11-2014

Automatic morphological trait characterization for corn plants via 3D holographic reconstruction

Supawadee Chaivivatrakula
Ubon Ratchathani University and Asian Institute of Technology

Lie Tang
Iowa State University, lietang@iastate.edu

Matthew N. Dailey
Asian Institute of Technology

Akash D. Nakarmi
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/abe_eng_pubs

Part of the Agriculture Commons, and the Bioresource and Agricultural Engineering Commons

The complete bibliographic information for this item can be found at https://lib.dr.iastate.edu/abe_eng_pubs/1063. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
Automatic morphological trait characterization for corn plants via 3D holographic reconstruction

Abstract
Plant breeding is an extremely important route to genetic improvements that can increase yield and plant adaptability. Genetic improvement requires careful measurement of plant phenotypes or plant trait characteristics, but phenotype measurement is a tedious and error-prone task for humans to perform. High-throughput phenotyping aims to eliminate the problems of manual phenotype measurement. In this paper, we propose and demonstrate the efficacy of an automatic corn plant phenotyping system based on 3D holographic reconstruction. Point cloud image data were acquired from a time-of-flight 3D camera, which was integrated with a plant rotating table to form a screening station. Our method has five main steps: point cloud data filtering and merging, stem segmentation, leaf segmentation, phenotypic data extraction, and 3D holographic visualization. In an experimental study with five corn plants at their early growth stage (V3), we obtained promising results with accurate 3D holographic reconstruction. The average measurement error rate for stem major axis, stem minor axis, stem height, leaf area, leaf length and leaf angle were at 7.92%, 15.20%, 7.45%, 21.89%, 10.25% and 11.09%, respectively. The most challenging trait to measure was leaf area due to partial occlusions and rolling of some leaves. In future work, we plan to extend and evaluate the usability of the system in an industrial plant breeding setting.

Keywords
Phenotyping, Trait characterization, 3D holographic reconstruction, Corn plant, Point cloud data

Disciplines
Agriculture | Bioresource and Agricultural Engineering

Comments

Creative Commons License
This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.

This article is available at Iowa State University Digital Repository: https://lib.dr.iastate.edu/abe_eng_pubs/1063
Automatic Morphological Trait Characterization for Corn Plants via 3D Holographic Reconstruction

Supawadee Chaivivatrakula\textsuperscript{a,b,*}, Lie Tang\textsuperscript{b}, Matthew N. Dailey\textsuperscript{b}, Akash D. Nakarmi\textsuperscript{c}

\textsuperscript{a}Information Technology for Agriculture and Rural Development, Ubon Ratchathani University, Ubon Ratchathani, Thailand 34190.
\textsuperscript{b}Computer and Information Management, Asian Institute of Technology, Pathumthani, Thailand 12120.
\textsuperscript{c}Agricultural and Biological Systems Engineering, Iowa State University, Ames, IA, 50011.

Abstract

Plant breeding is an extremely important route to genetic improvements that can increase yield and plant adaptability. Genetic improvement requires careful measurement of plant phenotypes or plant trait characteristics, but phenotype measurement is a tedious and error-prone task for humans to perform. High-throughput phenotyping aims to eliminate the problems of manual phenotype measurement. In this paper, we propose and demonstrate the efficacy of an automatic corn plant phenotyping system based on 3D holographic reconstruction. Point cloud image data were acquired from a time-of-flight 3D camera, which was integrated with a plant rotating table to form a screening station. Our method has five main steps: point cloud data filtering and merging, stem segmentation, leaf segmentation, phenotypic data extraction, and 3D holographic visualization. In an experimental study with five corn plants at their early growth stage (V3), we obtained promising results with accurate 3D holographic reconstruction. The average measurement error rate for stem major axis, stem minor axis, stem height, leaf area, leaf length and leaf angle were at 7.92\%, 15.20\%, 7.45\%, 21.89\%, 10.25\% and 11.09\%, respectively. The most challenging trait to measure was leaf area due to partial occlusions and rolling of some leaves. In future work, we plan to extend and evaluate the usability of the system in an industrial plant breeding setting.

