface geometric feature information from the original point cloud data, and provide effective routes for automated model construction in the reverse engineering design of industrial components.

2. Method

The proposed method uses PCA based on the normal vector distribution matrix to classify feature points as surface, edge, or corner points. For the set of surface points, the region growing method with a distance constraint is applied to obtain divided patches. These patches can be classified as regular quadratic, complex, and free-form surfaces according to their shape. In this study, we apply implicit and parametric surface models to reconstruct the feature surface. First, the RANSAC algorithm is used to analyze the implicit expression on patches, and the remaining edge and corner points that are not classified are substituted into the implicit expression to expand the patches. Second, the patches that fail to find a surface are expanded based on the distance constraint in the unclassified point space. Finally, the discrete point set of completely expanded patches is fitted using NURBS. The flowchart of the proposed method, called multistep segmentation and optimization (MSSO), is shown in Figure 1.

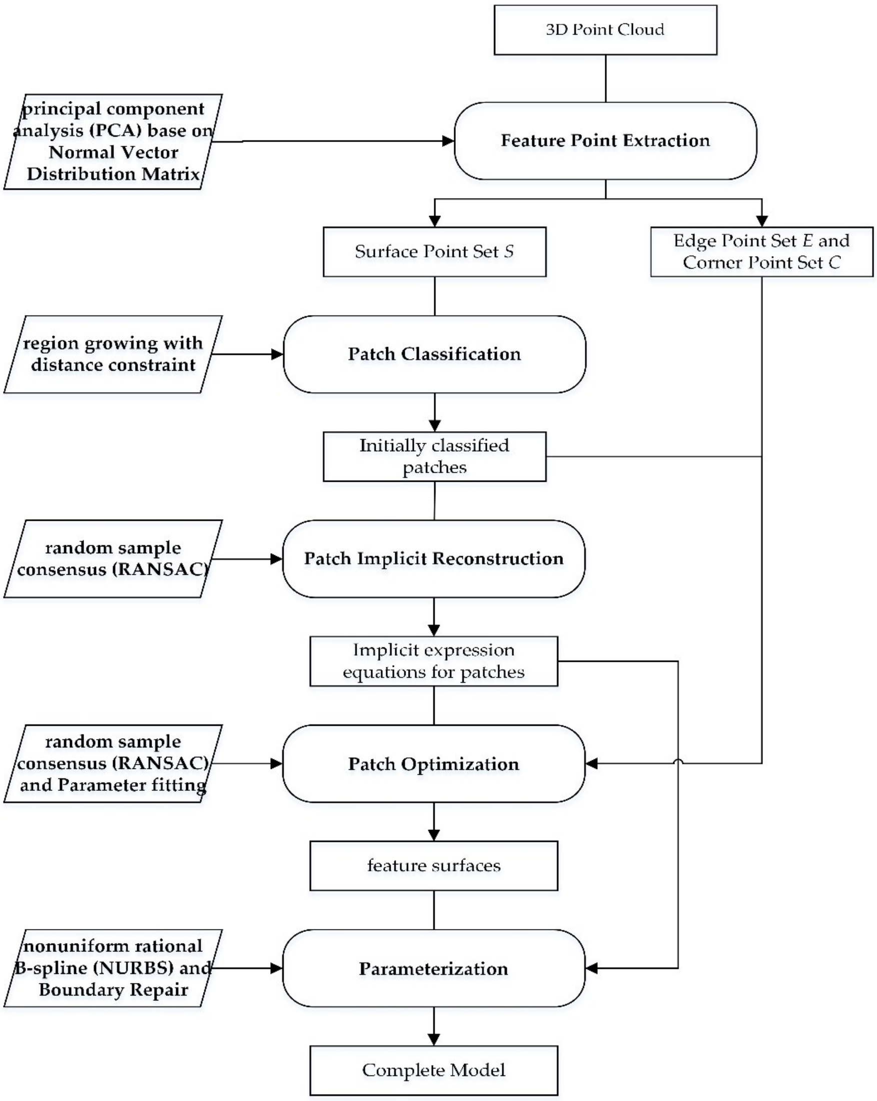

3D Point Cloud
principal component analysis (PCA) base on Normal Vector Distribution Matrix
Feature Point Extraction
Surface Point Set S
Edge Point Set E and Corner Point Set C
region growing with distance constraint
Patch Classification
Initially classified patches
random sample consensus (RANSAC)
Patch Implicit Reconstruction
Implicit expression equations for patches
random sample consensus (RANSAC) and Parameter fitting
Patch Optimization
feature surfaces
nonuniform rational B-spline (NURBS) and Boundary Repair
Parameterization
Complete Model

Figure 1. Flowchart of the feature surface extraction method.

2.1. Feature Point Segmentation Based on the Normal Vector Distribution Matrix

The distribution features of discrete points can be obtained by analyzing the local neighboring point coordinates via PCA. Point misclassification occurs when the distribution of discrete points is inconsistent with the actual topology relation of the surface. The red dots in Figure 2 illustrate the K nearest neighbors [40] of a point on the edge. The neighboring points are arranged in a planar shape, indicating that the points are incorrectly divided into the surface feature using PCA [19].

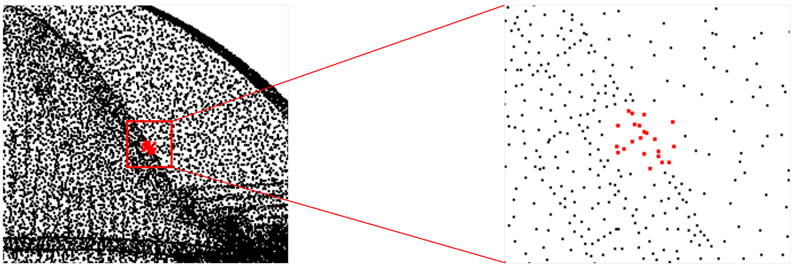


Figure 2. *K* nearest neighbor set of a point on the edge.

The original 3D point cloud contains point coordinates but lacks topological connection information between points. A point normal vector reflects spatial characteristics in a local surface area. Therefore, the neighboring relation between normal vectors should be obtained before other procedures. The fandisk model's original point cloud is shown in Figure 3a, and its normal vectors are calculated with the local plane fitting method [41]. The local neighbors of each point can be approximately regarded as a plane due to the large number of points in the point cloud model [42]. For discrete point p in the point cloud, its K nearest neighbors set $Q = \{q_1, q_2, \ldots, q_K\}$ is searched, and Equation (1) is used to calculate the plane:

$$P(n, d) = \arg\min \sum_{i=1}^{k} \theta(\| p - q_i^{\circ} \|)(n \cdot q_i - d)^2 \tag{1}$$

where d is the distance between point p and fitting plane P and n is the normal vector of the fitting plane. The effect of the distance from neighbor point q_i to point p on local features is considered, and the distance from p to projection point q_i° of neighbor point q_i on the fitting plane is used as a variable of the weight function θ. A moving least squares method is used to fit the plane and calculate the plane equation, and its normal vector is the normal vector of the point. Figure 3b indicates the directions of the normal vectors with purple lines. Figure 3c presents a locally enlarged display of normal vectors where the red line illustrates the surface edge. In this figure, the normal vectors on the same surface are in the same direction, and the angle between normal vectors of two adjacent surfaces is equally divided by the normal vector of a point on the edge. In this study, we use point normal vectors to replace the point coordinate information in PCA to improve the performance of local feature extraction.

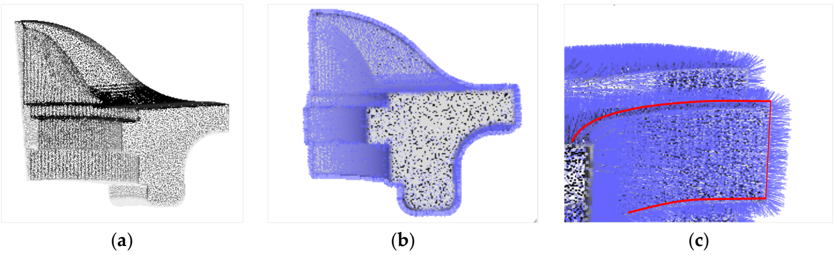


Figure 3. Normal vector of the fandisk model [43]: (**a**) original point cloud; (**b**) normal vectors; and (**c**) locally enlarged display of normal vectors.

