2018

3D Perception-based Collision-Free Robotic Leaf Probing for Automated Indoor Plant Phenotyping

Yin Bao
Iowa State University, yinbao@iastate.edu

Dylan Shah
Iowa State University

Lie Tang
Iowa State University, lietang@iastate.edu

3D Perception-based Collision-Free Robotic Leaf Probing for Automated Indoor Plant Phenotyping

Abstract

Various instrumentation devices for plant physiology study such as spectrometer, chlorophyll fluorimeter, and Raman spectroscopy sensor require accurate placement of their sensor probes toward the leaf surface to meet specific requirements of probe-to-target distance and orientation. In this work, a Kinect V2 sensor, a high-precision 2D laser profilometer, and a six-axis robotic manipulator were used to automate the leaf probing task. The relatively wide field of view and high resolution of Kinect V2 allowed rapid capture of the full 3D environment in front of the robot. The location and size of each plant were estimated by k-means clustering where "k" was the user-defined number of plants. A real-time collision-free motion planning framework based on Probabilistic Roadmaps was adapted to maneuver the robotic manipulator without colliding with the plants. Each plant was scanned from the top with the short-range profilometer to obtain high-precision 3D point cloud data. Potential leaf clusters were extracted by a 3D region growing segmentation scheme. Each leaf segment was further partitioned into small patches by a Voxel Cloud Connectivity Segmentation method. Only the patches with low root mean square errors of plane fitting were used to compute leaf probing poses of the robot. Experiments conducted inside a growth chamber mock-up showed that the developed robotic leaf probing system achieved an average motion planning time of 0.4 seconds with an average end-effector travel distance of 1.0 meter. To examine the probing accuracy, a square surface was scanned at different angles, and its centroid was probed perpendicularly. The average absolute probing errors of distance and angle were 1.5 mm and 0.84 degrees, respectively. These results demonstrate the utility of the proposed robotic leaf probing system for automated non-contact deployment of spectroscopic sensor probes for indoor plant phenotyping under controlled environmental conditions.

Keywords

Plant phenotyping, 3D perception, Agricultural robotics, Leaf probing, Motion planning

Disciplines

Bioresource and Agricultural Engineering | Computer-Aided Engineering and Design | Ergonomics | Operational Research

Comments

This is a manuscript of the article Bao, Yin, Dylan S. Shah, and Lie Tang. "3D Perception-based Collision-Free Robotic Leaf Probing for Automated Indoor Plant Phenotyping." *Transactions of the ASABE (in press)* (2018). DOI: 10.13031/trans.12653. Posted with permission.

3D PERCEPTION-BASED COLLISION-FREE ROBOTIC LEAF PROBING FOR AUTOMATED INDOOR PLANT PHENOTYPING

Y. Bao, D. Shah, L. Tang

The authors are **Yin Bao, ASABE Member,** Graduate Student, and **Dylan Shah,** Graduate Student, and **Lie Tang**, **ASABE Member**, Associate Professor, Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, IA, USA. **Corresponding author:** Lie Tang, 2346 Elings Hall, 605 Bissell Road, Ag and Biosystems Engineering, Iowa State University, Ames, IA 50011; phone: 515-294-9778; e-mail: lietang@iastate.edu.

Presented at the 2017 ASABE Annual Meeting as Paper No. 1700369

ABSTRACT.

Various instrumentation devices for plant physiology study such as spectrometer, chlorophyll fluorimeter, and Raman spectroscopy sensor require accurate placement of their sensor probes toward the leaf surface to meet specific requirements of probe-to-target distance and orientation. In this work, a Kinect V2 sensor, a high-precision 2D laser profilometer, and a six-axis robotic manipulator were used to automate the leaf probing task. The relatively wide field of view and high resolution of Kinect V2 allowed rapid capture of the full 3D environment in front of the robot. The location and size of each plant were estimated by k-means clustering where "k" was the user-defined number of plants. A real-time collision-free motion planning framework based on Probabilistic Roadmaps was adapted to maneuver the robotic manipulator without colliding with the plants. Each plant was scanned from the top with the short-range profilometer to obtain high-precision 3D point cloud data. Potential leaf clusters were extracted by a 3D region growing segmentation scheme. Each leaf segment was further partitioned into small patches by a Voxel Cloud Connectivity Segmentation method. Only the patches with low root mean square errors of plane fitting were used to compute leaf probing poses of the robot. Experiments conducted inside a growth chamber mock-up showed that the developed robotic leaf probing system achieved an average motion planning time of 0.4 seconds with an average end-effector travel distance of 1.0 meter. To examine the probing accuracy, a square surface was scanned at different angles, and its centroid was probed perpendicularly. The average absolute probing errors of distance and angle were 1.5 mm and 0.84 degrees, respectively. These results demonstrate the utility of the proposed robotic leaf probing system for automated non-contact deployment of spectroscopic sensor probes for indoor plant phenotyping under controlled environmental conditions.

Keywords.

Plant phenotyping, 3D perception, Agricultural robotics, Leaf probing, Motion planning

Agricultural robotics is undergoing rapid growth propelled by the need for automated plant phenotyping. Many existing automated phenotyping systems focus on imaging either the whole plant or the canopy using various cameras such as RGB, hyperspectral, and thermal cameras. Moreover, plant 3D reconstruction has been conducted with different range sensors including stereo cameras, Time-of-Flight of light sensors, LiDAR sensors, and Computed Tomography sensors (Hartmann et al., 2011; Araus and Cairns, 2014; Li et al., 2017). However, few systems address automatic instrumentation of plant activities and plant composition at the molecular level for plant physiology research. Chlorophyll fluorometer is widely used to measure chlorophyll fluorescence which indicates the rate of photosynthesis and the level of plant stress. Raman spectrometer can be used to identify and quantify valuable plant substances (Schulz and Baranska, 2007). The portable models of these devices require positioning a fiber optic probe above plant leaves at a close distance and with a precise angle. For example, MINI-PAM-II from WALZ (Effeltrich, Germany) measures fluorescence with a fiber-optic probe. The standard leaf clip has a 60-degree angle between the sample plane and the probe. The distance between them is adjustable but typically is set at 8 mm. Agility Raman Spectrometer from BaySpec (California, USA) also uses a fiber-optic probe. The probing angle is required to be 90 degrees, and the distance between the sample and the probe tip is set up based on the focal lengths of the interchangeable lenses in the probe (e.g., 6 mm and 15 mm). Automated leaf probing for the above instruments is highly challenging because of two main reasons: First, it needs to detect individual plant leaves in a complex scene with partially occluded leaves. Different species may have different leaf shapes, and even within the same species plant leaves grow and may curl at some point. Second, a robotic manipulator with at least five degrees of freedom (DOF) is necessary to place the probe with respect to the leaf surface with a required pose (position and orientation). Motion planning of each robot joint should be carried out to avoid any collision with the plants while bringing the probe in position.

Up until now robotic leaf probing research has mostly focused on leaf segmentation. Alenya et al. (2013) developed an eye-in-hand robotized leaf probing system using a multi-axis robot arm and a Time-of-Flight (ToF) sensor. The word “eye-in-hand” refers to equipping the end effector of a robotic manipulator with a machine vision system for guidance. A segmentation algorithm was developed to extract the single leaf segment in front of the ToF sensor using both the intensity and depth images. A grasping point was predetermined on a leaf contour model. By fitting the model to the segmented leaf, the grasping point was found. The algorithm was computationally expensive and the processing time was 25 s per image. Foix et al. (2015)

developed an active exploration process to incrementally gain information about the leaf in the view until probing could be performed. A depth sensor was fixed above the plant to acquire the top-view image of the whole plant for motion planning. Ahlin et al. (2016) used a deep convolutional neural network to detect plant leaves in the images acquired by an RGB camera mounted on the end effector of a robotic manipulator. Sparse feature points between frames were used to compute the 3D location of the leaves for guiding the end effector. However, this work only addressed how to bring the end effector close to the leaf.

Previous related work focused on probing a single plant and had little emphasis on collision-free motion planning (Alenya et al., 2013; Foix et al., 2015). However, the typical scenario for plant science research is that multiple plants grow together under a specific environmental condition, thus, developing a collision-free motion planning algorithm is indispensable for a robotic arm to execute leaf probing tasks in a cluttered environment where plants and other obstacles exist. The technical challenges in automated leaf probing include plant detection and localization, individual leaf segmentation, and precise leaf probing without disturbing other parts of the plants. This study proposes a robotic leaf probing pipeline using an eye-in-hand system including a Kinect V2 sensor, a high-precision 2D laser profilometer, and a six-DOF robotic manipulator (fig. 1). The primary objective of this study was to evaluate the performance of the proposed robotic leaf probing system which combines 3D plant perception and collision-free motion planning capabilities. Specifically, the system accuracy was evaluated in terms of the probing position and probing angle of the robot end effector to a square surface; motion planning performance was evaluated in connection with the planning time and the travel distance of the robot arm for leaf probing with multiple plants in a growth chamber mock-up.

