

A Leaf Segmentation and Phenotypic Feature Extraction Framework for Multiview Stereo Plant Point Clouds

Dawei Li[✉], Member, IEEE, Guoliang Shi, Weijian Kong[✉], Sifan Wang, and Yang Chen

Abstract—As a plant organ with the largest surface area, leaves are the main place where photosynthesis and respiration take place. High-throughput phenotyping of crop leaves is of great significance for breeding, growth monitoring, and increasing crop yield. Due to the highly complex and diversified plant structures, automated leaf segmentation and phenotypic feature extraction remain to be challenging tasks. In this article, we propose a novel five-stage framework that comprises multiview stereo point cloud reconstruction, preprocessing, stems removal in canopy, leaf segmentation, and leaf phenotypic feature extraction to carry out leaf phenotyping on two types of ornamentals—*Maranta arundinacea* and *Dieffenbachia picta*. The phenotypic traits such as the leaf area, leaf length, width, and leaf inclination angle for each single leaf are calculated and compared with ground truths. The experimental results show that the average accuracy of calculated leaf area of the two species reached 96.8% and 97.8%, respectively. The average errors of both the calculated leaf length and width of *Maranta arundinacea* are less than 4.0%, and for *Dieffenbachia picta*, the average errors of calculated leaf length and width are both no higher than 4.7%. The average errors of calculated leaf inclination angle for the two plant species are 2.9° and 3.0°, respectively.

Index Terms—Leaf phenotyping, leaf segmentation, multiview stereo (MVS), point cloud.

I. INTRODUCTION

PHENOTYPE is a group of biological traits that link the interaction between genotype and environment. Plant phenotypes intuitively reflect the growth state of plants, and observing the plant phenotypes is the basis of all breeding processes [1]. Plant phenotyping [2] is the comprehensive assessment of complex plant traits such as growth, development, tolerance, resistance, architecture, physiology, ecology, yield, and the basic measurement of individual quantitative parameters that form the basis for more complex traits. The quantitative analysis of plant phenotypes generally comprises the following aspects:

Manuscript received February 29, 2020; revised March 26, 2020; accepted April 20, 2020. Date of publication April 27, 2020; date of current version May 29, 2020. This work was supported in part by the Fundamental Research Funds for the Central Universities of China under Grant 2232019G-09, and in part by the National Natural Science Foundation of China under Grant 61603088 and Grant 61603089. (Corresponding author: Weijian Kong.)

The authors are with the College of Information Sciences and Technology, Donghua University, Shanghai 201620, China, and also with the Engineering Research Center of Digitized Textile & Fashion Technology Ministry of Education, Donghua University, Shanghai 201620, China (e-mail: daweilid@dhru.edu.cn; 2181346@mail.dhu.edu.cn; kongweijian@dhru.edu.cn; prizefighter@163.com; 425716127@qq.com).

Digital Object Identifier 10.1109/JSTARS.2020.2989918

leaf characteristics, stem characteristics, fruit traits, and root morphology [3]. As the organ that makes up the largest surface area of the plant, leaf is the main place where photosynthesis and respiration take place. Therefore, the leaf area, leaf length, width, and leaf inclination angle of plants are among the most critical biological factors [4]. With the growing demand for plant phenotype analysis, traditional manual measurement methods, which are often tedious and labor intensive [5], [6], are not applicable to high-throughput phenotyping tasks. In addition, the field of precision agriculture industry has been imposing higher standards on accuracy and precision for plant phenotyping at different scales [7].

Consequently, there exists a greater interest among academic communities and the agricultural industry in developing rapid and nondestructive techniques for high-throughput phenotyping. Generally, the first step of plant phenotyping requires accurate measurements of geometric features in specimens. This data acquisition process is also called plant digitizing [8], methods of which can be grouped into two classes—contact and contactless. The former class requires a certain form of probe to touch the surface of the plant for generating a 3-D model of the measured plant sample [9]–[12]. Despite high accuracy, the contact approaches rely on sophisticated manual operations and expertise. Therefore, researchers started to resort to noncontact techniques and significant progress has been made since then. At present, Light Detection and Ranging (LiDAR) sensing systems are widely used for 3-D reconstruction of tall trees [13], [14], corn [15], [16], cotton [17], and several other cash crops [18]–[20]. Although LiDAR has a high reconstruction accuracy, the equipment is so expensive that it is not suitable for the cost-sensitive agricultural industry. Structured light [21], [22] and time-of-flight [23] sensors also become popular tools for plant reconstruction because of their real-time performances [24], [25]. In addition, in order to further reduce plant digitizing costs [26]–[29], binocular stereo vision systems are used to reconstruct and analyze greenhouse plants and economic crops. However, this technique is only applicable to short plants, and is not able to handle dense canopies.

3-D reconstruction from multiple images, also referred as multiview stereo (MVS), has achieved great progress in recent years [30]. MVS is a low-cost method having high accuracy and resistance to light changes and imaging noise. By capturing images from different views and matching feature points, a complete 3-D model can be reconstructed, which avoids the

occlusion problem of the single-perspective imaging. Santos *et al.* [8] made a survey on plant digitalization, and showed that the structure from motion (SfM) method (one of the most important MVS methods) can be used for the phenotypic analysis of plants. Subsequently, Klodt and Cremers [31] evaluated the accuracy of MVS reconstruction on barley. Pound *et al.* [32] conducted accurate grid reconstruction of rice and wheat using surface estimation and boundary optimization. Johann *et al.* [33] segmented tomato plants at the organ level. In addition, Wang *et al.* [34] compared the precision and accuracy of 3-D laser scanning, MVS reconstruction, and 3-D digitized estimates on plant phenotyping.

Effective segmentation and feature analysis of 3-D point cloud is the key to digitization and automatic plant phenotyping. But how to effectively separate stem part and canopy for plants, and further, how to segment all leaves separately and extract phenotypic traits remain to be difficult questions. For dense crops, the segmentation tasks are even more challenging. In recent years, many new methods for the segmentation and analysis of 3-D models of different plants have been proposed. Paproki *et al.* [35] realized the separation of the leaves and the stem system of the cotton plant with a “hybrid” segmentation pipeline, and fitted leaf sagittal and coronal planes to calculate leaf length and width. Muller-Linow *et al.* [36] designed a half-automated graph-based method to segment individual leaves of young sugar beet plants. Duan *et al.* [37] proposed an octree algorithm to separate point clouds into several primary groups, and they extracted leaf midribs with local polynomial functions and leaf width with quantile regression. Itakura *et al.* [38] used simple-projection and attribute-expansion methods to segment leaves and analyzed the segmentation accuracy on six plants. Jin *et al.* [39] proposed median normalized vector growth to segment stems and leaves of maize, and extracted phenotypic traits at different levels including leaf, stem, and the individual plant. Su *et al.* [40] and Li *et al.* [41] used the Difference of Normals (DoN) [42] to segment leaves of Magnolia and maize, respectively; and leaf area index can then be estimated from segmented leaves. In addition, Zermas *et al.* [43] used nodes in 3-D skeleton to split corn overlapping leaves in 3-D point clouds. At the same time, the self-organized map [44] algorithm is used to carry out meshing on leaf point clouds and to calculate the leaf area. This method is also robust to certain species that have holes on leaves.

Most of the current point-cloud-based methods for leaf segmentation or leaf phenotypic analysis are not suitable for species with dense canopies. Some methods are tailor-made for only one or a specific class of plants with simple structures. In addition, existing methods have problems on automation, accuracy, and even real-time performance, which clearly restrict promotion and application. Motivated by the above issues, in this article, we first acquire 3-D point clouds of two types of plants by MVS, and then apply a stems removal process to obtain stem-free dense point clouds. After that, a region growing algorithm based on multiple features is used to segment individual leaves in the canopy. Finally, the phenotypic traits such as the leaf area, length, width, and the leaf inclination angle for each single leaf are extracted by using facet oversegmentation and the minimum

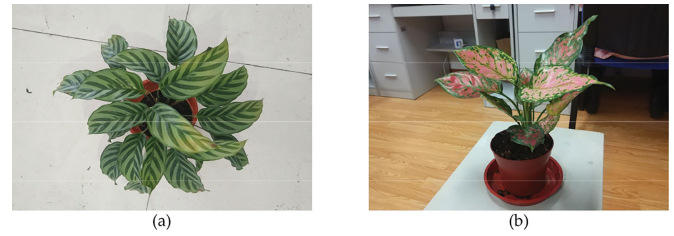


Fig. 1. Two types of potted ornamentals studied in this article. (a) is a sample plant of *Maranta arundinacea*, and (b) is a sample of *Dieffenbachia picta*.

3-D bounding box. Our contributions can be summarized into the following four aspects:

- 1) We successfully separate pure leaf parts from nonleaf parts (stems and background points) with a process designed for stems removal, which includes DoN thresholding and leaf point filling back.
- 2) We design an automatic individual leaf segmentation algorithm that is capable of separating overlapping leaves. Curvature features are used to remove the overlapped leaf edges. Second, then the canopy point clouds of two types of ornamentals are accurately segmented into individual leaves based on multifeature region growing.
- 3) We propose an accurate leaf area estimation algorithm based on facet oversegmentation and Delaunay triangulation for presegmented canopy point cloud. Comparing with two other methods for leaf area calculation, the proposed method leads in accuracy, reaching an average leaf area accuracy of 96.8% for *Maranta arundinacea* and 97.8% for *Dieffenbachia picta*, respectively. We also carry out parameter tuning of leaf area estimation and find the optimal parameter configuration.
- 4) Finally, principal component analysis (PCA) is used to generate the 3-D minimum bounding box for each segmented leaf to rapidly extract phenotypic information such as leaf length, width, and leaf inclination angle. Experiments show that both the average errors of leaf length and width of *Maranta arundinacea* are both less than 4.0%, and for *Dieffenbachia picta* the errors are both no higher than 4.7%. The average errors of calculated leaf inclination angle for the two plant species are 2.9° and 3.0°, respectively.

II. MATERIALS AND METHODS

A. Platform and Experimental Subjects

The processing unit is a desktop with an Intel Core i7-4770 CPU, 16 GB RAM, and an NVIDIA GTX 1080Ti GPU. The software environment includes Microsoft VS2015 with a PCL library [45] and VisualSfM [46], [47], which are all operated under Windows 10. In experiments, we used the camera (IMX378, Sony, Tokyo, Japan) from a cell phone (MI 5s, MI, China) to capture images of our experiment subjects for generating point clouds.

Two types of potted greenhouse ornamentals, *Maranta arundinacea* and *Dieffenbachia picta* (Fig. 1), are adopted as research

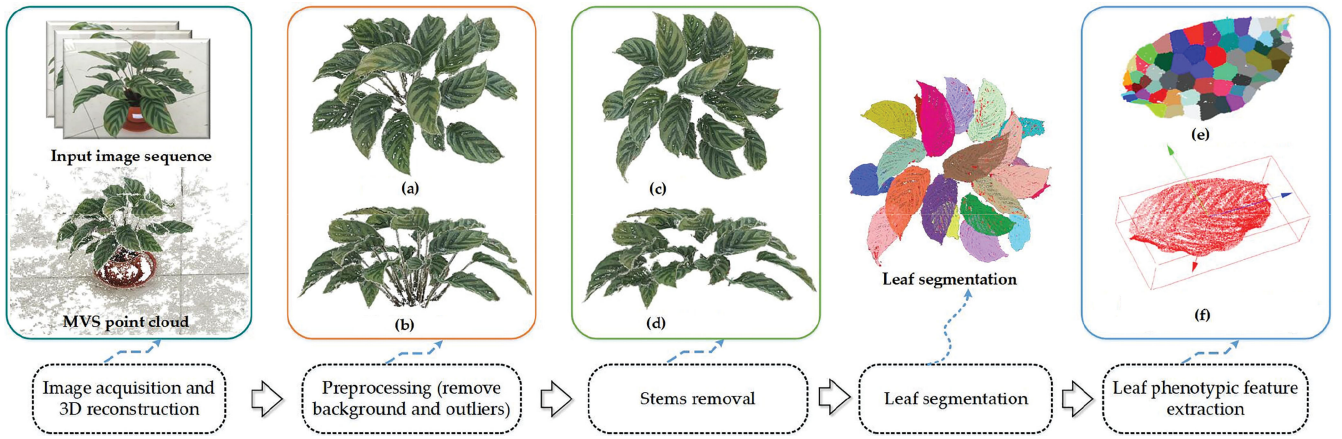


Fig. 2. Overview of the proposed method. (a) Top view. (b) Side view. (c) Top view. (d) Side view. (e) Leaf area calculation. (f) Calculation of leaf length, width, and inclination angle.

subjects in this article. A total of four samples with two in each plant species are experimented. The leaves of the two types differ a lot in texture, color, and shape, so that they are good samples to test the generality of the proposed leaf segmentation and phenotyping method. Comparing to *Dieffenbachia picta*, samples of *Maranta arundinacea* are taller, and have denser foliages.