For each discrete point, the normal vector distribution matrix is used to segment the feature points. First, the K nearest neighbors set of point p_i is searched. Second, the normal vectors of the nearest neighbor points are calculated by Equation (1), and the normal vectors are substituted in Equation (2) to construct the normal vector distribution matrix:

$$A = \begin{bmatrix} n_1 & n_2 & \cdots & n_K \end{bmatrix} \begin{bmatrix} n_1 & n_2 & \cdots & n_K \end{bmatrix}^T \tag{2}$$

where n_i ($i = 1 \ldots K$) is the normal vector of a neighboring point, and A is a symmetric, semipositive definite matrix. Third, the three non-negative eigenvalues are acquired by applying PCA on the normal vector distribution matrix and are normalized, as shown in Equation (3):

$$\delta_i = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

where λ_i ($i = 1, 2, 3$) is the eigenvalue and $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and δ_i ($i = 1, 2, 3$) is the normalized eigenvalue. Finally, the points are segmented according to the magnitude of normalized eigenvalues.

In PCA, the first normalized eigenvalue δ_1 is the largest and is much greater than the second and third normalized eigenvalues (i.e., δ_2 and δ_3), which indicates that the neighboring sets' normal vectors of the points on the surface are more consistent. Furthermore, δ_2 and δ_3 are approximately equal and much smaller than δ_1. The normal vectors of neighboring points on the edge are mostly toward the normal direction of their adjacent surfaces, and δ_2 is much larger than δ_3. The normal vectors of corner points are not uniform; δ_2 approximately equals δ_3, and they approach δ_1. On the basis of the relationship between the magnitudes of the second and third normalized eigenvalues, we set two thresholds e_0 and e_1 ($e_0 < e_1$) to segment the feature points. The type of point p_i is marked as $T(p_i)$ and is determined based on Equation (4).

$$T(p_i) = \begin{cases} 1, & \delta_2 \leq e_0 \\ 2, & \delta_2 \geq e_1,\ \delta_3 \leq e_0 \\ 3, & \delta_3 \geq e_1 \end{cases} \tag{4}$$

Point p_i is on the surface, edge, or corner when $T(p_i)$ is equal to 1, 2, or 3, respectively.

To obtain a significant contrast, first the nearest neighbor set $Q = \{q_1, q_2, \ldots, q_K\}$ of discrete point p in the point cloud is searched when PCA is used to segment feature points based on point coordinates. This procedure is similar to methods that are based on normal vectors. Second, covariance matrix C is constructed by the point's coordinates and the neighbors' coordinates, as shown in Equation (5):

$$C = \frac{1}{K} \begin{bmatrix} q_1 - \overline{q} & q_2 - \overline{q} & \cdots & q_K - \overline{q} \end{bmatrix} \begin{bmatrix} q_1 - \overline{q} & q_2 - \overline{q} & \cdots & q_K - \overline{q} \end{bmatrix}^T \tag{5}$$

where $\overline{q}$ is the center of gravity of point set Q. Matrix C is analyzed by PCA to obtain normalized eigenvalues δ_{ci} ($i = 1, 2, 3$ and $\lambda_{c1} \geq \lambda_{c2} \geq \lambda_{c3}$). Finally, the points are segmented according to the magnitude of normalized eigenvalues, which is similar to the rule used in feature point segmentation based on normal vectors. Table 1 shows the steps, notation, and explanations for feature point segmentation based on points' coordinates and normal vectors. Figure 4a demonstrates the results of feature point segmentation using PCA based on point coordinates, and Figure 4b displays the results using the normal vector distribution matrix with the same parameters and algorithm. The blue, pink, and yellow points denote the surface, edge, and corner points, respectively. Many edge points in Figure 4a are misclassified as surface points. Several misclassified corner points in Figure 4a are corrected in Figure 4b by using the normal vector distribution matrix. Thus, the feature points in this study are initially segmented using the normal vector distribution matrix. We establish that the surface, edge, and corner points are stored in sets S, E, and C, respectively. Surface points on different surfaces are separated by sets E and C rather than connected, and the separated surface points initially form the patch area.

Table 1. Steps, notation, and explanations for feature point segmentation based on points' coordinates and normal vectors.

Steps	Notation		Explanations
	Segmentation Based on Point Coordinates	**Segmentation Based on Normal Vector**	
Search for K nearest neighbor points	p	p	Discrete point
	K	K	Number of nearest neighbor points
	Q	Q	K nearest neighbor set
	q_i	q_i	ith point of K nearest neighbor set
Fit local plane	-	P	Fitting local plane
	-	d	Distance between point p and fitting plane P
	-	n	Normal vector of the fitting plane
	-	q_i°	ith projection point of a neighbor point on the fitting plane
	-	θ	Weight function
Calculate normal vector of discrete points	-	C	Covariance matrix constructed by point's coordinate
	-	$\overline{q}$	Center of gravity of point set Q
	-	n_i	ith normal vector of discrete points
PCA	C	A	Covariance matrix constructed by point's coordinate or normal vector
	$\overline{q}$	-	Center of gravity of the point set Q
	λ_{ci} $(i = 1, 2, 3)$	λ_i $(i = 1, 2, 3)$	ith eigenvalues
Segment feature points	δ_{ci} $(i = 1, 2, 3)$	δ_i $(i = 1, 2, 3)$	ith normalized eigenvalues
	e_{c0}	e_0	First threshold
	e_{c1}	e_1	Second threshold

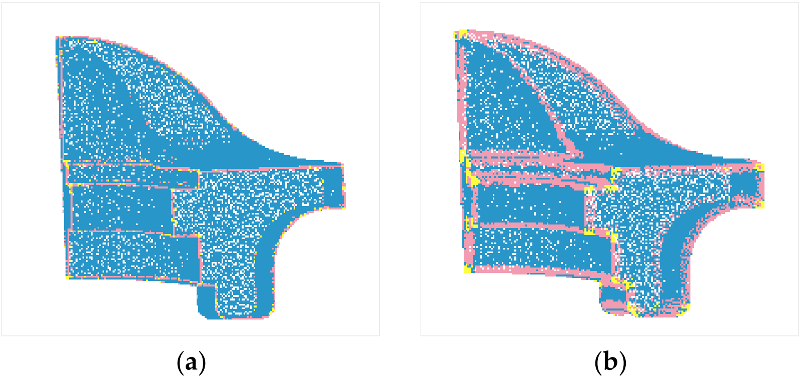


Figure 4. PCA results of point cloud segmentation by using (**a**) point coordinates and (**b**) the normal vector distribution matrix. The blue, pink, and yellow points denote the surface, edge, and corner points, respectively.

2.2. Patch Classification

The distance-constrained region growing algorithm is applied to separate a single patch. This algorithm facilitates the analysis of geometric features and improves the accuracy of surface reconstruction. The flowchart for classifying set S that stores surface points by using the region growing algorithm with a distance constraint is shown in Figure 5.

