

Article

Leaf Segmentation on Dense Plant Point Clouds with Facet Region Growing

Dawei Li ^{1,2}, Yan Cao ¹, Xue-song Tang ^{1,2,*}, Siyuan Yan ¹ and Xin Cai ^{1,2}

¹ College of Information Science and Technology, Donghua University, Shanghai 201620, China; daweili@dhu.edu.cn (D.L.); 2171339@mail.dhu.edu.cn (Y.C.); 2171353@mail.dhu.edu.cn (S.Y.); xcai@dhu.edu.cn (X.C.)

² Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, China

* Correspondence: tangxs@dhu.edu.cn; Tel.: +86-21-6779-2315

Received: 15 August 2018; Accepted: 22 October 2018; Published: 25 October 2018



Abstract: Leaves account for the largest proportion of all organ areas for most kinds of plants, and are comprise the main part of the photosynthetically active material in a plant. Observation of individual leaves can help to recognize their growth status and measure complex phenotypic traits. Current image-based leaf segmentation methods have problems due to highly restricted species and vulnerability toward canopy occlusion. In this work, we propose an individual leaf segmentation approach for dense plant point clouds using facet over-segmentation and facet region growing. The approach can be divided into three steps: (1) point cloud pre-processing, (2) facet over-segmentation, and (3) facet region growing for individual leaf segmentation. The experimental results show that the proposed method is effective and efficient in segmenting individual leaves from 3D point clouds of greenhouse ornamentals such as *Epipremnum aureum*, *Monstera deliciosa*, and *Calathea makoyana*, and the average precision and recall are both above 90%. The results also reveal the wide applicability of the proposed methodology for point clouds scanned from different kinds of 3D imaging systems, such as stereo vision and Kinect v2. Moreover, our method is potentially applicable in a broad range of applications that aim at segmenting regular surfaces and objects from a point cloud.

Keywords: individual leaf segmentation; point cloud; greenhouse plant; facet over-segmentation; local K-means clustering; region growing

1. Introduction

Leaves account for the largest proportion of all organ areas for most kinds of plants and comprise the main part of the photosynthetically active material in a plant. Leaves contain important information about the surface morphology and structure of a plant; therefore, observation of leaves can reveal its growth status. Changes in leaf morphology, texture, or color normally reflects biotic stress (plant diseases and pests) or abiotic stress (drought), so automatic leaf segmentation, recognition, and classification methods can provide rapid early warnings for agriculture facilities and, ultimately, help to increase plant output. Studies have used traditional imaging techniques and analyzing tools to carry out 2D leaf recognition and classification. Neto et al. used Gustafson-Kessel clustering and a genetic algorithm to segment individual leaves from different environments, and tested the method successfully on soybean, sunflower, pigweed, and velvetleaf plants [1]. Xu et al. extracted various color and texture features of leaves to identify nitrogen- and potassium-deficient tomatoes, and showed that the system could diagnose disease about 6–10 days before experts could reach the same assessment [2]. Kalyoncu and Toygar segmented and classified leaves of different species using a

linear discriminant classifier [3], achieving an accuracy of 70% on the Leafsnap dataset [4]. Zhang et al. proposed a novel cucumber disease recognition approach combining three pipelined procedures: segmenting diseased leaf images by K-means clustering, extracting shape and color features from lesion information, and classifying diseased leaf images using sparse representation [5].

Plant phenotyping is intended to measure complex traits related to growth, yield and adaptation to stress with a certain accuracy and precision at different scales of organization, from organs to canopies [6,7]. As an important task in high-throughput phenotyping, individual leaf segmentation has been drawing considerable attention from biology research and agricultural industry facilities. Recently, plant phenotyping software tools have sprouted up, such as PlantCV [8] and Leaf-GP [9]. Automatic leaf segmentation, alignment, and tracking algorithms are the key components of those tools, and are mostly based on 2D imaging. Scharr et al. made a recent survey of state-of-the-art 2D leaf segmentation methods for plant phenotyping, and evaluated the results of four different methods [10]. Given a fluorescence plant video, Yin et al. proposed a method that performed multi-leaf joint segmentation, alignment, and tracking (SAT) for *Arabidopsis* [11]. Viaud et al. proposed an *Arabidopsis* leaf segmentation and tracking system, in which the plant images were first segmented by a watershed algorithm and then refined by an ellipsoid-shaped model [12]. Recently, some researchers have introduced deep learning in counting or segmenting individual leaves from canopy images. Dobrescu et al. used a modified Resnet50 net to count leaves of *Arabidopsis* and tobacco plant images in the Computer Vision Problems in Plant Phenotyping (CVPNP) 2017 Leaf Counting Challenge dataset [13]. Although their method outperforms the winner of the previous CVPNP challenge, it is not able to locate the exact positions of leaves. Morris employed a pyramid-like Convolutional Neural Network (CNN) to segment leaves from images of dense foliage [14], which restricts the segmentation to the leaf boundary searching level.

Current leaf phenotyping algorithms are mostly image-based methods that often leverage machine learning and pattern recognition techniques. Despite the evident progress that has been made, leaf phenotyping remains restricted to processing several kinds of standard crops and plants (e.g., wheat [8,9] and rosette plants [10,12,15]). Moreover, the structure of the canopy is usually complicated, resulting in overlapping among leaves, which poses a great challenge to 2D leaf segmentation algorithms. As the cost of hardware continues to fall and image technology advances, 3D imaging has become a promising solution for sensing and phenotyping in cost-sensitive agricultural applications. Compared to traditional 2D imaging and vision techniques, 3D imaging not only acquires pixel-level colors, but also acquires the most important depth information of the scene. The spatial structure information of the plant is highly beneficial for studying its growth and its phenotypic traits. Recently, research on individual leaf segmentation with depth and range images or 3D point clouds has emerged. Some related literature has focused on tasks such as separating canopy volume from foliage point clouds scanned from a Lidar [16], segmenting single trees from forest point clouds [17,18], and estimating the leaf area density (LAD) of the plant [19]. Teng et al. built a segmentation and classification system for leaves, by which individual leaves can be segmented from point clouds by a joint 2D/3D approach [20]. Chaurasia and Beardsley designed a superpixel graph clustering algorithm to carry out initial segmentation of single-lobed leaves from point clouds, and then used Iterative Closest Point (ICP) matching to refine the segmentation results [21]. Others have focused on low-cost 3D sensors and techniques to generate dense plant point clouds, from which plant organs are further segmented. Li et al. [22] established a low-cost stereo system to reconstruct point clouds from six kinds of greenhouse plants, showing high accuracy and invariance to illumination changes. Paproki et al. [23] used multi-view 3D reconstruction and 3D meshes to generate a model of *Gossypium hirsutum* from 64 images, and then segmented leaves, petioles, and internodes from the model via a pipeline of four steps. Xia et al. exploited the Kinect v1 sensor to carry out in situ leaf segmentation in a greenhouse and reached a total segmentation rate of 87.97% [24]. Several research groups even turned to 4D point cloud data, with time as the 4th dimension for facilitating automatic organ segmentation. Li et al.

conducted a spatial and temporal analysis on 4D point cloud data of plants; budding and bifurcation events could be accurately segmented in a non-real-time way [25].

