Plant Event Detection from Time-Varying Point Clouds

Tian Gao[1], Jianxin Sun[1], Feiyu Zhu[1], Henry Akrofi Doku[2], Yu Pan[1], Harkamal Walia[2], Hongfeng Yu[1]
[1] Department of Computer Science and Engineering
[2] Department of Agronomy and Horticulture
University of Nebraska-Lincoln, Lincoln, NE, USA

***Abstract*—Studying the growth dynamics of developing plants is of critical importance in plant sciences. The traditional methods rely on either manual measurement, which involves tedious labor work, or 2D image-based approaches, which cannot fully characterize plants in 3D. Given the advances of scanners and 3D reconstruction methods, scientists begin to pay more attention to 3D models to improve accuracy. However, existing methods mostly focus on the growth of a whole plant rather than its detailed substructures. In this paper, we have developed an end-to-end pipeline to detect the key events on both the whole plant and the specific components. Our method is achieved by building 3D models from images, segmenting individual components, and capturing traits. We implement an experiment on maizes for evaluation and successfully detect events in the process of growth.**

***Index Terms*—plant growth analysis, point cloud, event detection, skeleton, leaf segmentation**

I. Introduction

Studying the growth dynamics of developing plants is one of the most critical topics for plant scientists to gain knowledge of the relationship between plant phenotypes and environmental stresses [1]. One of the main traditional methods is to record the growth of the plants manually. However, they usually involve tedious labor work, increase the chances of damaging plants, and are lack of the availability of large-scale measurements. Another type of widely used methods is based on 2D images, which can be scaled to large-scale observations. However, such 2D methods cannot fully describe the complete structure of a plant as images are 2D projections of 3D objects. During the process of projection, part of 3D geometric information is inevitably lost, and extra efforts are needed to recover them for certain analysis requirements. For example, image data have to be calibrated when plant scientists need to quantify phenotype traits, such as leaf length and leaf azimuth [2], [3].

Thanks to the development of imaging and computer vision techniques, 3D phenotyping approaches are drawing increasing attention to plant scientists recently. 3D methods have shown promising potentials to address the limitations of 2D methods. By representing plants in 3D, 3D models (such as point clouds or polygon meshes) are able to describe the complete shape of plants. In addition, similar to the image-based approaches, 3D methods do not require intensive manual work and thus can be scaled to large-scale experiments.

However, there are several key challenges associated with current 3D phenotyping methods. First, it is difficult to obtain a complete 3D model of a plant due to occlusion resulted from the complex structure of plants. Exterior tissues of a plant (e.g., leaves) can often occlude the interiors (e.g., stems and other leaves). Therefore, a generated 3D model may be incomplete [4] and lead to less accurate results in further analysis. Second, the resolution of a 3D model may be not high enough, and small tissues of the plants may be too subtle to be accurately captured as a result. For example, buddings may be wrongly considered as noise and ignored due to their small sizes. So the widely used depth-camera based solutions [5] may not be suitable for capturing subtle components because of their comparably insufficient resolutions. Third, some components such as leaves are non-rigid and will not stay still if they are disturbed by wind or motion applied to the plant. In some popular Structure From Motion (SFM) based methods, plants are placed on a turntable and keep rotating such that cameras can capture images from different view angles. As a result, in the process of rotating, the leaves of the plants can vibrate. Thus, reconstructed point cloud will include obvious noises, especially at the tips of leaves [2]. Researchers have attempted to solve the vibration issue by building complex imaging systems [6], [7]. However, these imaging systems are usually expensive and require professional mechanical techniques to implement.

In this paper, we propose a plant imaging platform as well as an end-to-end solution to detect events of plant growth based on time-varying 3D point clouds. In our imaging system, a plant is placed still at a fixed position, and two cameras at different heights rotate around the plant while taking images. The images are captured from different view angles by the two cameras. This can reduce the occlusion and help capture the components of a plant in the center. Our pipeline reconstructs point clouds with images captured by high-resolution cameras. A generated point cloud is detailed to facilitate us to recognize tiny tissues and conduct analysis. As a plant remains still in the center, our system can minimize the vibration of the plant leaves. Our experiments show that the point clouds can be generated with high stability and less noises. Meanwhile, the system is easy to build, and all the necessary materials are affordable. With the point cloud of a plant, we segment all the individual leaves and calculate traits from the point cloud of leaves. Finally, we detect the critical events on each leaf based on its traits over time.

II. Related Work

A. 3D Reconstruction

3D reconstruction is a hot topic in computer vision, and there are many methods to generate the point cloud of an object. One of the most widely used approaches involves depth sensors that are used to generate the point cloud directly. In these methods, a plant is usually fixed, and multiple sensors move around the plant to scan it for obtaining raw point clouds from different view-angles. Then, these point clouds are merged together to generate the final one [8], [9]. The problem with depth sensors is that the resolution of a scanner is usually not high enough to capture more details. Therefore, the resulting point clouds can suffer from imperfections, such as missing points and noises. Though scientists have developed various methods to improve the models [10]–[12], these imperfections can hardly be eliminated and hence reduce the accuracy of analysis based on these models.

Besides these methods based on depth sensors, image-based solutions are also quite popular. For example, Multi-View Stereo (MVS) and Structure from Motion (SfM) calculate the positions of points in 3D space by pairing images from different view angles. One of the main advantages of these methods is that the point cloud is relatively more detailed as the commonly used cameras have very high resolutions. These methods have been proved to be a successful implementation in 3D model reconstruction and growth estimation of plants [13], [14].

