Noname manuscript No.
(will be inserted by the editor)

A self-adaptive segmentation method for a point cloud

Yuling Fan · Meili Wang · Shaojun Hu · Dongjian He · Jian Chang · Jian J Zhang

Received: date / Accepted: date

Abstract The segmentation of a point cloud is one of the key technologies for three-dimensional reconstruction, and the segmentation from three-dimensional views can facilitate reverse engineering. In this paper, we propose a self-adaptive segmentation algorithm, which can address challenges related to the region-growing algorithm, such as inconsistent or excessive segmentation. Our algorithm consists of two main steps: automatic selection of seed points according to extracted features and segmentation of the points using an improved region-growing algorithm. The benefits of our approach are the ability to select seed points without user intervention and the reduction of the influence of noise. We demonstrate the robustness and effectiveness of our algorithm on different point cloud models and the results show that the segmentation accuracy rate achieves 96%.

Keywords Point Cloud; Segmentation; Seed Point; Region Growing

1 Introduction

With the increasing use of computer graphics technology in conjunction with agricultural knowledge, research on the morphological structure and physiological function of plants is entering a digital and visual stage. Digital content creation using real-world data has gained a great deal of attention over the past decades [1]. Segmentation is necessary in reverse engineering, where models are reconstructed from points acquired on the surface of the object by laser scanners. Additionally, agricultural and forestry plants have different structures with respect to their leaves, branches, and fruits. Thus, different parts of the models need to adopt different modeling methods to guarantee the precision and effectivity of the reconstruction. Segmenting the point cloud of different structures effectively is one of the key technologies for high-precision reconstruction of reverse engineering, and is also useful in other applications, such as three-dimensional (3D) city modeling, feature recognition, geometry compression, and industrial site modeling [9]. We propose an algorithm for segmenting an unorganized set of points of a 3D object and dividing the points into several proper subsets with similar attributes, which mainly include distance, density, normal, and curvature.

Y. Fan · M. Wang · J. Hu
College of Information Engineering, Northwest A&F University, China
E-mail: yuling_fan@nwsuaf.edu.cn

D. He ✉
College of Mechanical and Electronic Engineering, Northwest A&F University, China
E-mail: dongjian_h168@sina.com

J. Chang · J. Zhang
National Centre for Computer Animation, Media School, Bournemouth University, Bournemouth, UK

1.1 Related work

In current reverse engineering, a point cloud is typically divided into regions with similar topological structures to facilitate surface reconstruction. Existing segmentation methods mainly include edge-based segmentation, surface-based segmentation, and a combination of these two methods.

In edge-based segmentation, the boundary line connected by boundary points is the fundamental base for the segmentation. Wang et al. [24] proposed a segmentation algorithm for point cloud models of buildings. Their algorithm extracts buildings according to the boundary of the point cloud models. The boundary corresponds to discontinuities in depth or height, and therefore distinguishes one building from other buildings and objects on the ground. However, this method is very sensitive to noise. Dai et al. [6] developed a segmentation method for a point cloud distributed in the principal direction of tree models. The algorithm calculates the principal curvature and direction of the points from the tree point cloud models and uses this information to define an energy function. It then determines the segmentation of the leaves and branches according to the defined energy function and obtains the final segmentation results by separating the leaves and branches. The algorithm has a high operating efficiency, but its application is limited to tree models. Guillaume et al. [12] calculated the curvature tensor-based triangle mesh and used it to segment points into surface patches. Then, they adjusted the boundaries to obtain the segmentation of patch edges.

Surface-based segmentation uses local surface attributes as a similarity measure and merges the points with similar attributes [20]. It can also provide information with abstract expressions, which is useful for expression and analysis [16]. This method usually performs better than edge-based segmentation. Richtsfeld et al. [21] proposed a segmentation method using radial reflection to estimate model shapes. The algorithm mainly extracts the model shapes of surfaces. Rabbani et al. [20] proposed a constraint-based segmentation algorithm that can extract smooth regions in point cloud models. However, its segmentation results depend heavily on the parameter settings. Yamauchi et al. [25] proposed mesh segmentation with a mean shift algorithm. It is based on normal clustering using an adaptive mean shift algorithm and performs segmentation using the region-growing algorithm. The authors algorithm was originally proposed for the segmentation of images [5].

The mixed method involves combining methods based on edges and surfaces. Zhang et al. [28] used a statistical method to extract feature lines; however, it is affected by noise and sampling quality. Lari [15] proposed a segmentation algorithm that can quickly extract linear features in the point cloud data instead of segmenting the entire point cloud data, which leads to some limitations regarding the segmentation result. Zhana et al. [27] proposed a segmentation algorithm based on a color model to segment buildings, but this algorithm cannot be applied to point cloud models that do not contain color information, thereby greatly limiting its application potential. Dorninger et al. [8] proposed a point cloud segmentation algorithm based on parametric space. The algorithm clusters point cloud data in the space defined by the parameterization of the point cloud. However, this method is difficult to apply to unorganized point cloud data models. Gomes et al. [11] used 3D moving fovea to process parts of a scene using different levels of resolution. This approach can be used to identify objects in a point cloud. Gelfand et al. [10] presented shape segmentation using local slippage analysis. The shapes are defined as symmetrical, which includes cylinders, planes, spheres, and surfaces of revolution. The method merges initial surfaces and is sensitive to the size of patches. In [17], Marshall et al. proposed an improved least-squares fitting algorithm, which can segment the primitives (cylinders, spheres, and cones) from range data. In [16], Li et al. presented an improved algorithm to fit primitives using global relations, which can be obtained through constrained minimization. Pu et al. [19] performed building segmentation and extracted features using surface growing according to direction and size derived from convex hulls. Ochmann et al. [18] filtered out clutter outside the building, which was caused by mirrors and windows. This method obtains point labeling of a buildings room and is also homogeneous within each room. Kaick et al. [14] proposed a shape segmentation algorithm, which can optimize decomposition based on characterization according to the expected geometry of a shape. Demir et al. [7] used similarities to segment and detect the shape of a point cloud.

1.2 Overview and contributions

The aforementioned point cloud segmentation methods are mostly used for point cloud models or 2.5-dimensional depth images in specific application scenarios. Many of the methods involve a large number of parameters that have a substantial influence on the final segmentation result. These segmentation algorithms encounter many limitations when applied to unorganized 3D point cloud data. 3D scanners capture unorganized point cloud data, which makes it difficult to determine the topological relationship between points. The feature information corresponding to different parts is vastly different and it is difficult for these algorithms to obtain an ideal segmentation result. To address the aforementioned issues, we propose a self-adaptive point cloud segmentation algorithm to effectively segment different unordered point cloud data and lay a foundation for the 3D high-precision reconstruction of plants. The complete segmentation algorithm is provided in Algorithm 1. It is mainly divided into two steps: select seed points and segment the points. The first step mainly consists of the calculation of representativeness and diversity values. The second step mainly consists of the calculation of constraints. The calculations are described in detail in Sections 3 and 4, respectively.

To summarize, our contributions are as follows:

- Our point cloud segmentation algorithm automatically selects seed points without user intervention, thereby suitably expressing similarity and guaranteeing the consistency of the segmentation results.
- The algorithm can confirm the number of seed points automatically instead of requiring user interaction. Our algorithm enhances its adaptability by using an automatic calculation.
- The algorithm can also rearrange the location of a noised point, thereby reducing the effect of noise.
- We consider the connectivity of points and use a semiautomatic region-growing algorithm by reducing the number of parameters, thereby balancing between the degree of under- and over-segmentation.