Keywords: Phenotyping, Trait characterization, 3D holographic reconstruction, Corn plant, Point cloud data
1. Introduction

The growing world population and the lack of access to new arable land needed to maintain agricultural sustainability (Araus et al., 2008) is making plant breeding more important than ever before, as it can increase crop yield and plant adaptability. Genetic improvement in plant breeding focuses on exactly this problem, through the selection for desired plant phenotypes during the plant breeding cycle (Poehlman and Sleper, 1995). Genetic improvement requires careful measurement of plant phenotypes, which is a tedious and error-prone task when done manually. High-throughput phenotyping is a new technique that returns a large quantity of data and addresses the problems of manual phenotyping, so it has recently received a great deal of attention for the plant trait discovery task (Cabrera-Bosquet et al., 2012; Furbank and Tester, 2011). High-throughput phenotyping uses various technologies, such as near-infrared spectroscopy spectral reflectance, photography, and sonar. For example, Khanna et al. (2011) used several spectral analyses to classify aquatic macrophytes species.

For many years, botanists have observed the structure of plants and proposed many models to describe plant parts. Some models describe overall structure, and some describe specific parts of plants. Furthermore, visualization of plant models has also been enhanced with computer graphics techniques. For example, Ijiri et al. (2005) developed 3D simulation of inflorescences, Ding et al. (2008) conducted 3D modeling of a plant structure, and Yao et al. (2010) proposed a flower blooming 3D model. However, this work has focused on general 3D models without exploiting the unique phenotypic characteristics of specific plants. Some groups of researchers produce 3D models of rice plant from images and barley plants from 3D sensors (Wernecke et al., 2007; Watanabe et al., 2005). Other researchers have developed
methods for corn phenotype discovery and 3D visualization. Dornbusch et al. (2007) proposed a corn plant modeling procedure based on merging multiple 3D point cloud inputs and a mathematical model. Although they achieved excellent results, they did not perform automatic image capture and used only one single plant to demonstrate the use of the model. de Moraes Frasson and Krajewski (2010) also developed a 3D digital model of corn plants, but their method required attaching numerous markers to specific locations on a plant. None of the systems described in the literature are capable of fully automatic extraction of phenotypic data from their 3D models. The objective of the research reported upon in this paper was to develop a fully automatic system that is capable of characterizing corn plant morphological phenotypes in a controlled setting in a high-throughput fashion. These data include stem diameter, number of leaves, leaf length, leaf area, and leaf angle.

2. Materials and methods

An overview of our system is shown in Figure 1. Our modeling system interacts with a plant screening station consisting of a time-of-flight (ToF) camera (SR-4000, MESA Imaging, Switzerland), a turntable, a stepper motor, and a computer station. The modeling system sends commands to the plant screening station, including commands to precisely rotate the turntable and to acquire a 3D point cloud at a specific viewing angle. After point cloud acquisition, we perform point cloud data filtering, merging, stem segmentation, leaf segmentation, phenotypic data extraction, and 3D holographic visualization. The output of each step is depicted in Figure 2.

Figure 3 shows the experimental screening station, with the SR-4000 camera facing the plant turntable. We performed an initial calibration to identify the point $p_0$ (the center of the turntable's coordinate system at the top of the plant pot) in the camera coordinate system.
The methods described in this report could be used with some modification for corn plants at other growth stages (up to V10, after which there will be tassels and fruits). For the purpose of algorithm validation, five corn plants at the representative V3 growth stage were used.

2.1. Point cloud data filtering and merging

The point cloud data filtering and merging step comprised five intermediate steps: acquiring input point clouds, filtering, rotation and transform, fine alignment using iterative closest point (ICP) method, and filtering of the merged point cloud. The filtering, ICP, and merged point cloud filtering steps made use of the Point Cloud Library’s (PCL) built-in functions (Rusu and Cousins, 2011). Figure 5 shows sample results for each intermediate step of the process.

2.1.1. Acquiring point clouds

For each predefined acquisition angle, we sent a command to rotate the turntable and then a command to acquire a 3D point cloud. To ensure that the plant would settle from vibrations before image acquisition, we inserted a one-minute interval between turntable rotation and point cloud acquisition. The data from the ToF camera were in an XYZ format according to the camera’s Cartesian coordinate system depicted in Figure 4. The result of this step was a set of point clouds

\[ P_{0_i,1} = \{ p_{0_i,1}^j = (x_{0_i,1}^j, y_{0_i,1}^j, z_{0_i,1}^j) \}_{1 \leq j \leq N_{0_i}} \]

for various rotation angles \( 0 \leq \theta_i < 2\pi \), \( 1 \leq i \leq N_0 \). \( N_{0_i} \) is the number of points in point cloud \( P_{0_i,1} \), and \( N_0 \) is the total number of point clouds (acquisition angles).
Figure 1: Overview of the system, including screening system, modeling system, and main steps in modeling system.