B. Framework

The proposed method can be divided into five stages, and the diagram of the framework is shown in Fig. 2. The first stage is to construct accurate 3-D point clouds of plants by utilizing the MVS technique. In this stage, images of the subject are first captured from various directions. Then, the scale-invariant feature transform (SIFT) descriptor [48] is used to detect key matching points and to search the correspondences among images. Later, a sparse point cloud can be generated via bundle adjustment [49]. At last, clustering views for multiview stereo (CMVS) [50] is applied to the sparse model for producing a dense point cloud of the subject plant. The output point cloud of the first stage contains a lot of background information and noise that may interfere with the leaf segmentation step that follows. Therefore, in the second stage, we preprocess the point clouds by removing noncanopy areas (e.g., pot and ground) and suppressing noise points. In the third stage, we design a stems removal process which employs DoN [42] and Euclidean clustering to remove the stem system in the canopy. And after DoN, a neighborhood point search is applied to fill back some leaf points that are falsely removed together with the stems. In the fourth stage, the curvature feature of each point in the point cloud is computed first, and the points whose curvature values are higher than a threshold are removed from the point cloud because they are very likely to belong to the overlapping areas among different leaves. Afterwards, a region growing algorithm based on multifeatures is applied to segment individual leaves in the canopy. In the last stage, we measure phenotypic features such as the leaf area, leaf length, width, and leaf inclination angle for each segmented leaf of the sample plants in a fully automatic way. For calculating

the leaf area, we apply the facet oversegmentation algorithm [51] on each single leaflet; then each leaf surface is decomposed into many 3-D facets that are spatially flat and smooth. PCA [52] is applied for calculating the normal of each facet, and by rotating each facet to align its normal with the z -axis (the gravity direction) in the 3-D coordinate system of the total plant, we can project the 3-D facet onto the XOY horizontal plane as a 2-D manifold. After applying Delaunay triangulation [53] on the projected 2-D manifold of each facet, the area of a facet is approximated by the sum of all triangle areas. Furthermore, the area of a leaf is then estimated by summing up all its facet areas. For calculating the length, width, and inclination angle of a leaf, we construct a 3-D minimum bounding box for each leaf. The orientation of the bounding box reveals the leaf inclination angle, and the dimension of the box packages its leaf length and width.

C. 3-D Reconstruction of Plants

SFM [54] is an effective MVS method for constructing a 3-D point cloud with a series of 2-D images taken around the subject. In order to generate an accurate point cloud, images should have a short baseline and incorporate the subject as complete as possible. An SFM pipeline can be summarized as follows:

- 1) Detecting the feature points of each image.
- 2) Pairing the feature points between images.
- 3) Calculating the intrinsic and extrinsic parameters of the camera, and generating a sparse 3-D point cloud.

Specifically, the SIFT descriptor is employed to extract feature points. The approximate nearest neighbor algorithm [55] is utilized to match feature points under a small amount of computation, and the feature pairs below a threshold are removed. On remaining feature pairs, the fundamental matrix, camera parameters, and the projection matrix are estimated by using Random Sample Consensus [56]. The matched feature pairs can be further refined in the estimation of the fundamental matrix. Bundler adjustment [49] is applied for the final optimization.

Images are processed one by one in the above pipeline to add points to the 3-D point cloud. The sparse point cloud generated



Fig. 3. Camera location distribution for MVS imaging of a sample of *Maranta arundinacea* by VisualSFM and several image examples of the sequence. (a) Camera location for MVS imaging. (b) Image 10. (c) Image 15. (d) Image 20.

by SFM only covers the main skeleton of the object and lacks surface and texture information. Therefore, we utilize CMVS to build a dense point cloud based on the main 3-D structure. The points with strong color and geometric consistencies are added to the sparse SFM model to produce a dense and texture-rich point cloud in CMVS. A GUI-based end-to-end 3-D reconstruction tool—VisualSFM [46], [47], which encapsulates the basic SFM pipeline and CMVS, can be easily operated to generate a dense point cloud. In Fig. 3(a), we generate the point cloud of a sample plant by using VisualSFM, together with the camera location distribution for capturing the image sequence in the same coordinate system with the plant model. We also show several image examples in the image sequence in Fig. 3(b)–(d). The point cloud generated by MVS is unitless, so we place a standard box with the plant to scale the point cloud into its real-world size. The ground plane is treated as the horizontal plane of the plant coordinate system, and the plane normal is defined as the gravity direction.

D. Point Cloud Preprocessing

Due to the limitations of the method and the experimental conditions, the reconstructed plant point cloud is rife with noise, outliers, and background points (an example can be seen in the MVS point cloud in Fig. 2). These points can seriously interfere with leaf segmentation and plant phenotyping algorithms, resulting in incorrect results. Three filters are sequentially applied to suppress background points and noise points in the generated point clouds. We set up a plant coordinate system by referring to the gravity direction as the Z-axis, and the XOY plane represents the horizontal plane. The first filter is a spatial region filter that only keeps the points inside a $0.5 \times 0.5 \times 0.5$ m cube defined in

our plant coordinate system, so that the points from the ground and the pot can be easily removed by applying this filter. The second filter uses the color information to remove noncanopy points. If the color of a point is very close to the bright white or the pure black, then the point should be removed because it probably comes from the background or noise. The third filter is a radius-based outlier filter [45]. If the number of points in the sphere of radius R centered at the current point \mathbf{x} is lower than a threshold, then \mathbf{x} is considered as an outlier and will be discarded.

E. Stems Removal

In this subsection, we try to remove the stem points from the preprocessed point cloud by using normal features. The reasons for stems removal are two-fold. First, we are only interested in studying the leaf phenotypic features in this research. Second, points from stems can easily interfere with the identification and analysis of leaves in the point cloud. The normal is an important spatial feature for unorganized point clouds. Normals of points on a flat surface are smooth everywhere and do not change abruptly. But for the points on an irregular local structure, their normals vary greatly. On most plants, leaves are relatively flat and the normal directions seldom vary on the local region of the same leaf. But the normals on the stem differ greatly because of the cylindrical shape of the stem. Therefore, it is possible to distinguish leaf areas from the stem system in the point cloud by computing the changes of normals in a local region. DoN is a multiscale filtering operator that can be used to process massive 3-D point clouds [42]. By calculating the normal difference of the same point at different scales, we can judge whether the point is from a flat surface or not. A common approach for computing normals is to establish a set contains points in a local neighborhood with a fixed radius for each point, and then carry out PCA on the set to estimate the point normal. Formally, the DoN operator Δ_n for any point p_i in a point cloud P is defined as follows:

$$\Delta_n(p_i, r_1, r_2) = \left\| \frac{\mathbf{n}(p_i, r_1) - \mathbf{n}(p_i, r_2)}{2} \right\| \quad (1)$$

where r_1 and r_2 (r_1 is smaller) represent the two different support radii at point p_i , respectively. $\mathbf{n}(p_i, r)$ is the normal of p_i with the support radius r . $\Delta_n(p_i, r_1, r_2)$ represents the magnitude of difference between the two normals under support radii r_1 and r_2 , respectively. Note that for each single plane, there exists two normal vectors in the opposite directions, and both are correct. Therefore, to avoid this ambiguity, the angle between the two normal vectors is calculated first to examine whether they are at the same side before calculating DoN by (1). If $\mathbf{n}(p_i, r_1) \cdot \mathbf{n}(p_i, r_2) < 0$, then one of the normal has to be reversed. The result of (1) takes the L2-norm because the magnitude of difference is much more stable than its direction in calculation. DoN tends to be small for a point lies on a flat region, but large for a point on a rugged surface. Therefore, we can separate leaf regions from stem regions in a canopy point cloud by thresholding the DoN values.

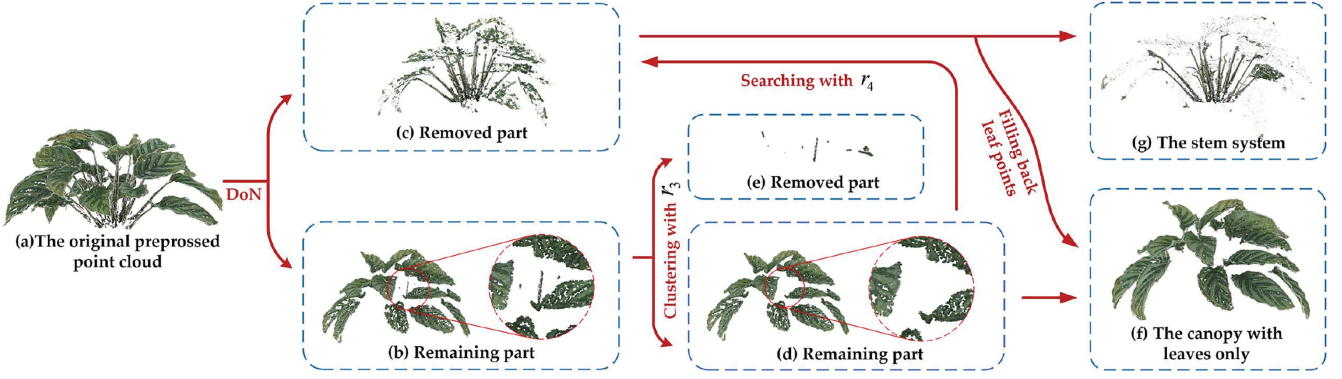


Fig. 4. Demonstration of the stems removal process with the canopy point cloud of *Maranta arundinacea*. (a) shows the original preprocessed point cloud of the sample plant. (b) shows the remaining point cloud after DoN. (c) shows the points removed by thresholding on DoN values, and most of the removed points are from stems. (d) is the remaining point cloud after Euclidean clustering. (e) is the isolated stem segments that are removed after Euclidean clustering. (f) is the final canopy point cloud that contains only leaves. It is obtained by filling back the falsely removed leaf points from the previous DoN result of (c). (g) shows the removed stem system from (a). By performing this process, recall of the leaf points increases dramatically from 19.4% to 97.4%, while the Precision remains unchanged.

After thresholding on DoN outputs, some stem parts (especially the ends near leaves) may still remain in the point cloud. A Euclidean clustering with radius r_3 is first performed on the point cloud after DoN, and all isolated clusters containing less than 200 points are removed to further filter fragments of remaining stem parts. Some leaf points may be falsely filtered as stem by the DoN threshold, resulting in false negatives. In order to suppress false negatives, we carry out a neighborhood point search to fill back some leaf points that are falsely removed together with the stems. After that, we traverse each point on the remaining point cloud, and conduct a search on the removed part of plant by DoN for points within a radius r_4 . Once searched, the points are filled back into the current point cloud, and these points are actually incorrectly removed by the previous DoN. Fig. 4 shows the full stems removal process for *Maranta arundinacea*; it outputs a canopy containing only leaves. After stems removal, the *Maranta arundinacea* canopy point cloud contains 637 353 points, the recall rate of leaf points increases from 19.4% (after DoN) to 97.4% (final canopy). Although multiple parameters exist in this process, they are not hard to tune, and we provide our suggestion in Section III-B.

F. Leaf Segmentation

At this point, we have removed the plant stems in the canopy point cloud and obtained the point cloud containing only leaves. Since different leaves are now detached from the stem system that connects them, they can be more easily separated in the 3-D space. Nevertheless, the leaves that are connected and overlapped with each other are still hard to separate, especially for those in a dense canopy. The “annoying” overlapping phenomenon among individual leaves seems to be extrinsically complicated (because leaves can overlap at various different positions and angles), but the solution is intrinsically simple. In order to separate overlapping leaves, we first try to recognize where they overlap by thresholding the curvature of every single point, and then remove those ambiguous points that have a curvature larger than a value. The curvature of each point is calculated in three small steps: 1) searching k -nearest neighbors of the

point ($k = 20$); 2) conducting PCA on the neighborhood, and compute the three eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$; 3) the curvature is calculated as $s = \lambda_3 / (\lambda_1 + \lambda_2 + \lambda_3)$. The smaller s means a smoother neighborhood [57], and the point is less likely to locate in the overlapping area.