Algorithm 1 Self-adaptive segmentation method for a point cloud

```
Inputs: Unorganized set of points $U = \{p_i\}_{i\in I} \subset R^3$; neighbor-finding function $\Omega()$.
 1: Calculate the initial radius and the points whose total distance is minimum(See Eq.(1)).
 2: while($\Omega(x).size$) is changed do
 3:   Calculate neighborhood size $\alpha$(See Eq.(2)).
 4: end while
 5: for $i$=0 to $U.size$ do
 6:   Calculate density $\rho_i^{Rep}$(See Eq.(3)).
 7: end for
 8: Sort the points in descending order
 9: for $j$=0 to $U.size$ do
10:   Calculate distance $\delta_j^{Div}$(See Eq.(4)).
11:   Calculate attribute value $sp_j^{Att}$
12: end for
13: Get seed list $S_L$
14: Get the normal and point connectivity(See Sect.3)
15: Segment the point cloud on the basis of list $S_L$(See Algorithm 2 and Sect.3)
```

2 Seed point calculation

In reverse engineering, mass research is a bottom-up method, which starts from the seed points and uses the region-growing algorithm. One problem for this method is the difficulty of selecting seed points. Thus, we present an algorithm that can automatically select seed points, which have a high dense density compared to its surrounding neighbors with lower density and a large diversity with other seed points.

2.1 Feature calculation

Representativeness values calculation Representativeness can be measured by a point that has a higher density than its neighbors. There has been extensive research on density calculation, but most works require artificial parameters to set the radius. To reduce user intervention and enhance adaptability, the implementation of an automatic process is necessary.

The input is an unorganized set of points $U = \{p_i\}_{i\in I} \subset R^3$. A bounding box that has the minimal area and encloses all the points can be constructed. Then we can obtain the initial radius as follows:

$$\alpha_0 = \frac{dist_d}{\sqrt[3]{N}} \tag{1}$$

where $dist_d$ is the diagonal length of the input U's bounding box and N is the number of points in U. Additionally, it is necessary to compute a point $mp \in U$ (see Fig. 1), which has the total minimum distance to a set of points. It can be obtained as follows:

$$mp = \arg\min \sum_{i\in I} ||mp - p_i|| \tag{2}$$

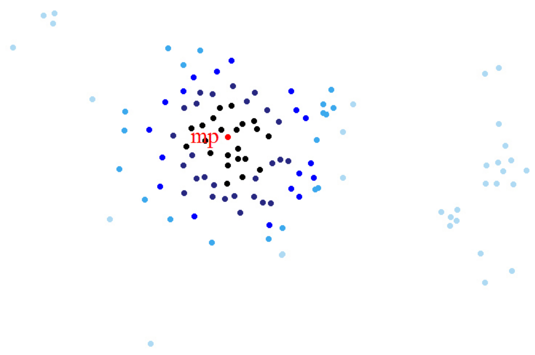

Fig. 1: Point mp has the total minimum distance

According to the initial neighborhood size and point mp, the neighborhood size can be computed automatically. It is adapted during iteration processing and can be described as follows:

$$\alpha_i = \alpha_{i-1} + \Delta L$$
$$\Delta L = \frac{1}{K}\frac{\sum_{p_i\in Nhbd_{mp}} \theta||mp - p_i||\varphi(\mathbf{n}, \mathbf{n}_{p_i})\mathbf{n}_{p_i}}{\sum_{p_i\in Nhbd_{mp}} \theta||mp - p_i||\varphi(\mathbf{n}, \mathbf{n}_{p_i})} \tag{3}$$

where $||\cdot||$ is the L_2-norm, $\mathbf{n}$ is the normal vector of the point, $Nhbd_{mp} = \{p_i|p_i \in U \wedge ||mp - p_i|| < \alpha_i\}$ under a neighborhood size α_i and K is the number of points in $Nhbd$; that is, the points can be obtained from fixed distance neighbors (FDNs). For a given point and radius α_i, the FDNs select all the points within the area (Fig. 2). For an FDN search, the number of neighbors K changes according to the density. The weight and spatial functions are defined by

$$\theta\left(r\right) = e^{-r^2/(\alpha/2)^2}, \varphi(\mathbf{n}, \mathbf{n}_i) = e^{-\left(\frac{1-\mathbf{n}^T\mathbf{n}_i)}{1-\cos(\mathbf{n}^T\mathbf{n}_i)}\right)}.$$

Eq. (3) is calculated by iterating until the value of K is constant. The computation of the radius is suitable for different models and enhances adaptability.

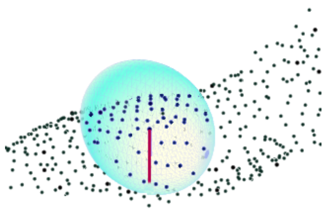

Fig. 2: Neighbors of a point

After neighborhood size α has been obtained, the representativeness value, that is, the density of each point in U, is defined as

$$\rho_i^{Rep} = 1 + \sum_{j \in I\setminus\{i\}} \theta(||p_i - p_j||) \tag{4}$$

To avoid excessive segmentation, we sort all the data points in descending order according to density.

Diversity values calculation To ensure the diversity of the seed points, the diversity can be measured by computing the minimum distance between point p_i and other points with densities that are higher than that of point p_i [22]. We define the diverse distance as follows:

$$\delta_i{}^{Div} = \min\nolimits_{j<i} ||p_i - p_j|| \tag{5}$$

For the point with the highest density, it can be noted that

$$\delta_i{}^{Div} = \max_{i \neq j} ||p_i - p_j|| \tag{6}$$

The diversity values are similar to those obtained using the maximum marginal relevance algorithm [2], which is used to remove points that are similar to those already selected. The maximum marginal relevance algorithm compares a point with selected points, whereas we compare a point with all other points, thus our algorithm has higher global diversity.

2.2 Automatic selection of seed points

As mentioned previously, the seed point is characterized by a higher density than its neighbors and by a relatively large distance from points with other higher densities to ensure the stability of this process. Therefore, we determine it using the following formula:

$$sp_i^{Att} = \log \rho_i^{Rep} + \log \delta_i^{Div} \tag{7}$$

We use the aggregation and flame datasets to evaluate the performance of selecting seed points in two dimensions, as shown in Fig. 3. Clearly, the algorithm tends to find seed points that are both dense and a large distance from other seed points. Next, we sort the points in descending order according to the value of sp. Then we generate the set of seed points S_L.

For the region-growing process, we select one point in S_L as a seed point in the sequence. Until the seed point operation is complete, the subsequent points are selected as seed points individually. Our algorithm can thus adaptively select seed points and automatically set the number of seed points. It can address the problem of inconsistent segmentation results caused by the random selection of seed points.

3 Segmentation

After seed point selection, a semi-automatic region-growing algorithm is used for segmentation. The region-growing algorithm is a surface-based method and involves clustering points with similar attributes into the same region with respect to the seed points.

In the region-growing process, another difficulty is determining whether the point should be added in a given region because the decision is susceptible to noise. To address this difficulty, we consider the connectivity of points and use an improved normal as a constraint.

