IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2018, accepted December 31, 2018, date of publication January 11, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891959

3D Incomplete Point Cloud Surfaces Reconstruction With Segmentation and Feature-Enhancement

MEILI WANG[1,2,3], MING JU[1], YULING FAN[1], SHIHUI GUO[4], MINGHONG LIAO[4], HUIJUN YANG[1,2,3], DONGJIAN HE[2,3,5], AND TAKU KOMURA[6]

[1]College of Information Engineering, Northwest A&F University, Yangling 712100, China
[2]Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling 712100, China
[3]Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, China
[4]School of Software, Xiamen University, Xiamen 361005, China
[5]College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China
[6]School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB, U.K.

Corresponding author: Shihui Guo (guoshihui@xmu.edu.cn)

This work was partially funded by Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China (2018AIOT-09). National Natural Science Foundation of China (61702433), Key Research and Development Program of Shaanxi Province (2018NY-127). The authors acknowledge the Shenzhen Key Lab of Visual Computing and Visual Analytics for the source data and the models.

ABSTRACT Raw point cloud data are often noisy, superfluous, and with topological defects, such as holes. These issues cause inaccurate geometric representation in 3D reconstruction. As a result, surface reconstruction from point cloud data is a highly challenging problem. In this paper, we address the aforementioned issues by taking advantage of the embedded information of segmentation, skeletonization, and user guidance. First, we pre-process the point cloud data with three steps, relocating each point, upsampling the point data, and optimizing normals to enhance the features and geometric details; second, a segmentation method converts the input cloud into separate parts; finally, we construct curve skeletons for each part and guide the surface reconstruction with minimal user interaction, where the parts of refined smooth shapes are fused to generate the final results. The comparison studies confirmed that the proposed method is able to produce state-of-the-art results in terms of preserving sharp features, handling missing data, and requiring minimal user intervention.

INDEX TERMS Surface reconstruction, segmentation, enhanced features, curve skeleton.

I. INTRODUCTION

Fully developed three-dimensional (3D) scanners can produce 3D point cloud representing complex models, which is an important data source of computer graphics research. Nowadays, point cloud can be easily acquired for our purposes, but the data is usually noisy, sparse and unorganized. A large amount of existing works have been paid to the reconstruction of meaningful surface models from the point cloud in the past three decades [1]–[3]. However, surface reconstruction from incomplete point cloud is still a challenging and unresolved problem [4].

Our work is inspired by the observation that the points of one division share the same characteristics and represent a primitive substructure. Segmentation is considered useful for analyzing various aspects of a scene such as locating and recognizing parts, classification and feature extraction. A recent work used the deep neural network (PointNet [5]) to tackle the problem of segmenting point cloud. Although there are many segmentation and reconstruction algorithms [5]–[7], few works utilize the segmentation information in surface reconstruction, in particular for challenges of handling incomplete point cloud. Meanwhile, the majority of existing works requires intensive user-interactive editing in the pipeline of surface reconstruction [8], [9].

Based on the aforementioned motivations, we propose a novel method that integrates both segmentation and curve skeletonization to accomplish the goal of surface reconstruction from incomplete point cloud. Our method advances the state-of-the-art by its performance of preserving sharp features, handling missing data and requiring minimal user intervention. Specifically, the novelty of this work lies in:

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

- We combine the extracted information of segmentation and skeletonization to address the challenges of mesh reconstruction from incomplete point cloud. This is, to the best of our knowledge, the first work to take advantage of these two aspects in the domain of point cloud reconstruction and achieve the state-of-the-art performance.
- We propose an anisotropic operator for point cloud simplification. This operator applies heterogeneous transformation, attraction or repulsion, to areas with different densities. This leads to a better uniform distribution after the process of simplification and lays the critical foundation for the follow-up procedures.
- We propose a method of prioritized feature enhancement to insert points along sharp boundaries of low density. This method improves the result of the existing method by preserving and enhancing the sharp edges in the original point cloud.

II. RELATED WORK

A. POINT CLOUD SEGMENTATION

3D point cloud segmentation classifies parts of point cloud with the same properties into the same region. Nguyen and Le [6] presented a detailed survey on existing segmentation methods, which include edge-based methods, region-based methods and model-based methods. They analyzed and discussed the advantages and disadvantages of these segmentation methods for point cloud. Gomes *et al.* [3] used a 3D moving fovea to process parts of a scene with different levels of resolution, which could easily recognize and identify objects in point cloud. Ochmann *et al.* [10] defined the reconstruction task as a labeling problem. They applied point cloud segmentation to define content into rooms and the outside area as a pre-processing filter to accelerate their computing and simplify the processing, which inspired our approach of applying segmentation for full reconstruction. Aliaga *et al.* [7] introduced a weighted minimum set cover to maximize the repetition of similar segments to simultaneously detect similarities especially for existing 3D architectural models. Qi *et al.* [5] proposed a deep neural network PointNet to segment point cloud. Their method provided a unified approach to object classification, part segmentation and semantic segmentation. Song *et al.* [11] introduced SSC-Net, an end-to-end 3D ConvNet, to produce a complete 3D voxel representation from a single-view depth map. Different from the aforementioned works, we integrate the segmentation information as part of the reconstruction process.

B. CURVE SKELETONS

A few methods [12]–[14] have been developed for curve skeleton extraction from incomplete point cloud. Guo *et al.* [13] and Andrei *et al.* [15] evolved deformable snakes based on surface tension control or smoothness priors for topological and geometric reconstruction. Tagliasacchi *et al.* [12] and Huang *et al.* [14] both obtained medial curve skeletons and proposed smooth cross-sectional curve interpolation to follow the skeletons for surface construction. Tagliasacchi *et al.* [16] extended curve skeletons with medial sheets to reconstruct non-cylindrical geometry. In existing works, surface reconstruction with incomplete data does not primarily consider sharp features. Inspired by the fact that a curve skeleton is an abstraction of data that can tolerate missing data, we have extended the existing method to radical cases in which a large amount of data is missing. Using the segmentation results, our approach allows fewer user interaction for guidance to preserve and enhance sharp features.