The complete individual leaf segmentation is realized by applying a region growing algorithm that considers multiple features including point distance, the normal difference in angle, and curvature difference between the seed point and the searching point [45]. We demonstrate three common cases of leaf overlapping from the preprocessed point clouds in Fig. 5 to help explaining how the proposed segmentation works. On the second row of Fig. 5, we enlarge the overlapping areas and use dots and grids to display the real 3-D orientations of the leaves, respectively. The third row shows the normalized curvature of each point rendered in different colors, which clearly show that the points in overlapping areas usually have high curvature values. After removing the high curvature points, all individual leaves are correctly segmented with the adopted region growing technique on the fourth row of Fig. 5.

G. Leaf Phenotypic Feature Extraction

1) *Leaf Area Calculation*: In our previous work [58], we attempted to estimate the area of each individual leaf directly from 3-D point clouds. The Greedy Projection Triangulation (GPT) algorithm [59] is used to generate the leaf mesh, and then the leaf area is equivalent to the sum of all meshed triangular areas. Fig. 6(a) shows the result of GPT for a single *Maranta arundinacea* leaf. In order to reduce estimation errors caused by limited imaging accuracy and noise [the errors result in very rugged triangle mesh shown in the side view of Fig. 6(b)], it is necessary to conduct both downsampling and smoothing before GPT (shown on the rightmost part of Fig. 6). Despite its advantage in accuracy over traditional methods, the estimated area by [58] is usually larger than the ground truth, and the method cannot be applied to a class of plants that have porous leaves (such as *Monstera deliciosa*).

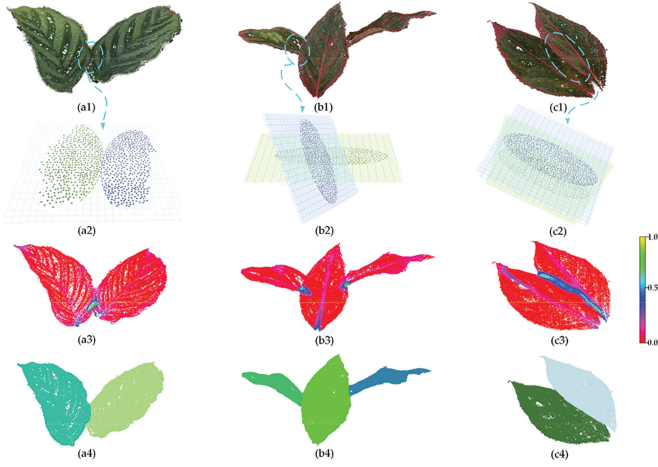


Fig. 5. Three leaf overlapping cases in the sample point clouds and the corresponding segmentation results of the proposed leaf segmentation technique. Columns 1 to 3 show the three different overlapping cases, respectively. (a1) shows two coplanar *Maranta arundinacea* leaves overlap at the boundary. In (a2), we enlarge the overlapping area and use dots and grids to show the 3-D orientations of the two leaves in this area, respectively. The plane of grids stands for the leaf plane. In (a3), we calculate the normalized curvature of each single point from the two leaves and demonstrate the values with different colors. It is straightforward to find that the points near the overlapping boundary have high curvature values compared to other leaf points. After removing the influence of high curvature points, the two leaves are segmented in (a4), in which each leaf is rendered with a different color. (b1) shows three *Dieffenbachia picta* leaves connecting with each other. In (b3), we render the curvature values with different colors. The three leaves are well segmented in (b4). (c1) shows two *Dieffenbachia picta* leaves touch with each other from two parallel layers. (c3) visualizes the curvature values. The final individual leaf segmentation result of (c1) is given in (c4). The curvature threshold is fixed to 0.3 throughout the article.

Instead of the direct calculation on 3-D structures, Koma *et al.* [14] project the leaf boundary onto a 2-D plane perpendicular to the average leaf normal, and then calculate the leaf area as the area of 2-D leaf shape. This method is effective for flat leaves, but fails on curved leaves because the projection of a curved surface must be smaller than the area of the surface itself. The method frequently outputs a smaller area than the real case as most natural leaves are not so flat.

GPT meshing [59] calculates the area from a very small scale that spans every three nearest local points on the 3-D surface. Though it enjoys a high precision, the estimated area is sensitive to local noise. The projection method [14] projects and calculates the leaf area from a global leaf scale. Though it is robust against imaging quality and noise, the estimation of area is not as accurate as [59]. We propose an accurate leaf area estimation algorithm that combines the benefits of both abovementioned methods. We first introduce a middle-scale information—the oversegmented facets, to partition the leaf surface into a set of facet pieces. Each facet is then projected onto a unique 2-D plane according to its normal, and the area of each facet is approximated by the projected 2-D area computed by GPT. Finally, the leaf area is the summation of areas of all its facets. Next, we will introduce the four steps of the proposed algorithm in detail: 1) facet oversegmentation, 2) facet projection, 3) Delaunay triangulation, and 4) outlier triangle removal.

a) Facet oversegmentation: Li *et al.* [51] first applied the facet oversegmentation technique to the problem of leaf segmentation in point clouds. Different from the leaf segmentation

task, in this step, we use this technique to partition the leaf into middle-scale regions. In oversegmentation, the iterative principle component analysis is first used to calculate the spatial features (including normal vectors and smoothness) of each point. A series of points with large smoothness are selected as seed points to start the initial growth of the facets. The growth from seed points must satisfy the following three conditions

- 1) The distance between candidate points and seed points should be less than a threshold r .
- 2) The angle between the candidate point and the seed point is less than a threshold θ .
- 3) The distance between the candidate point and the plane where the seed point locates should be less than a threshold σ_1 .

Candidate points that satisfy the above three conditions will be incorporated into the current facet. If a candidate point is not assigned to any seed points, it will become a new seed and participate in the calculation again. Finally, local k -means clustering is further used to refine the edges of all facets that have been preliminarily partitioned. The granularity of facets is mainly related to parameters K and r . K represents the number of neighborhood points used to calculate the spatial features, and r represents the minimum radius of a facet allowed in computation. In Section III-D.3, we will also discuss the influences of the two parameters on the calculation results.

b) Facet projection: In this step, each facet is projected onto the XOY plane for calculating its area. The average normal \mathbf{p} of the facet is first estimated by PCA. We now rotate the average normal to align with the standard normal $\mathbf{q} = [0, 0, 1]^T$, then the facet is automatically aligned with the XOY plane. In order to implement the alignment, the three-by-three rotation matrix $\mathbf{R}(\theta)$ from \mathbf{p} to \mathbf{q} should be first calculated according to Rodrigues Rotation Formula.

$$\mathbf{R}(\theta) = \mathbf{I} + \Omega \sin \theta + \Omega^2 (1 - \cos \theta) \quad (2)$$

in which \mathbf{I} is the three-by-three identity matrix. θ is the rotation angle between \mathbf{p} and \mathbf{q} . ω is the unit vector of the rotation axis defined as

$$\omega = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p}| \cdot |\mathbf{q}|} = [\omega_x, \omega_y, \omega_z]^T. \quad (3)$$

Ω is the crossproduct matrix of ω

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (4)$$

By left-multiplying the rotation matrix $\mathbf{R}(\theta)$ with original coordinates of all points in the facet, the facet is aligned with the XOY plane. By unanimously setting the Z -coordinates of all points on the aligned facet to zero, the facet is projected onto the exact standard 2-D plane XOY , which greatly facilitates the further calculation.

c) Delaunay triangulation: In this step, the Delaunay triangulation algorithm is used to reconstruct the projected 2-D facets, which is also much faster than the 3-D GPT meshing [59] because it works with only 2-D. The reconstructed 2-D mesh has three advantages: 1) the boundary of the mesh is convex and regular; 2) the triangulation result is unique; 3) and generally,

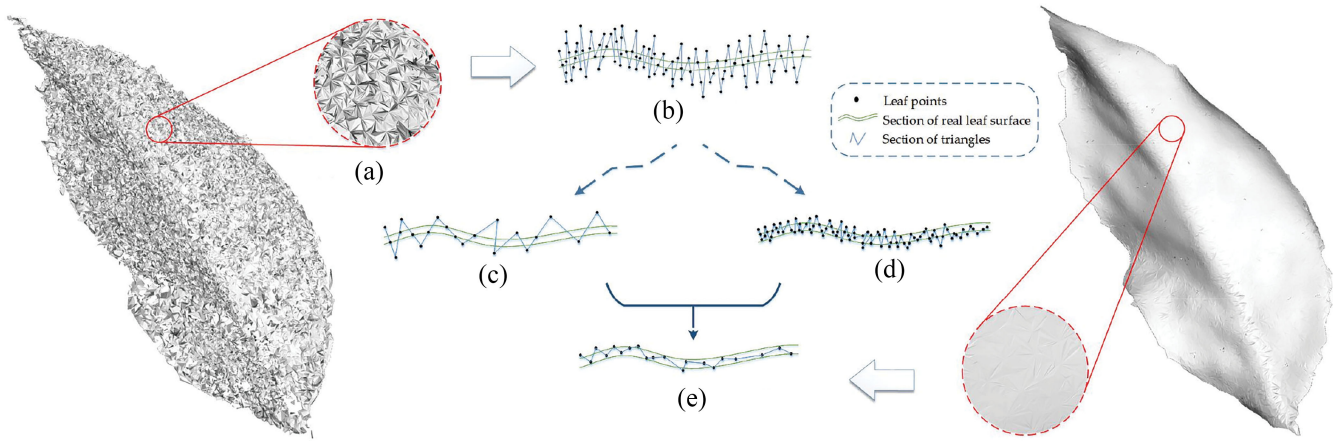


Fig. 6. Using GPT to mesh the leaf segment and estimate the leaf area. On the leftmost of this figure is a leaf point cloud of *Maranta arundinacea* directly meshed by GPT. (a) is a detailed top-view of a small part. Due to the rugged leaf surface caused by noise, the meshed leaf surface displays a diamond-like pattern. (b) is the side view sketch of the area in (a), and the rugged surface generates crisscrossing triangle meshes. (c) is the side view sketch of the area in (a) after point cloud downsampling. (d) is the side view sketch of the area in (a) after point cloud smoothing. (e) is the side view sketch of the area in (a) after both downsampling and smoothing. On the rightmost of this figure is the meshed leaf point cloud after global downsampling and smoothing; and it exhibits a highly regular and flat surface.

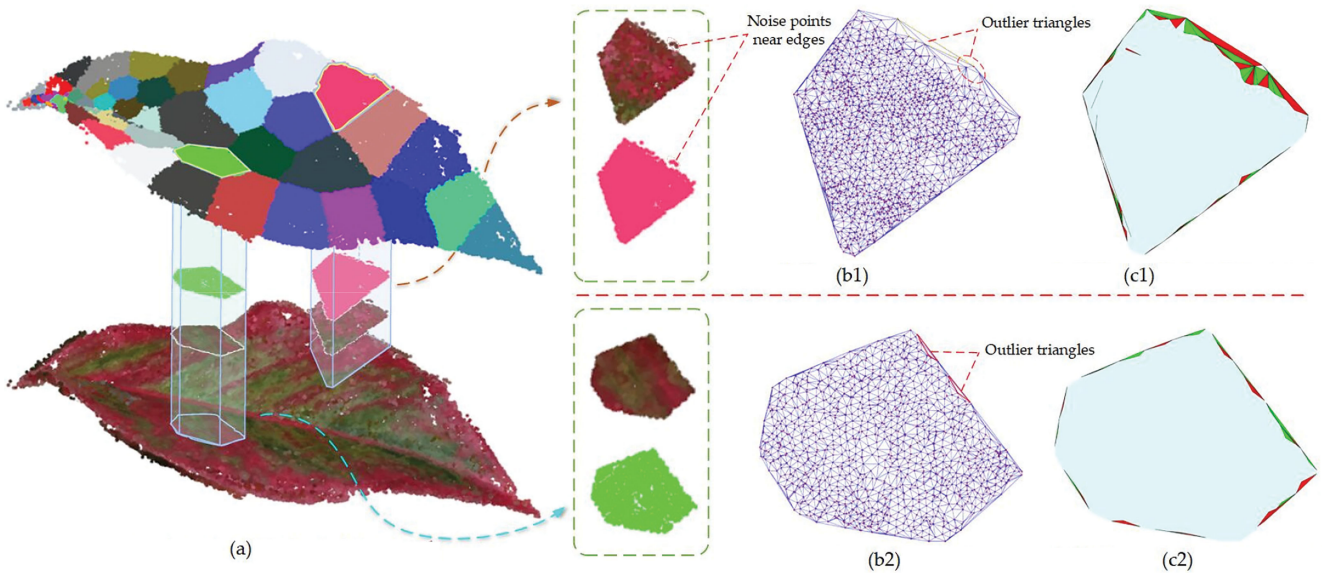


Fig. 7. Results of Delaunay triangulation of representative facets after facet oversegmentation. (a) is the result of leaf point cloud after facet oversegmentation, where the original point cloud is at the bottom and the facet results are at the top. We selected two small facets located at the edge and center, respectively, for analysis. (b1) and (b2) are the results of Delaunay triangulation subdivision of two small facets selected in (a) after rotating and projection. In (c1) and (c2), the outlier triangles are removed by the edge length threshold, where the red and green parts are the outlier triangles removed, and the cyan parts are the retained parts. When calculating leaf area, only cyan parts will be counted.

the triangles in the mesh are not overlapped, which contributes to an accurate area estimation.