3.1 Estimation of normal

Traditionally, the normal is equivalent to the normal of the least-squares plane of the point and its neighborhood. Using principal component analysis (PCA) [8] to estimate the normal produces poor approximations because of the existence of surface discontinuities and noise. Surface discontinuities are mainly caused by equally weighting the incorrect contributions of points [3]. Thus, we use an improved constrained nonlinear least-squares algorithm to adaptively determine the weight of each point contribution, which can be expressed as

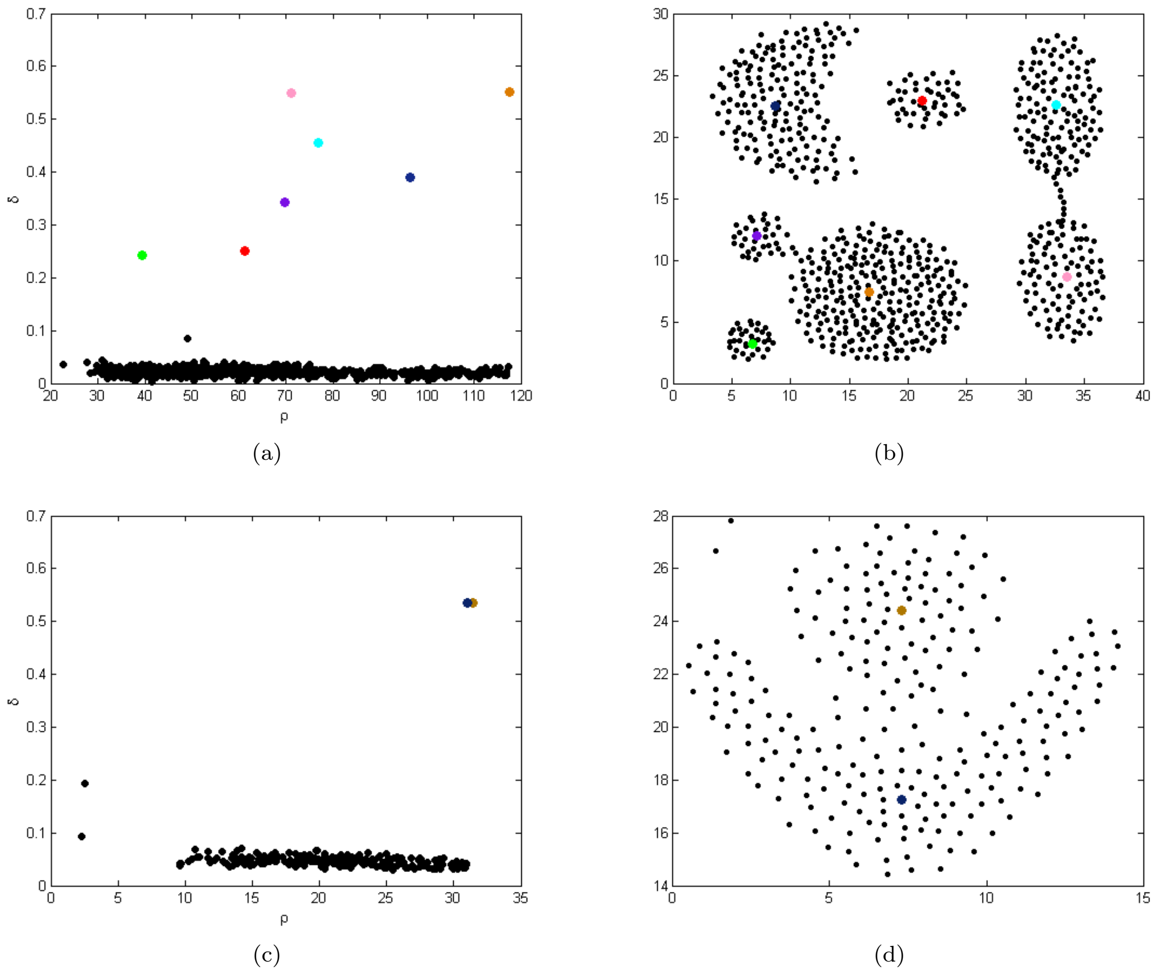

Fig. 3: Identify seed points and illustrate their location in datasets: **a** the decision of seed points in the aggregation dataset, colored points represent the seed points; **b** identify the location of seed points in the aggregation dataset; **c** present the seed points in the flame dataset; **d** the location of seed points in the flame dataset

follows:

$$\begin{aligned} &\arg\min_{\mathbf{n}} \frac{1}{2}\sum_{k=1}^{K} e^{-\lambda(o_k^T\mathbf{n})^2}(o_k^T\mathbf{n})^2 \\ &\qquad s.t.||\mathbf{n}||_2 = 1 \\ &\qquad \lambda = \frac{|K|}{|N|} \end{aligned} \tag{8}$$

where $o_k = p_k - p$, p_k is the neighbors of point p, $\mathbf{n}$ represents the normal vector and weighting $e^{-\lambda(o_k^T\mathbf{n})}$ can adaptively deflate the contribution of the high orthogonality mismatch defined by λ.

If the input point cloud has severe noise, we rearrange the location of the noised point. We use the following to obtain the location of adjustment:

$$\arg\min_{\tilde{p},\mathbf{n}} \frac{1}{2}\sum_{i=1}^{k} e^{-\lambda((p_k-\tilde{p})^T\mathbf{n})^2}((p_k-\tilde{p})^T\mathbf{n})^2 \tag{9}$$

where $\tilde{p}$ is the adjusted location. Figs. 4(a) and (b) show the adjustment of noise by which we can reduce the influence of noise. We apply PCA and Eq. (9) to the mimosa model, which is provided in [13]. The results are shown in Fig. 4. Comparing Figs. 4(c) and (d), we observe the improvement of the algorithm in the estimation of the normal.

3.2 Point Connectivity

To measure connectivity, the adjacency matrix is constructed, which is obtained from the surface curvature estimation that describes the connectivity for the unorganized set of points.

Based on FDN information and the curvature, an adjacency matrix SA is built, and the matrix is symmetric. If p_i, p_j are connected, $SA_{i,j} = 1$, otherwise

