# 3D Reconstruction of Plant Leaves for High-Throughput Phenotyping

Feiyu Zhu[1], Suresh Thapa[2], Tian Gao[1], Yufeng Ge[2], Harkamal Walia[3], Hongfeng Yu[1]

[1] Department of Computer Science and Engineering
[2] Department of Biological Systems Engineering
[3] Department of Agronomy and Horticulture
University of Nebraska-Lincoln, Lincoln, NE, USA

*Abstract*—Generating 3D digital representations of plants is indispensable for researchers to gain a detailed understanding of plant dynamics. Emerging high-throughput plant phenotyping techniques can capture plant point clouds that, however, often contain imperfections and make it a changeling task to generate accurate 3D reconstructions. We present an end-to-end pipeline to reconstruct surfaces from point clouds of maize and rice plants. In particular, we propose a two-step clustering approach to accurately segment the points of each individual plant component according to maize and rice properties. We further employ surface fitting and edge fitting to ensure the smoothness of resulting surfaces. Realistic visualization results are obtained through post-processing, including texturing and lighting. Our experimental study has explored the parameter space and demonstrated the effectiveness of our pipeline for high-throughput plant phenotyping.

*Index Terms*—3D reconstruction, high-throughput plant phenotyping, point cloud.

## I. INTRODUCTION

Capturing the dynamics of plant growth is a necessity for plant biologists to gain new insights into complex relationships between genes and phenotypes of plant organization under different environmental stresses [4]. During the process from a small seed to a full-grown plant with many leaves, the 3D structure of a plant exhibits dramatic changes. High-throughput plant phenotyping becomes indispensable to quantify the dynamics of high spatial or temporal resolutions with accurate 3D representations [9].

High-throughput phenotyping generally mainly employs two types of approaches to obtain 3D information of objects. The first type are active approaches that use active sensors, e.g., LiDAR (Light Detection and Ranging) [25], [28], to directly capture 3D structural information by generating discretized 3D point clouds [13]. The second type are passive approaches that use passive sensors, e.g., regular cameras, to take 2D images from multiple views of an object and generate 3D point clouds from these 2D images [24]. 3D reconstruction, converting a point cloud to a geometry that is represented by triangles [23], [28], is then typically applied to facilitate researchers to gain a better understanding of the 3D structures of target objects.

Compared to traditional application domains (e.g., 3D modeling in movies and games), high-throughput plant phenotyping has imposed new challenges in 3D reconstruction:

First, due to the subtle and complex structure of a plant and the resolution limit of imaging devices, a resulting 3D point cloud often contains various imperfections, such as severe noises and missing points [17].

Second, a plant researcher usually desires to obtain accurate 3D digital descriptions that can capture subtle features (e.g., most dynamically changed areas on a leaf) during a plant growth process, and such high accurate descriptions typically are not demanded in most movie or game scenes where visually appealing 3D approximations could suffice.

Third, most existing high-throughput plant platforms need to move either plants or imaging devices to capture the entire plant structure from different views [12], [16], and thereby are hard to guarantee stabilized imaging conditions during motions. In particular, unlike rigid objects, plant leaves can be easily vibrated to incur blurred or noised imaging results. This issue can be exacerbated with an increased movement speed in order to get a higher throughput.

In this paper, we present a pipeline to reconstruct the 3D geometries of plants from their 3D point clouds. We holistically address the key steps from noise reduction to plant structure reconstruction. In particular, we develop a 3D reconstruction method for plant leaves by leveraging certain natural properties of leaves to address the imperfections in their 3D point clouds. Post-processing, such as texturing and lighting, has been applied to resulting 3D geometries to enhance reconstruction results. We target specific plants (i.e., maize and rice) in this work, and our developed techniques can be potentially applied to other plants. Our approach can greatly facilitate researchers in studying 3D plant structures and capturing detailed features from high-throughput plant phenotyping.

## II. RELATED WORK

Extensive research work has been conducted for 3D reconstruction from point clouds. Berger et al. presented a comprehensive survey of the techniques [2]. In the plant science community, these techniques have been exploited to quantify the structural properties of different plants and their components [1], [3], [29]. In the computer vision and graphics communities, several approaches have been proposed for plant reconstruction and modeling. For example, Ijiri et al. [14] used simple primitives to approximate flower components via X-ray
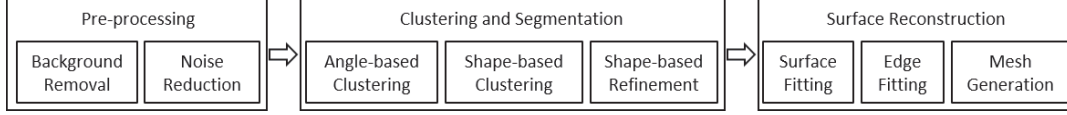
Fig. 1. The main steps of our pipeline.

Computed Tomography. Livny et al. [19] used a lobe-based representation to approximate the geometry of tree data.

As a plant can contain many parts (e.g., leaves and stalks) with distinct geometry properties, the whole point cloud of a plant usually cannot be reconstructed into a single geometry. Instead, the points that belong to each component should be segmented first, and then are tackled individually by 3D reconstruction algorithms. Therefore, the classification of points is a critical step in reconstructing plant or plant-like structures. Quan et al. [23] treated a point cloud of a plant as a graph and used a graph cut algorithm to cluster the leaves. Liu et al. [18] used a Gauss map to classify the surface types from a point cloud of a facility with pipelines. They showed that the surfaces can be classified by identifying the rings or patches in a Gauss map when the types of surfaces are limited in a point cloud. However, for plants, the surface of an actual leaf is not flat. For example, a maize leaf is wavy along the edges. Thus, if the normals of a plant point cloud are projected onto a unit spherical surface, the points on the surface will be scattered, which makes the Gauss map less effective for clustering points. Li et al. [17] determined the label of a point to a certain group by designing and minimizing an energy function. Their approach can extract and track the topological events like budding and bifurcation during the growth of certain plant species (e.g., Dishlia) from time-varying 3D point clouds. Schnabel et al. [26] used the RANSAC algorithm to extract shapes by randomly selecting minimal points from data and constructing corresponding models. However, the RANSAC algorithm may take many iterations and incur high computation costs, as it is based on a random search. In addition, it can produce false models if the point cloud has a lot of noises.