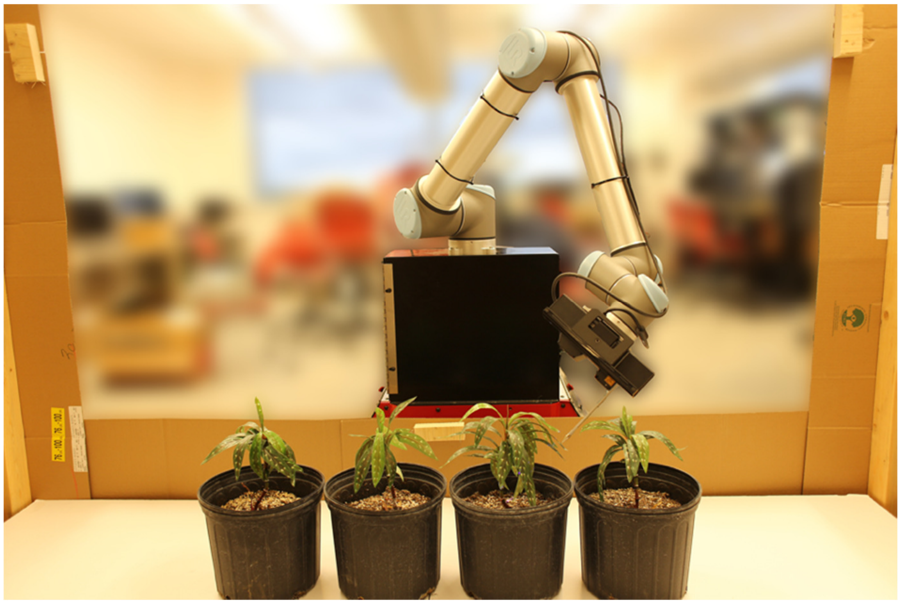


Figure 1. Robotic leaf probing of artificial plants in a growth chamber mock-up. The system includes a six-axis UR10 robotic manipulator, a Kinect V2 sensor, a Keyence LJ-V7300 2D in-line laser profilometer, and a probe.

MATERIALS AND METHODS

THE ROBOT AND THE SENSORS

The robot consisted of three modules: an unmanned ground vehicle (UGV), a six-axis robotic manipulator, and a sensing unit (Shah and Tang, 2016; Shah, Tang, et al., 2016). The UGV was custom-built with omnidirectional Mecanum wheels and magnetic guide sensors. The robotic manipulator is a Universal Robots UR10 (Odense, Denmark) with a reach radius of 1.3 m, a repeatability of ±0.1 mm, and a maximum tool speed of 1 m/s. The sensing unit includes a Kinect V2 sensor (Microsoft Inc., Washington, USA) and a Keyence LJ-V7300 2D in-line laser profilometer (Osaka, Japan). The Kinect V2 sensor outputs depth images with a resolution of 512×424 and a field of view (FOV) of 70×60 degrees at 30 frames per second. The distance measurement of Kinect V2 could deviate from the true distance up to approximately 0.03 m within a range of 0.4 m to 6 m (Lachat et al., 2015). Despite its relatively low accuracy in depth sensing, Kinect V2 sensor was successfully used for many applications, especially in real-time 3D modeling of natural scenes (Newcombe et al., 2011). Its long sensing range and wide FOV make it suitable for a rapid 3D mapping of a complete growth chamber environment where the plant locations and sizes are unknown. On the other hand, the laser profilometer has a relatively short measurement range of 0.155 m to 0.445 m but a high repeatability of 0.005 mm for depth measurement. The scan line widths on the near side and the far side are 110 mm and 240 mm, respectively, with a spatial resolution of 800 pixels. This type of sensor is often used for surface profile inspection in manufacturing industry. Hence, it is suitable for reconstructing accurate 3D plant models in this application.

ROBOTIC LEAF PROBING PIPELINE OVERVIEW

By using the Kinect's rough but rapid chamber-level environment mapping, the profilometer's high-resolution distance sensing capability, and the precise motion control of the robotic manipulator, a pipeline of accurate leaf probing can be developed (fig. 2). The 3D point cloud from the Kinect V2 can be used to locate each plant and estimate its size. In addition, the same point cloud defines the occupied workspace which the robot should avoid. Despite its low cost and its efficiency in environment mapping, the depth sensing of Kinect V2 tends to be inaccurate and noisy. However, robotic leaf probing relies on accurate depth sensing and surface normal estimation to compute the required position and angle between a leaf surface patch and the probe. Therefore, the short-range high-precision 2D laser profilometer is used to scan each plant from the top with a sweeping motion of the end effector. Piece-wise smooth point clusters are extracted with 3D region growing segmentation to obtain leaf segments which are large parts of plant leaves. Furthermore, each leaf segment is partitioned into spatially connected small point clusters, also known as supervoxels. The probing poses (position and orientation) of the end effector are only computed for the supervoxels with low root mean square (RMS) errors of plane fitting. Collision-free motion planning in six-dimensional configuration space (C-space or joint space) is performed to search for a sequence of robot configurations which would bring the probe to the desired pose without colliding with the plants. Such motion planning in high dimensional space is computationally expensive. Minimum robot operation time means minimum disturbance to plant growth. Therefore, a real-time motion planning framework is used. It pre-builds a reusable data structure that links the cells in a 3D grid to random configurations in C-space and requires minimum computation during online planning.

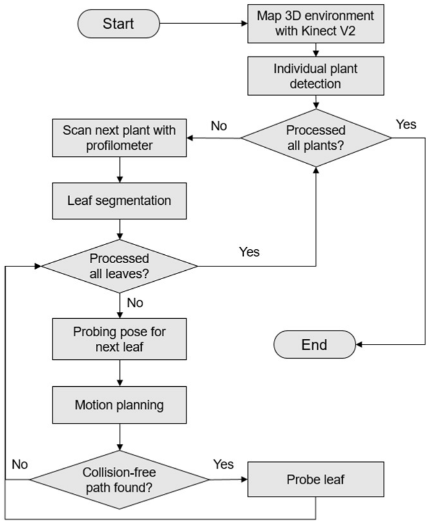

Start
Map 3D environment with Kinect V2
Individual plant detection
Processed all plants?
No
Yes
Scan next plant with profilometer
Leaf segmentation
Processed all leaves?
Yes
No
Probing pose for next leaf
End
Motion planning
Collision-free path found?
No
Yes
Probe leaf

Figure 2. The robotic leaf probing pipeline.

REAL-TIME COLLISION-FREE MOTION PLANNING

Motion planning plays a critical role in robotic leaf probing. In the cluttered growth chamber environment, not all plant leaves can be probed by the manipulator. Even for the reachable leaves, finding a collision-free path is a challenge. Fortunately, motion planning for high-degree-of-freedom robots has been well studied. The current state-of-the-art systems are built upon sampling-based algorithms such as Probabilistic Roadmap (Kavraki et al., 1996) and Rapidly-Exploring Random Tree (LaValle, 1998). The basic idea is to explore the configuration space by random sampling and connect neighboring configurations if the connections do not cause collisions in the robot's 3D workspace. The resultant graph is searched to find the shortest path between a start configuration and a goal configuration. Leven and Hutchinson (2002) proposed a real-time motion planning framework for changing environments based on Probabilistic Roadmap (PRMCE). Offline, it builds the roadmap, an undirected weighted graph in which a node represents a random self-collision-free configuration in C-space and an edge represents a self-collision-free path weighted by the swept volume between its two connecting configurations (left side of fig. 3). Then a 3D occupancy grid is constructed (right side of fig. 3). The workspace surrounding the robotic manipulator is tessellated into cube cells. Each cell stores a binary variable to indicate whether the cell is occupied by obstacles. In addition to occupancy state, each cell stores a list of all the edges whose swept volume is in collision with the cell. Online (during operation), if a cell is detected to be occupied with a depth sensor, the associated edges are temporarily blocked until the cell becomes free space later. In this way, any path on the roadmap is always collision-free. Constructing the roadmap and occupancy grid may require heavy computation due to a large amount of collision checking involved, but it only has to be done once offline for a given robotic manipulator. Consequently, the online motion planning mainly involves a graph search which takes relatively little time. Kunz et al. (2010) improved PRMCE for a seven-DOF robotic manipulator and used a fixed depth camera to update the 3D occupancy grid in real time. In their application, less than 100 ms was required for obstacle detection and path planning combined on a single core CPU clocked at 3 GHz. More recently, Murray et al. (2016) exploited the parallelism in the collision checking of PRMCE and implemented it on a field-programmable gate array (FPGA), achieving motion planning on a chip for less than 1 ms. All the advances proved the practicality of PRMCE. Therefore, the PRMCE algorithm was implemented for the robotic leaf probing system.