In this paper, we propose an individual leaf segmentation approach for dense plant point clouds using facet over-segmentation and facet region growing. The method can be divided into three steps: (1) point cloud pre-processing; (2) facet over-segmentation; and, (3) facet region growing for individual leaf segmentation. In the first step, the raw point clouds acquired from 3D sensors are filtered and preprocessed to leave out non-leaf areas. Then iterative principle component analysis (IPCA) is used to compute the spatial characteristics of each point in the point cloud (e.g., normal of the fitted plane in a neighborhood, and smoothness). In the second step, we first deploy a number of seed points and then apply a clustering algorithm to the plant point cloud with those seed points being the initial cluster centers to generate small facets. The facets generated have a smooth spatial structure and are uniformly distributed on the point cloud. In the third step, we use facet adjacency and coplanarity as conditions to carry out facet region growing with a breadth-first search strategy. After region growing, multiple facets will be combined to form a larger spatial structure that is finally regarded as a segmented leaf provided that it covers enough points. The experimental results show that the proposed method is effective in segmenting individual leaves from greenhouse ornamentals such as *Epipremnum aureum*, *Monstera deliciosa*, and *Calathea makoyana*, with precision and recall above 90%. The results also reveal the wide applicability of the proposed methodology for point clouds scanned from different kinds of 3D imaging systems, such as stereo vision and Kinect v2.

2. Materials and Methods

2.1. Platform and Experiment Subjects

2.1.1. Platform

The processing unit was a desktop, assembled by the authors, with an Intel Core i7-7700 CPU (Intel, Santa Clara, CA, USA) and 16 GB RAM. The software environment included VS2013 (Microsoft, Redmond, WA, USA) with the Point Cloud Library (PCL) [26], which were all operated in Windows 10. In the experiment, two types of imaging sensors with tripods were adopted to scan sample plants for point clouds. The first system was a binocular stereo vision system proposed in [22] (as illustrated in Figure 1a). This stereo vision platform consisted of two high-definition webcams (HD-3000 series, Microsoft, Redmond, WA, USA), a supporting board (LP-01, Fotamate, Jiangmen City, China) with a scale line, and a tripod (VCT-668RM, Yunteng Photographic Equipment Factory, Zhongshang City, China). The second sensor was a structured light sensor [27] (Kinect V2, Microsoft, Redmond, WA, USA) that obtains depth information by capturing reflections from the projected infrared light pattern. The Kinect sensor was mounted on the same type of tripod as used for the stereo system (as illustrated in Figure 1b).

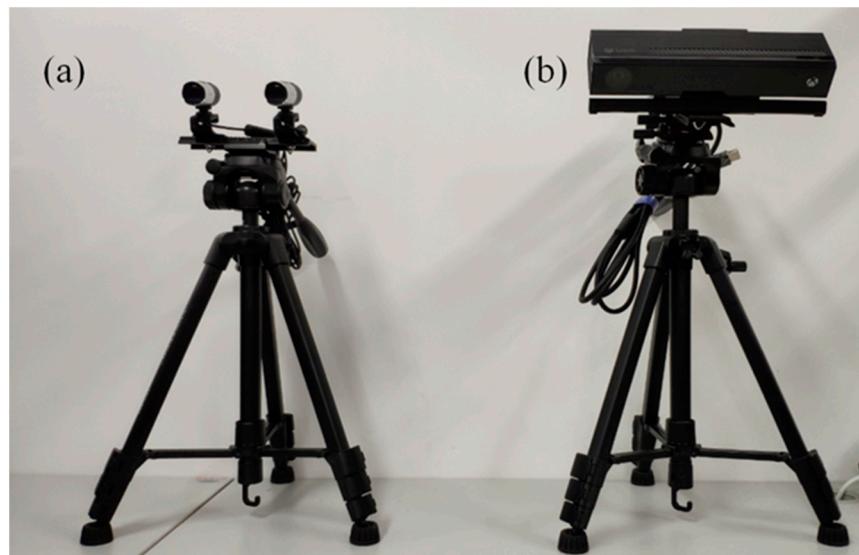


Figure 1. Two types of imaging sensors with tripods: (a) shows the binocular stereo vision system used in this research; (b) shows a Kinect V2 sensor mounted on a tripod.

2.1.2. Experiment Subjects

Three types of greenhouse ornamentals were adopted as research subjects in this paper: *Epipremnum aureum* (Linden & André) G.S. Bunting, *Monstera deliciosa* L., and *Calathea makoyana*. The Kinect V2 sensor was used to acquire the point cloud of an *Epipremnum aureum* sample plant and a *Calathea makoyana* sample plant. The binocular stereo vision system was used to reconstruct the 3D point cloud of a *Monstera deliciosa* sample plant.

2.2. Framework

Our individual leaf segmentation approach for plant point clouds consisted of three steps; the overview is shown in Figure 2. The first step was the pre-processing of the captured point cloud. Since the point cloud acquired from a 3D sensor has a large amount of noise points and non-leaf areas, such as pots and the ground, it was necessary to filter the original plant point cloud to generate a point cloud containing only leaves for further processing. In addition, the spatial characteristics of each point were also calculated in the pre-processing step, including the normal of the fitted plane in the neighborhood and the smoothness. In the second step, we carried out facet over-segmentation on the preprocessed point cloud data. In this step, a large number of uniformly distributed 3D facets that have a flat spatial structure were generated on the point cloud. The aim of over-segmentation is to cluster the points that share the same local spatial characteristics in advance. Therefore, it is a fine solution for the issue of individual leaf segmentation because aggregating bigger structures, such as facets to a leaf, is much easier than directly aggregating original points to a leaf in the cloud. We first utilized the computed spatial characteristics of each point in the first step to coarsely cluster the leaf points that are nearby and coplanar into the same facet. Then, to make the boundaries of over-segmented facets more regular, we employed local K-means clustering to refine all facets. In the third step, we realized individual leaf segmentation based on a facet region growing strategy. In this step, adjacency and co-planarity among facets were the conditions of facet region growing. After region growing, multiple facets were combined to form a larger spatial structure that could finally be regarded as a segmented leaf if it covers a sufficient number of points.

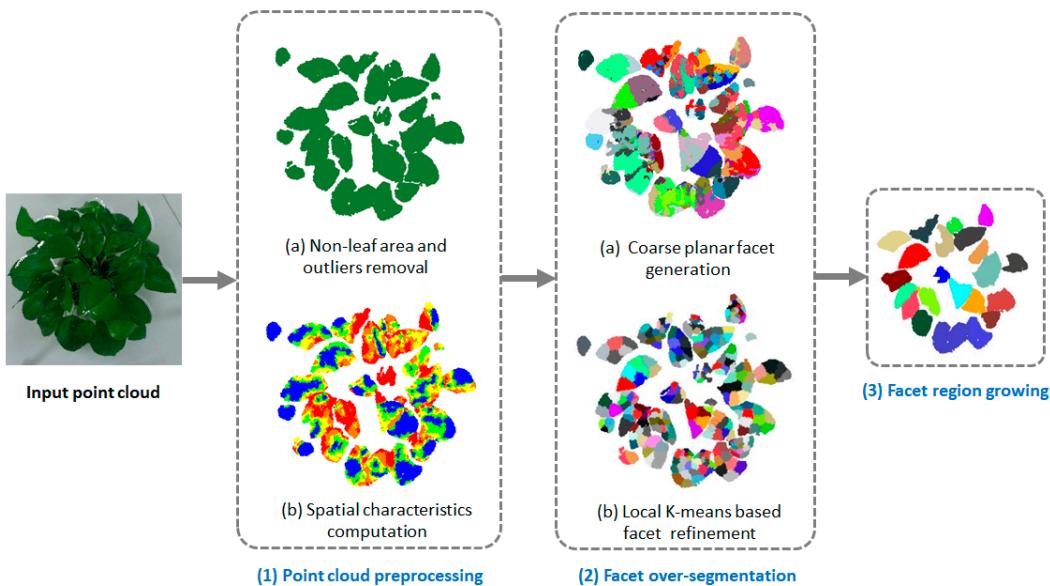


Figure 2. Overview of the proposed method.