After different components have been identified from the 3D point cloud of a plant, the structure of each component can be reconstructed. Surface fitting is used to find the triangular mesh that can represent the point cloud and the corresponding object. Several methods are commonly used. Marching cubes [20] has been used for surface reconstruction in which a tangent plane at each sample is estimated using the k-nearest neighbors, and then the distance to the plane is used to compute a signed distance function. Poisson surface reconstruction is a well-known technique for creating watertight surfaces from oriented point clouds with normals [15]. Moving Least Squares (MLS) [11], [21] has been used to find a smooth surface, a polynomial approximation of the local neighborhood, from a local planar parametrization. B-spline surface reconstruction, which describes the surface with B-spline functions, is also available for point clouds [7].

However, it is challenging to directly apply these methods to plant data to generate appropriate reconstruction results. The point cloud of a plant can contain a relatively high amount of noise, which can incur significant interference for 3D reconstruction processes. In particular, leaves of certain plants (e.g., maize and rice) can be characterized by their long and narrow structures, incurring a less accurate estimation of the surface in a small local region of a leaf.

## III. Our Approach

We develop an end-to-end pipeline to reconstruct the 3D geometry of a plant from its 3D point cloud. The pipeline consists of three main steps: data acquisition and preprocessing, leaf clustering and segmentation, and 3D model reconstruction. Figure 1 summarizes the main steps.

- First, we collect the point cloud of a plant by either active approaches or passive approaches. We apply the preprocessing operations to remove the background points and reduce the noises in the original point cloud.
- Next, we segment the 3D data points into individual leaves through a two-step clustering approach. Different clusters belong to the same leaf are detected and merged.
- Finally, we reconstruct the surface of each leaf using surface and curve fitting techniques and combine all the models of leaves together. We apply certain postprocessing (e.g., adding the textures of leaves and the models of stalks and pots) and enhance the final model of the plant to resemble the appearance of a real plant.

### A. Pre-processing

*1) Point Could Generation:* In this work, we target maize and rice plants. We place the plant on a rotary table and the table can be rotated 360 degrees. We use two methods to generate the point clouds of plants.

In the first method, we use a line laser scanner to vertically scan a plant during its rotation to directly obtain the 3D point cloud of the plant [28]. The resolution of the points is approximately 5 millimeters. We aggregate the depth information collected to get the 3D point cloud.

In the second method, we use the 2D images taken by the LemnaTec high-throughput plant phenotyping system located at the University of Nebraska-Lincoln's Greenhouse Innovation Center. Specifically, we use the images of plants taken by one camera from a side view and at every 72 degrees during the rotation. We employ the structure-from-motion (SFM) technique [22], [27] to reconstruct the 3D point cloud from these 2D images.
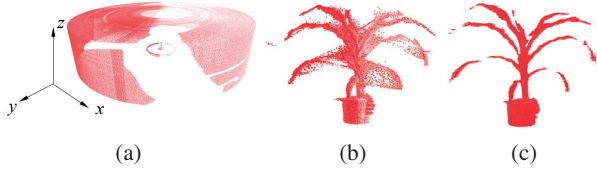
Fig. 2. Pre-processing a point cloud of a maize plant: (a) the original point cloud containing a plant (in the middle) and its background, (b) the point cloud after background removal, and (c) the point cloud after noise reduction.

*2) Background Removal:* A point cloud often contains a part of the environment (e.g., the ceiling or walls of a room), which is called the background. The background is usually very distinctive from the points of a plant in terms of their depth values, and thus can be relatively easily filtered out according to a depth range of the point cloud around its center. Figure 2(a) shows an example of a point cloud containing both a plant and the ambient background, and Figure 2(b) shows the result after background removal.

*3) Noise Reduction:* Because of the resolution limit of the devices (i.e., laser scanners and cameras) and the fine structure of a plant, the resulting point cloud can often contain a relatively high level of noise. In particular, there are significant noisy outliers near the edge of each leaf. These points will impair the quality of reconstruction. For example, clustering methods may not be directly applied to a point cloud with many outliers to generate optimal results.

We notice that the density of noisy points is typically lower than the density of plant points. We compute the density of a point $p_i$, $D(p_i)$, in a point cloud $P$ by finding the number of points within a spherical kernel $k$ centered at $p_i$:

$$D(p_i) = \frac{1}{\gamma^2} \sum_{p_j \in P} \delta(\|p_i, p_j\|) \qquad (1)$$

where $\gamma$ is the radius of $k$, $\|\cdot\|$ denotes the Euclidean distance between two points, and $\delta$ is a distance weight metric defined as:

$$\delta(\|p_i, p_j\|) = \begin{cases} 1, & if \|p_i, p_j\| \leq \gamma \\ 0, & if \|p_i, p_j\| > \gamma \end{cases} \qquad (2)$$

A point whose density is lower than a threshold $d$ is considered as outliers and thus is removed from the point cloud. Given a point $p_i$ and the point cloud center $c$, we set the threshold $d$ to be inversely proportional to the distance between $p_i$ and $c$:

$$d = k/(1 + \|p_i, c\|) \qquad (3)$$

where $k$ is a predefined coefficient. This is because the density and the noise level are higher for regions closer to the center.

After noise reduction, the plant structure becomes more clear in the point cloud, as shown in Figure 2(c). However, this may also incur a new problem where the points belonging to the same leaf (particularly with a thin structure) can become discontinuous, as some points at the intermediate positions along the leaf can be filtered out as noisy outliers.
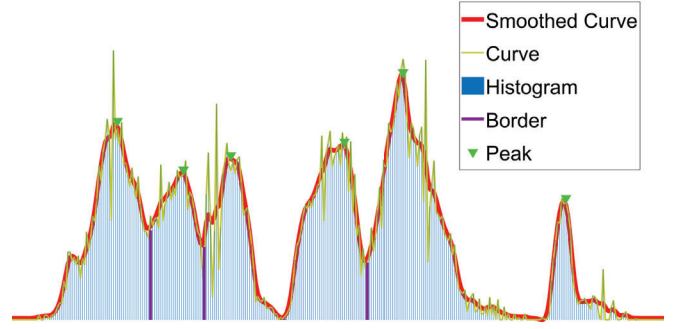


Fig. 3. A histogram of $\theta$ values.