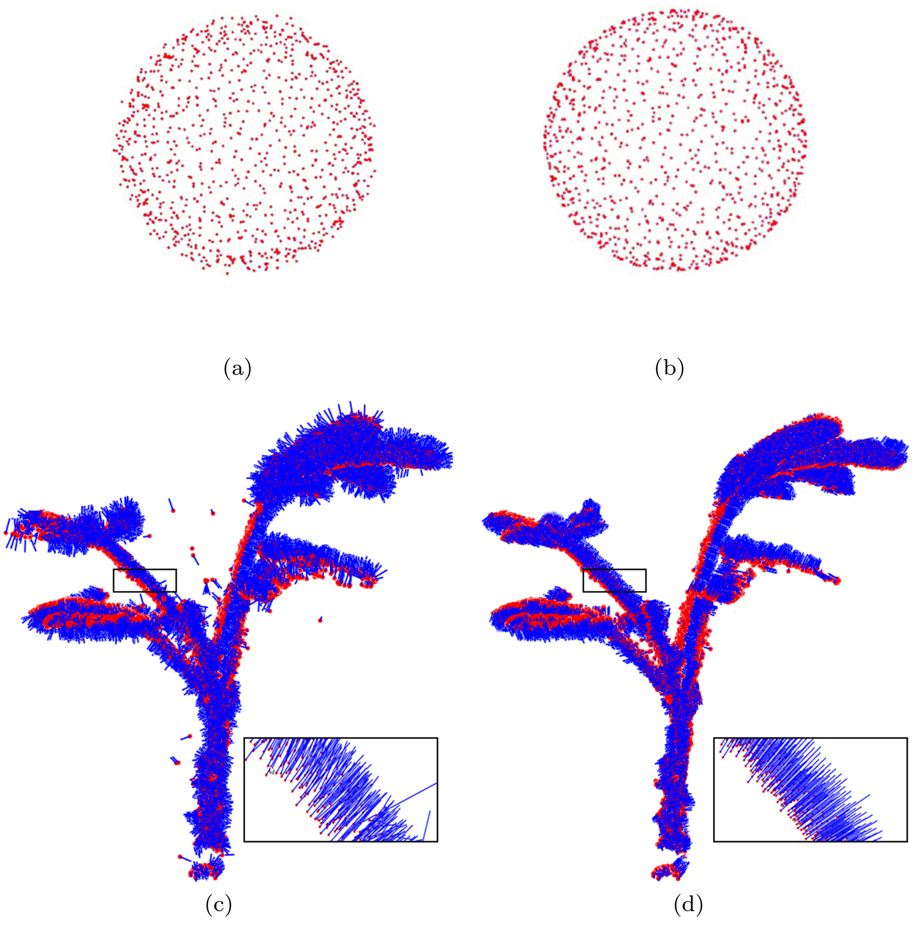

Fig. 4: Results of normal estimation: **a** nonlinear least-squares algorithm to adjust noise, **b** adjusted result of noise, **c** PCA normal estimation, **d** nonlinear least squares normal estimation

$SA_{i,j} = 0$. Considering connectivity, we assume that if points p_i, p_j are FDNs of each other and the curvature is less than the mean curvature in FDNs, then $SA_{i,j} = 1$.

We now consider computing the curvature using the method of moment-based surface analysis, which is robust to noise [4]. For a surface M and neighborhood ball B with radius α, the zero moment of point p can be defined as

$$M_\alpha^0(p) := \int_{B_\alpha(p)\cap M} p dp \tag{10}$$

and the first moments as

$$\begin{aligned} M_\alpha^1(p) &= \int_{B_\alpha(p)\cap M} (p - M_\alpha^1(p)) \otimes (p - M_\alpha^0(p)) dp \\ &= \int_{B_\alpha(p)\cap M} p \otimes p dp - M_\alpha^0(p) \otimes M_\alpha^0(p) \end{aligned} \tag{11}$$

where $q \otimes w = (q_i w_j)_{i,j=1,2,3}$. Because of the definition of a moment-based surface via local integration, these moments are robust to noise. Additionally, the curvature at point p can be computed using the zero and first moments shift:

$$\varsigma = G(\frac{||M_\alpha^0(p) - p||\lambda_{\min}}{\alpha\lambda_{\max}}) \tag{12}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the eigenvalues at point p in the first moment, $G = \frac{1}{\alpha + K^2}$ with neighborhood size α obtained from Eq. (3), and K is the number of neighbors.

3.3 Region growing

The basic purpose of the segmentation algorithm is to subdivide the points into meaningful subsets with high similarity, while avoiding over- and under-segmentation.

The details of the segmentation steps are provided in Algorithm 2 and the implementation process is further described below.

We denote the current region by the set Rc, the current seed region by Sc, and the segment list by $SegList$. A seed is chosen from the set of seed points S_L.

Algorithm 2 Point cloud segmentation

```
Inputs: Unorganized set of points $U = \{p_i\}_{i \in I} \subset R^3$ ;
    Seed List $S_L = \{seed_1, seed_2, ..., seed_n\}$; neighbor-finding function $\Omega()$; Symmetric adjacency matrix $SA$.
Initialize: Region List $R_L = \emptyset$, Available points list $A \leftarrow U$
 1: while $\{A\}$ is not empty do
       Current region $\{Rc\} \leftarrow \emptyset$, Current seeds $\{Sc\} \leftarrow$ the seed point from $S_L$
 2:   for $i$=0 to $Sc.size()$ do
 3:     Find the neighbors of the current seed point
        $\{Nbhd\} \leftarrow \Omega(Sc[i])$
 4:     for $j$=0 to $Nbhd.size()$ do
 5:       Current neighbor $p_l \leftarrow Nbhd_j$
 6:       if $p_l \in$A and $cos(|\langle \beta_{p_l}, \beta_{seed} \rangle|) < \sigma$ then
 7:         Remove $p_l$ from $\{A\}$
            $p_l \leftarrow \{Rc\}$
 8:         if $SA(seed, p_l) = 1$ then
 9:           $p_l \rightarrow Sc$
10:         end if
11:       end if
12:     end for
13:   end for
14:   Add current region to the segment list $Rc \rightarrow R_L$
15: end while
```

The implementation process is as follows:

1. Select the first data point $seed_1$ in S_L as the initial current seed point and insert it into the current seed region Sc.
2. Obtain the neighbors of the current seed point that satisfy $\cos(|\langle \beta_{p_l}, \beta_{\text{seed}} \rangle|) < \sigma$. The neighbor will be added to the current region Rc. σ represents the threshold of the normal. As $\sigma \to 1$, we have fewer segments, and in the extreme case, all the points belong to one segment. Similarly, as $\sigma \to 0$, we have more segments, and in the extreme case, each point belongs to one segment. Thus, σ provides a balance between over- and under-segmentation.
3. If the points are connected, add the neighbor to the current seed region Sc and remove it from S_L.
4. Delete the current seed point in seed region Sc and remove the data point from U.
5. Select the next data point in the current seed region Sc as the current seed point, return to Step (2), and execute until the seed region Sc is null.
6. Save Rc in the segment list R_L. Select the next data point in S_L as the current seed point, return to Step (1), and end the segmentation when all the points are segmented.

4 Result and Analysis

To evaluate the segmentation result of our proposed algorithm, we compare it with the segmentation algorithms of Rabbani et al. [20] and Rodriguez et al. [22].

(1) Segmentation of mimosa

Figure 5 shows the segmentation result for the mimosa point cloud using the three algorithms (the seed points are circled in the figure).

Comparing parts 1 to 3 in Fig. 5(a) and part 1 in Fig. 5(c), it is clear that the Rabbani et al. algorithm [20] excessively segments single leaves, and from part 4, we observe that the algorithm failed to separate the stems from the leaves of the mimosa and lacked sufficient segmentation strength. Comparing parts 1 and 2 in Fig. 5(b) and part 1 in Fig. 5(c), we observe that the Rodriguez et al. algorithm [22] also excessively segmented single leaves and insufficiently segmented the connection parts of the stems and leaves. In Figs. 5(c) and 5(d), we can observe that each part of the model has a seed point, which has a higher density and is far from the other seed points. Moreover, the algorithm effectively distinguished between stems and leaves. It obtained accurate resulting segments and avoided the problems of over- and under-segmentation.

(2) Segmentation of a plant

Comparing part 1 in Figs. 6(a) and (c), we observe that the Rabbani et al. algorithm [20] excessively segmented single leaves and from part 2, we observe that the algorithm failed to separate the stems from the leaves of the plants. Comparing part 1 in Figs. 6(b) and (c), we note that the Rodriguez et al. algorithm [22] also excessively segmented single leaves. As shown in Fig. 6(d), our algorithm maintained point connectivity and effectively segmented the stems and leaves.

