HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Preprint Submitted to Elsevier

PDE patch-based surface reconstruction from point clouds

Zaiping Zhu[a,1], Anzong Zheng[b], Andrés Iglesias[c,d], Shuangbu Wang[a], Yu Xia[a], Ehtzaz Chaudhry[a], Lihua You[a] and Jianjun Zhang[a]

[a] *The National Center for Computer Animation, Bournemouth University, Fern Barrow, Poole, Dorset, BH12 5BB, United Kingdom*
[b] *Humain Ltd, 1 Hill Street, Belfast BT1 2LA, United Kingdom*
[c] *Department of Information Science, Faculty of Sciences, Toho University, 2-2-1 Miyama, 274-8510 Funabashi, Japan*
[d] *Department of Applied Mathematics and Computational Sciences, E.T.S.I. Caminos, Canales y Puertos, University of Cantabria, Avda. de los Castros, s/n, 39005 Santander, Spain*

ARTICLE INFO

Article history:
Received xx Month 20xx
Revised xx Month 20xx
Accepted xx Month 20xx
Available online xx Month 20xx

Keywords:
Shape reconstruction
point clouds
fourth-order partial differential equation
segmentation
point cloud parameterization

ABSTRACT

By extending the work published at ICCS 2021 [1], in this paper we propose a new method for using multiple explicit PDE surface patches to reconstruct complex 3D shapes from point clouds. Our proposed method includes segmenting a given point cloud into several subsets, parameterizing the points, and fitting one PDE patch to the parameterized points in each of the subsets to reconstruct 3D shapes. Several surface reconstruction examples are presented to demonstrate our proposed method. The comparison with polygon-based surface reconstruction shows fewer design variables and better quality of our method.

1. Introduction

Surface reconstruction from point clouds is an important element of reverse engineering, which has been widely adopted in many domains, such as automotive, aerospace, and ship building industries, robotics, biomedical engineering, cultural heritage, and many others [2, 3, 4]. Many kinds of representations have been proposed for reconstructed 3D shapes, and they mainly can be divided into two categories: explicit surfaces and implicit surfaces. Each category includes various subclasses. For explicit surfaces, Bézier, B-spline, and NURBS surfaces have been widely explored. Implicit surface representation like level set function has also been broadly researched. However, all of these methods have some disadvantages in common as they require big data storage and heavy geometry processing.

To overcome such shortcomings of existing representations, an explicit partial differential equation (PDE) based method for surface reconstruction from point clouds is proposed in this paper. Compared to other types of 3D representations, PDE surfaces have several advantages. First of all, they require a smaller storage capacity when representing the same 3D shape. Secondly, adjacent PDE-based surface patches can naturally preserve position, tangent and even curvature continuities when they share the same boundary conditions. Thirdly, a sculpting force can be applied and the parameters in a PDE can be adjusted to create complex 3D shapes.

This work is inspired by the process of manually creating 3D shapes, which usually applies the method of patch-based modelling. Techniques developed in point cloud data including boundary extraction and part segmentation have also motivated our work. Based on these ideas, we propose to reconstruct PDE surfaces from point clouds by segmenting a 3D point cloud into several subsets and fitting the points in each of the subsets with one explicit PDE patch. Because of the advantages of PDE surfaces, these patches can easily achieve and naturally maintain required continuities when proper boundary conditions are formulated from the points in each of the subsets. Such a patch-by-patch reconstruction method has remarkable benefits. Since it is difficult to reconstruct a complicated 3D shape with only a single PDE patch, combining multiple PDE patches enables any complicated 3D shape to be reconstructed. In addition, fitting a 3D point cloud patch by patch would improve the reconstruction efficiency as multiple segmented patches can be reconstructed simultaneously, making the process very suitable for parallelization.

The paper is organized as follows. Related work is reviewed in

E-mail address(es): s5319266@bournemouth.ac.uk (Z. Zhu), anzong89@gmail.com (A. Zheng), iglesias@unican.es (A. Iglesias), swang1@bournemouth.ac.uk (S. Wang), yxia@bournemouth.ac.uk (Y. Xia), echaudhry@bournemouth.ac.uk (E. Chaudhry), lyou@bournemouth.ac.uk (L. You), jzhang@bournemouth.ac.uk (J. Zhang).

[1] Corresponding author

https://doi.org/xx.xxxx/j.jocs.20xx.xx.xxx
1877-7503/¬© 20xx. Publishing services by Elsevier B.V.

Section 2. Our proposed surface reconstruction pipeline is summarized in Section 3. In Section 4, the explicit PDE model will be firstly explained in detail, followed by point cloud decomposition and parameterization. At the very last of Section 4, multiple PDE patches will be used to fit the point clouds representing some types of 3D shapes. The experimental results and their analysis, along with a comparison with a popular polygon-based surface reconstruction method, will be presented in Section 5. Finally, we will conclude the paper and outline some directions for future work in Section 6.

2. Related work

There are many research works reported in the literature to address the problem of surface reconstruction from point clouds. Comprehensive and detailed literature reviews can be found in [5, 6, 7]. From the perspective of surface representation, surface reconstruction can be roughly divided into explicit and implicit based representations. Parametric surfaces are one type of explicit surfaces. They include very popular surfaces in computer-aided design and computer graphics such as Bézier surfaces [8], B-spline surfaces, and NURBS surfaces [9], etc. In addition, triangulated surfaces also belong to explicit surfaces. Implicit surfaces include level set surfaces, Poisson surfaces, and many others. Another approach is based on the use of partial differential equations (PDEs) for surface reconstruction; see [10] for an overview on the field. In recent years, different machine learning methods, mostly based on neural networks and deep learning [11, 12], metaheuristic techniques such as genetic algorithms and nature-inspired optimization algorithms [13, 14], or combinations of both [15] have also been used for 3D reconstruction from point clouds. In this section, we will briefly review some of the recent works, which are closely related to this paper.

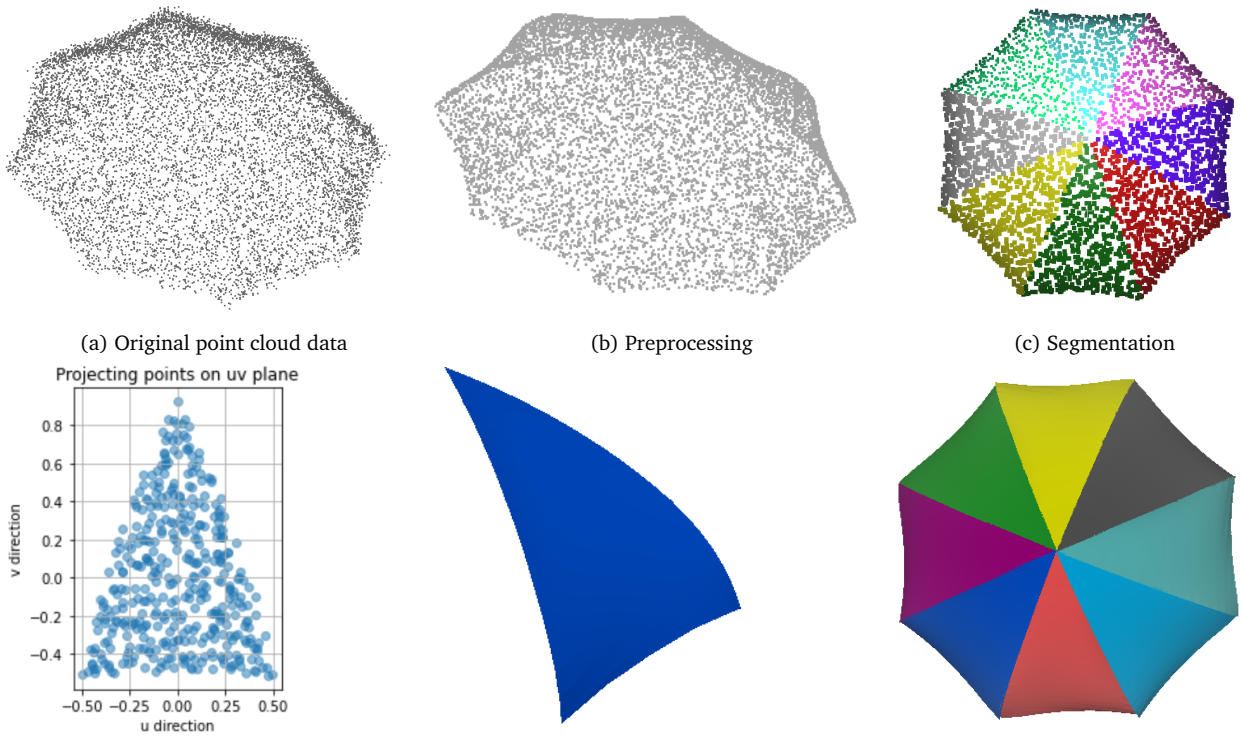
For parametric surfaces, Sharma et al. presented a parametric surface network for 3D point clouds and it was an end-to-end trainable model [16]. They firstly decomposed a 3D point cloud into several basic geometric primitives and B-spline patches, and the number of patches was automatically determined. Then they fit each segmented portion of the point cloud with a parametric patch. Post-processing was also necessary to better fit the point cloud, but producing seamless boundaries was still a challenge. Lee et al. proposed an approach to preserve sharp features for B-spline surface reconstruction from point cloud [17]. One challenge of B-spline surface reconstruction is that sharp features tend to be smooth out. Lee et al. took advantage of the curvature information of the B-spline patch and identified segments of sharp features, which could be preserved by adding more points to those regions. However, their method was only applied to a relatively simple shape of point clouds. The B-spline approach also contains some limitations. For instance, the rigid structure of the B-spline recurrence formula and its constraints on the order of the parameters and the basis functions can lead to models that have significantly larger number of degrees of freedom than it should be actually required. On the other hand, it is well known that B-splines are not well suited to model T-junctions, and hence, it is difficult to apply them to structural modeling involving such features. NURBS surfaces have been proposed to overcome the shortcomings of B-spline surfaces, as they are a generalization of the classical polynomial B-splines. Dimitrov et al. presented a new approach for NURBS surfaces fitting to unorganized point clouds [18]. They used intermediate B-spline surfaces to parameterize the points in a point cloud and the reconstructed NURBS surface was refined. A main challenge of fitting NURBS surfaces to point clouds lies in the proper parameterization of the input points. Several methods have been proposed to overcome such a difficulty. Bo et al. [19] fitted a parametric surface to a point cloud by minimizing the squared orthogonal distance from the surface to point cloud, thus it becomes a nonlinear least-squares minimization problem. They use an initial surface to approximate the point cloud, and for each point, they compute the closest point on the surface. They kept updating the surface by minimizing a quadratic function until the fitting error was smaller than a certain threshold.