C. 3D RECONSTRUCTION

The task of reconstruction is from point cloud to automatically construct mesh models, which are convenient to store, edit, render, and animate. The core aims of reconstruction are to recover missing data and preserve model shapes. In the past 20 years, surface reconstruction technology has made considerable progress [2], ranging from explicit reconstruction to implicit function-based reconstruction. Recent works have focused on priori knowledge-oriented model reconstruction and interactive reconstruction in response to serious deficiencies in point cloud. Ochmann *et al.* [10] presented an automatic approach to the parametric reconstruction of 3D building models from indoor point cloud. It took human inputs as an initial guess, and later treated the reconstruction problem as a labeling problem solved by energy minimization. Zhang *et al.* [17] introduced a statistical model using a Poisson distribution to extract feature points from point cloud. In contrast to the model of Ochmann *et al.* [10], their approach did not require any prior knowledge of the surface. However, their reconstruction is vulnerable to data noise and sampling quality, which have been addressed in our proposed method to enhance the reconstruction result. Sameer [1] reconstructed architectural scenes from sparse 3D point cloud. Yin *et al.* [8] considered 3D objects as a combination of generalized cylinders and proposed the Morfit algorithm to reconstruct generalized cylinders. Similar to their curve based solutions, we adopt the curve skeleton as an important interlayer and Morfit algorithm developed in [8] for our reconstruction. Our method provides higher quality surfaces in terms of smoothness and feature preservation compared to previous ones.

III. OVERVIEW

An overview of the proposed approach is shown in Figure 1. The input point cloud (Figure 1(a)) is typically unoriented, unevenly distributed, incomplete and contains noise and outlier. The approach includes two steps: pre-processing and reconstruction. The pre-processing step generates a new point set that enhances the sharp features and preserves the potential shape of the raw data. The pre-processing step (see Section IV) includes three substeps: simplification (Figure 1(b)), upsampling (Figure 1(c)), and normal optimization (Figure 1(d)). Simplification relocates each point to

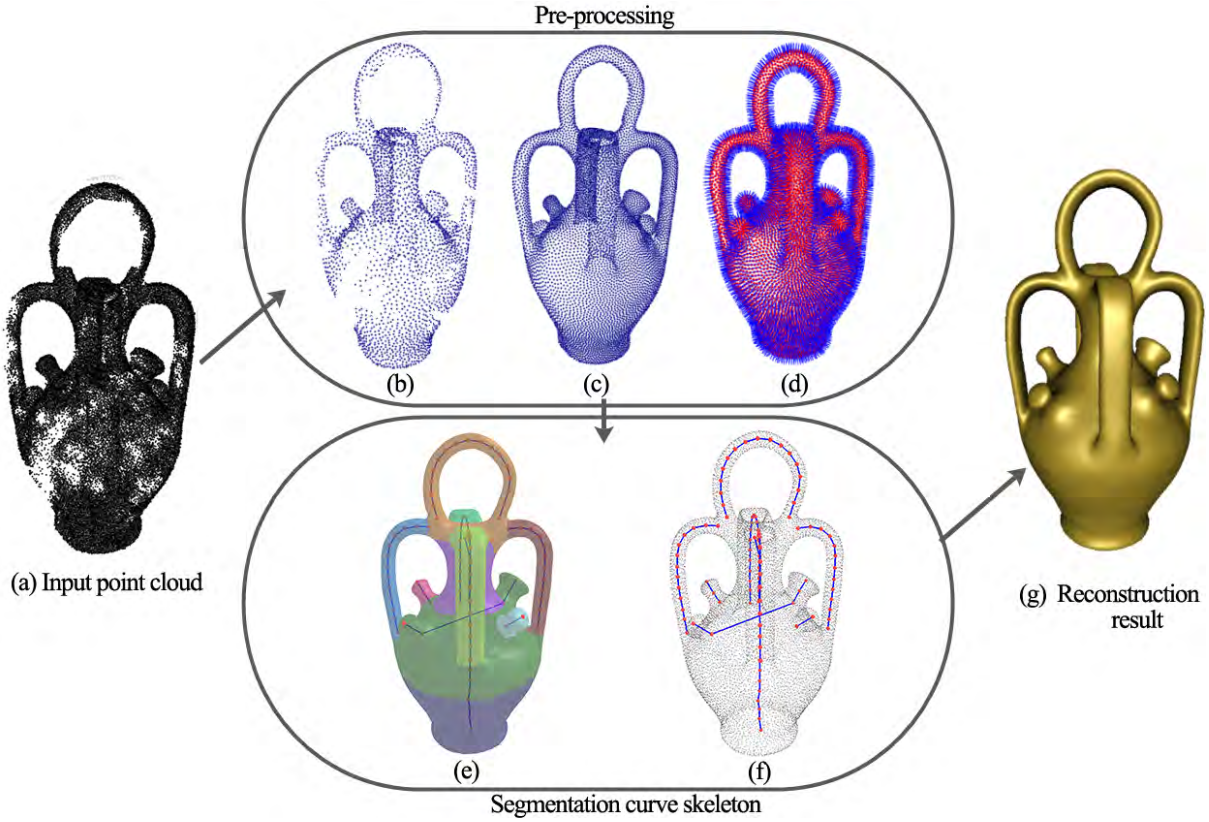


FIGURE 1. Surface reconstruction. (a) Incomplete point cloud input. (b) Points simplified by redistributing the center of the points. (c) Enhanced features. (d) Normals optimized by iterative updates. (e) Segmented results. (f) Curve skeleton. (g) Reconstruction.

the center of its local neighborhood, which is equivalent to a smoothing filter and can reduce undesired noise. Upsampling improves the quality of point cloud data by selectively inserting new points to enhance features such as sharp edges. Normal optimization produces accountable normals and is specifically tailored to consider sharp features by clustering candidates into different groups for optimization. After the pre-processing, we segment the point cloud into meaningful parts (Figure 1(e)), each of which contains only primitive geometry in terms of its topological structure so that we may use a general cylinder to represent each part in the next step. The curve skeletons are identified for the segmented parts by finding the L1 media skeletons (Figure 1(f)). Finally, we use the curve skeletons and point information to generate general cylinder structures and stitch them together to create fine surfaces (Figure 1(g)) with interactive user editing.