(3) Segmentation test of maize

Fig. 6 shows the result of segmenting a 3D point cloud of maize using the three algorithms.

Comparing part 1 in Figs. 7(a) and (c), it is clear that the Rabbani et al. algorithm [20] resulted in over-segmentation. Comparing parts 1 and 2 in Fig. 7(c), we observe that the Rabbani et al. algorithm [20] resulted in under-segmentation. Comparing part 1 in Figs. 7(b) and (c), it is clear that the Rodriguez et al. algorithm [22] also resulted in under-segmentation. Finally, comparing part 2 and part 1 in Fig. 7(b), the Rodriguez et al. algorithm [22] resulted in over-segmentation. Fig. 7(c) shows the segmentation result of our algorithm. Fig. 7(d) shows that the seed points have a high density

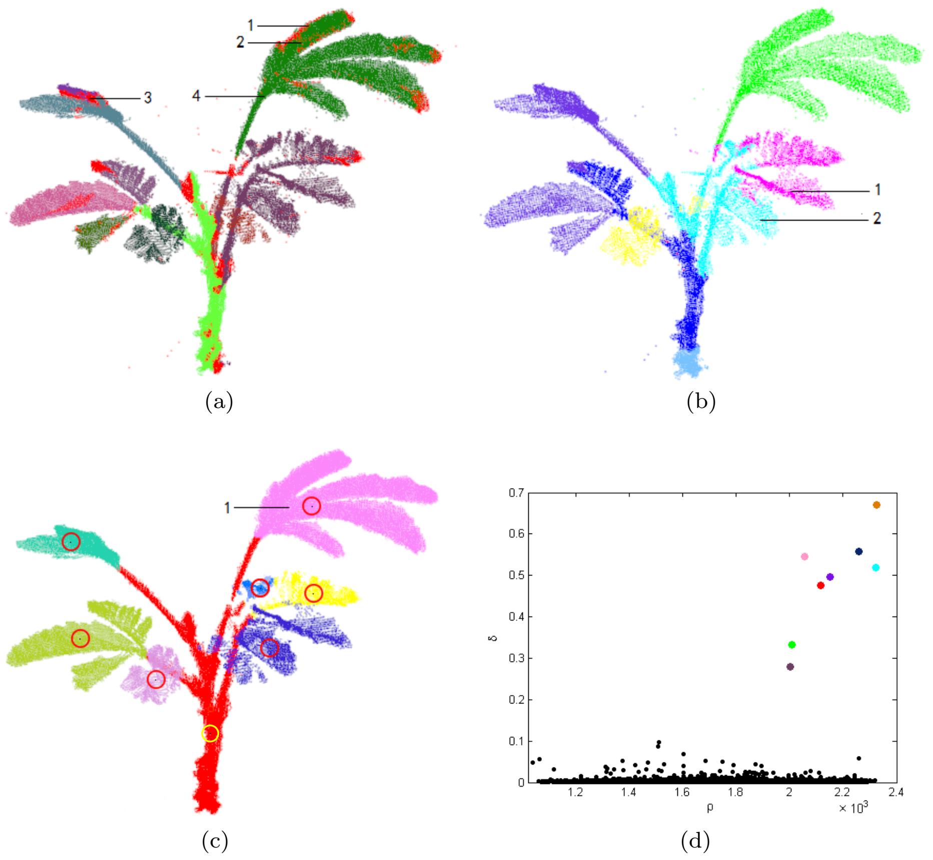

Fig. 5: Results obtained by the different methods of segmenting a single object: **a** segmentation of Rabbani et al. [20]; **b** segmentation of Rodriguez et al. [22]; **c** our method: eight seed points are selected, the representativeness and diversity values are shown in **d**; **d** the decision of seed points in two dimension

and an obvious division from the other seed points. The result demonstrates that the seed points are reasonable and effective. Our method can accurately segment a single object from multiple objects.

(4) Segmentation of other models

We conducted further tests on further models. Fig. 8 presents some segmentation results of models that we obtained from [14] and [26]. We note that the algorithm is robust when applied to these models and preserves the connectivity of the point cloud. Although the focus of our algorithm is the segmentation of a single point cloud model into meaningful subsets, we also show in Fig. 9 a test that applied our algorithm to scenes with multiple models. In the test, we segmented an indoor scene. We can observe that if the model has obvious differences in parts and structures, our method can segment it effectively.

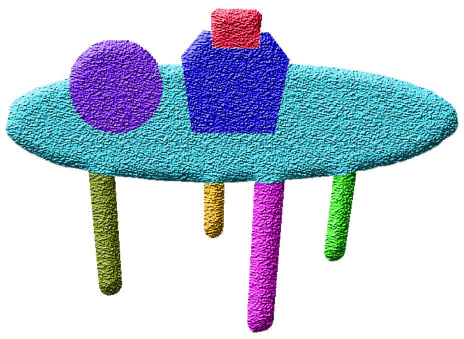

Fig. 9: Segmented indoor scene with multiple models

(5) Robustness test to noise

We added different levels of Gaussian noise to the models to evaluate the robustness of our approach. Noise of 0.3% and 0.6% were added to the models. The resulting segments are shown in Fig. 10.

To evaluate the accuracy of clustering, we used the common metric of purity [23], which calculates the max-

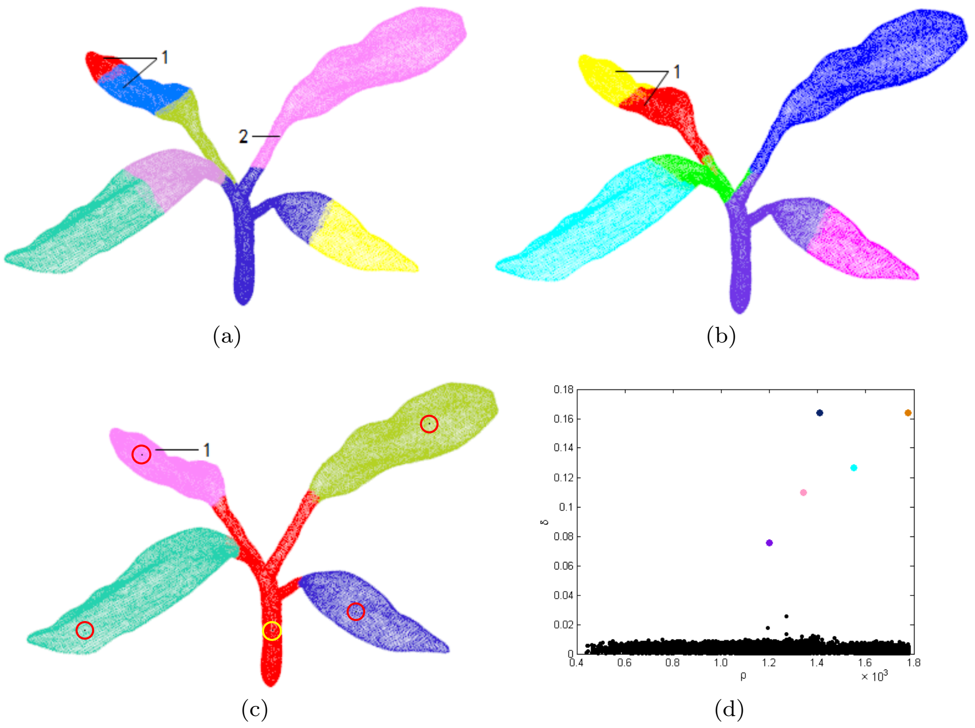

Fig. 6: Comparison of the plant model segmentation results using the three algorithms: **a** segmentation of Rabbani et al. [20]; **b** segmentation of Rodriguez et al. [22]; **c** our method: five seed points are selected; **d** the decision of seed points in two dimension.