### B. Clustering and Segmentation

In order to correctly segment the points of a leaf into one group, we employ a two-step clustering approach based on the leaf shape. First, we cluster the points according to the angles in their cylindrical projections by leveraging the symmetrical property of maize or rice plants. Second, we further cluster the points according to their shape connectivity. Third, we detect the clusters that belong to the same leaf and combine their partial point clouds. In this way, we can improve the segmentation of a point cloud, and make each segmentation correspond to an individual plant component (e.g., a leaf or a stalk).

*1) Angle-based Clustering:* In general, the leaves of a maize or rice plant grow up roughly around a vertical axis at its center. This is a unique property that allows us to first segment the points of different leaves according to their angles with respect to the central axis. To this end, we transform the points from their original Cartesian coordinates $(x, y, z)$ to the cylindrical coordinates $(r, \theta, z)$, where $\theta$ and $r$ denote the azimuthal coordinate and the radial coordinate, respectively.

We classify the points by analyzing the distribution of $\theta$ values. Figure 3 shows an example of the histogram of $\theta$ values of the leaves of a plant. We first construct a fitted histogram curve (i.e., the red smoothed curve in Figure 3) and compute the local extrema of this curve to detect the peaks. The peaks in the histogram correspond to the main directions of leaves. For example, there are several peaks (i.e., the green triangles) detected in Figure 3. Second, we compute the width of each peak as the distance between the borders (i.e., the purple lines). A border is the horizontal position of the lowest valley between two neighboring peaks. The range between two neighboring borders is used as the range of the peak fall between these two borders. Another possible way to determine the width of a peak is to use fuzzy sets [30], where all the bins that fall between the extension of the two slopes of a peak are assigned to that peak. In this way, some bins can be assigned to two or more peaks, and should be recorded and treated in a different way in later steps. In our study, we use the former method to define the peak width.

According to the position and the width of a peak, we can easily find the points covered by the peak and correspond them

approximately to one direction. Figure 5(a) shows an example of the angle-based clustering result. As only the histogram of $\theta$ has been computed and analyzed, the angle-based clustering allows us to quickly identify the leaves of the major directions at a comparably low computational cost.

However, due to the fact that the points belonging to two neighboring leaves may have the same angles in the cylindrical coordinates, these points may be in the same bin of the histogram (e.g., the blue clusters in Figure 5(a)). This problem is commonly referred as aliasing in signal processing. We address this problem to refine the angle-based clustering result using a shape-based clustering method.

*2) Shape-based Clustering:* Ideally, the points of each peak can be isolated based on their distributions in the histogram. Due to aliasing, there may be more that one leaf corresponding to the points of each peak, and these leaves are spatially disconnected. Intuitively, these points can be further separated or clustered using more spatial information. However, not all clustering methods can be applied here. For example, the *k*-means clustering method calculates the distances of the points to the center in each cluster and tends to group the points in a circular or spherical cluster. As a leaf typically has a long surface, the *k*-means clustering may produce many clusters from the points of even one leaf. Therefore, a shape-based clustering method is more suitable in this case. We choose DBSCAN [8] in our study.

In DBSCAN, a seed point is usually randomly selected to initialize the clustering process. It is desired that the point is close to the center of an actual cluster, which can enhance the clustering quality. However, in practice, it is difficult to select such a seed point from a large-scale point cloud. In particular, if the point falls on the overlapping edges of two clusters, undesirable results that falsely classify the points can be generated. To avoid this problem caused by randomization, we select a point within the bin of a peak in the histogram as the seed point. This is because the points of a peak most likely correspond to the center of a cluster (i.e., the central line of a leaf), and using such a seed point can facilitate us to classify the points of a leaf. If all the peak points have been used in clustering and there are still points unclassified, we then randomly choose a remaining point as a seed point to continue the clustering.

Figure 5(b) shows an example of the DBSCAN clustering result. We can clearly see that DBSCAN can successfully separate the unsegmented points of multiple leaves generated in the angle-based clustering. In addition, we apply DBSCAN on each cluster generated in the angle-based clustering, rather than the entire point cloud, and thereby can significantly lower the computational cost.

*3) Shaped-based Refinement:* Due to the point loss in the noise removal step, it is possible that one leaf, particularly a thin and long leaf, may be segmented into multiple clusters, e.g., the clusters in the blue circle of Figure 5(b). We combine the clusters that belong to one leaf together using two conditions. First, we call two clusters $C_i$ and $C_j$ neighboring clusters if they are close to each other in terms of the $\theta$
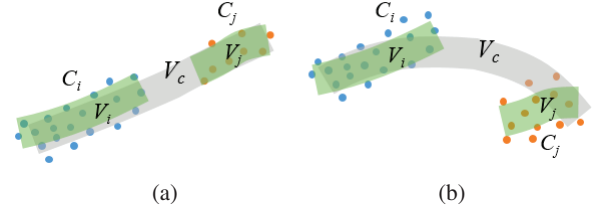


Fig. 4. Using regression volumes to determine the combination of two candidate clusters $C_i$ and $C_j$.

values. If two clusters belong to one leaf, they should form a neighboring cluster in the 2D histogram. Therefore, we only need to check if two neighboring clusters can form a single leaf. Second, for the two neighboring clusters, if their radial ranges overlap, they should not be combined into one cluster or be regarded as a single leaf. This is because each leaf grows in only one direction, and the radial range of one cluster should not overlap with the range of the other cluster if both of them belong to the same leaf.

We use these two conditions to find the clusters that potentially belong to the same leaf. We further need to determine if these clusters can be actually combined using a heuristic metric. For two candidate clusters $C_i$ and $C_j$, we construct their individual regression volumes $V_i$ and $V_j$. We note that each regression volume may not contain all the point of a cluster. We count the point numbers within $V_i$ and $V_j$ as $N_i$ and $N_j$, respectively. Then, we construct a regression volume $V_c$ of the union $C_c = C_i \bigcup C_j$, and count the point number within $V_c$ as $N_c$. If $N_c/(N_i+N_j) > e$, where $e$ is a predefined threshold, we consider that the combination of $V_i$ and $V_j$ largely match with $V_c$, and thereby determine that $C_i$ and $C_j$ can be combined, as shown in Figure 4(a). Otherwise, $C_i$ and $C_j$ belong to different leaves, and should not be combined, as shown in Figure 4(b). We set $e = 0.9$, which gives us appropriate results in our study.