IV. PRE-PROCESSING

A. SIMPLIFICATION

The original point cloud is first simplified with our anisotropic operator. This operator is an improved work of Locally Optimal Projection (LOP) operator [18] by applying heterogeneous transformation, attraction or repulsion, to adjacent areas with different densities. Given the original captured point set $P = \{p_j\}_{j\in J} \subset R^3$, our operator projects an arbitrary point set $X = \{x_i\}_{i\in I} \subset R^3$ onto the original captured set P, and minimizes the sum of distances from projected set $Q = \{q_i\}_{i\in I}$ to set P. This can be represented as:

$$Q^{k+1} = \underset{X=\{x_i\}_{i\in I}}{argmin} \{E_s(X, P, Q^k) + E_r(X, Q^k)\} \tag{1}$$

E_s is an attractive energy that attracts the points in set Q towards the local medians of original set P, while E_r is a repulsive energy that strives to equally distribute the q_i. $|I|$ and $|J|$ are the sizes of sets X and P respectively, and i and j denote the indices of the respective sets. An illustration of Equation 1 is given in Figure 2.

Standard LOP operator is likely to produce non-uniform distribution of cloud points, creating the sparse region highlighted in the model of Japanese Lady (Figure 3(b)). We overcome this problem by considering the anisotropic point density of adjacent regions. Attraction and repulsion energy are defined to relocate points from regions of higher density to ones of lower density. This improves the uniform distribution of the point cloud, in comparison to the original one. Our algorithm updates each $q_i^{(k)} \in Q^{(k)}$ in iteration k using the following equations. The first iteration is:

$$q_i^{(1)} = \frac{\sum_{j\in J} p_j \varsigma(n_j, p_j - q_i^{(0)})}{\sum_{j\in J} \varsigma(n_j, p_j - q_i^{(0)})} \tag{2}$$

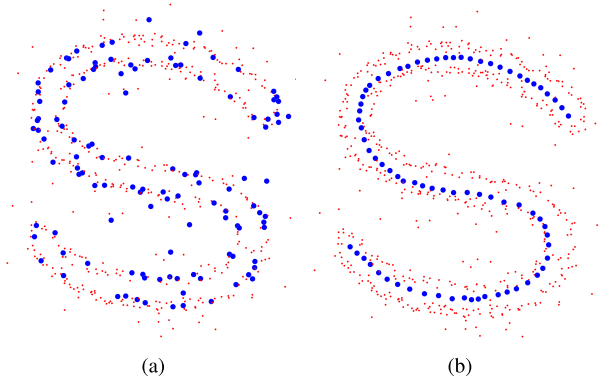


FIGURE 2. Projected set generated with attraction and repulsion defined by Equation 1. (a) Arbitrary set X. (b) Projected set Q.

We define the weighted densities for each point p_j in P and q_i in Q during the iteration k by

$$\begin{aligned} \rho_j &= 1 + \sum\nolimits_{j' \in J \setminus \{j\}} \varsigma(n_j, p_j - p_{j'}) \\ w_i^{(k)} &= 1 + \sum\nolimits_{i' \in I \setminus \{i\}} \theta(||q_i^{(k)} - q_{i'}^{(k)}||), \quad k = 0, 1, 2, ... \end{aligned} \tag{3}$$

where n_j is the normal vector of the point p_j. The normal vector can be calculated by PCA. ρ and w are the local adaptive weighted densities, which allow higher densities around sharp features. Finally, the projected result for $q_i^{(k)}$ can then be defined as

$$\begin{aligned} q_i^{(k+1)} &= \sum_{j \in J} p_j \frac{\alpha_j^i / \rho_j}{\sum_{j \in J} \alpha_j^i / \rho_j} \mu \sum_{i' \in I \setminus \{i\}} ||q_i^{(k)} - p_j|| \frac{w_{i'}^{(k)} \beta_{i'}^i}{\sum_{i' \in I \setminus \{i\}} w_{i'}^{(k)} \beta_{i'}^i} \\ \alpha_j^i &= \frac{\theta(||q_i^{(k)} - p_j||)}{||q_i^{(k)} - p_j||}, \\ \beta_{i'}^i &= \frac{\theta(||q_i^{(k)} - q_{i'}^{(k)}||) \eta(||q_i^{(k)} - q_{i'}^{(k)}||)}{||q_i^{(k)} - q_{i'}^{(k)}||} \end{aligned} \tag{4}$$

where $||.||$ is the L_2-norm, and μ is the weight parameter which balances attraction and repulsion. ρ and w are the adaptive density weights, and ς decides the normal directions indicating the approximate location of the edges. In our experiments, we set $\mu = 0.4$, $\eta(r) = -r$. θ and ς can be expressed as follows:

$$\begin{aligned} \theta(r) &= e^{-r^2/(h/4)^2} \\ \varsigma(n_j, p_j - p_{j'}) &= e^{-(n_i^T (p_j - p_{j'}))^2 / \sigma_p^2} \\ \sigma_p &= dist_{bb} / \sqrt{|J|} \end{aligned} \tag{5}$$

where θ is a rapidly decreasing smooth weight function with support radius h, which defines the size of the influence neighborhood and can be adjusted adaptively. We use a default value of $h = 4\sqrt{dist_{bb}/|J|}$ where $dist$ is the diagonal length of set P's bounding box.

Figure 3 shows the results of simplifying the Japanese Lady model using our proposed method and the standard LOP [18].