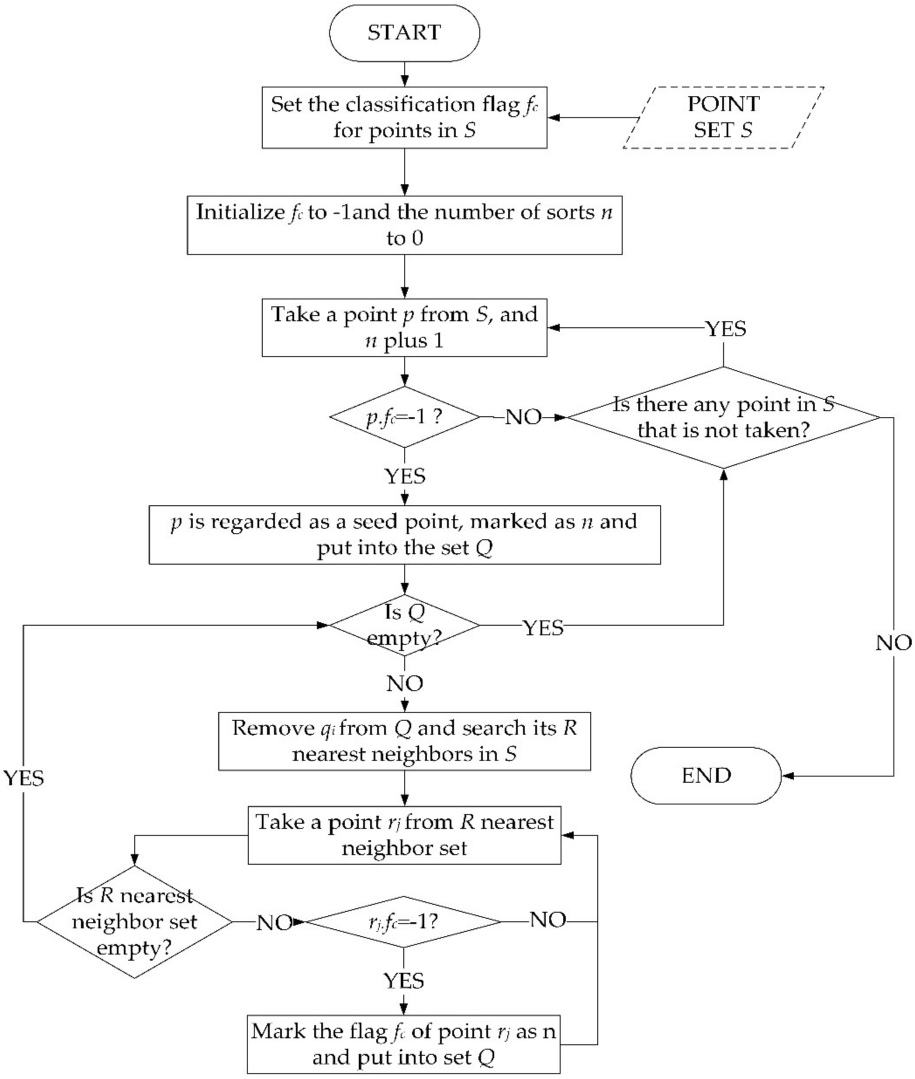

START
Set the classification flag f_c for points in S
POINT SET S
Initialize f_c to -1and the number of sorts n to 0
Take a point p from S, and n plus 1
YES
$p.f_c$=-1 ?
NO
Is there any point in S that is not taken?
YES
p is regarded as a seed point, marked as n and put into the set Q
Is Q empty?
YES
NO
NO
Remove q_i from Q and search its R nearest neighbors in S
YES
END
Take a point r_j from R nearest neighbor set
Is R nearest neighbor set empty?
NO
$r_j.f_c$=-1?
NO
YES
Mark the flag f_c of point r_j as n and put into set Q

Figure 5. Flowchart of patch classification.

Given the unclassified surface point set S from point cloud segmentation, the method for patch classification is implemented as follows:

1. A classification flag f_c is set for each point in surface point set S. The flag is initialized to -1, and the number of sorts, n, is initialized to 0.
2. The algorithm is terminated when all the points in point set S are extracted. Otherwise, n is incremented, and a point p is extracted from point set S to determine whether its classification flag f_c is -1 one by one. f_c is regarded as a seed point, marked as n, and stored in the temporary point set Q when its value is -1. If f_c is not -1, another point is obtained from S.
3. If set Q is empty, then Step 2 is repeated. Otherwise, element q_i is removed from Q, and the R nearest neighbors set of q_i is searched.
4. Flag f_c of point r_j in the R nearest neighbors set of q_i is marked as n and stored in set Q when its value is -1.

The points in set S with an equal flag f_c are on the same patch. The points are stored based on their category, and each element of $S = \{S_1, S_2, \ldots\ S_n\}$ is a category. The original point display is shown in Figure 6a, and the classification display is presented in Figure 6b, where point sets with different colors represent different patches.

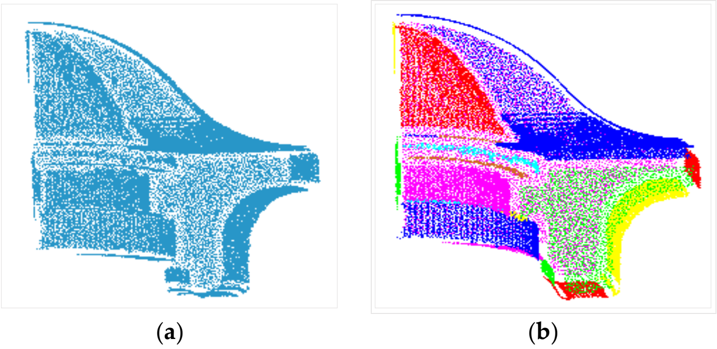


Figure 6. Comparison of results before and after patch classification: (**a**) original points display and (**b**) classification display.

2.3. Surface Implicit Reconstruction

Special quadric surfaces are typical structures in industrial components [31]. Determining if the discrete points are on the surface is easy when the implicit expression is used. Each patch after classification has a single structure, and many inliers exist. The RANSAC algorithm can be used to build the surface implicit equation effectively. In this study, the RANSAC algorithm is applied to three types of regular quadric surface models, namely, plane, cylinder, and sphere, to estimate the model parameters. The coordinates of discrete points can be combined with the normal vector to quickly calculate the parameters of a regular special quadric surface.

2.3.1. 3D Surface Model

A geometric primitive is a curve or surface that can be described by an equation with a number of free parameters [44]. The manual model surface can be described as a set of surface patches, and each surface patch is considered to be a geometric primitive [45]. Therefore, it is key for industrial component feature extraction to extract geometric primitives reliably. Eighty-five percent of mechanical components can be described by plane, spherical, cylindrical, and conical surfaces [31]. In this study, the plane, cylindrical surface, and spherical model are considered because of their relatively simple and common equations.

3D Plane Model

The implicit expression of a plane in 3D space is as follows:

$$Ax + By + Cz + D = 0 \tag{6}$$

where A, B, C, and D are constants that describe the characteristics of the plane space; and x, y, and z are variables that represent the 3D coordinates of a point on the plane.

The normal vector of a plane is perpendicular to the vectors of the plane. Thus, the point normal form equation of a plane with a point (x_0, y_0, z_0) and normal vector (n_1, n_2, n_3) is expressed as follows:

$$n_1(x - x_0) + n_2(y - y_0) + n_3(z - z_0) = 0 \tag{7}$$

where x, y, and z represent the 3D coordinates of a point on the plane.

According to Equations (6) and (7), the normal vector of each point on the plane is equal to the normal vector of the plane. The plane equation can be obtained by using any point on the plane with coordinates and normal vector information.

3D Cylindrical Surface Model

The distance from a point on a cylindrical surface to the central axis of the cylinder is equal to the radius of the cylinder. A cylindrical surface is represented by seven parameters at any position in the 3D space. These seven parameters include center of the cylinder C_p (x_p, y_p, z_p), vector of the central axis C_n (x_n, y_n, z_n), and radius C_R. Given the 3D coordinates and normal vector of any two points in the 3D point cloud as $p_1(x_1, y_1, z_1)$, $p_2(x_2, y_2, z_2)$, $n_1(x_{n1}, y_{n1}, z_{n1})$, and $n_2(x_{n2}, y_{n2}, z_{n2})$, the equations of the seven parameters are as follows:

$$C_n = \frac{(p_1 + s \cdot n_1) - (p_2 + s \cdot n_2)}{\sqrt{\| p_1 + s \cdot n_1 \|^2 + \| p_1 + s \cdot n_1 \|^2}} \tag{8}$$

$$\begin{cases} l = p_1 - C_p \\ \cos\langle C_n, l\rangle = \frac{C_n \cdot l}{\sqrt{\|C_n\|^2 + \|l\|^2}} \\ C_R = \| l \| \sin\langle C_n, l\rangle \end{cases} \tag{9}$$

where s and t are constant parameters, C_n is the central axis direction vector of the cylinder, C_p is the point on the center axis of the cylinder, and C_R is the radius of the cylinder.

3D Spherical Model

All points on the spherical surface have the same distance from the center of the sphere, and the normal vector of these points is toward the center of the sphere. The sphere's center coordinates $S_c(x_c, y_c, z_c)$ and the radius S_R of the sphere can determine the position of the sphere. On the basis of the geometry of the sphere, the spherical model is represented by Equation (10).

$$\begin{cases} \boldsymbol{p}_1 - \boldsymbol{S}_c = S_R \boldsymbol{n}_1 \\ \boldsymbol{p}_2 - \boldsymbol{S}_c = S_R \boldsymbol{n}_2 \end{cases} \tag{10}$$

Center coordinates $\boldsymbol{S}_c$ and radius S_R of the sphere can be obtained by substituting points $\boldsymbol{p}_1(x_1, y_1, z_1)$, $\boldsymbol{p}_2(x_2, y_2, z_2)$, and their normal vectors $\boldsymbol{n}_1(x_{n1}, y_{n1}, z_{n1})$, $\boldsymbol{n}_2(x_{n2}, y_{n2}, z_{n2})$ into Equation (10).