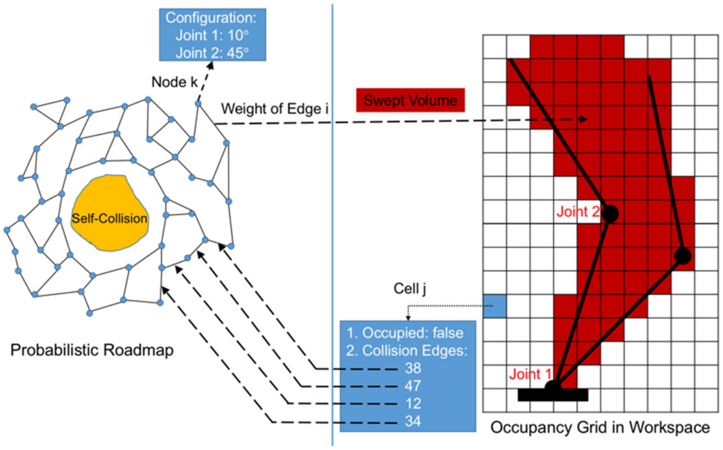


Figure 3. Probabilistic Roadmap in Changing Environments (PRMCE). Left: Probabilistic Roadmap, where each node represents a random self-collision-free configuration while each edge indicates a self-collision-free path weighed by the swept volume between its two connecting configurations. Right: Occupancy Grid in the workspace of a two-axis robotic manipulator, where each cell stores the occupancy state and the list of edges whose swept volume collided with the cell.

Robot Modeling and Self-Collision Checking

Each robot component (probe, sensors, robot arms, and mobile base) was modeled as an oriented bounding box (OBB) to perform collision checking efficiently (fig. 4). During the roadmap construction, the randomly generated joint positions might result in self-collision. Based on the separating axis theorem, collision checking of two OBBs came down to at most 15 non-intersection tests (Eberly, 2002). If each movable robot component did not collide with any other component (e.g., the first arm and the second arm), the configuration was free of self-collision. Because the robot was always situated at a location with respect to the growth chamber, some parts of the growth chamber were incorporated into the robot model, and thus self-collision checking also examined collision with the growth chamber (fig. 4).

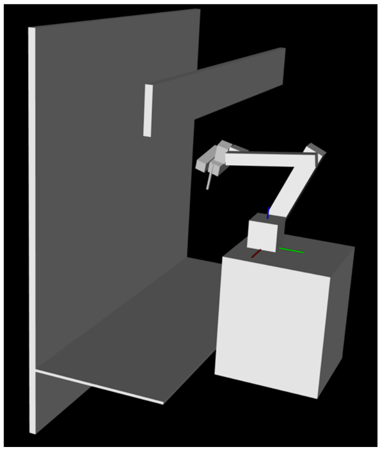
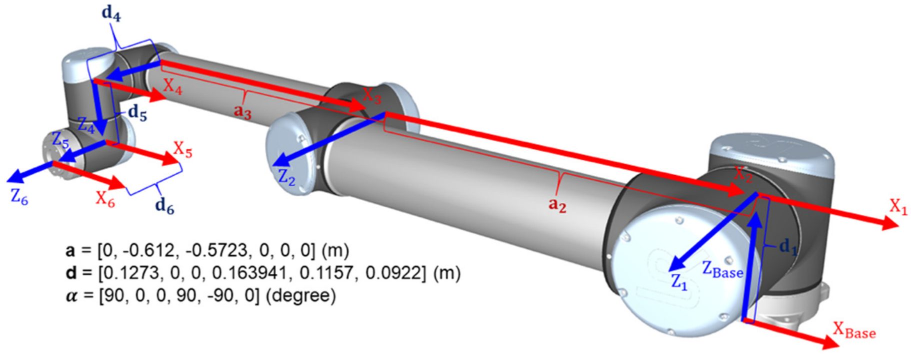


Figure 4. Robot and growth chamber modeling with oriented bounding boxes.

Roadmap Generation

The roadmap was an undirected weighted graph. First, N random self-collision-free configurations were generated as the nodes. Each of the six joint positions was randomly sampled from a uniform distribution. Because the robotic manipulator typically moved inside the growth chamber, each joint was limited to a specific range with the maximum range less than 180 degrees. A range of 45 degrees was allowed for the last joint (end effector) to prevent twisting the sensor cables. Second, edges were created to establish connectivity in the graph. Each node was connected to at most *K* neighbors if the edge did not result in self-collision. The weight of an edge represented the distance between the two connected configurations. The distance was measured by the Euclidian norm of *M* reference points on the robotic manipulator. The reference points were the origins of the reference frames of the robotic manipulator. The reference frames followed the commonly used Denavit-Hartenberg convention (Denavit, 1955; Hartnberg and Denavit, 1964) and the parameters were provided by Universal Robots (fig. 5). Therefore, the distance metric measures the overall displacement of all joints in the 3D workspace.



a = [0, -0.612, -0.5723, 0, 0, 0] (m)
d = [0.1273, 0, 0, 0.163941, 0.1157, 0.0922] (m)
α = [90, 0, 0, 90, -90, 0] (degree)

Figure 5. Reference frames defined by the Denavit-Hartenberg convention for UR10.

Mapping Workspace to Roadmap

Swept volume was represented as a set of cells in the 3D occupancy grid. The volume of a single configuration was computed by voxelizing the OBBs of the robotic manipulator and sensors in the 3D occupancy grid. The OBB voxelization was based on 3D line voxelization (Cohen-Or and Kaufman, 1997). Two parallel edges of a rectangle face on an OBB were voxelized, and then the two edges were used to rasterize the rectangle face. Finally, the rectangle face was extruded to the size of the OBB, and the set of cells representing the volume of the OBB was obtained. The swept volume between two configurations was obtained by voxelizing a series of configurations in the path. The process started with voxelizing the two end configurations while keeping track of which cells had been voxelized. Then the halfway configuration was voxelized, and the number of newly voxelized cells was counted. This recursive bisection voxelization was continued until no new cell was found (Leven and Hutchinson, 2002).

An edge in the roadmap corresponded to the swept volume between two configurations. For mapping the workspace to the roadmap, every cell in the 3D occupancy grid needed to maintain a list of edges whose associated volume contained the cell. The lists were updated during the voxelization of the swept volume of all the edges in the roadmap. Hence, the lists were completed once all the edges were voxelized.

Online Planning

During the online operation stage, the point cloud data from both the Kinect V2 and the 2D laser profilometer were used to update the 3D occupancy grid. If a cell was occupied, the weights of the edges stored in the cell were temporally changed to infinity until the cell became unoccupied by reset. The collision-free motion planner started with a collision test for a direct path between a start configuration and a goal configuration because the direct path had the absolute shortest distance. Collision checking required voxelization of the swept volume described in the previous section. During the process, collision was detected if any cell was found occupied. If the direct path failed, the two configurations were connected to the roadmap via nearest neighbor search and collision checking. The A* graph search algorithm (Hart et al., 1968) was used to search the path due to its optimal performance in pathfinding. If the length of the path was not infinite, a collision-free path was considered found. The path consisted of a sequence of configurations. Because the configurations were randomly generated, the robotic manipulator might not undergo a smooth motion. Hence the unnecessary nodes in the path were removed to reduce the travel distance and smooth the trajectory. Specifically, the path between node n^i and node n^{i+2} was examined for collision. If the path was collision-free, node n^{i+1} was deleted and then the process is repeated until 1) node n^{i+2} was or beyond the last node after a deletion or 2) collision was detected between node n^i and node n^j $(j > i)$. If node n^j was not the last node, the trimming process restarted at node n^j.

3D MAPPING IN A GROWTH CHAMBER

The 3D environment in a growth chamber could be quickly mapped with the Kinect V2 installed on the end effector of the robot arm. Several poses above the plants could be manually selected as long as all plants could be imaged in the bird's eye view. The distance between the plants and the Kinect V2 needed to be at least 0.5 m (the minimum working distance of Kinect V2). In our experiment, a distance of 1.3 m was used, and all the plants could be imaged with three poses. A typical issue with ToF sensors is flying pixels, which are pixels that observe depth discontinuities at object boundaries (Kolb et al., 2009). As a result, the depth measurements of flying pixels were between the foreground depth and background depth, causing free space to be wrongly labeled as occupied space. They can be filtered by thresholding the maximum depth difference in a small local neighborhood (e.g. a 3×3 window) of each pixel in the depth image.

The filtered depth image was converted to a 3D point cloud in the camera coordinate system using the Kinect software development kit. Then the point cloud was transformed back to the base coordinate system of the robotic manipulator based on $\boldsymbol{p}_{base} = T_{base2hand}T_{hand2cam}\boldsymbol{p}_{cam}$, where $\boldsymbol{p}_{cam}$ and $\boldsymbol{p}_{base}$ denoted the 4×1 homogeneous coordinates of a point in the camera coordinate system and in the robot base coordinate system, respectively; and $T_{hand2cam}$ and $T_{base2hand}$ were the 4×4 homogeneous transformation matrices from end-effector pose to camera pose and from robot base to end-effector pose, respectively. $T_{base2hand}$ came from the forward kinematics of a given configuration and the Denavit-Hartenberg parameters of the robotic manipulator. $T_{hand2cam}$ was estimated via hand-eye calibration. Because hand-eye calibration is well developed for regular 2D cameras, a classic 2D approach was used with the intensity image of the ToF sensor instead of the depth image. $T_{hand2cam}$ was solved by the least-squares solution proposed by Park et al. (1994). After the multi-view registration, the point cloud did not have a uniform point density. There would be more points in the overlapping regions. To reduce unnecessary computation for the rest of the processing pipeline, the point cloud was downsampled and regularized with a VoxelGrid filter (Rusu, 2010).

