Parametric surface reconstruction from 3D point data using Partial differential equation and bilinearly blended Coons

Zaiping Zhu[a], Shuangbu Wang[b,*], Lihua You[a], Jianjun Zhang[a]

[a] The NCCA, Bournemouth University, Poole, BH12 5BB, UK
[b] Institute of Smart City and Intelligent Transportation, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China

Abstract

Existing methods for parametric surface reconstruction from 3D point data typically segment the points into multiple subsets, each fitted with a parametric surface patch. These methods face two blue primary issues. First, they fail to achieve positional continuity between adjacent patches, resulting in gaps or overlaps. Second, parameterizing the data within each subset is a challenging task. In this paper, we address these problems by proposing a novel surface reconstruction method based on Partial differential equation (PDE) deformation surfaces and bilinearly blended Coons patches. Our approach involves first extracting four boundaries for each subset. Next, we generate a bilinearly blended Coons patch from these boundaries. Any errors between the points in the subset and their corresponding points on the Coons patch are minimized or eliminated using a PDE deformation surface, which employs as many unknown constants as necessary to achieve this goal. The proposed method offers several advantages. Firstly, it ensures good positional continuity between adjacent patches, as they share common boundaries. Secondly, reconstruction errors can be easily controlled by adjusting the hyper-parameters in the PDE deformation equation, thereby changing the number of unknown constants as needed. Thirdly, the challenge of point parameterization within each subset is effectively addressed by using the bi-

*Corresponding author
Email addresses: zzhu@bournemouth.ac.uk (Zaiping Zhu), shuangbuwang@swjtu.edu.cn (Shuangbu Wang), lyou@bournemouth.ac.uk (Lihua You), jzhang@bournemouth.ac.uk (Jianjun Zhang)

linearly blended Coons patch. We validate our method on various datasets of differing complexities and shapes, with results demonstrating the effectiveness and advantages of our approach.

Keywords:
Parametric surface Reconstruction, Fourth-order PDE, PDE deformation surfaces, Bilinearly blended Coons patches, Positional continuity, Parameterizing point clouds

1. Introduction

Reconstructing surfaces from 3D point data is of great scientific and practical significance in various fields such as cultural heritage, automotive, robotics, and many others [1, 2, 3]. This process can be categorized into explicit and implicit methods based on whether explicit or implicit representations are used. Each method has its own advantages and disadvantages, and the choice should be made based on the specific tasks and their requirements. In this paper, we focus on parametric surface reconstruction from point clouds, a popular approach for surface reconstruction using explicit representation. Compared to other explicit representations like polygon surfaces, parametric surfaces offer certain benefits. They are defined by mathematical equations, which reduces storage requirements. Moreover, theoretically, different levels of continuity can be achieved between adjacent parametric surface patches by specifying various continuity conditions, which is crucial for many applications.

Current approaches for reconstructing parametric surfaces from point clouds typically involve dividing the 3D point data into subsets and approximating each subset with a parametric surface patch. However, these methods face two significant challenges. First, gaps or overlaps may occur between adjacent patches, necessitating additional postprocessing to resolve these issues. Second, the parameterization of point clouds or points within subsets remains an important and unresolved problem. To address these challenges, we propose a new reconstruction method that integrates partial differential equation (PDE) based deformation surfaces with bilinearly blended Coons patches. This method enables the reconstruction of surfaces with parametric representation from 3D point data or subsets thereof. We use bilinearly blended Coons patches to create initial parametric surfaces that pass through the boundaries of each subset. Subsequently, we propose a PDE involving as many unknown constants as needed, derived from the governing equation for the elastic bending of thin plates. The particular solution to this PDE is used to create a deformation surface, which is superimposed on the Coons patch to fit the interior points without altering the boundaries of the Coons patches. This ensures that adjacent parametric patches share the same boundary and are connected seamlessly without gaps or overlaps. Additionally, the bilinearly blended Coons patch serves as an ideal base surface for easily and properly parameterizing point clouds. For each point in the point clouds or subsets, the closest point on the corresponding Coons patch and its u and v parametric values can be accurately determined.

The organization of this paper is as follows: Section 2 provides an overview of related work. In Section 3, we outline the pipeline of our proposed method. Section 4 explores bilinearly blended Coons patches and PDE deformation surfaces in detail. The results of surface reconstruction from various datasets are presented in Section 5. Finally, Section 6 concludes the paper and discusses future work.

2. Related work

For surface reconstruction from 3D point data, there are numerous surface representations to choose from. Broadly, these representations can be categorized into two primary types: explicit and implicit. Each type has its own advantages and disadvantages. For a more comprehensive and detailed introduction to various methods of surface reconstruction from point clouds, please refer to these review papers [4, 5, 6, 7, 8].

Methods in the first category include parametric surfaces and triangular surfaces, etc. Triangular surfaces are typically reconstructed using Delaunay triangulations and Voronoi diagrams [9, 10]. These methods use numerous flat triangles to approximate curved shapes, which can require very high storage. A parametric surface, on the other hand, is defined by a mathematical expression so it is a very compact representation. Parametric surfaces also have the advantages of easily generating points on the surface and intuitively altering the surface shape by adjusting the position of control points or other parameters in the mathematical expression. However, determining whether a point lies on which side of a parametric surface is not straightforward. Examples of parametric surfaces include Bézier surfaces [11], B-spline surfaces [12], NURBS surfaces [13], and PDE surfaces [21], etc.

For parametric surface reconstruction from 3D point data, it is often necessary to use multiple patches, each defined by a mathematical expression, to construct a complex shape. A significant challenge in this type of surface reconstruction is achieving good continuity between patches, as existing parametric surface representations struggle to address this issue. Another challenge is parameterizing point clouds or the points within subsets of a point cloud to obtain suitable parameters for each point. Zhu et al. [14] provide a comprehensive survey on various methods of parameterizing 3D point clouds for parametric surface reconstruction.

In this paper, we propose using bilinearly blended Coons patches and a particular solution to a fourth-order PDE involving a lateral force to address these challenges. Our method offers two key advantages. First, it guarantees positional continuity between adjacent surface patches. Second, the bilinearly blended Coons patch serves as an excellent base surface for parameterizing point clouds or points within subsets of a point cloud to obtain the correct parameters for each point.

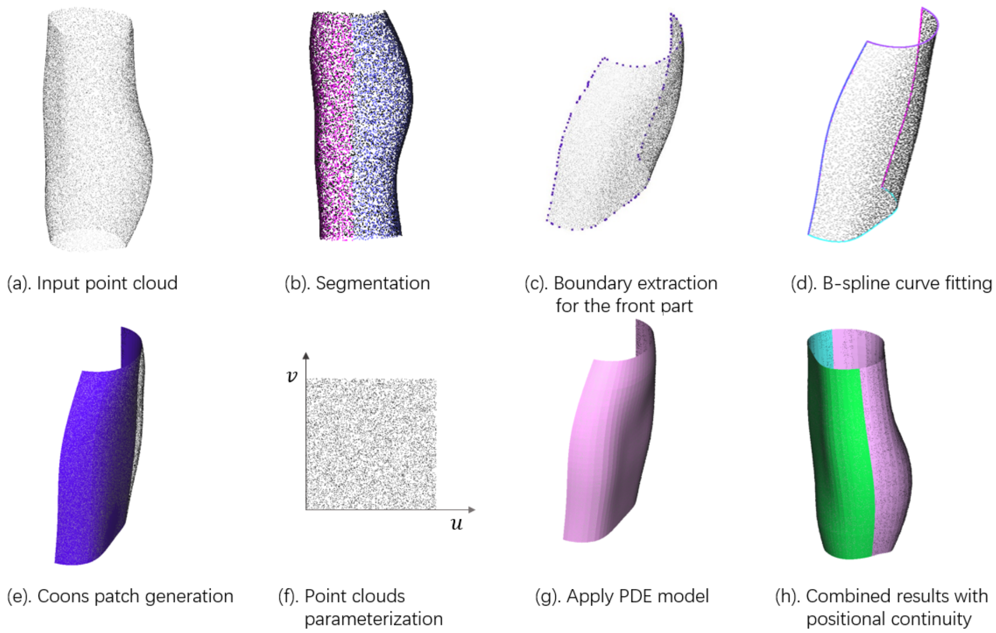
Implicit representation has become a very popular choice, especially as deep learning techniques have been widely integrated into the pipeline for reconstructing 3D surfaces. Implicit representation is powerful because it can theoretically represent arbitrarily complex shapes and is highly compatible with deep learning techniques for surface reconstruction. Compared to explicit representation, it is much easier to determine whether a point lies inside or outside the surface. However, generating a point from an implicit representation and altering the shape of surfaces is not straightforward. Additionally, implicit surfaces need to be converted into explicit representations like polygon surfaces or voxels for display in the digital world. Typical examples of implicit representation include level sets [15], distance functions [16], algebraic surfaces [17], and Constructive Solid Geometry [18]. Among distance function representations, Poisson surface reconstruction [19] is a classic method, which solves for an approximate indicator function of the inferred solid, whose gradient best matches the normal of the point set. The reconstructed surface is obtained by extracting an appropriate isosurface of the indicator function using adaptations of the Marching Cubes algorithm. Implicit B-spline surfaces have also been investigated and used for reconstruction [20].

