sensors

MDPI

Article

Surface Fitting for Quasi Scattered Data from Coordinate Measuring Systems

Qing Mao [1], Shugui Liu [1,*], Sen Wang [1] and Xinhui Ma [2]

[1] State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China; maoqing.love@163.com (Q.M.); wang.sen.happy@163.com (S.W.)
[2] School of Engineering and Computer Science, University of Hull, Hull HU6 7RX, UK; xinhui.ma@hull.ac.uk
* Correspondence: sgliu@tju.edu.cn; Tel.: +86-136-1210-3881

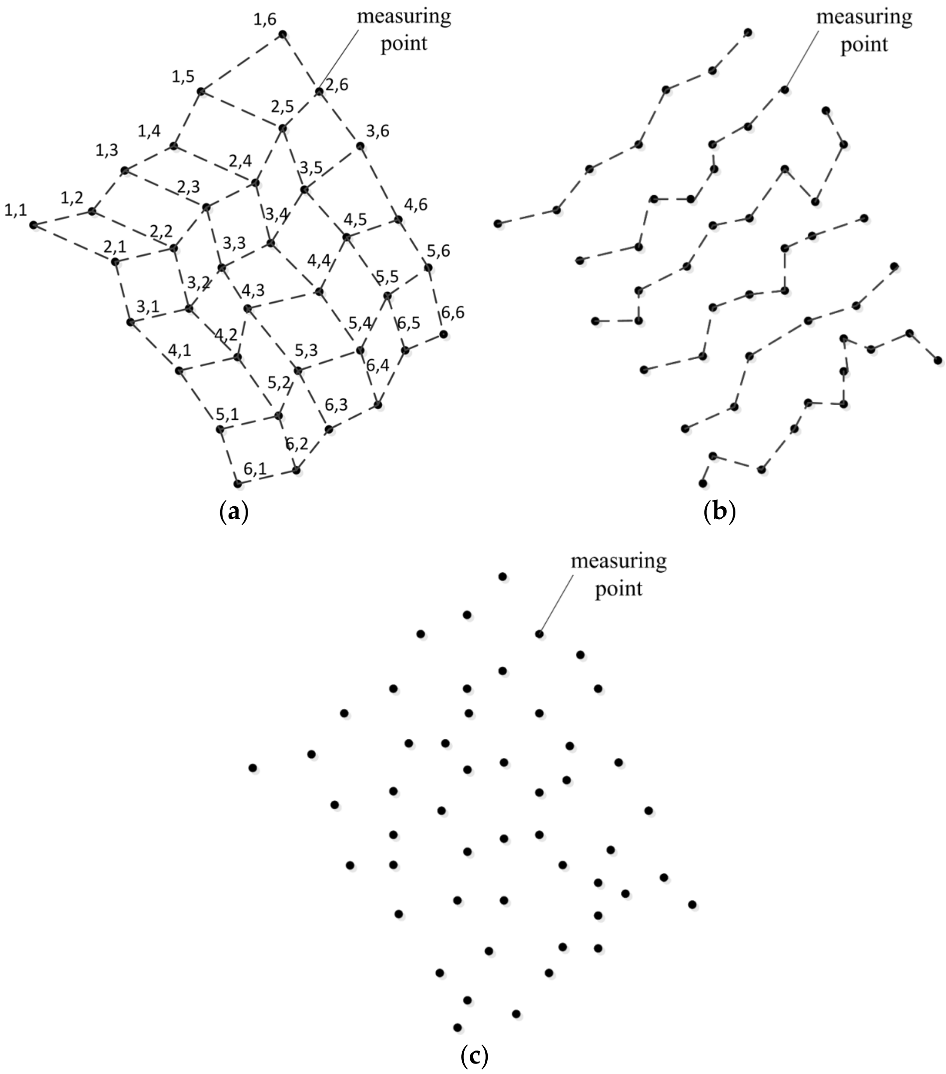
Received: 21 October 2017; Accepted: 10 January 2018; Published: 13 January 2018

Abstract: Non-uniform rational B-spline (NURBS) surface fitting from data points is wildly used in the fields of computer aided design (CAD), medical imaging, cultural relic representation and object-shape detection. Usually, the measured data acquired from coordinate measuring systems is neither gridded nor completely scattered. The distribution of this kind of data is scattered in physical space, but the data points are stored in a way consistent with the order of measurement, so it is named quasi scattered data in this paper. Therefore they can be organized into rows easily but the number of points in each row is random. In order to overcome the difficulty of surface fitting from this kind of data, a new method based on resampling is proposed. It consists of three major steps: (1) NURBS curve fitting for each row, (2) resampling on the fitted curve and (3) surface fitting from the resampled data. Iterative projection optimization scheme is applied in the first and third step to yield advisable parameterization and reduce the time cost of projection. A resampling approach based on parameters, local peaks and contour curvature is proposed to overcome the problems of nodes redundancy and high time consumption in the fitting of this kind of scattered data. Numerical experiments are conducted with both simulation and practical data, and the results show that the proposed method is fast, effective and robust. What's more, by analyzing the fitting results acquired form data with different degrees of scatterness it can be demonstrated that the error introduced by resampling is negligible and therefore it is feasible.

Keywords: scattered data; NURBS fitting; resampling; iterative projection optimization; hierarchical fitting

1. Introduction

The problem of surface fitting appears repetitively in computer aided design, cultural relic representation, reverse engineering, object shape detection and many other fields during the last 20 years or so. Its essence is to build a mathematical model that approximates the object as accurately as possible from measured information. At present, 3D coordinate measuring systems are the main sources of the measured data in these fields [1,2], so finish surface fitting from data acquired from 3D coordinate measuring systems is significant. For now, among all these measuring systems, 3D coordinate measuring machines (CMMs) are representative and mature products which have gained widespread acceptance for their advantages of high accuracy [3]. Besides traditional CMMs, portable 3D vision coordinate measurement machines (PCMMs) such as laser or white light scanners [4,5] and portable light pen 3D vision coordinate measuring systems [6,7], were developed in recent years in order to meet the needs of large-scale on the spot engineering metrology. Generally, the distribution of the measured data acquired from traditional CMMs and the most recently developed PCMMs are neither gridded nor completely scattered [8]. Typically, the data points are scanning points successively sampled from iso-parametric or section curves on a surface, which makes them easily organized into rows [1,2]. In order to describe it clearly, we name this kind of data quasi-scattered data to distinguish it from grid data and completely scattered data in [8]. Figure 1 is a simple sketch and it shows the difference between these three kinds of data. Grid data can easily be organized into rows and columns, and the number of points in each row and column is the same. Quasi-scattered data is scattered in space, but the points are stored according to the order of measurement, so they can easily be organized into rows, but the number of points in each row is different. Completely scattered data has no specific storage order and the points are randomly distributed. In this paper, we only assume that the initial input information is a (possibly massive) set of quasi-scattered 3D points, a typical output obtained from most of the above devices.



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

(**c**)

Figure 1. Three different kinds of distribution of data points: (**a**) grid data; (**b**) quasi-scattered data; (**c**) completely scattered data (quasi-scattered data is stored according to the order of measurement which is the main difference from completely scattered data).

According to the final representation of the fitted surface, the existing related researches can be divided into three categories: fitting methods based on polygonal mesh model [9–13]; fitting methods based on constructive solid geometry (CSG) model [14]; fitting methods based on free-form parametric surface models [15–22]. Non-uniform rational B-spline (NURBS) surface, inheriting all the advantages of B-spline surface, is the most common free-form parametric surface. Most importantly, with the advantages of flexibility and versatility, it provides a good choice for a wide variety of compact and smooth shapes [23–27], so NURBS surface is adopted in this paper.

The problem of NURBS surface fitting from 3D points has been analyzed from several points of view, and for grid data, there are many efficient methods, such as the works related in [19,21,24,25,28,29]. But for scattered data (both quasi-scattered data and completely scattered data), the fitting problem becomes much more difficult because of the limitation that the control points in NURBS surface modeling technique should be organized as a regular grid structure [30]. In [18], a method based on a hierarchical fitting idea is proposed, however, the merged knot vector may make the method face the problem of node redundancy, especially when the number of data points is large and the point distributions of different rows vary a lot. In [28,31] scattered data points are projected to base surfaces to do parameterization according to parameters of the projected points. This kind of methods are useful only when data points can be projected in an unambiguous way [23]. In [30] patch partitioning and polyhedral approximation are adopted to get an organized quadrilateral mesh used as the input of NURBS fitting, however the pre-processing is time consuming and the continuity between different quadrilateral patches is hard to guarantee. In [24] a method combining a hybrid optimization algorithm and iterative scheme (HOAAI) is proposed, which converges rapidly in grid data surface fitting with high accuracy, and when used for scattered data fitting, pre-processing work also need to be done for its strong constraints on the parametrization approach. In order to solve the problem of surface fitting from scattered 3D data, a simulated annealing algorithm [32] and some evolutionary intelligence techniques [23,33,34] were introduced to enhance the robustness. However, these methods are hard to handle. When the measured object is complex or the data size is large, it is not easy to ensure practicability. Under these circumstances, these methods may become low-efficiency because the introduced artificial intelligence techniques are somewhat time-consuming. In [35,36], feature sensitive parameterization is conducted for point-clouds organized in the form of triangulated meshes. They use constant knot vector and allocate more parameter space to area of interest, which results in more control points in the corresponding area. In [35], a kind of area preserving parameterization, stretch minimizing parameterization, is conducted on the image manifold of the original surface. However, the computation of image manifold is not easy and stretch minimizing parameterization requires solving a large non-linear optimization problem, which needs expensive computation. In [36], a hybrid method that combines harmonic mapping and elastic spring is adopted to generate initial parameterization, followed by an adaptive re-parameterization procedure based on relaxation field to refine fitting result. Compared with [35], it requires less computational cost, but the result is not as good geometrically.

In theory, the abovementioned methods provided for completely scattered data fitting are also an available choice for our problem, with no regard to their defects as stated above. If the quasi-scattered data is treated as completely scattered to finish the fitting problem, existing valuable information of measurement order is ignored and extra work must be done for the input data arranging [24,37,38] or initial parameterization [24,28,32–34]. For quasi-scattered data which can be easily triangulated, the methods described in [35,36] are applicable. However, the parameterization for quasi-scattered data doesn't have to be so complicated, because they have already been organized into rows, which is a more efficient structure to manipulate. For now, there is no efficient and general method that can be applied to finish the problem of NURBS surface fitting from quasi-scattered data which appears frequently in reverse engineering applications (see, e.g., [1,2,18]). In this paper, a new method based on resampling, hierarchical fitting and iterative projection is put forward to reconstruct a general NURBS surface from quasi-scattered 3D data. As is known, point projection and data parameterization are two essential problems when iterative projection idea is used for surface fitting. In the provided method, the projection result provides optimized result for subsequent parameterization, and in return the search space of point projection is pointed out accurately by previous parameterization result of the latest iteration. By this way, these two problems are merged together. The main contribution of this paper is that resampling is introduced in surface fitting of quasi-scattered data and a new resampling approach, which makes use of parameterization information of curve fitting and changes of curvature, is proposed. The whole method provided for quasi-scattered data fitting has the advantages of a simple principle, fast computational speed, wide application, and therefore has great utilization value. This paper is structured as follows: the basic definitions, principles and related pervious work for NURBS fitting are introduced in Section 2. Detailed descriptions of the proposed method are provided in Section 3, including the process of the iterative projection optimization scheme, and the implementation of resampling. Section 4 conducts a series of numerical experiments. Finally, discussions and conclusions are provided in Section 5.

2. Pervious Work

2.1. NURBS Curve and Surface

Suppose $U = \{u_0, u_1, \ldots, u_{t-1}, u_t\}$ is a non-decreasing sequence of real numbers, which is called knot vector and the real numbers u_i are its knots. Then the B-spline basis function $N_{i,p}(u)$ of p degree (or equivalently, order $p+1$) can be defined by the recurrence relations, as follows [27,39,40]:

$$N_{i,0}(u) = \begin{cases} 1, & if\ u_i \leq u < u_{i+1} \\ 0, & otherwise \end{cases} \tag{1}$$

and for $0 \leq i < r$ with $p \geq 1$:

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2}$$

Here the interval $[u_i, u_{i+p})$ is the support domain of $N_{i,p}(u)$, because the value of $N_{i,p}(u)$ is always zero outside $[u_i, u_{i+p})$. In this paper, NURBS curves and surfaces are defined on the this kind of knot vectors, in which knots distribute non-uniformly and the end knots appear the same times as the order of the B-spline.