Some works solved PDE explicitly and applied PDE to surface reconstruction. Ugail and Kirmani adopted an elliptic PDE equation and used a set of curves that represented the original shape as the boundary conditions to solve the PDE equation analytically [20]. Given that an analytical solution was obtained and applied, that approach was highly efficient. Rodrigues et al. solved a Laplace equation explicitly and applied it to 3D data compression and reconstruction [21].

Implicit surfaces, which use approximation techniques, have also been utilized to reconstruct smooth surfaces from point clouds. Poisson surface reconstruction is a typical example of implicit-based surface reconstruction techniques. Kazhdan et al. computed a 3D indicator function based on the observation that the indicator gradient was zero except at points near the object surface, and the value of the indicator function was set to 1 and 0 inside and outside the 3D object respectively [22]. The reconstructed surface was obtained by extracting a suitable isosurface. This technique can behave well even if there is noise in the point set. However, their method tends to over smooth the reconstructed surface. To address such problems, Kazhdan et al. modified the Poisson reconstruction algorithm by adding positional and derivative constraints [23]. For implicit PDE based 3D reconstruction, Duan et al. presented a PDE-based deformable surface to evolve its shape to reconstruct 3D surfaces, where the input data can be either volumetric data or unorganized point clouds and multi-view 2D images [24]. Franchini et al. proposed a method to reconstruct a 3D shape from an unorganized point set by adopting a PDE-based deformable surface [25]. Their method can also be applied to Boolean operation between various data. Linz et al. developed a technique to reconstruct 3D shapes from implicit PDE definition [26].

Apart from the techniques mentioned above for 3D surface reconstruction from point clouds, with the advancement of neural networks, several research works have applied neural networks to mesh reconstruction. At the object level, many methods learned priors from the dataset [27, 28, 29]. However, these methods do not generalize well to unseen objects during training since they learn priors at the object level. To make this technique more general, Badki et al. proposed to learn local shape (patch-level) prior of objects for mesh reconstruction [30]. The learned local shape features serve as a dictionary of local features, and they can be used to reconstruct 3D shapes even from unseen categories. Williams used a similar method to learn geometric prior for surface reconstruction [31]. They fit many neural networks in parallel while enforcing global consistency to compute an atlas of multiple mappings, which can reconstruct complex objects very well. However, gap or jagged areas may exist because of inconsistent mapping between different patches. Based on this observation, Deng et al. proposed a method to achieve global consistency between local mapping by incorporating consistency of local surface normal and minimizing a prescribed stitching error [32]. They also obtained adjacent patches that nearly coincide, a clear indication of good stitching.

Different from the above works, our proposed surface reconstruction method is based on a closed-form solution to a fourth-order partial differential equation and multiple PDE patches defined by the closed-form solution. With our proposed method, a point cloud with any complex shape can be segmented into some



(a) Original point cloud data (b) Preprocessing (c) Segmentation

(d) Parameterization (e) Patch fitting (f) Reconstructed surface

Fig. 1. The pipeline of surface reconstruction from point clouds.

subsets, each of which defines a simple shape. One PDE patch is used to fit the points in each of the segmented subsets, and all obtained PDE patches represent the reconstructed shape defined by the point cloud.

Since our proposed surface reconstruction method uses a closed-form solution to a fourth-order PDE, it is faster and more accurate than existing methods. Specifically, the existing methods focusing on implicit PDE based reconstruction are inefficient and inaccurate because they involve numerically solving a PDE. Although some other works have applied explicit PDE based methods to 3D reconstruction, we are unaware of any surface reconstruction methods, which use analytical 4-sided PDE patches defined by a closed-form solution to a partial differential equation. In comparison with the numerical implicit PDE-based surface reconstruction, our proposed method relies on analytical solutions, and hence it is more efficient and accurate. Compared with other explicit PDE surface reconstruction methods not using analytical 4-sided PDE patches, our proposed method is more powerful since it can reconstruct more complex shapes.

3. Surface reconstruction pipeline

In our previous work [1], we have used the closed-form solution to a fourth-order partial differential equation to define a 4-sided PDE patch. However, a single 4-sided PDE patch is incapable of reconstructing complicated shapes from point clouds. To tackle this problem, we will propose a new method of using multiple 4-sided PDE patches to reconstruct complicated shapes from point clouds .

Inspired by [33], the general pipeline of our proposed surface reconstruction method consists of six steps as shown in Fig. 1. For an input point cloud shown in Fig. 1(a), preprocessing is carried out to change the input point cloud into the one shown in Fig. 1(b). Then, the preprocessed point cloud in Fig. 1(b) is segmented into some subsets depicted in Fig. 1(c). Next, the points in each of the subsets are parameterized to obtain their parametric values in Fig. 1(d). After that, one 4-sided PDE patch in Fig. 1(e) is used to fit the points in each of the subsets (note that the 3-sided patch in that figure is a particular case of 4-sided patches). After fitting the points in all the subsets, a reconstructed 3D shape Fig. 1(f) consisting of multiple 4-sided PDE patches is obtained.

Point clouds can be obtained by using a scanner or just taking advantage of open-source data. As indicated by [34], a depth detection device can also be used to obtain the 3D point cloud data representing a certain object. The obtained point clouds need to be preprocessed to remove noise and to complete the data, usually through simplification and compression. There are a lot of techniques available. Han et al. reviewed the state-of-the-art techniques for filtering point sets [35]. When a point cloud is fitted to two or more surface patches, the point cloud should be segmented into some subsets whose number is the same as the number of the surface patches.

4. Method

In this section, we introduce our proposed PDE-based surface reconstruction method. First, we present a PDE model for surface reconstruction and obtain its closed-form solution in Section 4.1. Then, the segmentation of point clouds is introduced in Section 4.2. For parametric surface reconstruction, the points in each of the subsets must be parameterized. We discuss the parameterization of point clouds in Section 4.3. Having obtained the parametric values of parametric variables u and v for all the points in each of the segmented subsets, 4-sided PDE patches are used to fit these points. The method used to fit the parameterized points in each of the segmented subsets is described in Section 4.4. After the parameterized points in all segmented subsets have been fitted, a reconstructed 3D shape consisting of the reconstructed PDE

patches is obtained.

4.1. PDE model and closed-form solutions

A PDE patch is defined by the solution to a vector-valued partial differential equation. It can be regarded as a parametric surface involving two variables, denoted as u and v. Generally, both u and v are defined on the interval $[0,1]$, forming four boundaries of a 3D surface patch. Similar to our previous work in [1], the vector-valued partial differential equation we consider is:

$$\mathbf{a}_1\frac{\partial^4\mathbf{X}(u,v)}{\partial u^4}+\mathbf{a}_2\frac{\partial^4\mathbf{X}(u,v)}{\partial v^4}=\mathbf{F}(u,v) \tag{1}$$

where $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{X}(u,v)$, and $\mathbf{F}(u,v)$ are three-dimensional vectors containing x, y and z components. Note that in this paper vectors are denoted in bold. To simplify the notation in the paper, the following mathematical operations are defined:

$$\begin{array}{ll}\mathbf{f}''''(u)=\frac{d^4\mathbf{f}(u)}{du^4} & \mathbf{g}''''(v)=\frac{d^4\mathbf{g}(v)}{dv^4}\\ e^{\mathbf{a}_1}=[e^{a_{1x}},e^{a_{1y}},e^{a_{1z}}] & \mathbf{a}_1\mathbf{a}_2=[a_{1x}a_{2x},a_{1y}a_{2y},a_{1z}a_{2z}]^T\\ \frac{\mathbf{a}_1}{\mathbf{a}_2}=\left[\frac{a_{1x}}{a_{2x}},\frac{a_{1y}}{a_{2y}},\frac{a_{1z}}{a_{2z}}\right]^T & \sqrt[n]{\frac{\mathbf{a}_1}{\mathbf{a}_2}}=\left[\sqrt[n]{\frac{a_{1x}}{a_{2x}}},\sqrt[n]{\frac{a_{1y}}{a_{2y}}},\sqrt[n]{\frac{a_{1z}}{a_{2z}}}\right]^T\end{array} \tag{2}$$