In addition to the aforementioned methods for reconstructing 3D surfaces from 3D data points, PDE surfaces have also been adopted using both explicit and implicit representations. For example, Zhu et al. [21] used 4-sided patches defined by an analytical resolution to a fourth-order PDE for reconstructing parametric surfaces from 3D point data, notable for its efficiency and accuracy. Ugail and Kirmani employed an elliptic PDE equation and analytically solved it by utilizing a set of curves as the boundary condition, resulting in a highly efficient approach [22]. Rodrigues et al. explicitly resolved a Laplace equation and used it for compressing and reconstructing 3D data [23]. In the context of 3D surface reconstruction using implicit PDE models, many approaches have also been proposed [24, 43, 25]. For a more detailed introduction to these methods, please refer to [21].

While implicit PDE-based surface reconstruction methods are powerful in reconstructing various complex shapes from point clouds, they involve heavy numerical calculations and require high storage costs. Although existing explicit PDE-based surface reconstruction methods overcome these issues, they cannot guarantee that reconstructed PDE patches are seamlessly connected, and parameterizing point clouds remains challenging. This paper introduces a new surface reconstruction method using explicit PDE-based deformations and bilinearly blended Coons patches to address these challenges.

3. Pipeline

The pipeline of our proposed method is illustrated in Fig. 1. Specifically, given a point cloud data shown in Fig. 1(a), we first segment it into multiple subsets as shown in Fig. 1(b). For each subset, we then extract its boundaries and corner points using an existing method, as depicted in Fig. 1(c). The points on each extracted boundary are fitted with a B-spline curve that passes through the two endpoints of the boundary, as shown in Fig. 1(d). Next, we use these boundary curves to generate a bilinearly blended Coons patch, illustrated in Fig. 1(e). This Coons patch serves as an excellent base surface for the parameterization of the points within the subset, allowing us to obtain their (u, v) parameters, as depicted in Fig. 1(f). Our proposed PDE patch is then applied to deform the interior part of the Coons patch, ensuring that the combined parametric surface fits the points in the subset well without altering the boundaries of the Coons patch, as shown in Fig. 1(g). Since adjacent parametric surface patches share the same boundary, they are seamlessly connected with positional continuity. The final result, shown in Fig. 1(h), demonstrates that the adjacent patches are seamlessly connected, eliminating the need for postprocessing required by other methods.



(a). Input point cloud

(b). Segmentation

(c). Boundary extraction for the front part

(d). B-spline curve fitting

(e). Coons patch generation

(f). Point clouds parameterization

(g). Apply PDE model

(h). Combined results with positional continuity

Figure 1: Pipeline of our proposed method.

4. Proposed approach

Our proposed approach is detailed in this section. Firstly, the method for obtaining multiple subsets using point cloud segmentation techniques and extracting the boundaries of each segmented subset will be discussed in Section 4.1. Secondly, we explain how to fit B-spline curves to the extracted boundary points in Section 4.2. Thirdly, the definition of the Coons patch from four parametric curves is provided, and a bilinearly blended Coons patch from the four B-spline boundary curves is generated in Section 4.3. Following this, Section 4.4 covers the parameterization of the points within the subset using the generated Coons patch. Finally, we describe fitting our proposed PDE model to the offset between the Coons patch and the interior points of the subset. The combined PDE model and Coons patch fit the points very well, ensuring positional continuity between adjacent parametric patches that share the same boundary condition.

4.1. Segmentation and boundary extraction of 3D point data

Various approaches exist for segmenting 3D point data, including methods using attributes, 3D models, and edges, etc. For a comprehensive and

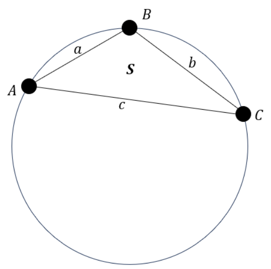


Figure 2: Curvature calculation of points on the boundary.

detailed introduction to these methods, please refer to [26]. For simplicity, this paper adopts a geometric primitive (3D models) method to segment an input point cloud into an appropriate number of subsets. To detect the boundary points of a given point set, many techniques have been proposed [27, 28, 29]. Specifically, Chen et al. [27] proposed an improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to detect point cloud boundaries, but their method mainly focuses on planar points. Mineo proposed a novel algorithm for point cloud boundary detection by calculating the local resolution of the point cloud with the aid of the K-nearest neighbour method [28]. Alpha-shape [29], a classical method for detecting the convex or concave regions of a given point set, can also be used for point cloud detection. Here, we adopt the technique in [29] as it meets our needs in most cases. In cases where the extracted boundary is not satisfactory, we will fine-tune the parameters of the boundary detection method to achieve better results. Specifically, if the extracted boundary points contain outliers or too few points, we will adjust the parameters of the alpha-shape method, as this method is dependent on both the parameters and the specific input. For the detected boundary points, we also need to identify the corner points, which will divide the boundary points into four or three segments (curves). To accomplish this, we first calculate the curvature of each point using the following formulation [30]:

$$k = (4 * S)/(a * b * c)$$

where S is the area formed by the point of interest (point B in Fig. 2) with its two adjacent points (points A and C), and a, b, and c are the lengths of AB, BC, and AC, respectively, as shown in Fig. 2. The top four or three curvature points will be treated as the corner points.

4.2. B-spline curve fitting

For the points on each boundary, we need to use a parametric curve to fit them, ensuring that the curve passes through the first and last points. While we could use a Bézier curve, it requires a higher degree for more complex shapes defined by the points in a subset. This is not desirable due to the computational expense and lack of local control. On the other hand, the B-spline curve is a better choice due to its flexibility and capability to fit the data without increasing the curve's degree. A B-spline curve is defined by the following expression:

$$\mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,k} \qquad\qquad t \in [0,1]$$

where $\mathbf{P}_i (i = 0, 1, 2, \ldots, n)$ are the control points, k is the order of the B-spline curve, and $B_{i,k}(t)$ are the basis functions, which are defined recursively as follows:

$$B_{i,1}(t) = \begin{cases} 1 & \text{for } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t) \qquad\qquad \text{if } k > 1$$

In the above equations, $\{t_i\}_{i=0}^{n+k}$ is the knot vector, a non-decreasing array of numbers from 0 to 1. To ensure the curve passes through its two control points at the ends, we set $t_0 = t_1 = \ldots = t_{k-1} = 0$ and $t_{n+k} = t_{n+k-1} = \ldots = t_{n+1} = 1$. The remaining values of the knot vector are uniformly distributed, as this approach is relatively simple and meets our needs. Having determined the appropriate parametric curve for fitting, the next step is to parameterize the points on each of the extracted boundaries. There are several methods available, such as Chord length, Centripetal, Foley, and Universal, etc. We adopt the Chord length method for parameterizing the points on each of the extracted boundaries, as it is one of the most widely used methods. Given the knot vector and the parameters obtained for the points on each of the extracted boundaries, the B-spline curves for these boundaries can be readily obtained through least squares fitting.

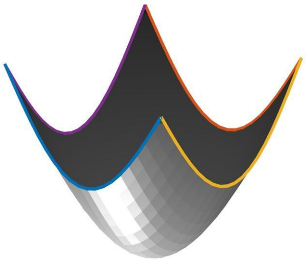


Figure 3: Bilinearly blended Coons patch from 4 boundary curves.