2.3. Point Cloud Pre-Processing

2.3.1. Removal of Non-Leaf Areas and Outliers

During the 3D point cloud reconstruction process, most imaging sensors will perform interpolation in certain local areas of the original point cloud data, causing the boundary points of different planes in the point cloud to leap across different surfaces. Although interpolation smoothed the measurement errors for interior points, the spatial variances of boundary points became much larger than those of the interior points, which may result in connection of leaves that are actually isolated. Therefore, the removal of outliers in the point cloud is an important step in the framework. The paper applied different filtering methods on point clouds generated by two types of imaging methods. Three filtering operations were employed on the point cloud of the *Epipremnum aureum* sample plant. Firstly, we filtered out the points belonging to the ground by using the z-axis coordinates of the space. Second, if the number of points in the sphere of radius r centered at the current point was lower than a threshold n_1 , then the current point will be considered an outlier and discarded (i.e., the RadiusOutlierRemoval function in the PCL library). Lastly, we calculated the average distance between the k nearest neighboring points and the current point, and removed the neighboring point whose distance was larger than an upper bound derived by adding the average spacing to a coefficient n_2 multiplied by the standard deviation (the StatisticalOutlierRemoval function in the PCL library). Consequently, non-leaf points and outliers in the *Epipremnum aureum* point cloud were filtered. Only the first and the third operations were performed on the *Monstera deliciosa* point cloud to carry out filtering. Only the first and the second filtering operations were performed on the *Calathea makoyana* point cloud; there was no need to apply the last filtering operation because the leaves of *Calathea makoyana* are more regular than the leaves of *Epipremnum aureum*.

2.3.2. Using IPCA to Compute Spatial Characteristics of Each Point

We specify the spatial characteristics of each point \mathbf{x}_i in the point cloud χ in two parts: the normal \mathbf{n}_i of the fitted plane in a neighborhood of $\mathbf{x}_i = [x_i, y_i, z_i]^T$ and smoothness s_i of \mathbf{x}_i . The fitted plane $f_i = (\mathbf{X}_{3 \times K}, \mathbf{x}_i, \mathbf{n}_i)$ of \mathbf{x}_i is a 3-tuple, in which the K nearest neighborhood matrix $\mathbf{X}_{3 \times K}$ (including \mathbf{x}_i itself) of \mathbf{x}_i is calculated by iterative principle components analysis (IPCA). In each iteration, a plane passing through \mathbf{x}_i will be re-estimated by performing PCA on the inliers in the neighborhood. The notation $d(i, j)$ is defined as the Euclidean distance from the point \mathbf{x}_j in $\mathbf{X}_{3 \times K}$ to the fitted plane

f_i of \mathbf{x}_i . If $d(i, j)$ is larger than the fixed threshold parameter σ_1 , point \mathbf{x}_j will be removed from $\mathbf{X}_{3 \times K}$ before next iteration. The calculation of $d(i, j)$ is as follows:

$$d(i, j) = \frac{|\mathbf{n}_i^T \cdot (\mathbf{x}_j - \mathbf{x}_i)|}{|\mathbf{n}_i|} \quad (1)$$

The IPCA iterative process will cease when the size of $\mathbf{X}_{3 \times K'}$ remains unchanged, which also means K' decreases from K and finally becomes stable. The fitted plane $f_i = (\mathbf{X}_{3 \times K'}, \mathbf{x}_i, \mathbf{n}_i)$ calculated by IPCA will be very helpful in representing the spatial structure around each point \mathbf{x}_i . In IPCA, the three-by-three covariance matrix \mathbf{C}_i of $\mathbf{X}_{3 \times K}$ needs to be calculated for updating the fitted plane:

$$\mathbf{C}_i = \frac{1}{K} \mathbf{X}_{3 \times K} \cdot \mathbf{X}_{3 \times K}^T \quad (2)$$

In Equation (2), the data matrix $\mathbf{X}_{3 \times K}$ has been centralized. λ_1, λ_2 , and λ_3 are the three descending eigenvalues of the covariance matrix \mathbf{C}_i formed by $\mathbf{X}_{3 \times K}$. Therefore, the unit eigenvector corresponding to λ_3 can be regarded as the normal \mathbf{n}_i of the fitted plane of \mathbf{x}_i . The smoothness indicator s_i is defined as the ratio between λ_2 and λ_3 [28,29]. The greater the smoothness, the flatter the neighborhood of \mathbf{x}_i . Both the normal \mathbf{n}_i and the smoothness s_i will be updated in each iteration of IPCA. The pseudocode of the IPCA algorithm is given in Algorithm 1.

Algorithm 1 IPCA for computing the spatial characteristics of each point.

Input: Point Cloud χ , \mathbf{x}_i is any point in χ .

Parameters: Initial number of points in the neighborhood K .

Output: The unit normal vector \mathbf{n}_i , and the smoothness s_i .

```

1      for each point  $\mathbf{x}_i$  in  $\chi$  do
2          Initialize  $\mathbf{x}_i$ 's  $K$ -nearest neighbors data matrix  $\mathbf{X}_{3 \times K}$ .
3          repeat
4              Compute the covariance matrix  $\mathbf{C}_i$  of  $\mathbf{X}_{3 \times K}$  by Equation (2).
5              Compute the eigenvalues in descending order  $\lambda_1, \lambda_2$ , and  $\lambda_3$ , and their corresponding
6              eigenvectors  $\mathbf{v}_1, \mathbf{v}_2$ , and  $\mathbf{v}_3$  of  $\mathbf{C}_i$  by Eigenvalue Decomposition.
7               $\mathbf{n}_i \leftarrow \mathbf{v}_3, s_i \leftarrow \lambda_2 / \lambda_3$ .
8              Compute the distance  $d(i, j)$  between the point  $\mathbf{x}_j$  and  $\mathbf{x}_i$ 's current fitted plane by
9              Equation (1).
10             if  $d(i, j) > \sigma_1$  then
11                 remove the point  $\mathbf{x}_j$  from  $\mathbf{X}_{3 \times K}$ 
12             end if
13             until  $\mathbf{X}_{3 \times K}$  remains the same
14         end for

```

2.4. Facet Over-Segmentation

2.4.1. Seed Point Selection and Coarse Planar Facet Generation

The normal \mathbf{n}_i of the fitted plane in the neighborhood and the smoothness s_i of \mathbf{x}_i in the point cloud χ now have been computed according to Section 2.3.2. The coarse facet generation will be performed based on the normal and smoothness indicators. The essence of the process is to cluster the points in the point cloud with the same spatial characteristics. We carry out this coarse clustering by deploying a number of seed points first and then clustering similar points in a local neighborhood to each seed point. In order to produce reasonable seed points, we randomly select one point \mathbf{x}_i in χ and let the point \mathbf{x}_j with the greatest smoothness in the K -nearest neighborhood $\mathbf{X}_{3 \times K}$ of \mathbf{x}_i to be a seed point. Next, based on the seed point \mathbf{x}_j , a facet whose points have the same spatial characteristics will be established by local region growing. There are three conditions for local point region growing: (i) the Euclidean distance between a candidate grow point \mathbf{x}_k and the seed point \mathbf{x}_j must be less than a threshold r_1 ; (ii) the angle between the normals of the fitted planes of the seed point \mathbf{x}_j and \mathbf{x}_k needs to be less than θ ; and (iii) the distance from the point \mathbf{x}_k to the fitted plane f_j of \mathbf{x}_j should be less than

σ_1 . The above conditions are used to judge whether x_k and x_j are near and coplanar, and the three conditions must be satisfied at the same time to allow the facet region of x_j to include x_k . If a point x_k has been included in a facet by the local region growing, then it will be labelled as used and extracted from χ . With the help of the above procedure, coarse planar over-segmented facets will be generated. The detailed pseudocode for this step is given in the first half of Algorithm 2.

2.4.2. Local K-Means Clustering Based Facet Refinement

In Section 2.4.1, points in χ with the same spatial characteristics have been coarsely clustered into over-segmented facets, and we have also obtained the collection χ_{seed} of seed points. The size of facets can be controlled by the parameters K and r_1 ; the parameter tuning process will be further discussed in Section 3. The idea of our local K-means clustering is similar to simple linear iterative clustering (SLIC) [28].