In the future, we will investigate how to use a general expression for $\mathbf{F}(u,v)$ in order to make PDE patch-based surface reconstruction more powerful. In this paper, we set $\mathbf{F}(u,v)$ to 0, which makes Eq. (1) homogeneous. In that case, we can use the method of separation of variables to derive its four closed-form solutions. Assuming that the variables u, v in Eq. (1) are separable, $\mathbf{X}(u,v)$ be expressed as follows:

$$\mathbf{X}(u,v)=\mathbf{f}(u)\mathbf{g}(v) \tag{3}$$

Substituting Eq.(3) back to Eq. (1), we get:

$$\begin{array}{l}\mathbf{a}_1\mathbf{g}(v)\frac{d^4\mathbf{f}(u)}{du^4}+\mathbf{a}_2\mathbf{f}(u)\frac{d^4\mathbf{g}(v)}{dv^4}=0\Rightarrow\\ \mathbf{a}_1\mathbf{f}''''(u)\frac{1}{\mathbf{f}(u)}=-\mathbf{a}_2\mathbf{g}''''(v)\frac{1}{\mathbf{g}(v)}\end{array} \tag{4}$$

By setting both sides in Eq. (4) to $\mathbf{c}_0$, which is a vector-valued constant, We get Eq. (5):

$$\mathbf{a}_1\mathbf{f}''''(u)\frac{1}{\mathbf{f}(u)}=-\mathbf{a}_2\mathbf{g}''''(v)\frac{1}{\mathbf{g}(v)}=\mathbf{c}_0 \tag{5}$$

With the above treatment, the partial differential equation in Eq. (1) is transformed into two ordinary differential equations given in Eq. (5). The first ordinary differential equation is:

$$\mathbf{a}_1\mathbf{f}''''(u)=\mathbf{c}_0\mathbf{f}(u) \tag{6}$$

From Eq. (6), we know that $\mathbf{f}(u)$ can be taken to be:

$$\mathbf{f}(u)=e^{\mathbf{r}u} \tag{7}$$

From the above equation, we obtain the fourth-order derivative of $\mathbf{f}(u)$ as:

$$\mathbf{f}''''(u)=\mathbf{r}^4e^{\mathbf{r}u}$$

Substituting the expressions of $\mathbf{f}(u)$ and $\mathbf{f}''''(u)$ into Eq. (6), we obtain:

$$\mathbf{a}_1\mathbf{r}^4e^{\mathbf{r}u}=\mathbf{c}_0e^{ru} \tag{8}$$

From Eq. (8), we obtain:

$$\mathbf{r}^4=\frac{\mathbf{c}_0}{\mathbf{a}_1} \tag{9}$$

To solve Eq. (9) for $\mathbf{r}$ two cases should be considered. The first case is $\frac{\mathbf{c}_0}{\mathbf{a}_1}>0$ and the second case is $\frac{\mathbf{c}_0}{\mathbf{a}_1}<0$, where the inequalities must be understood as component-wise.

For the first case, $\frac{\mathbf{c}_0}{\mathbf{a}_1}>0$, we have $\frac{\mathbf{c}_0}{\mathbf{a}_1}=\left|\frac{\mathbf{c}_0}{\mathbf{a}_1}\right|>0$. To simplify mathematical notations, we let:

$$\mathbf{q}_1=\sqrt[4]{\frac{\mathbf{c}_0}{\mathbf{a}_1}}=\sqrt[4]{\left|\frac{\mathbf{c}_0}{\mathbf{a}_1}\right|}$$

Now we can obtain the four roots of Eq. (9) as follows:

$$\mathbf{r}_1=\mathbf{q}_1,\quad \mathbf{r}_2=-\mathbf{q}_1,\quad \mathbf{r}_3=i\mathbf{q}_1,\quad \mathbf{r}_4=-i\mathbf{q}_1 \tag{10}$$

where $i^2=-1$. Substituting Eq. (10) back to Eq. (7), we obtain $\mathbf{f}(u)$ as follows:

$$\mathbf{f}(u)=\mathbf{c}_1e^{\mathbf{q}_1u}+\mathbf{c}_2e^{-\mathbf{q}_1u}+\mathbf{c}_3cos(\mathbf{q}_1u)+\mathbf{c}_4sin(\mathbf{q}_1u) \tag{11}$$

where $\mathbf{c}_1,\mathbf{c}_2,\mathbf{c}_3,\mathbf{c}_4$, are vector-valued constants.

For the second case, $\frac{\mathbf{c}_0}{\mathbf{a}_1}<0$,we let:

$$\mathbf{q}_2=\frac{\sqrt{2}}{2}\sqrt[4]{\left|\frac{\mathbf{c}_0}{\mathbf{a}_1}\right|}=\frac{\sqrt{2}}{2}q_1$$

We can also obtain the four roots of Eq. (8) as follows:

$$\mathbf{r}_1=\mathbf{q}_2(1+i),\quad \mathbf{r}_2=-\mathbf{q}_2(1+i),\quad \mathbf{r}_3=\mathbf{q}_2(1-i),\quad \mathbf{r}_4=-\mathbf{q}_2(1-i)$$

In such a case, $\mathbf{f}(u)$ can be expressed as follows:

$$\begin{array}{ll}\mathbf{f}(u)= & e^{\mathbf{q}_2u}\left[\mathbf{c}_1cos(\mathbf{q}_2u)+\mathbf{c}_2sin(\mathbf{q}_2u)\right]+\\ & e^{-\mathbf{q}_2u}\left[\mathbf{c}_3cos(\mathbf{q}_2u)+\mathbf{c}_4sin(\mathbf{q}_2u)\right]\end{array} \tag{12}$$

The second ordinary differential equation given in Eq. (5) can be written as:

$$-\mathbf{a}_2\mathbf{g}''''(v)=\mathbf{c}_0\mathbf{g}(v) \tag{13}$$

We can use the same method as solving $\mathbf{f}(u)$ to obtain the solution of $\mathbf{g}(v)$ for the two cases: $\frac{\mathbf{c}_0}{\mathbf{a}_2}>0$ and $\frac{\mathbf{c}_0}{\mathbf{a}_2}<0$.

For the first case, $\frac{\mathbf{c}_0}{\mathbf{a}_2}>0$, we let:

$$\mathbf{q}_3=\sqrt[4]{\left|\frac{\mathbf{c}_0}{\mathbf{a}_2}\right|}$$

We can get the solution of $\mathbf{g}(v)$ below:

$$\mathbf{g}(v)=\mathbf{c}_5e^{\mathbf{q}_3v}+\mathbf{c}_6e^{-\mathbf{q}_3v}+\mathbf{c}_7cos(\mathbf{q}_3v)+\mathbf{c}_8sin(\mathbf{q}_3v) \tag{14}$$

where $\mathbf{c}_5,\mathbf{c}_6,\mathbf{c}_7,\mathbf{c}_8$ are vector-valued constants.

For the second case, $\frac{\mathbf{c}_0}{\mathbf{a}_2}<0$, we define:

$$\mathbf{q}_4=\frac{\sqrt{2}}{2}\sqrt[4]{\left|\frac{\mathbf{c}_0}{\mathbf{a}_2}\right|}=\frac{\sqrt{2}}{2}\mathbf{q}_3$$

Under such a case, we obtain the expression of $\mathbf{g}(v)$ as follows:

$$\begin{array}{ll}\mathbf{g}(v)= & e^{\mathbf{q}_4v}\left[\mathbf{c}_5cos(\mathbf{q}_4v)+\mathbf{c}_6sin(\mathbf{q}_4v)\right]+\\ & e^{-\mathbf{q}_4v}\left[\mathbf{c}_7cos(\mathbf{q}_4v)+\mathbf{c}_8sin(q\mathbf{q}_4v)\right]\end{array} \tag{15}$$

Since $\mathbf{f}(u)$ and $\mathbf{g}(v)$ both have two forms, which are Eq. (11) and Eq. (12) for $\mathbf{f}(u)$ and Eq. (14) with Eq. (15) for $\mathbf{g}(v)$, they can be substituted into Eq. (3) to obtain four solutions of $\mathbf{X}(u,v)=\mathbf{f}(u)\mathbf{g}(v)$. We use $\mathbf{X}_1(u,v)$, $\mathbf{X}_2(u,v)$, $\mathbf{X}_3(u,v)$, and $\mathbf{X}_4(u,v)$, to denote the four solutions, which are obtained below.