B. Time-varying Plant Modeling

The growth dynamics of plants are of researchers' interest in botany [15]. Unlike destructive measurements where researchers measure plants for one time after harvesting them [16], non-destructive methods allow scientists to observe the dynamics of plant growth. With the help of time-varying data, researchers are able to quantify the differences of the same plant on different days or create the animation of plant growth with high quality. Traditionally, capturing of time-varying data involves tedious manual measurement, and researchers have tried various methods to reduce labor. For example, Li et al. have explored video as the data source to model a growing tree [17]. Fernandez et al. presented a semi-manual approach to track plant growth at cell resolution [18]. In our work, we develop a method that is based on 3D point clouds and corresponding skeletons to track and analyze the dynamics of plant growth.

C. Skeleton Extraction

A skeleton of an object is a curve that describes the topology of the object. Traditional, scientists mainly extract skeletons from 2D images, and the thinning algorithms to fetch the skeleton is relatively easy. As the availability and the power of 3D scanner increases, researchers are not only satisfied with the skeletons from 2D images since they are not fully accurate and need to be calibrated due to the missing dimension [19]–[21]. They begin to pay more attention to the skeletons extracted from 3D models directly. Compared to the 2D methods, 3D skeleton extraction is more complex because of the additional information. However, the result is more accurate, and more traits can be directly generated [8]. Examples of skeleton extraction algorithms include rotational symmetry axis (ROSA) based method [22], L1-medial skeleton [11] and Laplacian skeleton extraction [23]. The first two methods are successfully implemented in cylindrical-shaped point clouds, and the last one is able to extract curves directly from point cloud without extra mesh reconstruction.

D. Event Detection

Researchers are interested in extracting the hidden information behind the time-varying data usually presented in videos or a series of images. Brendel et al. [24] considered the videos as training data and presented a learning-based graph model to detect and localize relevant activities. Pirsiavash et al. [25] proposed a method to detect the activities of daily living, which involved long-scale temporal structures and complex object interactions using first-person camera views. Recently, time-varying 3D data received increasing interest from researchers, given its potential to present complex events with 3D information. Shotton et al. [26] proposed a real-time method to recognize human poses with depth images.

Our pipeline in this paper is similar to Li's work [2], in which the authors proposed a framework to study the growth dynamics of plants by comparing the point clouds. However, instead of detecting events on the whole plant, we separate the individual leaves and focus on events on each leaf.

III. Methods

The end-to-end pipeline we develop in this work to detect events on plants involves four steps, namely, (i) image acquisition with our imaging system; (ii) point cloud reconstruction using MVE [27]; (iii) plant segmentation; and (iiii) traits calculation for event detection. The overall pipeline is illustrated in Fig. 1.

A. Image Acquisition

We construct an efficient plant imaging platform and rotate the cameras around plants to capture images from multiple view-angles for 3D reconstruction (Fig. 2). There are two main parts of the platform: (i) a rotating imaging system; and (ii) color checkerboards and backdrops.

1) Imaging System: To rotate the cameras, we have a rotary double-ring lazy susan bearing turntable ring apparatus. The inner ring is fixed on the ground while the outer ring can move freely. The outer ring is connected to a wooden board, and two robot cars are connected to the other end of the board, providing power for rotation. The plants in pots are placed in the center of the ring apparatus, and a blue mesh of the same size is placed on the soil in the pots for calibration in the further steps. The wooden board holds two Sony α6500 cameras, and the position of the cameras is adjustable according to the size of the plant. The resolution of our cameras is high

Fig. 1: Overview of our end-to-end pipeline. Left: our imaging system. Middle: 3D reconstruction and a selected leaf. Right: traits calculation for event detection of the selected leaf.

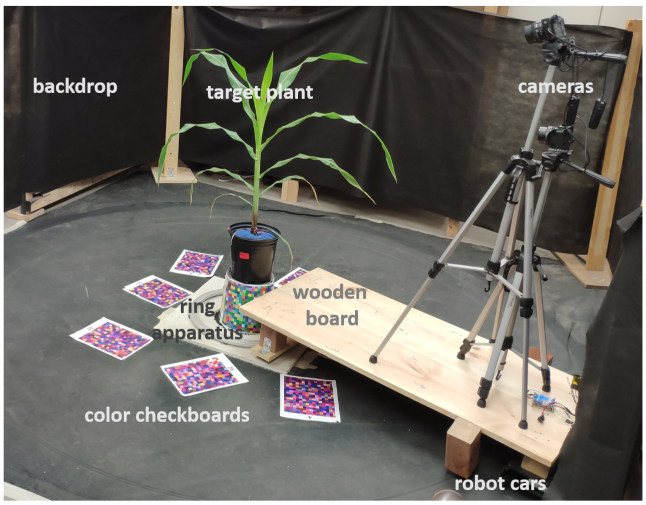


Fig. 2: Plant imaging platform

enough to capture all the details of the plants, and the high stabilization minimizes the distortion caused by the camera motion. A camera built-in application called "time-lapse" is utilized to capture images in sequence at a constant speed. In our experiment, to balance the accuracy and computing speed, the parameters of the system are adjusted so that it takes one minute to finish rotation, and each camera takes 60 images during motion. We used quick response (QR) codes to label the plants, and these QR codes can be recognized in the later image processing steps. All the components of the imaging system are commercial off-the-shelf at a relatively low cost.

2) Color Checkerboards and Backdrops: The other parts of the plant imaging platform include black backdrops and color checkerboards. The backdrops around the apparatus are used to minimize noises in the background while the color checkerboards on the ground and posted on the pot help reconstruct point clouds in a more stable fashion. These color checkerboards will generate extra feature points in the 3D reconstruction process to recover camera parameters, including the intrinsic calibration (i.e., radial distortion of the lens and the focal length) and the extrinsic calibration (i.e., the position and orientation of the camera) [27].