Figure 2: Illustration of output of each main step in the modeling system.
Figure 3: Screening station setup. (a) Station setup, SR-4000 camera, a turntable, and a desktop computer. (b) A plant on the turntable in the view of the camera.

Figure 4: Coordinate system for input point clouds extracted from the screening system.

Figure 5: Results of intermediate steps of point cloud filtering and merging process.
2.1.2. Filtering

The point clouds generated by the ToF camera normally contained quite a few sparsely distributed outlier points. We used the statistical filtering algorithm of Rusu et al. (2008), which considers a point as either an inlier or an outlier based on the mean and variance of its distance to its \( k \)-nearest neighbors. Points with atypically sparse neighbor distance distributions (points with nearest neighbor distance greater than \( \mu + \alpha \sigma \), where \( \mu \) is the average and \( \sigma \) is the standard deviation of the nearest neighbor distances over the entire point cloud) were filtered out as outliers. We set \( \alpha \) to be conservative in this first filtering phase then less conservative during a later filtering the merged point cloud. The result was the filtered point cloud \( P_{0,2} \) with outliers removed.

\[
P_{0,2} = \text{FILTER}(P_{0,1}, k, \alpha)
\]

\[
P_{0,2} \subseteq P_{0,1}
\]

2.1.3. Rotation and transform

Next, we rotated each individual point cloud into rough alignment with each other using the turning angle \( \theta \) used at the acquisition time. We applied the necessary rotation and translation (Trucco and Verri, 1998; Forsyth and Ponce, 2002)

\[
P_{0,3} = \text{TRANSFORM}(P_{0,2}, \text{ROTATION}_y(\theta), p_0),
\]

where \( \text{TRANSFORM}(P, R, p) \) translates the points in \( P \) by \( -p \) (so that \( p \) is moved to the origin), rotates the resulting points with rotation matrix \( R \), and then translates the points by \( p \). \( \text{ROTATION}_y(\theta) \) returns the rotation matrix for a rotation of \( \theta \) around the y axis, and \( p_0 = (x_0, y_0, z_0) \) is the center point for the rotation, which we measured manually one time, when the turntable was first set up.
2.1.4. Fine alignment

After the rough alignment achieved by the previous step, we performed a more accurate registration of the point clouds using the ICP algorithm (Zhang, 1992; Rusinkiewicz and Levoy, 2001). We used the basic ICP implementation from the PCL library. We set the initial target point cloud to be $P_{0,3}$, the cloud acquired at a rotation of $0^\circ$, then we added views one by one, aligning each to the target:

$$P_k = \text{ITERATIVE-CLOSEST-POINT}((P_{0,3})_{1 \leq i \leq N_0}, m, r, t_i).$$

where $P_k$ is the resulting merged point cloud. $m$ is the maximum number of iterations. If the algorithm performs $m$ iterations without finding any matching points, it stops without returning any points at all. $r$ is the maximum rotation threshold. ICP will not rotate the source point cloud by an angle more than $r$. $t_i$ is the translation threshold, in Euclidean distance. ICP will not translate the point cloud by a distance more than $t_i$.

2.1.5. Filtering the merged point cloud

Since corn leaves are very thin, to make sure we keep all points on the leaf and only filter out true outliers during the initial filtering process, we used a conservative value of $\alpha$ for the statistical filtering of individual point clouds. At this point, some outlier points remained. Therefore, after we merged the individual point clouds, getting a denser set of points for every leaf, we applied the filter one more time with a less conservative outlier threshold to remove any remaining outlier points:

$$P_5 = \text{FILTER}(P_4, k, \alpha)$$

$P_5$ is the filtered version of $P_4$. As before, $k$ is the number of nearest neighbor distances to consider, and $\alpha$ is the number of standard deviations from the mean neighboring point distance to consider a point as an inlier.
2.2. Stem segmentation

To identify the stem in the merged point cloud $P_5$, we first noted that the stem could be approximately modeled as an elliptical cylinder. We thus searched for a segment of the merged point cloud that could be considered as an elliptical cylinder. The stem segmentation step was composed of seven intermediate steps: point cloud slicing, morphological closing, contour extraction, least squares ellipse fitting, ellipse linking, elliptical cylinder extrusion, least squares line fitting, and elliptical cylinder point cloud generation. We implemented these steps using routines in the OpenCV library (OpenCV Community, 2013). Figure 6 shows sample results for each intermediate step of the process.