Fig. 7(a) demonstrates the facet oversegmentation process on a real leaf, and we focus on two representative facets (one near the boundary, the other is near the center) for showing the Delaunay triangulation. Fig. 7(b1) and (b2) shows the results of Delaunay triangulation of the two facets after facet projection, respectively. Most generated triangles in the two projected areas are regular and they successfully cover their, respectively, facet. However, under the influence of noise points at the edges, several excessively large triangles are generated. Those big and slender triangles are outliers that should not be counted into the

facet area. In order to eliminate the errors introduced by outlier triangles, we add an extra outlier removal step after triangulation.

d) Outlier triangle removal: After Delaunay triangulation algorithm, all side lengths of triangles are ranked. Let s_{median} be the median of all side lengths, and λ be a coefficient. If the three side lengths of a triangle are all shorter than $\lambda s_{\text{median}}$, the triangle area is added to the facet area. Otherwise, the triangle is removed. This simple thresholding measure adapts itself well to different plant species and point clouds from different sensors. Fig. 7(c1) and (c2) shows the results of outlier removal by painting red and green colors to the removed triangles. Note that the majority of outliers lies at the boundary of facets. In

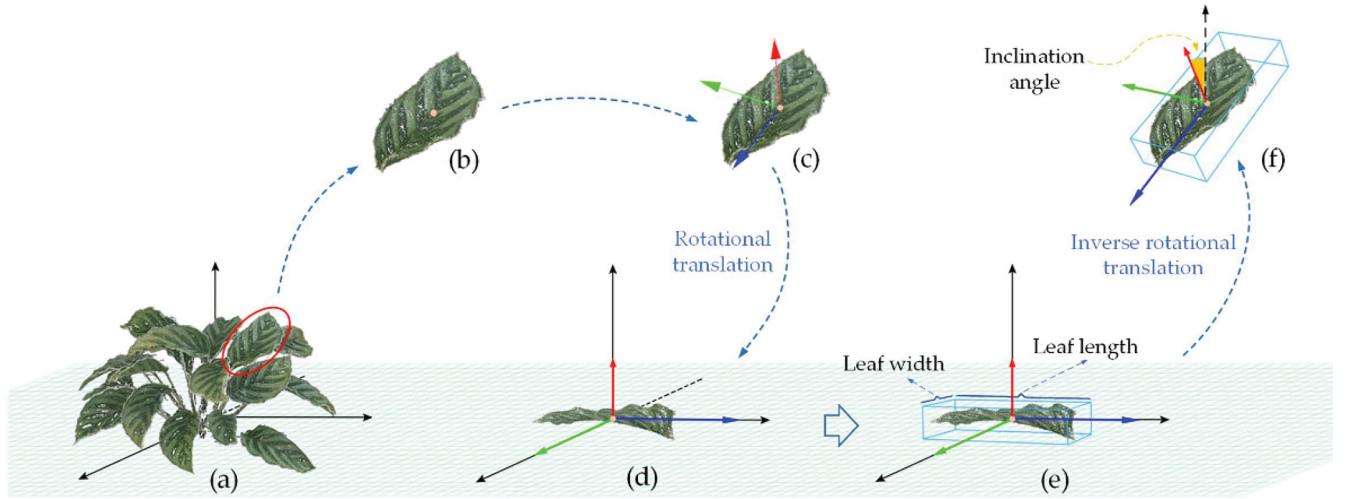


Fig. 8. Demonstration of the proposed leaf phenotypic feature extraction. (a) shows the original point cloud of a plant canopy under the plant (world) coordinate, in which the Z-axis is on the direction of gravity. A leaf from the canopy is separately studied in (b), the center of mass is shown in a yellow dot. (c) shows the established leaf coordinate system in which the three orthogonal PCA components on the leaf point set; the red arrow represents the direction of the leaf normal, while the blue and the green arrows represent the length direction and width direction, respectively. In (d), we use a rotational and translational transformation to align the leaf with the plant coordinate system. (e) shows the 3-D bounding box constructed by finding the max and min values in all three coordinates, respectively. (f) shows the bounded leaf transformed back into its original position by the inverse transformation, and the inclination angle is the organ angle between the leaf normal and the world z-axis.

Section III-D.3, we will further discuss the influence of parameter λ on the results of leaf area estimation.

2) Calculations of Leaf Length, Width, and Inclination Angle:

Except from leaf area, the leaf length, width, and its inclination angle are also important plant phenotypic parameters, which can directly reflect plant growth. Leaf inclination angle is the angle between the leaf normal and the zenith axis (z-axis). In fact, leaf inclination angle is also the angle between the leaf surface and the horizontal plane XOY . For calculating the leaf length and width, we try to fit a minimum 3-D bounding box to each individual leaf point cloud. The height of the box is aligned with the average normal of the leaf. The length and width of the bounding box then represent the leaf length and leaf width, respectively.

In fact, finding the minimum 3-D bounding box for leaves faces an ambiguity problem. Imagine that even on a 2-D case, a leaf image can have infinite 2-D bounding boxes centering at the leaf central point. For computing the 2-D leaf bounding box, [60] finds the bounding box that has the smallest area by rotating the bounding box according to the center point of a leaf image with a small angle at a time. Unfortunately, generalization of this method to the 3-D case is infeasible because the exhaustive search in 3-D space has a much higher time complexity than the 2-D case. Considering PCA can calculate orthogonal principal directions of a data cluster, it seems to be suitable for the estimation of three principal directions of the minimum 3-D bounding box because the leaf length, leaf width, and the leaf normal are regarded as mutually orthogonal. Generally, a leaf has a flat ellipse-like structure. So the principal component corresponding to the largest eigenvalue calculated by PCA can represent the direction of leaf length, and the second-largest principal component represents leaf width; the smallest one gives the direction of leaf normal. We obtain

three eigenvalues λ_1 , λ_2 , and λ_3 in descending order, whose corresponding eigenvectors are α_1 , α_2 , and α_3 after doing PCA on a leaf point cloud. An eigenvector matrix $\mathbf{A} = [\alpha_1, \alpha_2, \alpha_3]$ can be obtained by arranging the eigenvectors in columns. The coordinate of leaf center $\mathbf{c} = [x_c, y_c, z_c]^T$ is defined as center of mass of leaf in the plant coordinate system. Fig. 8 demonstrates our leaf phenotypic feature extraction. Fig. 8(a) shows the plants point cloud in the world coordinate system, where the z-axis of the coordinate system is the direction of gravity. Fig. 8(b) is one single leaf point cloud obtained by the previous segmentation algorithm. In Fig. 8(c), the center of mass of the leaf is first calculated and marked by a yellow dot, and the three principal component directions calculated by PCA are also shown. In order to facilitate the calculation of the minimum bounding box, we carry out a rotational and translational transformation to the original leaf point cloud to align the plant coordinate system. Fig. 8(d) is the transformed leaf point cloud. Now the outline of the minimum 3-D bounding box can be determined by the points that have lowest or highest values along each axis. Fig. 8(e) shows the generated minimum 3-D bounding box of the leaf in the plant coordinate system. The leaf length is the side length along the X-axis, and the leaf width is the side length along the Y-axis. Fig. 8(f) shows the 3-D bounding box after inverse transformation, the inclination angle is labeled in the orange color.

The homogeneous rotational and translational matrix \mathbf{V} that transforms the coordinate system of leaf to the plant coordinate system is given by

$$\mathbf{V} = \begin{bmatrix} \mathbf{A}^T & -\mathbf{A}^T \mathbf{c} \\ 0 & 1 \end{bmatrix}. \quad (5)$$

Every point \mathbf{x}_i of the leaf is first changed into the homogeneous coordinate as $\mathbf{x}_i = [x_i, y_i, z_i, 1]^T$, and then

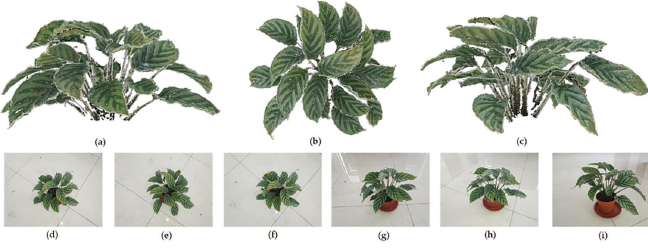


Fig. 9. Reconstructed point cloud of *Maranta arundinacea* sample 1 and several images used in SFM reconstruction. (a) Front view. (b) Top view. (c) Side view. (d) Image 1. (e) Image 11. (f) Image 21. (g) Image 31. (h) Image 41. (i) Image 51.

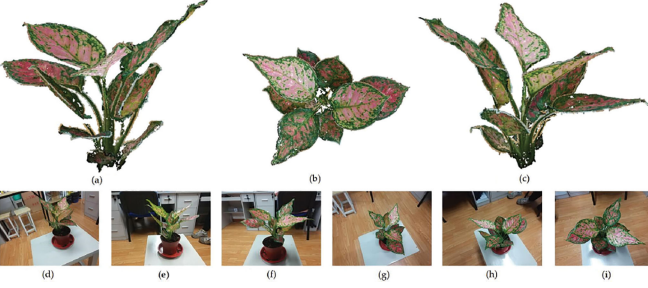


Fig. 10. Reconstructed point cloud of *Dieffenbachia picta* sample 1 and several images used in SFM reconstructions. (a) Front view. (b) Top view. (c) Side view. (d) Image 3. (e) Image 10. (f) Image 18. (g) Image 23. (h) Image 37. (i) Image 46.

left-multiplied by the four-by-four matrix \mathbf{V} to generate its coordinate $\mathbf{x}_i^w = [x_i^w, y_i^w, z_i^w, 1]^T$ in the plant coordinate system. The transformation pair between the leaf and the plant coordinate systems is given by

$$\begin{cases} \mathbf{x}_i^w = \mathbf{V} \cdot \mathbf{x}_i \\ \mathbf{x}_i = \mathbf{V}^{-1} \cdot \mathbf{x}_i^w \end{cases} \quad (6)$$

By using (6), we can easily generate a minimum 3-D bounding box for each single leaf and map it back to the original leaf.

III. RESULTS AND DISCUSSION

A. 3-D Reconstruction Results

We capture 102 images and 73 images at a resolution of 4000×3000 for the two sample plants of *Maranta arundinacea*, respectively. After the feature extraction and matching by VisualSFM, two dense point cloud with 1400333 points and 770037 points are constructed by CMVS, respectively. For the two *Dieffenbachia picta* sample plants, 58 images and 47 images at the same resolution are captured, respectively. We then construct two dense point clouds with 691735 points and 917327 points, respectively. The original dense point clouds contain much background information and noise. Three kinds of preprocessing filters (mentioned in Section II-D) are sequentially applied to effectively remove nonplant and noise points. After preprocessing, the *Maranta arundinacea* sample 1 has 688374 points, and the sample 2 has 331992 points, while 226878 points remain in the *Dieffenbachia picta* sample 1, and 138226 points remain in sample 2.