2.3.2. Model Parameter Solution Based on the RANSAC Algorithm

The shape of the isolated patch area cannot be directly represented by a fixed mathematical model due to its uncertainty. A few sample points are randomly selected to calculate the model parameters for the three models stated in Section 2.3.1 and the inliers of the three models using the RANSAC algorithm. The model type and inliers are recorded when the number of inliers is larger than the threshold, and the recording times are marked as the frequency of the model to which the patch belongs. After multiple cycles, the ratio of model frequency to the total number of cycles is recorded as the probability of the model to which the patch belongs. The model with the maximum likelihood is the optimal expression of the surface.

One point is required when using the point coordinates and normal vector to solve a plane, whereas two points are required to solve a cylinder or sphere. We randomly select three points as small samples, and the points can be used for necessary calculation and preliminary verification. Selecting the minimum number of points on the basis of satisfying calculations and verifications can reduce calculation. Patch S_i is defined in surface point set S and loops n. The specific steps of the algorithm are as follows:

1. Three points are randomly extracted from S_i when the number of iterations is less than n, and the cosine values c_1, c_2, and c_3 of the angles between the normal vectors of the three points are calculated, and Step 2 is executed. When the number of cycles is equal to n, Step 4 is executed.
2. The patch may be a plane if the angles among the three normal vectors are roughly equal, that is, the absolute values of c_1, c_2, and c_3 are approximately 1. Four parameters of the plane are

calculated, and the inliers of the parametric model are calculated. The frequency of the plane model is incremented by 1 when the number of inliers is greater than the threshold, and the optimal parameters are updated and Step 1 is executed. Otherwise, Step 3 is executed.

3. The parameters of the cylindrical or sphere model are calculated using two points, and the third point is used for verification. The inliers of the parametric model are calculated when the third point satisfies the calculated model. The frequency of the cylindrical or sphere model is incremented by 1 when the number of inner points is greater than the threshold, and the optimal parameters are updated. Continue to Step 1.
4. Point set S_i does not conform to the above three models when the maximum frequency of the model is less than 3 after n iterations, and the model output is 0. Otherwise, the maximum probability model is calculated, all the model inliers under the corresponding optimal model parameters are optimized using the least squares method, and the obtained model parameters are the final outputs.

2.4. Patch Optimization

2.4.1. Patch Expansion

Considering that edge point and corner point sets contain several surface points during point cloud segmentation, the patch should be expanded to obtain a complete patch. The implicit equation of each patch is easy to obtain from the solved model parameters of the patch. After substituting the point coordinates into the implicit equation of the neighboring surface, a point is deemed to belong to the surface when the point coordinate satisfies the implicit equation. On this basis, the edge and corner points can be classified into the patch they belong to.

Situations wherein patches belong to the same surface in the space may exist, and these patches satisfy the same equation expression, although they are separated from one another in the actual object. A distance limit should be added on the basis of satisfying the implicit equation to prevent misclassification. In this study, the minimum distance from a point in edge or corner point sets to the patch is used as the distance limit. The point is considered as the extension point of the patch when the distance from a point to a patch is smaller than threshold μ, and the coordinate of the point satisfies the implicit equation of the patch.

The results of the original patch before and after expansion are displayed in Figure 7. The regular surface exhibits significant expansion in space, as shown in Figure 7b.

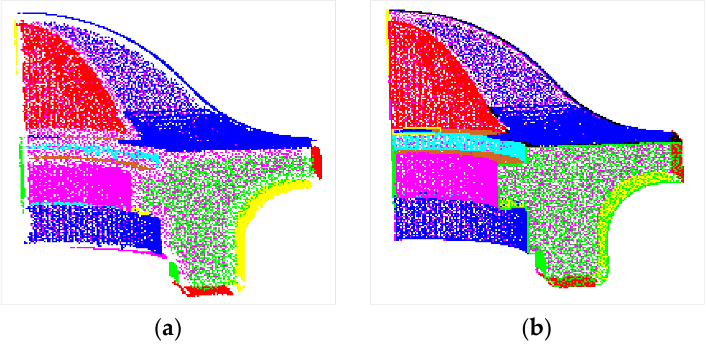

(**a**) (**b**)

Figure 7. Comparison of results before and after patch expansion: (**a**) before patch expansion display and (**b**) after patch expansion display. The patches are marked by different colors.

2.4.2. Patch Merging and Resegmentation

Over-segmentation occurs in patch classification. The two patches selected by the black rectangles in Figure 8 are divided into several surfaces, and the patches that belong to the same surface should be combined. On the basis of the surface implicit equations calculated by the RANSAC algorithm, we can determine if the patches belong to the same spatial surface. When patches belong to the same spatial surface and the distance between the nearest points of the patches is less than the distance constraint, the patches are merged.

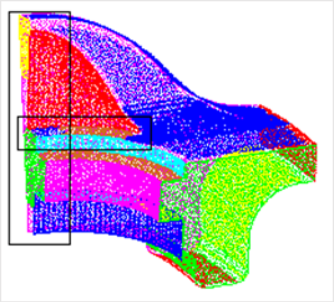


Figure 8. Over-segmentation in patch classification.

Several surfaces that fail to find the model parameters in implicit reconstruction are considered complex or free-form surfaces. A complex surface is composed of multiple regular surfaces. Meanwhile, several points satisfying the regular surface equation are assumed to exist on a free-form surface. The RANSAC algorithm is used to resegment complex or free-form surfaces for further extraction of a regular or local regular surface from them. The points are substituted in the plane, cylindrical surface, and spherical models because the number of point sets and the type of regular surface contained in the complex or free-form surface are not prior knowledge. After each regular quadric surface is determined and added to regional surface set S, the remaining points in S_i continue to be searched until the model can no longer be extracted or only a few remaining points exist. The distances from the unclassified points to the surfaces are calculated when points that are not classified to the surface set still exist. The points are then categorized into the nearest surface based on the distance constraints. In this manner, expansion and optimization of all regional surfaces are completed, and the point cloud model is divided into a number of feature surfaces according to shape characteristics.

2.5. Surface Parameterization

After completing the feature surface extraction of the point cloud model, each surface is represented via parameter expression to facilitate the display and reuse of each platform. This study uses the NURBS module packaged in Point Cloud Library [43] to program NURBS fitting on the Visual Studio platform. Considering the effect and speed of the algorithm, the NURBS polynomial order is set to the experimental value 3, and the number of vertices in each parameter direction is set to the experimental value 100. Then, each feature surface is automatically fitted, and the fitting result based on the fandisk model is shown in Figure 9.

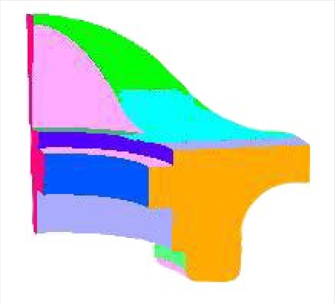


Figure 9. Result of feature surface extraction based on the fandisk model.

3. Experiment and Results

3.1. Testing Data

In our study, three groups of point cloud data [46] (i.e., fandisk, support frame, and connecting piece models) were used to conduct the experiments. The point density of the point clouds is uneven. The point density of edge and corner points is high, whereas it decreases rapidly in the surface areas. The fandisk model is a complex industrial component model [31] that consists of regular and complex surfaces, with an indistinctive difference in the area of surfaces. The fandisk model has various feature surfaces with obvious corner points. The support frame model is the simplest among the three models, and it comprises regular surfaces, with a more significant difference in the area of surfaces than in the fandisk model and obvious corner points. The connecting piece model has moderate complexity in surface structure among the three models, and it consists of regular surfaces and only one complex surface, with a significant difference in the area of surfaces and no obvious corner points. The corresponding data are shown in Table 2, where *Boxx* is the length of the oriented bounding box of the point cloud model, *Boxy* is the width and *Boxz* is the height. The model display is presented in Figure 10.

Table 2. Point cloud data of the three models.