INDIVIDUAL PLANT EXTRACTION AND SCANNING

Given the known geometry and dimensions of the growth chamber, the plants including the pots were cropped out of the 3D point cloud of the growth chamber environment by thresholding the ranges of X, Y, and Z coordinates. The next step was to resolve the location and size of each plant and determine the scanning trajectory of the 2D laser profilometer. Because the plants/pots are placed at different locations in a growth chamber, and the quantity is typically fixed from the beginning of an experiment, k-means clustering was used to partition the 3D points into clusters representing individual plants based on their X, Y, and Z coordinates. Here, "k" was the user-defined number of plants. Finally, an axis-aligned bounding box was extracted for each cluster, and the top face defined the scanning region.

A 2D laser profilometer works as follows. It emits a thin laser sheet and detects the laser line on an object's surface with a 2D camera. The depth of any pixel on the laser line is calculated via triangulation. To generate the 3D model of a plant, the profilometer was moved translationally by the robotic manipulator along a direction perpendicular to the laser sheet to cover the surfaces of a plant canopy. Assuming that the profilometer started scanning at the same time as the end effector began acceleration, the time when line profile number n was being scanned was calculated as $t = (n-1)/f$, where f was a constant scan frequency. The translational motion of the end effector had a trapezoidal speed profile with three phases: acceleration, constant speed, and deceleration. Then the 3×1 translation vector $\overrightarrow{V_n}$ of the profilometer at scan n was calculated as

$$\overrightarrow{V_n} = \begin{cases} \frac{1}{2}at^2\,\overrightarrow{V_e}, & t < t_a \\ v\left(t - \frac{t_a}{2}\right)\overrightarrow{V_e}, & t_a \le t \le t_d \\ \left\{v(t - t_d) - \frac{d(t-t_d)^2}{2}\right\}\overrightarrow{V_e}, & t_d < t \end{cases}, \tag{1}$$

where

$a =$ acceleration constant

d = deceleration constant

$t_a =$ time when acceleration ended

t_d = time when deceleration started

v = velocity constant

$\overrightarrow{V_e}$ = 3×1 normalized translation vector of the end effector

A point $\boldsymbol{q}_{scan}^{n}$ in scan n was transformed to the point $\boldsymbol{p}_{base}^{n}$ in the robot base frame based on

$$\boldsymbol{p}_{base}^{n} = T_{base2hand}T_{hand2scan}\begin{pmatrix}\boldsymbol{q}_{scan}^{n} + \overrightarrow{V_n} \\ 1\end{pmatrix}, \tag{2}$$

where $T_{hand2scan}$ denotes the transformation matrix from end effector to 2D laser profilometer. $T_{hand2scan}$ was determined via hand-eye calibration. Note that $\boldsymbol{q}_{scan}^{n}$ is not in homogeneous coordinates. Carlson et al. (2015) proposed a simple and robust method for hand-eye calibration between a 2D laser profilometer and a six-DOF robotic manipulator using planar constraints. The calibration data were easy to acquire and the procedure was described as follows: (1) Place a planar object such as a piece of flat sheet metal in front of the robot; (2) Collect multiple line scans on the planar surface along with the end-effector poses of the robotic manipulator. Note that the laser lines should not be collinear otherwise it would not be possible to estimate the plane coefficients; (3) Change position and orientation of the planar object and repeat previous two steps at least two times. The calibration data were used in an optimization method. It started with an initial guess of $T_{hand2scan}$ which was measured with a ruler and a protractor. Then it iteratively optimized the rotation matrix and the translation vector in $T_{hand2scan}$ to minimize the sum of the squared distances between the 3D points and the associated planes. The optimization procedure mainly involved some well-developed methods of linear algebra: Principle Component Analysis (Pearson, 1901; Hotelling 1933), Ordinary Least Squares (Lawson and Hanson, 1995), and Singular Value Decomposition (Golub and Reinsch, 1970).

LEAF SEGMENTATION AND EXTRACTION OF PROBING POINTS AND DIRECTIONS

The high-precision 3D scan using the profilometer enabled accurate surface normal estimation. The surface normal was estimated based on Principle Component Analysis (PCA) of the local neighborhood around a 3D point. Specifically, eigenvalue decomposition was performed on the 3×3 covariance matrix for the set of points. The surface normal was then approximated by the normalized eigenvector corresponding to the minor eigenvalue.

As plant leaf surfaces tend to be smooth, neighboring points on the same leaf should have similar surface normal direction and low curvature. The 3D region growing segmentation with smoothness constraint (Rabbani et al., 2006) was used for leaf segmentation. An implementation is available in Point Cloud Library (PCL, Rusu and Cousins, 2011). It starts with a seed defined by the point of minimum curvature and then merges a neighbor if 1) the angle between the seed's normal and the neighbor's normal is less than a smoothness threshold and 2) the plane fitting residual with *K*-nearest neighbors is less than a residual threshold. If a neighbor is merged, it is queued as a new seed. Once all the neighbors of a seed are evaluated, the seed is removed from the queue. The region growing stops when there is no seed left in the queue. The visited points are labeled. Then the same process is repeated on the remaining unlabeled points until all points are assigned to a cluster. In this application, the leaf segmentation aimed to find large smooth parts of leaves instead of complete individual leaves. The number of leaf segments depends on the geometry of the leaf surface. For instance, a soybean leaf is relatively small, flat, and round, hence a leaf segment is likely the whole leaf, whereas a maize leaf is elongated and often twisted, hence multiple leaf segments may be extracted from one leaf. Since the ultimate goal of this application was to find flat and small surface patches on leaves for probing, the parameters of the 3D region growing segmentation should be tuned for different plant species when controlling the smoothness of segmented surfaces so that larger leaf segments could be produced. The tuning process started with large values of the *K*, the residual threshold, and the smoothness threshold, which resulted in no segmentation. Among the three parameters, the smoothness threshold was the most effective one for separating leaves. So the smoothness threshold was first decreased gradually until most of the complete individual leaves were segmented. Then the *K* and the residual threshold were fine-tuned to further separate overlapping leaves with a smooth transition of surface normal.

The segmentation results were further filtered based on their size and position; and the scanned point cloud was downsampled with a VoxelGrid filter so that each voxel in 3D space contained at most one point. Then each segment's surface area was estimated by the product of the number of points in the cluster and the voxel size. Segment position was represented by the cluster centroid. Many locations on a leaf segment could be probed without a collision, especially for elongated leaf shapes of plants such as maize and sorghum. Voxel Cloud Connectivity Segmentation (Papon et al., 2013) was used to partition a leaf segment into several small patches of similar sizes, namely, supervoxels. The potential probing points were based on the supervoxels with low RMS errors of plane fitting. Although leaf surface was smooth, it was more desirable to probe planar patches which offered more accurate surface normal estimates. Supervoxel size was a user-defined parameter to control the surface patch size and was reasonable to set to a value based on the diameter of the probe. Finally, the probing location was chosen to be the point closest to the centroid of the supervoxel and the probing direction is parallel to the normal of the plane fitted to that supervoxel.

Probing Pose Determination

A probing stick was mounted next to the laser profilometer parallel to the Z-axis of robot tool frame (the last reference frame in fig. 5). The probe tip position in robot tool frame was obtained by tool center point (TCP) calibration. A number of robot tool frames were recorded as the robot end effector was moved such that the probe tip stayed at a fixed point in 3D space while the probe body was rotated around that point. Every pair of frame i and frame j forms a linear equation,

$$\left(R_j - R_i\right)\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \vec{T_\iota} - \vec{T_J}, \tag{3}$$

where R and $\vec{T}$ denote the rotation matrix and the translation vector of robot tool frame, respectively. The probe tip position $[X_P, Y_P, Z_P]^T$ can be solved in a least-squares fashion. Figure 6 illustrates one of several robot tool frames recorded for TCP calibration. This was done by carefully guiding the end effector to align the probe tip with the fixed 3D calibration point. This process was facilitated by the teach pendant which is a handheld device to remotely control the robotic manipulator. Given a probing pose $T_{base2probe}$, the robot tool frame was simply calculated as

$$T_{base2tool} = T_{base2probe}\begin{bmatrix} 1 & 0 & 0 & -X_P \\ 0 & 1 & 0 & -Y_P \\ 0 & 0 & 1 & -Z_P \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

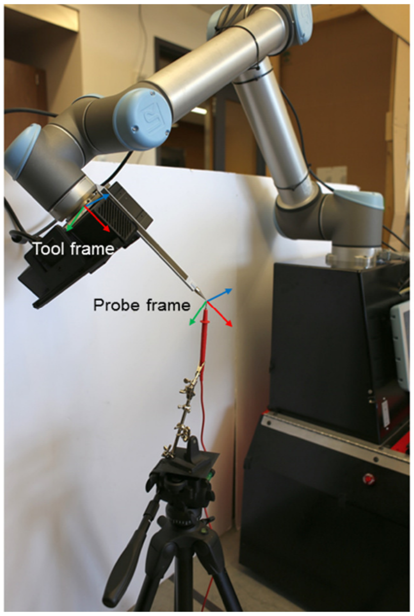


Figure 6. Probe tip calibration. The end effector was carefully guided to a pose where the probe tip touched the calibration point. Tool frame refers to the last reference frame of the robotic manipulator. Probe frame is obtained by translating tool frame to probe tip. Red arrow: X-axis; Green arrow: Y-axis; Blue arrow: Z-axis.