B. Point Cloud Reconstruction

In this step, we consider a set of images as input and reconstruct the 3D point cloud. First, we preprocess the images and filter out the background to speed up computation. Then, we exploit MVE to generate a raw point cloud. Finally, we post-process the raw point cloud and generate the final point cloud that only includes the plant of interest.

1) Image Preprocessing: The purpose of image preprocessing is to decrease the number of pixels to be processed in the images and hence reduce time cost and the size of the point cloud files. First, the RGB (red, green and blue) images are converted into HSV (hue, saturation and value) color space. Then, the background is removed if the hue, saturation, and value are not in the ranges of 0.2-0.5, 0.5-1, and 0.2-0.7, respectively. These thresholds work for all the images given the constant light condition in our work. After background removal, the images are denoised by removing all the isolated pixels.

2) Point Cloud Generation: With the preprocessed image sequences, we implement MVE [27] to generate point clouds of the plants. Firstly, MVE reconstructs camera parameters (such as the position and orientation of the camera) and a 3D scene by detecting and matching feature points. Then, with the 3D scene, depth maps are constructed, and the number of depth maps corresponds to the number of the input images. Finally, these depth maps are merged to generate a dense 3D point cloud as output. The pipeline of MVE is shown in Fig.3.

3) Point Cloud Postprocess: With the raw point cloud, we implement a two-step filtering and generate the final point cloud that includes the plant only. Firstly, we implement 3D clustering on the raw point cloud to remove the background. For each cluster, we check the number of green points and total points. Since the plant is the only green object in the scene, the cluster with the highest percentage of green points is considered as the target one that includes the plant in the pot. To define a green point, we are using the Visible Atmospherically Resistant Index (VARI), which is designed to emphasize vegetation in the visible portion of the spectrum [28]. The VARI of a point can be calculated using Equation 1:

$$VARI = \frac{Green - Red}{Green + Red - Blue}, \tag{1}$$

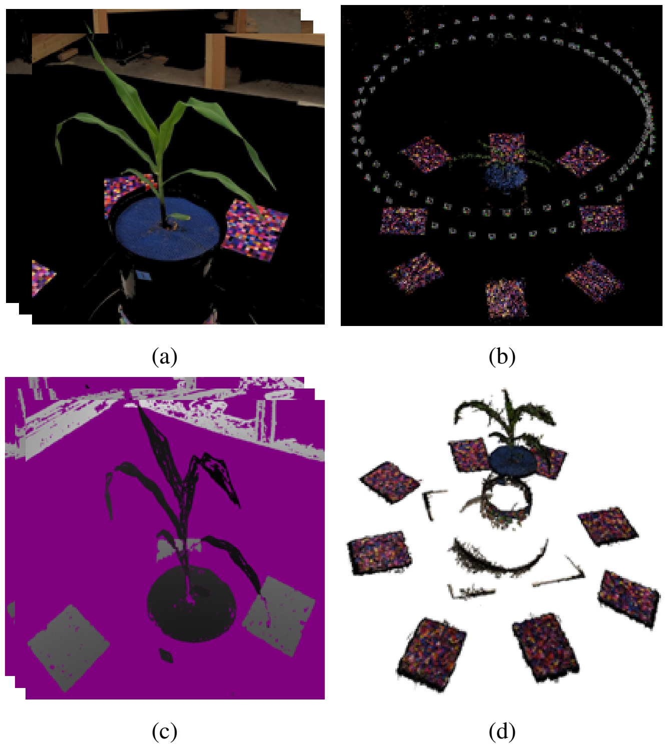


Fig. 3: The pipeline of MVE. (a) Input images. (b) Reconstructed 3D scene including cameras and sparse point cloud. (c) Depth maps. (d) A dense 3D point cloud.

where Green, Red and Blue are the corresponding values of RGB channels, respectively. A point is considered as green if its VARI is large than a threshold. In this work, we set the threshold as 0.1.

After removing the background, the cluster needs to be further filtered since the blue mesh for calculation will be in the cluster as well. Since the blue mesh is almost flat, we remove all the green points in the cluster and fit a plane for the blue mesh after denoising. Then, points near this fitting plane are considered as mesh, and the rest parts belong to the plant. Finally, we capture two point clouds: the blue mesh for calculation and the plant. The process of two-step filtering is illustrated in Fig. 4.

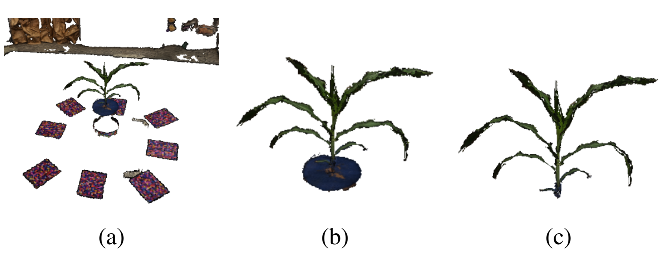


Fig. 4: Point cloud postprocess. (a) The raw point cloud generated by MVE. (b) The point cloud of the plant in pot with background removed. (c) The final point cloud of the plant with blue mesh removed.

Different point clouds may have different local origins and scales. To facilitate our analysis, we build a global coordinate system for all point clouds. Since it is known that the blue mesh has the same size among point clouds, we use it as a reference to perform the calibration. Assume the original local Cartesian coordinate system of the blue mesh is X-Y-Z. First, we create a new Cartesian coordinate system X'-Y'-Z'. The three bases of this new Cartesian coordinate system are calculated using the principal component analysis (PCA) method, and the origin is the middle point of the mesh. Then, we transform each point cloud from its original coordinate system to this new one. After that, we project the point clouds from 3D space to 2D space (i.e., X'-Y' Cartesian coordinate system). For the projected points, we fit the points on edge to a circle and calculate the radius R of this circle. Then, the coordinates of the point clouds are rescaled by dividing R. The projected blue mesh is illustrated in Fig. 5.