Model	Surfaces	Points	Average Point Spacing (*dm*)	Minimum Point Spacing (*dm*)	Bounding Box Size		
					Boxx (*dm*)	*Boxy* (*dm*)	*Boxz* (*dm*)
Fandisk model	21	55,105	0.0426	0.006	4.8281	5.2445	2.6805
Support frame model	38	200,018	0.0052	0.0002	1.0000	1.0000	0.6000
Connecting piece model	12	1,514,884	0.0024	0.0005	0.9975	0.9264	0.7838

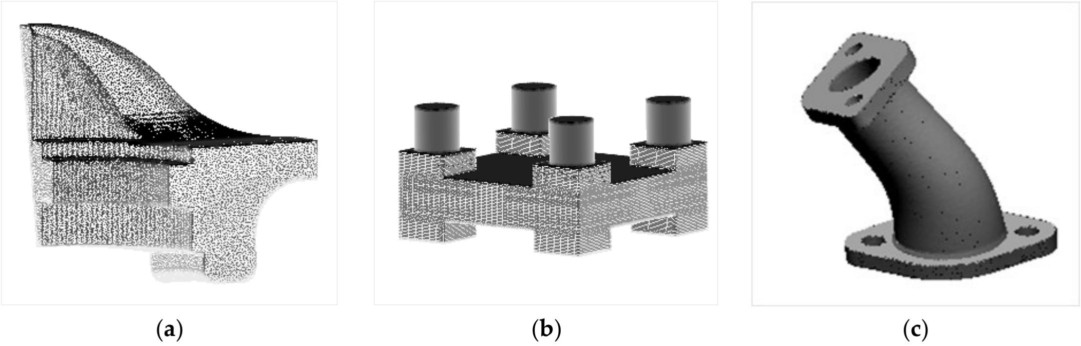

(**a**) (**b**) (**c**)

Figure 10. Model display: (**a**) fandisk model; (**b**) support frame model; (**c**) connecting piece model.

3.2. Parameter Tuning

In feature point segmentation, the main parameters are the K value of the K nearest neighbors set and thresholds e_0 and e_1. In patch classification and surface reconstruction, the main parameters are the nearest neighbor search radius R when classifying a patch, cycles n_1 in the RANSAC algorithm when calculating the implicit expression, cycles n_2 and the minimum inner points κ in the RANSAC algorithm when resegmenting to determine rule surfaces, and distance threshold μ between a point and the nearest point on the surface when expanding a patch. Threshold ε denotes the error range between data and theoretical values, and it is applied in many algorithms. In this study, surface complexity is classified into three grades, the simplest of which is the regular surface. The second grade is the complex surface, which is composed of multiple regular surfaces. The most complex surface is the free-form surface, which is the surface that cannot be a combination of any regular surface.

K, e_0, and e_1 directly affect feature point classification. K affects the range of feature point segmentation, and a K value that ranges from 15 to 20 can be adjusted to become large when the number of model points is large, the point space is small, and the complexity of the surface is low. The values of e_0 and e_1 are set by the sampling method. First, we manually select the sample points, which are surface, edge, and corner points, each with 50 points. Second, we calculate the normalized eigenvalues δ_1, δ_2, and δ_3 of all sample points by using PCA based on normal vectors. Finally, we set two thresholds e_0 and e_1 according to the relationship between the magnitudes of the second and third normalized eigenvalues. The two thresholds satisfy the following conditions: e_0 is larger than the second normalized eigenvalues of the surface point and the third normalized eigenvalues of the edge point, and e_1 is smaller than the second normalized eigenvalues of the edge point and the third normalized eigenvalues of the corner point. For an easy comparison of the probabilities of model types, n_1 is set as an odd number and changes with the complexity of the surface of the industrial component. The larger the values of n_1 and n_2 are, the more accurate the result is, and the higher the computational efficiency is. Usually, n_1 ranges from 50 to 150, and n_2 ranges from 100 to 200. n_1 and n_2 are set to a large value when complex or free-form surfaces exist. In the fandisk model, n_1 and n_2 are set with larger values compared with those in the other models because complex surfaces exist in the fandisk model. The value of κ influences the optimization of the feature surface, and κ usually ranges from 500 to 600. A large number of surfaces that require resegmentation are obtained when the value of κ is small. The feature surface of the connecting piece model is simple, and the feature surface obtained in patch classification is accurate. Thus, κ is set to a large number (i.e., 1000). The value of μ as the distance limit threshold of patch extension is related to the width of the K nearest neighbors set in the feature point extraction process. Here, μ is set to half of the width (K multiplied by average point spacing). No resegmentation occurs in the support frame model. Thus, n_2, κ, and μ have no effect on their extraction results. R is generally set to a value that is 1 to 10 times larger than average point spacing and ranges with the complexity of the surface. The lower the grade of complexity is, the smaller the value of R is. The value of ε is related to average point spacing in the point cloud model, and it is set to a value that is close to average point spacing when the point spacing is less than 0.01. Otherwise, the value of ε is approximately 5 to 10 times less than that of average point spacing. The parameter settings in the experiment are shown in Table 3.

Table 3. Point cloud data of the three models.

Parameter	Fandisk Model	Support Frame Model	Connecting Piece Model
K (*ea*)	15	16	20
e_0	0.0045	0.005	0.035
e_1	0.006	0.01	0.1
n_1 (*time*)	125	25	55
n_2 (*time*)	200	-	150
κ (*ea*)	600	-	1000
μ (*dm*)	0.35	-	0.025
R (*dm*)	0.07	0.02	0.02
ε (*dm*)	0.006	0.005	0.002

3.3. Results

The result of feature point segmentation for the fandisk model is shown in Figure 11a, the feature surface extraction is shown in Figure 11b, and the feature surface reconstruction is shown in Figure 11c. The experimental results from the support frame and connecting piece models are shown in Figures 12 and 13, respectively.

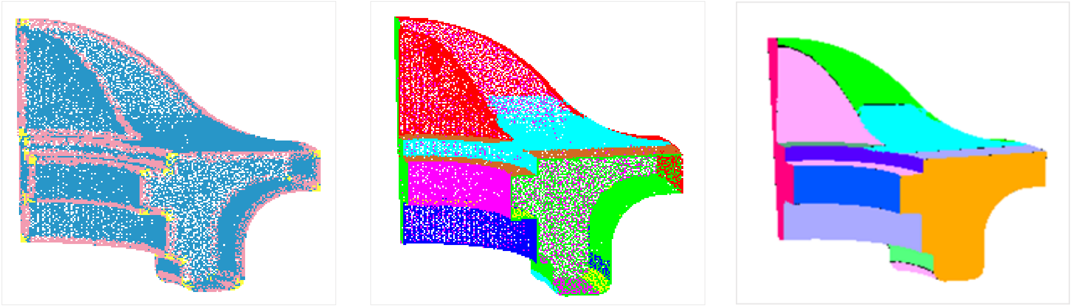


Figure 11. Experimental results of the fandisk model: (**a**) feature point segmentation; (**b**) feature surface extraction; (**c**) feature surface reconstruction.

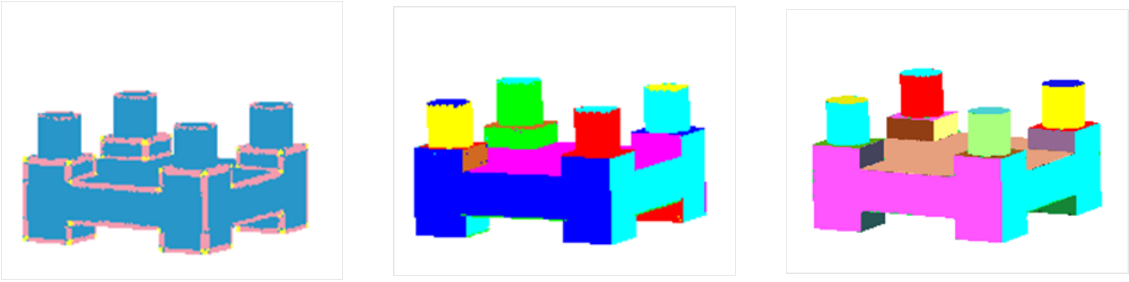


Figure 12. Experimental results of the support frame model: (**a**) feature point segmentation; (**b**) feature surface extraction; (**c**) feature surface reconstruction.

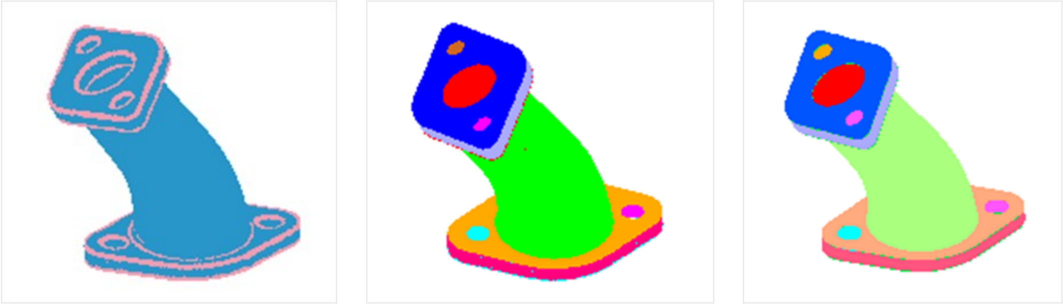


Figure 13. Experimental results of the connecting piece model: (**a**) feature point segmentation; (**b**) feature surface extraction; (**c**) feature surface reconstruction.