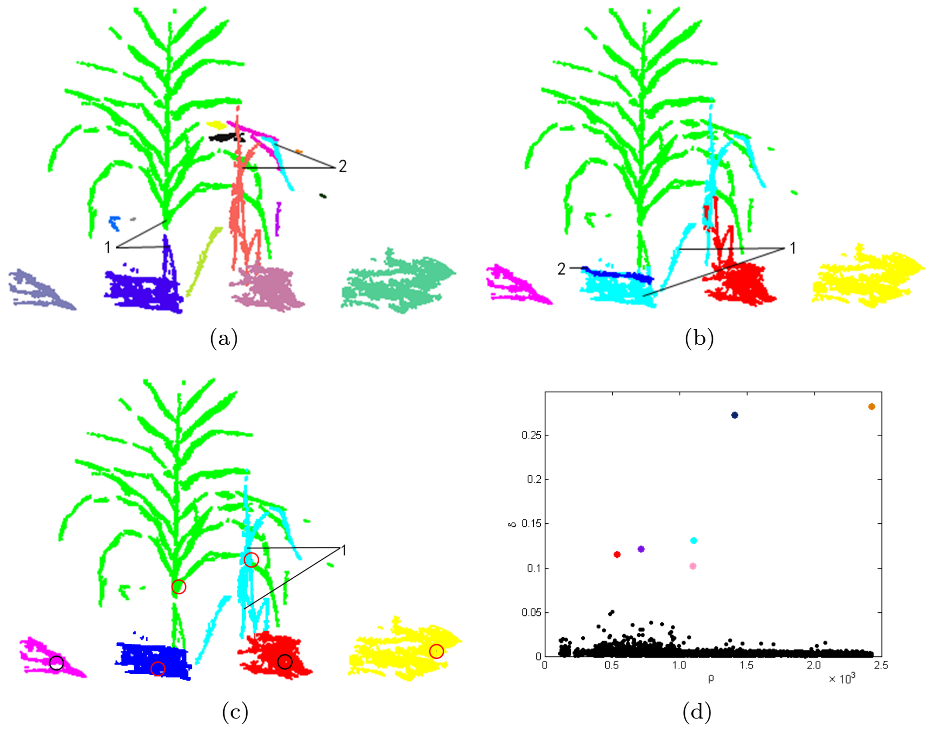

Fig. 7: Results obtained by the different methods when segmenting a single object from multiple objects: **a**, **b** and **c** the original model. **b**; **c**; and **d** Rabbani et al. [20], Rodriguez et al. [22], and our method, respectively: six seed points are selected; **d** the decision of seed points in two dimension.