Figs. 9 and 10 show the 3-D reconstruction results of *Maranta arundinacea* sample 1 and *Dieffenbachia picta* sample 1 using SFM, respectively. Comparing to the plants in images, the reconstructed point clouds are complete and correct, and nearly no holes can be observed on the leaves, which satisfies the following leaf phenotyping requirements. The quality and quantity of images directly affect the 3-D reconstruction result. When less images are used, the method produce many holes on leaves. When too many images are used in reconstruction, the processing time will dramatically increase; and more importantly, the CMVS may produce redundant points, causing leaves to be thicker than real and eventually causing problems to the leaf segmentation algorithms. From the experience from experiments, we choose more than 40 short-baseline images to generate plant point cloud. Because *Maranta arundinacea* has a more complex canopy than *Dieffenbachia picta*, we input more *Maranta arundinacea* images to recover its canopy structure than the latter.

B. Result of Canopy Without Stems

The radius scales r_1 and r_2 are two important parameters in DoN thresholding. Ioannou *et al.* [42] argue that when the scale ratio r_2/r_1 is around 10.0, the planar and nonplanar components can be well separated on point clouds about urban street views. Take the *Maranta arundinacea* sample 1 and *Dieffenbachia picta* sample 1 as examples. The average point distance of *Maranta arundinacea* is 0.002 m, and the average diameter of its stem is about 0.016 m. For *Dieffenbachia picta*, the average point distance is 0.0005 m and the average diameter of stem is 0.005 m. The point density of the two clouds is very different from the point clouds of street view. Therefore, we need to select a new scale ratio r_2/r_1 to distinguish leaves from the stem system in our point clouds. After a parameter tuning experiment, we find out that the small radius r_1 should be larger than two times of the average point distance to extract enough surface information, and r_2 should be smaller than the average stem diameter in the point cloud to concentrate the local scale. In this article, we fix r_1 to be twice of the average point distance, and r_2 to be twice of r_1 . Therefore, for *Maranta arundinacea* sample 1, we set $r_1 = 0.004$ m, $r_2 = 0.008$ m. For *Dieffenbachia picta* sample 1, we set $r_1 = 0.001$ m, $r_2 = 0.002$ m.

It is also important to determine the value for thresholding the DoN result $\Delta_n(\mathbf{x}_i, r_1, r_2)$. If the magnitude of normal difference is larger than the threshold, we think that the point is located on a rugged surface, which also means the point is likely from the nonleaf part in the point cloud. Because the two different normals are unit vectors and the angle varies between 0° and 90° , the result of DoN lies between 0 and 0.72. We normalize the DoN to the interval of $[0, 1]$ to facilitate calculation. Fig. 11 shows the results of *Maranta arundinacea* sample 1 using different thresholds for $\Delta_n(\mathbf{x}_i, 0.004, 0.008)$, in which (a1) and (a2) show two views of the original preprocessed point cloud, respectively. The point cloud contains 688374 points in total, with the stem ground truth containing 34040 points, the leaf ground truth containing 654334 points. We only show the points whose DoN magnitudes are below the threshold in Fig. 11(b)–(d). Fig. 11(b1)

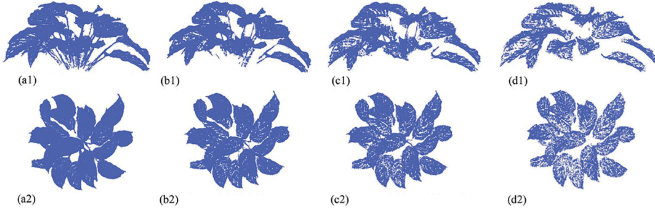


Fig. 11. DoN results of the *Maranta arundinacea* sample 1 point cloud using different thresholds for $\Delta_n(\mathbf{x}_i, 0.004, 0.008)$. Three DoN threshold values (0.1, 0.05, and 0.03) are used for stems removal. (a1) Original point cloud. (b1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.1$. (c1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.05$. (d1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.03$. (a2) Original point cloud. (b2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.1$. (c2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.05$. (d2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.008)| < 0.03$.

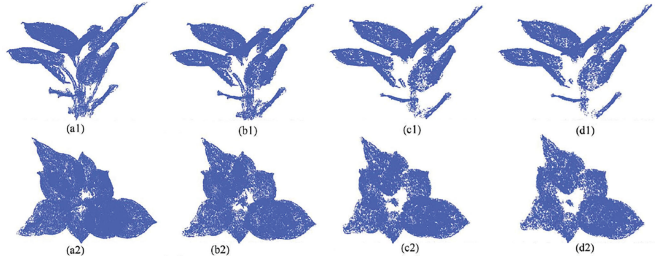


Fig. 12. DoN results of the *Dieffenbachia picta* sample 1 point cloud using different thresholds for $\Delta_n(\mathbf{x}_i, 0.001, 0.002)$. Three threshold values (0.2, 0.1, and 0.09) are used for stems removal. (a1) Original point cloud. (b1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.2$. (c1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.1$. (d1) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.09$. (a2) Original point cloud. (b2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.2$. (c2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.1$. (d2) $|\Delta_n(\mathbf{x}_i, 0.004, 0.002)| < 0.09$.

and (b2) is the results with the threshold at 0.1, and the number of points left is 512 961. Fig. 11(c1) and (c2) is the results with the threshold at 0.05, and the number of points is 275 239. (d1) and (d2) are the results with the threshold at 0.03, and the number of points is 126 940. Fig. 12 shows the results of *Dieffenbachia picta* using different thresholds for $\Delta_n(\mathbf{x}_i, 0.001, 0.002)$. The point cloud contains 226 878 points in total, in which the stem ground truth contains 11 570 points, and the leaf ground truth contains 215 308 points after manual segmentation. Fig. 12(b1) and (b2) is the results when the threshold is 0.2, and the number of left points is 181 873. Fig. 12(c1) and (c2) is the results with threshold at 0.1, and the number of left points is 124 902. (d1) and (d2) are results with threshold at 0.09, and the number of points is 113 544. For MVS point clouds, we suggest the threshold of DoN to be lower than 0.1. Finally, we set 0.03 for *Maranta arundinacea*, 0.09 for *Dieffenbachia picta*.

In Figs. 11(d1) and 12(d1), the stems of the two samples are completely removed, showing the proposed DoN filter to be feasible. Though small holes begin to show up on leaves in the two cases, the overall geometry of leaves is intact and well-preserved. The quantitative analysis of Figs. 11(d1) and 12(d1) shows that the precision values of leaf points after DoN are both close to 100%, but the recall rates are only 19.4% and 52.7%, respectively. Therefore, we apply the process in Fig. 4 to fill back some leaf points that were incorrectly removed by DoN to suppress false negatives.

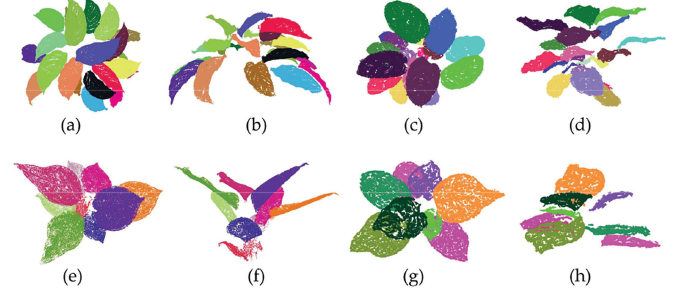


Fig. 13. Individual leaf segmentation results of four samples based on curvature thresholding and region growing. (a) and (b) are the top view and side view of the segmented *Maranta arundinacea* sample 1 result, respectively. (c) and (d) are the top view and side view of the segmented *Maranta arundinacea* sample 2 result, respectively. (e) and (f) are the top view and the side view of the segmented *Dieffenbachia picta* sample 1, respectively. (g) and (h) are two views of the segmented *Dieffenbachia picta* sample 2.

The parameters r_3 and r_4 are used to carry out Euclidean clustering and point filling, respectively. r_3 and r_4 can be both chosen as 2 to 10 times of the average spacing of the point cloud. In this article, we fix the parameters as $2r_1 = r_2 = r_3 = r_4$. After the stems removal process, the canopy point cloud of *Maranta arundinacea* sample 1 reaches 637 353 points, and its recall of leaf points increases from 19.4% (after DoN) to 97.4%. After stems removal, the canopy point cloud of *Dieffenbachia picta* sample 1 now contains 212 392 points, and the recall for leaf points increases from 52.7% (after DoN) to 98.6%.

C. Leaf Segmentation Results

In this subsection, the point cloud containing only leaves (from Section III-B) will be segmented into individual leaves by applying the approach described in Section II-F. On several kinds of plants, we have observed that the points from overlapped-leaf regions generally have a large curvature value that lies in the interval of $[0.4, 0.6]$. On all sample plants, we directly remove those points that have a curvature higher than 0.3. Fig. 13(a) and (b) is the results of the proposed leaf segmentation approach of *Maranta arundinacea* sample 1 in two different views, respectively. A total of 21 leaves are successfully segmented by the proposed method from a total of 23 leaves in ground truth of the plant, with an accuracy of 91.3%. Fig. 13(c) and (d) are the segmentation results of *Maranta arundinacea* sample 2 in two different views, respectively. A total of 20 leaves are segmented from a total of 21 leaves in the ground truth, with an accuracy of 95.2%. In Fig. 13(e) and (f), 9 leaves are finally segmented by the proposed approach from the original 10 leaves of *Dieffenbachia picta* sample 1, with an accuracy of 90%. In Fig. 13(g) and (h), all eight leaves from the original *Dieffenbachia picta* sample 2 are successfully segmented by the proposed method, with 100% accuracy. The missing leaves of the samples are all very small leaves locating at the bottom of the plant, where also receives the poorest quality in MVS reconstruction. If the quality of the reconstructed point cloud is further improved (using Lidar), the accuracy of leaf segmentation can also increase.

TABLE I
THREE METHODS ARE COMPARED OVER THE LAA MEASURE ON THE TWO TYPES OF PLANTS

Methods	Plant Types	Leaf 1	Leaf 2	Leaf 3	Leaf 4	Leaf 5	Leaf 6	Leaf 7	Leaf 8	Leaf 9	Leaf 10	Leaf 11	Leaf 12	Average
[59]	<i>Maranta arundinacea</i> Sample 1	93.7	93.4	91.2	90.9	95.7	93.1	90.9	94.0	92.1	91.2	96.6	93.5	93.0
	<i>Maranta arundinacea</i> Sample 2	86.3	90.1	92.6	92.2	85.0	84.2	89.5	90.6	86.7	89.9	87.0	/	88.6
	<i>Dieffenbachia picta</i> Sample 1	92.2	92.9	90.3	90.9	88.0	90.5	91.2	88.5	/	/	/	/	90.6
	<i>Dieffenbachia picta</i> Sample 2	93.0	89.7	93.1	90.7	96.7	93.4	95.8	98.1	/	/	/	/	93.8
[14]	<i>Maranta arundinacea</i> Sample 1	96.4	97.9	96.8	96.1	95.4	95.4	93.9	91.9	94.9	97.1	92.6	91.0	95.0
	<i>Maranta arundinacea</i> Sample 2	95.8	96.5	95.5	94.8	94.2	96.2	95.9	94.4	98.3	97.8	93.5	/	95.7
	<i>Dieffenbachia picta</i> Sample 1	96.4	95.0	98.6	94.6	98.7	96.8	92.4	96.9	/	/	/	/	96.2
	<i>Dieffenbachia picta</i> Sample 2	94.9	95.3	96.3	95.1	98.8	94.7	95.2	96.2	/	/	/	/	95.8
Ours	<i>Maranta arundinacea</i> Sample 1	95.0	98.8	98.7	99.9	96.8	99.5	98.8	96.5	96.6	92.5	93.3	94.0	96.7
	<i>Maranta arundinacea</i> Sample 2	97.4	96.8	96.4	97.9	97.2	95.5	96.9	96.1	98.0	98.4	95.8	/	96.9
	<i>Dieffenbachia picta</i> Sample 1	98.5	98.7	98.5	96.2	99.8	98.2	95.2	98.2	/	/	/	/	97.9
	<i>Dieffenbachia picta</i> Sample 2	97.1	97.5	98.9	96.1	97.1	97.6	99.0	98.2	/	/	/	/	97.7

The highest values are in bold face.