Multiplying Eq. (11) with Eq. (14), we get $\mathbf{X}_1(u,v)$ below:

$$\begin{array}{ll}\mathbf{X}_1(u,v)= & \left[\mathbf{c}_1e^{\mathbf{q}_1u}+\mathbf{c}_2e^{-\mathbf{q}_1u}+\mathbf{c}_3cos(\mathbf{q}_1u)+\mathbf{c}_4sin(\mathbf{q}_1u)\right]\\ & \left[\mathbf{c}_5e^{\mathbf{q}_3v}+\mathbf{c}_6e^{-\mathbf{q}_3v}+\mathbf{c}_7cos(\mathbf{q}_3v)+\mathbf{c}_8sin(\mathbf{q}_3v)\right]\end{array} \tag{16}$$

Multiplying Eq. (11) with Eq. (15), we get $\mathbf{X}_2(u,v)$ below:

$$\begin{array}{ll}\mathbf{X}_2(u,v)= & \left[\mathbf{c}_1e^{\mathbf{q}_1u}+\mathbf{c}_2e^{-\mathbf{q}_1u}+\mathbf{c}_3cos(\mathbf{q}_1u)+\mathbf{c}_4sin(\mathbf{q}_1u)\right]\\ & \left(e^{\mathbf{q}_2u}\left[\mathbf{c}_1cos(\mathbf{q}_2u)+\mathbf{c}_2sin(\mathbf{q}_2u)\right]+\right.\\ & \left.e^{-\mathbf{q}_2u}\left[\mathbf{c}_3cos(\mathbf{q}_2u)+\mathbf{c}_4sin(\mathbf{q}_2u)\right]\right)\end{array} \tag{17}$$

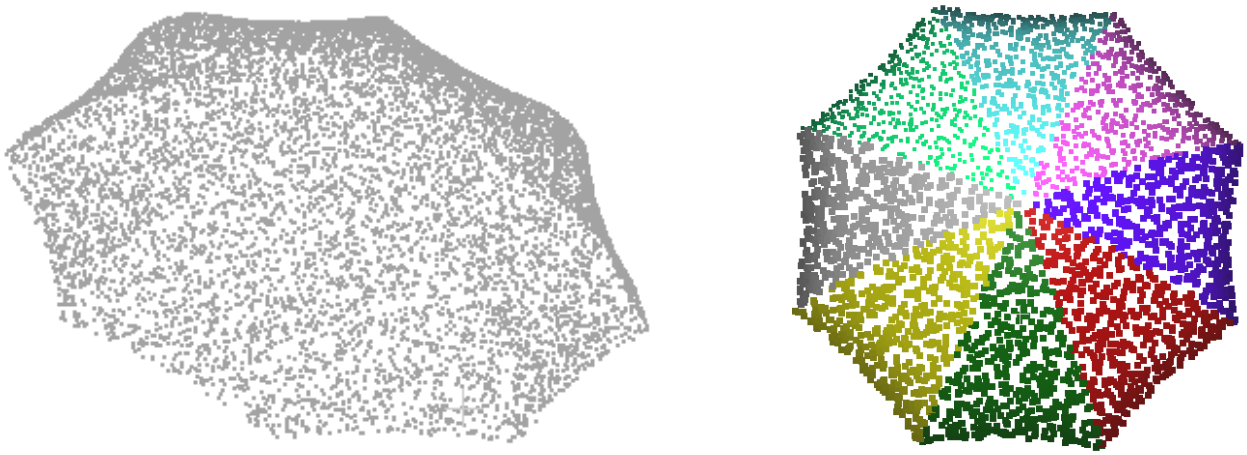
Multiplying Eq. (12) with Eq. (14), we get $\mathbf{X}_3(u,v)$ below:

$$\begin{array}{ll}\mathbf{X}_3(u,v)= & \left(e^{\mathbf{q}_2u}\left[\mathbf{c}_1cos(\mathbf{q}_2u)+\mathbf{c}_2sin(\mathbf{q}_2u)\right]+\right.\\ & \left.e^{-\mathbf{q}_2u}\left[\mathbf{c}_3cos(\mathbf{q}_2u)+\mathbf{c}_4sin(\mathbf{q}_2u)\right]\right)\\ & \left[\mathbf{c}_5e^{\mathbf{q}_3v}+\mathbf{c}_6e^{-\mathbf{q}_3v}+\mathbf{c}_7cos(\mathbf{q}_3v)+\mathbf{c}_8sin(\mathbf{q}_3v)\right]\end{array} \tag{18}$$

Multiplying Eq. (12) with Eq. (15), we get $\mathbf{X}_4(u,v)$ below:

$$\begin{array}{ll}\mathbf{X}_4(u,v)= & \left(e^{\mathbf{q}_2u}\left[\mathbf{c}_1cos(\mathbf{q}_2u)+\mathbf{c}_2sin(\mathbf{q}_2u)\right]+\right.\\ & \left.e^{-\mathbf{q}_2u}\left[\mathbf{c}_3cos(\mathbf{q}_2u)+\mathbf{c}_4sin(\mathbf{q}_2u)\right]\right)\\ & \left(e^{\mathbf{q}_4v}\left[\mathbf{c}_5cos(\mathbf{q}_4v)+\mathbf{c}_6sin(\mathbf{q}_4v)\right]+\right.\\ & \left.e^{-\mathbf{q}_4v}\left[\mathbf{c}_7cos(\mathbf{q}_4v)+\mathbf{c}_8sin(\mathbf{q}_4v)\right]\right)\end{array} \tag{19}$$

Each of the above four solutions can be used to define 4-sided PDE patches for surface reconstruction. In this paper, $\mathbf{X}_4(u,v)$ is



(a) (b)

Fig. 2. Segmentation of a point cloud of the umbrella example: (a) Original point set; (b) segmented point subsets.

adopted to reconstruct 3D surfaces from point clouds.

Conducting the multiplication operation in Eq. (19) and letting:

$$
\begin{aligned}
\mathbf{f}_1(u,v) &= e^{\mathbf{q}_2 u} e^{\mathbf{q}_4 v} cos(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_2(u,v) &= e^{\mathbf{q}_2 u} e^{\mathbf{q}_4 v} cos(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_3(u,v) &= e^{\mathbf{q}_2 u} e^{\mathbf{q}_4 v} sin(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_4(u,v) &= e^{\mathbf{q}_2 u} e^{\mathbf{q}_4 v} sin(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_5(u,v) &= e^{\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} cos(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_6(u,v) &= e^{\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} cos(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_7(u,v) &= e^{\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} sin(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_8(u,v) &= e^{\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} sin(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_9(u,v) &= e^{-\mathbf{q}_2 u} e^{\mathbf{q}_4 v} cos(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_{10}(u,v) &= e^{-\mathbf{q}_2 u} e^{\mathbf{q}_4 v} cos(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_{11}(u,v) &= e^{-\mathbf{q}_2 u} e^{\mathbf{q}_4 v} sin(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_{12}(u,v) &= e^{-\mathbf{q}_2 u} e^{\mathbf{q}_4 v} sin(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_{13}(u,v) &= e^{-\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} cos(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_{14}(u,v) &= e^{-\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} cos(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)\\
\mathbf{f}_{15}(u,v) &= e^{-\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} sin(\mathbf{q}_2 u) cos(\mathbf{q}_4 v)\\
\mathbf{f}_{16}(u,v) &= e^{-\mathbf{q}_2 u} e^{-\mathbf{q}_4 v} sin(\mathbf{q}_2 u) sin(\mathbf{q}_4 v)
\end{aligned}
\tag{20}
$$

Eq. (19) can be transformed into:

$$
\mathbf{X}(u,v) = \sum_{j=1}^{16} \mathbf{d}_j \mathbf{f}_j(u,v) \tag{21}
$$

where $\mathbf{d}_j$, $(j = 1,\ldots,16)$ are the vector-valued unknowns. Note that $\mathbf{X}(u,v)$ in Eq. (21) is a parametric surface, which defines a 4-sided PDE patch. We discuss how to determine the unknowns $\mathbf{d}_j$, $(j = 1,\ldots,16)$ in the following subsection.

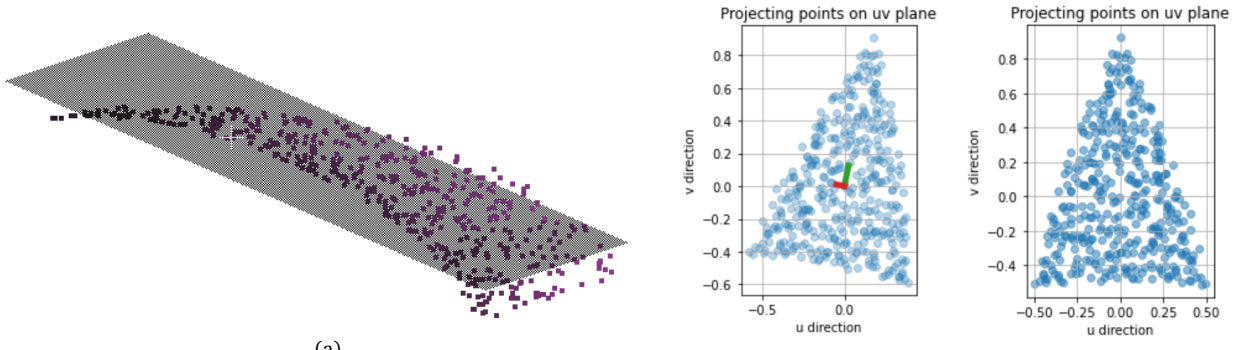
4.2. Segmentation of point cloud

Like mesh segmentation, there are also a lot of works focusing on the segmentation of point clouds. The existing point cloud segmentation methods can be classified into three categories: semantic segmentation, instance segmentation, and part segmentation [36]. Deep learning has also been introduced into point cloud segmentation. One of the most commonly adopted methods is applying different neural networks to part segmentation of point clouds. For example, PointNet is a very popular neural network architecture, which applies a specialized deep neural network to point clouds for several tasks, including object classification and part segmentation [37]. Later work also proposed many methods to improve the performance of part segmentation of point clouds. The graph convolution is one of the most used neural network architectures. Please refer to [38] for a comprehensive review of deep learning-based 3D segmentation.