To construct a regression volume $V_i$ of a point cluster $C_i$, we first project all the points of $C_i$ into the $r$-$z$ plane. Second, we fit a quadratic curve $f_i$ for all the points of $C_i$ in the $r$-$z$ plane using the least square method. Then, we compute the median value $\theta_m$ for the $\theta$ values of points in $C_i$. Finally, we construct $V_i$ as

$$V_i(r, \theta, z) = \begin{cases} |f_i(r) - z| < \delta_z \\ |\theta - \theta_m| < \delta_\theta \end{cases} \qquad (4)$$

where $\delta_z$ denotes the range of $V_i$ in the $r$-$z$ plane, and $\delta_\theta$ denotes the range of $V_i$ in the $r$-$\theta$ plane. We set $\delta_z = 5$ and $\delta_\theta = 15°$ in our study.

Using this method, we can effectively detect and combine the clusters of the same leaf. Figure 5(c) shows an example, where the two circled clusters are merged as a single cluster using our refinement method, where these two clusters are separated in the previous angle-based clustering and the shape-based clustering, as shown in Figure 5(a) and (b).
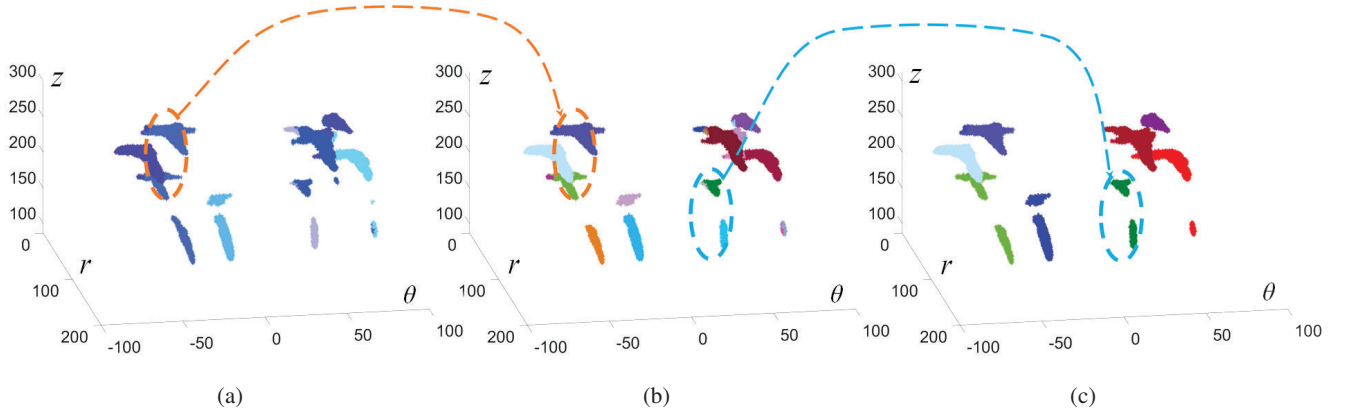
Fig. 5. An example of our clustering results in the space of $r$, $\theta$, and $z$. (a) We use the angle-based clustering to obtain the initial segmentation of the point cloud. However, due to the aliasing problem, the points of two nearby leaves may be clustered together (e.g., the circled blue clusters). (b) We further apply the shape-based clustering to separate these points. (c) We detect and combine the clusters that belong to the same leaf (e.g., the circled green cluster).
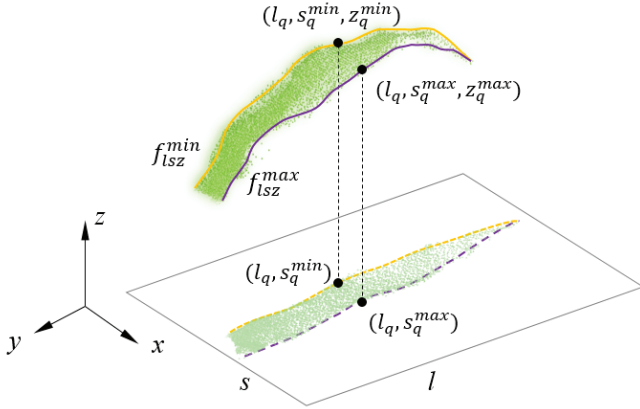


Fig. 6. Transform a leaf from its original $x$, $y$, $z$ coordinate system into a new $l$, $s$, $z$ coordinate system, where the $l$ and $s$ axes are in the $x$-$y$ plane. The point cloud of the leaf is projected into the $l$-$s$ plane. Then, we detect the 2D edge points in the $l$-$s$ plane. For an edge point, we can find its corresponding 3D point in the $l$, $s$, $z$ coordinate system. We use these 3D edge points to form two 3D curves $f_{lsz}^{min}$ and $f_{lsz}^{max}$ to fit the 3D edges of the leaf.



Fig. 7. The edge points are found using a moving strip along the long axis $L$ of a leaf.

We use a local regression method called locally estimated scatterplot smoothing (LOESS) [5], which is similar to MLS. LOESS can reconstruct a continuous surface even with the presence of the discontinuity of leaf points.

*2) Edge Fitting:* Although surface fitting can generate a smooth surface for a leaf, it can result in very serrated lines for the edges. To construct the edges of a leaf, we first detect the 3D edge points and then use smooth 3D splines to fit the edges.