Give a knot vector U and a chain of 3D control points $\{\mathbf{P}_i, i = 0, 1, \ldots, n\}$, a NURBS curve with p degree is defined as follows [25]:

$$\mathbf{C}(u) = \frac{\sum\limits_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum\limits_{i=0}^{n} N_{i,p}(u) w_i}, \ 0 \leq u \leq 1 \tag{3}$$

where w_i is the corresponding weight of $\mathbf{P}_i$, and $n+1$ is the number of control points. Similarly, give two vectors U, V and a grid of 3D control points $\{\mathbf{P}_{i,j}, i = 0, 1, \ldots, n; j = 0, 1, \ldots, m\}$ which forms a bidirectional net, a NURBS surface with degree (p, q) can be described by:

$$\mathbf{S}(u, v) = \frac{\sum\limits_{i=0}^{n} \sum\limits_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum\limits_{i=0}^{n} \sum\limits_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \ 0 \leq u, v \leq 1 \tag{4}$$

2.2. Basis of NURBS Curve and Surface Fitting

When the curve degree and the number of control points are already given, the basic process of NURBS curve fitting can be described summarily as follows: (1) parameterize the data points, which means to associate each point with a corresponding parameter; (2) determine the knot vector based on the result of parameterization; (3) calculate the control points based on least squares fitting. The number of control points is ascertained according to a recursive scheme based on the feedback of fitting result which means initializing the number of control points with an empirical value first and subsequently increasing or decreasing it by comparing the fitting precision with given requirement recursively. The curve degree is generally selected according to practical situation. For most applications, a cubic NURBS curve is accurate enough to express the shape of an object.

1. Parameterization of data points. As reported in [22,28,40], the primarily methods to parameterize organized data points are uniform parameterization, which is not recommended when data is unevenly spaced, centripetal parameterization, which is superior when the curvature of the measured object varies intensively, and chord length parameterization, which is most widely used and adopted in the initial part of the proposed method. Taking NURBS curve fitting from data points $\mathbf{Q} = \{\mathbf{Q}_i, i = 0, 1, \ldots, r\}$ as example, the parameterization result with chord length method $\overline{U} = \{\overline{u}_i, i = 0, 1, \ldots, r\}$ is described as follows:

$$\begin{cases} \overline{u}_0 = 0 \\ \overline{u}_i = \overline{u}_{i-1} + \frac{|\mathbf{Q}_i - \mathbf{Q}_{i-1}|}{\sum_{j=1}^{j=r} |\mathbf{Q}_j - \mathbf{Q}_{j-1}|}, \quad i = 1, 2, \ldots, r-1 \\ \overline{u}_r = 1 \end{cases} \tag{5}$$

2. Determination of knot vector. After the parameterization of data points, knot vector $U = \{u_i, i = 0, 1, \ldots n + p + 1\}$ is initialized based on the parameterization result $\{\overline{u}_i, i = 0, 1, \ldots, r\}$, as follows:

$$u_i = \begin{cases} 0, & i = 0, \ldots, p \\ (jt - \lfloor jt \rfloor)\overline{u}_{\lfloor jt \rfloor - 1} + (1 - jt + \lfloor jt \rfloor)\overline{u}_{\lfloor jt \rfloor}, & i = p + j \text{ and } j = 1, \ldots, n - p \\ 1, & i = n + 1, \ldots, n + p + 1 \end{cases} \tag{6}$$

   where n is the number of control points, t is equivalent to $(r + 1)/(n - p + 1)$ and $\lfloor jt \rfloor$ is the maximum positive integer that is less than or equal to jt. In this way, at least one parameter $\overline{u}_i$ is guaranteed in every knot interval, which assures related matrix is well-conditioned.

3. Calculation of control points. When the weights w_i are already given, least square curve fitting error can be formulized as

$$E_{ls} = \sum_{k=0}^{r} |\mathbf{Q}_k - \mathbf{C}(\overline{u}_k)|^2 = \sum_{k=0}^{r} \left(\mathbf{Q}_k - \sum_{i=0}^{n} R_{i,p}(\overline{u}_k)\mathbf{P}_i \right)^2 \tag{7}$$

Parameterization result $\{\overline{u}_i, i = 0, 1, \ldots, r\}$ and knot vector $\{u_i, i = 0, 1, \ldots, n + p + 1\}$ are obtained with the method above, $R_{i,p}(\overline{u}_k)$ is equal to $N_{i,p}(\overline{u}_k)w_i / \sum_{i=0}^{n} N_{i,p}(\overline{u}_k)w_i$ and control points $\{\mathbf{P}_i, i = 0, 1, \ldots, n\}$ are the unknowns. Rewrite Equation (7) as the following matrix form:

$$E_{ls} = \left(\mathbf{P}^T \boldsymbol{N}^T - \mathbf{F}^T \right) (\boldsymbol{N}\mathbf{P} - \mathbf{F}) \tag{8}$$

Here $\mathbf{P}$ is the matrix form of control points $\{\mathbf{P}_i, i = 0, 1, \ldots, n\}$, $\mathbf{F}$ is the vectorization result of the data points $\{\mathbf{Q}_k, k = 0, 1, \ldots, r\}$, and $\boldsymbol{N}$ is the coefficient matrix composed of basic functions:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_0 \\ \vdots \\ \mathbf{P}_n \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \mathbf{Q}_0 \\ \vdots \\ \mathbf{Q}_r \end{bmatrix}, \boldsymbol{N} = \begin{bmatrix} R_{0,p}(\overline{u}_0) & \cdots & R_{n,p}(\overline{u}_0) \\ \vdots & \vdots & \vdots \\ R_{0,p}(\overline{u}_r) & \cdots & R_{n,p}(\overline{u}_r) \end{bmatrix}$$

Then minimize the expression in Equation (8) using least square method and the control points can be solved.

The basic framework of NURBS surface fitting from gridded data can be built on the foundation of NURBS curve fitting using the hierarchical fitting idea. That is, for a given set of points

$\mathbf{Q} = \{\mathbf{Q}_{i,j}, i = 0, 1, \ldots, s; j = 0, 1, \ldots, r\}$, points $\mathbf{Q}_{i,j}$ are first fitted row by row with NURBS curves, and then the same operation is done on the resulting control points column by column to produce the final surface control points [18,21,28]. We call the fitting following row direction the u-direction fitting and that in column direction the v-direction fitting. In consideration of the application background of this paper, local modification (which often appears in the field of shape design) is not considered, and uniform weights are applied in the proposed method. The weights provide the possibility of shape modification after surface fitting, if needed [25].

2.3. Methods of NURBS Surface Fitting Based on Iterative Projection Optimization

Iterative projection optimization idea is not a novel choice among the methods of surface fitting. The major steps are: (1) project data points onto an appropriate base space, (2) do some operation, for example subdivision [41], based on the projection result, to get initial parameters, (3) get the knot vector and carry on surface fitting with the achieved parameterization result, (4) if the fitting quality doesn't satisfy the given requirement, do iteration to modify the projection result and the parameterization. So when it is used for NURBS surface fitting, an iteration scheme and a projection algorithm are absolutely necessary.

In references [28,41,42], relevant research about point projection has been presented, and the basic process is usually performed in an iterative fashion, which generally has two steps: first to obtain a good start value; and second to operate the iteration until the distance converges to its minimum. With good start value, method based on Newton iteration is an excellent choice not only for its convergence speed but also for its stability. In this paper, approach based on Newton iteration is applied to calculate the projection points in NURBS curve and surface, and the basic process is provided in the next subsection.

2.3.1. Calculating the Projection Point in Curve

A clear explanation of projection point used in this paper is listed here: $\mathbf{C}(u)$ is a given space curve, $\mathbf{Q}$ is a given point in the space, $\mathbf{C}(u^{pro})$ is a point on the curve, if $|\mathbf{C}(u^{pro}) - \mathbf{Q}|$ is the least distance between $\mathbf{Q}$ and $\mathbf{C}(u)$, then $\mathbf{C}(u^{pro})$ is the projection point of $\mathbf{Q}$ on $\mathbf{C}(u)$ in this paper.

In the left picture of Figure 2, vectors $\vec{n}$, $\vec{t}$, $\vec{b}$ are respectively the principal normal vector, the tangent vector, and the binormal vector of curve $\mathbf{C}(u)$ at point $\mathbf{C}(u^{pro})$, and plans Π_{T1}, Π_{N1}, Π_{T2} are the corresponding osculating plane, normal plane and rectifying plane. By our definition of $\mathbf{C}(u^{pro})$ as a point on curve closest to the given point $\mathbf{Q}$, $\mathbf{Q}$ has to be in the curve's normal plane at $\mathbf{C}(u^{pro})$. Thereout, an objective function is given as follows to calculate the projection point $\mathbf{C}(u^{pro})$ on curve $\mathbf{C}(u)$:

$$f(u) = \mathbf{C}'(u) \bullet (\mathbf{C}(u) - \mathbf{Q}) \tag{9}$$

where $\mathbf{C}'(u)$ is the derivative of curve $\mathbf{C}(u)$ with respect to parameter u.

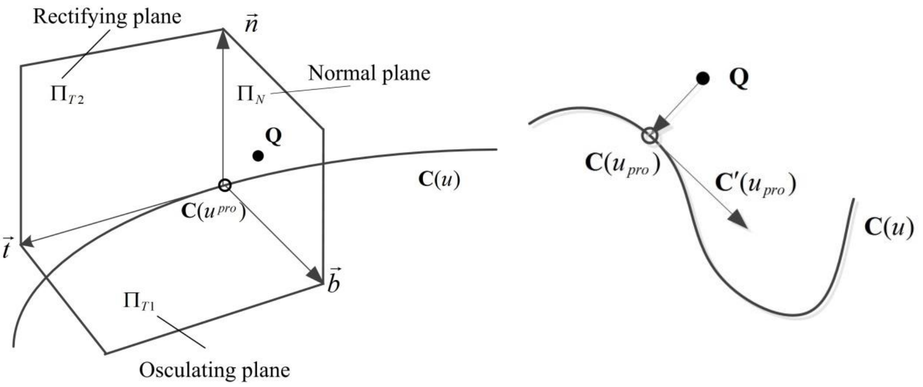


Figure 2. The projection point in a curve.

According to the definition of normal plane, the projection point is the solution of $f(u) = 0$. That is $\mathbf{C}'(u^{pro})\bullet(\mathbf{C}(u^{pro}) - \mathbf{Q}) = 0$. Assuming that the start value u_{0-pro} is already known, u_{i-pro} is the result of *i*th Newton iteration, then the result of the (*i* + 1)th iteration is described as:

$$u_{(i+1)-pro} = u_{i-pro} - f(u_{i-pro})/f'(u_{i-pro}) \tag{10}$$

The convergence criteria are usually Euclidean distance $\left|\mathbf{C}(u_{i-pro}) - \mathbf{Q}\right|$ and cosine value of the angle between $\mathbf{C}'(u_{i-pro})$ and $\mathbf{C}(u_{i-pro}) - \mathbf{Q}$. If u_{i-pro} satisfies the given convergence condition, u_{i-pro} is regarded as the u^{pro}, otherwise continue Newton iteration.

2.3.2. Calculating the Projection Point in Surface

Suppose $\mathbf{S}(u,v)$ is a given surface, $\mathbf{Q}$ is a given point in the space, $\mathbf{S}(u^{pro},v^{pro})$ is a point on the curve, if $|\mathbf{S}(u^{pro},v^{pro}) - \mathbf{Q}|$ is the least distance between $\mathbf{Q}$ and $\mathbf{S}(u,v)$, then $\mathbf{S}(u^{pro},v^{pro})$ is the projection point of $\mathbf{Q}$ on $\mathbf{S}(u,v)$ in this paper.