The above methods have a problem when they are used to segment a point cloud into some subsets. The problem is that the boundaries between two reconstructed patches are not very smooth because the points in the regions around the boundaries are sparse and irregular. To tackle such a problem, the method of edge detection of point clouds could be an effective tool. For instance, Bazazian proposed a weakly supervised learning approach to detect the edges of point clouds [39].

A simple feasible strategy to reconstruct PDE-based surfaces patch by patch using our proposed method is to use any of the existing methods reviewed in [38] to segment a point cloud into several subsets. When the shapes defined by the points in some segmented subsets are too complicated, it is necessary to split these segmented subsets obtained from existing methods into smaller subsets. Using this procedure (usually called *segmentation refinement*), we can obtain a reconstructed shape of high quality.

We will reconstruct several objects using our proposed PDE-based method, including sphere, cylinder, umbrella, and table. The point clouds of these objects will be segmented automatically or manually into some subsets. Specifically, the point sets of the sphere and the cylinder can be segmented into several subsets automatically based on the coordinates of the point sets. As for the point set of the table, which are composed of planes, an automatic method can also be applied to obtain several subsets (planes), which will be introduced in detail in the experimental section. Lastly, we perform segmentation of the point set of the umbrella using a software called *'CloudCompare'*. However, it is also possible to perform automatic segmentation of this point set using more sophisticated techniques. This will be part of our future work in the field. Fig. 2 shows the original point cloud and the segmented result of an umbrella that we will reconstruct later. The points in each of the subsets will be parameterized, and our proposed PDE-based reconstruction method will be applied to reconstruct a PDE patch by fitting these points. The parameterization of point clouds and the fitting process will be described in detail in the following subsections.



(a) (b) (c)

Fig. 3. Parameterization of points in a subset: (a) Fitting plane. (b) Projecting points to a $u-v$ plane. (c) Aligned projected points with u and v directions.

4.3. Point cloud parameterization

Proper parameterization of point clouds has a big impact on final reconstruction quality. Various parameterization methods have been proposed in the literature. Ma and Kruth introduced commonly used methods of parameterization of point sets for B-spline and NURBS surfaces, including uniform parameterization, cumulative chord length parameterization, centripetal parameterization and base surface parameterization [40]. The first three methods are more suitable for a point set with a regular pattern, but when a point set is scattered, the last approach may be a better choice. Other approaches for data parameterization can be found, for instance, in [41, 42].

In this work, we apply different parameterization methods to different types of models. For relatively simple geometry primitives such as a sphere and a cylinder, since they have an analytical representation, we use their parametric mathematical equations to achieve their parameterization. For example, a sphere can be parameterized with two angles in a spherical coordinate system, which can be normalized to get the parametric values of parametric variables u and v on the unit interval for the points on the sphere. Since the points on a whole sphere cannot be reconstructed with a single PDE patch, we first segment the point cloud defining a sphere into two equal subsets, which can readily be done automatically. However, in our experiments we found that the points in the two subsets still cannot be well reconstructed by two PDE patches. Therefore, we further segment the sphere into four equal subsets or eight equal subsets. We also found that the former option is preferable since four subsets lead to good reconstruction quality and the number of the design variables for four subsets is half of that for eight subsets. Finally, we parameterize the points in each of the four subsets in a spherical coordinate system to obtain their u and v parametric values.

A cylinder also has a parametric mathematical expression. Similar to the parameterization of a sphere, we segment the point cloud defining a cylinder into 2 subsets. For the points in each of the two subsets, we obtain the parametric values of two parametric variables u and v from the two variables angle and height defining the points in each of the two subsets.

The third model we reconstruct is a table that is completely composed of planes. Segmenting the table into planes can also be done automatically. The details of the automatic segmentation of the table will be given in Section 5. For each segmented plane, it can be regarded as a $u-v$ plane. Normalizing the coordinate of the points on every plane would give us the parametric values of the parametric variables u and v for each point on the segmented plane. Then, the points on each of the planes are used to reconstruct a PDE patch. Combining all the reconstructed PDE patches generates the final reconstructed 3D shape of the table.

Finally, an umbrella is also reconstructed. The point set is firstly segmented into eight equal subsets, as shown in Fig. 2. For the points in each of the segmented subsets, we find a plane that best fits the point set, as shown in Fig. 3(a). Then the 3D points in the subset are projected onto the plane, which can also be regarded as a $u-v$ plane. Since the principal component analysis (PCA) axes of the projected points on the $u-v$ plane do not align with the u and v directions as shown in Fig. 3(b), we apply a rotation transformation to the projected points to make their PCA axes aligning with the u and v directions as shown in Fig. 3(c). Finally, normalizing operations are applied to the points on the $u-v$ plane to obtain the parametric values of the points in each of the segmented subsets.

4.4. Fitting

After parameterizing the points in each of the segmented subsets, we obtained their parametric values u_n and v_n for each point $\mathbf{X}_n$ in the subsets. Then we fit the PDE patch to the points. As discussed above, applying our developed PDE patch to surface reconstruction from point clouds requires to find the 16 vector-valued unknowns $\mathbf{d}_j$, $(j = 1,2,3,\ldots,16)$ so that the PDE patch $\mathbf{X}(u,v)$ will best approximate the points in the subset.

If there are N points $\mathbf{X}_n$ $(n = 1,2,3,\ldots,N)$ in a subset to be reconstructed by one PDE patch $\mathbf{X}(u,v)$, the squared sum of the errors between the known points $\mathbf{X}_n$ and the unknown points $\mathbf{X}(u_n,v_n)$ can be determined with the following equation:

$$\mathbf{E} = \sum_{n=1}^{N} \left[\mathbf{X}(u_n,v_n) - \mathbf{X}_n\right]^2 = \sum_{n=1}^{N} \left[\sum_{j=1}^{16} \mathbf{d}_j \mathbf{f}_j(u_n,v_n) - \mathbf{X}_n\right]^2 \tag{22}$$

To minimize the error $\mathbf{E}$ and find the 16 vector-valued unknowns, we apply the method of least squares, given by the following equation:

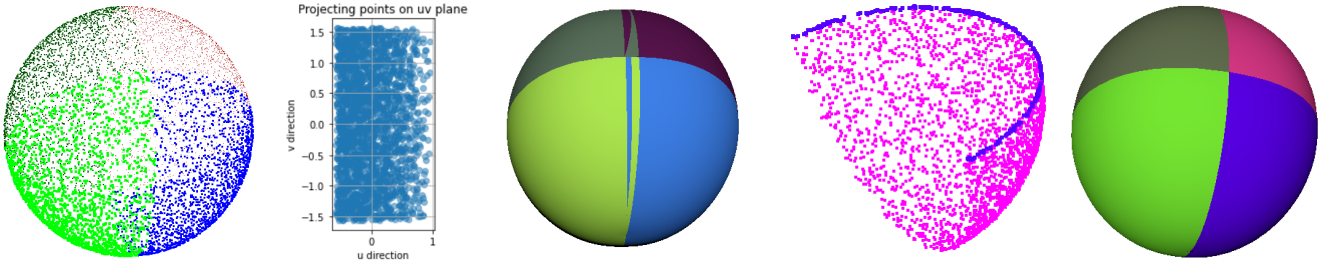
$$\frac{\partial \mathbf{E}}{\partial \mathbf{d}_k} = 0 \qquad (k = 1,2,3,\ldots,16) \tag{23}$$

Substituting Eq. (22) into Eq. (23), the following system of equations is obtained:

$$\sum_{j=1}^{16} \mathbf{d}_j \sum_{n=1}^{N} \mathbf{f}_j(u_n,v_n)\mathbf{f}_k(u_n,v_n) = \sum_{n=1}^{N} \mathbf{X}_n \mathbf{f}_k(u_n,v_n) \tag{24}$$

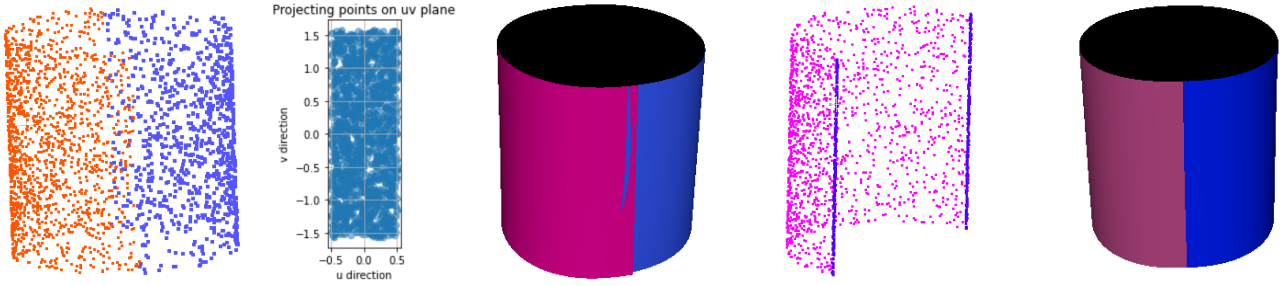
for $k = 1,2,3,\ldots,16$. Therefore, there are 16 equations in Equation (24) that must be solved to find the 16 vector-valued unknowns $\mathbf{d}_k$ $(k = 1,\ldots,16)$.