It is difficult to directly detect the 3D points of the edges of a leaf due to the noises along the edges. In this work, we leverage the symmetrical property of a maize or rice leaf to obtain the informative 2D projection of the point cloud of the leaf. Then, we find the 2D edge points and their corresponding 3D points to fit the 3D edges of the leaf. Figure 6 illustrates the process. We first transform the point cloud of a leaf from its original $x$, $y$, $z$ coordinate system into a new $l$, $s$, $z$ coordinate system using the principal component analysis (PCA) method. We first project all the points of the leaf into the $x$-$y$ plane. For the projected 2D points of the leaf, we define the long axis $l$ of the leaf as the line from the leaf base to the leaf tip. The short axis $s$ is defined as an axis perpendicular to $l$ in the 2D plane. We use the classic PCA method to find these two axes. If $\lambda_1$ and $\lambda_2$ are the two eigenvalues of the covariance matrix

## C. Surface Reconstruction

Once the points of a leaf are segmented from the entire point cloud, we reconstruct a surface to fit these points using the Cartesian coordinates. The points on the leaf surface and the edges are tackled in different ways.

*1) Surface Fitting:* Least squares methods are a classic tool for surface fitting. However, a direct application of least squares tends to generate a smooth surface that can lose certain local details of the leaf. A method using local information is more suitable to reconstruct the leaf surface and capture local details. Moving least squares (MLS) [11] is widely used to generate a surface for data points. Instead of constructing a global approximation, MLS constructs and evaluates a local polynomial continuously over the entire domain. MLS can be viewed as a local regression method.
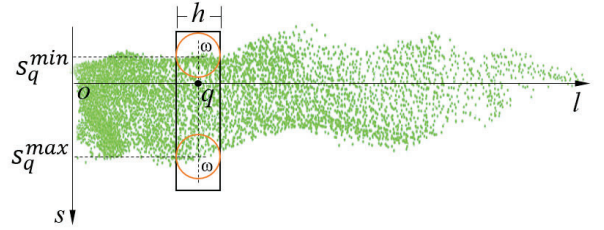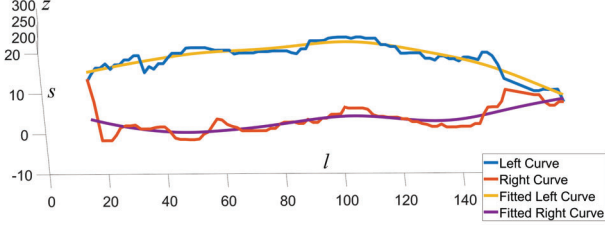
Fig. 8. The results of fitting smooth edges to the leaf using a smoothing spline.



Fig. 9. A comparison of histogram shapes using different numbers of bins: (a) $N = 50$, (b) $N = 100$, (c) $N = 200$, and (d) $N = 500$.

of all the points in the 2D plane and $\lambda_1 > \lambda_2$, the eigenvector $v_1$ corresponding to $\lambda_1$ is used as $l$, and $v_2$ corresponding to $\lambda_2$ is used as $s$. We project all the points onto $l$ and select the point with the smallest projected value as the origin $o$ in the 2D plane. We use the original $z$ axis as the third axis in the new coordinate system. Given the structure of a maize or rice leaf, the most informative 2D projection of the point cloud can be obtained in the $l$-$s$ plane.

Second, we find the 2D edge points in the $l$-$s$ plane. As shown in Figure 7, we move a small strip along $l$. At any 2D position $q(l_q, 0)$ on the $l$ axis, the strip is parallel to the $s$ axis and is centered at $q$. We denote the width of the strip as $h$. We find the range $[s_q^{min}, s_q^{max}]$ to contain most the points inside the strip and exclude some outlier points along the $s$ axis. We set $s_q^{min} = m - 2\sigma$ and $s_q^{max} = m + 2\sigma$, where $m$ and $\sigma$ are the mean and the standard deviation of the $s$ values of the points inside the strip, respectively. We use the two 2D points $(l_q, s_q^{min})$ and $(l_q, s_q^{max})$ to represent the 2D edge points within the strip.

Then, we aim to find the corresponding 3D edge points of $(l_q, s_q^{min})$ and $(l_q, s_q^{max})$. For $(l_q, s_q^{min})$, we compute the average $z$ value, $z_q^{min}$, of all the points within a small circle $\omega$ centered at $(l_q, s_q^{min})$, and thereby obtain its corresponding 3D edge point $(l_q, s_q^{min}, z_q^{min})$. The radius of $\omega$ is set to $h/2$. Similarly, we find another 3D edge point $(l_q, s_q^{max}, z_q^{max})$. We find all the 3D edge points using this moving strip. We empirically set $h$ as 2% of the total length of the points along the $l$ axis.

After we find all the 3D edge points, we use smoothing splines [6] to fit the edges. We generate two curves $f_{lsz}^{min}$ and $f_{lsz}^{max}$ corresponding to the 3D point sets of $(l_q, s_q^{min}, z_q^{min})$ and $(l_q, s_q^{max}, z_q^{max})$, as shown in Figure 6. Figure 8 shows an example of smooth edge fitting, where the blue and orange lines correspond to the original detected edges with serrated shapes, and the yellow and purple lines correspond to the smooth fitted edges. Note that the $l$, $s$, $z$ coordinate system is constructed from the $x$, $y$, $z$ coordinate system using the PCA method. We can easily transform these two curves $f_{lsz}^{min}$ and $f_{lsz}^{max}$ back to the $x$, $y$, $z$ coordinate system as $f_{xyz}^{min}$ and $f_{xyz}^{max}$, respectively.

*3) Mesh Generation:* After surface fitting, we discretize the surface functions to a 3D triangular mesh. We first use the Delaunay triangulation algorithm [10] to generate a triangular mesh in the $x$-$y$ plane. Then, for each vertex of a triangle,
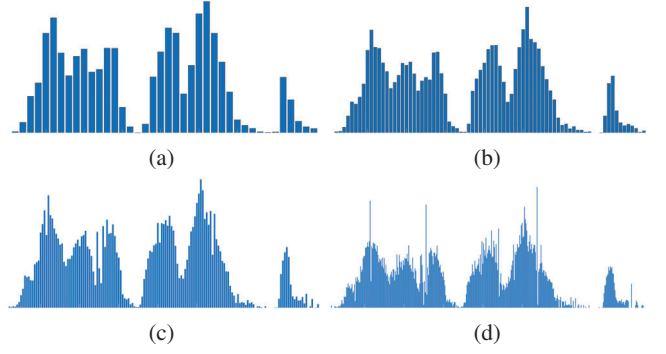
we use its $x$ and $y$ values to compute its $z$ value through the functions of the fitted surface. In this way, we can generate a 3D triangle mesh from the fitted surface.