2.2.1. Point cloud slicing

We sliced the merged point cloud along the $y$ axis, projecting the points in each slice orthogonally onto the $x-z$ plane. We took each point’s $(x, z)$ coordinates and treated them as pixels in a 2D binary image. For each slice $s \in 1..N_s$, we extracted the point cloud

$$P_{s,6} = \left\{ P_{s,6}^j = (x_{s,6}^j, y_{s,6}^j, z_{s,6}^j) \in P_5 \mid \begin{align*}
(y_{\min} + \frac{y_{\max} - r - y_{\min}}{N_y} \times s - r) \leq y_{s,6}^j \\
(y_{\min} + \frac{y_{\max} - r - y_{\min}}{N_y} \times s + r)
\end{align*} \right\}$$

then projected $P_{s,6}$ to

$$I_{s,1} = \left\{ \mathbf{q}_{s,1} = (x_{s,1}, y_{s,1}, z_{s,1}) \mid x_{s,1} = x_{s,6}^j / \ell_4; y_{s,1} = y_{s,6}^j / \ell_4 \right\}$$
where $I_{s,1}$ is the binary image formed by the projection of $P_{s,6}$ to the $x - z$ plane. $r$ is half of the slice thickness (the thickness of each slice is $2r$). $y_{\text{min}}$ and $y_{\text{max}}$ are the minimum and maximum values of $y$ in $P_5$. $t_q$ is the pixel size in the binary image.

2.2.2. Morphological closing

We treated the projection of each slice as a binary image and applied the morphological closing operation

$$(I_{s,2}, O_{s,1}) = \text{CLOSING}(I_{s,1}; d_1)$$

$I_{s,2}$ is the binary image after performing morphological closing on $I_{s,1}$. $O_{s,1}$ is the set of connected components remaining after the closing operation. $d_1$ is a disk-shaped structuring element.
2.2.3. Contour extraction

We took only those connected components that were large enough to be considered as the projection of the stem, then we found the outer contour of each resulting connected component.

\[
O_{s,2} = \{o'_{s,2} \in O_{s,1} \mid |o'_{s,2}| \geq t_c\}
\]

\[
O_{s,2} \subseteq O_{s,1}
\]

\[
T_s = \{\text{CONTOUR}(o'_{s,2})\}_{j \in 1 \ldots |O_{s,2}|}
\]
where \( |\sigma_{s,2}^j| \) is the area of connected component \( \sigma_{s,2}^j \). \( O_{s,2} \) is thus the set of connected components in \( O_{s,1} \) containing at least \( t_c \) points. \( T_s \) is the set of contours of the connected components in slice \( s \).

2.2.4. Least squares ellipse fitting

We applied least squares ellipse fitting for each contour.

\[
E_{s,1} = \{\text{LEAST-SQUARES-ELLIPSE-FITTING}(T_s^j)\}_{j \in [1, |T_s|},
\]

\( E_{s,1} \) is the set of ellipses output from least squares ellipse fitting for each of the contours in \( T_s \).

2.2.5. Ellipse linking

In this step, we aimed to find a subset of the ellipses in \( E_{s,1} \) whose \( x - z \) projections overlapped the most with each other over all slices of the point cloud. Two ellipses with \( x - z \) projections \( A \) and \( B \) are deemed to overlap each other if \( \frac{A \cap B}{A} \geq t_k \) and \( \frac{A \cap B}{B} \geq t_k \). We found, for each ellipse, the set of overlapping ellipses in other slices. We called the largest such overlap set the linkage set and placed the elements of the linkage set in a doubly linked list sorted by slice.