Note that in Eq. (24), $\mathbf{f}_j(u_n,v_n)$ and $\mathbf{f}_k(u_n,v_n)$ involve the constants $\mathbf{q}_2$ and $\mathbf{q}_4$, which can be treated as two design variables and optimized to obtain the optimal PDE patch which best fits the points $\mathbf{X}_n$ $(n = 1,\ldots,N)$. However, Eq. (8) would become non-



(a) (b) (c) (d) (e)

Fig. 4. PDE-based reconstruction from the point cloud of a sphere: (a) Segmented point cloud of a sphere. (b) Projecting the points in a subset to a $u-v$ plane. (c) Reconstructed shape with small overlaps consisting of four PDE patches without adding points in the regions around boundaries. (d) The points in a subset after adding points to the regions around boundaries. (e) The final result without overlaps obtained by adding points to the regions around boundaries.



(a) (b) (c) (d) (e)

Fig. 5. PDE -based reconstruction from the point cloud of a cylinder: (a) Segmented point cloud of a cylinder. (b) Projecting the points in a subset to a $u-v$ plane. (c) Reconstructed shape with small overlaps consisting of two PDE patches without adding points in the regions around boundaries. (d) The points in a subset after adding points to the regions around boundaries. (e) The final result without overlaps obtained by adding points to the regions around boundaries.

linear if we treat $\mathbf{q}_2$ and $\mathbf{q}_4$ as design variables, which makes it more difficult to find the 16 vector-valued unknowns $\mathbf{d}_k$. Owing to this fact, in this paper we treat $\mathbf{q}_2$ and $\mathbf{q}_4$ as two constants and set $\mathbf{q}_2 = \mathbf{q}_4 = \mathbf{0.1}$, which provides good results in all our experiments.

5. Results

5.1. Experiments and results

For the points in each of the subsets obtained from segmenting a point cloud, we use Eq. (21) to reconstruct a PDE patch $\mathbf{X}(u,v)$. After the points in all the subsets have been used to reconstruct PDE patches, the reconstructed 3D shape consisting of the reconstructed PDE patches is obtained. In what follows, we present some examples to illustrate this process.

The first example is to reconstruct a sphere from its point cloud. As discussed above, the point cloud describing a sphere is segmented into four subsets as shown in Fig. 4(a). For the points in each of the four subsets, Eq. (21) is used to reconstruct one PDE patch from the points. To do this, the points in the subset are projected to a $u-v$ plane as shown in Fig. 4(b). After the four PDE patches have been reconstructed, they are automatically connected together to represent the reconstructed shape from the point cloud. However, we found that the reconstructed patches have small overlaps between two PDE patches such as those between the grey and red patches and between the green and blue patches as shown in Fig. 4(c). To address this problem, we increase the number of the points around boundaries by first upsampling the original point cloud, then keeping the points around the boundaries using a self-defined filter based on the coordinates of the point cloud, and finally combining them with the original point cloud. Fig. 4(d) shows the points in one subset after increasing the number of the points in the regions around boundaries. With this treatment, we obtain the reconstructed shape shown in Fig. 4(e), which indicates that the small overlaps have been removed and a good reconstruction result has been obtained.

The second example is to reconstruct a cylinder from its point cloud. As discussed in Section 4.2, the point cloud has been segmented into two equal subsets as shown in Fig. 5(a). Then, the points in each of the subsets are projected to a $u-v$ plane shown in Fig. 5(b). After using Eq. (21) to reconstruct the two PDE patches shown in Fig. 5(c), small overlaps occur again. With the same treatment as discussed above, we add more points in the regions around the boundaries between the two PDE patches as shown in Fig. 5(d). By adding new points, the small overlaps disappear as shown by the two reconstructed PDE patches depicted in Fig. 5(e).

The third example is to reconstruct a table from its point cloud. The points in the point cloud of a table are on some planes. This example is used to demonstrate that our proposed PDE-based method can be used to reconstruct not only 3D shapes consisting of curved surfaces but also 3D shapes consisting of flat planes or a combination of curved surfaces and flat planes.

Even though a plane seems to be a simple primitive, plane detection and reconstruction is important in many fields such as robotic perception and image processing. For example, robots need to detect ground where it is safe to walk, as well as the walls

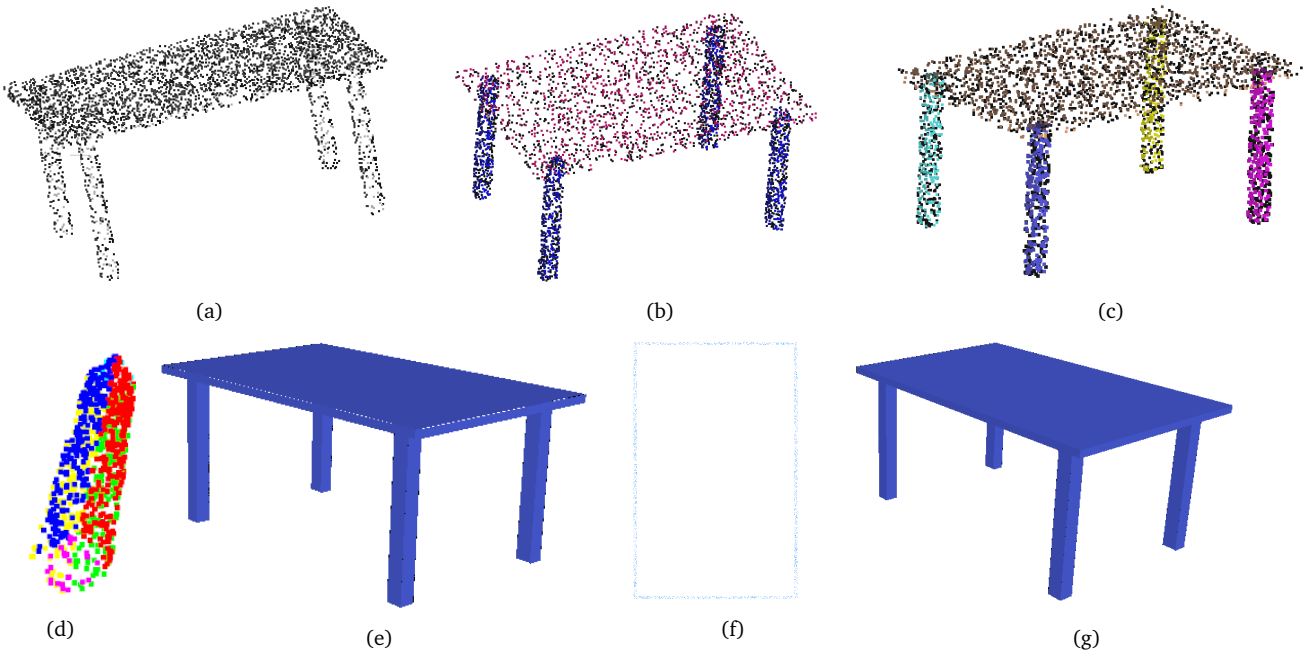


Fig. 6. PDE-based reconstruction from the point cloud of a table (left-right,top-bottom): (a) Point cloud of a table. (b) Segmenting the point cloud of the table into a top part and a bottom part. (c) Using the K-means clustering algorithm to segment the bottom part into four subparts. (d) Using the RANSAC algorithm to segment each of the subparts into six subsets. (e) Reconstructed shape with small gaps between two adjacent PDE patches. (f) Points in a subset after adding points to the regions around boundaries. (g) The final result without gaps obtained by adding points to the regions around boundaries.