4.3. Bilinearly blended Coons patches

A bilinearly blended Coons patch is a parametric surface defined by four boundary curves, and it passes through these boundaries. Specifically, given four boundary curves $\mathbf{P}(u,0)$, $\mathbf{P}(u,1)$, $\mathbf{P}(0,v)$ and $\mathbf{P}(1,v)$ which are all parametric curves with the parametric variables u and v normally defined in the range $[0,1]$, the corresponding equation of the bilinearly blended Coons patch is given by the following equation [31]:

$$\mathbf{S}_c(u,v) = \begin{pmatrix} 1-u & u & 1 \end{pmatrix} \begin{pmatrix} -\mathbf{P}_{00} & -\mathbf{P}_{01} & \mathbf{P}(0,v) \\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & \mathbf{P}(1,v) \\ \mathbf{P}(u,0) & \mathbf{P}(u,1) & (0,0,0) \end{pmatrix} \begin{pmatrix} 1-v \\ v \\ 1 \end{pmatrix} \tag{1}$$

where $\mathbf{P}_{00}$, $\mathbf{P}_{01}$, $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$ are the four corner points where the boundary curves intersect, with $\mathbf{P}_{00}$ being the intersection of $\mathbf{P}(u,0)$ and $\mathbf{P}(0,v)$. Similarly, $\mathbf{P}_{01}$, $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$ are the remaining three corner points where the other pairs of adjacent boundary curves intersect. This is illustrated in Fig. 3.

4.4. Point cloud parameterization for 3D surface fitting

For unstructured 3D point data, the parameters associated with each point are unknown and not easily obtained. The process of obtaining suitable parameters for each point is called point cloud parameterization. Numerous techniques have been proposed to achieve this. One of the most widely used methods involves using a base surface, which can be a plane [32, 33], a sphere [34, 35], or a surface patch that approximates the shape of the 3D point data [36, 37]. In our approach, a bilinearly blended Coons patch serves as the base surface. Given the base surface and a point in a 3D point data, we need to find the closest point on the base surface and use this point to determine the parameter values u and v of the original point, which is a nonlinear problem. Many methods have been proposed to tackle this nonlinear problem, which can be roughly divided into five categories: Newton-Raphson method [38], subdividing method [39], solver methods [40], clipping method [41], and geometric method [42]. Each method has its advantages and disadvantages. After testing some methods from each category, we found that the geometric method best fits our needs due to its effectiveness and efficiency. We adopt the geometric method proposed in [42] to find the closest point and its associated parameters on a parametric surface to a given point. Specifically, given a parametric surface and a test point whose parameters are to be obtained, a normal curvature sphere of the surface is constructed. The radius and center of this sphere, along with the initially guessed parameters, are specified. Next, the intersection point between the line segment defined by the test point and the center of the normal curvature sphere is found. Lastly, to iteratively update the parameters of the test point, the iterative formula is derived using Taylor's expansion of the parametric surface. This method is independent of the initial parameter values and converges quickly. Since there may be tens of thousands of points in a subset, using the geometric method point by point is inefficient. To tackle this issue, we use parallel computing to improve the efficiency of point cloud parameterization, as parameterizing each point using the geometric method is independent of the others.

4.5. PDE deformation surfaces

PDE deformation surfaces are crucial in removing or minimizing the errors between the points in a segmented subset and the corresponding points on the bilinearly blended Coons patch. These surfaces can be derived from the particular solution to the PDE proposed below. The bending equation of an elastic thin plate is:

$$D(\frac{\partial^4 \mathbf{w}}{\partial x^4} + 2\frac{\partial^4 \mathbf{w}}{\partial x^2 \partial y^2} + \frac{\partial^4 \mathbf{w}}{\partial y^4}) = \mathbf{q_w} \tag{2}$$

where x and y are the position variables,$\mathbf{w}$ is the lateral displacement,$\mathbf{q_w}$ is the lateral force, and

$$D = \frac{Eh^3}{12(1-\mu^2)} \tag{3}$$

is called the flexural rigidity, which is determined by Young's modulus E, Poisson's ratio μ, and the thickness h of the elastic plate. If we replace the position variables x and y with the parametric parameters (u and v), and replace the lateral displacement with the position displacements x, y, z, Eq. (2) is changed into the following partial differential equations:

$$D(\frac{\partial^4 w}{\partial u^4} + 2\frac{\partial^4 w}{\partial u^2 \partial v^2} + \frac{\partial^4 w}{\partial v^4}) = q_w \\ (w = x, y, z) \tag{4}$$

For simplicity, we will not use Young's modulus E, Poisson's ratio μ, and the thickness h to determine the flexural rigidity D. Instead, we set its value to 1, i.e., D=1.

Since the purpose of using Eq (4) is to add deformations to the bilinearly blended Coons patch to remove or minimize the errors between the point clouds and the Coons patch, only the particular solution to Eq. (4) is necessary. Although the deformations may be complex, they can be mathematically represented with a Fourier series. To ensure positional continuity between reconstructed patches, the Fourier series can be expressed as the sine series. Considering these factors, the mathematical expression of the lateral force $\mathbf{q_w}$ are taken to be the following forms:

$$q_w = \sum_{m=1}^{M}\sum_{n=1}^{N} q_{wmn} sin(m\pi u) sin(n\pi v) \\ (w = x, y, z) \tag{5}$$

where M and N are hyper-parameters that can be adjusted to control the surface reconstruction errors. For example, when the shape of a point cloud is complex and the reconstruction error exceeds the specified tolerance, we can increase the values of M and N to reduce the disparity between the reconstructed surface and the point data within the specified threshold, which will be demonstrated in Section 5.

According to Eq. (5), the displacement w can be taken to be

$$w = \sum_{m=1}^{M}\sum_{n=1}^{N} w_{mn} sin(m\pi u) sin(n\pi v) \\ (w = x, y, z) \tag{6}$$

From Eq. (6), we obtain the fourth partial derivatives of w with respect to u, u and v, and v, respectively. They have the forms of:

$$\begin{aligned}\frac{\partial^4 w}{\partial u^4} &= \pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} m^4 w_{mn} sin(m\pi u) sin(n\pi v)\\ \frac{\partial^4 w}{\partial u^2 v^2} &= \pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} m^2 n^2 w_{mn} sin(m\pi u) sin(n\pi v)\\ \frac{\partial^4 w}{\partial v^4} &= \pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} n^4 w_{mn} sin(m\pi u) sin(n\pi v)\end{aligned} \tag{7}$$

Substituting the above Eq. (5) and Eq. (7) into Eq. (4), we obtain

$$\begin{aligned}&D[\pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} m^4 w_{mn} sin(m\pi u) sin(n\pi v) + 2\pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} m^2 n^2 w_{mn} sin(m\pi u) sin(n\pi v)\\ &+ \pi^4 \sum_{m=1}^{M}\sum_{n=1}^{N} n^4 w_{mn} sin(m\pi u) sin(n\pi v)] = \sum_{m=1}^{M}\sum_{n=1}^{N} q_{wmn} sin(m\pi u) sin(n\pi v)\end{aligned} \tag{8}$$

The above equation can be simplified as:

$$\sum_{m=1}^{M}\sum_{n=1}^{N} [D\pi^4 (m^4 + 2m^2 n^2 + n^4) w_{mn} - q_{wnm}] sin(m\pi u) sin(n\pi v) = 0 \tag{9}$$

From Eq. (9), we the following w_{mn}:

$$\begin{gathered} w_{mn} = \frac{q_{wmn}}{D\pi^4 (m^4 + 2m^2 n^2 + n^4)}\\ (w = x, y, z; m = 1, 2, 3, ..., M; n = 1, 2, 3, ..., N)\end{gathered} \tag{10}$$

By Substituting Eq. (10) into Eq. (6), the particular solution shown below can be obtained

$$\begin{gathered} w = \frac{1}{D\pi^4} \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{q_{wmn}}{(m^4 + 2m^2 n^2 + n^4)} sin(m\pi u) sin(n\pi v)\\ (w = x, y, z)\end{gathered} \tag{11}$$