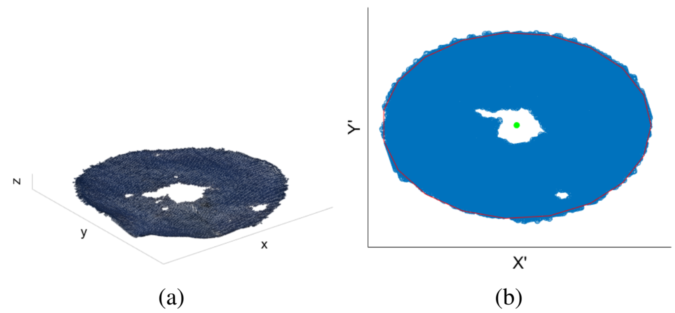


Fig. 5: Mesh projection. (a) Point cloud of the mesh. (b) Projection of the mesh, where red curve is the fitting circle and the green point is the center of the circle.

C. Plant Segmentation

The purpose of plant segmentation is to segment all the individual leaves for traits calculation. We follow two steps to achieve this: (i) stem detection; and (ii) individual leaves segmentation. The process of segmentation is shown in 6.

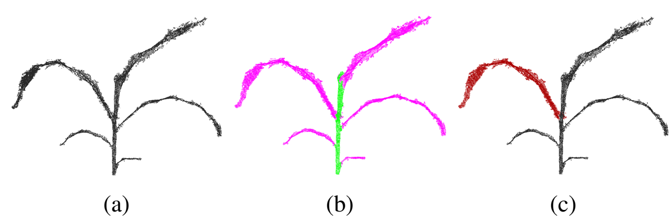


Fig. 6: Leaf segmentation. (a) The original point cloud. (b) Segmented stem and leaves, where green points are the segmented stem and pink points are the leaves. (c) A individual leaf marked red.

1) Stem Detection and Segmentation: Stem detection is a binary 3D semantic segmentation on a plant point cloud by determining the label of each point in a point cloud. It utilizes extrinsic 3D information of the point cloud and segments it into two subsets: points that belong to the plant stem and the rest of the plant. An accurate stem detection is a critical step that helps extract structural parameters of the plant with high fidelity.

The point cloud is shifted to the center by overlapping the centroid of the point cloud with the origin of the Cartesian coordinate system. For a better extraction of the points around the stem, we roughly regulate the orientation of the raw point cloud by rotating the point cloud such that the stem is pointing toward the positive Z'-axis. The rotation matrix is calculated by evaluating the normal orientation of the plane that fits the blue mesh. The colors associated with 3D points are mostly green and provide less useful information for stem detection, and thereby the color information is deleted from the point cloud.

The point clouds of all plants in our work share a common feature, which is that all plants have a single stem, and this stem is the only cylinder-like structure of the point cloud. The rest of the point cloud belongs to the leaves and has thin layers with organic contour. We utilize this observation and localize the stem of the point cloud by fitting a single cylinder. The fitted cylinder can also be used as a geometric approximation of the stem surface.

Even though we roughly correct the orientation of the point cloud, the stem of the input point cloud is not perfectly aligned with the Z'-axis. Therefore, the fitting approach has to preserve permutation invariance to the input 3D point cloud. The robust fitting approach, RANSAC [29], can properly address the problem and is used in our work to perform cylinder fitting on the point cloud. In order to increase the robustness, we need to exclude the points that are far from the stem location. We first construct a naive cylinder along the Z'-axis, where its cylinder centroid axis goes through the stem root point, the point having minimal value on the Z'-axis. Secondly, only the points fall inside this naive cylinder volume are kept as the input point cloud and fed to the RANSAC pipeline. We apply the cylinder based plant stem fitting method [30] [31] to find the optimal cylinder parameters. Table I shows all 7 fitted cylinder parameters. Specifically, the vector (x, y, z) is the coordinate of the center of the circle on one side of the cylinder, the vector (a, b, c) is the vector presenting the direction of the cylinder, and its norm represents the length of the cylinder, and d represents the radius of the cylinder.

Index	Label	Cylinder Parameters
1	x	x coordinate of bottom circle
2	y	y coordinate of bottom circle
3	z	z coordinate of bottom circle
4	a	cylinder direction on x-axis
5	b	cylinder direction on y-axis
6	c	cylinder direction on z-axis
7	d	cylinder radius

TABLE I: Fitted cylinder parameters

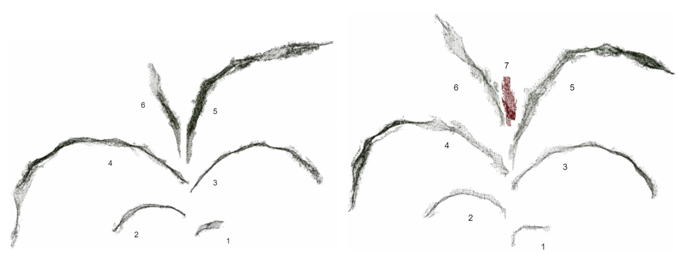
Once we have the cylinder parameters, a transformation matrix can be constructed from those parameters for accurately correcting the orientation of the original point cloud. Here, we make the stem of the plant growing upward along the positive Z'-axis. Based on the cylinder model, we separate the original point cloud into two point clouds, stem and leaf point clouds. We first construct a boundary radius by increasing the modeled radius of the cylinder by 5%, which will be used as a threshold to segment the point cloud. Then, we calculate the euclidean distance between every point of the point cloud and the centroid axis of the cylinder. All points with a shorter euclidean distance comparing to the boundary radius are labeled as stem points, and the rest are labeled as leaf points.