To evaluate the performance of the proposed MSSO, we refer to [47,48] and quantitatively evaluate the results of feature surface segmentation at the point level by using four measures: *Corr* (correctness rate of feature surface extraction), *Comp* (completeness rate of feature surface extraction), *Qual* (quality of feature surface extraction), and *T* (execution time of the extraction). The first three measures pertain to accuracy, and the last pertains to time. The accuracy measures are defined as follows:

$$Corr = \frac{N_{tp}}{N_{tp} + N_{mp}} \tag{12}$$

$$Comp = \frac{N_{tp}}{N_{tp} + N_{op}} \tag{13}$$

$$Qual = \frac{N_{tp}}{N_{tp} + N_{mp} + N_{op}} \tag{14}$$

where N_{tp} is the number of true positive feature points contained in the detected surfaces, N_{mp} is the number of misclassification points contained in the detected surfaces, and N_{op} is the number of omitted points. The sum of the three types of points is the total number of points N_{ap} in the model. Larger values of *Comp*, *Corr*, and *Qual* and a small value of *T* are desirable.

The fandisk, support frame, and connecting piece models are evaluated. To compute the four measures for evaluating the surface segmentation results, we count the number of the three types of points to which the detected points belong in each model and the execution time. The results of the location evaluation (correctness, completeness, and quality) and execution time of extraction are shown in Tables 4 and 5.

Table 4. Defining assessment quantities.

Model	N_{ap}	N_{tp}	N_{mp}	N_{op}
Fandisk model	55,105	54,688	222	195
Support frame model	200,018	199,693	218	107
Connecting piece model	151,484	149,943	220	1321

Table 5. Result of evaluation and execution time of feature surface extraction.

Model	*Corr* (%)	*Comp* (%)	*Qual* (%)	*T* (s)
Fandisk model	99.60	99.64	99.24	54.36
Support frame model	99.89	99.95	99.84	22.55
Connecting piece model	99.85	99.13	98.98	43.29

As shown in Table 5, the correctness, completeness, and quality of feature surface extraction in the fandisk and support frame models exceed 99%, and the models' completeness is better than their correctness. The correctness and completeness in the connecting piece model exceed 99%, but the quality is low (98.98%) and completeness is worse than correctness. Feature extraction in the three models can be completed within 55 s. The fandisk model takes the longest time, which is 54.36 s, to extract the feature surface.

4. Discussion

4.1. Comparative Studies

The proposed method is compared with other software [49] and algorithms [9,50,51] in terms of three aspects, namely, operational complexity, execution time, and accuracy.

The operational complexity of the proposed method is compared with that of a commercial software called Geomagic Studio [49], which is a reverse engineering application developed by Raindrop. Surface construction with Geomagic Studio involves three phases, namely, point cloud, polygons, and patches. In the point cloud phase, users are required to erase outliers manually or automatically, reduce noise by manually setting different smoothness levels, and down-sample. In the polygon phase, users set parameters to wrap the object with a polygon mesh, fill holes, clean up redundancy, adjust the quantity of polygons, and smooth and correct the polygon mesh. In the last phase, surfaces are automatically reconstructed with one parameter for a sample model. However, for a complex model, users are required to perform four steps, namely, constructing contours, constructing patches, constructing grids, and fitting surfaces. Each step of Geomagic Studio in constructing the abovementioned surfaces requires the user to manually tune parameters (approximately a dozen). However, with our method, surfaces can be automatically constructed with nine parameters entered before the execution of the program, instead of manual processing. Thus, the proposed method facilitates automation of feature surface extraction.

Zhou [50] proposed a new approach to reconstruct and optimize feature surfaces at large scales with the division and joining of surfaces. Chen et al. [51] proposed a method to compute a triangular mesh and complex surfaces with restricted Delaunay triangulations from an unorganized point cloud. They both extracted feature surfaces from discrete point cloud data, and verified the execution time of the algorithm based on a fandisk model. We compare the two methods with the proposed method in terms of execution time, and the comparative results are shown in Table 6.

Table 6. Comparison of execution time results based on the fandisk model.

Method	Points	Execution Time (s)	*Qual* (%)
Zhou [50]	23,400	79	99.04
Cheng et al. [51]	66,100	140.88	-
Our method	55,105	54.36	99.24

The hardware parameters, such as CPU and RAM, of this method are similar to those in [50], and the number of points in the fandisk model in [50] is less than half of that in this study. However, compared with the method in the literature, our method requires less time to extract surfaces, as shown in Table 6. The method of [51] takes 140.88 s to construct feature surfaces, which is approximately three times longer than that of the proposed method and exceeds the impact of the hardware and number of points. Therefore, the proposed method performs well in terms of efficiency.

In terms of accuracy, the proposed method is compared with the method of Cao et al. [9], who developed a novel surface fitting scheme to automatically reconstruct a genus-0 object into a continuous parametric surface. The method is experimentally validated on the fandisk model. The fitting error is related to the number of control points and number of iterations. The initial fitting iteration is obtained through experiments. The accuracy results are shown in Table 7. The proposed method does not require control points, and the accuracy of the proposed method is higher than that of Cao et al.'s method with control points (the number is 582).

Table 7. Comparison of accuracy results based on the fandisk model.

Method	Points	Control Points	*Qual* (%)
Cao et al. [9]	56,000	582 4994	98.84 99.97
Our method	55,105	0	99.24

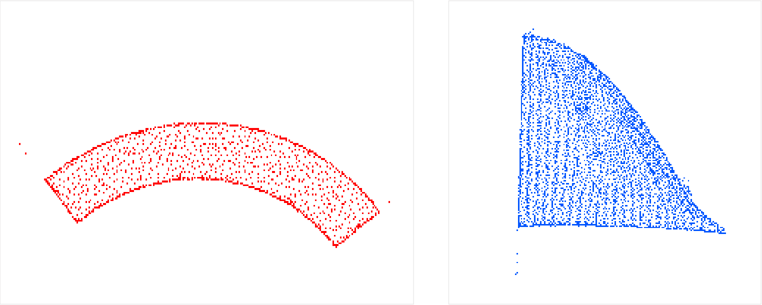
4.2. Result Analysis

The results of the location evaluation (correctness, completeness, and quality) and execution time of feature surface extraction are shown in Table 5. The correctness rate of the fandisk model is the lowest among the three models, because it has the most complex surface structure and the lowest point density. The completeness rate of the connecting piece model is the lowest, and its completeness rate is much lower than its correctness rate compared with the difference in the other models. A large and irregular surface is observed in the connecting piece model, resulting in omission. Moreover, all of the location evaluations of the support frame model are the highest among the three models because the surface structure of the support frame model is regular, and its area is apposite. Furthermore, all of the surfaces can be represented by an expression in surface implicit reconstruction. The details of the error sources are analyzed in Section 4.3. The execution time of the fandisk model is the longest, and that of the support frame model is the shortest, although the fandisk model has the least number of discrete points. According to the analysis in Section 3.2, the larger the cycles n_1 and n_2 are, the lower the computational efficiency. That is, the more complex the surface structure is, the longer the execution time. Thus, the complexity of the surface exerts a greater impact on operating efficiency than does the point density of point cloud in the 3D model. In our method, surfaces with low complexity are expected, that is, regular and complex surfaces containing plane, cylindrical, and spherical surfaces only rather than free-form surfaces will achieve greater results. Meanwhile, a high point density of the point cloud is expected when there are free-form surfaces.

Using PCA in the normal vectors to define the point features for classification is accurate, as proven in Section 2.1. Thus, normal vectors are introduced in this method. However, this process requires considerable computing resources to segment the point cloud using normal vectors to define point features rather than point coordinates. However, we determine that computing resources will not be consumed in large quantities due to the small number of points in the industrial component point cloud compared with buildings and complex man-made objects [13]. Nevertheless, the issue that the proposed method requires a large amount of computing resources should be taken seriously.