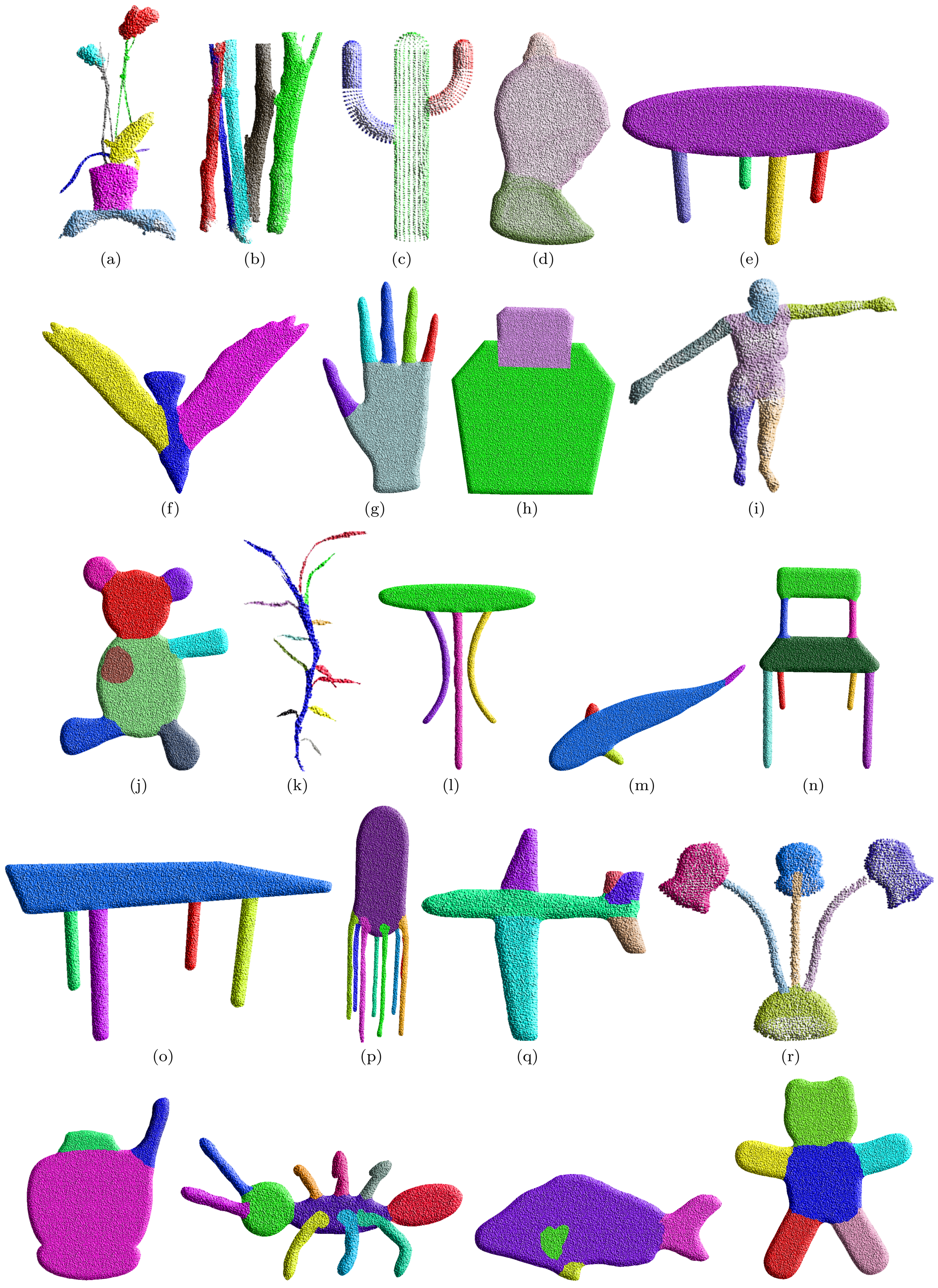

Fig. 8: Extended evaluation of our method on different objects

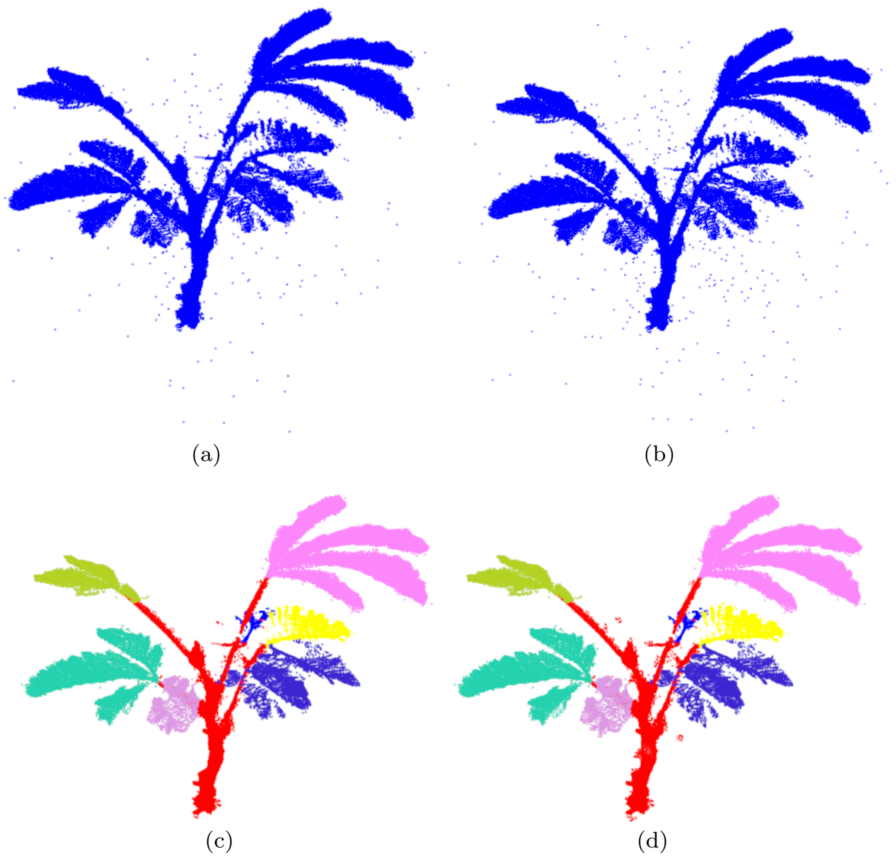

Fig. 10: Segmented the models with different levels of Gaussian Noise. **a** 0.3% noise; **b** 0.6% noise; **c**, and **d** are the segmentation results with noise

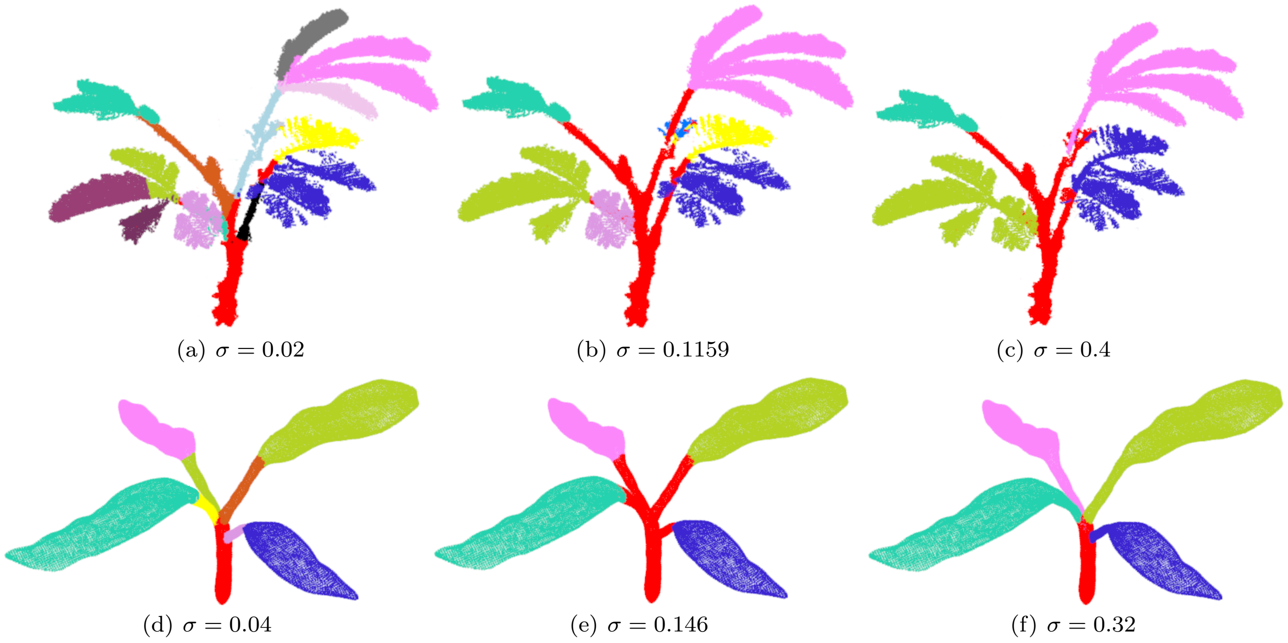

Fig. 11: Segmented the point cloud using different strengths σ

Table 1: The purity of point cloud model

S/N	Model	Data size	Region growing [20]	Rodriguez et.al [22]	Our method
			parameter/purity	parameter/purity	parameter/purity
1	Mimosa	40551	K=50, ς=0.152, σ=0.1159/0.54	d_c=0.015, Num=8/0.78	σ=0.1159/0.97
2	Plant	23096	K=30, ς=0.166, σ=0.146/0.83	d_c=0.1456, Num=7/0.86	σ=0.146/0.98
3	Maize	23320	K=30, ς=0.45, σ=0.40/0.77	d_c=0.16, Num=6/0.89	σ=0.40/0.97

imum number of points in cluster i corresponding to class j:

$$purity(R, C) = \frac{1}{N} \sum_i \max_j |region_i \cap class_j| \qquad (13)$$

where $R = \{region_1, region_2, \cdots, region_i\}$ represents the set of segmenting in R_L; N stands for the total number of points in U, and $C = \{class_1, class_2, \cdots, class_i\}$ represents the result of segmenting.

In the Rabbani et al. algorithm [20], three parameters need to be set: the number of neighbors K, curvature threshold ς, and normal threshold σ. In the Rodriguez et al. algorithm [22], two parameters need to be set: cutoff distance d_c and cluster number Num. In our method, only one parameter needs to be set: normal threshold σ. We segmented the point cloud using different strengths. Fig. 11 illustrates the effect of σ. It is clear that σ provides a balance between over- and under-segmentation.

For the experiments, Table 1 provides the necessary data. Fig. 12 shows the purity of the three algorithms. We observe that our algorithm effectively achieved the segmentation of the point cloud, with an increase in the purity of the segmentation of 0.26 and 0.13 compared with the Rabbani et al. [20] and Rodriguez et al. [22] algorithms, respectively.