Without loss of generality, perpendicularly probing a point on the plant leaf surface was investigated. Given a probing point and its normal direction, the probing pose is not uniquely defined. The robot tool frame may rotate around the surface normal axis while the robot does not self-collide or collide with the environment. The solution set could be infinite, but only one solution is required. Therefore, a finite number of probing pose candidates were tested by rotating the end effector around the surface normal axis with a step size of ten degrees. The collision-free candidate with the least rotation of the last wrist joint was chosen to avoid twisting the sensor cables. Finally, the configuration of the robotic manipulator was computed using the analytic inverse kinematics solutions implemented in the ROS (Quigley et al., 2009) package *ur_kinematics* (Hawkins, 2013).

EXPERIMENT SETUP

Except for calibration, all the experiments were conducted with a growth chamber mock-up in front of the robot and four pots of artificial plants were placed in a row at the center of the chamber floor (fig. 1). The pot size was 0.28 m in diameter, and the plant height was approximately 0.28 m. The average leaf length and maximum leaf width were 0.13 m and 0.03 m, respectively. The sensors and the robotic manipulator were controlled by an industrial PC equipped with an Intel Core i5-4300U CPU clocked at 1.9 GHz with 8 GB RAM. OpenCV and PCL were utilized for 3D perception and visualization. PRMCE was implemented using the Boost Graph Library for the roadmap construction and A* graph search. The precomputed data structure was saved and loaded with Boost Serialization Library. *K*-nearest neighbor search was done using the FLANN library (Muja and Lowe, 2009).

RESULTS

In this section, the results of all key aspects of the robotic leaf probing pipeline are reported.

CALIBRATION

For TCP calibration, eight random robot tool frames were recorded for probing a calibration probe tip fixed in robot base frame. For each data acquisition, the teach pendant was used to carefully orient and position the probe (fig. 6). The number of linear equations equals the number of unique pairs of poses $\binom{8}{2}$. The least-squares solution resulted in a translation vector of the probe tip in robot tool frame $(0.0866, 0.0268, 0.2485)^T$ (m). Because the position of the calibration probe tip was unknown in robot base frame, the positions of the probe tip in robot base frame are computed using the TCP calibration result. The standard errors of the means of the positions of the probe tip in X, Y, and Z dimensions of robot base frame are presented in table 1. The errors also include human error when aligning the tool probe tip and calibration probe tip.

Table 1. Standard errors of the means of 8 probe tip positions in robot base frame in X, Y and Z dimensions.

Dimension	X	Y	Z
Standard Error (mm)	0.16	0.16	0.17

For hand-eye calibration of the laser profilometer, nine planes with different orientations were used and six lines on each plane were scanned to estimate the plane equation. In total, 32681 data points were acquired. Figure 7 shows scanning a line on one plane with the profilometer, and the flat sheet metal was not moved during the scanning of the six lines on the plane.

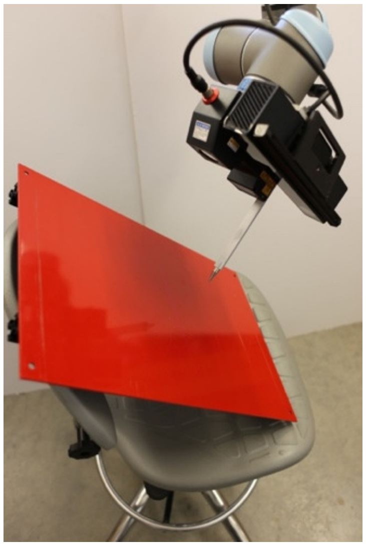


Figure 7. Scanning a line on a planar object for laser profilometer hand-eye calibration.

Based on the 3D computer-aided design models of the UR10, the Keyence laser profilometer, and the custom adapter connecting the two components, the initial hand-to-eye transformation was

$$T_{hand2scanner} = \begin{bmatrix} 0 & 1 & 0 & 0.076 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.094 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The optimization was run for ten iterations, returning the final estimate

$$T_{hand2scanner} = \begin{bmatrix} 0.0066 & 0.9999 & -0.0087 & 0.0761 \\ -0.9999 & 0.0067 & 0.0042 & 0.0018 \\ 0.0043 & 0.0087 & 0.9999 & 0.0941 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The root mean squared (RMS) distance between the points to their associated planes is illustrated in figure 8. The final RMS distance was 0.39 mm.

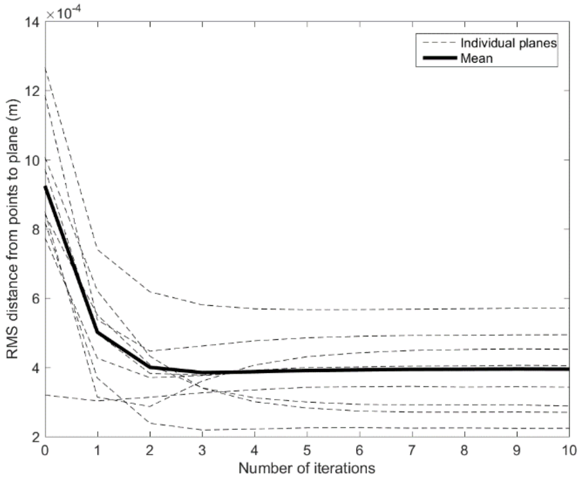


Figure 8. Convergence results for profilometer hand-eye calibration data gathered from nine planes. The RMS distance between measurement points and their associated planes are shown for individual planes and the mean over all planes.

3D MAPPING AND PLANT PERCEPTION

The environment inside the growth chamber mock-up was mapped with three views (a, b, and c in fig. 9). The 3D point cloud of the environment was regularized by VoxelGrid filter to the resolution of 2 mm whenever a new view was merged. The plants were first cropped out the point cloud. Then individual plants were segmented (d in fig. 9) by using the implementation of *k*-means clustering in OpenCV which supports a robust seeding procedure (Arthur et al., 2007). The average runtime of the plant segmentation in the combined point cloud acquired by Kinect V2 was approximately 200 ms.

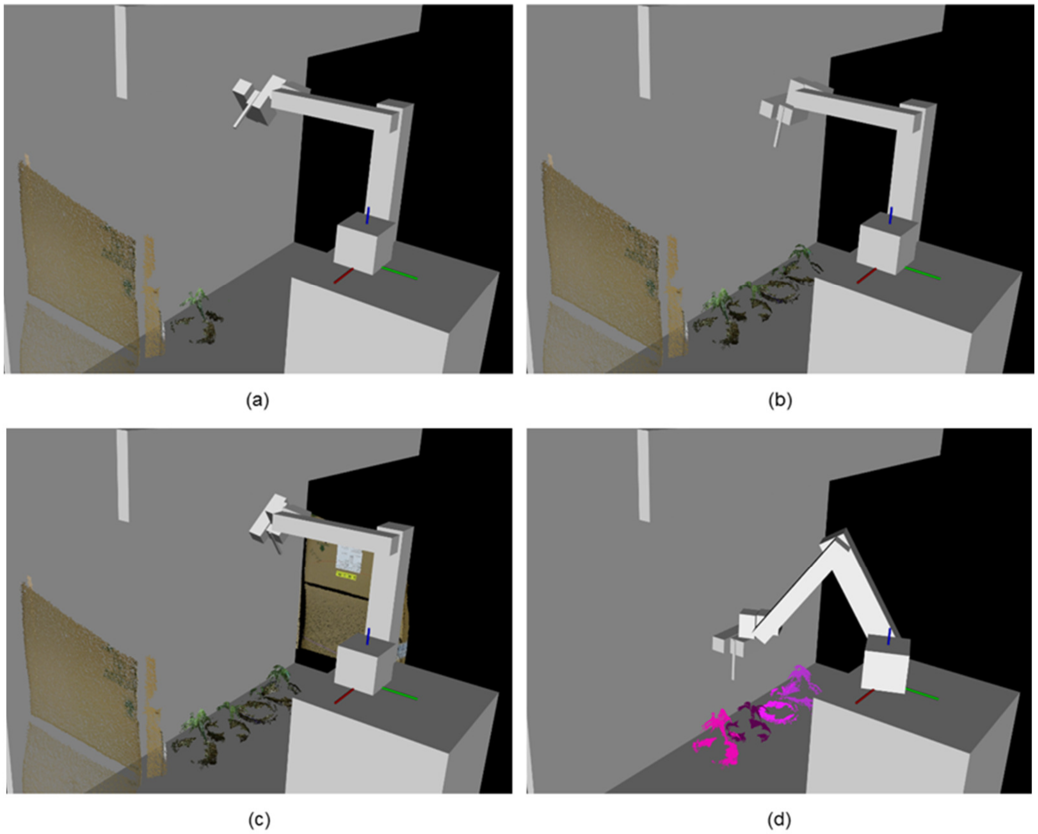


Figure 9. 3D mapping in the growth chamber mock-up and plant segmentation. (a) Imaging pose 1. (b) Imaging pose 2. (c) Imaging pose 3. (d) Plant segmentation via *k*-means clustering.