When multiple ellipses in a single slice were linked, their relative ordering was arbitrary. The resulting list

\[
E_2 = \text{LINKING}(E_{1,1}, \ldots, E_{N_s,1}, t_k)
\]

was used for elliptical cylinder extrusion in the next step.
2.2.6. Elliptical cylinder extrusion

To obtain an initial estimate of the vertical stem center line, we began with the two bottom-most ellipses in the linkage set $E_2$. Assuming these two ellipses were $e_1$ and $e_2$, we found the ellipse $e_3$ best modeling these two ellipses, i.e.,

$$e_3 = \text{LEAST-SQUARES-ELLIPSE-FITTING}(\text{SAMPLE}(e_1) \cup \text{SAMPLE}(e_2)),$$

We then passed $e_3$ along with the full linkage set $E_2$ to Algorithm 1, which attempts to extrude $e_3$ upward through the set of linked ellipses not too much larger than $e_3$, excluding any small ellipses at the top of the linkage set. To accomplish this, we started at the top of the linkage set then moved downward, first discarding any ellipses smaller than a large multiple of $e_3$’s area, then continuing downward until the first ellipse smaller than a smaller multiple of $e_3$’s size was found.
2.2.7. Elliptical cylinder point cloud generation

We calculated a line representing the vertical stem center (L) in Algorithm 1 taking $E_2$ (a linkage set of ellipses), $e_3$ (an ellipse representing the cross section of stem shape), $t_m$ (a large area threshold), $t_n$ (a large area threshold).

In this step, we generated a point cloud $P_6$ in the form of a mesh with a horizontal cross section in the shape of $e_3$ and a vertical stem center following line L. The resulting mesh modeled the plant stem estimated by the previous steps.

$$L = \text{VERTICAL-STEM-CENTER}(E_2, e_3, t_m, t_n)$$
$$P_6 = \text{ELLIPITICAL-CYLINDER-POINT-CLOUD}(e_3, L)$$

2.3. Leaf segmentation
Recall that we began stem segmentation based on the merged and filtered point cloud \( P_5 \). Once the stem has been identified and modeled as an elliptical cylinder, the next step was to identify each leaf segment in \( P_5 \). The leaves of a young corn plant are nearly horizontally opposed in \( x - y \) plane. Therefore, an initial leaf segmentation could be performed based on the projection of the point cloud onto the vertical plane in which the extent of the point cloud’s projection onto the \( x \) axis was largest. In our experimental setup, the first image was always taken with the plant aligned like this. Therefore, in our case, we simply projected each point onto the \( x - y \) plane, but of course, the appropriate projection could easily be determined automatically through a search for a rotation under which the \( x \) data have the widest extent, and this rotation could be used as an initial view.

We thus took each projected point’s \((x, y)\) coordinates and treated them as pixels in a 2D binary image. Next, we removed pixels corresponding to the rectangular bounding box (the smallest rectangle fitting to the stem pixels) containing projections of points on the stem. We then applied morphological closing followed by morphological opening to the rest of the binary image. After that, we applied the leaf-cutting method described in the next sections.
Figure 7 shows sample results of each intermediate step of the process.

2.3.1. Binary image of leaf point cloud

As mentioned above, we first projected the point cloud onto the x − y plane and removed the projection of the stem to obtain binary image I₃:

\[ I₃ = \{ q^i \in \text{PROJECTION-TO-XY-PLANE}(P_b) - \text{BOUNDING-BOX}(\text{PROJECTION-TO-XY-PLANE}(P_b)) \} , \]

where

\[ \text{PROJECTION-TO-XY-PLANE}(P) = \{ q^i = (x^i_p, y^i_p) | x^i_p = x^i / t_q; y^i_p = y^i / t_q; p^i = (x^i, y^i, z^i); p^i \in P \} \]

\( t_q \) is the pixel size for the binary image.
2.3.2. Morphological operations

To remove isolated points and fill holes in the leaf point cloud projection, we performed morphological closing and morphological opening operations:

$$(I_4, O_4) = \text{OPENING}(\text{CLOSING}(I_3, d_2), d_3),$$

where $I_4$ is the binary image returned after the closing operation followed by the opening operation. $O_4$ is the set of connected components after the morphological operations have been applied. $I_3$ is the projected binary image of merged point cloud $P_5$ with the stem region removed. $d_2$ is a disk-shaped structuring element for the closing operation, and $d_3$ is a disk-shaped structuring element for the opening operation. This step segmented the leaves along the stem into separate connected components, generally leaving a clump of merged leaves at the top of the plant.