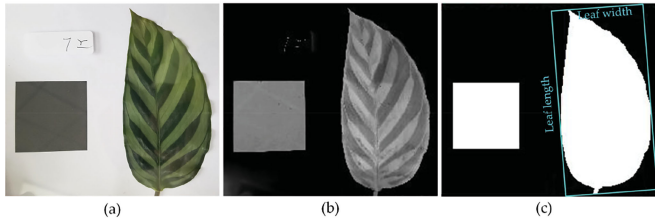


Fig. 14. Calculating ground truths of the leaf area, leaf length, and width using 2-D image processing. In (a), we clamp each leaf (nondestructively) with a 16 cm² reference square between two parallel pieces of glasses, and place them right below a camera to capture the image. (b) is the grayscale image that removes the background by frame differencing. (c) shows the binary image by applying the Otsu's method [61] on (b), and the 2-D minimum bounding box with clearly labeled truth leaf length and width is the result from applying [60].

D. Leaf Area Calculation Results

We select segmented leaves that are large enough to test the proposed leaf phenotypic feature extraction method. A total of 12 leaves and 11 leaves are selected from *Maranta arundinacea* sample 1 and sample 2, respectively. Eight leaves are selected from *Dieffenbachia picta* sample 1 and sample 2, respectively.

The ground truth of each leaf area is obtained by contrasting the pixels from the flattened leaf region with a standard square in standard images. In order to verify the accuracy and effectiveness of the proposed method, we compare it with GPT meshing [59] and the projection method in [14]. Experimental results show that the proposed method obtains the minimum error across the three methods on almost all leaves. To assure fairness, we separately conduct parameter tuning for the method [59], [14], and the proposed area calculation method to obtain their best results, respectively.

1) *Measuring the Ground Truth*: In order to quantitatively evaluate the leaf area estimation, an easy image processing method is designed to calculate the ground truth of each leaf area by comparing the flattened 2-D leaf shape with a reference square. The ground truths of the leaf length and width can also be computed in this way by following the method in [60]. Fig. 14 shows how the ground truths of leaf area, length, and width are measured from a single leaf in a nondestructive way. The ground

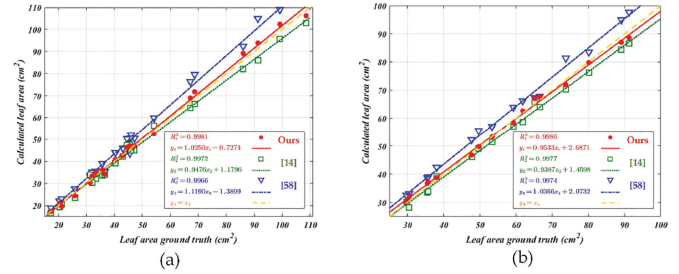


Fig. 15. Comparison of the three leaf area calculation methods on the four sample plants of the two species. (a) shows the results on *Maranta arundinacea*, in which each point is a leaf and the fitted line of our method are painted in red, the data points and the fitted line of [59] are painted in green, and the points and fitted line of [14] are painted in blue. The standard regression line $y = x$ is represented by an orange dotted line. (b) shows the results on *Dieffenbachia picta*. It can be seen that the proposed method has the best regression result against the ground truth of leaf areas across the three methods.

truth of leaf inclination is measured by a digital inclinometer (DXL360S, Jingyan, Shenzhen, China).

2) *Results of Leaf Area Estimation*: We compare the proposed leaf area calculation method with GPT meshing [59] and the 2-D projection method [14] in quantitative experiments. A new quantitative measure—leaf area accuracy (LAA), is defined in (7) to quantitatively evaluate the performances. The comparative results of the two plant types are shown in Table I, from which can be seen that the LAA of our method is superior to the other two methods in most cases. The best results in Table I are in bold face.

$$\text{LAA} = 1 - \frac{|\text{calculated leaf area} - \text{ground truth}|}{\text{ground truth}} \times 100\%. \quad (7)$$

Fig. 15 shows the regression comparison of three leaf area calculation methods on two plants. Note that the regression line of our method is the closest to $y = x$ across all compared methods on two species.

3) *Parameter Tuning for Leaf Area Estimation*: In our proposed method for estimating the area of each single leaf, facet oversegmentation is first applied on each leaf, and then Delaunay triangulation is applied on each projected facet to compute the 2-D area; finally, the leaf area is approximated by summing up all facet areas. The parameters in facet oversegmentation and

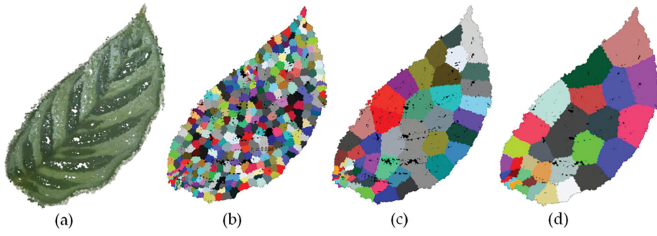


Fig. 16. Relationship between facet oversegmentation parameters r_1 , K with the number of facets. (a) is the original point cloud of *Maranta arundinacea* sample 1. (b) is the oversegmentation result with $r = 0.01$ m, $K = 100$; the number of facets is 375. (c) is the result with $r = 0.04$ m, $K = 150$; the number of facets is 58. (d) is the result with $r = 0.18$ m, $K = 200$; the number of facets is 26.

the outlier triangle removal can affect the accuracy of leaf area calculation; thus, it is necessary to tune the parameters for the optimal result.

The parameters influence the oversegmentation step by affecting the granularity of the facets (the sizes of facets on each leaf). The main parameters that affect the segmentation granularity are K and r . According to suggestions in [51], the other two parameters in the facet oversegmentation are fixed all over the experiments in this article as $\theta = 23^\circ$ and $\sigma_1 = 0.025$ m. Parameter K represents the number of nearest neighbor points when calculating the spatial characteristics of each point. This value not only affects normals of the facets, but also affects the number of facets generated. The higher the K value is, the smaller the number of facets is. The parameter r controls the minimum distance between the centers of adjacent facets, so the larger the value is, the smaller the number of facets is. In order to avoid conflicting situation in parameters' contributions, we generally increase K while increasing r . The minimum value of r should not be less than five times of the average spacing to gather enough points for each facet. For *Maranta arundinacea*, we set r in the interval of $[0.01, 0.18]$ m, and for *Dieffenbachia picta*, we set r in the interval of $[0.005, 0.14]$ m. The parameter K has less influence on the oversegmentation than r , and we set its value in the interval $[100, 200]$. We adjust the values of r and K to form 12 different parameter configurations for each plant to find the optimal parameters for generating facets and calculating leaf areas. First, we verify that the changes in parameters can produce different granularity. Second, we will show that the LAA is actually robust to changes in granularity. Fig. 16 shows different parameter settings can produce different facet oversegmentation results on a *Maranta arundinacea* leaf.

We conducted facet oversegmentation on leaves of *Maranta arundinacea* and *Dieffenbachia picta* with 12 groups of parameters, respectively. The numbers of facets under all parameter configurations are shown in Fig. 17(a1) and (b1). With the increase of the two parameters, the number of facets first decreases and then remains stable. This is because when r is sufficiently large, the other two auxiliary parameters (θ and σ_1 in [51], [58]) take over the role of dividing points into facets. The LAA results under 12 different parameter configurations for leaves of each plant are shown in Fig. 17(a2) and (b2), respectively. For *Maranta arundinacea*, the LAAs of most leaves stabilize above 96% after $r \geq 0.027$, and the number of facets for each

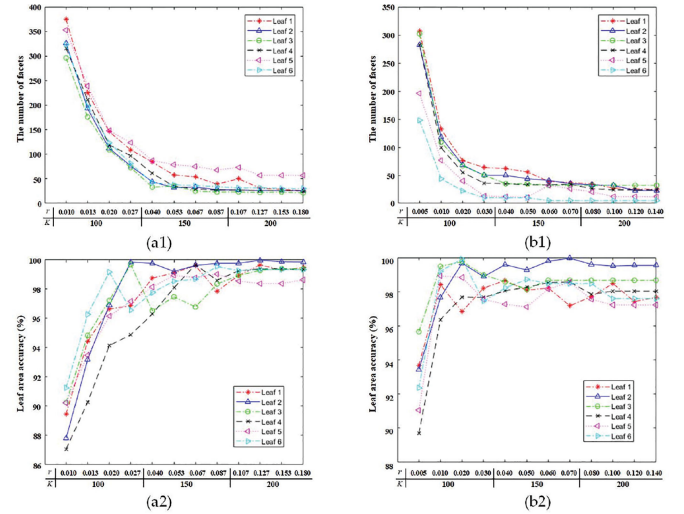


Fig. 17. Parameter tuning for facet oversegmentation parameters. (a1) and (b1) show the influence of parameters on the number of generated facets for leaves of *Maranta arundinacea* and *Dieffenbachia picta*, respectively. (a2) and (b2) show the relationship of parameter configurations and the LAAs. A total of 12 parameter configurations are tested on each type of plant, and the majority of them can produce a calculation accuracy higher than 96%.

leaf is below 150. For *Dieffenbachia picta*, the LAAs of most leaves stabilize above 96% after $r \geq 0.02$, and the number of facets for each leaf is below 100. It can be observed that when the number of facets is too large (more than 250), the accuracy of leaf area calculation will decrease. Therefore, the number of facets should be upper-bounded to suppress calculation error. It should be noted that our calculation method is actually robust to change of parameters for facets generation because the worst LAA of leaves in Fig. 17 still stay around 90%.

We finally set $r = 0.067$ m, and $K = 150$ for *Maranta arundinacea*; $r = 0.060$ m, and $K = 150$ for *Dieffenbachia picta*. In fact, the two parameters can be easily chosen from broad intervals as long as the number of facets on each leaf is kept below 100.

In the step of outlier triangle removal in leaf area calculation, we remove the spurious triangles around facet edges by using the median value of all lengths of triangles s_{median} multiplied by the parameter λ as a threshold. The larger the threshold is, the fewer triangles are removed. Therefore, λ has influence on the result of leaf area estimation. We first fixed the parameters r and K to the suggested values, and then tune the parameter λ in the interval $[2, 10]$ to obtain the LAA results of the two plants, respectively. The results of six *Maranta arundinacea* leaves are shown in Fig. 18(a), in which all reach the highest accuracy for leaf area calculation when $\lambda = 3.0$. Fig. 18(b) shows the results of *Dieffenbachia picta*, in which all six leaves also reach the highest accuracy when $\lambda = 3.0$. Therefore, it is concluded that for those plants have similar leaf structures with the two plant types, λ can be fixed at 3.0 to obtain good performances.

E. Results of Leaf Length, Width, and Inclination Angle

Fig. 19 shows the results of using the method in Section II-G to calculate the 3-D bounding boxes and the average leaf normals for all leaves of the two types. The first row of Fig. 19 shows

TABLE II
ESTIMATION ERRORS OF THE PROPOSED METHOD OF CALCULATING LEAF LENGTH AND WIDTH

Plant Types	Category	Leaf 1	Leaf 2	Leaf 3	Leaf 4	Leaf 5	Leaf 6	Leaf 7	Leaf 8	Leaf 9	Leaf 10	Leaf 11	Leaf 12	Average
<i>Maranta arundinacea</i> Sample 1	Length	5.5	2.4	4.4	3.4	3.8	4.5	4.5	2.7	4.1	5.5	2.5	3.2	3.9
	Width	2.8	5.2	1.3	7.4	2.2	2.0	7.1	4.7	5.3	4.1	2.2	3.2	3.9
<i>Maranta arundinacea</i> Sample 2	Length	2.0	2.6	4.3	1.4	4.5	2.6	3.4	4.9	2.8	1.1	3.4	/	3.0
	Width	3.6	4.6	0.7	2.1	0.9	5.8	2.4	6.5	2.8	5.4	4.5	/	3.6
<i>Dieffenbachia picta</i> Sample 1	Length	5.9	4.0	2.8	4.5	4.2	6.5	5.2	4.0	/	/	/	/	4.6
	Width	1.2	3.6	9.5	5.8	3.9	7.4	3.4	2.6	/	/	/	/	4.7
<i>Dieffenbachia picta</i> Sample 2	Length	5.5	6.3	2.2	3.4	4.4	1.8	4.7	1.2	/	/	/	/	3.7
	Width	2.1	6.0	5.7	4.5	7.4	4.7	4.9	2.0	/	/	/	/	4.7

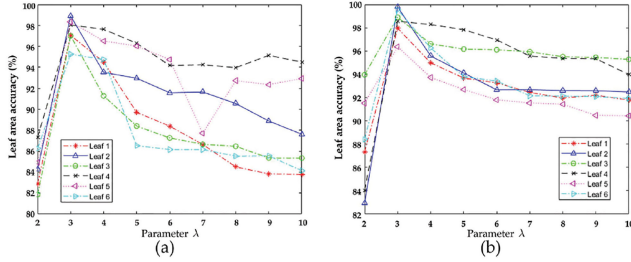


Fig. 18. Relationship between the parameter λ in outlier triangle removal step and LAA. (a) and (b) are the results of parameter tuning on *Maranta arundinacea* and *Dieffenbachia picta*, respectively. The x-axis is λ , and the y-axis is LAA. The best performance for both plants are obtained when $\lambda = 3.0$.