For facets refinement, a seed point x_{seed} will be first randomly selected from χ_{seed} and then a search sphere of radius r_2 centered at x_{seed} is formed. It should be noted that the sphere may possibly contain multiple seed points in χ_{seed} . In each sphere, the Euclidean distances from each point x_i in χ to all seed points (if there are any) will be calculated respectively. After all seed points in χ_{seed} have been processed with the above measure, each point x_i in χ is classified to its nearest seed point. After all points in the cloud have been classified, each cluster now becomes a new facet, and the position of each seed point will be updated as the mean of the points in the facet it belongs to. We repeat the above local K-means clustering until the cluster centers of facets become stable. In experiments, we have observed that the boundaries of facets became stable after 3 to 5 iterations on several kinds of point clouds. Therefore, we fix the maximum number of iterations for local K-means to be 10. The detailed pseudocode for facet refinement is given in the second half of Algorithm 2.

Algorithm 2 Facet over-segmentation.

Input: Unit normal vector n_i , and the smoothness s_i of each point x_i .

Parameters: r_1 , θ , and σ_1

Output: the collection F of facets, the seed point set χ_{seed} .

```

1       $\chi_{\text{seed}} \leftarrow \emptyset$ 
2      for each unused point  $x_i$  in  $\chi$  do
3          set the point  $x_j$  with the largest  $s_j$  in  $X_{3 \times K}$  of  $x_i$  as the seed point  $x_{\text{seed}}$ 
4           $\chi_{\text{seed}}.\text{push\_back}(x_{\text{seed}})$ 
5          for each unused point  $x_k$  in  $\chi$  do
6              if  $x_k$  and  $x_{\text{seed}}$  satisfy the three conditions at the same time.
7                  (i)  $|x_k - x_{\text{seed}}| \leq r_1$ 
8                  (ii)  $\text{acos}(n_k \cdot n_{\text{seed}}) \leq \theta$ 
9                  (iii)  $d(\text{seed}, k) \leq \sigma_1$ 
10             then grow  $x_k$  belongs to the region of  $x_{\text{seed}}$ , and label  $x_k$  as used
11             end if
12         end for
13     end for
14     Set each region of  $x_{\text{seed}}$  in  $\chi_{\text{seed}}$  as a facet.
15     Set distance  $d_i = \infty$  for each point  $x_i$  in  $\chi$ .
16     repeat
17         for each cluster center  $x_{\text{seed}}$  in  $\chi_{\text{seed}}$  do
18             for each point  $x_i$  in a sphere of radius  $r_2$  centered at  $x_{\text{seed}}$  do
19                 Compute the distance  $D = |x_{\text{seed}} - x_i|$  (multiple seeds may exist)
20                 if  $D \leq d_i$  then
21                      $d_i \leftarrow D$ 
22                     classify  $x_i$  to the cluster of  $x_{\text{seed}}$ 
23                 end if
24             end for
25         end for
26         Points that do not belong to any facet are classified to its nearest seed points. Each
27         cluster of a seed  $x_{\text{seed}}$  now becomes a new facet  $f_{\text{seed}}$ , and all facets form a collection  $F$ .
28         until the positions of seeds remain stable.

```

2.5. Facet Region Growing for Individual Leaf Segmentation

In Section 2.4, we have accomplished facet over-segmentation and obtained the collection \mathbf{F} of refined facets. The main idea of this section is to carry out facet region growing from each facet f_i in \mathbf{F} according to the facet adjacency and the coplanarity. Then, multiple facets will be combined into a larger structure that can be finally regarded a segmented individual leaf if it covers a sufficient number of points.

The facet region growing from a starting facet f_i to a facet f_j must satisfy two conditions: (i) the facet f_j is adjacent to f_i ; and. (ii) if x_i and x_j are the centers of the two facets respectively, $d(i, j)$, which represents the distance from x_j to the facet f_i , is less than a threshold σ_2 . If facets f_j and f_i satisfy both of these conditions, they are regarded as being located on the same leaf. A breadth-first search strategy is employed in the facet region growing, which is demonstrated by Figure 3. In the breadth-first facet region growing, we begin by defining a starting facet labeled by the number “1” in Figure 3a. The first round of facet growing incorporates three nearby facets with label “2” into the region (Figure 3b). Finally, after 4 rounds of growing, the total structure now covers 9 small facets that are assigned the same red color (Figure 3e). The pseudocode for facet region growing is listed in Algorithm 3.

Algorithm 3 Facet region growing.

Input: The collection of facts \mathbf{F} .

Parameters: σ_2

Output: The collection of individual leaves.

```

1      for each unused facet  $f_k$  in  $\mathbf{F}$  do
2          a temporal facet queue for breadth-first search  $\mathbf{A} \leftarrow \text{NULL}$ 
3          Set  $f_k$  as the starting facet of a new individual leaf.
4           $\mathbf{A.push\_back}(f_k)$ , and label  $f_k$  as used.
5          repeat
6               $f_i = \mathbf{A.pop\_front}$ 
7              for each unused facet  $f_j$  in  $\mathbf{F}$  do
8                  if  $f_j$  is adjacent and  $d(i, j) \leq \sigma_2$  then
9                       $\mathbf{A.push\_back}(f_j)$  and label  $f_j$  as used
10                     Grow  $f_i$  to  $f_j$ .
11                 end if
12             end for
13         until  $\mathbf{A} = \text{NULL}$ 
14     end for

```

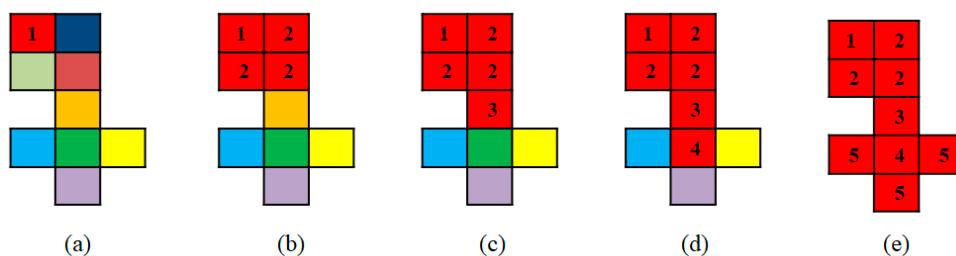


Figure 3. Breadth-first searching strategy for facet region growing: (a) shows the facets before region growing started; (b) shows the result of the first round of facet region growing; (c) shows the result of the second round of facet region growing; (d) shows the result of the third round of facet region growing; (e) shows the result of the final round of facet region growing.

3. Results and Discussion

3.1. Point Cloud Pre-Processing Results

The Kinect V2 sensor is used to acquire point clouds of an *Epipremnum aureum* sample plant and a *Calathea makoyana* sample plant, each containing a total of 217,088 points. The binocular stereo vision system is used to reconstruct the 3D point cloud of a *Monstera deliciosa* sample plant that contains 200,000 points. Three filtering operations are employed on the point cloud of the *Epipremnum aureum* sample plant. Firstly, we filter out the non-leaf points according to greenness and the z-axis coordinates of the point cloud. Secondly, we compare the number of points in the radius $r = 0.015m$ around each point with a threshold $n_1 = 85$. If the number of points is lower than 85, the current point will be considered an outlier and discarded. Last, we calculate the average distance between the $k = 40$ nearest neighboring points and the current point, removing the neighboring points whose distance is larger than an upper bound of adding the average spacing to $n_2 = 1$ multiplied by the standard deviation. However, only two filtering operations are performed on the *Monstera deliciosa* point cloud, with parameters set as $k = 25$ and $n_2 = 1$. Generally speaking, the radius-based filtering method can better preserve the dense parts of the point cloud, while the filtering method based on the average distance within a neighborhood is well suited for removing isolated outliers. The original and the pre-processed point clouds are shown in Figure 4.