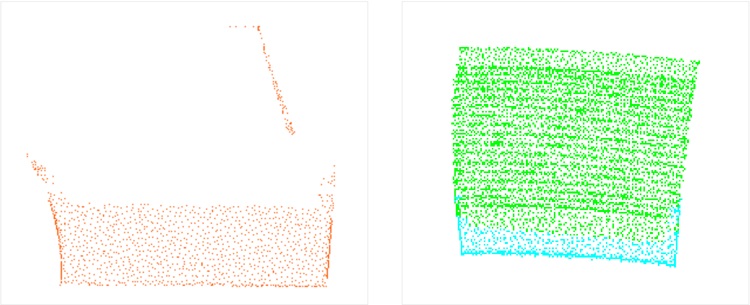
4.3. Error Analysis

The experimental results show that the fandisk model contains two free-form surfaces. Hence, two cases of misclassification occur. In the first case, the points outside the area are also classified in the patch when the regular surface is extended and the threshold of the distance constraint is large (Figure 14). In the second case, the patches extracted from the complex surface are disconnected from one another because the connection of extracted patches is not considered when extracting the inliers. As shown in Figure 15a, two plane point sets that are not connected to each other are obtained by subdividing the free-form surface. Figure 15b illustrates the misclassification in which patch overlap occurs between the blue plane and the green cylindrical surface.



(**a**) (**b**)

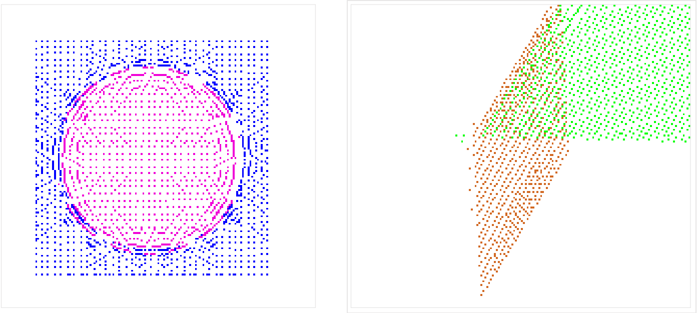
Figure 14. Outer points are extended to regular surfaces in the fandisk model: (**a**) plane; (**b**) cylindrical surface.



(**a**) (**b**)

Figure 15. Patches extracted from the free surface are disconnected in the fandisk model: (**a**) two unconnected plane point sets; (**b**) patch overlap between the plane and cylindrical surface.

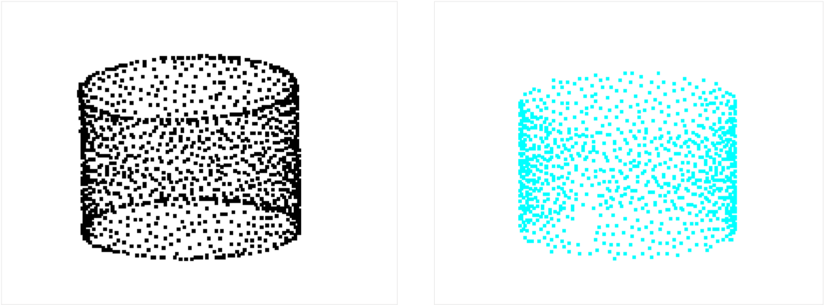
The support frame model only has two types of plane and cylindrical surfaces without a complex surface. Therefore, this model exerts a better extraction effect than the fandisk model. Although the distances for the inliers are relatively strict in regular area extension, several misclassification points are still observed at the junction between the patches. Figure 16 illustrates the misclassification for the two situations. Such a misclassification point occurs because the distance from the point at the boundary to the nearest point of two adjacent surfaces simultaneously satisfies the threshold, and the points that have been marked are no longer considered a close patch.



(**a**) (**b**)

Figure 16. Display of boundary misclassification points in the support frame model: (**a**) the junction of the plane and cylindrical surface; (**b**) the junction of two planes.

The feature surface structure is relatively simple in the connecting piece model. Hence, no obvious misclassification occurs. However, several missing points are observed in the feature surface extraction results, especially in regular surfaces such as cylinders and planes, as presented in Figure 17. Several inliers are regarded as outliers and removed when RANSAC is executed. Omission points can cause incomplete model boundaries without affecting surface reconstruction because the small holes will be filled after NURBS reconstruction.



(**a**) (**b**)

Figure 17. Display of boundary misclassification points in the connecting piece model: (**a**) cylindrical surface in the original model; (**b**) extracted cylindrical surface.

In the three groups of models, misclassification of the boundary points causes the breakage of the boundary junctions. In this case, the boundary of feature surfaces in software such as Geomagic and ProE should be manually repaired. Overall, the algorithm has high automaticity for extracting feature planes and shows a good extraction effect for regular patches, that is, the quadric surfaces of the plane, cylinder, and sphere that are applied in this study. The boundary of the feature surface is incomplete due to misclassification, which can be solved by manual repair.

5. Conclusions

To improve the efficiency of model reconstruction in reverse engineering, this study developed a new feature surface extraction method for 3D point cloud models. First, PCA based on a normal vector distribution matrix was used, and two thresholds were set to extract feature edge, surface, and corner points. Compared with the PCA method based on point coordinates, the proposed method extracted feature points completely. The RANSAC algorithm was applied to establish the implicit equation and extend the incomplete feature surface. NURBS fitting was then performed to reconstruct each complete feature surface. Two forms of implicit and parameter expressions were used for surface reconstruction to improve the accuracy and practicability of the developed model. The quality of feature surface extraction exceeded 99% in the fandisk and support frame models, and the quality in the connecting piece model also reached 98.98%. Feature surface extraction with the three models was completed within 55 s, and the support frame model took 22.55 s to extract the feature surface. This study provided a method for setting the threshold parameter in the algorithm, which can improve automation of feature surface extraction.

This algorithm can obtain relatively complete feature surface vector information, and it is more suitable for regular and complex surfaces containing plane, cylindrical, and spherical surfaces only rather than free-form surfaces. In our future studies, we plan to expand the types of fitting surfaces, such as circular conical and paraboloid, in implicit expressions to better describe the free-form surfaces of industrial components. Although a threshold setting method was proposed in this study, setting appropriate parameters is still difficult for non-professionals because of the lack of adaptive setting of the parameter threshold. Subsequent studies may consider the adaptive adjustment of model parameters. In addition, we will further evaluate different industrial component models and extend this approach to extract features from the point clouds of other objects. Therefore, the proposed method is expected to be improved to automatically analyze the characteristics of various point cloud models.

Author Contributions: Y.W. and J.W. developed the main method to extract feature surfaces from industrial components. X.C. supervised the work and led the work to revise the algorithm. M.L. conducted the experiment on the 3D point cloud data. T.C. and T.Y. revised the paper.

Funding: This work was supported by the Navigation and Location-based service (NAL) Lab, Peking University, and funded by The National Key Research and Development Program of China (No. SQ2017YFC150110).

Acknowledgments: The authors would like to thank the reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, Y.; Wang, C.; Cheng, J. Line segment extraction for large scale unorganized point clouds. *ISPRS J. Photogramm. Remote Sens.* **2015**, *102*, 172–183. [CrossRef]
2. Li, L.; Li, C.; Tang, Y.; Du, Y. An Integrated Approach of Reverse Engineering Aided Remanufacturing Process for Worn Components. *Robot. Comput. Robot. Integr. Manuf.* **2017**, *48*, 39–50. [CrossRef]
3. Patel, M.; Desai, H. A Qualitative Survey on 3D Data Acquisition Techniques. *Int. J. Sci. Res. Dev.* **2015**, *3*, 781–785.
4. Zheng, S.; Wang, X.; Ma, D. A Convenient 3D Reconstruction Method of Small Objects. *Geomat. Inf. Sci. Wuhan Univ.* **2015**, *40*, 147–152.
5. Ni, H.; Lin, X.; Ning, X.; Zhang, J. Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. *Remote Sens.* **2016**, *8*, 710. [CrossRef]
6. Zhang, Y. Research on Automatic Pipe Network Modeling Based on 3D Point Cloud Data. Master's Thesis, Beijing University of Civil Engineering and Architecture, Beijing, China, 2016.
7. Várady, T.; Facello, M.; Terék, Z. Automatic extraction of surface structures in digital shape reconstruction. *Comput. Aided Des.* **2007**, *39*, 379–388. [CrossRef]
8. Li, J.; Wang, R. Robust Denoising of Point-sampled Surfaces. *WSEAS Trans. Comput.* **2009**, *8*, 153–162.
9. Cao, J.; Li, X.; Chen, Z.; Qin, H. Spherical DCB-Spline Surfaces with Hierarchical and Adaptive Knot Insertion. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1290–1303. [PubMed]