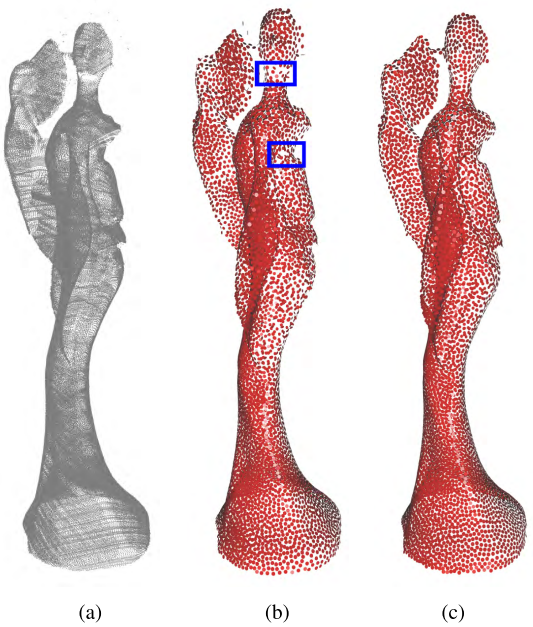


FIGURE 3. Comparison of point cloud simplification by different methods on the *Japanese Lady* model. (a) Raw data. (b) Our Method.

In our method, owing to the density weights ρ and w, the attraction is relaxed by ρ with respect to the attractive energy, and the repulsion in dense areas is strengthened by weight w with respect to the repulsive energy. Hence, our algorithm obtains more uniformly-spaced points than the standard LOP.

B. FEATURE ENHANCEMENT

We propose a method of prioritized feature enhancement to insert points along sharp boundaries of low density. For each insertion, this method adds a new point $b + dn$ in two steps: finding the near-sparsest insertion base b and optimizing the projection distance d to move the point onto the latent surface. Different from existing methods in feature enhancement [19], we aim to insert more points along sharp boundaries and at low-density regions. To achieve this goal, we introduce a priority factor as:

$$\begin{aligned} P(s_i) = \max_{s_{i'} \in N s_i} (2 - n_{s_i}^T n_{s_{i'}})^\rho D(\frac{s_i + s_{i'}}{2}, s_{i'}) \\ minimize_{s_{i'} \in N s_i} D(b, s_{i'}) = ||b - s_{i'} - n_{s_{i'}}^T (b - s_{i'}) n_{s_{i'}}|| \end{aligned} \tag{6}$$

where b is the base location, s_i is an existing point and the set N_{s_i} is its neighbors. Higher priority ρ is given to place points along sharp edges. In order to get the results of sharp feature enhancement, we use the default value $\rho = 3$.

To minimize the weighted projection distance, we introduce the anisotropic neighbors to enhance features. Finally, the calculation of projection distance d is as follows:

$$d(b, n) = \frac{\sum_{s_i \in N_b} (n^T (b - s_i)) \varsigma(n, b - s_i) \varphi(n, n_{si})}{\sum_{s_i \in N_b} \varsigma(n, b - s_i) \varphi(n, n_{si})} \tag{7}$$

FIGURE 4. Comparison results of feature enhancement by different methods on the *plant* model. (a) Simplified points. (b) Our method.

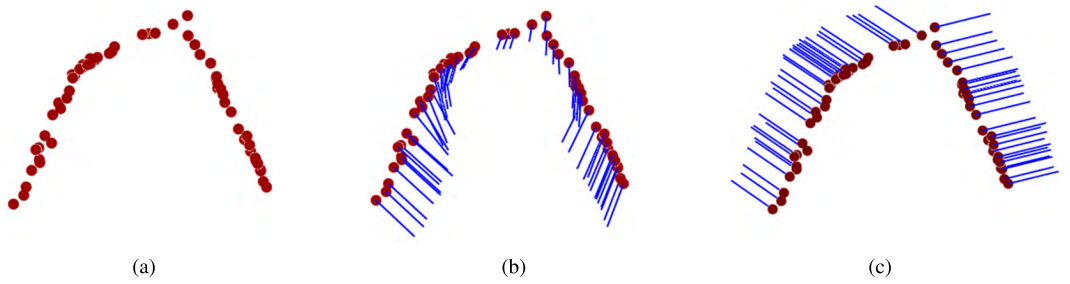


FIGURE 5. Normal optimization. (a) Sample model. (b) PCA normal. (c) Anisotropic normal.

where $\varphi(n, n_i) = e^{-(\frac{1-n_i^T n_{i'}}{1-\cos\alpha})^2}$, $\alpha = 15^0$ and ς can be obtained from the Equation (5).

In comparison to the uniform distribution of inserted points from the EAR method [19], our method inserts more points at the low-density region and preserves the sharp edges (Figure 4).

C. NORMAL OPTIMIZATION

Standard method for computing the normal direction of a point, such as Principal Component Analysis(PCA) [19], is to utilize the information of its local neighbors. Errors may occur when the point is near sharp areas (see Figure 5(b)); we therefore estimate the normal based on an anisotropic neighborhood, as in the framework of bilateral normal smoothing [20].

For an existing point $z_i = (p_i, n_i)$, our goal is to minimize the normal differences between all points on the surface and their neighbors:

$$f(p_i, n_i) = \sum_{z_{i'} \in N z_i} ||n_i - n_{i'}||^2 \theta(||z_i - z_{i'}||)\varphi(n_i, n_{i'}) \quad (8)$$

This can be achieved by iteratively updating n_i with

$$n_i \leftarrow \frac{\sum_{z_{i'} \in N z_i} ||n_i - n_{i'}||^2 \varsigma(n_i, z_i - z_{i'})\varphi(n_i, n_{i'})n_{i'}}{\sum_{z_{i'} \in N z_i} ||n_i - n_{i'}||^2 \varsigma(n_i, z_i - z_{i'})\varphi(n_i, n_{i'})} \quad (9)$$

where ς, φ can be obtained from Equation (5) and (6) respectively. The results in Figure 5(c) show that the normals are well aligned in the appropriate directions.

V. SURFACE RECONSTRUCTION

A. SEGMENTATION

We adopt a self-adaptive segmentation algorithm of two main steps: automatic selection of the centers according to the extracted features and segmentation of the points according to the centers [21].