Each plant was scanned with the laser profilometer at a sweeping speed of 0.1 m/s with the maximum scan rate of 2000 Hz. The resultant spatial resolution in the sweeping direction was 0.05 mm. Because the nominal spatial resolution in the scanline direction provided by the laser profilometer was 0.225 mm, the 0.1 m/s sweeping speed ensured that the combined resolution was mainly limited by the laser profilometer. Potentially the sweeping speed could be increased by approximately four times to achieve equal resolutions on both axes, but increasing the sweeping speed also increases the synchronization error between the laser profilometer and the robotic manipulator and thus reduces probing accuracy. The 0.1 m/s sweeping speed was found to be an acceptable balance. The scanned point cloud was filtered through a voxel grid of size 1 mm. Large leaf segments were found by 3D region growing segmentation, and small planar patches were extracted by supervoxel segmentation. The key parameters of the two segmentation algorithms are listed in table 2 and table 3. A supervoxel segment was considered as a probing candidate if the RMS value of plane fitting was less than 0.5 mm. Figure 10 demonstrates the extracted leaf segments and potential probing points. The average runtime of the two segmentation algorithms per plant was approximately 500 ms. The number of leaf segments depended on the smoothness and curvatures of the leaves. In the third and fourth plants in figure

10, many leaves either were not smooth or had high curvatures, thus were rejected by the 3D region growing segmentation algorithm. On the other hand, some leaf segments could include more than one leaf when multiple leaves overlapped each other with a smooth transition of their surface normals, which prevented them from being separated by the 3D region growing segmentation algorithm (e.g., the bottom left leaf of the second plant in fig. 10). The goal of leaf segmentation was not to find every individual leaf, but to ensure that the extracted probing points were from some large smooth leaf surfaces.

Table 2. Key parameters for the 3D region growing segmentation.

Number of neighbors	Residual threshold	Smoothness threshold	Minimum cluster size	Maximum cluster size
6	0.01	7°	1000	10000

Table 3. Key parameters for the supervoxel segmentation.

Voxel resolution	Seed resolution	Spatial importance	Normal importance
1 mm	20 mm	0.5	1.0

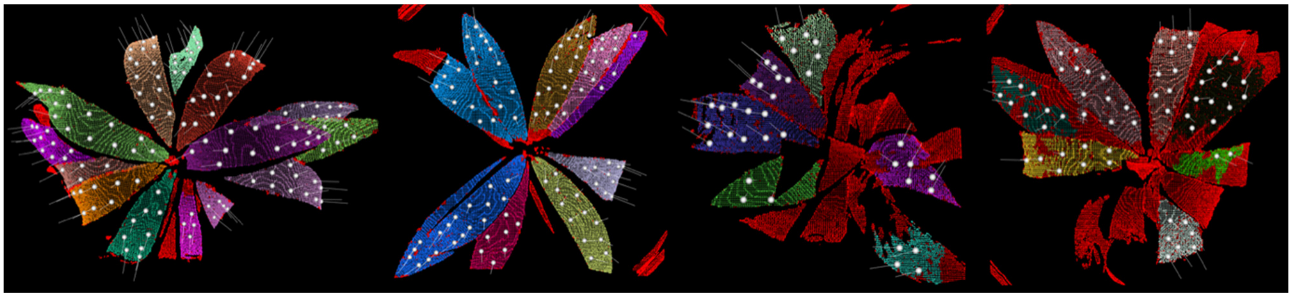


Figure 10. Leaf segmentation results and probing location candidates. The images are top views of the 3D point clouds scanned by the laser profilometer. Red points are rejected by 3D region growing segmentation. Any other color represents a different leaf segment. A feasible probing location is shown with a white sphere and the surface normal.

ROBOTIC PROBING ACCURACY

This robotic probing system was tested to scan and probe the centroid of a 90 mm × 90 mm square surface of a calibration block. The calibration block was first placed in front of the robot such that the square edges were either parallel or perpendicular to the X-axis and the Y-axis of robot base frame. As a result, the square surface was perpendicular to the Z-axis of robot base frame. Then the square surface was scanned such that the laser sheet was perpendicular to the surface and parallel to the Y-axis moving in X direction (fig. 11).

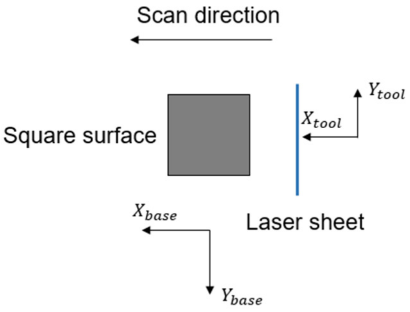


Figure 11. Top view of the scanning setup for the robotic probing accuracy test.

The rest of the processing pipeline would automatically extract the square surface as well as the probing pose, and execute the probing. Once the probe reached its destination, the position of robot tool frame was recorded as $\boldsymbol{t}_{robot}$. Figure 12 shows probing the square centroid perpendicularly. Then probe was carefully guided in absolute translation by using the teach pendant to probe the physical centroid, resulting a new position of robot tool frame, $\boldsymbol{t}_{manual}$. Probing position error is defined as the Euclidean distance between $\boldsymbol{t}_{robot}$ and $\boldsymbol{t}_{manual}$. As for probing orientation, the angle between the probe and the square surface was measured, and the error was defined as the absolute deviation from 90 degrees due to the targeted perpendicular probing action. The angle was measured by using a digital protractor with an accuracy of ± 0.1 degrees. The full process was repeated three times. In addition to the leveled surface, the same process was done when the calibration block was rotated around X-axis and Y-axis of the robot base frame, respectively, by 0, ± 30, and ± 60 degrees. The probing position errors and angle errors with different surface orientations are shown in figure 13. In summary, the system achieved a probing position error of 1.5 ± 0.20 mm and a probing angle error of 0.84 ± 0.18 degrees with a 95% confidence interval. More illustrations showing how plants might be probed from different angles are given in figure 14.

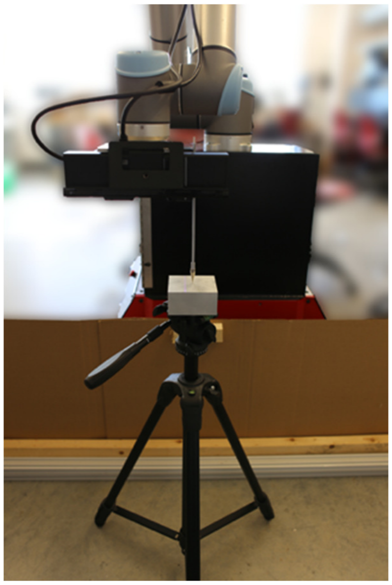


Figure 12. Probing the centroid of 0.09×0.09 m square surface perpendicularly.

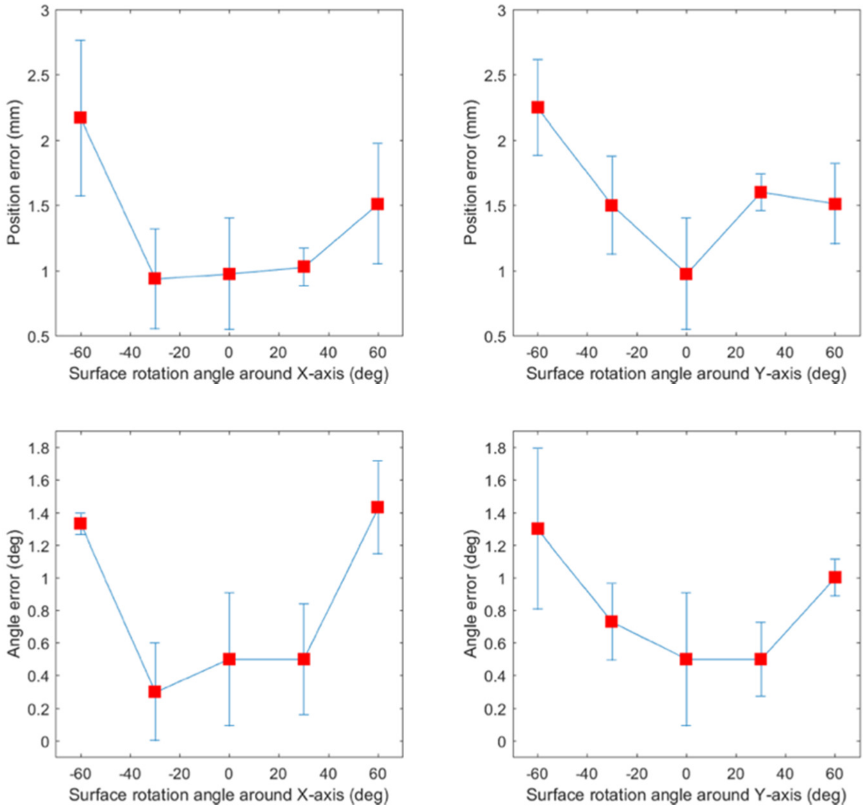


Figure 13. Probing errors with different surface orientations. Each surface orientation was probed three times. Error bars indicate 95% CI. Top: absolute probing position errors when the square surface was rotated around X-axis (left) and Y-axis (right) of robot base frame. Bottom: absolute probing angle error when the square surface was rotated around X-axis (left) and Y-axis (right).