## IV. RESULTS AND DISCUSSION

### A. Number of Bins

In this work, the histogram of the $\theta$ values is an important guide in how the point cloud should be divided. An appropriate choice of the number of bins determines if the peaks can be successfully detected or not. We experiment the number of bins $N$ as 50, 100, 200, and 500, and compare the histogram results in Figure 9 (a), (b), (c) and (d), respectively. We can see that a small $N$ cannot represent the histogram in a smooth shape, while a very large $N$ can cause high pulses in the histogram. We found that $N = 200$ is a suitable choice for the histogram and have used it in our study.

### B. Histogram Curve

The number of peaks that can be detected in the histogram of the $\theta$ values is mainly determined by three factors: the smoothness of the fitted histogram curve used to detect the peaks, the minimum distance between peaks, and the minimum height of a peak. Figure 10 compares the effects of these factors. Figure 10(a) shows the original histogram curve that may not be smooth and can contain many peaks. Not all these peaks correspond to actual leaves. For example, in Figure 10(a), several peaks (within an orange circle) appear at the low-count bins that are likely generated by noise. These false peaks can be partially removed by setting the minimum height $minHeight$ of the peak and the minimum distance $minDistance$ between the peaks, as shown in Figure 10(b). But there are still false peaks found on the histogram curve (e.g., the ones in the magenta circle of Figure 10(b)).

The problem can be alleviated by further smoothing the histogram curve. Moving average filter is commonly used in signal processing, which takes the weighted sum of the neighboring points of one point and uses the sum as the new value for the point. Moving average can reduce many false peaks detected in the histogram curve, as shown in Figure 10(c). However, a false peak (in the cyan circle of Figure
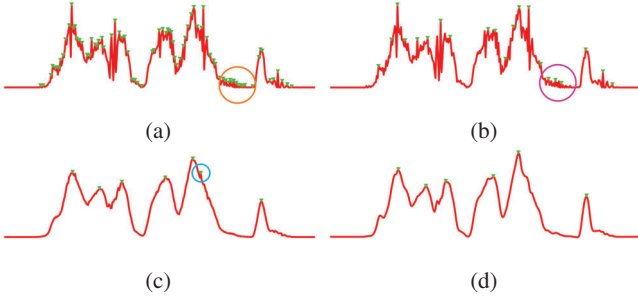
Fig. 10. The number of peaks (denoted by the green triangles) that can be found for different curves: (a) the original histogram curve, (b) the original histogram curve with $minHeight = 5$ and $minDistance = 10$, (c) the curve generated using the moving average filter, and (d) the curve generated using curve fitting.

10(c)) is found at a place where only one peak should exist. A spline based curve fitting method, such as smoothing spline, can be used here to better describe the distribution of peaks in a histogram curve. The fitted histogram curve using a smoothing spline is smoother than the ones obtained using moving average. The peaks can also be correctly found without any false peaks, as shown in Figure 10(d). Therefore, we suggest using a histogram curve fitted with smoothing spline for the task of peak detection.

### C. Parameters of DBSCAN

The two key parameters, the search radius $R$ and the minimum number of points $minPoints$, largely determine the effectiveness of DBSCAN. DBSCAN is sensitive to these parameters such that changing them can generate many different results. Figure 11 shows a comparison of the different clustering results obtained by changing these two parameters. The plots in the left column show the effect of changing the search radius $R$. When $R = 5$, all the points can be correctly clustered, where the points in one cluster belong to only one leaf, as shown in Figure 11(a). However, in Figure 11(b), when the radius $R$ is increased to 10, there is a false classification where the points (within the red circle) that belong to two leaves are in one cluster. As shown in Figure 11(c), when the radius $R$ is further increased to 100, the false classification gets more severe where a cluster (e.g., the one in the red circle) can contain the points of more than two leaves.

The minimum number of points $minPoints$ for a point to grow can also affect the clustering result. As shown in Figure 11(d), when $minPoints = 0$, many small clusters are detected. Some of these clusters are from the tips of leaves and some are from noise (within the red circle). When $minPoints$ increases to 10 in Figure 11(e), some tiny clusters will disappear. However, when $minPoints$ is set to a high value like 100, there points that belong to one leaf will be cut into many medium clusters, as shown in Figure 11(f). Although the separated clusters that belong to a leaf can be combined using our shape-based refinement method, it is still better to make the clustering results as accurate as possible,
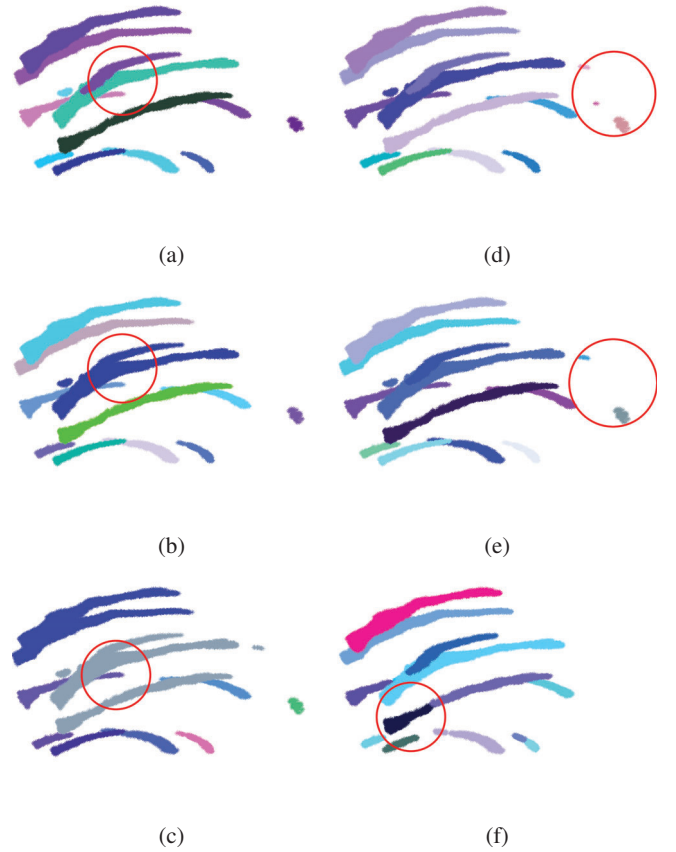


Fig. 11. A comparison of results of using different parameters: (a) $R = 5$, $minPoints = 50$; (b) $R = 10$, $minPoints = 50$; (c) $R = 20$, $minPoints = 50$; (d) $R = 5$, $minPoints = 0$; (e) $R = 5$, $minPoints = 10$; and (f) $R = 5$, $minPoints = 100$.

which can also reduce the computation costs. We used $R = 5$ and $minPoints = 50$ for the point cloud in this work.