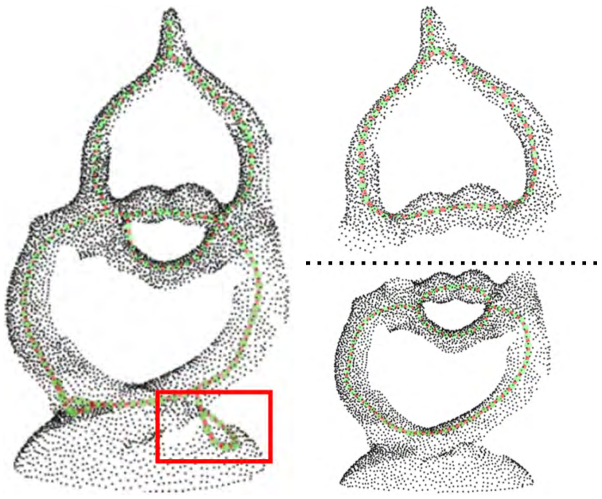
The centers can be selected with their representativeness and diversity values. Representativeness of a point can be measured by the relative density in comparison with its neighbors. The diversity can be measured by computing the minimum distance between a point and other points with higher densities. A point is selected as a center if it has a high density compared to its surrounding neighbors with lower density and a large diversity with respect to other points with high density.

We define the representativeness value δ and the diversity value ν as:

$$\begin{aligned} sp_i^{Att} &= \log \nu_i + \log \delta_i \\ \nu_i^{Rep} &= 1 + \sum_{i' \in I} \theta(||p_{i'} - p_i||) \\ \delta_i^{Div} &= \min_{i'<i} ||p_{i'} - p_i|| \end{aligned} \quad (10)$$

where θ is the same as in Equation (5).

Next, we sort the points in a descending order according to the value of sp. Center points are efficiently extracted from points with higher rankings. Our method can effectively select centers that are both dense and distant from other centers [21]. After obtaining the centers, each remaining point is assigned to the same cluster as its nearest neighbor of higher density.



(a) (b)

FIGURE 6. Curve skeleton extraction. (a) Curve skeleton with unsegmented point cloud. The red rectangle denotes undesired skeletons. (b) Curve skeleton of segmented point cloud.

B. L1-CURVE SKELETON

To obtain the geometric representation, the L1-medial skeleton [14] is used to represent the curve skeleton. It iteratively projects points to produce the L1-medians of the local neighborhoods. We use the following definition to obtain a set of projected points $M = \{m_l\}_{l\in L}$ in each segmented clusters:

$$m_l^{k+1} = \frac{\sum_{i\in I} z_i \alpha_i^l}{\sum_{i\in I} \alpha_i^l} + \mu\sigma_l^k \frac{\sum_{l'\in L\setminus\{l\}} (m_l^k - m_{l'}^k)\beta_{l'}^l}{\sum_{l'\in L\setminus\{l\}} \beta_{l'}^l} \quad (11)$$

where

$$\beta_{l'}^l = \frac{\theta(||m_l^k - m_{l'}^k||)}{||m_l^k - m_{l'}^k||^2}, \quad \alpha_i^l = \frac{\theta(||m_l^k - m_i^k||)}{||m_l^k - m_i^k||},$$

$\sigma_l = \frac{\lambda_l^2}{\lambda_l^0+\lambda_l^1+\lambda_l^2}$, λ is eigenvalue, and $\lambda_l^0 \le \lambda_l^1 \le \lambda_l^2$.

The aforementioned iterative projection produces a set of points $M = \{m_l\}_{l\in L}$, which can form a skeleton of the underlying shape. It can be seen that our method improves the standard method of skeleton extraction by introducing the segmentation procedure and removing the undesired skeleton parts (Figure 6 marked in red rectangle).

After obtaining the skeleton of data, we apply an interactive Morfit [8] to reconstruct each part of the segmentation. The separate parts are fused as an ensemble of general cylinders around the curve skeleton. The Morfit [8] algorithm allows users to edit the model on a larger scale. For the L1-center skeleton of the model, users can adjust the model by disconnection, connection, pruning, extension, and deformation, and hence unambiguously express the model topology.

VI. RESULTS AND DISCUSSIONS

A. IMPLEMENTATION AND PERFORMANCE

Our method is implemented in C++ on a standard PC environment (CPU: Intel Core i5-7300HQ 2.50GHz, memory: 8 GB, GPU: NVIDIA GeForce GTX 1050). Figure 7 shows the reconstruction results on four models: hand, horse, ant and vase. Number of user interactions are included in the figure captions. Models of hand, ant and vase contain incomplete and sparse regions, which our method can successfully handle. Table 1 lists the timecost of segmentation, skeletonization and mesh reconstruction for all demonstrated examples in this work. Note that for the purpose of consistency, all models are preprocessed to the same number of cloud points. The results show that the procedures of segmentation and mesh reconstruction cost significantly less time than the procedure of skeletonization.

TABLE 1. Timecost of demonstrated examples. Unit: second.

	Segmentation	Skeletonization	Mesh Reconstruction
Hand	2.15	9.71	3.38
Horse	2.08	23.35	4.21
Ant	2.04	47.24	6.74
Vase	2.13	31.47	1.63
Lady	0.38	10.29	1.05
Mouse	2.05	17.52	7.34
Plant	2.24	10.56	2.37

B. COMPARISON OF SEGMENTATION RESULTS WITH POINTNET

In Figure 8 we compared our method with the PointNet [5] method. Our method produces semantically-meaningful segmentation for the models of mouse, hand, bird, pig and laptop (Figure 8(a)-(e)). For the handbag case (Figure 8(f)), our method divides the handle strap into two halves, while the PointNet method regards it as one part. PointNet successfully divides the laptop into two parts (the same as our method) but produces inappropriate segmentation results for cases of mouse, hand, bird and pig. The results reveal that our method does not require any training data and thus is not limited to specific models, while PointNet can face challenges for models which are excluded from its training dataset.

Table 2 lists the timecost of segmentation by PointNet and our method. The method of PointNet is implemented in Python on a standard PC platform (CPU: Intel Core i7-6700 CPU 3.40GHz, memory: 16GB, GPU: NVIDIA GeForce GTX 1080ti). Although PointNet runs on a platform with higher computing capability than our method, the timecost of our method is still less than that of PointNet. Additionally, the training of PointNet costs over 12 hours on the aforementioned platform.