For surface $\mathbf{S}(u,v)$, only when $\mathbf{Q}$ is in the surface's normal line at point $\mathbf{S}(u^{pro},v^{pro})$, $|\mathbf{S}(u^{pro},v^{pro}) - \mathbf{Q}|$ is the least distance between $\mathbf{Q}$ and $\mathbf{S}(u,v)$. Based on this, an matrix function $\mathbf{f}(u,v)$ which consists of two dot product functions $f_1(u,v)$ and $f_2(u,v)$, is built as follows:

$$\mathbf{f}(u,v) = \begin{bmatrix} f_1(u,v) \\ f_2(u,v) \end{bmatrix} = \begin{bmatrix} \mathbf{S}_u(u,v)\cdot(\mathbf{Q}_{l,k} - \mathbf{S}(u,v)) \\ \mathbf{S}_v(u,v)\cdot(\mathbf{Q}_{l,k} - \mathbf{S}(u,v)) \end{bmatrix} \tag{11}$$

where $\mathbf{S}_u(u,v)$ and $\mathbf{S}_v(u,v)$ are the first partial derivatives of $\mathbf{S}(u,v)$ in u-direction and v-direction, respectively. Then its projection point on $\mathbf{S}(u,v)$ is the point with a parameter pair $(u_{l,k},v_{l,k})$ where $\mathbf{f}(u_{l,k},v_{l,k}) = 0$ is met. Let matrix $\gamma_{i-pro} = \left[\Delta u_{i-pro}, \Delta v_{i-pro}\right]^T = \left[u_{(i+1)-pro} - u_{i-pro}, v_{(i+1)-pro} - v_{i-pro}\right]^T$ where (u_{i-pro}, v_{i-pro}) is the output of the *i*th Newton iteration and $(u_{(i+1)-pro}, v_{(i+1)-pro})$ is the output of the (*i* + 1)th iteration. γ_{i-pro} can be gained in the *i*th iteration by solving below matric equation:

$$\gamma_{i-pro} = -\mathbf{J}_{i-pro}^{-1}\mathbf{f}_{i-pro} \tag{12}$$

All the notations in Equation (10) are in matrix form, among which $\mathbf{f}_{i-pro}$ is the value of object matrix function $\mathbf{f}(u,v)$ at (u_{i-pro}, v_{i-pro}), and $\mathbf{J}_{i-pro}$ is the Jacobian matrix of $\mathbf{f}(u,v)$ evaluated at (u_{i-pro}, v_{i-pro}). Then the output of the (*i* + 1)th iteration is:

$$\begin{aligned} u_{(i+1)-pro} &= u_{i-pro} + \Delta u_{i-pro} \\ v_{(i+1)-pro} &= v_{i-pro} + \Delta v_{i-pro} \end{aligned} \tag{13}$$

Give convergence criterions based on Euclidean distance $\left|\mathbf{S}(u_{i-pro}, v_{i-pro}) - \mathbf{Q}\right|$, cosine value of the angle between $\mathbf{S}(u_{i-pro}, v_{i-pro}) - \mathbf{Q}$ and $\mathbf{S}_u(u_{i-pro}, v_{i-pro})$, and cosine value of angel angle between $\mathbf{S}(u_{i-pro}, v_{i-pro}) - \mathbf{Q}$ and $\mathbf{S}_u(u_{i-pro}, v_{i-pro})$. If γ_{i-pro} satisfies the judgement criteria, $\mathbf{S}(u_{i-pro}, v_{i-pro})$ is the final projection point.

3. New Method of Surface Fitting from Scattered Data Points

As described in the introduction, for quasi-scattered data, a set of 3D points $\mathbf{Q} = \{\mathbf{Q}_{i,j}, i = 0, 1, \ldots, s; j = 0, 1, \ldots, r_i\}$ can be regarded as the measured data of a surface $\mathbf{S}$, where r_i is the number of data points in the ith row, and for any $i, j \in 0, 1, \ldots, s$, r_i does not have to be equal to r_j. Assuming that points in the same row are captured from approximate iso-parametric or section curves, and each of these approximate lines never intersects with the others. Under this assumption, if curve $\mathbf{C}_l(u)$ is the curve fitting from points $\mathbf{Q}_{l,j}, j = 0, 1, \ldots, r_l$, and with fitting error ignored, it must be a part of $\mathbf{S}$ and will not intersect with other fitting curves, so in order to overcome the problem resulting from the randomness of point numbers in each row, a new method is given here: (1) first fits the points in $\mathbf{Q}$ row by row with NURBS curves, (2) resamples on the resulting curves $\mathbf{C}_l(u), l = 0, 1, \ldots, s$ and ensure equivalent number of points in each row, and (3) constructs a NURBS surface based on the resampled data $\overline{\mathbf{Q}}$. Considering that no local modification is used in the proposed method, uniform weights are applied in both curve fitting and surface fitting, and for convenience of calculation, all weights are set equal to constant 1.

In the first and third part of this method, an iterative projection optimization idea is used, and it is not novel in NURBS object fitting. For example, in reference [24], a method based on iterative projection optimization is proposed, and it is outstanding in existing methods for its high accuracy and convergence speed when it comes to grid data. In [17], an error term called squared distance minimization (SDM) is introduced for planar curve fitting. It is defined by a curvature-based quadratic approximant of squared distances from data points to a fitting curve. A method based on this error term converges with fewer iterations comparing with methods based on point distance, but for space curves, the description of error term SDM may become too complicated. However, the curve fitted from measured points in one row of quasi-scattered data is generally a space curve, especially when the measured data is acquired by portable 3D coordinate measurement machines, so to be more general, point distance, which is widely used in practice for parametric curve and surface fitting, is selected in this paper.

In our method, new algorithms based on iterative projection optimization are given for NURBS curve and surface fitting, in which problems of data parameterization and point projection are merged together to reduce the time cost. A new resample approach which makes use of parameterization information of curve fitting and changes of curvature is proposed, which makes our method has advantages in accuracy and efficiency especially for surface fitting from quasi scattered data. The flowchart of this method is shown in Figure 3, and concrete algorithms and implementing procedures of three major parts: NURBS curve fitting, resampling and NURBS surface fitting are introduced separately in detail in Sections 3.1–3.3.

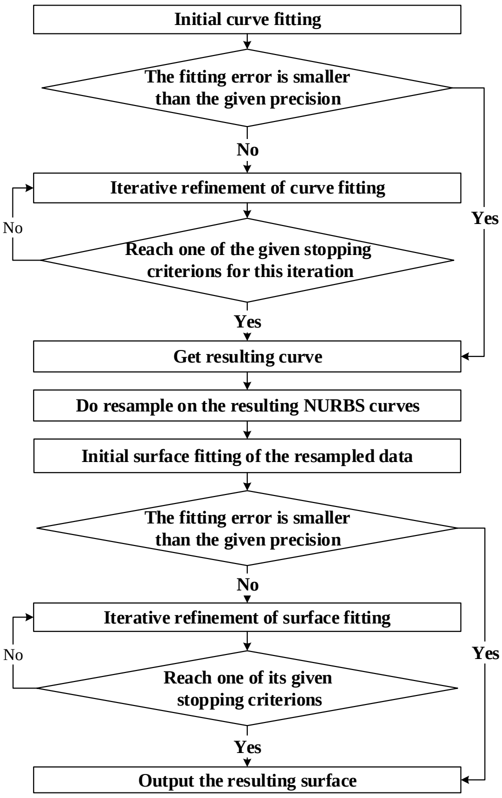

Initial curve fitting
The fitting error is smaller than the given precision
No
Iterative refinement of curve fitting
No
Yes
Reach one of the given stopping criterions for this iteration
Yes
Get resulting curve
Do resample on the resulting NURBS curves
Initial surface fitting of the resampled data
The fitting error is smaller than the given precision
No
Iterative refinement of surface fitting
No
Yes
Reach one of its given stopping criterions
Yes
Output the resulting surface

Figure 3. Framework of the method proposed in this paper.

3.1. NURBS Curve Fitting in Our Method

As can be seen in Figure 3, in order to construct a NURBS surface from quasi-scattered data $\mathbf{Q} = \{\mathbf{Q}_{i,j}, i = 0,1,\ldots,s; j = 0,1,\ldots,r_i\}$, a set of NURBS curves must be obtained which accurately reflect the characteristics of measured object. In this part iterative projection optimization idea is used for its obvious advantages in simplicity and generality. The curve fitting for data points in each row is an iterative process where an initial fitting curve is firstly achieved in accordance with the thought provided in Section 2.2 and then followed by an iterative refinement based on projection. For the calculation of projection point, Newton iteration principle is used because with a good start value, it converges rapidly and stability. During one iteration of this algorithm, every $\mathbf{Q}_{i,j}$ is projected onto the output of last iteration with its parameterization result in last iteration as the initial value of projection, and subsequently, reparameterize the data points based on the distribution of their corresponding projection points. With the given definitions of relevant symbols,

$\mathbf{Q}_l$	$\mathbf{Q}_l = \left\{\mathbf{Q}_{l,j}, j = 0,1,\ldots,r_l\right\}$, data points in lth row of the given data $\mathbf{Q}$
$k_{\max}$	The maximum number of iteration
ε_0	The given fitting precision
p	Curve degree
$n+1$	The number of control points
$\overline{\boldsymbol{U}}_l^{(k)}$	$\overline{\boldsymbol{U}}_l^{(k)} = \{\overline{u}_{l,j}^{(k)}, j = 0,1,\ldots,r_l\}$, parameterization result in kth iteration
$\boldsymbol{U}_l^{(k)}$	The knot vector in kth iteration
$\mathbf{P}_l^{(k)}$	$\{\mathbf{P}_l^{(k)} = \mathbf{P}_{l,j}^{(k)}, j = 0,1,\ldots r_l\}$, control points in kth iteration
$\mathbf{C}_l^{(k)}(u)$	The resulting curve of kth iteration
$u_{l,j}^{(k),\text{pro}}$	The resulting parameter of point projection from point $\mathbf{Q}_{i,j}$ to curve $\mathbf{C}_l^{(k)}(u)$
$\overline{e}_l^{(k)}$	The average distance from any point in $\mathbf{Q}_l$ to its projection point on $\mathbf{C}_l^{(k)}(u)$
$\mathbf{C}_l(u)$	The final fitting curve corresponding to $\mathbf{Q}_l$

The detail process of curve fitting for $\mathbf{Q}_l$ is as follows Algorithm 1:

Algorithm 1

```
1.  $k = 0$ (initial curve fitting)
2.  Do parameterization for $\mathbf{Q}_l$ by Equation (5), save the result as $\overline{\boldsymbol{U}}_l^{(0)} = \{\overline{u}_{l,j}^{(0)}, j = 0, 1, \ldots, r_l\}$
3.  Determine the knot vector $\boldsymbol{U}_l^{(0)}$ based on $\overline{\boldsymbol{U}}_l^{(0)}$ by Equation (6).
4.  Calculate control points $\mathbf{P}_l^{(0)}$ according to Equations (7) and (8) with parameterization result $\overline{\boldsymbol{U}}_l^{(0)}$ and
    knot vector $\boldsymbol{U}_l^{(0)}$.
5.  Obtain the curve $\mathbf{C}_l^{(0)}(u)$ based on $\boldsymbol{U}_l^{(0)}$ and $\mathbf{P}_l^{(0)}$.
6.  for $j = 0, 1, \ldots, \mathrm{r}_l$ do
7.      Do point projection from point $\mathbf{Q}_{l,j}$ to curve $\mathbf{C}_l^{(0)}(u)$: let $\overline{u}_{l,j}^{(0)}$ be the start value of Newton iteration,
        calculate the projection parameter $u_{l,j}^{(0),\mathrm{pro}}$ following the way presented in Section 2.3.1.
8.  end for
9.  Calculate the average distance $\overline{e}_l^{(0)}$
10. if $\overline{e}_l^{(0)} \leq \varepsilon_0$ do
11.     $\mathbf{C}_l(u) \leftarrow \mathbf{C}_l^{(0)}(u)$
12.     end the algorithm
13. $k \leftarrow k + 1$
14. Update parameterization result $\overline{\boldsymbol{U}}_l^{(k)}$ by projection result: $\overline{u}_{l,j}^{(k)} \leftarrow u_{l,j}^{(k-1),\mathrm{pro}}$
15. Determine $\boldsymbol{U}_l^{(k)}$ from updated $\overline{\boldsymbol{U}}_l^{(k)}$ by Equation (6)
16. Calculate control points $\mathbf{P}_l^{(k)}$ according to Eq. (7)-(8) with $\overline{\boldsymbol{U}}_l^{(k)}$ and $\boldsymbol{U}_l^{(k)}$, and gain the curve $\mathbf{C}_l^{(k)}(u)$
17. for $j = 0, 1, \ldots, \mathrm{r}_l$ do
18.     Do point projection from point $\mathbf{Q}_{l,j}$ to curve $\mathbf{C}_l^{(k)}(u)$ by the presented way in Section 2.3.1 to get $u_{l,j}^{(k),pro}$,
        and in this projection step $\overline{u}_{l,j}^{(k)}$ is the start value of Newton iteration.
19. end for
20. Calculate the average distance $\overline{e}_l^{(k)}$.
21. if $\overline{e}_l^{(k)} \leq \varepsilon_0$ or $k = k_{\max}$ do
22.     $\mathbf{C}_l(u) \leftarrow \mathbf{C}_l^{(k)}(u)$
23.     end the algorithm
24. go back to step 13
```

In the kth iteration of the iterative projection optimization procedure, projection parameters gained from the $(k-1)$th iteration are used to modify the previous parameterization in order to optimize the fitted curve. The projection point is calculated using Newton iteration approach presented in Section 2.3.1, with previous parameterization result used as the initial value. For example, in Figure 4, $\mathbf{C}_l^{(k-1)}(u)$ is the fitted curve of the $(k-1)$th iteration, $\overline{u}_{l,j}^{(k-1)}$ is the corresponding parameterization result of point $\mathbf{Q}_{l,j}$, when do point projection from $\mathbf{Q}_{l,j}$ to $\mathbf{C}_l^{(k-1)}(u)$, parameter $\overline{u}_{l,j}^{(k-1)}$ is used as the initial value, and the resulting projection parameter is $\overline{u}_{l,j}^{(k-1),\mathrm{pro}}$. In the kth iteration, projection parameter $\overline{u}_{l,j}^{(k-1),\mathrm{pro}}$ is used to update the parameterization, after which corresponding parameter of point $\mathbf{Q}_{l,j}$ is modified to $\overline{u}_{l,j}^{(k)}$, and the fitted curve is updated to $\mathbf{C}_l^{(k)}(u)$. When projecting $\mathbf{Q}_{l,j}$ to $\mathbf{C}_l^{(k)}(u)$, $\overline{u}_{l,j}^{(k)}$ becomes the initial value.