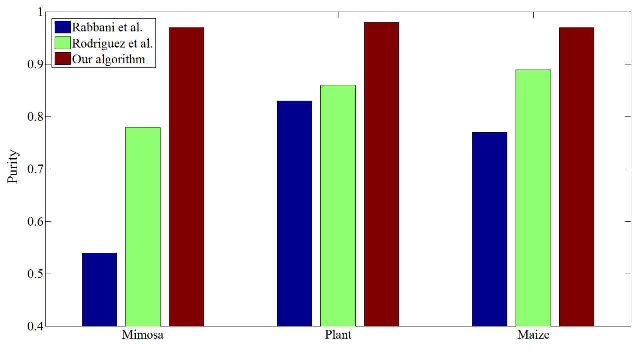

Fig. 12: Quantitative evaluation of the segmentation algorithm

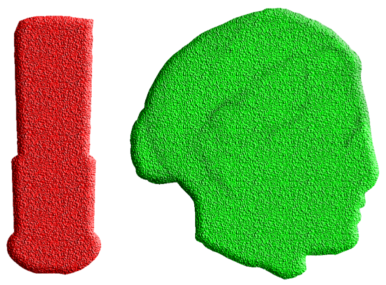

Fig. 13: Segmented point cloud with a limitation

5 Conclusion and Limitation

(1) Our self-adaptive point cloud segmentation algorithm automatically selects seed points and guarantees the consistency of the segmentation results. The algorithm reduces both the setting of parameters and the effect of noise, which can enhance the adaptability of the algorithm. The only user-specified parameter required by our method provides a balance between over- and under-segmentation. According to the test results, our algorithm effectively segmented the point cloud and achieved a segmentation rate of up to 96%.

(2) However, the algorithm demonstrated better segmentation results for models with obvious differences in parts and structures, and exhibited a certain limitation for models with similar structures; the limitation can be seen in Fig. 13.

Acknowledgements This work was partially funded by the National High-tech research and Development Program(863 Program: 2013AA10230402), National Natural Science Foundation of China(61402374), and the China Postdoctoral Science Foundation(2014M562457). The authors acknowledge the authors of [15], Shenzhen Key Lab of Visual Computing and Visual Analytics for the source data and the models.

References

1. Aiteanu, F., Klein, R.: Hybrid tree reconstruction from inhomogeneous point clouds. The Visual Computer **30**(6-8), 763–771 (2014)
2. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing

summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335–336. ACM (1998)
3. Castillo, E., Liang, J., Zhao, H.: Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. In: Innovations for Shape Analysis, pp. 283–299. Springer (2013)
4. Clarenz, U., Griebel, M., et al: Feature sensitive multiscale editing on surfaces. The Visual Computer **20**(5), 329–343 (2004)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(5), 603–619 (2002)
6. Dai, M., Zhang, X., et al: Segmentation of point cloud scanned from trees. In: Workshop on Community Based 3D Content and Its Applications in Mobile Internet Environments, ACCV (2009)
7. Demir, I., Aliaga, D.G., Benes, B.: Coupled segmentation and similarity detection for architectural models. ACM Transactions on Graphics (TOG) **34**(4), 104 (2015)
8. Dorninger, P., Nothegger, C.: 3d segmentation of unstructured point clouds for building modelling. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **35**(3/W49A), 191–196 (2007)
9. Fayolle, P., Pasko, A.: Segmentation of discrete point clouds using an extensible set of templates. The Visual Computer **29**(5), 449–465 (2013)
10. Gelfand, N., Guibas, L.: Shape segmentation using local slippage analysis. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 214–223. ACM (2004)
11. Gomes, R.B., da Silva, et al: Efficient 3d object recognition using foveated point clouds. Computers & Graphics **37**(5), 496–508 (2013)
12. Guillaume, L., Florent, D., Atilla, B.: Curvature tensor based triangle mesh segmentation with boundary rectification. In: Computer Graphics International, 2004. Proceedings, pp. 10–25. IEEE (2004)
13. Huang, H., Wu, S., et al: L1-medial skeleton of point cloud. ACM Trans. Graph. **32**(4), 65–1 (2013)
14. Kaick, O.V., Fish, N., Kleiman, Y., Asafi, S., Cohen-Or, D.: Shape segmentation by approximate convexity analysis. ACM Transactions on Graphics (TOG) **34**(1), 4 (2014)
15. Lari, Z., Habib, A.: A novel hybrid approach for the extraction of linear/cylindrical features from laser scanning data. ISPRS Ann Photogramm Remote Sens Spat. Inf Sci **2**, 151–156 (2013)
16. Li, Y., Wu, X., et al: Globfit: Consistently fitting primitives by discovering global relations. In: ACM Transactions on Graphics (TOG), vol. 30, p. 52. ACM (2011)
17. Marshall, D., Lukacs, G., Martin, R.: Robust segmentation of primitives from range data in the presence of geometric degeneracy. Pattern Analysis and Machine Intelligence, IEEE Transactions on **23**(3), 304–314 (2001)
18. Ochmann, S., Vock, R., Wessel, R., Klein, R.: Automatic reconstruction of parametric building models from indoor point clouds. Computers & Graphics **54**, 94–103 (2016)
19. Pu, S., Vosselman, G., et al: Automatic extraction of building features from terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36**(5), 25–27 (2006)
20. Rabbani, T., Van Den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36**(5), 248–253 (2006)
21. Richtsfeld, M., Vincze, M.: Point cloud segmentation based on radial reflection. In: Computer Analysis of Images and Patterns, pp. 955–962. Springer (2009)
22. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
23. Song, S., Li, C., Zhang, X.: Turn waste into wealth: On simultaneous clustering and cleaning over dirty data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1115–1124. ACM (2015)
24. Wang, J., Shan, J.: Segmentation of lidar point clouds for building extraction. In: American Society for Photogramm. Remote Sens. Annual Conference, Baltimore, MD, pp. 9–13 (2009)
25. Yamauchi, H., Lee, S., et al: Feature sensitive mesh segmentation with mean shift. In: Shape Modeling and Applications, 2005 International Conference, pp. 236–243. IEEE (2005)
26. Yücer, K., Sorkine-Hornung, et al: Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. ACM Transactions on Graphics (TOG) **35**(3), 22 (2016)
27. Zhana, Q., Liangb, Y., Xiaoa, Y.: Color-based segmentation of point clouds. Int Arch Photogrammetry, Remote Sens Spat Inf Sci **38**, 248–252 (2009)
28. Zhang, Y., Geng, G., et al: A statistical approach for extraction of feature lines from point clouds. Computers & Graphics **56**, 31–45 (2016)

Yuling Fan received her B.E. degree in College of Information Engineering from Northwest A&F University in 2015. She is currently pursuing M.E. degree in College of Information Engineering, Northwest A&F University. Her research interests include computer graphics, point cloud processing, and virtual reality.

Meili Wang is an associate professor at the College of Information Engineering, Northwest A&F University. She received her PhD degree in computer graphics in 2011 at the National Centre for Computer Animation, Bournemouth University. Her research interests include computer graphics, geometric modelling, image processing, visualization and virtual reality.

Dongjian He is currently a Professor in College of Mechanical and Electronic Engineering at Northwest A&F University. His research interests include computer graphics, image analysis and machine vision. He received a B.E., M.E. and D.E. in agricultural engineering from Northwest A&F University in 1982, 1985 and 1998, respectively. He was a lecturer in the College of Mechanical and Electronic Engineering at Northwest A&F University from 1987 to 1992, and an Associate Professor from 1992 to 1999. He is a member of the China Computer Federation, vice chairman of Shaanxi Society of Image and Graphics, vice chairman of Electrical Information and Automation Committee of CSAE, and a member of a council of Chinese Society for Agricultural Machinery.