2) Leave Segmentation: In the leaf point clouds, since the stem is already removed, points that belong to different leaves are relatively far away from each other. We implement 3D point cloud clustering on these leaf point clouds, and each cluster can be considered as an individual leaf. Each leaf is stored as an individual point cloud file for further calculation of traits.

D. Traits Calculation

We calculate four traits of both the whole-plant level and the detailed leaf level. The four traits are (i) number of leaves; (ii) leaf length; (iii) leaf projection area; and (iiii) green point percentage.

1) Number of Leaves: The first trait of a plant is the number of leaves. The number can be used to detect some plant-level events, such as budding and decay, as this kind of event will lead to changes of the leaf number. The leaves counting process is shown in Fig. 7.



(a) (b)

Fig. 7: Counting number of leaves. (a) Point cloud of all leaves on day 2. (b) Point cloud of all leaves on day 3. The new leaf is marked red.

2) Leaf Length: In this work, we calculate the length of a leaf by capturing the skeleton and estimating its length. The skeleton is a curve that describes the geometric shape of a leaf. To capture the skeleton, we first implement the classical restricted Laplace operator [32] to the point clouds of the leaves. The restricted Laplace operator shrinks the point cloud with the following three main steps:

First, a point cloud P is iteratively generate by solving Equation 2:

$$\begin{bmatrix} W_L^t L^t \\ W_H^t \end{bmatrix} P^{t+1} = \begin{bmatrix} 0 \\ W_H^t P^t \end{bmatrix}, \tag{2}$$

where t is the iteration step; L is a $n \times n$ weighted Laplacian matrix generated with Delaunay neighborhood points; and W_L

and W_H are the diagonal matrices to control the intensity of the contraction and the attraction forces of the original points respectively.

Second, W_L and W_H are updated using Equation 3:

$$\begin{aligned} W_L^{t+1} &= s_L W_L^t \\ W_{H,i}^{t+1} &= W_{H,i}^0 \frac{S_i^0}{S_i^t} \end{aligned} \tag{3}$$

where S_i^0 and S_i^t are the current and initial neighborhood extent of point p_i respectively; and $s_L = 3.0$ is a fixed parameter to control iteration speed.

Third, a new Laplacian matrix L^{t+1} is reconstructed using the new point cloud P^{t+1}.

The iteration stops when the convergence of the solution is reached, or the count of iteration is greater than 20. As shown in Fig. 8(b), the contraction of the point cloud is thin and skeleton shaped, which approximates a curve.

Though the contraction has already represented the geometric shape of the leaf, the number of points in it still remains the same. In order to capture the skeleton curve, it is necessary to implement sampling to reduce the number of points. In this work, the sampling approach we implement is the farthest-point method with a ball of radius r. The radius r is adaptive to the shape of the leaf. Specifically, if the shape of the leaf is flat, we will sample the contraction using a ball with a large radius. On the other hand, if the shape of the leaf is complicated due to bending or twist, we will use a small ball in order to obtain an accurate local shape. To determine whether the local region of the leaf has a complicated shape, we introduce $l(v)$ to describe the complexity on each point v [33]. As defined in Equation 4, C is a 3×3 covariance matrix; $v_1, ..., v_k$ are k local neighbors of a point p; $\bar{v}$ is the centroid of $v_1, ..., v_k$; V_l and λ_l are the eigenvector and eigenvalue of the matrix C where $\lambda_0 \leqslant \lambda_1 \leqslant \lambda_2$. Fig. 8(c) illustrates the sampled points on contraction.

$$\begin{aligned} l(v) &= \frac{\lambda_2}{\lambda_0 + \lambda_1 + \lambda_2} \\ CV_l &= \lambda_l V_l, l \in 0, 1, 2 \\ C &= \begin{bmatrix} v_1 - \bar{v} \\ ... \\ v_k - \bar{v} \end{bmatrix}^T \begin{bmatrix} v_1 - \bar{v} \\ ... \\ v_k - \bar{v} \end{bmatrix} \\ \bar{v} &= \frac{1}{k} \sum_{i=1}^{k} v_i \end{aligned} \tag{4}$$

The next step is to connect the sampled isolated points to form a skeleton curve. In this work, we connect each point to its nearest neighbors and consider the corresponding segments as parts of the whole curve. Finally, all the lengths of the segments are summed up as the length of the leaf. A connected skeleton curve is shown in Fig. 8(d).

3) Leaf Projection Area: Leaf projection area is the area of the leaf projected on the ground. To calculate this area, we first find the projected 2D points of the leaf on the ground by projecting all the 3D points in the original X'-Y'-Z' coordinate system into the X'-Y' coordinate system. For the projected 2D points, we find a closed curve as the boundary [4]. The area of the region inside the closed curve is considered as the projection area. An example of leaf projection area is illustrated in Fig. 9