10. Hu, J.; Wang, X.; Qin, H. Improved, Feature-centric EMD for 3D surface modeling and processing. *Graph. Models* **2014**, *76*, 340–354. [CrossRef]
11. Merigot, Q.; Ovsjanikov, M.; Guibas, L. Voronoi-Based Curvature and Feature Estimation from Point Clouds. *IEEE Trans. Comput.* **2010**, *17*, 743–756. [CrossRef] [PubMed]
12. Altantsetseg, E.; Muraki, Y.; Matsuyama, K.; Konno, K. Feature line extraction from unorganized noisy point clouds using truncated Fourier series. *Vis. Comput.* **2013**, *29*, 617–626. [CrossRef]
13. Demarsin, K.; Vanderstraeten, D.; Roose, D. Meshless Extraction of Closed Feature Lines Using Histogram Thresholding. *Comput.-Aided Des. Appl.* **2013**, *5*, 589–600. [CrossRef]
14. An, Y. Geometric Properties Estimation and Feature Identification from 3D Point Cloud. Ph.D. Thesis, Dalian University of Technology, Dalian, China, 2011.
15. Zou, D. The Feature Extraction and Segmentation Analysis of Point Clouds. Ph.D. Thesis, Nanjing Normal University, Nanjing, China, 2012.
16. Zhang, Y.; Geng, G.; Wei, X.; Su, H.; Zhou, M. Point Clouds Simplification with Geometric Feature Reservation. *J. Comput.-Aided Des. Comput. Graph.* **2016**, *28*, 1420–1427.
17. Angelo, L.; Stefano, P. Geometric segmentation of 3D scanned surfaces. *Comput. Aided Des.* **2015**, *62*, 44–56. [CrossRef]
18. Liu, Z.; Jiang, K.; Lin, J. Interactive extraction of boundary of specified target feature on scattered point cloud. *Comput. Eng. Appl.* **2016**, *52*, 186–190.
19. Qin, J.; Wan, Y.; He, P.; Chen, M.; Lu, W.; Wang, S. An Automatic Building Boundary Extraction Method of TLS Data. *Remote Sens. Inf.* **2015**, *30*, 3–7.
20. Berger, M.; Tagliasacchi, A.; Seversky, L.; Alliez, P.; Guennebaud, G.; Levine, J.; Sharf, A.; Silva, G. A Survey of Surface Reconstruction from Point Clouds. *Comput. Graph. Forum* **2016**, *36*, 301–329. [CrossRef]
21. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the EG/SIGGRAPH Symposium on Geometry Processing, Boston, MA, USA, 13–17 August 2006; pp. 61–70.
22. Liu, S.; Wang, C. Orienting Unorganized Points for Surface Reconstruction. *Comput. Graph.* **2010**, *34*, 209–218. [CrossRef]
23. Kai, W.; Bo, P. Feature curve extraction from point clouds via developable strip intersection. *J. Comput. Des. Eng.* **2015**, *3*, 102–111.
24. Lu, X.; Yao, J.; Tu, J.; Li, K.; Li, L.; Liu, Y. Pairwise Linkage for Point Cloud Segmentation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 201–208. [CrossRef]
25. Mizoguchi, T.; Date, H.; Kanai, S.; Kishinami, T. Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing. In Proceedings of the 4th International Conference on Geometric Modeling and Processing, Pittsburgh, PA, USA, 26–28 July 2006; pp. 644–654.
26. Vo, A.; Truong-Hong, L.; Laefer, D.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [CrossRef]
27. Wang, X.; Zou, L.; Shen, X.; Ren, Y.; Qin, Y. A Region-growing Approach for Automatic Outcrop Fracture Extraction from A Three-dimensional Point Cloud. *Comput. Geosci.* **2017**, *99*, 100–106. [CrossRef]
28. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for Point-cloud Shape Detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [CrossRef]
29. Arikan, M.; Wimmer, M.; Maierhofer, S. O-snap: Optimization-based Snapping for Modeling Architecture. *ACM Trans. Graph.* **2013**, *32*, 1–15. [CrossRef]
30. Zeineldin, R.; El-Fishawy, N. A Survey of RANSAC Enhancements for Plane Detection in 3D Point Clouds. *Menoufia J. Electron. Eng.* **2017**, *26*, 519–537.
31. Wu, S. Feature Extraction and Shape Detection in Reverse Engineering. Master's Thesis, Zhejiang University, Hangzhou, China, 2011.
32. Fuhrmann, S.; Goesele, M. Fusion of Depth Maps with Multiple Scales. In Proceedings of the ACM SIGGRAPH Asia Conference, HongKong, China, 12–15 December 2011.
33. Tagliasacchi, A.; Olson, M.; Zhang, H.; Hamarneh, G.; Cohen-Or, D. VASE: Volume-Aware Surface Evolution for Surface Reconstruction from Incomplete Point Clouds. *Comput. Graph. Forum* **2011**, *30*, 1563–1571. [CrossRef]
34. Wu, G.; Zhang, Y. A Novel Fractional Implicit Polynomial Approach for Stable Representation of Complex Shapes. *J. Math. Imaging Vis.* **2017**, *55*, 89–104. [CrossRef]

35. Rouhani, M.; Sappa, A.; Boyer, E. Implicit B-spline Surface Reconstruction. *IEEE Trans. Image Process.* **2015**, *24*, 22–32. [CrossRef] [PubMed]
36. Xu, L.; Wu, G. Review of Implicit Surface Reconstruction from Point Cloud Dataset. *Comput. Sci.* **2017**, *44*, 19–23.
37. Bohm, W.; Farin, G.; Kahmann, J. A Survey of curve and surface methods in CAGD. *Comput. Aided Geom. Des.* **1984**, *1*, 1–60. [CrossRef]
38. Barazzetti, L.; Banfi, F.; Brumana, R.; Previtali, M. Creation of Parametric BIM Objects from Point Clouds Using Nurbs. *Photogramm. Rec.* **2016**, *30*, 339–362. [CrossRef]
39. Fu, H.; Liu, X.; Li, J. NURBS Surface Reconstruction Based on Terrestrial LiDAR Point Cloud. *J. Geomat.* **2017**, *42*, 17–19.
40. Linsen, L.; Prautzsch, H. Local versus global triangulations. In Proceedings of the Eurographics, Manchester, UK, 5–7 September 2001; pp. 257–263.
41. Pauly, M.; Keiser, R.; Kobbelt, L.; Gross, M. Shape modeling with point-sampled geometry. *ACM Trans. Graph.* **2003**, *22*, 641–650. [CrossRef]
42. Hoppe, H.; Derose, T.; Duchamp, T.; Mcdonald, J.; Stuetzle, W. Surface Reconstruction from Unorganized Points. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 26–31 July 1992; pp. 71–78.
43. PCL-The Point Cloud Library. 2012. Available online: http://pointclouds.org/ (accessed on 12 April 2018).
44. Roth, G.; Levine, M. Geometric Primitive Extraction Using a Genetic Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 901–905. [CrossRef]
45. Li, Z.; Li, D.; Li, Z.; Wang, Z. New method on extracting geometric primitives. *Infrared Laser Eng.* **2001**, *30*, 4–7.
46. Weber, C.; Hahmann, S.; Hagen, H.; Bonneau, G. Sharp Feature Preserving MLS Surface Reconstruction based on Local Feature Line Approximations. *Graph. Models* **2012**, *74*, 335–345. [CrossRef]
47. Zeng, C.; Wang, J.; Lehrbass, B. An Evaluation System for Building Footprint Extraction from Remotely Sensed Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 1640–1652. [CrossRef]
48. Truong-Hong, L.; Laefer, D. Quantitative evaluation strategies for urban 3D model generation from remote sensing data. *Comput. Graph.* **2015**, *49*, 82–91. [CrossRef]
49. Fu, W.; Zhang, H. Reverse Engineering Design Based on Geomagic Studio Software. *Tool Eng.* **2007**, *41*, 54–57.
50. Zhou, X. Piecewise Method for Large Scale Reconstruction and Optimization. Master's Thesis, Zhejiang University, Hangzhou, China, 2017.
51. Cheng, S.; Wang, Y. Reconstructing Piecewise Curved Complexes. *Surf. Reconstr.* **2008**. Available online: http://hdl.handle.net/1783.1/3090 (accessed on 8 May 2018).