C. COMPARISON OF RECONSTRUCTION RESULTS WITH POISSON AND MORFIT

In this section, we compared the reconstruction results obtained by our method with existing methods, including Poisson [17] and Morfit [8].

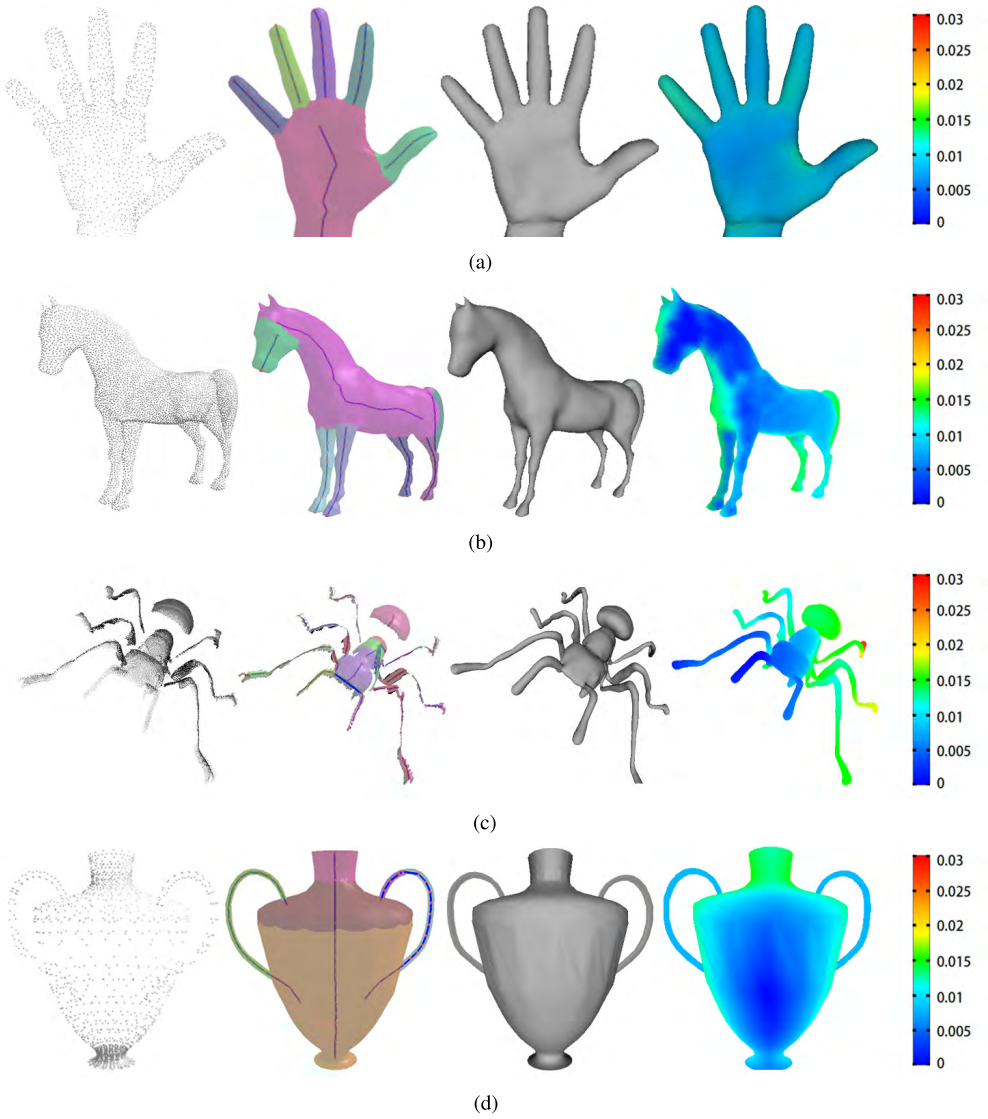


FIGURE 7. Reconstruction results on different models. The columns from left to right show the raw data, curve skeletons of each segmented part (indicated by different colors), reconstruction results and the error plots. (a) Hand user interaction: #BE = 1, #CE = 1, #PS = 0, #SS = 0; maximum error = 0.013. (b) Horse user interaction: #BE = 4, #CE = 4, #PS = 2, #SS =2; maximum error = 0.028. (c) Ant user interaction: #BE = 8, #CE = 5, #PS = 11, #SS =1; maximum error = 0.027. (d) Vase user interaction: #BE = 2, #CE = 0, #PS = 4, #SS =0; maximum error = 0.013.

TABLE 2. Timecost comparison of segmentation between PointNet and our method. Unit: second.

	mouse	hand	bird	pig	laptop	bag
PointNet	3.17	3.25	3.15	3.16	2.54	2.49
Our Method	2.05	2.15	2.08	1.98	1.71	1.88

In Figure 9, the results of the lady model show that our algorithm can deal with incomplete point cloud (Figure 9(e)), which provides the most faithful reconstruction result with only one path stroke and editing. Figure 9(a)-(d) are the raw data, the extracted skeleton model, the reconstructed model using the Poisson-based algorithm and the reconstructed model using the Morfit algorithm. For the left arm of the lady body, it can be seen that the shape from our method is better fitted to the original model than the other two methods with enhanced details. For the enlarged part, our reconstruction result is smoother.

In Figure 10, we presented the experiment results on the mouse model. The results show that the Poisson method fails to reconstruct the arms, tail and whiskers of the mouse due to sparse cloud points. The Poisson method also introduces erroneous part on the non-existent nose part. In comparison, the Morfit method is capable of reconstructing the majority geometry of the original point cloud. As can be seen from Figure 10(e), our method can produce distinct edges and contours of the eyelid and foot toes (highlighted in blue box).