### D. Complexity

The design in our pipeline has considerably lowered the runtime complexity compared to alternative approaches. In particular, there are two types of search operations in the pipeline. One is the search of points that belong to a cluster in DBSCAN. The other one is the search of clusters in combining clusters that belong to a leaf.

Given $n$ points, DBSCAN has an average runtime complexity of $O(n \log n)$. As we use the histogram to find peaks and apply an initial angle-based clustering of the point cloud, we employ DBSCAN on each cluster from the angle-based clustering, instead of the whole point cloud. If the point cloud contains $N_p$ peaks in the histogram, then the average complexity is $N_p * O(\frac{n}{N_p} \log \frac{n}{N_p}) = O(n \log n) - \log N_p * O(n)$, which is smaller than the original complexity $O(n \log n)$ of DBSCAN.

In the step of shape-based refinement, a direct approach is to compare any two clusters and see if they can form a bigger cluster. If there are $N_c$ clusters, the complexity of this

approach is $O(\frac{N_c(N_c-1)}{2!}) = \frac{1}{2}O(N_c(N_c-1))$. In our method, as we only compare neighboring clusters in terms of $\theta$ values, the complexity is $O(N_c - 1)$, which is much smaller than the original complexity $\frac{1}{2}O(N_c(N_c-1))$.

### E. Reconstruction Results

We compare the reconstructed meshes for all the leaves of a maize plant using our method and existing approaches. Figure 12(a) shows the result of the Poisson surface reconstruction method [15]. We can clearly see the bumpy surface caused by noises, which are difficult to be tackled only using local Poisson disks. Figure 12(b) shows the result of the B-spline surface reconstruction [7], where the surface is much smoother compared to the Poisson surface. However, this method cannot obtain appropriate results in the region with high noises. For example, in the highlighted region in Figure 12(b), the reconstructed surface is twisted to over-fit the local noisy point cloud. Our method addresses this issue by applying local regression, and thereby can significantly improve the reconstruction result, as shown in Figure 12(c). We note that edge fitting can further improve the surface reconstruction for leaves of plants, as shown in the images Figure 12(d).

We further quantitatively evaluate the reconstructed surfaces using the variance of surface normals. Table I shows the reconstruction results of eight leaves of a maize plant among the Poisson method, the B-spline method, and our method with and without edge fitting. We can clearly see that our method can generate much smoother leaf surfaces with smaller normal variances. In particular, when applying edge fitting, the average variance of our method is only 28.7% and 56.8% of ones of the Poisson and B-spline methods, respectively. Figure 13 shows the real maize plant and our reconstructed leaves. Our method can generate a reconstructed representation approximating the real plant.

We enhance a final model to resemble the appearance of a real plant by adding the textures of leaves and stalks, as well as the pot when the plant grows. Figure 14 shows examples of maize and rice plants. Our 3D reconstruction results can facilitate researchers to gain an intuitive but detailed understanding of plants, and lead to possible new scientific discoveries.

TABLE I
A COMPARISON OF THE VARIANCE OF SURFACE NORMALS.

| Method | Leaf1 | Leaf2 | Leaf3 | Leaf4 | Leaf5 | Leaf6 | Leaf7 | Leaf8 | Average |
|---|---|---|---|---|---|---|---|---|---|
| Poisson | 0.85 | 0.88 | 0.91 | 0.87 | 0.88 | 0.86 | 0.87 | 0.84 | 0.87 |
| B-spline | 0.21 | 0.88 | 0.29 | 0.48 | 0.79 | 0.15 | 0.15 | 0.60 | 0.44 |
| Our method[1] | 0.41 | 0.31 | 0.21 | 0.38 | 0.26 | 0.20 | 0.35 | 0.17 | 0.29 |
| Our method[2] | 0.33 | 0.28 | 0.18 | 0.26 | 0.25 | 0.18 | 0.33 | 0.16 | 0.25 |

## V. CONCLUSION

We present a pipeline to reconstruct surface representations from point clouds of plants, where the point clouds can be generated by either active approaches or passive approaches. After removing background and noises, we design a two-step

[1]Our method without edge fitting.
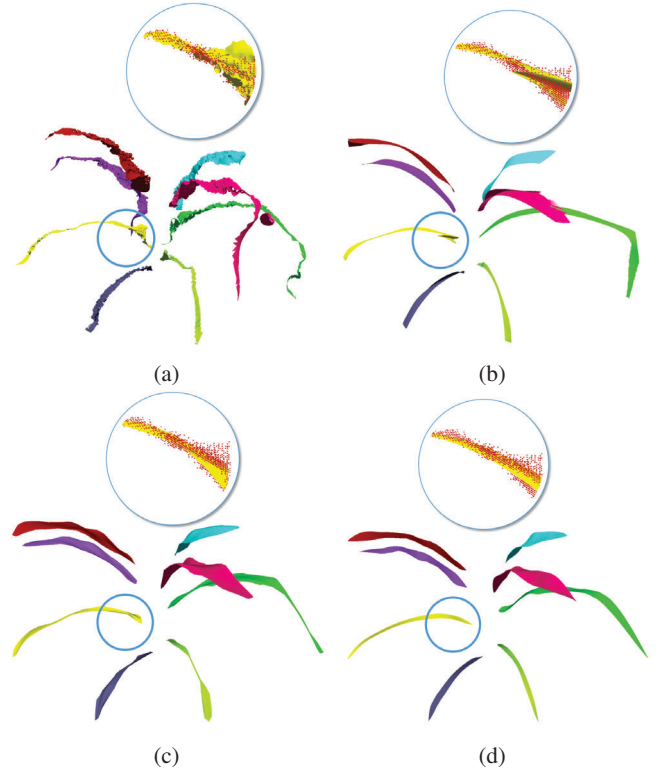[2]Our method With edge fitting.