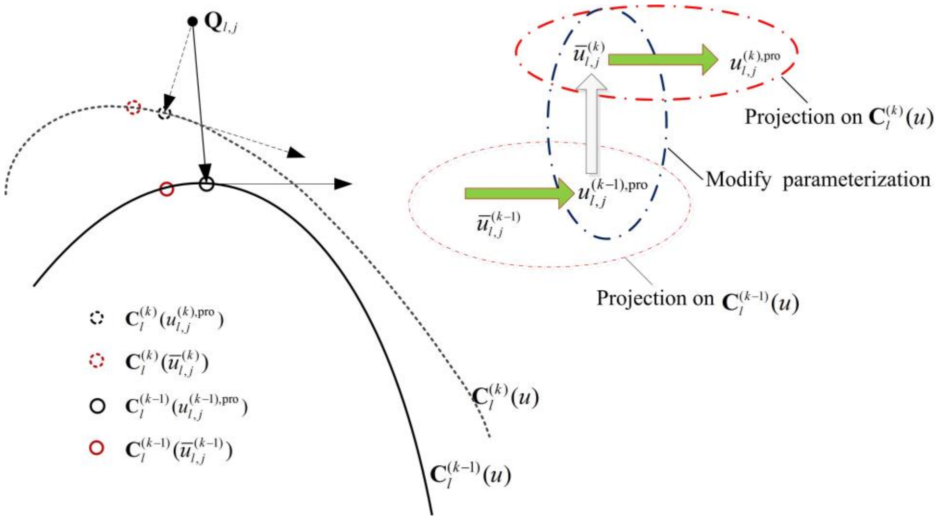

$\mathbf{Q}_{l,j}$
$\bar{u}_{l,j}^{(k)}$
$u_{l,j}^{(k),\text{pro}}$
Projection on $\mathbf{C}_l^{(k)}(u)$
Modify parameterization
$u_{l,j}^{(k-1),\text{pro}}$
$\bar{u}_{l,j}^{(k-1)}$
Projection on $\mathbf{C}_l^{(k-1)}(u)$
○ $\mathbf{C}_l^{(k)}(u_{l,j}^{(k),\text{pro}})$
○ $\mathbf{C}_l^{(k)}(\bar{u}_{l,j}^{(k)})$
○ $\mathbf{C}_l^{(k-1)}(u_{l,j}^{(k-1),\text{pro}})$
○ $\mathbf{C}_l^{(k-1)}(\bar{u}_{l,j}^{(k-1)})$
$\mathbf{C}_l^{(k)}(u)$
$\mathbf{C}_l^{(k-1)}(u)$

Figure 4. The process of iterative projection optimization during curve fitting.

The time complexity of the algorithm is $O(r_l)$. Calculation of projection takes up the largest part of time cost. The authors sampled a row of 4000 data points from the model in example 1, and fit them using this algorithm. For every iteration when $k > 0$, more than 90% time is spent on the computation of projection. In this algorithm, parameterization result provides a good initial value for the calculation of projection point which makes Newton iteration method reach optimal projection result quickly. For the whole process, the maximum iteration number $k_{\max} = 3$ is enough for most issues.

3.2. The Resample Approach in Our Method

In this paper, resampling is applied and it plays an important role in our method, because based on it the input quasi-scattered data can be converted into grid data. It is significant for the fitting accuracy and time cost of the whole method.

Today some well-known sampling approaches are usually uniform sampling, patch size based sampling, curvature-based sampling, the equal arc length sampling and the equal parameter sampling [43–45]. Uniform sampling first generates a straight line with sample nodes uniformly distributed on it, and then projects these nodes to the curve to get final sample points. It is not recommended when the curve is complex. Patch size-based sampling [43,44] divides the object into a set of ordinal units based on the knot vector. These units are ranked based on their own geometry sizes and points are distributed according to this ranking. That is, unit of the higher rank contains the more sample points, in which case, some important information in unit of very lower rank might be ignored. In curvature-based sampling, the sampling object is divided into independent units and for each unit the most critical points are first selected depending on its maximum and minimum curvature. Then more sample points are added to each unit based on the whole distribution of overall critical points, and the curvature variation can be reflected accurately by the resulting points. The equal arc length sampling, which adjusts the sampling density automatically from the curve slope, is also sensitive to curvature change. The equal parameter sampling, distributing sample points equally in parameter domain, is usually a nice choice when sample on parameter model with nonsignificant curvature changes.

As Figure 3 shows, in order to reconstruct a surface from a given set of quasi-scattered data $\mathbf{Q} = \{\mathbf{Q}_{i,j}, i = 0,1,\ldots,s; j = 0,1,\ldots,r_i\}$, NURBS curves are obtained, and resampling is carried out on each of the resulting curves. During the process of resampling, three problems should be taken into consideration: (1) the number of sampling points is unknown, (2) guarantee that the number of resampled points in each row is the same, and (3) the resampled data should reflect the shape of surface without distortion. As mentioned above, if we just consider the first and the second problem, equal parameter sampling technique is enough and simple to realize. However, when the surface is complex in curvature, the resampling becomes troublesome through existing sample approaches.

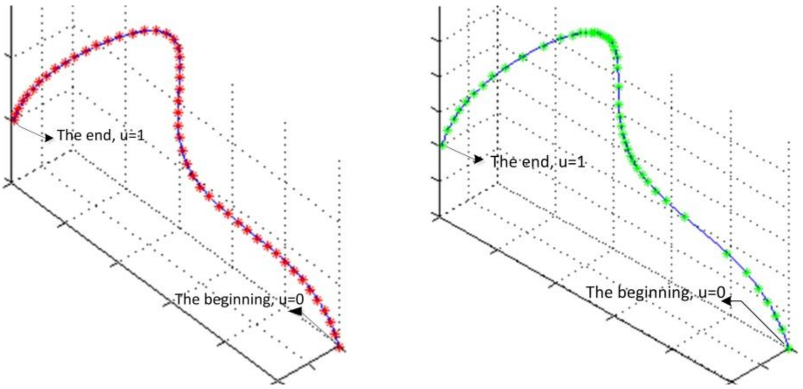
Here a new resampling algorithm is given based on the parameterization result of curve fitting and changes of curvature. The basic idea is to sample more points in the area that curvature of curve changes rapidly and to sample less points in the area that the curve is gentle. The number of resampling points is set to be a little larger than the maximum number of original data points in each row. By this way, the resampling points not only preserve the information of original data completely but also include some important shape information acquired from the fitting curve. If the number is too large, additional resampling error may be introduced.

In our method, resampling is conducted on the curve $\mathbf{C}_l(u)$ fitted from each row of the measured data. When resampling on the parameter curve, the curvature variation with respect to the parameter should be gained first. In this paper, we add three more parameters between every two parameters in parameterization result, and calculate the curvature at both parameterization result and the added ones to get the curvature variation. For the reason that NURBS curve is a kind of spline curve, which means it is piecewise and segmented by knot vector, the calculated curvature has slight fluctuation in each segment. Therefore, GLPF (Gaussian low-pass filter) is adopted to smooth the curvature plot, and the parameters that achieve local maximum are saved as local peaks.

Calculate the curvature integral value between any two adjacent parameters. For two adjacent parameters u_a and u_b, $|u_a - u_b| \cdot |curvature(u_a) + curvature(u_a)|/2$ is used as the approximate curvature integral value. Because the parameterization in initial curve fitting is based on chord-length, the geometric meaning of the curvature integral is that it is approximately proportional to the angle that the tangent vectors have rotated from parameter u_a to u_b. Adding the results together, we gain the whole integral value.

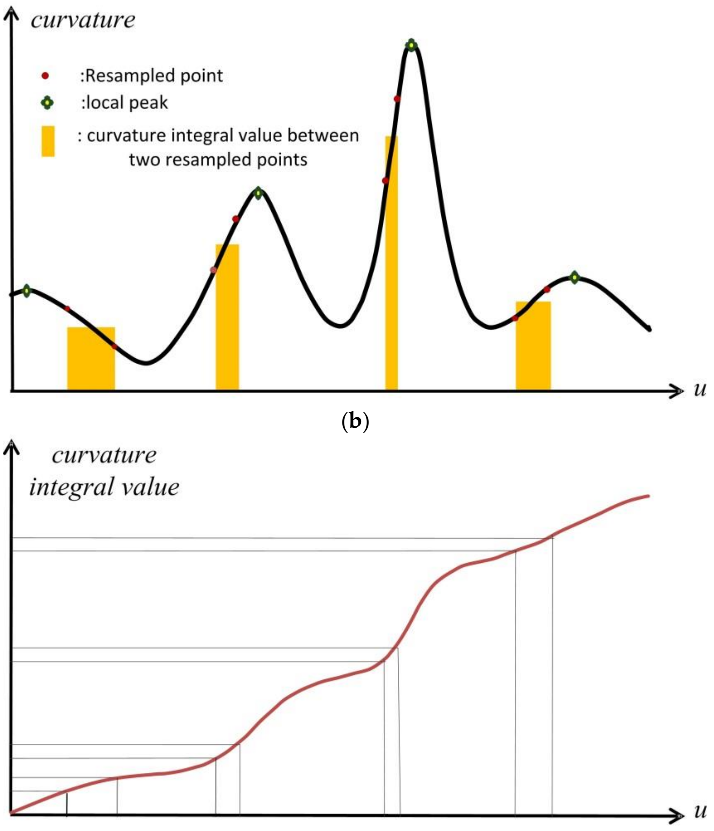
Next the whole integral value is divided into r parts (where r + 1 is the pre-set number of resampling point) equally, and the resulting unit controls the distribution of resampling points. That is, the curvature integral value between any two adjacent resampled parameters should be equal. By this way, it is guaranteed that there are more resampled points in the highly curved regions.

Figure 5 gives a clear explanation of the above process. Figure 5a–d are respectively the measured data using equal parameter sampling technique, the curvature variation after GLPF, the curvature integral value, and the resample result using the provided algorithm.



(**a**) (**d**)

Figure 5. *Cont.*



(**b**)

(**c**)

Figure 5. The process of resampling. (**a**–**d**) are respectively the measured data using equal parameter sampling technique, the curvature variation after GLPF, the curvature integral value, and the resampled points using the provided approach.