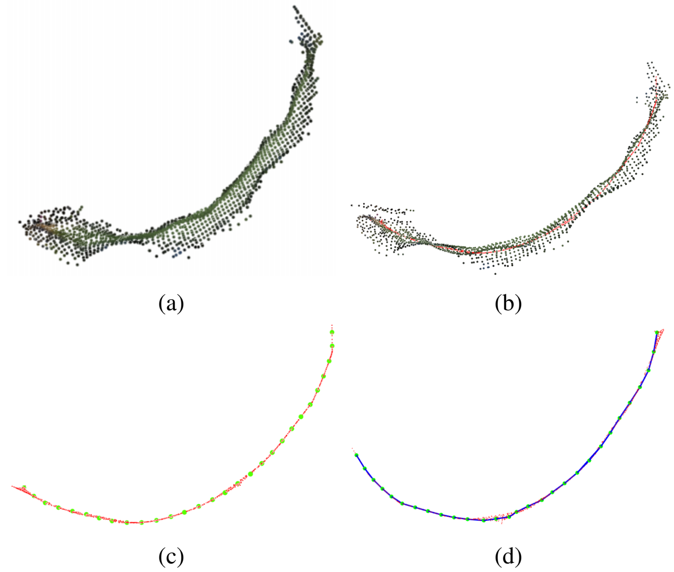


Fig. 8: Leaf length. (a) The original point cloud of a leaf. (b) The contraction and the point cloud. Contraction is marked red. (c) The sampled points and the contraction. The points are marked green. (d) The skeleton curve. The segments connecting sampled points are marked blue.

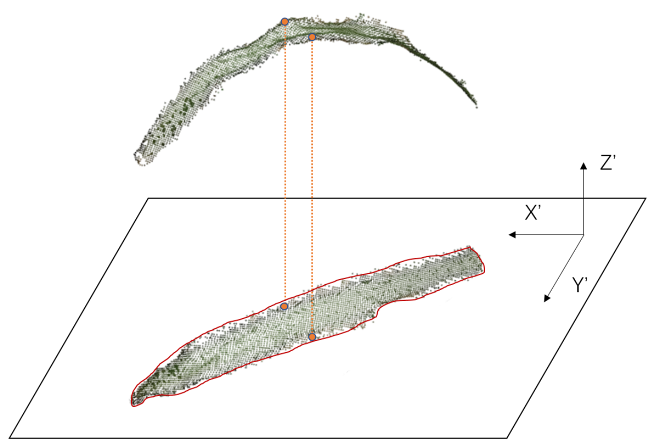


Fig. 9: An example of leaf projection area. The area inside the red curve is the projected area of the leaf.

4) Green Point Percentage: As the last trait, we measure the color of the leaf by calculating the percentage of green points. The percentage will decrease when a leaf becomes yellow. As stated in Equation 1, we calculate the VARI to define a green point. The threshold of VARI to determine a green point is 0.1.

IV. Results and Discussion

In order to evaluate our pipeline, we conduct an experiment using 36 maize plants for event detection. We grow all these plants from the seeds and begin to capture images when they are several inches in height. The data acquisition process remains over a period of 4 weeks. With the images, we build the 3D models of the plants, calculate the traits, and smooth the results to reduce noises. Traits of some leaves from one plant are shown in Fig. 10.

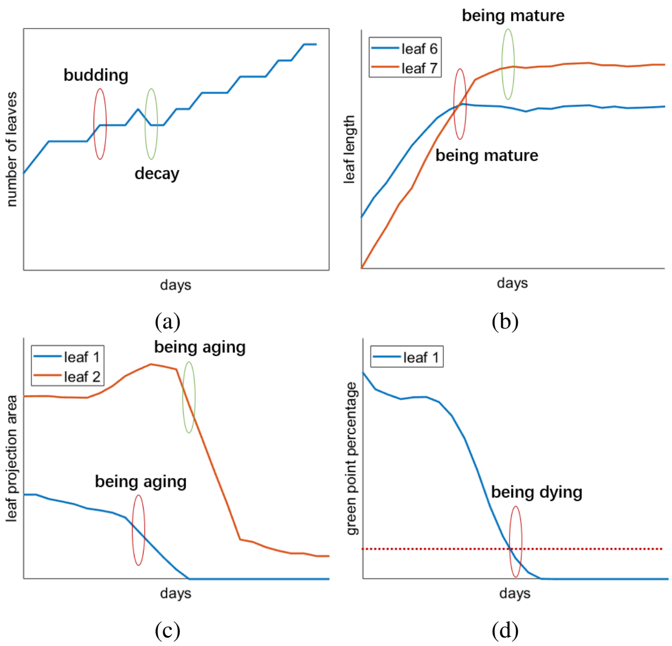


Fig. 10: Traits of some leaves from one plant and the corresponding events. (a) The number of leaves. (b) The leaf length. (c) The projection area. (d) The green point percentage of a leaf. The red dotted line illustrates the threshold.

Then we define some key events in the process of plant growth and detect them based on these traits. There are five events we are interested in: Budding, Decay, Being mature, Being aging and Being dying.

1) Budding and Decay: We detect the budding and decay of leaves by counting the number of leaves. If the number increases, there should be a new budding leaf. On the other hand, if the number decreases, there should be a decaying leaf. Fig. 7 and Fig. 10(a) show the new budding leaf detected by counting leaf number.

2) Being Mature: The event of being mature is related to the length of a leaf. When a leaf is young, it keeps growing, and its length keeps increasing. After a period of time, the leaf is mature, and its length becomes stable. So the event of being mature is detected if the increased length compared to the last day is close to zero. Fig. 11 and Fig. 10(b) show an example of being mature.

3) Being Aging: To detect the event of being aging, we analyze the trait of leaf projection area. It is observed that, when a leaf is getting old, it will bend toward the ground due to gravity. As a result, the corresponding projected area will decrease continuously. So the event of being aging is detected if the leaf projection area is decreasing by 10% for two days. Fig. 12 and Fig. 10(c) illustrate an example of being aging.