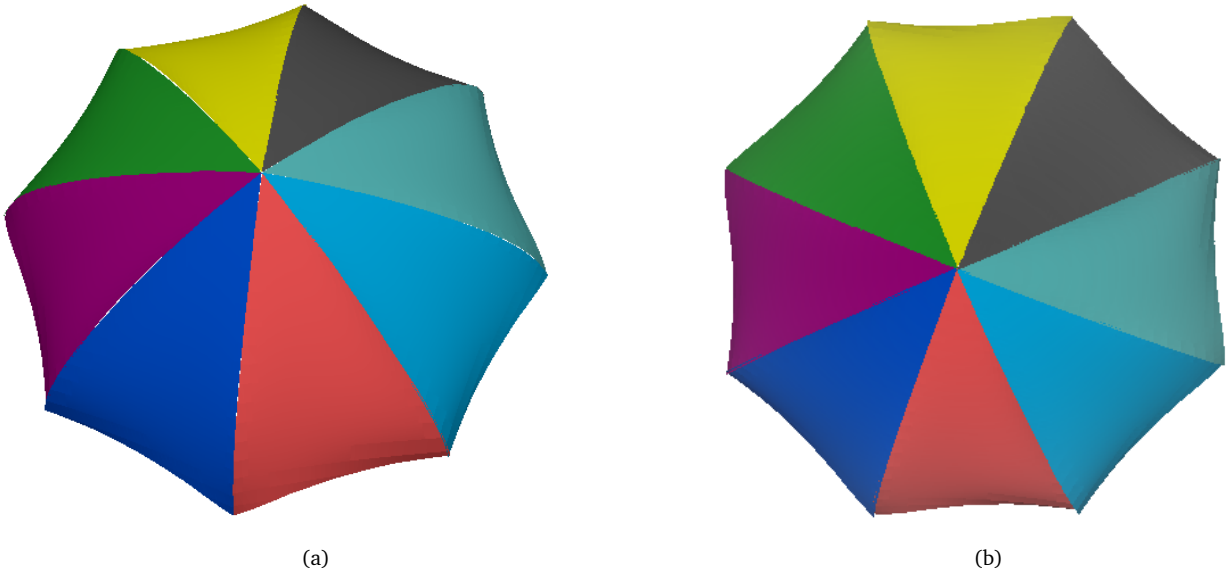


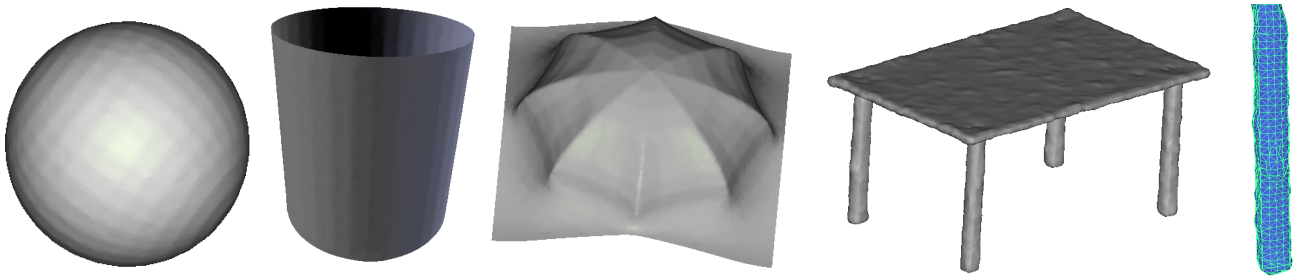
Fig. 7. (a) Reconstructed umbrella; (b) Final result after post-processing.

and other artificial (and mostly linear and flattened) areas for collision avoidance.

For a point cloud of this type, no manual segmentation is needed. Usually, it takes three steps to segment a point cloud. In the first step, we segment the point cloud in Fig. 6(a) of the table into the top part and leg part shown in Fig. 6(b) using the part segmentation algorithm of point clouds as we introduced in Section 4.2.

Then in the second step, we use a K-means clustering algorithm to the leg part and change it into four subparts shown in Fig. 6(c). Each subpart represents a leg. Finally, in the third step, we use a plane detection algorithm called RANSAC, which is a state-of-the-art plane detector, from Point Cloud Library (PCL) to segment each of the four subparts into six subsets, which are six planes of a leg. With this treatment, the points in each of the subparts are segmented into those in the six subsets (planes) of a leg as shown in Fig. 6(d).

For the points in each of the segmented subsets, we reconstruct a PDE patch by applying our developed PDE-based reconstruction method. Fig. 6(e) shows the reconstructed PDE patches from the point cloud of the table. It can be observed that there are very small gaps between reconstructed PDE patches. To tackle this



(a) (b) (c) (d) (e)

Fig. 8. 3D shapes reconstructed using the Poisson surface reconstruction technique: (a) Reconstructed sphere; (b) Reconstructed cylinder; (c) Reconstructed umbrella; (d) Reconstructed table; (e) Close view of a leg of the reconstructed table.

Table 1
Number of the design variables needed by our proposed PDE-based method and the polygon-based method to reconstruct 3D shapes from different point clouds.

	Sphere	**Cylinder**	**Table**	**Umbrella**
Polygon-based	726	192	45,279 (using Poisson technique) 120 (Modeling by software)	86,613
PDE-based	192	96	1080	384

problem, we add more points in the regions around the boundaries. Fig. 6(f) shows the result of adding more points in the regions around the boundaries of a plane. The final result after adding more points is shown in Fig. 6(g), which removes the gaps and is of good quality.

The last example is to reconstruct an umbrella from its point cloud. As discussed in Section 4.2, the point cloud of the umbrella is segmented into eight subsets. The points in each of the eight subsets are projected to a $u-v$ plane. Since the projected points do not fill the top left region and the top right region as shown in Fig. 3(c), trimming the reconstructed PDE patch corresponding to the two regions is necessary. We use the boundaries of the obtained subsets to trim the reconstructed PDE-based patches. Fig. 7(a) shows the reconstructed umbrella after trimming the reconstructed PDE patches. As we can see, there are some very small gaps. This problem can be addressed by sampling more points in the regions around the boundaries of subsets. In this example, we manually eliminate the gaps and obtain the result shown in Fig. 7(b).

5.2. Comparison with a polygon-based method

To illustrate the advantages of our proposed PDE-based surface reconstruction technique, we also reconstruct polygon surfaces from the same point clouds used in this paper by applying a widely adopted method called Poisson surface reconstruction that was briefly mentioned in Section 2. The main aspect we consider is the number of variables needed to reconstruct the same 3D shape while keeping the reconstructed surface with high quality. The errors between the surface defined by the original points set and the reconstructed surfaces, including the PDE surface and the polygon surface are calculated to demonstrate the quality of the result. Specifically, for the error between the surface defined by the original point set and the reconstructed PDE surface, we calculate it through the following equations:

$$
\begin{aligned}
Err_M &= \frac{1}{N}\sum_{n=1}^{N}|\mathbf{X}_n - \mathbf{X}(u_n, v_n)| \\
Err_D &= \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\Big||\mathbf{X}_n - \mathbf{X}(u_n, v_n)| - Err_M\Big|\right)^2}
\end{aligned}
\tag{25}
$$

where Err_M indicates the average error between the two surfaces, Err_D stands for the standard deviation of the errors regarding all corresponding points, and $|.|$ indicates the Eucidean distance between two points (vectors). For the error between the surface and the reconstructed polygon surface, we take advantage of a tool in a point cloud processing software called CloudCompare and the tool is designed to calculate the difference between two surfaces.

Fig. 8 shows the polygon surfaces reconstructed from the point cloud of the sphere, cylinder, table and umbrella, which we used to reconstruct PDE patches in the above subsection.

Comparing the reconstructed shapes shown in Fig. 8 obtained from the Poisson surface reconstruction algorithm with those obtained with our proposed PDE-based method, it is clear that our proposed PDE-based method has better quality of reconstructed 3D shapes.

Table 1 gives the number of the design variables required by the polygon-based method and PDE-based method to reconstruct different 3D shapes. We can see that for the point clouds defining 3D shapes with curved surfaces, our proposed PDE-based surface reconstruction method requires much fewer design variables than the polygon-based method.

To demonstrate the quality of the reconstructed surface, we use the polygon-based method to reconstruct the 3D shapes in Table 1 to make their vertex number roughly the same as the number of variables needed in corresponding PDE-based surface, thus the number of variables needed to represent the same 3D shape would be roughly the same. Table 2 shows the number of variables required to represent various 3D shapes for both the polygon-based

Table 2
Number of variables required to represent different 3D shape for polygon-based method and proposed PDE-based method after simplification of the reconstructed polygon mesh.

	Sphere	Cylinder	Table	Umbrella
Polygon-based	$74 \times 3 = 222$	$32 \times 3 = 96$	$377 \times 3 = 1131$	$184 \times 3 = 552$
PDE-based	192	96	1080	384

Table 3
The mean error and its deviation after simplification between the surface defined by different point clouds and the reconstructed PDE surface and the polygon surface respectively (a/b: a refers to the mean error, b refers to the deviation).

	Sphere	Cylinder	Table	Umbrella
Polygon-based	0.0091/0.0099	0.0045/0.0071	01083/0.3961	0.0120/0.0613
PDE-based	0.0026/0.0055	0.0029/0.0062	$9.0 \times 10^{-6}/1.9 \times 10^{-5}$	0.0045/0.0094

method and our proposed PDE-based method after simplifying the reconstructed polygon mesh.

Then we calculated the average error and its standard deviation between the surface defined by the original point set and the reconstructed surfaces. Table 3 shows the results and we can see that both the mean error and the standard deviation between the PDE surface and original point set is smaller than that between the polygon surface and the original point set. In conclusion, our proposed PDE-based surface reconstruction method from point cloud outperforms a classical polygon-based surface reconstruction technique from point cloud regarding the number of variables and the quality of the output 3D shape.

6. Conclusions

In this paper, we develop a new method to reconstruct 3D shapes from point clouds with multiple PDE patches. Each PDE patch is based on an explicit closed-form solution to a vector-valued fourth-order partial differential equation, which is efficient and accurate due to the feature of analytical closed-form solutions of the obtained mathematical expressions. In comparison with the polygon-based method, our proposed PDE-based surface reconstruction method involves much fewer design variables. Since multiple PDE patches are used in reconstructing 3D shapes from point clouds, many complex shapes can be reconstructed with our proposed PDE-based method. By adding more points in the regions around the boundaries shared by two PDE patches, good positional continuity can be obtained by the method proposed in this paper, which avoids the postprocessing of reconstructed surfaces.