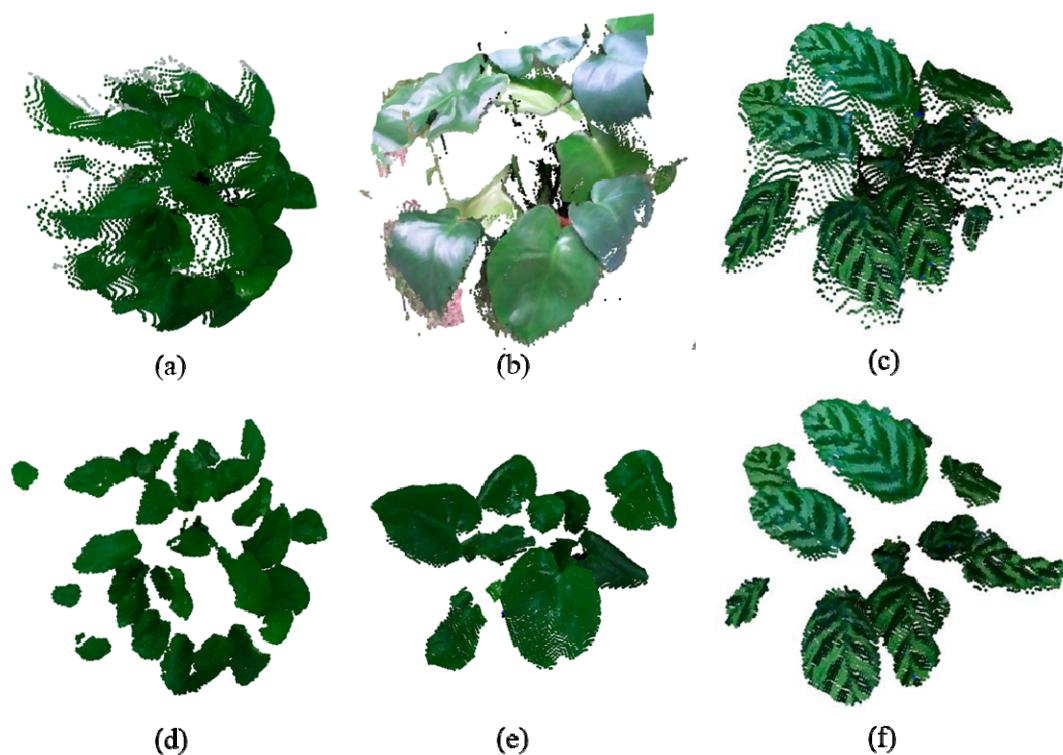


Figure 4. The original and pre-processed point clouds of the three greenhouse sample plants: (a) the original point cloud of *Epipremnum aureum*; (b) the original point cloud of *Monstera deliciosa*; (c) the original point cloud of *Calathea makoyana*; (d) the pre-processed result on *Epipremnum aureum*; (e) the pre-processed point cloud on (b); and, (f) the pre-processed result on (c).

3.2. Results of Facet over-Segmentation

Facet over-segmentation consists of two parts. The first part mainly selects seed points and carries out the coarse facet generation in the preprocessed plant point cloud. The second part conducts local K-means clustering to refine boundaries of facets. The threshold r_1 roughly defines the spacing between two different seed points in 3D space. However, coarse planar facet generation processes

each point only once; namely, when the current seed point has grown into a facet, the remaining seeds can only continue to grow on the remaining points in the point cloud. Thus, the coarse facets may not have the suitable structure and points in the point cloud may not be assigned to the correct seed points. Therefore, it is crucial to refine the boundaries of the facets by applying local K-means clustering, which automatically classifies each point to its closest facet. Local K-means clustering also updates the seed points (centers) of the refined new facets, respectively. The two parts combined can generate regular and uniform facets on plant point clouds. The granularity of the generated facets can be controlled by parameters K and r_1 . K is the number of points in the neighborhood when performing IPCA, which not only influences the normal of the fitted plane, but also influences the number of generated seed points (i.e., the number of facets). The smaller the value of K , the more seed points are generated. The parameter r_1 controls the minimum distance between seed points after the generation of coarse facets. Therefore, a smaller r_1 represents more facets input for local K-means clustering. Figure 5b shows the over-segmentation result of *Epipremnum aureum* when K is 100 and r_1 is 0.05 m; Figure 5c shows the over-segmentation when K is 40 and r_1 is 0.05 m; and, Figure 5d demonstrates the results when K is 20 and r_1 is 0.03 m. Figure 5f shows the over-segmentation result of the *Monstera deliciosa* point cloud when K is 100 and r_1 is 0.05 m; Figure 5g displays the result when K is 40 and r_1 is 0.05 m; and, Figure 5h is the result when K is 40 and r_1 is 0.03 m. Figure 5j shows the over-segmentation result of the *Calathea makoyana* point cloud when the value of K is 100 and r_1 is 0.05 m; Figure 5k shows the result when K is 40 and r_1 is 0.05 m; and, Figure 5l is the result obtained when K is 40 and r_1 is 0.03 m. It can be clearly seen that the granularity of the facets changes as K and r_1 decrease, and that a smaller K or r_1 tends to produce smaller facets.

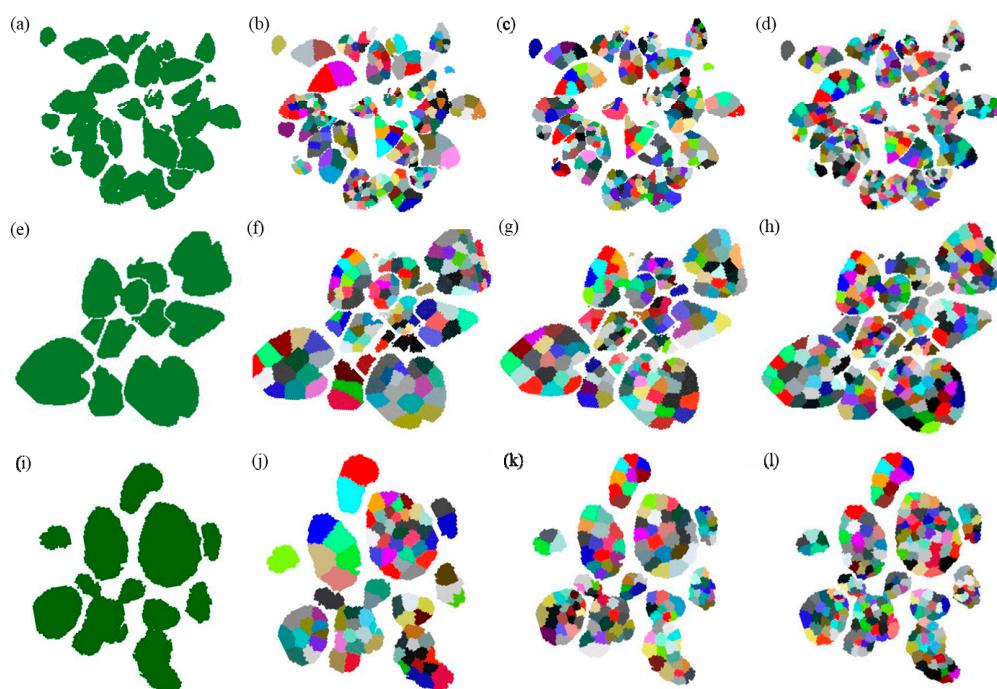


Figure 5. The facet over-segmentation of *Epipremnum aureum*, *Monstera deliciosa*, and *Calathea makoyana*; point clouds (a,e,i) show the top-views of the original three point clouds, respectively; (b) the over-segmentation result of *Epipremnum aureum* when K is 100 and r_1 is 0.05 m; (c) the result when K is 40 and r_1 is 0.05 m; (d) the result when K is 20 and r_1 is 0.03 m. (f) The over-segmentation result of *Monstera deliciosa* when the value of K is 100 and r_1 is 0.05 m; (g) the result when K is 40 and r_1 is 0.05 m; and, (h) the result obtained when K is 40 and r_1 is 0.03 m. (j) The over-segmentation result of *Calathea makoyana* when the value of K is 100 and r_1 is 0.05 m; (k) the result when K is 40 and r_1 is 0.05 m; and, (l) the result obtained when K is 40 and r_1 is 0.03 m.