2.3.3. Leaf cutting

The leaves at the top of the plant normally remain connected. In this step, we separated the clump of merged leaves into isolated leaf regions using Algorithms 2 and 3.

$$\{P_{l,t}\}_{l \in 1..N_1} = \text{LEAF-CUTTING}(I_4, P_5, O_4, (x_t, y_t))$$

where $I_4$ is the binary image from the previous step. $P_5$ is the original merged and filtered point cloud. $O_4$ is the set of connected components from the previous step. $(x_t, y_t)$ is the point in the plane representing the top end of the vertical stem center. $P_{l,t}$ is a segmented point cloud representing leaf $l$.

In the first pass (Algorithm 2), we chopped the topmost connected leaf component into candidate individual leaves at points of concavity around the component’s contour. In the
second pass (Algorithm 3), we ensured that any points in the point clouds for the two topmost leaves were properly associated by projecting onto the $y-z$ plane, chopping at points of concavity, and moving points between the resulting point clouds as necessary.

An assumption was made that only the topmost leaf segment contained more than one leaf and that the leaves in this segment could all be chopped at a point of inflection along the segment’s contour.

2.3.4. Leaf surface estimation

In this step, we smoothed and resampled data for each leaf point cloud using the moving least squares (MLS) method (Alexa et al., 2003). The algorithm fitted a 2D manifold to the 3D point cloud data and resampled the points to place on the estimated surface. In addition to estimating the manifold, the method also provided surface normal and curvature estimates and up-sampled or down-sampled the point set appropriately.

$$P_{l,8} = \text{LEAF\text{-}SURFACE\text{-}ESTIMATION}(P_{l,7}, t_r),$$

where $t_r$ is the sphere radius used for determining the k-nearest neighbors of a point. We configured MLS for polynomial estimation based on a k-d tree over the point set.

2.3.5. Attach leaf to cylinder

In some cases, after stem segmentation and leaf segmentation, we ended up with gaps between some of the leaf point clouds and the estimated stem cylinder. To fill these gaps, we added points to interpolate from the leaf to the stem.

$$P_{l,9} = \text{ATTACH\text{-}LEAF\text{-}TO\text{-}CYLINDER}(P_{l,8}, P_6),$$
where $P_6$ is the cylinder point cloud representing the plant stem.

2.4. Phenotypic data extraction

After the plant stem and individual leaves were segmented, we performed phenotype extraction and visualization. Due to the flat but curved shape of corn leaves, we adopted the MLS method to estimate a smooth 2D manifold for each leaf and re-sampled the leaf point clouds $P_{1,7}$ before extracting phenotypes and rendering for 3D holographic visualization in next section.
Algorithm 2 LEAF-CUTTING

Input: \(I\): binary image mapping leaf segments.
Input: \(P\): merged and filtered point cloud.
Input: \(O\): set of connected components in \(I\).
Input: \((x_t, y_t)\): projection of stem center point at the top of the stem.
Input: \(d\): structuring element for morphological operations.
Output: \(\{P_l\}_{l=1}^{N_l}\): a set of leaf point clouds indexed by leaf number \(l\).

1: \(O' \leftarrow \text{MINIMUM-Y-ASCENDING-SORT}(O)\)
2: \(O'\) is a stack of connected components.
3: \(d' \leftarrow \text{POPD}(O')\)
4: \(Q \leftarrow \text{UPWARD-CONCAVITY(CONVEXITY(CONTOUR(d'))})\)
5: \(\triangleright\) Upward-facing points of concavity along contour of top-most connected component, sorted clockwise.
6: for \(i = 0\) to \(|Q| - 1\) do
7: \(\left(x_1, y_1\right) = q^{(i-1)} \mod |Q|\); \(\left(x_2, y_2\right) = q^i\); \(\left(x_3, y_3\right) = q^{(i+1)} \mod |Q|\)
8: if \(y_2 \geq y_t\) then
9: \(d' \leftarrow \text{BLACK-LINE-DRAWING}(d', x_2, y_2, x_t, y_t)\)
10: else
11: \(d' \leftarrow \text{BLACK-LINE-DRAWING}(d', x_2, y_2, x_t, y_2)\)
12: end if
13: \((o_0, o_1) \leftarrow \text{CONNECTED-COMPONENT}(d')\)
14: \(\triangleright\) \(o_0, o_1\) are two connected components split from \(d'\).
15: Let \(w_0\) be the distance from the center of \(o_0\) to \((x_t, y_t)\).
16: Let \(w_1\) be the distance from the center of \(o_1\) to \((x_t, y_t)\).
17: if \(w_0 > w_1\) then
18: \(\text{PUSH}(O', o_0)\)
19: \(d' \leftarrow o_1\)
20: else
21: \(\text{PUSH}(O', o_1)\)
22: \(d' \leftarrow o_0\)
23: end if
24: end for
25: \(N_l \leftarrow |O'|\)
26: while \(|O'| \neq 0\) do
27: \(l \leftarrow |O'|\)
28: \(P_l \leftarrow \text{REPROJECTION-TO-3D-SPACE}(P, \text{POP}(O'))\)
29: \(\triangleright l\) is a leaf point cloud number \(l = 1, 2, 3, \ldots, N_l\).
2.4.1. Stem diameter extraction