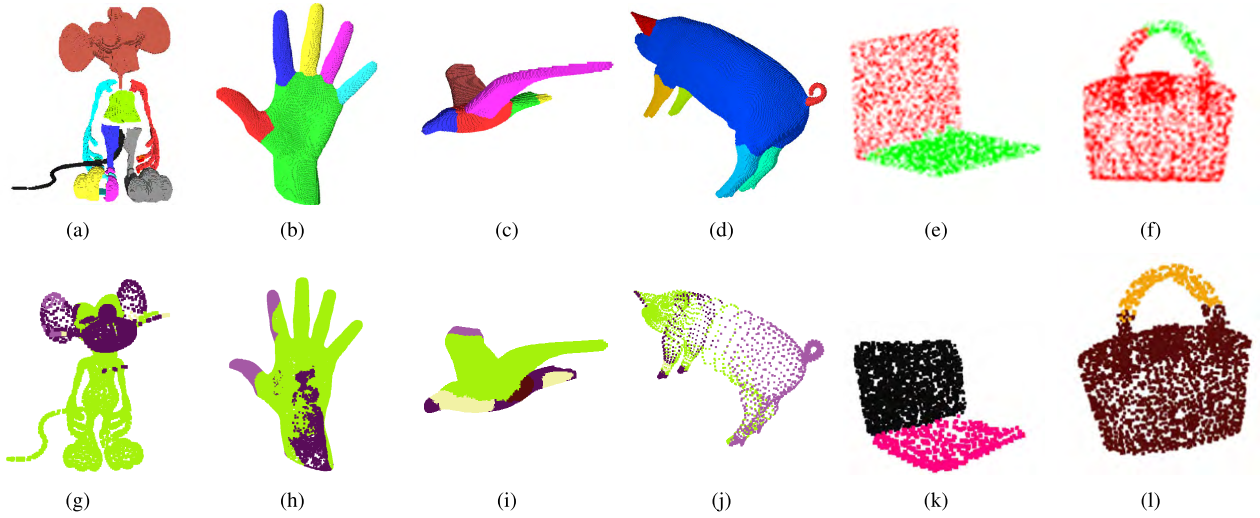


FIGURE 8. Comparison of segmentation results between our method and PointNet. (a)-(f) our method. (g)-(l) PointNet.

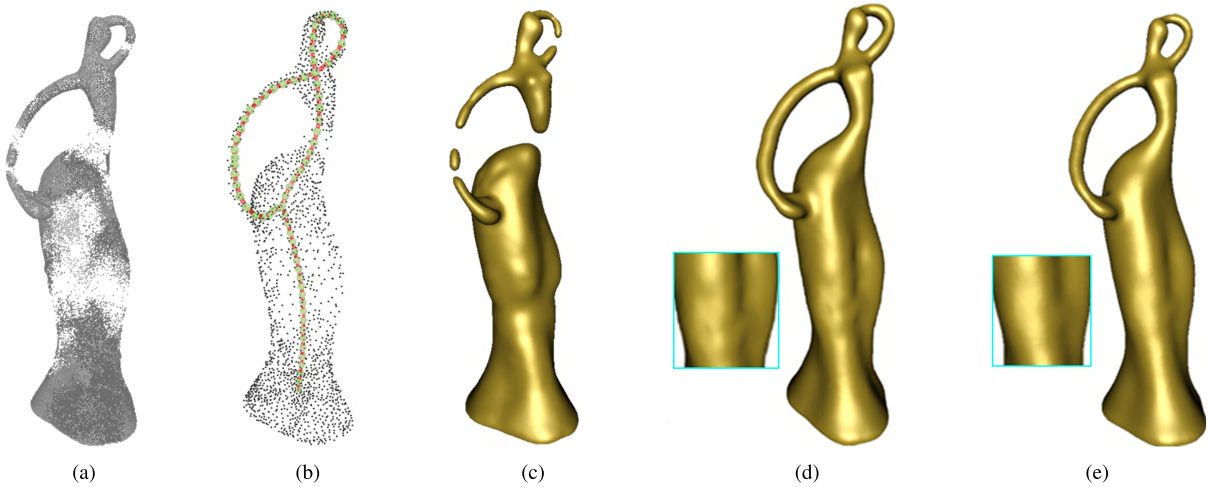


FIGURE 9. Comparison results on the Lady model. (a) Raw data. (b) Skeleton. (c) Poisson. (d) Morfit. (e) Our Method.

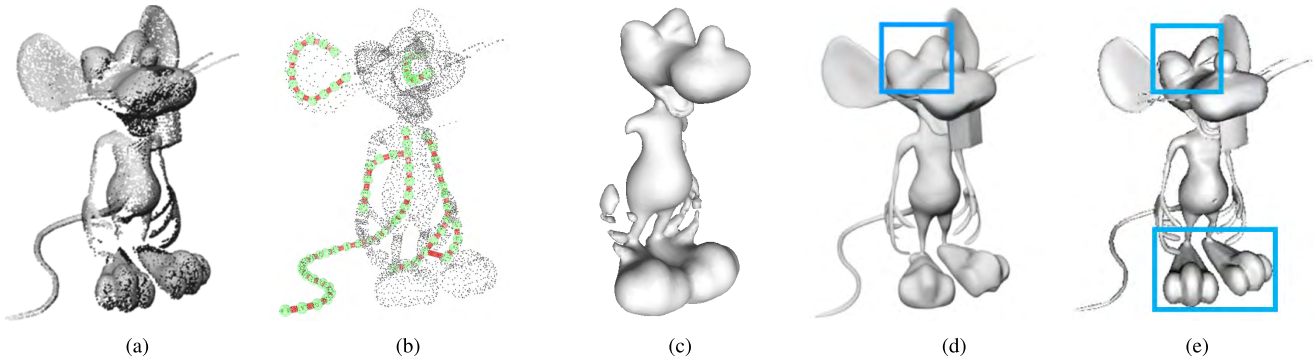


FIGURE 10. Comparison results on the mouse model. (a) Raw data. (b) Skeleton. (c) Poisson. (d) Morfit. (e) Our Method.

Obviously, the details of the result are closer to the original model and represent a more accurate surface.

Figure 11 compares the results of our method and Morfit [8] of a plant model obtained from [14]. For the plant model, Figure 11(b) shows the reconstructed results of each part after the segmentation of the point cloud. Our algorithm achieves better performance than Poisson [17] and Morfit [8] methods by keeping the sharp features on the leaves.