3.3. Result of Individual Leaf Segmentation based on Facet Region Growing

The result of facet region growing is mainly influenced by the parameter σ_2 , which is used to prevent the growing region from extending to nearby leaves. The distance from the center of an adjacent facet to the current facet needs to be less than σ_2 when carrying out the breadth-first-search-based facet region growing. The result of individual leaf segmentation of the point cloud of *Epipremnum aureum* is shown in three different views in Figure 6a,d,g, in which σ_2 is 0.0055 m. The individual leaf segmentation results of the point cloud of *Monstera deliciosa* in three different views are shown in Figure 6b,e,h, where σ_2 is set as 0.0025 m. The individual leaf segmentation results of the point cloud of *Calathea makoyana* in three different views are shown in Figure 6c,f,i, where σ_2 is set as 0.0055 m (the same as for *Epipremnum aureum*).

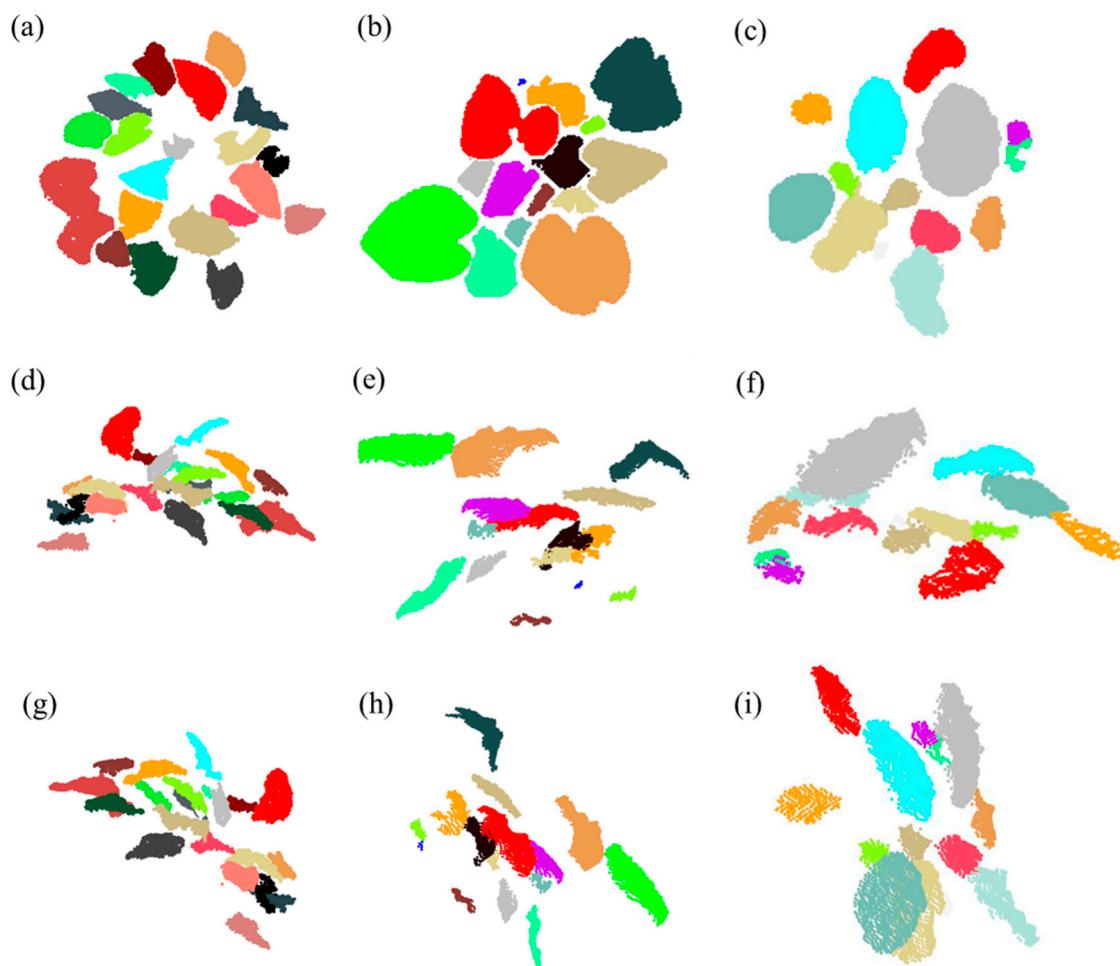


Figure 6. The results of individual leaf segmentation of *Epipremnum aureum*, *Monstera deliciosa*, and *Calathea makoyana*. The left column (a,d,g) shows three different views of the result of the point cloud of *Epipremnum aureum*, respectively. The middle column (b,e,h) shows three different views of the result of the point cloud of *Monstera deliciosa*, respectively. The right column (c,f,i) shows three different views of the result of the point cloud of *Calathea makoyana*, respectively.

3.4. Parameters

The parameters that appear in the proposed method are listed in Table 1. The parameters can be divided into two parts. The first part contains the filtering parameters, including r , n_1 , k , and n_2 . The second part contains the segmentation parameters, that is, the last six parameters. The parameter values for *Epipremnum aureum*, *Monstera deliciosa*, and *Calathea makoyana* are different, which is due to the different leaf types of the three species.

Table 1. Parameter settings for the three different point clouds.

Parameter	Description	Value for <i>Epipremnum</i>	Value for <i>Monstera</i>	Value for <i>Makoyana</i>
r	The radius parameter of the search sphere used for removing outliers.	0.015 m	—	0.015 m
n_1	A threshold that defines the minimum number of points in the search sphere.	85	—	50
k	The number of the nearest neighbors of the current point when computing the average spacing.	40	25	—
n_2	A threshold that is used to multiply the standard deviation of the average spacing.	1	0.1	—
K	The number of the nearest neighbor points used in IPCA.	20	40	40
r_1	A radius threshold used for coarse planar facet generation.	0.03 m	0.03 m	0.03 m
θ	An angle threshold for comparing two normals.	23°	23°	23°
σ_1	A threshold for measure the distance from a point to a plane.	0.025 m	0.025 m	0.025 m
r_2	A radius threshold used in local K-means clustering.	0.1 m	0.1 m	0.1 m
σ_2	A threshold that defines the distance from the center of one facet to another adjoining facet.	0.0055 m	0.0025 m	0.0055 m

The *Epipremnum aureum* plant has small and curved leaves and a crowded canopy, therefore we have to apply three different filters—the z-axis filter, the number-of-points-in-a-radius filter, and the statistical average spacing filter—together to remove the outliers. So, for a plant that has small and curved leaves, we strongly recommend applying all of the four filtering parameters to the point cloud. Conversely, for a plant with large and broad leaves, such as the *Monstera deliciosa* and *Calathea makoyana*, only two filters should be applied; the z-axis filter and the number-of-points-in-a-radius (or the statistical average spacing filter) are enough to generate a satisfactory point cloud. Larger leaves usually represent a simpler canopy structure because the area of leaves is already large enough to capture energy from the sun, and a simple canopy structure means that the leaves are easier to be segmented. In addition, we have also tested the number-of-points-in-a-radius filter and the statistical average spacing filter separately on the *Monstera deliciosa* sample plant, and the two processed point clouds are nearly identical. In our experience, the values for the four filtering parameters should be set to larger values on a plant with small and dense leaves.