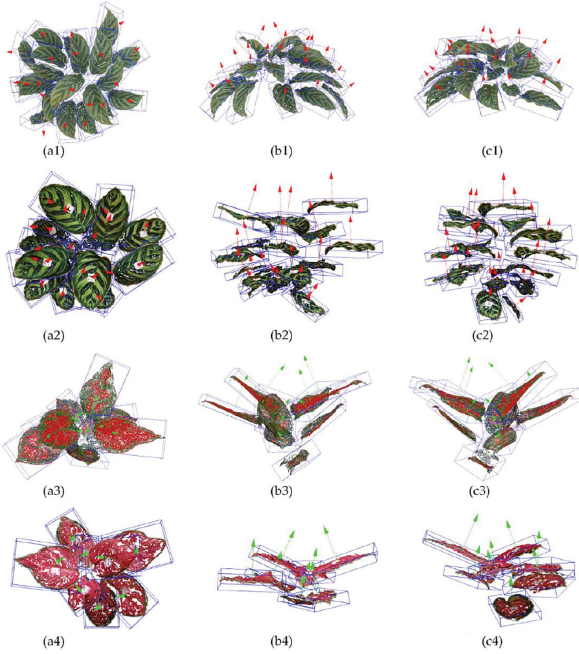


Fig. 19. Results with 3-D bounding boxes and leaf normals. The leaf normals are represented by arrows, and the 3-D bounding boxes are painted in blue. From the first row to the fourth row, results are *Maranta arundinacea* sample 1, *Maranta arundinacea* sample 2, *Dieffenbachia picta* sample 1, and *Dieffenbachia picta* sample 2. (a1) Top view. (a2) Front view. (a3) Side view. (a4) Top view. (b1) Front view. (b2) Side view. (b3) Top view. (b4) Front view. (c1) Side view. (c2) Top view. (c3) Front view. (c4) Side view.

the canopy point cloud of *Maranta arundinacea* sample 1 with 3-D bounding boxes and normals from three different views, respectively. The second row shows the results of *Maranta arundinacea* sample 2. The third and fourth rows show the

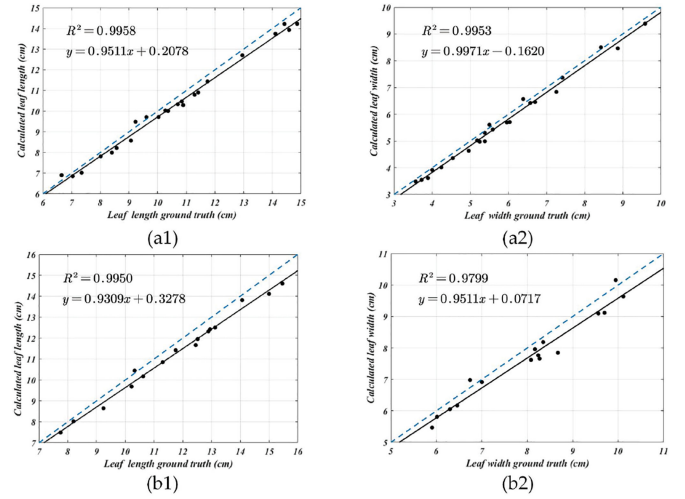


Fig. 20. Regressions of estimated leaf length and leaf width with ground truths for the two plants. (a1) and (a2) are the regression results of the leaf length and width of *Maranta arundinacea*, respectively. (b1) and (b2) are the regression results of the leaf length and width of *Dieffenbachia picta*, respectively. The results show the estimated values are close to their ground truths.

results of *Dieffenbachia picta* sample 1 and 2, respectively. Each leaf normal starts from the gravity center of that leaf.

Table II lists the estimation errors of the leaf length and width using our 3-D bounding box for the two types of plants. The error is defined as the ratio of the absolute difference between the estimated value and the ground truth to the ground truth in percentage. The ground truths of leaf length and width are obtained by the method shown in Fig. 14. The average errors of the estimated leaf length and leaf width are both below 4.7%. Fig. 20 shows the linear correlations between our calculated values and the ground truths on leaf length, width of the two plant types, respectively. All four fitted lines are close to $y = x$, showing high accuracies. Note that the calculated length and width values are generally smaller than their ground truths. This is because leaves are flattened when ground truth values are measured, making every ground truth leaf larger than its appearance in the real 3-D world.

We also compare the inclination angles calculated by the proposed approach with the ground truths measured by an inclinometer for each leaf in Table III. For *Maranta arundinacea* sample 1 and sample 2, the maximum leaf inclination angle error (difference) is no higher than 4.0° and 4.8° , respectively. For the leaves from *Dieffenbachia picta* sample 1 and sample 2, the

TABLE III
COMPARISON OF ESTIMATED VALUES WITH GROUND TRUTHS OVER THE LEAF INCLINATION ANGLE

Plant types	Inclination angle	Leaf 1	Leaf 2	Leaf 3	Leaf 4	Leaf 5	Leaf 6	Leaf 7	Leaf 8	Leaf 9	Leaf 10	Leaf 11	Leaf 12
<i>Maranta arundinacea</i> Sample 1	Measured	3.5	36.5	25.0	8.5	11.5	13.0	10.5	18.0	27.5	31.0	22.5	33.5
	GT	6.1	38.5	22.6	11.5	15.5	9.9	13.6	15.9	29.1	33.2	25.6	30.2
	Difference	2.6	2.0	2.4	3.0	4.0	3.2	3.1	2.1	1.6	2.2	3.1	3.3
<i>Maranta arundinacea</i> Sample 2	Measured	12.5	10.5	7.5	11.5	28.5	22.0	16.5	26.5	19.5	41.0	21.5	/
	GT	16.4	12.6	9.6	13.6	32.8	24.9	17.2	31.3	23.7	46.2	20.1	/
	Difference	3.9	2.1	2.1	2.1	4.3	2.9	0.7	4.8	4.2	5.2	1.4	/
<i>Dieffenbachia picta</i> Sample 1	Measured	45.0	9.0	35.5	36.5	51.5	9.0	31.5	40.0	/	/	/	/
	GT	47.5	12.9	39.3	33.2	49.0	10.5	29.2	36.5	/	/	/	/
	Difference	2.5	3.9	3.8	3.3	2.5	1.5	2.3	3.5	/	/	/	/
<i>Dieffenbachia picta</i> Sample 2	Measured	29.5	18.0	17.0	19.0	17.0	8.0	14.0	15.0	/	/	/	/
	GT	34.1	13.1	16.4	18.3	20.8	8.9	9.9	10.0	/	/	/	/
	Difference	4.6	4.9	0.6	0.7	3.8	0.9	4.1	5.0	/	/	/	/

TABLE IV
AVERAGE PROCESSING SPEEDS OF THE PROPOSED METHOD ON ALL STAGES FOR THE TWO TYPES OF PLANTS

Plant types	3D reconstruction (min)	Preprocessing (s)	Stems removal (s)	Leaf segmentation (s)	Leaf area calculation (s per leaf)	Leaf width, height, inclination calculation (s per leaf)
<i>Maranta arundinacea</i>	23.283	5.415	31.476	18.025	11.426	0.018
<i>Dieffenbachia picta</i>	13.716	1.870	12.333	2.225	5.264	0.024

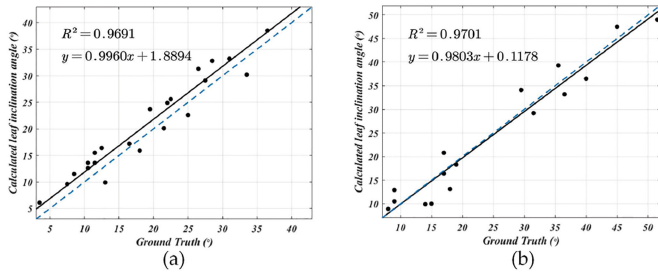


Fig. 21. Regressions of the estimated leaf inclination angle with ground truths for the two plants. The two fitted lines are close to the line $y = x$, showing the high accuracy of the proposed method.

maximum leaf inclination angle error (difference) is no higher than 3.9° and 5.0° , respectively. The average errors of calculated leaf inclination angle for the two plant types are 2.9° and 3.0° , respectively. The average measurement error of ground truth by the inclinometer is about 3.0° , which means our approach is comparable with the inclinometer measurement. The correlations between calculated inclination angles and the measured ground truths are given in Fig. 21, showing the effectiveness of our approach to extract inclination angles.

F. Processing Speed

The processing times for different stages of the proposed method are listed in Table IV. The processing time for plant *Maranta arundinacea* is longer than *Dieffenbachia picta* at all stages, this is because the former contains more points than the latter. The biggest bottleneck of the processing speed lies in the 3-D reconstruction stage. Except from the 3-D reconstruction stage by MVS, both of our leaf segmentation and the leaf phenotypic traits calculation have quasireal-time processing speeds.

IV. CONCLUSION

This article proposes a framework for automatic leaf segmentation and leaf phenotypic feature extraction for MVS plant point clouds. Experiments on two types of ornamental plants have shown that the proposed framework can effectively calculate phenotypic features such as the area, length, width, average normal, and the inclination angle of each single leaf in the canopy point clouds. The average accuracy of the calculated leaf area by our method reaches 96.8% for *Maranta arundinacea* and 97.8% for *Dieffenbachia picta*. The average errors of the calculated leaf length and width for *Maranta arundinacea* are both less than 4.0%, and the average error of the estimated leaf inclination angle is 2.9° . For *Dieffenbachia picta*, the average errors of calculated leaf length and width are both no higher than 4.7%, and the average error of the estimated leaf inclination angle is 3.0° .

At present, our leaf segmentation and phenotyping framework have been tested on several types of ornamentals. Although the tested samples differ greatly on aspects such as canopy structure and leaf shape, the effectiveness of our method is yet to be further validated on a broad range of plants. For example, the maize plant has very long and curved leaves, the bounding box of a single leaf can be huge and its boundaries are hence no longer reliable enough to deduce the dimension of its leaf inside. Therefore, the techniques for calculating the leaf length, width, and inclination angle for maize may have to be modified. In the future, we plan to apply the proposed framework on several more kinds of agricultural plants, and keep on improving our method.

REFERENCES

- [1] A. Walter, F. Liebisch, and A. Hund, "Plant phenotyping: From bean weighing to image analysis," *Plant Methods*, vol. 11, no. 1, 2015, Art. no. 14.
- [2] L. Li, Q. Zhang, and D. Huang, "A review of imaging techniques for plant phenotyping," *Sensors*, vol. 14, no. 11, pp. 20078–20111, 2014.