We extracted three values related to the stem diameter: the stem width along the major axis of the elliptical cross-section, the width orthogonal to the major axis of elliptical cross-section, and the stem height.

$$(h_1, h_2) = \textsc{EllipseSize}(e_3),$$

$$h_3 = \textsc{MinimumYValue}(P_{N_{i-1,9}}) - y_0,$$
where \( h_1 \) and \( h_2 \) are the major axis and minor axis lengths of the ellipse \( e_3 \) previously estimated that represented the cross-section of the stem. To get the height of the stem, assuming the y-coordinate of the bottom-most point in the point cloud was \( y_0 \), and the leaf point clouds \( P_{l,9} \) were sorted by minimum y value, we calculated height as the difference between the bottom-most y coordinate of the second leaf from the top of the plant and \( y_0 \). We used the second leaf from the top because it was typically mature, terminating near the top of the stem, whereas the top-most leaf was typically immature and partly occluded.

2.4.2. Leaf length and leaf area extraction

We calculated the length and area of each leaf by computing the boundary polygon of its point cloud (mapped to a surface by MLS). The leaf length was calculated as half the boundary polygon’s perimeter, and the area was calculated as the sum of the areas of the triangles in a tessellation of the boundary polygon, according to Algorithm 4.

\[
H_{l,4} = \text{LEAF-LENGTH-AND-LEAF-AREA-EXTRACTION}(P_{l,9}, t_x)
\]

where \( P_{l,9} \) is the point cloud for leaf \( l \), \( t_x \) is the spacing of the strips used for tessellation of the leaf surface. \( H_{l,4} \) contained the estimated leaf length and leaf area for leaf \( l \).

2.4.3. Angle extraction

To calculate the angle at which each leaf extended from the stem, we first fitted a plane to the leaf points then calculated the angle between the plane and the center line of the elliptical cylinder model of the stem. For plane fitting, we used the Random Sample Consensus (RANSAC) method (Hartley and Zisserman, 2000). The method iteratively selects a sample of three points, computes the plane spanning those points, determines the number of
inlier points, and repeats the process for k iterations, keeping a record of the best plane obtained thus far. k, the number of RANSAC iterations, is defined as follows:

\[ k = \frac{\log(1 - \rho)}{\log(1 - w^n)} \]

where \( p \) is the desired probability of finding the best plane by sampling three inliers and \( w \) is the current estimate of the probability that a single sample from the point set is an inlier (updated on each iteration as the maximum inlier ratio observed thus far). The sampling procedure began with a large initial value for \( k \) that was decreased as the observed inlier ratio \( (w) \) increased and continued until the number of iterations exceeded \( k \).
invalid model parameters, and return the best plane model.

Algorithm 4 LEAF-LENGTH-AND-LEAF-AREA-EXTRACTION

Input: \( \{P_l\}_{l=1,N_l} \): a set of leaf point clouds indexed by leaf number \( l \).
Input: \( t \): threshold for width distance.
Output: \( \{H_l\}_{l=1,N_l} \): estimated leaf lengths, indexed by leaf number \( l \).
Output: \( \{H'_l\}_{l=1,N_l} \): estimated leaf areas, indexed by leaf number \( l \).