There are not many differences in fixing the segmentation parameters for the three different species. Only the parameter K should be decided according to the average leaf size of the plant. For a plant with small leaves (such as *Epipremnum aureum*), we use a smaller K to generate enough facets for each single leaf. Conversely, for a plant with large leaves (such as *Monstera deliciosa* and *Calathea makoyana*), we use a larger K to avoid generating too many tiny pieces of facets on a single leaf. The parameters r , r_1 and σ_2 are related to the precision of the scanned point cloud. The average spacing of the dense points in the sample clouds is about 1.95 mm, while the spacing of sparse points is between 3 mm and 6 mm. Thus, all the distance thresholds are tuned with a small step of 0.5 mm to avoid producing unstable fluctuations in the segmentation result. n_1 , k , and K are the number thresholds that relate to the number of points in the point cloud. On average, there are about 400 to 500 points on a single leaf. Therefore, all the number thresholds are tuned with a small step of 5 to avoid producing unstable fluctuations in segmentation result.

It is also worth noting that if the facets are smaller, then the facet region growing will be easier because the distance from a facet center to another facet is smaller. This phenomenon is demonstrated

in Figure 7. When f_b breaks into two parts f_{b1} and f_{b2} , the distance from f_{b1} to f_a is much smaller than the distance from f_b to f_a . Therefore, if the granularity of over-segmentation is changed, the parameters for facet region growing should also be altered to avoid falsely combining facets on different leaves.

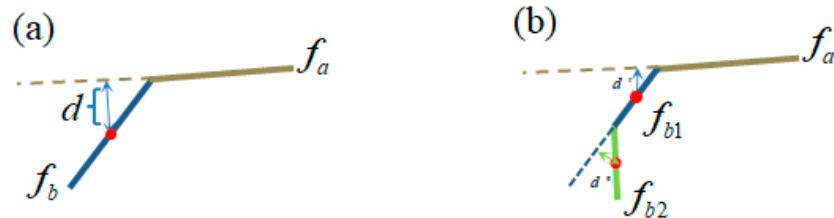


Figure 7. The granularity of over-segmentation affects the result of facet region growing. (a) shows the side view of a facet set containing only two adjacent facets, and (b) shows the side view of a facet set containing three adjacent facets. If the facet f_b in (a) breaks into two parts, f_{b1} and f_{b2} ; In (b), the distance from f_{b1} to f_a is much smaller than the distance from f_b to f_a , making the region growing to be easier.

3.5. Performance Evaluation

Individual leaves are calculated according to the facet growing result of the proposed method. We define the following metrics to better evaluate the proposed method, and the calculated metrics are analyzed at the individual leaf level.

- TP (True Positive): if a segmented leaf region covers more than 70% of the total number of the points of the real single leaf, the segmented leaf is then regarded as a TP.
- FP (False Positive): if two real leaves are segmented by the same segmentation region, then we regard it to be an FP.
- FN (False Negative): If more than 70% points of a real leaf are not covered by any segmentation, then we call it an FN.

Because we only provide point clouds of leaves for evaluating the proposed method, the True Negative (TN) does not exist. Based on the above definitions of TP, FP, and FN, we calculate three metrics—namely, the recall, precision, and F-measure—to quantitatively evaluate the proposed algorithm. The three metrics are explained as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (3)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (4)$$

$$\text{F - Measure} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \times 100\% \quad (5)$$

Table 2 lists the evaluation results of the proposed method on the three sample point clouds by these defined metrics. The proposed method successfully segmented 21 individual leaves from the point cloud of *Epipremnum aureum* with 23 real leaves, and segmented 15 individual leaves from the point cloud of *Monstera deliciosa* with 16 real leaves. The proposed method also correctly segmented 11 individual leaves from the point cloud of *Calathea makoyana* with 12 real leaves. The average precision of the method is higher than 90%, and the F-measure is higher than 95% on all three types of plants. The metrics show that the proposed method is able to obtain high TP numbers in the three cases. The FPs are mainly attributed to spatially adjacent leaves. For example, several leaves are overlapping and coplanar, making them extremely difficult to be correctly segmented even by a human (as illustrated in Figure 8a,b). We also manually evaluated the cover rates (defined as the ratio of a segmented leaf area to the original leaf area) of the three sample plants. The average cover

rate for the *Epipremnum aureum* sample plant is 94.35%; the maximum cover rate among all leaves is 100%, and the minimum is about 70.00%. The average cover rate of *Monstera deliciosa* roughly reaches 100% because our method generates almost perfect segmentation result on plants with large leaves. The average cover rate of *Calathea makoyana* is 92.08%; the maximum cover rate among all leaves of *Calathea makoyana* is 100%, and the minimum cover rate is about 50%.

Table 2. Quantitative results of the proposed method.

Plant Type	True Positive (TP)	Reference	False Positive (FP)	False Negative (FN)	Recall	Precision	F-Measure
<i>Epipremnum aureum</i>	21	23	2	0	100%	91.30%	95.45%
<i>Monstera deliciosa</i>	15	16	1	0	100%	93.75%	96.77%
<i>Calathea makoyana</i>	11	12	0	1	91.67%	100%	95.65%

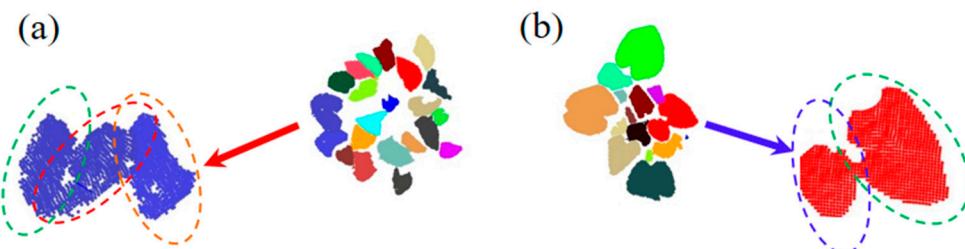


Figure 8. False Positives (FP) in two sample point clouds: (a) the two FPs for the point cloud of *Epipremnum aureum*; (b) the FP of the point cloud of *Monstera deliciosa* sample plant.

The proposed method took 13.04 s to complete the individual leaf segmentation on the point cloud of *Epipremnum aureum*, 4.85 s on the point cloud of *Monstera deliciosa*, and 4.50 s on the point cloud of *Calathea makoyana*. Thus, the proposed method can be used for quasi-real-time plant phenotyping.

Although the original goal of the proposed method was to segment individual leaves from a point cloud, we found that it has the potential to be applied to many applications that aim to segment regular surfaces and objects from a point cloud; for example, remote sensing, building information modeling (BIM), and simultaneous localization and mapping. In Figure 9, we apply the segmentation method to a point cloud containing a table surface and several objects (bottles and boxes), and obtain a satisfactory surface segmentation result, where all standing objects are correctly segmented from the table surface.



Figure 9. The segmentation result of our method on a point cloud containing a table surface and several objects (bottles and boxes): (a) the original point cloud with real colors, and (b) the segmentation result with different objects labeled in different colors, respectively.

4. Conclusions

In order to address the problems of image-based leaf segmentation methods, such as restrictions on species and vulnerability toward overlapping leaves, we propose a new individual leaf segmentation method for plant point clouds. The method can be divided into three steps: pre-processing,

facet over-segmentation, and facet region growing. Our method is not only suitable for the point clouds generated from two-camera stereo systems, but also handles the plant point clouds scanned from structured light sensors. In a performance assessment, our method achieved higher than 90% in average recall and higher than 95% in average F-measure for three different kinds of greenhouse ornamental plants. Although the original goal of our method was to segment individual leaves from a point cloud, we found that it has the potential to be applied to many applications that aim to segment regular surfaces and objects from a point cloud; for example, remote sensing, building information modeling (BIM), and simultaneous localization and mapping.