When resampling on the curves $\mathbf{C}_l(u)\ l = 0, 1, \ldots, s$, definitions of the relevant symbols in resample are given as follows:

$\overline{\boldsymbol{U}}_l$	The parameterization result of points in the lth row after curve fitting
$r + 1$	The pre-set number of points resampled from each fitted curve
$\overline{\mathbf{Q}}$	Save all the resampling points from curves $\mathbf{C}_l(u)$, $l = 0, 1, \ldots, s$
$\overset{\frown}{u}_{l,j}$	Parameter corresponding to $(j + 1)$th sampling point in the lth curve
$\overset{\frown}{\boldsymbol{U}}_l$	Save parameters of all sampling points of lth curve

The specific process of resampling is given as below Algorithm 2:

Algorithm 2

```
1. for $l = 0, 1, \ldots, s$ do
2.     for $j = 0, 1, \ldots, r_l - 1$
3.         add parameters $(\overline{u}_{l,j+1} - \overline{u}_{l,j})/4$, $(\overline{u}_{l,j+1} - \overline{u}_{l,j})/2$, $3(\overline{u}_{l,j+1} - \overline{u}_{l,j})/4$ into $[\overline{u}_{l,j},\ \overline{u}_{l,j+1}]$
4.     end for
5.     Save both parameterization result and the added parameters in array A
6.     Calculate curvature at every parameter of array A
7.     Use Gaussian low-pass filter to smooth the result in step 6, and save parameters where curvature
       achieves local maximums (local peak) in array B
8.     Calculate the curvature integral value between any two adjacent elements in A, and gain the whole
       integral value.
9.     Divide the whole integral value into $r$ parts equally, denote as $\Delta inte$
```

```
10  for $j = 1, \ldots, r-1$
11  Calculate the parameter where curvature integral is $\Delta inte \cdot j$, and save it as the $j$th resampled
    parameter $\widehat{u}_{l,j}$
12  end for
13  for $j = 0, 1, \ldots, r_l - 1$
14  if $u_{l,i}^{peak}$ exists in $(\widehat{u}_{l,j}, \widehat{u}_{l,j+1})$ do
1.  use $u_{l,i}^{peak}$ to replace one of $\widehat{u}_{l,j}, \widehat{u}_{l,j+1}$, the one nearer to $u_{l,i}^{peak}$ is replaced
15  end if
16  end for
17  Calculate point on $\mathbf{C}_l(u)$ at parameters in $\widehat{\boldsymbol{U}}_l$, and save the result in $\overline{\mathbf{Q}}$.
18  end for
```

With only consideration of the preservation of geometric properties, curvature-based sampling and the equal arc length sampling are all good choices, but it is extremely difficult for them to guarantee the number equality of resampled points for each row. The proposed algorithm overcomes this problem and keeps the advantage of curvature-based sampling.

In this algorithm, parameterization result and its subdivided result (the added parameters in step 3) are used together to get the change of curvature. It is reasonable, because distribution information of the measured data is reserved in the parameterization result.

Curvature integral value in this algorithm is approximate to the included angle of tangent vectors. When resampling based on the unit integral value $\Delta inte$, for any two contiguous resampled points the included angle of tangent vectors almost keeps the same. The resulting points can reflect the curvature change of the curve and proposed algorithm keeps the advantage of curvature-based sampling. The time complexity of the algorithm is $O(r_l)$.

3.3. NURBS Surface Fitting in Our Method

In this section, the detailed algorithm of NURBS surface fitting from the resampled data $\overline{\mathbf{Q}} = \left\{\overline{\mathbf{Q}}_{i,j}, i = 0, 1, \ldots, s; j = 0, 1, \ldots, r\right\}$ is given. In this algorithm hierarchical fitting idea, provided in Section 2.2, is applied. The whole framework is an iterative projection optimization process. As is known, when iteration projection idea is applied, in order to improve the fitting accuracy and arithmetic speed, advisable parameterization result and high-quality start value of point projection are especially important, in our method they are ensured by fusing point projection and data parameterization into each other during iteration with the same way as Section 3.1. Given the definitions of the relevant symbols:

$k_{\max}$	The maximum number of iteration
ε_0	The given fitting precision
p, q	Surface degree in u-direction and v-direction respectively
$\overline{\boldsymbol{U}}^{(k)}$	$\overline{\boldsymbol{U}}_l^{(k)} = \{\overline{u}_{l,j}^{(k)}, j = 0, 1, \ldots, r_l\}$, parameterization result in kth iteration
$\boldsymbol{U}^{(k)}$	The u-direction knot vector in kth iteration
$\overline{\mathbf{P}}^{(k)}$	$\overline{\mathbf{P}}^{(k)} = \{\overline{\mathbf{P}}_{i,j}^{(k)}, i = 0, 1, \ldots, s; j = 0, 1, \ldots, \mathrm{n}\}$ control points of u-direction in kth iteration
$\overline{\boldsymbol{V}}^{(k)}$	$\overline{\mathbf{V}}^{(k)} = \{\overline{v}_{i,j}^{(k)}, i = 0, 1, \ldots, s; j = 0, 1, \ldots, r\}$ v-direction parameterization result in kth iteration
$\boldsymbol{V}^{(k)}$	The v-direction knot vector in kth iteration
$\mathbf{P}^{(k)}$	$\mathbf{P}^{(k)} = \{\mathbf{P}_{i,j}^{(k)}, i = 0, 1, \ldots n; j = 0, 1, \ldots, m\}$, v-direction control points in kth iteration
$\mathbf{S}^{(k)}(u, v)$	The resulting NURBS surface of kth iteration
$(u_{i,j}^{(k),pro}, v_{i,j}^{(k),pro})$	The resulting parameter of point projection from $\overline{\mathbf{Q}}_{i,j}$ to $\mathbf{S}^{(k)}(u, v)$
$\overline{e}^{(k)}$	The average distance from any point in $\overline{\mathbf{Q}}$ to its projection point on $\mathbf{S}^{(k)}(u, v)$
$\mathbf{S}(u, v)$	The final fitting surface corresponding to $\overline{\mathbf{Q}}$

The details of this algorithm are as shown in the following Algorithm 3:

Algorithm 3

```
1.  begin iteration
2.    $k = 0$
3.    while (not termination condition) do
4.      for $l = 0, 1, \ldots, s$ do
5.        Do parameterization for points in $l$th line by Equation (5), save the result in $\overline{\boldsymbol{U}}^{(k)}$ denoted as $\overline{u}_{l,j}^{(k)}, j = 0, 1, \ldots, r$.
6.      end for
7.      Do average operation on $\overline{\boldsymbol{U}}^{(k)}$: $\overline{u}_j \leftarrow \sum_{l=0}^{s} \overline{u}_{l,j} / (s+1), j = 0, 1, \ldots, r$.
8.      Determine the knot vector $\boldsymbol{U}^{(k)}$ following Equation (6) with $\overline{u}_j, j = 0, 1, \ldots, r$.
9.      for $l = 0, 1, \ldots, s$ do
10.       Do curve approximation for points in $l$th line following Equations (7) and (8), and save resulting control points in $\overline{\mathbf{P}}^{(k)}$, denoted as $\overline{\mathbf{P}}_{l,j}^{(k)}, j = 0, 1, \ldots, n$.
11.     end for
12.     for $j = 0, 1, \ldots, n$ do
13.       Do parameterization for $\overline{\mathbf{P}}_{i,j}^{(k)}, i = 0, 1, \ldots, s$ by Equation (5) and save the result in $\overline{\boldsymbol{V}}^{(k)}$.
14.     end for
15.     Determine the knot vector $\boldsymbol{V}^{(k)}$ based on $\overline{\boldsymbol{V}}^{(k)}$ the same as that in $u$-direction.
16.     For $j = 0, 1, \ldots, n$ do
17.       Do curve approximation for points $\overline{\mathbf{P}}_{i,j}^{(k)}, i = 0, 1, \ldots, s$ following the way in $u$-direction fitting, and get control points $\mathbf{P}_{i,j}^{(k)}, i = 0, 1, \ldots, m$.
18.     end for
19.     Gain surface $\mathbf{S}^{(k)}(u, v)$ based on $\boldsymbol{U}^{(k)}$, $\boldsymbol{V}^{(k)}$ and $\mathbf{P}^{(k)}$.
20.     for $i = 0, 1, \ldots, s$
21.       for $j = 0, 1, \ldots, r$ do
22.         Do point projection from point $\mathbf{Q}_{i,j}$ to surface $\mathbf{S}^{(k)}(u, v)$ to get $(u_{i,j}^{(k),pro}, v_{i,j}^{(k),pro})$ by the way presented in Section 2.3.2 with $(\overline{u}_{i,j}^{(k)}, \overline{v}_{i,j}^{(k)})$ being the start value.
23.       end for
24.     end for
25.     Calculate the average distance $\overline{e}^{(k)}$.
26.     if $\overline{e}^{(k)} > \varepsilon_0$ and $k < k_{\max}$ do
27.       $k \leftarrow k + 1$, update $\overline{\boldsymbol{U}}^{(k)}$ with projection result, go back to step 7
28.     else if $\overline{e}^{(k)} \leq \varepsilon_0$ or $k = k_{\max}$ do
29.       $\mathbf{S}(u, v) \leftarrow \mathbf{S}^{(k)}(u, v)$
30.     end if
31.   end while
32. end iteration
```

In this algorithm, the hierarchical fitting idea and iterative projection optimization are combined with each other, and during the iteration process, every point is projected onto the output of last iteration with its parameterization result in last iteration as the initial value of projection, and subsequently, re-parameterize the data points based on the distribution of their corresponding projection points. In our method uniform weights are used in the fitting process. Let N be the whole number of data points, the time complexity of this algorithm is $O(N)$. In iterative projection methods the calculation of point projection is the most time-consuming part, while in this algorithm, the parameterization result of last iteration provides a suitable stat value with which the projection point is reached quickly by Newton method.

4. Experiment

In order to prove the validity of the proposed method, its performance has been tested on both simulation data and real-measured data. Some test results and a comparison with other previous methods are shown in this section, and a discussion about the factors that influence the fitting result is also presented in this chapter.

4.1. Illustrative Experiment

In this section, we do experiment on six different examples: three test problems from [24] (examples 3–5) and other three surfaces (examples 1, 2, 6). For each of example, a collection of grid data points is first fitted, and then the proposed method in Figure 3 is tested on a set of quasi scattered points. Results including fitting error, computation time, and information about final fitted surface are provided.

Example 1. *A quadric surface. This surface is given by the following equation:*

$$z = \frac{y}{1 + x^2/250000 + y^2/250000},\ x,\ y \in [-500\ \text{mm}, 500\ \text{mm}]$$

In this example, method for grid data fitting is tested on a collection of 334 × 334 data points, and a (4, 4) order fitting surface with 65 × 65 control points is obtained in 2.3 min, for which the average fitting error is 1.61×10^{-3} mm and the maximum error is 2.97×10^{-3} mm. We chose 97,612 points randomly from the grid data, test the proposed method in Figure 2 on the resulting data, and a (4, 4) order NURBS surface of 65 × 65 control points is achieved in 3.1 min with an average fitting error of 1.67×10^{-3} mm, a maximum fitting error of 3.01×10^{-3} mm. Figure 6 presents the randomly chosen data and its fitting surface.

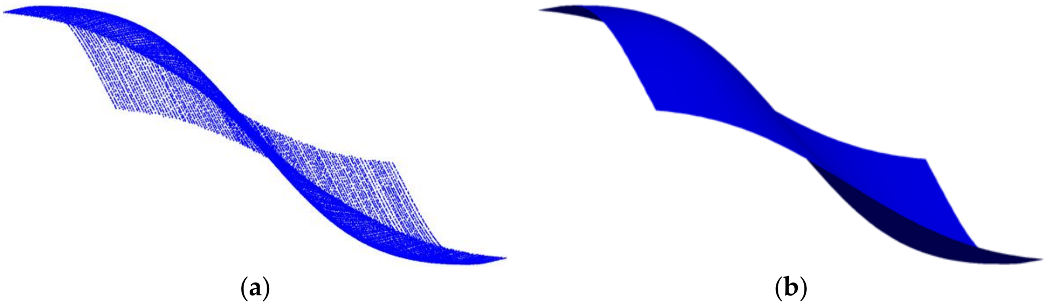


Figure 6. Surface fitting of a quadric surface: (**a**) the data points chosen randomly from grid data; (**b**) the fitting surface.

Example 2. *A hat surface. This surface is given as follows:*

$$z = \frac{100 \sin \sqrt{x^2/2500 + y^2/2500}}{\sqrt{x^2/2500 + y^2/2500}},\ x,\ y \in [-400\ \text{mm}, 400\ \text{mm}]$$

In this example, a set of 267 × 267 data points is fitted with a (4, 4) order NURBS surface with 68 × 68 control points first. The average and maximum fitting error are 1.65×10^{-3} mm and 3.34×10^{-3} mm respectively, and the time cost is 1.7 min. For a collection of 53,367 scattered points chosen from the grid data, a (4, 4) order NURBS surface is obtained using the proposed method, with an average error of 1.69×10^{-3} mm and a maximum error of 3.35×10^{-3} mm in 2.4 min. The randomly chosen data and its corresponding fitting surface are displayed in Figure 7.