Suppose the number of points in a given point cloud subset is I, then at the points $(u_i{,}v_i)$ $(i = 1, 2, 3, \dots, I)$, the errors between the points in the point cloud and the corresponding points obtained from the bilinearly blended Coons surface $S_c(u_i, v_i)$ are $\overline{w}_i$. To fit a parametric surface to the interior points, we minimize the following error function:

$$E_w = \sum_{i=1}^{I} [\overline{w}_i - \frac{1}{D\pi^4} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{q_{wmn}}{(m^4 + 2m^2n^2 + n^4)} sin(m\pi u_i) sin(n\pi v_i)]^2 \quad (12)$$

which can be solved using the least square method below:

$$\frac{\partial E_w}{\partial q_{wkl}} = 2 \sum_{i=1}^{I} [\overline{w}_i - \frac{1}{D\pi^4} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{q_{wmn}}{m^4 + 2m^2n^2 + n^4} sin(m\pi u_i) sin(n\pi v_i)][$$
$$- \frac{1}{D\pi^4} \frac{1}{(k^4 + 2k^2l^2 + l^4)} sin(k\pi u_i) sin(l\pi v_i)] = 0$$
$$(w = x, y, z; k = 1, 2, 3, \dots, M; l = 1, 2, 3, \dots, N) \quad (13)$$

By solving the above $M \times N$ equations, we obtain q_{wmn} (where $w = x, y, z$; $m = 1, 2, 3, \dots, M$; $n = 1, 2, 3, \dots, N$). Substituting these values into Eq. (11), we derive the PDE patch $\mathbf{S}PDE(u, v)$, which is expressed as $\mathbf{S}_{PDE}(u, v) = [x(u, v), y(u, v), z(u, v)]^T$. Finally, the reconstructed parametric surface $\mathbf{S}(u, v)$ is:

$$\mathbf{S}(u, v) = \mathbf{S}_c(u, v) + \mathbf{S}_{PDE}(u, v) \quad (14)$$

To assess the reconstruction accuracy of our proposed approach, we compute both the average error and the maximum error between the reconstructed parametric surface and the original I points $(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_I)$ in the point cloud subset using the following expression:

$$\begin{aligned} Err_{Mean} &= \frac{1}{I} \sum_{i=1}^{I} |\mathbf{P}_i - \mathbf{S}(u_i, v_i)| \\ Err_{Max} &= \max_{i=1,2,\dots,I} |\mathbf{P}_i - \mathbf{S}(u_i, v_i)| \end{aligned} \quad (15)$$

Notice that when the product $M \times N$ of the hyper-parameters M and N is equal to the number I of the points in the point cloud subset, the reconstruction error is removed, i.e., the reconstruction error is zero.

5. Results

To validate the effectiveness of our proposed method, we tested it on various datasets, including structured point clouds, unstructured point clouds of varying complexities, and data with different levels of noise.

5.1. Surface reconstruction from structured point clouds

There are mainly two types of point clouds: structured and unstructured. In structured point clouds, the relationship between points is known, whereas in unstructured point clouds, this relationship is unknown and more complex. To obtain a structured point cloud, we uniformly sample points on a parametric Bézier surface, knowing the position of each point and its corresponding parameters (u_i, v_i). The point cloud is shown in Fig. 4(a), with fewer points sampled for clarity.

The parametric equations of the four boundaries can be easily derived by fixing values for u as 0 and 1 and for v as 0 and 1 in the parametric equation of the given Bézier surface. From these four boundaries, we construct a bilinearly blended Coons patch based on Eq. (1), as shown in Fig. 4(b). The figure illustrates that while the bilinearly blended Coons patch passes through the four boundaries, it does not fit the interior points well, with a mean reconstruction error of 0.4343. To reduce this error, we apply the particular solution of Eq. (11) to the PDE in Eq. (4), adding deformations to the bilinearly blended Coons patch. With hyper-parameters $M = 5$ and $N = 5$, the resulting reconstruction surface, which combines the PDE patch and the bilinearly blended Coons patch, is shown in Fig. 4(c). As illustrated, the combined parametric surface fits all the points very well, significantly reducing the mean reconstruction error to 0.0038.

5.2. Surface reconstruction from unstructured point clouds

We first test our method with an unstructured point cloud of one patch with four sides. The point cloud is shown in black in Fig. 5(a), representing the front part of a skirt model. To obtain the Coons patch, we first identify the four boundaries of the point cloud, ensuring that one endpoint of adjacent boundaries is the same point, serving as the corner point of the Coons patch. The detected boundary points are shown in blue in Fig. 5(a), and the fitting B-spline curves are shown in Fig. 5(b). After obtaining the B-spline equations for the four boundaries, the bilinearly blended Coons patch is generated from these boundaries using Eq. (1), as shown in Fig. 5(c).

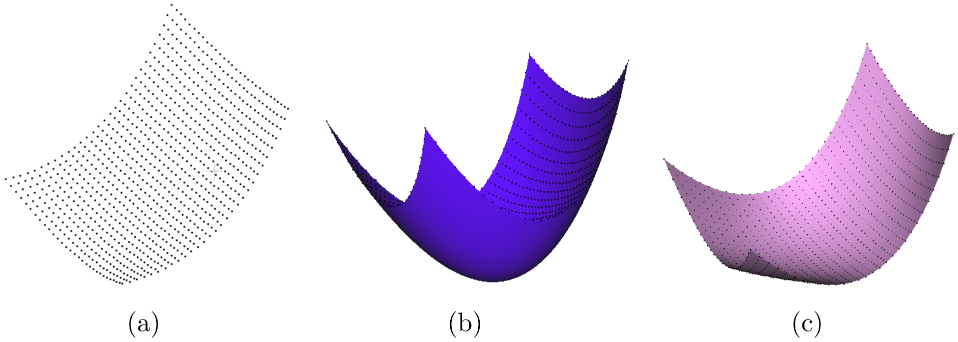


Figure 4: Reconstruction of the surface from constructed point clouds: (a).Input point clouds. (b).Generated Coons patch with an average error of 0.4343. (c).Final parametric surface with an average error of 0.0038

The Coons patch does not fit the point cloud well initially. To improve the fitting, we calculate the distances between the points in the point cloud and the corresponding points on the Coons patch. The PDE deformation surface from Eq. (11) is then used to compensate for this discrepancy. The final result is shown in Fig. 5(d), demonstrating that the reconstructed surface fits the point cloud very well.

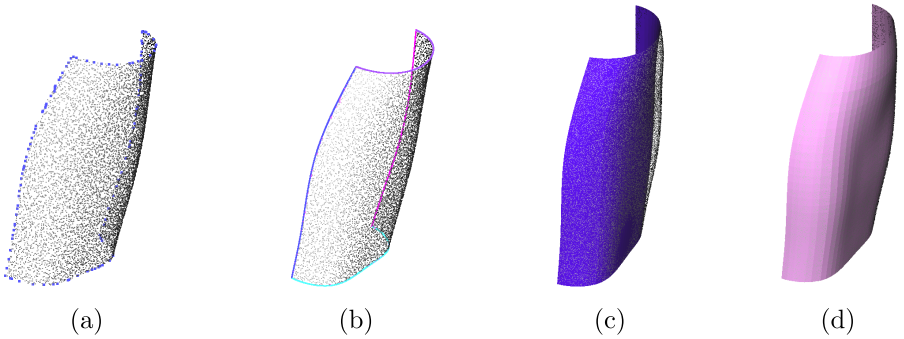


Figure 5: Reconstruction of the surface from unconstructed point clouds of the front part of a skirt model:(a).Input point clouds and extracted boundaries. (b).Reconstructed B-spline curves. (c).Generated Coons patch. (d).Final parametric surface

To further demonstrate the effectiveness and controllability of our method, we apply it to reconstruct the back part of the skirt model using the same procedure. Fig. 6(a), 6(b), and 6(c) show the original input point cloud, the reconstructed bilinearly blended Coons patch, and the final parametric surface, respectively. As expected, the Coons patch does not fit the point cloud well initially, but by adding our proposed PDE deformation surface, the final reconstruction result is satisfactory. As discussed in Section 4.2, the reconstruction error from our method is controllable. This can be achieved by adjusting the values of M and N. The reconstruction surface shown in Fig. 6(c) uses $M = 5$ and $N = 5$, with corresponding mean and maximum errors of 0.0074 and 0.0423, respectively. To reduce the reconstruction error, we can increase the values of M and N. Fig. 6(d) shows the result with $M = 10$ and $N = 10$, where the mean and maximum errors are reduced to 0.0041 and 0.0389, respectively.