In this research, we only handled the plant point clouds that are sufficiently dense, which requires that a leaf should contain at least 50 points. Leaves that are both coplanar and adjoining are extremely difficult to segment, even for a human. For our approach, the FPs (False Positives) which stem from incorrectly growing facets across adjacent leaves are unavoidable. Although a FP segmentation does not influence the total leaf area computed, it reduces the number of detected leaves in the point cloud, resulting in an increase in the average leaf area index. For future research, we are interested in integrating popular deep learning algorithms into the 3D leaf segmentation method to separate each individual leaf from heavily clustered foliage. We also plan to test the proposed method on the plant point clouds scanned by many other kinds of 3D sensors with distinct accuracies.

Author Contributions: The theoretical framework was designed by D.L.; Y.C. and S.Y. conducted the experiments; X.C. carried out a part of the data processing; D.L. and X.-s.T. organized data and wrote the paper.

Funding: This work was jointly supported by the fundamental Research Funds for the Central Universities of China under Grants 2232016D3-20, 2232017D-08, and 2232017D-13; the Natural Science Foundation of China under Grants 61603089, and 61603090; and Shanghai Sailing Programs 16YF1400100 and 17YF1426100.

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

References

1. Neto, J.C.; Meyer, G.E.; Jones, D.D. Individual leaf extractions from young canopy images using Gustafson–Kessel clustering and a genetic algorithm. *Comput. Electr. Agric.* **2006**, *51*, 66–85. [[CrossRef](#)]
2. Xu, G.; Zhang, F.; Shah, S.G.; Ye, Y.; Mao, H. Use of leaf color images to identify nitrogen and potassium deficient tomatoes. *Pattern Recognit. Lett.* **2011**, *32*, 1584–1590. [[CrossRef](#)]
3. Kalyoncu, C.; Toygar, O. Geometric leaf classification. *Comput. Vis. Image Understand.* **2015**, *133*, 102–109. [[CrossRef](#)]
4. Kumar, N.; Belhumeur, P.N.; Biswas, A.; Jacobs, D.W.; Kress, W.J.; Lopez, I.C.; Soares, J.V. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 502–516.
5. Zhang, S.; Wu, X.; You, Z.; Zhang, L. Leaf image based cucumber disease recognition using sparse representation classification. *Comput. Electr. Agric.* **2017**, *134*, 135–141. [[CrossRef](#)]
6. Li, L.; Zhang, Q.; Huang, D. A review of imaging techniques for plant phenotyping. *Sensors* **2014**, *14*, 20078–20111. [[CrossRef](#)] [[PubMed](#)]
7. Fiorani, F.; Schurr, U. Future scenarios for plant phenotyping. *Ann. Rev. Plant Biol.* **2013**, *64*, 267–291. [[CrossRef](#)] [[PubMed](#)]
8. Gehan, M.A.; Fahlgren, N.; Abbasi, A.; Berry, J.C.; Callen, S.T.; Chavez, L.; Doust, A.N.; Feldman, M.J.; Gilbert, K.B.; Hodge, J.G.; et al. PlantCV v2: Image analysis software for high-throughput plant phenotyping. *PeerJ* **2017**, *5*, e4088. [[CrossRef](#)] [[PubMed](#)]
9. Zhou, J.; Applegate, C.; Alonso, A.D.; Reynolds, D.; Orford, S.; Mackiewicz, M.; Griffiths, S.; Penfield, S.; Pullen, N. Leaf-GP: An open and automated software application for measuring growth phenotypes for arabidopsis and wheat. *Plant Methods* **2017**, *13*, 1176. [[CrossRef](#)] [[PubMed](#)]
10. Scharr, H.; Minervini, M.; French, A.P.; Klukas, C.; Kramer, D.M.; Liu, X.; Luengo, I.; Pape, J.-M.; Polder, G.; Vukadinovic, D.; et al. Leaf segmentation in plant phenotyping: A collation study. *Mach. Vis. Appl.* **2015**, *27*, 585–606. [[CrossRef](#)]
11. Yin, X.; Liu, X.; Chen, J.; Kramer, D.M. Joint Multi-Leaf Segmentation, Alignment, and Tracking for Fluorescence Plant Videos. *IEEE Trans. Pattern Anal. Mach. Intel.* **2018**, *40*, 1411–1423. [[CrossRef](#)] [[PubMed](#)]

12. Viaud, G.; Loudet, O.; Cournède, P.H. Leaf Segmentation and Tracking in Arabidopsis thaliana Combined to an Organ-Scale Plant Model for Genotypic Differentiation. *Front. Plant Sci.* **2017**, *7*, 2057. [CrossRef] [PubMed]
13. Dobrescu, A.; Giuffrida, M.V.; Tsafaris, S.A. Leveraging multiple datasets for deep leaf counting. In Proceedings of the ICCV2017, Venice, Italy, 22–29 October 2017; pp. 2072–2079.
14. Morris, D.D. A pyramid CNN for Dense-Leaves Segmentation. *arXiv*, 2018; arXiv:1804.01646v1.
15. Pape, J.-M.; Klukas, C. Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images. In Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP), Swansea, UK, 7–10 September 2015; pp. 3.1–3.12.
16. Ferrara, R.; Virdis, S.G.; Ventura, A.; Ghisu, T.; Duce, P.; Pellizzaro, G. An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN. *Agric. For. Meteorol.* **2018**, in press. [CrossRef]
17. Yang, B.; Dai, W.; Dong, Z.; Liu, Y. Automatic Forest Mapping at Individual Tree Levels from Terrestrial Laser Scanning Point Clouds with a Hierarchical Minimum Cut Method. *Remote Sens.* **2016**, *8*, 372. [CrossRef]
18. Li, W.; Guo, Q.; Jakubowski, M.K.; Kelly, M. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 75–84. [CrossRef]
19. Li, S.; Dai, L.; Wang, H.; Wang, Y.; He, Z.; Lin, S. Estimating Leaf Area Density of Individual Trees Using the Point Cloud Segmentation of Terrestrial LiDAR Data. *Remote Sens.* **2017**, *9*, 1202. [CrossRef]
20. Teng, C.-H.; Kuo, Y.-T.; Chen, Y.-S. Leaf segmentation classification and three-dimensional recovery from a few images with close viewpoints. *Opt. Eng.* **2011**, *50*. [CrossRef]
21. Chaurasia, G.; Beardsley, P. Editable Parametric Dense Foliage from 3D Capture. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) (2017), Venice, Italy, 22–29 October 2017; pp. 5305–5314.
22. Li, D.; Xu, L.; Tang, X.S.; Sun, S.; Cai, X.; Zhang, P. 3D Imaging of Greenhouse Plants with an Inexpensive Binocular Stereo Vision System. *Remote Sens.* **2017**, *9*, 508. [CrossRef]
23. Paproki, A.; Sirault, X.; Berry, S.; Furbank, R.; Fripp, J. A novel mesh processing based technique for 3D plant analysis. *BMC Plant Biol.* **2012**, *12*, 63. [CrossRef] [PubMed]
24. Xia, C.; Wang, L.; Chung, B.K.; Lee, J.M. In situ 3D segmentation of individual plant leaves using a RGB-D camera for agricultural automation. *Sensors* **2015**, *15*, 20463–20479. [CrossRef] [PubMed]
25. Li, Y.; Fan, X.; Mitra, N.J.; Chamovitz, D.; Cohen-Or, D.; Chen, B. Analyzing growing plants from 4D point cloud data. *ACM Trans. Graph.* **2013**, *32*, 1–10. [CrossRef]
26. pointclouds.org. Available online: <http://pointclouds.org/> (accessed on 24 October 2018).
27. Yang, L.; Zhang, L.; Dong, H.; Alelaiwi, A.; El Saddik, A. Evaluating and Improving the Depth Accuracy of Kinect for Windows v2. *IEEE Sensors J.* **2015**, *15*, 4275–4285. [CrossRef]
28. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
29. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. *Remote Sens.* **2017**, *9*, 433. [CrossRef]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).