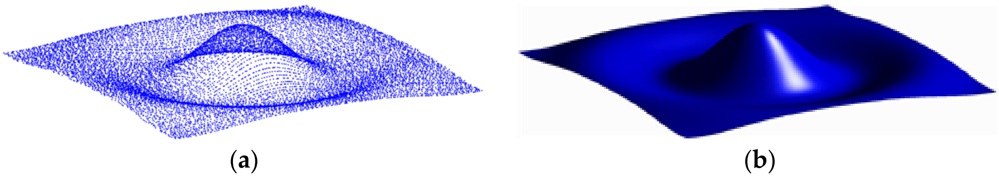


Figure 7. Surface fitting of a hat surface: (**a**) the scattered data chosen randomly from grid data; (**b**) the fitting surface.

Example 3. *A shell surface. This parametric surface is given as follows:*

$$\begin{cases} x = \frac{1}{5}(1 - \frac{v}{2\pi})\cos(2v)[1 + \cos(u)] + \frac{1}{10}\mathrm{co}(2v), & \\ y = \frac{1}{5}(1 - \frac{v}{2\pi})\sin(2v)[1 + \cos(u)] + \frac{1}{10}\sin(2v), & u, v \in [0, 2\pi] \\ z = \frac{v}{2\pi} + \frac{1}{5}(1 - \frac{v}{2\pi})sin(u), & \end{cases}$$

For this shell surface, grid data of 45 × 55 points is fitted, and a (4, 4) order NURBS surface is obtained in 87 s with an average error of 5.92×10^{-7}. We conduct our method provided in Figure 3 on 2102 scattered points, chosen randomly from the grid one, and a (4, 4) order NURBS surface with 11 × 14 control points is accepted as the final result after 2.1 min, with the average error and the maximum error of 6.63×10^{-7} and 3.83×10^{-6}, respectively. Figure 8 shows the distribution of the randomly chosen points and the corresponding fitting surface.

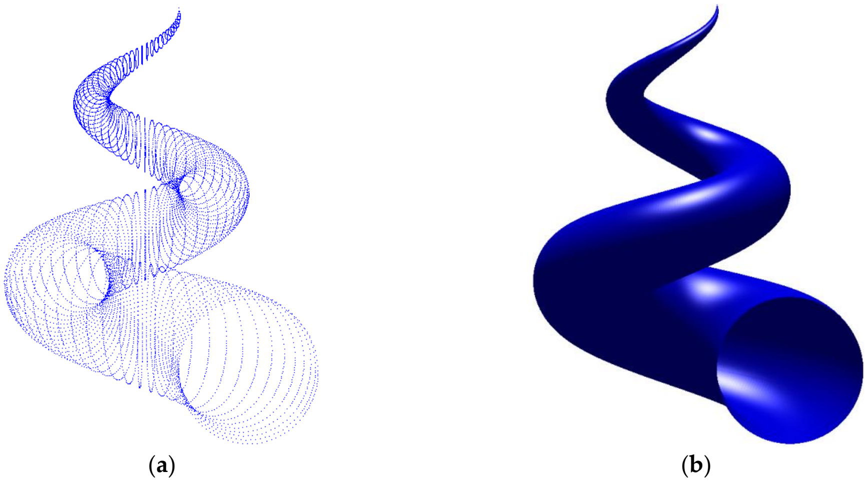


Figure 8. Surface fitting of a shell surface: (**a**) the data chosen randomly from grid data; (**b**) the fitting surface.

Example 4. *A posit surface. This posit surface is given in parametric form as follows:*

$$\begin{cases} x = A\cos(B + u)(2 + \cos(v)), & \\ y = C\cos(D - u)(2 + E\cos(F + v)), & u, v \in [0, 2\pi] \\ z = E\cos(F + u)(2 + G\cos(H - v)), & \end{cases}$$

where A = 0.655866, B = 1.03002, C = 0.74878, D = 1.40772, E = 0.868837, F = 2.43773, G = 0.495098, H = 0.377696.

For this non-zero genus surface, method for grid data is applied on a set of 100 × 105 grid points, and the proposed method is carried out on 8725 points chosen randomly from the grid data. For the former one, the average error, the maximum error and the time cost are 1.1×10^{-6}, 3.24×10^{-5} and 3.8 min successively. And for the scattered data, an average error of 2.35×10^{-6} is obtained in 5.2 min, where a (4, 4) order NURBS surface with 21 × 21 control points is accepted as the final resulting surface, the maximum error of which is 3.51×10^{-5}. Figure 9 provides the randomly chosen data and the fitted surface.

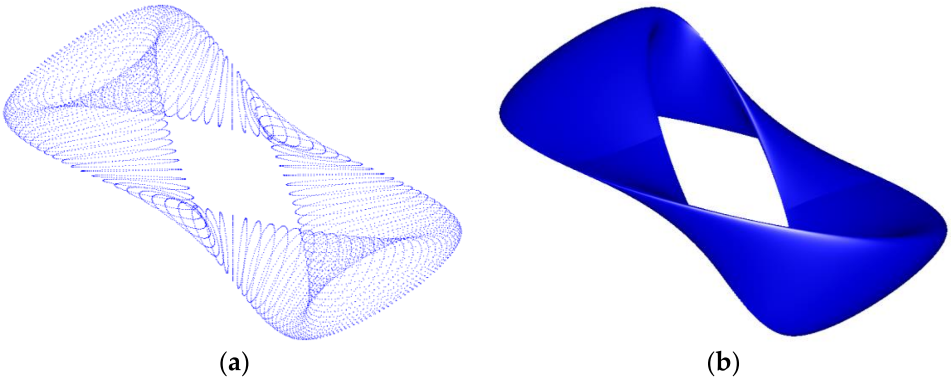


Figure 9. Surface fitting of a posit surface: (**a**) the data chosen randomly from grid data; (**b**) the fitting NURBS surface.

Example 5. *A pump surface. For the convenience of comparison, here the real measured data points provided in [24] is adopted. It is a grid data with 14 rows and the point number of every row is 370. Every data point is consisted of three variables: rotation speed, flow and moment, measured from the Mashan Pumped Storage Power Station in China [24], and the problem is to construct a model surface which reflects the concrete mathematical relations of the three variables. For this original data, method in [24] can reach a fitting accuracy of* 8.7×10^{-4} *in 1.3 min. Then for every row, we reject a part of points randomly and get a set of 4800 scattered data, from which a (5, 4) order NURBS fitted surface with 97 × 4 control points is achieved in 1.8 min by the proposed method, where the average error is* 8.9×10^{-4} *and the maximum error is* 1.71×10^{-3}*. Figure 10 presents the original data, the randomly chosen data and the fitting surface.*

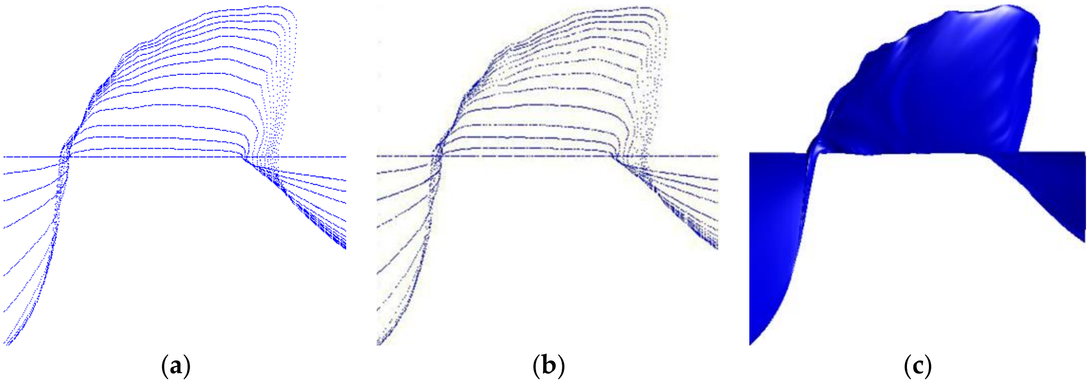


Figure 10. Surface fitting of a pump surface: (**a**) the original grid data; (**b**) the randomly chosen data; (**c**) the fitting NURBS surface.

Example 6. *A blade surface. In this example, data points are measured from a blade of aircraft engine by an in-situ automatic measuring machine system [46]. The left picture of Figure 11 is the scene photo (The inner room is the machining center). When the machining of one blade of the blisk is finished, the measuring machine rotates into the machining center to do measurement of the machined blade just like what is shown in right* *picture of Figure 11. The applied measuring probe is RSP2 probe, and it is installed on the Renscan5^{TM} probe body. The maximum permissible error value of REVO system is no more than 1 µm, and the uncertainty of the whole measuring system is less than (10 + L/30) µm (where L is the length value of the measured object, L: mm).*

The measuring machine system consists of six relative motion parts, as a result, six coordinate systems from the tip center to the machine zero position are established based on the quasi-rigid-body model for building the machine coordinate system. The workpiece coordinate system is established based on the center circle of the blisk and the locating hole, as shown in the right picture of the Figure 11. The center of the circle is used as the original point of the coordinate system. The Z-axis of the coordinate system is determined by the line that connects the centers of the circle and the locating hole. The Y-axis is determined by the vertical axis of the center circle, and then X-axis is determined by the cross product of Y-axis and Z-axis.

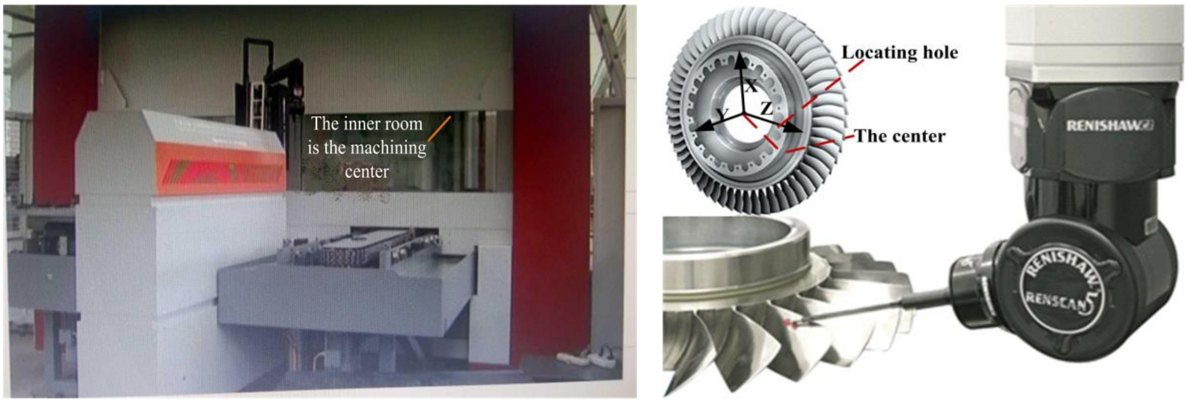
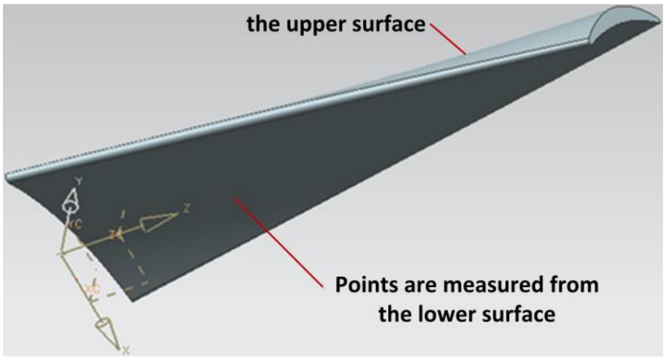


Figure 11. The applied coordinate measuring system: the left picture is the scene photo; and the right picture is the measurement on the workpiece.