- [3] C. Costa, U. Schurr, F. Loreto, P. Menesatti, and S. Carpentier, "Plant phenotyping research trends, a science mapping approach," *Front. Plant Sci.*, vol. 9, 2018, Art. no. 1933.
- [4] T. W. Gara, A. K. Skidmore, R. Darvishzadeh, and T. Wang, "Leaf to canopy upscaling approach affects the estimation of canopy traits," *GLSci. Remote Sens.*, vol. 56, no. 4, pp. 554–575, 2019.
- [5] N. Fahlgren, M. A. Gehan, and I. Baxter, "Lights, camera, action: High-throughput plant phenotyping is ready for a close-up," *Current Opinion Plant Biol.*, vol. 24, pp. 93–99, 2015.
- [6] A. Hartmann, T. Czauderna, R. Hoffmann, N. Stein, and F. Schreiber, "HTPheno: An image analysis pipeline for high-throughput plant phenotyping," *BMC Bioinf.*, vol. 12, 2011, Art. no. 148.
- [7] F. Fiorani and U. Schurr, "Future scenarios for plant phenotyping," *Annu. Rev. Plant Biol.*, vol. 64, no. 1, pp. 267–291, 2013.
- [8] T. T. Santos and A. A. de Oliveira, "Image-based 3D digitizing for plant architecture analysis and phenotyping," *Workshop on Industry Applications (WGARI) in SIBGRAPI 2012 (XXV Conf. Graphics, Patterns Images)*, Ouro Preto, Brazil, Aug. 2012, doi: [10.13140/2.1.4576.1608](https://doi.org/10.13140/2.1.4576.1608).
- [9] A. Lang, "Leaf orientation of a cotton plant," *Agricultural Meteorol.*, vol. 11, no. 73, pp. 37–51, 1973.
- [10] H. Sinoquet, B. Mouliat, and R. Bonhomme, "Estimating the 3D geometry of a maize crop as an input of radiation models: Comparison between 3D digitizing and plant profiles," *Agricultural Forest Meteorol.*, vol. 55, no. 3–4, pp. 233–249, 1991.
- [11] C. Godin, E. Costes, and H. Sinoquet, "A method for describing plant architecture which integrates topology and geometry," *Ann. Botany*, vol. 84, no. 3, pp. 343–357, 1999.
- [12] M. Rakocevic, H. Sinoquet, A. Christophe, and C. Varlet-Grancher, "Assessing the geometric structure of a white clover (*Trifolium repens* L.) canopy using 3-D digitising," *Ann. Botany*, vol. 86, no. 3, pp. 519–526, 2000.
- [13] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Trans. Graph.*, vol. 29, no. 6, 2010, Art. no. 151.
- [14] Z. S. Koma, M. Rutzinger, and M. Bremer, "Automated segmentation of leaves from deciduous trees in terrestrial laser scanning point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1456–1460, Sep. 2018.
- [15] S. Jin *et al.*, "Separating the structural components of maize for field phenotyping using terrestrial lidar data and deep convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2644–2658, Apr. 2020.
- [16] W. Su, M. Zhang, J. Liu, and Z. Sun, "Automated extraction of corn leaf points from unorganized terrestrial lidar point clouds," *Int. J. Agricultural Biol. Eng.*, vol. 11, no. 3, pp. 166–170, 2018.
- [17] S. Sun *et al.*, "In-field high throughput phenotyping and cotton plant growth analysis using lidar," *Front. Plant Sci.*, vol. 9, 2018, Art. no. 16.
- [18] J. A. Jimenez-Berni *et al.*, "High throughput determination of plant height, ground cover, and above-ground biomass in wheat with lidar," *Front. Plant Sci.*, vol. 9, 2018, Art. no. 237.
- [19] Q. Guo *et al.*, "Crop 3D—A lidar based platform for 3D high-throughput crop phenotyping," *Sci. China Life Sci.*, vol. 61, no. 3, pp. 328–339, 2018.
- [20] H. Yuan, R. S. Bennett, N. Wang, and K. D. Chamberlin, "Development of a peanut canopy measurement system using a ground-based lidar sensor," *Front. Plant Sci.*, vol. 10, 2019, Art. no. 203.
- [21] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik, "Evaluating and improving the depth accuracy of kinect for windows v2," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4275–4285, Aug. 2015.
- [22] N. Thuy, S. David, M. Nelson, M. Julin, and S. Neelima, "Structured light-based 3D reconstruction system for plants," *Sensors*, vol. 15, no. 8, pp. 18 587–18 612, 2015.
- [23] M. Vazquez-Arellano, D. Reiser, D. S. Paraforos, M. Garrido-Izard, M. E. C. Burce, and H. W. Griepentrog, "3-D reconstruction of maize plants using a time-of-flight camera," *Comput. Electron. Agriculture*, vol. 145, pp. 235–247, 2018.
- [24] A. Vit and G. Shani, "Comparing RGB-D sensors for close range outdoor agricultural phenotyping," *Sensors*, vol. 18, no. 12, 2018, Art. no. 4413.
- [25] X. Wang, S. Daljit, M. Sandeep, M. Geoffrey, and P. Jesse, "Field-based high-throughput phenotyping of plant height in sorghum using different sensing technologies," *Plant Methods*, vol. 14, no. 1, 2018, Art. no. 53.
- [26] R. Xiang, H. Jiang, and Y. Ying, "Recognition of clustered tomatoes based on binocular stereo vision," *Comput. Electron. Agriculture*, vol. 106, pp. 75–90, 2014.
- [27] C. Wang, X. Zou, Y. Tang, L. Luo, and W. Feng, "Localisation of litchi in an unstructured environment using binocular stereo vision," *Biosyst. Eng.*, vol. 145, pp. 39–51, 2016.
- [28] X. Xiong, W. Yang, D. Wu, and Q. Liu, "A high-throughput imaging facility for evaluation of oilseed rape biomass related traits," in *Proc. Asia Commun. Photon. Conf.*, vol. 13, 2016, pp. 1–3.
- [29] D. Li, L. Xu, X. Song Tang, S. Sun, X. Cai, and P. Zhang, "3D imaging of greenhouse plants with an inexpensive binocular stereo vision system," *Remote Sens.*, vol. 9, no. 5, 2017, Art. no. 508.
- [30] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 17–22, 2006, pp. 519–528.
- [31] M. Klodt and D. Cremers, "High-resolution plant shape measurements from multi-view stereo reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8928, 2017, pp. 174–184.
- [32] M. P. Pound, A. P. French, E. H. Murchie, and T. P. Pridmore, "Surface reconstruction of plant shoots from multiple views," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8928, 2014, pp. 158–173.
- [33] J. C. Rose, S. Paulus, and H. Kuhlmann, "Accuracy analysis of a multiview stereo approach for phenotyping of tomato plants at the organ level," *Sensors*, vol. 15, no. 5, pp. 9651–9665, 2015.
- [34] Y. Wang *et al.*, "Maize plant phenotyping: Comparing 3D laser scanning, multi-view stereo reconstruction, and 3D digitizing estimates," *Remote Sens.*, vol. 11, no. 1, 2019, Art. no. 63.
- [35] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp, "A novel mesh processing based technique for 3D plant analysis," *BMC Plant Biol.*, vol. 12, no. 1, 2012, Art. no. 63.
- [36] M. Muller-Linow, F. Pinto-Espinosa, H. Scharr, and U. Rascher, "The leaf angle distribution of natural plant populations: Assessing the canopy with a novel software tool," *J. Plant Methods*, vol. 11, no. 1, 2015, Art. no. 11.
- [37] T. Duan, S. Chapman, E. Holland, G. Rebetzke, Y. Guo, and B. Zheng, "Dynamic quantification of canopy structure to characterize early plant vigour in wheat genotypes," *J. Exp. Botany*, vol. 67, no. 15, pp. 4523–4534, 2016.
- [38] K. Itakura and F. Hosoi, "Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3D plant images," *Sensors*, vol. 18, no. 10, 2018, Art. no. E3576.
- [39] S. Jin *et al.*, "Stem-leaf segmentation and phenotypic trait extraction of individual maize using terrestrial lidar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1336–1346, Mar. 2019.
- [40] W. Su, D. Zhu, J. Huang, and H. Guo, "Estimation of the vertical leaf area profile of corn (*Zea mays*) plants using terrestrial laser scanning (TLS)," *Comput. Electron. Agriculture*, vol. 150, pp. 5–13, 2018.
- [41] L. Shihua, D. Leiyu, W. Hongshu, W. Yong, H. Ze, and L. Sen, "Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial lidar data and a voxel-based model," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1202.
- [42] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Vis. Transmiss.*, Zurich, Switzerland, 2012, pp. 501–508.
- [43] D. Zermas, V. Morellas, D. Mulla, and N. Papanikolopoulos, "Estimating the leaf area index of crops through the evaluation of 3d models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6155–6162.
- [44] S. T. Kohonen Termini, "Self-organizing maps," *Circolo Matematico di Palermo*, vol. 44, no. 3, pp. 506–506, 1995. [Online]. Available: <https://doi.org/10.1007/BF02844683T>
- [45] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1–4.
- [46] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Int. Conf. 3D Vis.*, 2013, pp. 127–134.
- [47] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 20–25, 2011, pp. 3057–3064.
- [48] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [49] B. Triggs, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms: Theory Pract.*, 1999, pp. 298–372.
- [50] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.
- [51] D. Li, Y. Cao, X. Song Tang, S. Yan, and X. Cai, "Leaf segmentation on dense plant point clouds with facet region growing," *Sensors*, vol. 18, no. 11, 2018, Art. no. 3625.

- [52] I. T. Jolliffe, "Principal component analysis," *J. Marketing Res.*, vol. 25, no. 4, 2002, Art. no. 513.
- [53] L. P. Chew, "Constrained Delaunay triangulation," *Algorithmica*, vol. 4, no. 1–4, pp. 97–108, 1989.
- [54] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.
- [55] T. Liu, A. W. Moore, A. Gray, and K. Yang, "An investigation of practical approximate nearest neighbor algorithms," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 825–832.
- [56] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [57] R. B. Rusu, *Semantic 3D Object Maps for Everyday Robot Manipulation*. Berlin, Germany: Springer, 2013, doi: [10.1007/978-3-642-35479-3](https://doi.org/10.1007/978-3-642-35479-3).
- [58] D. Li, Y. Cao, G. Shi, X. Cai, and S. Yan, "An overlapping-free leaf segmentation method for plant point clouds," *IEEE Access*, vol. 7, pp. 129054–129070, 2019.
- [59] Z. C. Marton, R. B. Rusu, and M. Beetz, "On fast surface reconstruction methods for large and noisy point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, Kobe, Japan, May 12–17, 2009, pp. 3218–3223.
- [60] W. Yuan, D. Hu, "Measurement of leaf blade length and width based on moment," *Comput. Eng. Appl.*, vol. 49, no. 16, pp. 188–191. doi: [10.3778/j.issn.1002-8331.1111-0488](https://doi.org/10.3778/j.issn.1002-8331.1111-0488).
- [61] O. Nobuyuki, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

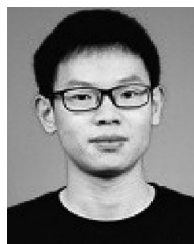


Dawei Li (Member, IEEE) received the B.Eng. degree in automation from Tongji University, Shanghai, China, in 2006, and the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2013.

During 2013–2015, he was a Postdoc with the Department of Computer Sciences and Technology, Tongji University, Shanghai, China. He is currently working as an Associate Professor with the College of Information Sciences and Technology, Donghua University, Shanghai, China. From 2009 to 2010, he

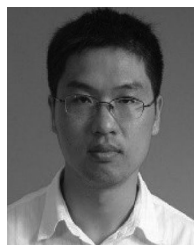
was a Visiting Researcher with the Genetic Algorithms Research and Applications Group, Michigan State University, East Lansing, MI, USA. His research interests include image processing, pattern recognition, plant phenotyping, and agricultural engineering.

Dr. Li was bestowed the Finalist for Best Paper Award at the 11th IEEE's International Conference on Control, Automation, Robotics, and Vision, in 2010.



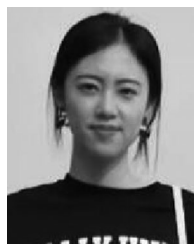
Guoliang Shi received the B.S. degree in automation from Changsha University of Science and Technology, Changsha, China, in 2018. He is currently working toward the M.S. degree in control theory and control engineering at Donghua University, Shanghai, China.

His research interests include 3-D point cloud processing, deep learning, and plant phenotyping.



Weijian Kong received the M.S. degree in pattern analysis and intelligent system and the Ph.D. degree in control theory and control engineering, both from Northeastern University, Shenyang, China, in 2008 and 2014, respectively.

He is currently a Lecturer with the College of Information Sciences and Technology, Donghua University, Shanghai, China. His research interest includes machine learning and plant-wide optimization for the complex industrial systems.



Sifan Wang received the B.S. degree in automation from Donghua University, Shanghai, China, in 2018. She is currently working toward the M.S. degree in control theory and control engineering at Donghua University, Shanghai, China.

Her research interests include image processing and artificial neural networks.



Yang Chen received the B.S. degree in electronic information science and technology from North China University of Science and Technology, Tangshan, China, in 2016. He is currently working toward the M.S. degree in control theory and control engineering at Donghua University, Shanghai, China.

His research interests include machine learning and image processing.