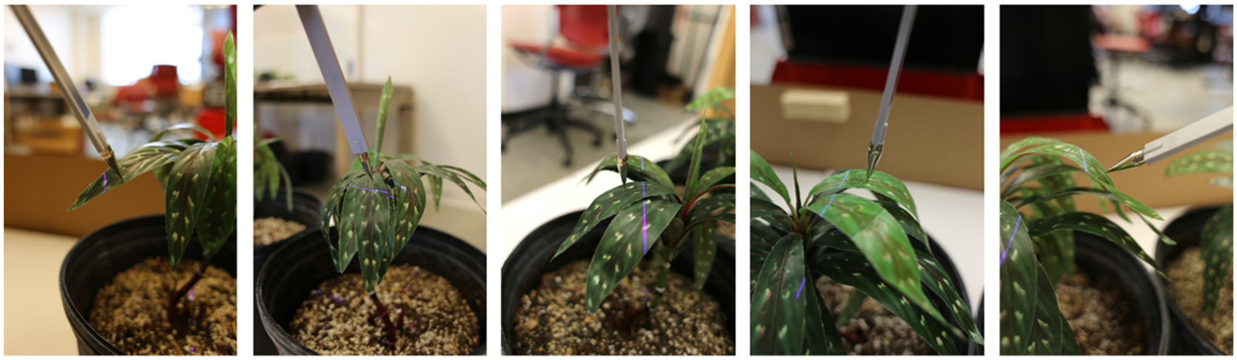


Figure 14. Leaf probing pose examples performed by the robotic system.

Motion Planning

The PRMCE motion planner was evaluated regarding planning time and path quality. The full robotic leaf probing routine was tested on the experimental setup illustrated in figure 1. The plants were processed in a decreasing order of their X coordinates in robot base frame. For each plant, the leaf segments were probed in the same order. For each leaf segment, the maximum number of probing points allowed was set to one.

The key parameters for the planner are number of nodes *N*, number of neighbors per node *K*, and 3D occupancy grid size. They were set to 5000, 20, and 0.04 m, respectively. The parameters were chosen based on the results by Kunz et al. (2010) except that the number of nodes *N* was about one-third of the original value to reduce the preprocessing time without compromising the performance. As a result, generating the reusable roadmap took 37 min and the precomputed data structure occupied ~1 GB RAM, but loading the file only took ~2.5 s.

In total, 35 leaf segments were found existed. The numbers of not probed leaf segments in each plant were 4, 8, 4, and 3 because no collision-free probing solutions were found. Among the remaining 16 leaf segments which were probed, the numbers of leaf segments in each plant were 4, 6, 2, and 4. Although multiple probing points were available for these leaves, probing multiple nearby points on the same leaf would not reveal the effectiveness of motion planning. Hence, only one probing point on a leaf segment was randomly selected and probed. The motion of the robotic manipulator was monitored in real time, and all of the leaves were successfully probed without collision. Motion planning happened when the robotic manipulator needed to move from one probing pose to the next. The planning was broken down into three stages: *Stage 1*- connecting to roadmap; *Stage 2* - A* graph search; *Stage 3*- path smoothing. Table 4 illustrates the average time and standard deviations of each stage and the complete motion planning.

Table 4. Average time and standard deviation of the three stages of motion planning.

Stage	Average time (ms)	Standard deviation (ms)
Connecting to roadmap	138.6	52.3
A* graph search	1.2	0.4
Path smoothing	276.5	320.2
Complete motion planning	416.3	355.6

The travel distances of two reference points on the robotic manipulator were used to quantify path quality, one at the tool frame (*Frame 6* in fig. 5) and one at the elbow frame (*Frame 2* in fig. 5). The elbow point was included because it connects the two longest arms of UR10. Table 5 shows the average travel distance and standard deviation for each reference point.

Table 5. Travel distances of tool frame and elbow frame of UR10.

Reference point	Average travel distance (m)	Standard deviation (m)
Tool	1.06	0.20
Elbow	0.92	0.18

DISCUSSION

The probing accuracy of the developed robotic leaf probing system was promising for automated instrumentation of sensor probes. This was achieved through the fusion of global and local environmental models, and accurate TCP and hand-eye calibrations. The errors of both probing position and probing angle tended to increase as the object surface became more slanted toward the laser profilometer. One possible reason for such behavior was the decrease in spatial resolution of the point cloud data when a slanted surface was scanned. This problem could be solved by rescanning each leaf segment in a fronto-parallel view although it would require more operation time. Another source of error was the uncertain delay between the time when the robotic manipulator started moving and the time when the laser profilometer started scanning. The UR10 was controlled via TCP/IP and the laser profilometer was triggered using its software development toolkit. It is possible to improve the probing accuracy if the robotic manipulator and the laser profilometer are synchronized via hardware.

With data from the high-precision profilometer, the 3D region growing segmentation with smoothness constraint algorithm was able to extract large piece-wise smooth leaf surfaces. However, overlapping leaves with similar surface normal directions tended to be merged into one segment. The 3D region growing segmentation is a generic point cloud segmentation method, thus is not necessarily capable of distinguishing between different types of leaves. However, it provided some geometric information about where the probing points were extracted. Detecting individual leaves of different shapes and sizes is a challenging problem. Potential solutions may lie in the state-of-the-art object detection algorithms using deep convolutional neural networks (Ren et al., 2015; Dai et al., 2016; Liu et al., 2016; Girshick et al. 2016; Redmon et al., 2016). Using RMS error of plane fitting on the supervoxel as a criterion to filter potential probing points worked as expected, especially when a supervoxel belonged to the edges of two overlapping leaves.

The motion planning experiment showed that the PRMCE planner was both fast and effective for robotic leaf probing in a growth chamber environment. As for planning time, *Stage 1* took approximately one-third of the total time. In this stage, the planner searched for the nearest configurations on the roadmap that could connect to the start and the goal configurations without collision. In addition to *K*-nearest neighbor search, it required collision checking up to *K* times for the start and the goal configurations, respectively. In *Stage 2*, the planner searched for a path on the roadmap. A* graph search proved to be highly efficient. The average time took ~1 ms, which was negligible compared to the other two stages. In *Stage 3*, unnecessary waypoints were removed from the path found in *Stage 2*. The running time depended on the number of waypoints in the original path and the specific scenario of the 3D workspace occupancy, resulting in a large standard deviation. Collision checking remained a major bottleneck since it required 3D voxelization of the OBBs representing the robotic manipulator and sensors. Its computational complexity largely depends on the size of 3D occupancy grid. The smaller the grid size is, the more 3D cells are visited, and the more accurate the collision checking is. Note that collision checking can be done independently for each OBB. Further speedup can be achieved by using parallel 3D voxelization of the OBBs with modern multi-core CPUs or GPUs. Finally, the paths found by PRMCE for moving from one probing pose to another were qualitatively reasonable. Usually, the path included two intermediate waypoints. The first waypoint moved the end effector safely away from the start configuration to a position higher than the plants. The second one served a similar purpose for the goal configuration. Because the used probe was fairly long (0.26 m), a path with only one intermediate waypoint was less likely to be collision-free.

This robotic leaf probing system was developed as a general method for automated non-contact instrumentation for molecular spectroscopy on plant canopy in a controlled environment. However, several limitations need to be overcome to deploy the system for probing real plants in a growth chamber across multiple growth stages. First, in case of densely populated plants, the *k*-means clustering method could not accurately differentiate overlapping canopies originating from different plants. It might be possible to track the plant organs starting from an early stage. Second, the maximum plant height which can be handled by the system is limited by the reach and dexterity of the robotic manipulator. For instance, the two long arms of UR10 might not allow the end effector to go over a tall plant without collision, but replacing long arms with multiple short arms would probably work in the same scenario. Third, the system assumes that the plants do not move. This assumption is valid in a growth chamber in which the fan can be shut down during the robot's operation. Plant motion might not be controllable in a greenhouse, in which case, a robotic gripper is needed to stabilize a leaf first before probing it. An interesting future research direction will be to investigate the coordination of real-time 3D sensing and real-time motion control of the robotic manipulator to achieve the gripping of a moving leaf. Fourth, the leaf segmentation method is not customized for different types of plants. If the prior knowledge of the leaf shape is known, the template matching approach used by Alenya et al. (2013) can allow users to define probing points on the template. Last, the leaves which can be probed without collision are most likely the top canopies with multiple plants in a growth chamber because the leaves below the top canopies are partially or fully occluded in the field of view of the vision system.

CONCLUSION

An eye-in-hand robotic system was developed to perform automated high-throughput 3D plant perception and leaf probing with real-time collision-free motion planning for a growth chamber environment. The automated probing pipeline consisted of four main steps: environment mapping, individual plant segmentation and scanning, probing pose extraction, and collision-free motion planning. Rapid environment mapping was realized by several snapshots with a Kinect V2 depth sensor guided by a six-axis robotic manipulator. The registered 3D point cloud was used for determining the workspace occupancy and individual plant segmentation. PRMCE allowed for real-time motion planning which avoided collision between the robot and the environment. High-precision 3D scanning of plant canopy was obtained by sweeping a 2D laser profilometer with the robotic manipulator at a close distance. The resultant 3D model allowed the system to achieve average absolute errors of probing position and probing angle of 1.5 mm and 0.84 degrees, respectively. The PRMCE motion planner was used to probe four artificial plants for a total of 16 leaves without collision with an average planning time of 416 ms and an average TCP travel distance of 1 m. These results are promising for the applicability of the approach presented for automated in situ plant phenotyping using instruments such as a fluorimeter and a Raman spectrometer. For future study, these spectroscopic sensors will be integrated into the system to study crop responses in different growing environments through close collaborations with plant scientists.