Twenty one sections of the blade are determined before the measuring process firstly. Because the twist angle of the blade is as much as 64° and the space between two adjacent blades is so small, one blade is divided into two surfaces (the upper surface and the lower surface). For every surface, do the measurement following these determined sections one by one. The data that used is quasi scattered data consisting of 21 rows. The maximum number of data points in a row is 118 and the minimum is 97. The whole data number is 2255. By the proposed method, a (4, 4) order NURBS surface of 24 × 9 control points is fitted in 17 s with an average fitting error of 1.22×10^{-3} mm, a maximum fitting error of 2.27×10^{-3} mm. Figure 12 presents the data and its fitting surface.



(**a**)

Figure 12. *Cont.*

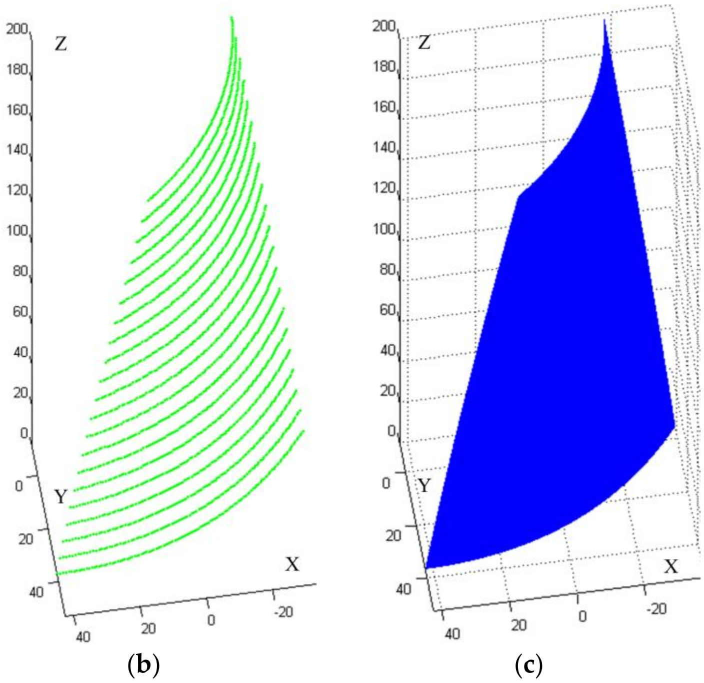


Figure 12. Surface fitting of a blade surface: (**a**) a diagram of the measurement; (**b**) the original grid data; (**c**) the randomly chosen data; (**c**) the fitting NURBS surface.

According to the above fitting results, our method performs well for scattered data chosen randomly from grid data, and the fitting results of these scattered data and grid data are in the same level with regard to both fitting error and time cost. For all experiments, the computations in this paper have been performed on a PC equipped with a core processor operating at 2.7 GHz with 2 GB of RAM. Calculation of projection takes up the dominant part of time cost. The source code has been implemented in VC++, while the resulting pictures are constructed in Matlab with the obtained control points, knot vectors, and the orders.

4.2. Comparison with Other Approaches

As stated, the proposed fitting method by curvature based resampling presents a good performance for surface fitting of problems analyzed above. Compared with other methods presented in the literature, it outstands when the fitting data is scattered, especially in terms of time cost. In order to support this claim, here we do a careful comparison with five other methods found in literature [8,18,23,24,33]. Table 1 shows the comparison results, including the average fitting error and the time cost.

The method given by Ma in [8], projects points to a base surface created from approximate boundary points or curves to do parameterization. However, it works properly only for simple surface with no self-intersecting, and hard to get a high fitting accuracy. As shown in the comparison results, it has no advantage on time cost and fitting accuracy when used for quasi scattered data fitting.

In [18], the rows are approximated independently of one another, and new knots are added into the knot vector during the curve fitting, then the merged knot vector is used for the fitting of the next row. This make the method face the problem of nodes redundancy. And in practice, it is often happens especially when the number of data points is large and the point distributions of different rows change a lot.

Table 1. This is a comparison of proposed method with four other alternative methods. NDP presents the number of data points. AFE is the average fitting error, and TC denotes the time cost.

Method	**Quadric Surface** **NDP: 97612** **AFE, TC**	**Hat Surface** **NDP: 53367** **AFE, TC**	**Shell Surface** **NDP: 2102** **AFE, TC**	**Posit Surface** **NDP: 8725** **AFE, TC**	**Pump Surface** **NDP: 4800** **AFE, TC**	**Blade Surface** **NDP: 2255** **AFE, TC**
[8] 1995, Base surface parameterization with LSQ	1.14×10^{-4}, several h	2.12×10^{-4}, several h	failed	failed	1.47×10^{-2}, 94.7 min	1.83×10^{-4}, 30–50 min
[18] 2000, Surface fitting based on hierarchical fitting and merged knot vector	failed	failed	4.51×10^{-4}, 7.6 min	1.35×10^{-3}, 14 min	4.91×10^{-3}, 8.4 min	2.58×10^{-4}, 5.9 min
[23] 2010, Particle swarm optimization	7.0×10^{-7}, >100 min	7.64×10^{-7}, >100 min	8.72×10^{-14}, 35–50 min	2.81×10^{-12}, >100 min	1.63×10^{-3}, 11.4 min	1.97×10^{-6}, 7.9 min
[33] Iteration genetic algorithm with LSQ fitting	1.79×10^{-5}, >100 min	8.7×10^{-5}, >100 min	2.83×10^{-6}, >100 min	9.1×10^{-6}, >100 min	7.43×10^{-3}, >100 min	4.11×10^{-6}, 16.4 min
[24] 2010, projected optimization and iteration	4.27×10^{-5}, 18.3 min	5.74×10^{-5} 15.9 min	3.81×10^{-6}, 11.2 min	1.80×10^{-6}, 45.7 min	3.42×10^{-3}, 32.3 min	6.72×10^{-6}, 2.1 min
Our method	1.67×10^{-6}, 3.1 min	1.69×10^{-6}, 2.4 min	6.63×10^{-7}, 2.1 min	2.35×10^{-6}, 2.8 min	8.9×10^{-4}, 1.8 min	1.22×10^{-6}, 17 s

Methods in [23], given by Gálvez, based on particle swarm optimization, can achieve perfect accuracy for examples 1–5, but its complexity and computational time are too hard to handle. Compared with it, our method has a great advantage on time cost.

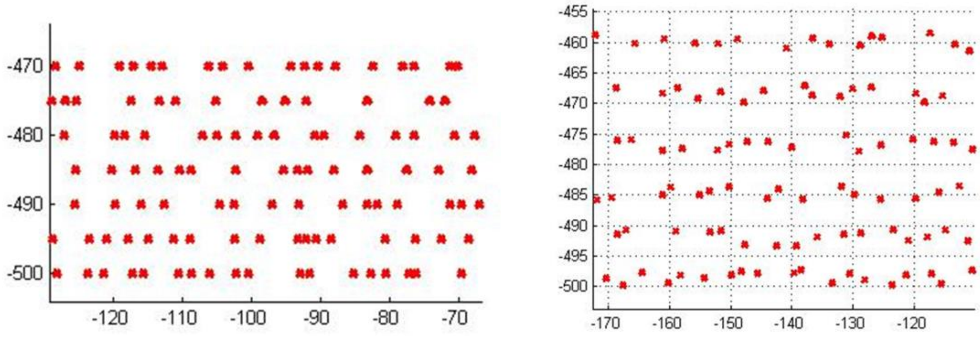
For the same quasi scattered points of examples 1–5, methods provided in [24,33] all report reasonable fitting accuracy at the same level with ours, but they all lose the competition on time expenditure.

To summarize, when used for quasi scattered data fitting, our method reaches enough accuracy with the lowest time cost compared with previous fitting approach.

4.3. Robustness

In order to test the robustness of the proposed method, we conduct the method on several data with varying degrees of scatterness acquired from examples 1 and 5. In example 1, model surface is presented in the implicit form of $z = z(x, y)$, while if we let $x = v, y = u$, it changes to an equivalent parametric form $x = x(u, v) = v, y = y(u, v) = u, z = z(u, v)$. To describe the experiment process on example 1 clearly, row-spacing, u-section line, v-interval, u-fluctuation and additional error introduced by resampling are introduced, and the special meanings of them are given as follows:

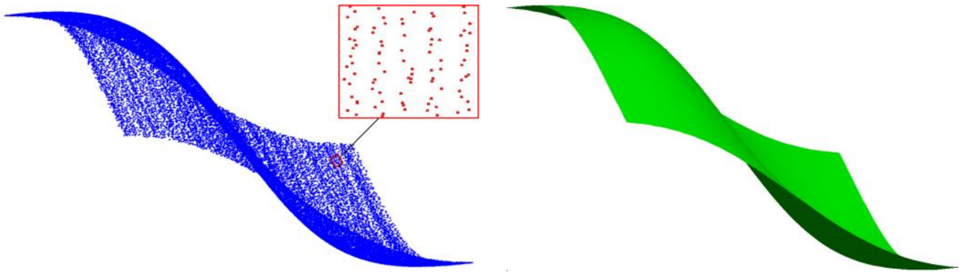
- *u-section line*: Points in such a section line which have the same coordinate in u direction.
- *row-spacing*: For two adjacent u-section lines, coordinate difference in u direction is row-spacing. If a set of section lines distribute uniformly, row-spacing is a constant value, otherwise it varies in a certain range.
- *v-interval* and *u-fluctuation*: When sampling along a u-section line, coordinate difference in v direction of two adjacent sample points is v-interval and the distance in u direction from a data point to the section line is u-fluctuation. If u-fluctuation is equal to zero anywhere, the sampling data points distribute as shown in Figure 13a, otherwise as shown in Figure 13b which is more similar to real measured data.
- *additional error introduced by resampling:* If the average distance between measured data and the final fitting surface $\mathbf{S}(u, v)$ is denoted as e called the average fitting error, and that between resampled data and $\mathbf{S}(u, v)$ is denoted as $\bar{e}$ then the absolute difference $|e - \bar{e}|$ is the additional error introduced by resampling.



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

Figure 13. Points distribution on the (u, v) domain when sampling along u-section line: (**a**) the case when u-fluctuation is equal to zero; (**b**) the case when u-fluctuation ranges from -1.5 to 1.5 randomly. In this picture, v direction is the horizontal direction, and u direction is the vertical direction.

After multiple tests on example 1, results of 12 representative experiments are listed in Table 2. Figure 14 gives the distribution of data points corresponding to the 11th experiment this table. In this table, v-interval, row-spacing, and u-fluctuation are used together to represent the scatterness of data distribution, the average error and additional error introduced by resampling are listed as the fitting results.



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

Figure 14. Distribution of data points corresponding to the 11th experiment in Table 2: (**a**) distribution in domain of quasi scattered data points; (**b**) the fitting surface.

Table 2. Experiment results of example 1. RND and NDP are the row number of data, the number of data points, respectively. RS, VIT and UFL are row-spacing, v-interval and u-fluctuation, successively. AFE and AER are respectively the average fitting error and the additional error introduced by resampling. In the RS, VIT and UFL columns, a~b means ranging from "a" to "b" randomly.

RS/mm	RND	VIT/mm	UFL/mm	NDP	AFE/mm	AER/mm
3	334	3~4.5	0.0	95,211	0.00169	0.00000
3	334	3~6	0.0	86,606	0.00171	0.00000
3	334	3~6	−0.6~0.6	86,606	0.00193	0.00001
3	334	3~6	−1.2~1.2	86,606	0.00226	0.00003
3~6	222	3~6	−0.6~0.6	57,520	0.00255	0.00003
3~6	222	3~6	−1.2~1.2	57,520	0.00316	0.00005
5	200	3~6	0.0	51,775	0.00254	0.00000
5	200	5~10	0.0	39,199	0.00304	0.00000
5	200	3~6	−1.0~1.0	51,775	0.00310	0.00006
5	200	3~6	−1.5~1.5	51,775	0.00379	0.00005
5~7.5	161	3~6	−1.0~1.0	41,668	0.00407	0.00006
5~7.5	161	3~6	−1.5~1.5	41,668	0.00484	0.00005

In example 5, the real measured data provided in [24] is adopted as the original measured data. For this data, method in [24] can reach a fitting accuracy of 8.8×10^{-4}. Here, several sets of scattered data points are acquired on the basis of the measured data. That is, each data point of the original measured data is reserved or rejected by random gating sampling, and the number of points in each row of the resulting data varies in a range randomly. Experiments are conducted on both the original measured data and the resulting scattered data to verify the robustness. Table 3 lists experimental results of the original measured data and four sets of resulting scattered data.

Table 3. Experiment results of example 5. Max-NRP and Min-NRP are the maximum and minimum number of data points in each row of the scattered data. The meanings of the other abbreviations are the same as Table 2.