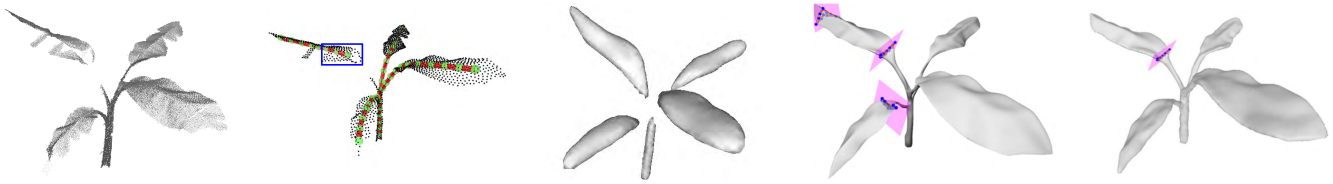


FIGURE 11. Comparison results on the plant model. (a) Raw data. (b) Skeleton. (c) Poisson. (d) Morfit. (e) Our Method.

TABLE 3. Comparison of the number of user interactions between Morfit and our method. #BE: number of skeletal branches edited/added/removed; #CE: number of profile curves edited; #PS: number of strokes to indicate sweep paths; #SS: number of sharpening strokes.

	Lady (Morfit)	Lady (Our)	Mouse (Morfit)	Mouse (Our)	Plant (Morfit)	Plant (Our)
#BE	0	0	0	0	0	0
#CE	0	0	6	0	3	0
#PS	3	1	6	1	4	1
#SS	0	0	0	0	0	0

D. USER INTERACTION IN EDITING PROCEDURE

Table 3 compares the number of user interactions in editing procedure for the models of Lady, mouse and plant. It can be seen that our proposed method requires a considerably smaller number of interactions than Morfit for all models. For example, for the Lady model, our method requires only 1 interaction while Morfit requires 3 interactions because our method segments the model into meaningful parts and then reconstructs them accordingly. It performs even better on the mouse and plant models since both models contain many curves and branches. Our method requires only 1 stroke to indicate sweep path while Morfit requires 12 editings for the mouse model and 7 interactions for the plant model. Figure 11(d) and 11(e) demonstrates part of the interactions highlighted in purple boxes.

VII. CONCLUSION

The goal of this work is to construct a mesh surface from a point cloud. Our method first includes a pre-processing step, which improves the quality of point cloud distribution and enhances sharp features. The main contribution of this work is to utilize the embedded information from both segmentation and skeletonization. The segmentation assists in the task of skeleton extraction by separating the original point cloud and improving the accuracy of the extracted skeleton. The curve skeleton informs the final procedure of mesh reconstruction, in particular for scenarios of incomplete point cloud. Our approach requires fewer user interactions without degenerating the quality and efficiency of the output. For the final outputs, our results present more feature details, whereas existing methods such as the Poisson-based method and Morfit [8] tend to overfit the surface and smooth out subtle details.

One of the limitations from our proposed method is the extraction of representative curve skeletons. For a model with high structure complexity, the method still requires additional user edits and produces excessive segments. Our future work is to investigate the case of objects with complex shapes and identify suitable solutions that are effective in terms of both usability and accuracy. In addition, the pipeline of our method is composed of a set of individual tools. Another future task is to automate this process and provide a complete framework to non-professional users. Exploring an end-to-end solution with the technique of deep neural network is worth our future efforts.

REFERENCES

[1] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Schematic surface reconstruction," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1498–1505.
[2] M. Berger, A. Tagliasacchi, and S. Lee, "State of the art in surface reconstruction from point clouds," in *Proc. Eurograph. State Art Rep.*, 2014, vol. 1, no. 1, pp. 161–185.
[3] R. Gomes *et al.*, "Efficient 3D object recognition using foveated point clouds," *Comput. Graph.*, vol. 37, no. 5, pp. 496–508, 2013.
[4] J. Mei, L. Zhang, S. Wu, Z. Wang, and L. Zhang, "3D tree modeling from incomplete point clouds via optimization and L_1-MST," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 5, pp. 999–1021, 2017.
[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, p. 4.
[6] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *Proc. IEEE 6th Conf. Robot., Automat. Mechatron. (RAM)*, Nov. 2013, pp. 225–230.
[7] D. Aliaga, G. Daniel, and B. Bedrich, "Coupled segmentation and similarity detection for architectural models," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 104.
[8] K. Yin, H. Huang, H. Zhang, M. Gong, D. Cohen-Or, and B. Chen, "Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–12, 2014.
[9] T. Itoh, "Surface reconstruction with an interactive modification of point normals," in *Proc. Int. Symp. Comput. Modeling Objects Represented Images*. Springer, 2010, pp. 263–274.
[10] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Comput. Graph.*, vol. 54, pp. 94–103, Feb. 2015.

[11] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 190–198.
[12] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 71.
[13] G. Li, L. Liu, H. Zheng, and N. J. Mitra, "Analysis, reconstruction and manipulation using arterial snakes," *ACM Trans. Graph.*, vol. 29, no. 6, p. 152, 2010.
[14] H. Huang *et al.*, "L_1-medial skeleton of point cloud," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 65.
[15] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly curve-skeleton computation for 3D shapes," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 323–328, 2007.
[16] A. Tagliasacchi, M. Olson, H. Zhang, G. Hamarneh, and D. Cohen-Or, "VASE: Volume-aware surface evolution for surface reconstruction from incomplete point clouds," *Comput. Graph. Forum*, vol. 30, no. 5, pp. 1563–1571, 2011.
[17] Y. Zhang, G. Geng, X. Wei, S. Zhang, and S. Li, "A statistical approach for extraction of feature lines from point clouds," *Comput. Graph.*, vol. 56, pp. 31–45, May 2016.
[18] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 22.
[19] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, 2013, Art. no. 9.
[20] Z. Ouml, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 493–501, 2009.
[21] Y. Fan, M. Wang, N. Geng, D. He, J. Chang, and J. J. Zhang, "A self-adaptive segmentation method for a point cloud," *Vis. Comput.*, vol. 34, no. 5, pp. 659–673, 2018.

Authors' photographs and biographies not available at the time of publication.