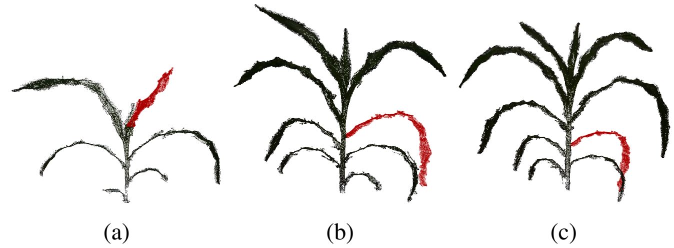


Fig. 11: An example of a mature leaf. The leaf marked red is the same one in different stage. (a) The leaf at immature stage. (b) The leaf at mature stage. The leaf length has increased compared to (a). (c) The leaf remaining mature. The leaf length remain the same compared to (b).

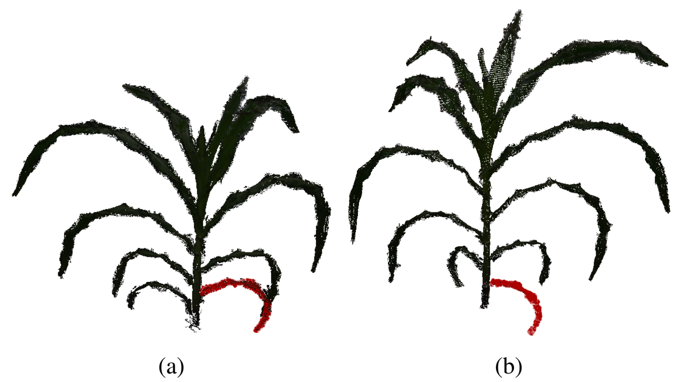


Fig. 12: An example of an aging leaf. The leaf marked red is the same one in different stage. (a) A normal leaf. (b) A aging leaf bending to the ground. The projected area has decreased compared to (b).

4) Being Dying: One of the main changes of the leaf, when it is dying, is that the color will become yellow. So the event of being aging is detected if the percentage of green points is lower than a threshold. In this work, the threshold is set to 15%. An example of being dying is illustrated in Fig. 13 and Fig. 10(d).

The events of leaves from one of the plants are summarized in Table II.

A. Discussion

In this section, we compared our methods with some existing methods. Li et al. [2] proposed a method to analyze the event of growing plants based on point cloud generated by structured light. They successfully detect events such as budding, bifurcation and decay on Anthurium, Dishlia and Dancing bean. However, all the events they found are in the whole-plant level. Compared to their results, we can go further into individual leaves and find more detailed events. Brichet et al. [20] proposed a pipeline for tracking the growth of detailed

(a) (b)

Fig. 13: An example of a dying leaf. (a) A normal leaf. (b) A dying leaf. The color is becoming yellow.

Date	Leaf Index	Event
day 2	leaf 7	budding
day 3	leaf 8	budding
day 6	leaf 6	being mature
day 7	leaf 1	being aging
day 7	leaf 9	budding
day 10	leaf 7	being mature
day 10	leaf 10	budding
day 11	leaf 1	being dying
day 12	leaf 1	decay
day 13	leaf 11	budding
day 14	leaf 8	being mature
day 14	leaf 2	being aging
day 15	leaf 12	budding
day 16	leaf 2	being dying
day 17	leaf 9	being mature
day 18	leaf 13	budding
day 21	leaf 10	being mature
day 21	leaf 14	budding
day 21	leaf 3	being aging
day 23	leaf 15	budding

TABLE II: Event summary

parts of maize, such as ear and silks. Though they calculate traits such as pixel count and stem width, their analysis may not be accurate since the data they are using is 2D images. As we know, 2D images are projections of 3D objects, and some of the geometric information will be lost in the process of projection. Compared to their results, we can calculate accurate traits directly based on 3D models.

B. Limitation

One of the limitations of our pipeline is that all our plants are grown indoors, and all the experiments are conducted in an imaging room. Though our imaging system is designed to be portable, it is not suitable for the plants living in the field since it is required that the plants stay still in the process of imaging. If we conduct the experiments in open space, there will be inevitable disturbances such as wind. As a result, the generated point clouds may be too noisy and cannot lead to a reasonable conclusion. Moreover, the changing sunlight may also be a problem resulting in the incomparable color of leaves.

Another limitation is that the reconstruction method, MVE, is designed to reconstruct 3D point clouds of general objects. It does not take into account any prior information about the reconstructed object. If the shape of the plants can be considered in the process of reconstruction, we may improve both efficiency and accuracy.

V. Conclusion

In this paper, we present an end-to-end pipeline to detect events of plants from 3D point clouds. After capturing images of the plants with our low-cost imaging platform, we obtain the time-varying 3D point clouds using image-based 3D model reconstruction. By filtering out the irrelevant parts in the point cloud, we generate the point clouds of all the separated leaves by detecting and segmenting the stem. Then, we calculate traits from the point clouds of either the whole plant or the individual leaves. Finally, we detect several key events from these traits.

With the proposed method, accurate point clouds of plants and skeletons of leaves can be derived. Our experiment over a period of 4 weeks shows that our pipeline has the ability to detect the events with time-varying data accurately. Therefore, our methods provide an efficient solution for plant scientists to process and analyze time-varying phenotyping data. The same pipeline can also be utilized to measure the effect of the stress environment or different genes on phenotyping.

For future work, we plan to improve our imaging system and conduct experiments on plants in the field by considering various ambient conditions such as wind and changing sunlight.

Acknowledgment

This work has been supported by the National Science Foundation through grants DBI-1564621, OIA-1736192, and IIS-1423487. The authors are grateful to the staff members at the University of Nebraska-Lincoln's Greenhouse Innovation Center for their support in the data collection process.

References

[1] 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*, vol. 26, no. 1, pp. 5–20, 2014.

[2] 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)*, vol. 32, no. 6, p. 157, 2013.