RND	NDP	Max-NRP	Min-NRP	AFE
14	5180	370	370	8.7×10^{-4} [1]
14	4800	367	311	8.9×10^{-4}
14	4389	353	265	8.9×10^{-4}
14	4088	365	228	9.0×10^{-4}
14	4011	360	192	8.9×10^{-4}

[1] Experiment of the first row is conducted on the original gridded data, and experiments of other rows are conducted on quasi scattered data.

By comparing the different experimental results in Table 2, we know that the fitting precision of our method has inverse relation to the u-fluctuation and the row-spacing. Resample is brought in to implement data transformation in our method, in order to quantify its impaction on final fitting precision, additional error brought by resample (AER) is calculated in Table 2. And as shown in the results, it is always zero when sampling along sections strictly (the u-fluctuation is zero), in which case resample of course has no impact on fitting precision but this is not likely to happen in actual measurement. In general, it is not zero but small enough to be ignored compared with average fitting error (AFE).

For the real-measured data points, the fitting accuracy of method provided in [24] based on iterative projection optimization is 8.8×10^{-4} and our method can reach a fitting accuracy of 8.7×10^{-4} in few seconds, what's more, the fitting accuracy of our method nearly unchanged after removing some of the data points randomly as listed in Table 3, which illustrates the robustness against data missing and demonstrates that our method can be better applied in surface fitting of real-measured quasi-scattered points.

In order to test the performance in circumstance that there are areas with densely packed rows and areas with large distance between individual rows, we set rowing-spacing to be different value in different parts of the surface in example 2. When areas with densely packed rows (rowing-spacing is set to be 0.5~1) exist in the central part ($180 < x < 220$) where curvature change highly and areas with large distance (rowing-spacing is controlled between 10~12.5) exist in the part that the curve change gently ($30 < x < 100$), the average fitting error is 6.53×10^{-3} mm. While when the areas with large distance exist in the central part, the average fitting error is 8.61×10^{-2} mm. So it is the distribution of measured data not the distribution of rows that affect the final fitting result.

5. Conclusions

In this paper, a new method of NURBS surface fitting from quasi-scattered data is proposed. It consists of three parts: NURBS curve fitting for each row of the original data, curvature-based resampling on resulting curves and NURBS fitting for the resampled data. The proposed resampling approach is mainly on the basis of the parameterization result of the preceding curve fitting, and integrates curve curvature and local peaks. In proposed resampling approach, parameterization result and its subdivided result are used together to get the curvature variation. Chord length parameterization, which is close to arc length parameterization, is used in this paper because quasi scattered is easy to be organized into rows. Therefore, the parameterization result reserves the distribution information of original measured data, which can reflect the key geometric information of the curve. Besides, in this approach curvature is integrated along parameter u to get the curvature integral unit that used to control the distribution of the resampled data. It is easy to operate. And as the experimental results shown, it almost introduces no additional error in subsequent surface fitting. Iterative projection optimization idea is applied in the fitting process, but in our method parameterization results are used as the start value of point projection, so that problems of parameterization and calculation of projection are boned together, which makes the iteration reach reasonable results quickly and precisely. For quasi scattered data, the proposed method is efficient and superior to previous methods especially in the time coat. Experimental results described above have proved this.

Acknowledgments: The authors wish to thank the State Administration of Science, Technology and Industry for National Defense of China for the financial support through the project JSJL2014206B001.

Author Contributions: Qing Mao and Shugui Liu conceived the whole fitting method; Qing Mao and Sen Wang designed and performed the experiments; Qing Mao analyzed the data and wrote the paper; Xinhui Ma provided great help on the revision of the manuscript.

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

References

1. Várady, T.; Martin, R.R.; Cox, J. Reverse engineering of geometric models—An introduction. *Comput. Aided Des.* **1997**, *29*, 255–268. [CrossRef]
2. Zhang, X.C. Study of Inspection Technologies Based on Point Cloud for Complex Surfaces and Its System Development. Ph.D. Thesis, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China, 2006.
3. Brezina, I. Coordinate Measuring Machines and Systems. *Measurement* **1997**, *20*, 287. [CrossRef]
4. Milroy, M.J.; Weir, D.J.; Bradley, D.C.; Vickers, G.W. Reverse engineering employing a 3D laser scanner: A case study. *Int. J. Adv. Manuf. Technol.* **1996**, *12*, 111–121. [CrossRef]
5. Chen, J.; Wu, X.; Wang, M.Y.; Li, X. 3D shape modeling using a self-developed hand-held 3D laser scanner and an efficient HT-ICP point cloud registration algorithm. *Opt. Laser Technol.* **2013**, *45*, 414–423. [CrossRef]
6. Liu, S.; Zhang, H.; Dong, Y.; Tang, S.; Jiang, Z. Portable Light Pen 3D Vision Coordinate Measuring System-Probe Tip Center Calibration. *Meas. Sci. Rev.* **2013**, *13*, 194–199. [CrossRef]
7. Pettersen, A.D. System for Point-By-Point Measuring of Spatial Coordinates. U.S. Patent US6166809, 26 December 2000.
8. Ma, W.; Kruth, J.P. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Comput. Aided Des.* **1995**, *27*, 663–675. [CrossRef]
9. Floater, M.S. Parameterization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.* **1997**, *14*, 231–250. [CrossRef]
10. Sen, S.; Zheng, S.Q. Near-optimal triangulation of a point set by simulated annealing. In Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, Technological Challenges of the 1990’s, Kansas City, MO, USA, 1–3 March 1992; ACM: New York, NY, USA, 1992; pp. 1000–1008.
11. Prestifilippo, G.; Sprave, J. Optimal triangulation by means of evolutionary algorithms. In Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems, Innovations and Applications (GALESIA ’97), Glasgow, UK, 2–4 September 1997; IET: Glasgow, UK, 1997; pp. 492–497.
12. Goinski, A. Evolutionary surface reconstruction. In Proceedings of the 2008 Human System Interactions Conference, Krakow, Poland, 25–27 May 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 464–469.
13. Kodama, T.; Li, X.; Nakahira, K.; Ito, D. Evolutionary computation applied to the reconstruction of 3-D surface topography in the SEM. *J. Electron. Microsc.* **2005**, *54*, 429–435. [CrossRef] [PubMed]
14. Keller, R.E.; Banshaf, W.; Mehnen, J.; Weinert, K. CAD surface reconstruction from digitized 3D point data with a genetic programming/evolution strategy hybrid. *Adv. Genet. Program.* **1999**, *3*, 41–65.

15. Borges, C.F.; Pastva, T. Total Least Squares Fitting of Bezier and B-Spline Curves to Ordered Data. *Comput. Aided Geom. Des.* **2002**, *19*, 275–289. [CrossRef]
16. Pottmann, H.; Leopoldseder, S. A concept for parametric surface fitting which avoids the parametrization problem. *Comput. Aided Geom. Des.* **2003**, *20*, 343–362. [CrossRef]
17. Wang, W.; Pottmann, H.; Liu, Y. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.* **2006**, *25*, 214–238. [CrossRef]
18. Piegl, L.A.; Tiller, W. Surface approximation to scanned data. *Vis. Comput.* **2000**, *16*, 386–395. [CrossRef]
19. Pan, R.; Skala, V. Continuous global optimization in surface reconstruction from an oriented point cloud. *Comput. Aided Des.* **2011**, *43*, 896–901. [CrossRef]
20. Gregorski, B.F.; Hamann, B.; Joy, K. Reconstruction of B-spline surfaces from scattered data points. In Proceedings of the 2000 Computer Graphics International Conference, Geneva, Switzerland, 19–24 June 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 163–170.
21. Piegl, L.A.; Tiller, W. Algorithm for approximate NURBS skinning. *Comput. Aided Des.* **1996**, *28*, 699–706. [CrossRef]
22. Lee, E.T. Choosing nodes in parametric curve interpolation. *Comput. Aided Des.* **1989**, *21*, 363–370. [CrossRef]
23. Gálvez, A.; Iglesias, A. Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Inf. Sci.* **2010**, *192*, 174–192. [CrossRef]
24. Xie, W.C.; Zou, X.F.; Yang, J.D.; Yang, J.D.; Yang, J.B. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. *Comput. Aided Des.* **2012**, *44*, 1127–1140. [CrossRef]
25. Piegl, L.A.; Tiller, W. *The NURBS Book*, 2nd ed.; Springer: New York, NY, USA, 1997; pp. 86–100.
26. Dimas, E.; Briassoulis, D. 3D geometric modelling based on NURBS: A review. *Adv. Eng. Softw.* **1999**, *30*, 741–751. [CrossRef]
27. Tiller, W. Rational B-Spline for curve and surface representations. *IEEE Comput. Graph.* **1983**, *3*, 61–69. [CrossRef]
28. Piegl, L.A.; Tiller, W. Parametrization for surface fitting in reverse engineering. *Comput. Aided Des.* **2001**, *33*, 593–603. [CrossRef]
29. Yin, Z. Reverse engineering of a NURBS surface from digitized points subject to boundary conditions. *Comput. Graph.* **2004**, *28*, 207–212. [CrossRef]
30. Park, I.K.; Sang, U.L. Geometric modeling from scattered 3-D range data. In Proceedings of the International Conference on Image Processing, Santa Barbara, CA, USA, 26–29 October 1997; IEEE: Santa Barbara, CA, USA, 1997; pp. 712–715.
31. Kineri, Y.; Wang, M.; Lin, H.; Maekawa, T. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. *Comput. Aided Des.* **2012**, *44*, 697–708. [CrossRef]
32. Riyazuddin, M. Visualization with NURBS Using Simulated Annealing Optimization Technique. Master's Thesis, Dept. ICS, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, 2004.
33. Gálvez, A.; Iglesias, A.; Puig-Pey, J. Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. *Inf. Sci.* **2012**, *182*, 56–76. [CrossRef]
34. Sarfraz, M.; Raza, S.A.; Baig, M.H. Computing Optimized Curves with NURBS Using Evolutionary Intelligence. *Lect. Notes Comput. Sci.* **2005**, *3480*, 806–815.
35. Lai, Y.K.; Hu, S.M.; Pottmann, H. Surface fitting based on a feature sensitive parametrization. *Comput. Des.* **2006**, *38*, 800–807. [CrossRef]
36. Marinić-Kragić, I.; Ćurković, M.; Vučina, D. Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting. *Eng. Appl. Artif. Intell.* **2018**, *67*, 39–51. [CrossRef]
37. An, X. Neural Network Approach to the Reconstruction of Freeform Surfaces. *Zhongguo Jixie Gongcheng/China Mech. Eng.* **1998**.
38. Hoffmann, M.; Várady, L. Free-form Surfaces for scattered data by neural networks. *J. Geom. Graph.* **1998**, *2*, 1–6.
39. Piegl, L.A. On NURBS: A survey. *IEEE Comput. Graphics Appl.* **1991**, *11*, 55–77. [CrossRef]
40. Piegl, L.A.; Tiller, W. Computing offsets of NURBS curves and surfaces. *Comput. Aided Des.* **1999**, *31*, 147–156. [CrossRef]
41. Selimovic, I. Improved algorithms for the projection of points on NURBS curves and surfaces. *Comput. Aided Geom. Des.* **2006**, *23*, 439–445. [CrossRef]

42. Chen, X.D.; Yong, J.H.; Wang, G.; Paul, J.C.; Xu, G. Computing the minimum distance between a point and a NURBS curve. *Comput. Aided Des.* **2008**, *40*, 1051–1054. [CrossRef]
43. Obeidat, S.M.; Raman, S. An intelligent sampling method for inspecting free-form surfaces. *Int. J. Adv. Manuf. Technol.* **2009**, *40*, 1125–1136. [CrossRef]
44. Elkott, D.F.; Elmaraghy, H.A.; Elmaraghy, W.H. Automatic sampling for CMM inspection planning of free-form surfaces. *Int. J. Prod. Res.* **2002**, *40*, 2653–2676. [CrossRef]
45. Hu, J.; Li, Y.; Wang, Y.; Cai, J. Adaptive sampling method for laser measuring free-form surface. *Int. J. Adv. Manuf. Technol.* **2004**, *24*, 886–890. [CrossRef]
46. Zhang, H.T.; Liu, S.G.; Li, X.H. A study on the key techniques of application of REVO five-axis system in non-orthogonal coordinate measuring machine. *Proc. Inst. Mech. Eng. B* **2017**, *4*, 730–736. [CrossRef]