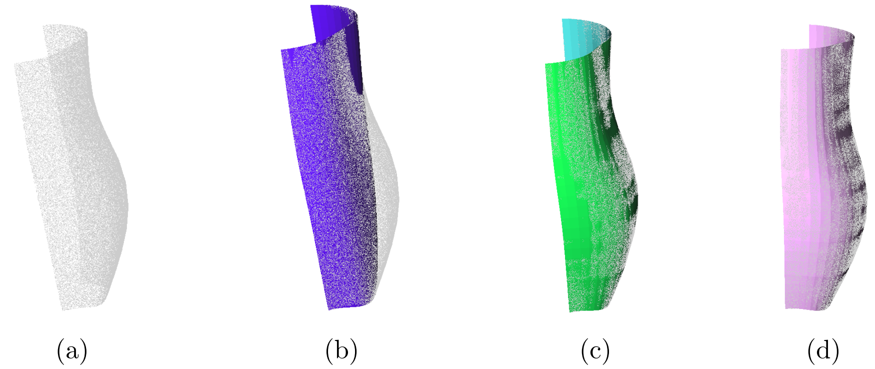


Figure 6: Reconstruction of the surface from unconstructed point clouds of the back part of a skirt model:(a).Input point clouds. (b).Generated Coons patch. (c).Reconstructed parametric surface with $M = 5, N = 5$. (d).Reconstructed parametric surface with $M = 10, N = 10$.

The point clouds shown in Fig. 5(a) and 6(a) are the front and back parts of the point cloud shown in Fig. 7(a), respectively. Their reconstructed surfaces are shown in Fig. 7(b). As depicted, the reconstruction surfaces fit the point cloud very well. Moreover, the two reconstructed surfaces are seamlessly connected, requiring no post-processing, unlike the PDE-based reconstruction method presented in [21].

We further test our method with examples shown in Fig. 8 and 9. In Fig. 8, we use one patch to reconstruct the flag shape from the point cloud shown in Fig. 8(a), with the reconstructed surface shown in Fig. 8(b). In Fig. 9, we segment the point cloud in Fig. 9(a) into 12 subsets, using a single parametric surface to approximate the 3D points in each subset. These reconstructed surfaces are seamlessly connected to form the pot shape shown in Fig. 9(b).

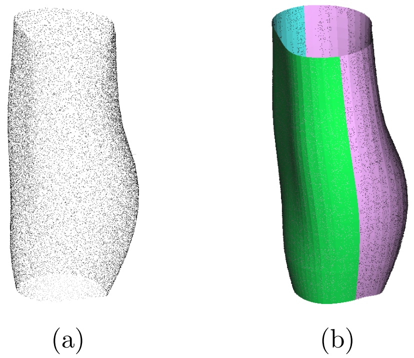


Figure 7: Reconstruction of the surface from unconstructed point clouds of a skirt model:(a).Input point clouds. (b).Reconstructed parametric surface after combination.

These examples further illustrate the capability of our method to effectively generate parametric surfaces that closely match various datasets.

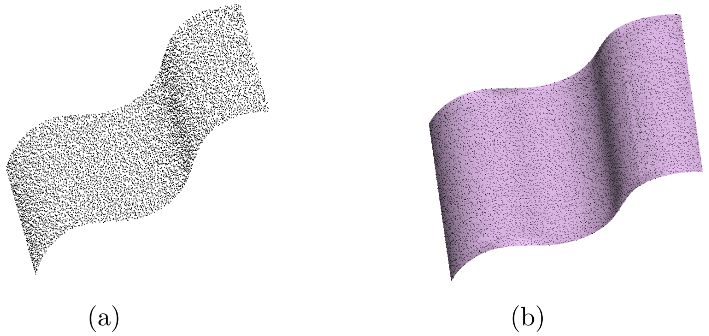


Figure 8: Reconstruction of the surface from unconstructed point data of a flag model:(a).Input point clouds. (b).Reconstructed parametric surface.

The above examples demonstrate our method's effectiveness in reconstructing four-sided patches from 3D point data. However, not all point clouds have four boundaries or can be segmented into subsets with four boundaries. Some point clouds may have three boundaries. In such cases, one boundary of the reconstructed patch degenerates to a single point, and our proposed method can still handle such datasets. Fig. 10(a) shows such a point cloud, segmented into eight subsets, each with three boundaries, as shown in Fig. 10(b). Our approach reconstructs the shape from the points within each subset, with the final result shown in Fig. 10(c), demonstrating that the reconstructed surfaces fit the point cloud very well.

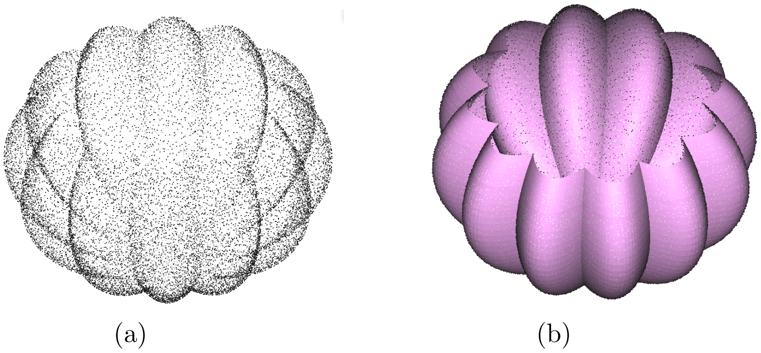


Figure 9: Reconstruction of the surface from unconstructed point data of a pot model:(a).Input point clouds. (b).Reconstructed parametric surface.

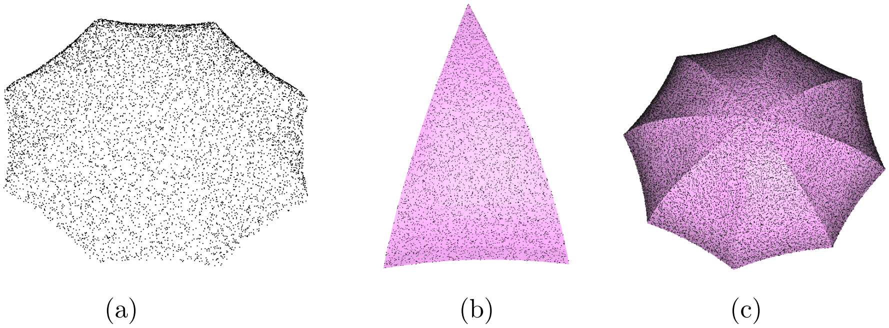


Figure 10: Surface reconstruction from unconstructed point clouds of an umbrella model:(a).Input point clouds. (b).Reconstructed parametric surface for a subset with three boundaries. (c).Final parametric surface after combination

5.3. The impact of hyper-parameters

To further investigate how different values of the hyper-parameters M and N affect reconstruction errors, we set $M = N = 5$ and $M = N = 10$ for surface reconstruction from the point clouds shown in Fig. 4, 5, 6, 8, 9, and 10. The mean errors and maximum errors are presented in Table 1 for $M = N = 5$ and in Table 2 for $M = N = 10$. Comparing the mean errors and maximum errors given in the tables, we can conclude that both the mean error and maximum error can be reduced by increasing the values of M and N, which further demonstrates the effectiveness and controllability of our suggested approach To provide a clearer comparison, we also plot the mean and maximum errors for different degrees of freedom ($M \times N$) across all datasets. Fig. 11 illustrates the mean fitting errors relative to the degrees of freedom, showing a consistent decrease in mean errors as the degrees of freedom increase in all cases. For the maximum errors, the trend is less clear when plotted together due to varying scales, so we present them in three separate subplots in Fig. 12. In all cases, the maximum errors decrease as the degrees of freedom increase. These figures demonstrate the flexibility and controllability of our method.

In Section 4.5, we take the value of D as 1 for simplicity and use it in Eq. (13) for linear least squares fitting. It is important to note that the value of D does not affect the reconstructed results or the fitting errors. This is because in Eq. (13), D can be incorporated into the expression within the sum, allowing q_{wmn}/D to be treated as a new variable. Therefore, altering D would change q_{wmn} correspondingly, leaving their ratio unchanged. To illustrate this, we varied the value of D in Eq. (13) for the least squares fitting multiple times, selecting random values between 10^{-3} and 10^{3}. Each time, both the average error and the maximum error remained constant, and the value of q_{wmn}/D were unchanged.