[3] C. Klukas, D. Chen, and J.-M. Pape, "Integrated analysis platform: an open-source information system for high-throughput plant phenotyping," *Plant physiology*, vol. 165, no. 2, pp. 506–518, 2014.

[4] F. Zhu, S. Thapa, T. Gao, Y. Ge, H. Walia, and H. Yu, "3D reconstruction of plant leaves for high-throughput phenotyping," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4285–4293.

[5] R. F. McCormick, S. K. Truong, and J. E. Mullet, "3D sorghum reconstructions from depth images identify qtl regulating shoot architecture," *Plant physiology*, vol. 172, no. 2, pp. 823–834, 2016.

[6] A. Chaudhury and J. L. Barron, "Machine vision system for 3D plant phenotyping," *IEEE/ACM transactions on computational biology and bioinformatics*, 2018.

[7] T. Nguyen, D. Slaughter, N. Max, J. Maloof, and N. Sinha, "Structured light-based 3D reconstruction system for plants," *Sensors*, vol. 15, no. 8, pp. 18 587–18 612, 2015.

[8] E. Puttonen, M. Lehtomaki, P. Litkey, R. Nasi, Z. Feng, X. Liang, S. Wittke, M. Pandzic, T. Hakala, M. Karjalainen, and N. Pfeifer, "A clustering framework for monitoring circadian rhythm in structural dynamics in plants from terrestrial laser scanning time series," *Front Plant Sci*, vol. 10, p. 486, 2019.

[9] 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 (Basel)*, vol. 18, no. 4, 2018.

[10] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[11] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 65:1–65:8, 2013.

[12] S. Fuhrmann and M. Goesele, "Floating scale surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, p. 46, 2014.

[13] Y. Bao, L. Tang, M. W. Breitzman, M. G. Salas Fernandez, and P. S. Schnable, "Field-based robotic phenotyping of sorghum plant architecture using stereo vision," *Journal of Field Robotics*, vol. 36, no. 2, pp. 397–415, 2019.

[14] F. Rovira-Más, Q. Zhang, and J. Reid, "Creation of three-dimensional crop maps based on aerial stereoimages," *Biosystems Engineering*, vol. 90, no. 3, pp. 251–259, 2005.

[15] L. Mundermann, Y. Erasmus, B. Lane, E. Coen, and P. Prusinkiewicz, "Quantitative modeling of arabidopsis development," *Plant Physiol*, vol. 139, no. 2, pp. 960–8, 2005.

[16] Y. Zhou, S. Srinivasan, S. V. Mirnezami, A. Kusmec, Q. Fu, L. Attigala, M. G. Salas Fernandez, B. Ganapathysubramanian, and P. S. Schnable, "Semiautomated feature extraction from rgb images for sorghum panicle architecture gwas," *Plant Physiol*, vol. 179, no. 1, pp. 24–37, 2019.

[17] C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall, *Modeling and generating moving trees from video*. ACM, 2011, vol. 30, no. 6.

[18] R. Fernandez, P. Das, V. Mirabet, E. Moscardi, J. Traas, J. L. Verdeil, G. Malandain, and C. Godin, "Imaging plant growth in 4d: robust tissue reconstruction and lineaging at cell resolution," *Nature Methods*, vol. 7, no. 7, pp. 547–U94, 2010.

[19] C. Klukas, D. Chen, and J. M. Pape, "Integrated analysis platform: An open-source information system for high-throughput plant phenotyping," *Plant Physiol*, vol. 165, no. 2, pp. 506–518, 2014.

[20] N. Brichet, C. Fournier, O. Turc, O. Strauss, S. Artzet, C. Pradal, C. Welcker, F. Tardieu, and L. Cabrera-Bosquet, "A robot-assisted imaging pipeline for tracking the growths of maize ear and silks in a high-throughput phenotyping platform," *Plant Methods*, vol. 13, no. 1, p. 96, 2017.

[21] X. Zhang, C. Huang, D. Wu, F. Qiao, W. Li, L. Duan, K. Wang, Y. Xiao, G. Chen, Q. Liu, L. Xiong, W. Yang, and J. Yan, "High-throughput phenotyping and qtl mapping reveals the genetic architecture of maize plant growth," *Plant Physiol*, vol. 173, no. 3, pp. 1554–1564, 2017.

[22] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 71.

[23] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," in *ACM transactions on graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 44.

[24] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 778–785.

[25] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2847–2854.

[26] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*. Ieee, 2011, pp. 1297–1304.

[27] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, "MVE-an image-based reconstruction environment," *Computers & Graphics*, vol. 53, pp. 44–53, 2015.

[28] A. A. Gitelson, Y. J. Kaufman, R. Stark, and D. Rundquist, "Novel algorithms for remote estimation of vegetation fraction," *Remote sensing of Environment*, vol. 80, no. 1, pp. 76–87, 2002.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[30] H. Medeiros, D. Kim, J. Sun, H. Seshadri, S. A. Akbar, N. M. Elfiky, and J. Park, "Modeling dormant fruit trees for agricultural automation," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1203–1224, 2017.

[31] N. M. Elfiky, S. A. Akbar, J. Sun, J. Park, and A. Kak, "Automation of dormant pruning in specialty crop production: An adaptive framework for automatic reconstruction and modeling of apple trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 65–73.

[32] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via Laplacian based contraction," in *2010 Shape Modeling International Conference*. IEEE, 2010, pp. 187–197.

[33] Z. Su, Y. Zhao, C. Zhao, X. Guo, and Z. Li, "Skeleton extraction for tree models," *Mathematical and Computer Modelling*, vol. 54, no. 3-4, pp. 1115–1120, 2011.