ACKNOWLEDGEMENTS

This research was supported by National Science Foundation under the project “MRI: Development of an ENVIRATRON – an accelerator for climate change research” (Grant No. 1428148).

REFERENCES

Ahlin, K., Joffe, B., Hu, A.-P., McMurray, G., & Sadegh, N. (2016). Autonomous leaf picking using deep learning and visual-servoing. *Proc. AgriControl 2016: The 5th IFAC Conf. on Sensing, Control and Automation Technol. for Agric.* Amsterdam, Netherland: Elsevier. https://doi.org/10.1016/j.ifacol.2016.10.033

Alenya, G., Dellen, B., Foix, S., & Torras, C. (2013). Robotized plant probing: Leaf segmentation utilizing time-of-flight data. *IEEE Robotics & Automation Magazine, 20*(3), 50-59. http://dx.doi.org/10.1109/MRA.2012.2230118

Araus, J. L., & Cairns, J. E. (2014). Field high-throughput phenotyping: The new crop breeding frontier. *Trends in Plant Science, 19*(1), 52-61. https://doi.org/10.1016/j.tplants.2013.09.008

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proc. 18th Annu. ACM-SIAM Symp. on Discrete Algorithms*. Philadelphia, PA: SIAM.

Carlson, F. B., Johansson, R., & Robertsson, A. (2015). Six DOF eye-to-hand calibration from 2D measurements using planar constraints. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.* Piscataway, NJ: IEEE. http://dx.doi.org/10.1109/IROS.2015.7353884

Cohen-Or, D., & Kaufman, A. (1997). 3D line voxelization and connectivity control. *IEEE Computer Graphics and Applications, 17*(6), 80-87. http://dx.doi.org/10.1109/38.626973

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. *Proc. Advances in Neural Information Process. Systems*.

Denavit, J. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *ASME J. Appl. Mech.*, 215-221.

Eberly, D. (2002). Dynamic collision detection using oriented bounding boxes. Geometric Tools, LLC. Retrieved from https://www.geometrictools.com/Documentation/DynamicCollisionDetection.pdf

Foix, S., Alenya, G., & Torras, C. (2015). 3D sensor planning framework for leaf probing. *IEEE/RSJ Int. Conf. on Intelligent Robots and*

Systems. Piscataway, NJ: IEEE. http://dx.doi.org/10.1109/IROS.2015.7354306

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 38*(1), 142-158. doi:http://dx.doi.org/10.1109/TPAMI.2015.2437384

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische mathematik, 14*(5), 403-420. doi:https://doi.org/10.1007/BF02163027

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Syst. Sci. and Cybernetics, 4*(2), 100-107. http://dx.doi.org/10.1109/TSSC.1968.300136

Hartenberg, R. S., & Denavit, J. (1964). *Kinematic synthesis of linkages.* New York City, NY: McGraw-Hill.

Hartmann, A., Czauderna, T., Hoffmann, R., Stein, N., & Schreiber, F. (2011). HTPheno: An image analysis pipeline for high-throughput plant phenotyping. *BMC Bioinformatics, 12*(1), 148. https://doi.org/10.1186/1471-2105-12-148

Hawkins, K. P. (2013). Analytic inverse kinematics for the universal robots UR-5/UR-10 arms. Atlanta, GA: Institute for Robotics and Intelligent Machines. Retrieved from https://smartech.gatech.edu/handle/1853/50782

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *J. of Educational Psychol., 24*(6), 417. http://dx.doi.org/10.1037/h0071325

Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation, 12*(4), 566-580. http://dx.doi.org/10.1109/70.508439

Kolb, A., Barth, E., Koch, R., & Larsen, R. (2009). Time-of-Flight sensors in computer graphics. *Proc. 30th Annu. Conf. of European Association for Computer Graphics*. Hoboken, NJ: Wiley.

Kunz, T., Reiser, U., Stilman, M., & Verl, A. (2010). Real-time path planning for a robot arm in changing environments. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.* Piscataway, NJ: IEEE. https://doi.org/10.1109/IROS.2010.5653275

Lachat, E., Macher, H., Mittet, M., Landes, T., & Grussenmeyer, P. (2015). First experiences with Kinect v2 sensor for close range 3D modelling. *The Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(5), 93. https://doi.org/10.5194/isprsarchives-XL-5-W4-93-2015

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Computer Science Dep., Iowa State University.

Lawson, C. L., & Hanson, R. J. (1995). *Solving least squares problems*. Philadelphia, PA: SIAM.

Leven, P., & Hutchinson, S. (2002). A framework for real-time path planning in changing environments. *The Int. J. of Robotics Res., 21*(12), 999-1030. https://doi.org/10.1177/0278364902021012001

Li, L., Zhang, Q., & Huang, D. (2014). A review of imaging techniques for plant phenotyping. *Sensors, 14*(11), 20078-20111. http://dx.doi.org/10.3390/s141120078

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conf. on Computer Vision.* Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-46448-0_2

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1), 2*(331-340), 2.

Murray, S., Floyd-Jones, W., Qi, Y., Sorin, D. J., & Konidaris, G. (2016). Robot Motion Planning on a Chip. *Proc. Robotics: Sci. and Systems.*

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., . . . Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking*. 10th IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR).* Piscataway, NJ: IEEE. https://doi.org/10.1109/ISMAR.2011.6092378

Papon, J., Abramov, A., Schoeler, M., & Worgotter, F. (2013). Voxel cloud connectivity segmentation-supervoxels for point clouds. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition.* Piscataway, NJ: IEEE. https://doi.org/10.1109/CVPR.2013.264

Park, F. C., & Martin, B. J. (1994). Robot sensor calibration: solving AX= XB on the Euclidean group. *IEEE Trans. on Robotics and Automation, 10*(5), 717-721. http://dx.doi.org/10.1109/70.326576

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., . . . Ng, A. Y. (2009). ROS: An open-source robot operating system*. ICRA Workshop on Open Source Software.*

Rabbani, T., Van Den Heuvel, F., & Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*(5), 248-253.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE. https://doi.org/10.1109/CVPR.2016.91*

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks*. Proc. Advances in Neural Information Process. Systems.*

Rusu, R. B. (2010). Semantic 3d object maps for everyday manipulation in human living environments. PhD diss. Garching b. München, Germany: Technische Universität München, Computer Science Department.

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *2011 IEEE Int. Conf. on Robotics and Automation.* Piscataway, NJ: IEEE. https://doi.org/10.1109/ICRA.2011.5980567

Schulz, H., & Baranska, M. (2007). Identification and quantification of valuable plant substances by IR and Raman spectroscopy. *Vibrational Spectroscopy, 43*(1), 13-25. https://doi.org/10.1016/j.vibspec.2006.06.001

Shah, D., & Tang, L. (2016). Development of an autonomous indoor phenotyping robot. ASABE Paper No. 162460767. St. Joseph, MI: ASABE. https://doi.org/10.13031/aim.20162460767

Shah, D., Tang, L., Gai, J., & Putta-Venkata, R. (2016). Development of a mobile robotic phenotyping system for growth chamber-based studies of genotype x environment interactions*. AgriControl 2016: The 5th IFAC Conf. on Sensing, Control and Automation Technol. for Agric.* Amsterdam, Netherland: Elsevier. https://doi.org/10.1016/j.ifacol.2016.10.046

Template for ASABE Journal Authors

Please complete a table for each author, in order, so that author names are indexed online.

See the ASABE website for more information for authors.

Author 1

First Name or initial	Middle Name or initial	Surname		ASABE member, yes/no? Role (job title, etc.)	E-mail (and phone for contact author)	C a ye
Yin		Bao		Yes	yinbao@iastate.edu	

Affiliation for Author 1

Organization	Address	Country	URL or other info.
Iowa State University	1201 Sukup Hall Ag & Biosystems Engineering Iowa State University Ames, Iowa, 50011	USA	

Author 2 (repeat Author and Affiliation tables for each author)

First Name or initial	Middle Name or initial	Surname		ASABE member, yes/no? Role (job title, etc.)	E-mail (and phone for contact author)	C a ye
Dylan	S	Shah		No	dylan.shah@yale.edu	

Affiliation for Author 2

Organization	Address	Country	URL or other info.
Iowa State University	1201 Sukup Hall Ag & Biosystems Engineering Iowa State University Ames, Iowa, 50011	USA	

Author 3 (repeat Author and Affiliation tables for each author)

First Name or initial	Middle Name or initial	Surname		ASABE member, yes/no? Role (job title, etc.)	E-mail (and phone for contact author)	C a ye
Lie		Tang		Yes	lietang@iastate.edu 515-294-9778	

Affiliation for Author 3

Organization	Address	Country	URL or other info.
Iowa State University	2346 Elings Hall 605 Bissell Road Ag & Biosystems Engineering Iowa State University Ames, IA, 50011	USA	