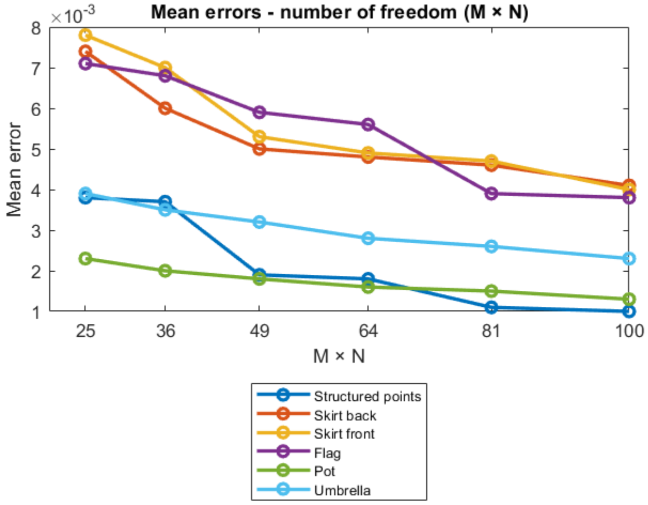


Figure 11: Mean errors respect to $M \times N$ for all the datasets

5.4. Surface reconstruction from point clouds with various levels of noise

To investigate the robustness of our method to various levels of noise (defined as l), we begin with the constructed point sets sampled from a Bézier surface that is used in Section 5.1, which is noise-free and can be

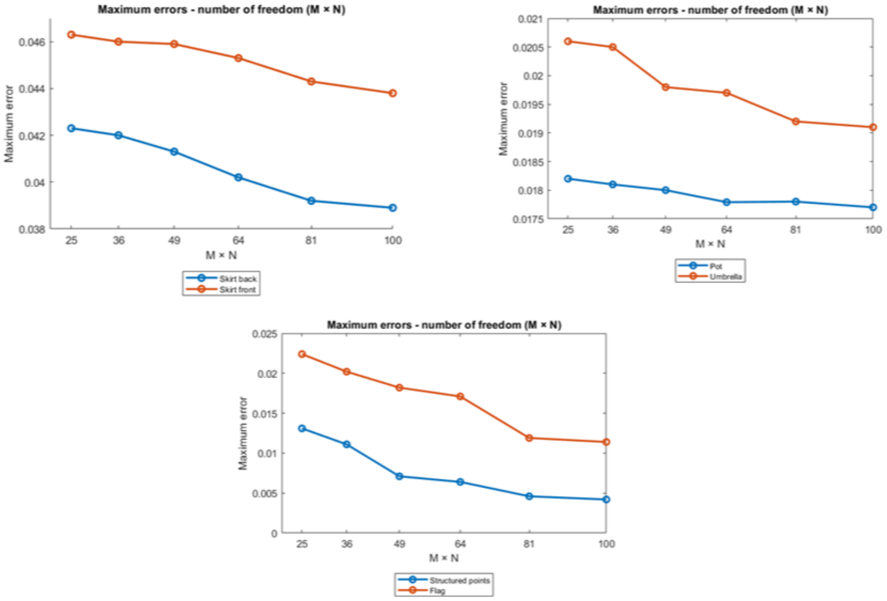


Figure 12: Mean errors respect to $M \times N$ for all the datasets

	Figure 4: Structured points	Figure 5: Front part of skirt	Figure 6: Back part of skirt	Figure 8: Flag	Figure 9: Pot	Figure 10: Umbrella
Mean error	0.0038	0.0078	0.0074	0.0071	0.0023	0.0039
Max error	0.0131	0.0463	0.0423	0.0224	0.0182	0.0206

Table 1: Reconstruction errors when $M = N = 5$.

used as the ground truth for calculating the fitting errors. Similar to other methods to test the performance on point clouds with various levels of noise [44, 45], which reconstruct B-Spline surfaces for sharp feature preservation and curvature estimation respectively, we define the noise level as follows: We first compute the smallest bounding box that encapsulates the noise-free point cloud, and the diagonal length of the bounding box is calculated and defined as d. Then each point is displaced in a random direction with a random magnitude, and the magnitude is drawn from a Gaussian distribution

	Figure 4: Structured points	Figure 5: Front part of skirt	Figure 6: Back part of skirt	Figure 8: Flag	Figure 9: Pot	Figure 10: Umbrella
Mean error	0.0010	0.0040	0.0041	0.0038	0.0013	0.0023
Max error	0.0042	0.0438	0.0389	0.0114	0.0177	0.0191

Table 2: Reconstruction errors when $M = N = 10$.

with zero mean and a certain standard deviation, which is calculated by $(l * d)\%$.

Given the noise-free point set, as shown in Fig.13, we set $l = 0.5$. The obtained noisy data using the aforementioned steps with noise levels $l = 0.5$, $l = 1.0$, and $l = 1.5$, together with the detected boundaries, are shown in the first row of Fig. 14. The base surfaces from the detected boundaries are shown in the second row. The last two rows show the reconstructed results and those superimposed on the structured point cloud that is free of noise. Notice we set $M = N = 10$ in this section. As we can see, our method gives a relatively good result when l is no more than 1.0. As l gets larger, the quality of the result decreases. However, it is important to note that the first step of the general pipeline for parametric surface reconstruction from noisy data involves preprocessing, which filters noise and outliers or adds

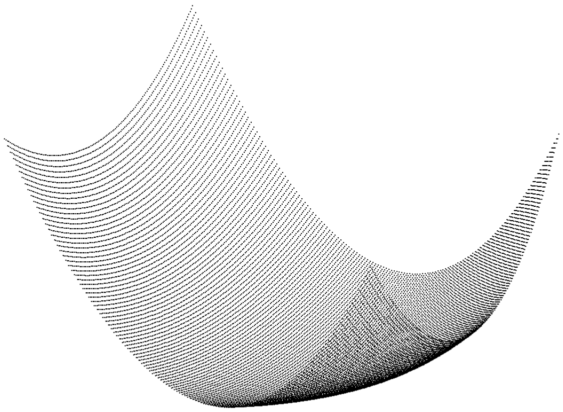


Figure 13: Noise free point cloud

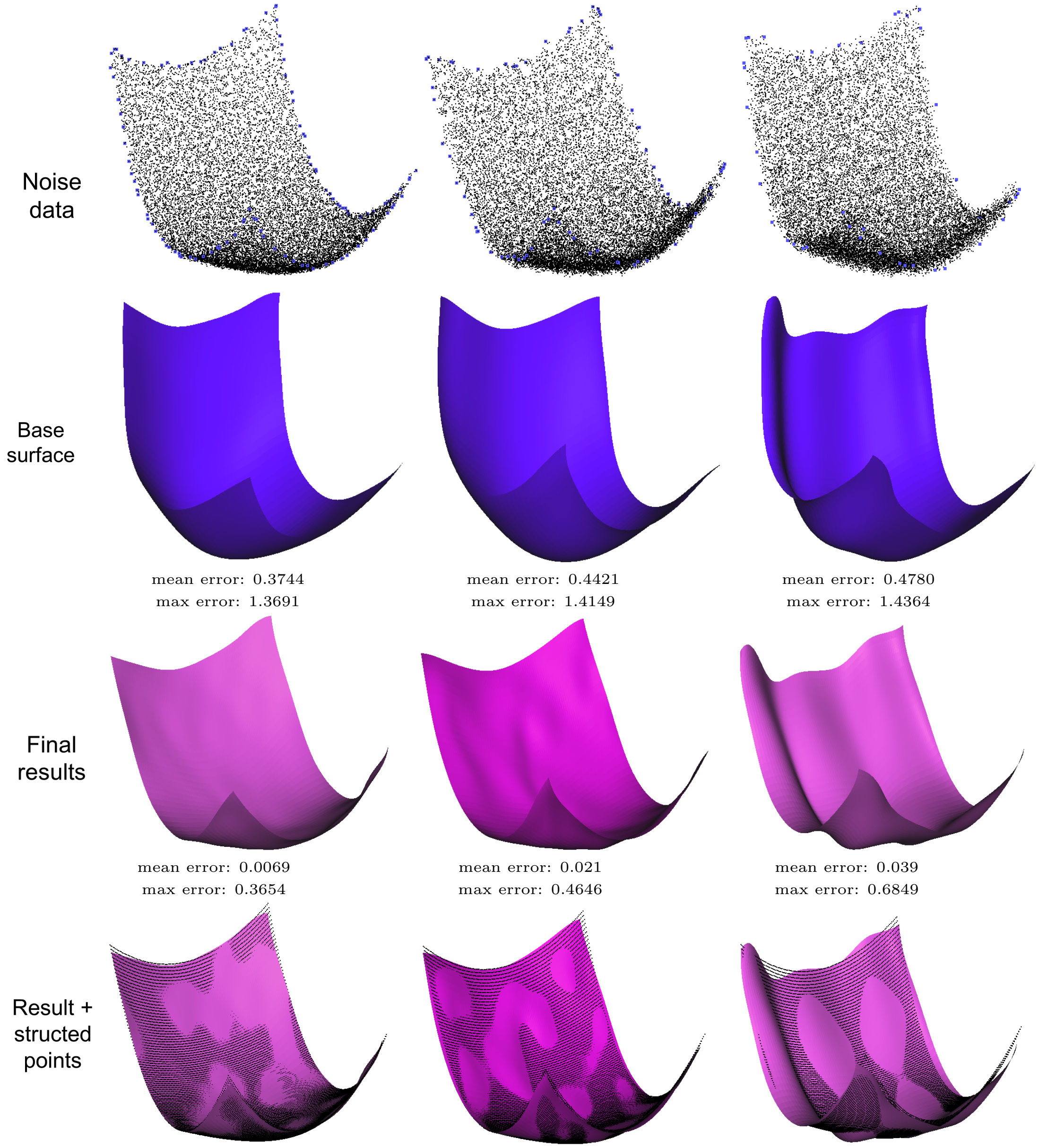
points to make the data complete. Since this section aims to investigate the robustness of our method to various levels of noise, preprocessing is not carried out to denoise the data. As the level of noise increases, the detected boundary degrades, and the generated base surface also deteriorates, as seen in the third row when $l = 1.5$. This affects the parameterization of points and the subsequent fitting process, thus impacting the final result. To illustrate, we use the boundary of the original structured point cloud to generate the base surface shown in Fig. 15(a), which is used for the noisy data ($l = 1.5$) parameterization, and the final result is shown in Fig. 15(b). We can see a better result can be obtained if a better base surface is generated, which can be achieved when the data is processed to filter the noise.