Fig. 12. A comparison of reconstructed meshes for leaves of a maize plant: (a) Poisson surface reconstruction, (b) B-spline surface reconstruction, (c) our method without edge fitting, and (d) our method with edge fitting.



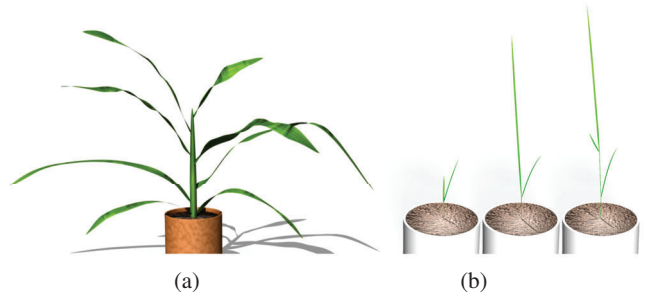Fig. 13. A comparison of (a) the real plant and (b) the reconstructed leaves.



Fig. 14. Rendering of 3D models with texture and lighting enhancements: (a) a maize plant and (b) a rice plant at three time points.

clustering approach and the refinement process to effectively and efficiently segment a point cloud into different components by leveraging the structural properties of the maize or rice plant. Then, we reconstruct each component using surface fitting and edge fitting to ensure the smoothness of resulting surfaces. The final results, with texture and lighting enhancement, provide a faithful digital 3D representation of the original plant in the real world. Our detailed experimental study has explored the effectiveness of parameter changes and provided useful guidelines on selecting parameters for practical usages. Our method significantly enhances the quality of 3D reconstruction for high-throughput plant phenotyping.

In our study, we leverage the constraints of the maize or rice plant. However, these constraints do not always hold for plants with more complex structures. In the future, we would like to investigate algorithms to automatically characterize plant structural properties and derive constraints for obtaining more accurate reconstruction results. In addition, we plan to extend our current pipeline to tackle time-varying 3D point cloud and detect important plant dynamics during its process of growth.

## VI. ACKNOWLEDGMENT

## REFERENCES

[1] D. Andújar, C. Fernández-Quintanilla, and J. Dorado. Matching the best viewing angle in depth cameras for biomass estimation based on poplar seedling geometry. *Sensors*, 15(6):12999–13011, 2015.

[2] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017.

[3] Y. Chéné, D. Rousseau, P. Lucidarme, J. Bertheloot, V. Caffier, P. Morel, É. Belin, and F. Chapeau-Blondeau. On the use of depth camera for 3D phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82:122–127, 2012.

[4] Y. H. Chew, R. W. Smith, H. J. Jones, D. D. Seaton, R. Grima, and K. J. Halliday. Mathematical models light up plant signaling. *The Plant Cell*, 26(1):5–20, 2014.

[5] W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[6] E. R. Cook and K. Peters. The smoothing spline: a new approach to standardizing forest interior tree-ring width series for dendroclimatic studies. *Tree-ring bulletin*, 1981.

[7] A. Dimitrov, R. Gu, and M. Golparvar-Fard. Non-uniform B-spline surface fitting from unordered 3D point clouds for as-built modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(7):483–498, 2016.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[9] N. Fahlgren, M. A. Gehan, and I. Baxter. Lights, camera, action: high-throughput plant phenotyping is ready for a close-up. *Current opinion in plant biology*, 24:93–99, 2015.

[10] D. A. Field. Laplacian smoothing and delaunay triangulations. *International Journal for Numerical Methods in Biomedical Engineering*, 4(6):709–712, 1988.

[11] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *ACM transactions on graphics (TOG)*, volume 24, pages 544–552. ACM, 2005.

[12] J. A. Gibbs, M. Pound, A. P. French, D. M. Wells, E. Murchie, and T. Pridmore. Plant phenotyping: an active vision cell for three-dimensional plant shoot reconstruction. *Plant physiology*, 178(2):524–534, 2018.

[13] B. Hofle. Radiometric correction of terrestrial LiDAR point cloud data for individual maize plant detection. *IEEE Geoscience and Remote Sensing Letters*, 11(1):94–98, 2014.

[14] T. Ijiri, S. Yoshizawa, H. Yokota, and T. Igarashi. Flower modeling via x-ray computed tomography. *ACM Transactions on Graphics (TOG)*, 33(4):48, 2014.

[15] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013.

[16] P. Kumar, J. Connor, and S. Mikiavcic. High-throughput 3D reconstruction of plant shoots for phenotyping. In *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, pages 211–216. IEEE, 2014.

[17] Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen. Analyzing growing plants from 4D point cloud data. *ACM Transactions on Graphics (TOG)*, 32(6):157, 2013.

[18] Y.-J. Liu, J.-B. Zhang, J.-C. Hou, J.-C. Ren, and W.-Q. Tang. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1700–1707, 2013.

[19] Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, and B. Chen. *Texture-lobes for tree modelling*, volume 30. ACM, 2011.

[20] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[21] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, volume 28, pages 493–501. Wiley Online Library, 2009.

[22] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[23] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 599–604. ACM, 2006.

[24] F. Remondino and S. El-Hakim. Image-based 3D modelling: a review. *The photogrammetric record*, 21(115):269–291, 2006.

[25] S. E. Reutebuch, H.-E. Andersen, and R. J. McGaughey. Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory. *Journal of Forestry*, 103(6):286–292, 2005.

[26] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[27] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[28] S. Thapa, F. Zhu, H. Walia, H. Yu, and Y. Ge. A novel lidar-based instrument for high-throughput, 3D measurement of morphological traits in maize and sorghum. *Sensors*, 18(4):1187, 2018.

[29] C. Xia, L. Wang, B.-K. Chung, and J.-M. Lee. In situ 3D segmentation of individual plant leaves using a RGB-D camera for agricultural automation. *Sensors*, 15(8):20463–20479, 2015.

[30] H.-J. Zimmermann. *Fuzzy set theory and its applications*. Springer Science & Business Media, 2011.