Regarding future work, there are several directions that have not been examined in this paper. We will investigate them in our forthcoming work. First, the problem of overlaps and gaps in the regions around the boundaries between two PDE patches are treated by adding more points in the regions. More effective methods should be developed to solve this problem.

Second, how parameters $\mathbf{q}_2$ and $\mathbf{q}_4$ involved in the mathematical expressions of PDE patches influence quality, efficiency, and capacity has not been investigated in this paper. They can be treated as two design variables and optimized to determine their optimal value.

The method proposed in this paper is used to reconstruct 3D shapes from point clouds. Shape reconstruction from images is another important topic. We will extend the method proposed in this paper to reconstruct 3D shapes from a single image or multiple images in our following work.

Surface reconstruction is to recreate surfaces from a given point cloud within the shortest possible time and with given quality criteria. In this regard, machine learning has been applied to shape reconstruction due to its ability in producing good results without restrictions on point cloud size and points' order as well as the capacity in generating meshes from a small number of point samples in an unorganized point cloud. In the future, we will combine our proposed PDE-based method with machine learning to develop a new PDE patch and machine learning-based shape reconstruction method.

Acknowledgments

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 China Scholarship Council. Andrés Iglesias thanks the project TIN2017-89275-R funded by MCIN/ AEI /10.13039/501100011033/ FEDER "Una manera de hacer Europa".

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of this article.

References

[1] Z. Zhu, E. Chaudhry, S. Wang, Y. Xia, A. Iglesias, L. You, J.J. Zhang: Shape reconstruction from point clouds using closed form solution of a fourth-order partial differential equation. Proc. of International Conference on Computational Science, ICCS'2021. *Springer Lecture Notes in Computer Science*, **12746**, 207–220 (2021).
[2] T. Varady, R. Martin: *Reverse Engineering*. In: G. Farin, J. Hoschek, M. Kim, M. (Eds.): Handbook of Computer Aided Geometric Design. Elsevier Science, Amsterdam, The Netherlands (2002).
[3] V. Raja, K.J. Fernandes: *Reverse Engineering: An Industrial Perspective*. Springer (2007).

[4] L. Gomes, O.R.P. Bellon, L. Silva: 3D reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters*, **50**, 3–14 (2014).

[5] M. Berger, A. Tagliasacchi, L.M. Seversky, P. Alliez, G. Guennebaud, J.A. Levine, A. Sharf, C.T. Silva: A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, **36**(1), 301–329 (2017).

[6] S.P. Lim, H. Haron: Surface reconstruction techniques: a review. *Artificial Intelligence Review*. **42**, 59–78 (2014).

[7] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, C. Silva: State of the art in surface reconstruction from point clouds. In: *Eurographics 2014-State of the Art Reports*, 161–185 (2014).

[8] A. Boltuc, E. Zieniuk: Parametric integral equation system (PIES) for solving problems with inclusions and non-homogeneous domains using Bézier surfaces. *Journal of Computational Science*, Article 101343 (2021).

[9] M. Kapturczak, E. Zieniuk: Modeling the boundary shape of the problems described by Navier–Lamé equations using NURBS curves in parametric integral equations system method. *Journal of Computational Science*, Article 101367 (2021).

[10] M.N.M. Othman, Y. Yusoff, H. Haron, L. You: An overview of surface reconstruction using partial differential equation (PDE). In: *IOP Conference Series: Materials Science and Engineering*, 012054 (2019).

[11] P. Gu, X. Yan: Neural network approach to the reconstruction of free-form surfaces for reverse engineering. *Computer-Aided Design*, **27**, 59-–64 (1995).

[12] R. Sharma, T. Schwandt, C. Kunert, S. Urban, W. Broll: Point cloud upsampling and normal estimation using deep learning for robust surface reconstruction. *Proc. of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2021*, Vol. 5, 70–79 (2021).

[13] A. Gálvez, A. Iglesias, J. Puig-Pey: Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. *Information Sciences*, **182**(1), 56–76 (2012).

[14] A. Gálvez, A. Iglesias: Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Information Sciences*, **192**, 174–192 (2012).

[15] X. Liu, M. Tang, J.H. Frazer: Shape reconstruction by genetic algorithms and artificial neural networks. *Engineering Computations*, **20**, 129—151 (2003).

[16] G. Sharma, D. Liu, S. Maji, E. Kalogerakis, S. Chaudhuri, R. Měch, Parsenet: A parametric surface fitting network for 3d point clouds. *Proc. of European Conference on Computer Vision*, Springer, 261–276 (2020).

[17] D. Lee, I. Quan, C. Wu, J. Wu, D. Tamir, N. Rishe, Optimizing B-Spline Surface Reconstruction for Sharp Feature Preservation. *Proc. of 10th Annual Computing and Communication Workshop and Conference, CCWC*, IEEE, 0359–0364 (2020).

[18] A. Dimitrov, M. Golparvar-Fard: Robust NURBS surface fitting from unorganized 3D point clouds for infrastructure as-built modeling. *Proc. of International Conference on Computing in Civil and Building Engineering*, 81–88 (2014).

[19] P. Bo, R. Ling, W. Wang, A revisit to fitting parametric surfaces to point clouds. *Computers & Graphics*. **36**, 534–540 (2012).

[20] H. Ugail, S. Kirmani: Method of surface reconstruction using partial differential equations. *Proc. of 10th WSEAS International Conference on Computers*, Athens, Greece,13–15 (2006).

[21] M. Rodrigues, A. Osman, A. Robinson: Partial differential equations for 3D data compression and reconstruction, *ADSA Advances in Dynamical Systems and Applications*, **8**, 303–315 (2013).

[22] M. Kazhdan, M. Bolitho, H. Hoppe: Poisson surface reconstruction. *Proc. of the Fourth Eurographics Symposium on Geometry Processing*, 61–70 (2006).

[23] M. Kazhdan, H. Hoppe: Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, **32**, 1–13 (2013).

[24] Y. Duan, L. Yang, H. Qin, D. Samaras: Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. *Proc. of European Conference on Computer Vision*, Springer, 238–251 (2004).

[25] E. Franchini, S. Morigi, F. Sgallari: Implicit shape reconstruction of unorganized points using PDE-based deformable 3D manifolds. *Numerical Mathematics: Theory, Methods and Applications*, **3**, 405–430 (2010).

[26] C. Linz, B. Goldlücke, M. Magnor, A point-based approach to PDE-based surface reconstruction. *Proc. of Joint Pattern Recognition Symposium*, Springer, 729–738 (2006).

[27] A. Kanazawa, S. Tulsiani, A.A. Efros, J. Malik: Learning category-specific mesh reconstruction from image collections. *Proc. of the European Conference on Computer Vision, ECCV*, 371–386 (2018).

[28] H. Kato, Y. Ushiku, T. Harada: Neural 3d mesh renderer. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 3907–3916 (2018).

[29] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4460–4470 (2019).

[30] A. Badki, O. Gallo, J. Kautz, P. Sen: Meshlet priors for 3d mesh reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2849–2858 (2020).

[31] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, D. Panozzo: Deep geometric prior for surface reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10130–10139 (2019).

[32] Z. Deng, J. Bednařík, M. Salzmann, P. Fua: Better patch stitching for parametric surface reconstruction. *Proc. of International Conference on 3D Vision, 3DV*, IEEE, 593–602 (2020).

[33] P.N. Azariadis: Parameterization of clouds of unorganized points using dynamic base surfaces. *Computer-Aided Design*, **36**, 607–623 (2004).

[34] H. Boukamcha, A.B. Amara, F. Smach, M. Atri: Robust technique for 3D shape reconstruction. *Journal of Computational Science*, **21**, 333–339 (2017).

[35] X.-F. Han, J.S. Jin, M.-J. Wang, W. Jiang, L. Gao, L. Xiao: A review of algorithms for filtering the 3D point cloud. *Signal Processing: Image Communication*, **57**, 103–112 (2017).

[36] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun: Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, DOI: 10.1109/TPAMI.2020.3005434 (*in press*).

[37] C.R. Qi, H. Su, K. Mo, L.J. Guibas: Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660 (2017).

[38] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, Y. Wang, Q. Fu, Y. Zou, A. Mian: Deep Learning based 3D Segmentation: A Survey. *ArXiv Preprint ArXiv:2103.05423.* (2021).

[39] D. Bazazian, M.E. Parés: EDC-Net: Edge Detection Capsule Network for 3D Point Clouds. *Applied Sciences*, **11**, Article 1833 (2021).

[40] W. Ma, J.-P. Kruth: Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, **27** 663–675 (1995).

[41] J. Barhak, A. Fischer: Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. *IEEE Transactions on Visualization and Computer Graphics*, **7**, 1–16 (2001).

[42] T. Wagner, T. Michelitsch, A. Sacharow: On the design of optimizers for surface reconstruction. *Proc. of the 2007 Genetic and Evolutionary Computation Conference—GECCO2007*, London, England, 2195—2202 (2007).