Finally, we plot the fitting errors along the way in all cases, as shown in Fig. 16. Our PDE model consistently reduces the error between the base surface (Coons patch) and the structured points in all cases.

6. Conclusion

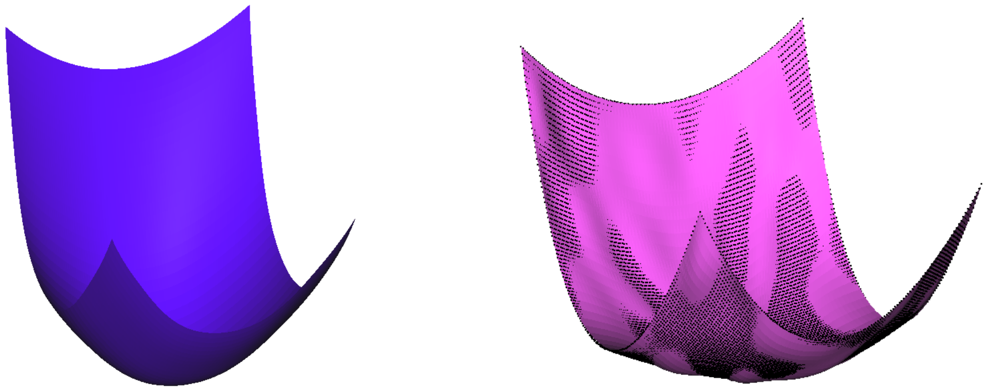
In this paper, we have developed a new physics-based method using PDE deformation surfaces and bilinearly blended Coons patches for parametric surface reconstruction from point clouds. Our approach involves modifying the governing equation of elastic bending of thin plates to obtain a partial differential equation (PDE) that incorporates numerous unknown constants in the lateral force, allowing us to minimize or eliminate surface reconstruction errors. We derived the particular solution of this PDE and used it to generate a PDE deformation surface. This surface is then superimposed on the bilinearly blended Coons surface, which is derived by interpolating the four boundaries of a point cloud or one of its subsets, to create the reconstruction surface. Our experimental results demonstrate the effectiveness of this surface reconstruction method, highlighting several advantages: Seamless connection between reconstructed surface patches; readily controllable reconstruction errors; and easy and proper parameterization of point data.

In theory, any surface type can be reconstructed by adjusting the number of surface patches. For complex input point data shapes, increasing the number of subsets through more segmentation is beneficial. However, automatic and optimal segmentation of point clouds remains a challenging task. In future work, we will develop advanced methods to achieve this. Given its power, deep learning has been utilized for segmenting point clouds into multiple instances, semantic regions, or parts. We plan to explore incorporating deep



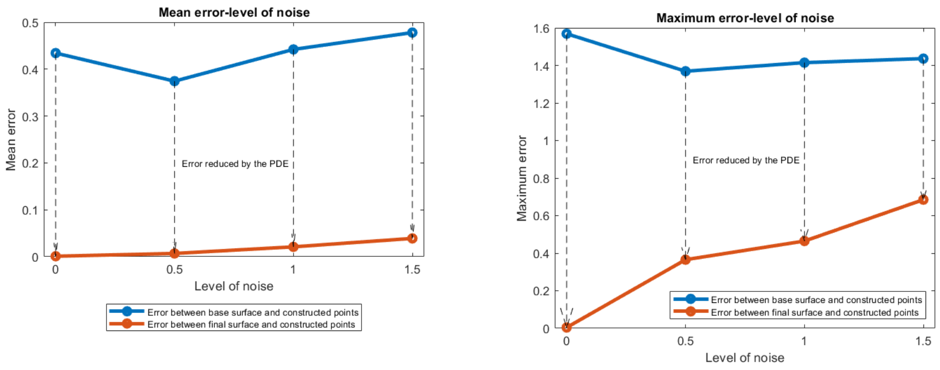
(a) $l = 0.5$ (b) $l = 1.0$ (c) $l = 1.5$

Figure 14: Reconstruction of the surface from unconstructed point data with $l = 0.5$(first column), 1.0(second column), 1.5(third column).
Row 1:Noisy point clouds; Row 2: Base surface; Row 3: Reconstructed parametric surface; Row 4: Reconstructed parametric surface with structured point cloud.



(a) (b)

Figure 15: Case $l = 1.5$: (a).Generated base surface from the boundary of the structured points; (b).Reconstructed parametric surface



(a) (b)

Figure 16: Errors between Coons patch with reconstructed surface and the constructed point clouds (a).Mean errors; (b).Maximum errors

learning into our pipeline to achieve automatic and optimal segmentation of point clouds.
Additionally, our current investigation focuses on surface reconstruction using four-sided and three-sided PDE deformation surfaces and bilinearly blended Coons patches. Future work will extend to parametric surface reconstruction from point clouds with only two boundaries. In such cases, we will propose a new PDE deformation surface, and the bilinearly blended Coons patch can be replaced with a loft surface. We will also investigate higher-order continuity for parametric surface reconstruction from point clouds, which is a more challenging task.

Acknowledgements

This research is supported by the PDE-GIR project, which has received funding from the European Union Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 778035. Zaiping Zhu is also sponsored by the China Scholarship Council.

References

[1] Farin, Gerald and Hoschek, Josef and Kim, M-S,Handbook of computer aided geometric design, Elsevier, (2002).

[2] Raja, Vinesh and Fernandes, Kiran J,Reverse engineering: an industrial perspective, Springer Science & Business Media, (2007).

[3] Gomes, Leonardo and Bellon, Olga Regina Pereira and Silva, Luciano, 3D reconstruction methods for digital preservation of cultural heritage: A survey, Pattern Recognition Letters, 50 (2014), pp. 3-14.

[4] Sulzer, Raphael and Landrieu, Loic and Marlet, Renaud and Vallet, Bruno, A Survey and Benchmark of Automatic Surface Reconstruction from Point Clouds, arXiv preprint arXiv:2301.13656, Elsevier, (2023).

[5] Berger, Matthew and Tagliasacchi, Andrea and Seversky, Lee M and Alliez, Pierre and Guennebaud, Gael and Levine, Joshua A and Sharf, Andrei and Silva, Claudio T, A survey of surface reconstruction from point clouds, Computer graphics forum, Wiley Online Library, 36(1), (2017), pp. 301-329.

[6] Lim, Seng Poh and Haron, Habibollah, Surface reconstruction techniques: a review, Artificial Intelligence Review, Springer, 42, (2014), pp. 59-78.

[7] Berger, Matthew and Tagliasacchi, Andrea and Seversky, Lee M and Alliez, Pierre and Levine, Joshua A and Sharf, Andrei and Silva, Claudio T, State of the art in surface reconstruction from point clouds, 35th Annual Conference of the European Association for Computer Graphics, Eurographics 2014-State of the Art Reports, CONF, The Eurographics Association, (2014).

[8] Shen, Yijun and Ren, Jieji and Huang, Nuodi and Zhang, Yang and Zhang, Xinquan and Zhu, Limin, Surface form inspection with contact coordinate measurement: a review, International Journal of Extreme Manufacturing, IOP Publishing, 5(2), (2023), Article 022006.

[9] Edelsbrunner, Herbert, Shape reconstruction with Delaunay complex, Latin American Symposium on Theoretical Informatics, Springer, (1998), pp. 119-132.

[10] Amenta, Nina and Bern, Marshall and Kamvysselis, Manolis, A new Voronoi-based surface reconstruction algorithm, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, (1998), pp. 415-421.

[11] Bołtuć, Agnieszka and Zieniuk, Eugeniusz, Parametric integral equation system (PIES) for solving problems with inclusions and non-homogeneous domains using Bézier surfaces, Journal of Computational Science, Elsevier, 51, (2021), Article 101343.

[12] Yuwen, Sun and Dongming, Guo and Zhenyuan, Jia and Weijun, Liu, B-spline surface reconstruction and direct slicing from point clouds, The International Journal of Advanced Manufacturing Technology, Springer, 27, (2006), pp. 918–924.

[13] Gálvez, Akemi and Iglesias, Andrés, Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points, Information Sciences, 174–192, Elsevier, 192, (2012).

[14] Zhu, Zaiping and Iglesias, Andres and You, Lihua and Zhang, Jian Jun, A review of 3D point clouds parameterization methods, International Conference on Computational Science, Springer, (2022), pp. 690-703.

[15] Fuhrmann, Simon and Goesele, Michael, Floating scale surface reconstruction, ACM Transactions on Graphics (ToG), ACM New York, NY, USA, 33(4), (2014), pp. 1-11.

[16] Boissonnat, Jean-Daniel and Cazals, Frédéric, Smooth surface reconstruction via natural neighbour interpolation of distance functions, Proceedings of the sixteenth annual symposium on Computational geometry, (2000), pp. 223-232.

[17] Yavartanoo, Mohsen and Chung, Jaeyoung and Neshatavar, Reyhaneh and Lee, Kyoung Mu, 3dias: 3d shape reconstruction with implicit algebraic surfaces, Proceedings of the IEEE/CVF International Conference on Computer Vision, (2021), pp. 12446-12455.

[18] Xiao, Jianxiong and Furukawa, Yasutaka, Reconstructing the world's museums, International journal of computer vision, Springer, 110, (2014), pp. 243-258.

[19] Kazhdan, Michael and Bolitho, Matthew and Hoppe, Hugues, Poisson surface reconstruction, Proceedings of the fourth Eurographics symposium on Geometry processing, 7(4), (2006).

[20] Rouhani, Mohammad and Sappa, Angel D and Boyer, Edmond, Implicit B-spline surface reconstruction, IEEE transactions on image processing, IEEE, 24(1), (2014), pp. 22-32.

[21] Zhu, Zaiping and Zheng, Anzong and Iglesias, Andrés and Wang, Shuangbu and Xia, Yu and Chaudhry, Ehtzaz and You, Lihua and Zhang, Jianjun, PDE patch-based surface reconstruction from point clouds, Journal of Computational Science, Elsevier, 61, (2022), Article 101647.

[22] Ugail, Hassan and Kirmani, Syed, Method of surface reconstruction using partial differential equations, Proceedings of the 10th WSEAS International Conference on Computers, Athens, Greece, 13-15, (2006).

[23] Rodrigues, Marcos and Osman, Abdusslam and Robinson, Alan, Partial differential equations for 3D data compression and reconstruction, ADSA Advances in Dynamical Systems and Applications, Research India Publications, 8(2), (2013), pp. 303-315.

[24] Duan, Ye and Yang, Liu and Qin, Hong and Samaras, Dimitris, Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces, Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III 8, Springer, (2004), pp. 238-251.

[25] Linz, Christian and Goldlücke, Bastian and Magnor, Marcus, A point-based approach to PDE-based surface reconstruction, Pattern Recognition: 28th DAGM Symposium, Berlin, Germany, September 12-14, 2006. Proceedings 28, (2006), pp. 729-738.

[26] Nguyen, Anh and Le, Bac, 3D point cloud segmentation: A survey, 2013 6th IEEE conference on robotics, automation and mechatronics (RAM), IEEE, (2013), pp. 225-230.

[27] Chen, Hui and Liang, Man and Liu, Wanquan and Wang, Weina and Liu, Peter Xiaoping, An approach to boundary detection for 3D point clouds based on DBSCAN clustering, Pattern Recognition, 108431, Elsevier, 124, (2022).

[28] Mineo, Carmelo and Pierce, Stephen Gareth and Summan, Rahul, Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction, Journal of Computational Design and Engineering, Oxford University Press, 6(1), (2019), pp. 81-91.

[29] Edelsbrunner, Herbert, Alpha shapes-a survey, Tessellations in the sciences: Virtues, techniques and applications of geometric tilings, (2011).

[30] Belyaev, Alexander G, A note on invariant three-point curvature approximations (singularity theory and differential equations), RIMS Kōkyûroku, RIMS-Research Institute for Mathematical Sciences Kyoto University, 1111, (1999), pp. 157-164.

[31] Salomon, David, Curves and surfaces for computer graphics, Springer Science & Business Media, (2007).

[32] Floater, Michael S and Reimers, Martin, Meshless parameterization and surface reconstruction, Computer Aided Geometric Design, Elsevier, 18(2), (2001), pp. 77-92.

[33] Zhang, Lei and Liu, Ligang and Gotsman, Craig and Huang, Hua, Mesh reconstruction by meshless denoising and parameterization, Computers & Graphics, Elsevier, 34(3), (2010), pp. 198-208.

[34] Choi, Gary Pui-Tung and Ho, Kin Tat and Lui, Lok Ming, Spherical conformal parameterization of genus-0 point clouds for meshing, SIAM Journal on Imaging Sciences, SIAM, 9(4), (2016), pp. 1582-1618.

[35] Zwicker, Matthias and Gotsman, Craig, Meshing Point Clouds Using Spherical Parameterization, PBG, SIAM, (2004), pp. 173-180.

[36] Ma, Weiyin and Kruth, Jean-Pierre, Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces, Computer-Aided Design, Elsevier, 27(9), (1995), pp. 663-675.

[37] Pottmann, Helmut and Leopoldseder, Stefan and Hofer, Michael, Approximation with active B-spline curves and surfaces, 10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings, IEEE, (2002), pp. 8-25.

[38] Mortenson, Michael E, Geometric modeling, John Wiley & Sons, Inc., (1997).

[39] Johnson, David E and Cohen, Elaine, Distance extrema for spline models using tangent cones, Proceedings of Graphics Interface 2005, (2005), pp. 169-175.

[40] Elber, Gershon and Kim, Myung-Soo, Geometric constraint solver using multivariate rational spline functions, Proceedings of the sixth ACM symposium on Solid modeling and applications, (2001), pp. 1-10.

[41] Chen, Xiao-Diao and Yong, Jun-Hai and Wang, Guozhao and Paul, Jean-Claude and Xu, Gang, Computing the minimum distance between a point and a NURBS curve, Computer-Aided Design, 40(10-11), (2008), pp. 1051-1054.

[42] Li, Xiaowu and Wu, Zhinan and Pan, Feng and Liang, Juan and Zhang, Jiafeng and Hou, Linke and others, A Geometric Strategy Algorithm for Orthogonal Projection onto a Parametric Surface, Journal of Computer Science and Technology, 34(6), (2019), pp. 1279-1293.

[43] Franchini, Elena, Serena Morigi, and Fiorella Sgallari, Implicit shape reconstruction of unorganized points using PDE-based deformable 3D manifolds, Numerical Mathematics: Theory, Methods and Applications, 3(4), (2010), pp. 405-430.

[44] Yuwen, Sun and Dongming, Guo and Zhenyuan, Jia and Weijun, Liu, B-spline surface reconstruction and direct slicing from point clouds, The International Journal of Advanced Manufacturing Technology, 27, (2006), pp. 918-924.

[45] Khameneifar, Farbod and Ghorbani, Hamid, On the curvature estimation for noisy point cloud data via local quadric surface fitting, Comput.-Aided Des. Appl., 16(1), (2019), pp. 140-149.