1: for each \( P_l \) do
2: \[ P_{x_{\text{min}}} = (x_{\text{min}}, y_{\text{min}}, z_{\text{min}}) \leftarrow \text{MINIMUM-X}(P_l) \]
3: \[ P_{x_{\text{max}}} = (x_{\text{max}}, y_{\text{max}}, z_{\text{max}}) \leftarrow \text{MAXIMUM-X}(P_l) \]
4: \( P_{\text{bottom}} \leftarrow \emptyset \)
5: ADD\( (P_{\text{bottom}}, P_{x_{\text{min}}}) \) \( \triangleright \) Add point \( P_{x_{\text{min}}} \) from \( P_l \) with minimum \( x \) value to \( P_{\text{bottom}} \)
6: \( i \leftarrow 0 \)
7: while \( (x_{\text{min}} + t \times i) < x_{\text{max}} \) do
8: \( P_0 \leftarrow \emptyset \)
9: for each \( P^j \in P_l \) do
10: \( \quad \text{if } \left( (x_{\text{min}} + t \times i) \leq x_j \leq (x_{\text{min}} + t \times (i + 1)) \right) \) then
11: \( \quad \text{ADD}(P_0, P^j) \)
12: \( \quad \)end if
13: \( \)end for
14: ADD\( (P_{\text{bottom}}, \text{MINIMUM-Z}(P_0)) \) \( \triangleright \) Add point from \( P_0 \) in slice \( i \) with smallest \( z \)
15: ADD\( (P_{\text{top}}, \text{MAXIMUM-Z}(P_0)) \) \( \triangleright \) Add point from \( P_0 \) in slice \( i \) with largest \( z \)
16: \( i \leftarrow i + 1 \)
17: end while
18: \( P_{\text{polygon}} \leftarrow P_{\text{bottom}} \)
19: ADD\( (P_{\text{polygon}}, P_{\text{top}}) \)
20: \( P_{\text{boundary}} \leftarrow 0 \)
21: for \( j = 0 \) to \( |P_{\text{polygon}}| - 2 \) do
22: \( \quad \text{boundary} \leftarrow \text{boundary} + \text{DISTANCE}(P_{\text{polygon},j}, P_{\text{polygon},j+1}) \)
23: end for
24: \( H_l \leftarrow \text{boundary}/2 \)
25: \( H'_l \leftarrow \text{POLYGON-AREA}(P_{\text{polygon}}) \) \( \triangleright \) calculate polygon area by triangulation method
26: end for

\[ H_{l,5} = \text{ANGLE(PLANE-FITTING}(P_{l,9}), L) \]

\( H_{l,5} \) is the angle between leaf number \( l \) (represented by point cloud \( P_{l,9} \) and the vertical stem center line \( L \)).
2.5. 3D holographic visualization

We ran the non-uniform rational B-splines (NURBS) algorithm (Wang et al., 2006) for each plant part separately, then we rendered them together.

\[
S_1 = \text{NURBS}(P_6), \quad S_{l,2} = \text{NURBS}(P_{l,9}),
\]

where \( S_1 \) is a surface model of the stem, and \( S_{l,2} \) is a surface model of leaf point cloud \( P_{l,9} \).

Figure 8 shows processes involved in automatic phenotyping and 3D holographic visualization and their corresponding outputs.

3. Experiments and results

In this section, we provide detailed descriptions of our experimental setup and results. We first provide details of how the experimental screening station described in Section 2 was set up and how the plant data were acquired. We provide a detailed description of the parameters used for point cloud data filtering and merging, stem segmentation, leaf segmentation, phenotype extraction, and 3D holographic visualization. We provide detailed per-plant and per-leaf results for each phenotype. Then we analyze the correlation between the ground truth phenotype measurements and the estimates from the 3D modeling procedure. We conclude with an overall summary of the results in Table 3.
Figure 8: Processes involved in automatic phenotyping and 3D holographic visualization and their corresponding outputs.
3.1. Point cloud data filtering and merging

Each plant was placed on the turntable, one by one. We acquired point clouds at every 5° degrees of rotation, returning 72 views of each plant, with θ = 0°, 5°, 10°, …, 355° sequentially. In the conservative per-view point filtering step, we used k = 10 nearest neighbors and a pairwise distance z-score threshold of α = 10. This resulted in removing an average of 0.2% of the points in each point cloud as noise. After an initial rotation of each point cloud using the angle at which it was acquired and the center of rotation p₀ fixed at the calibration, we ran ICP with maximum number of iterations m = 100, maximum rotation threshold r = 0.256°, and a maximum translation threshold t = 0.0003 m. In the more aggressive merged point cloud filtering step, we used k = 30 and α = 0.05. This resulted in removing a total of 18% of the points in the combined point cloud.

3.2. Stem segmentation

Based on the merged and filtered point cloud, we estimated each plant’s stem as described in Section 2. We sliced the merged point cloud along the y (vertical) direction into 50 slices (Nₛ = 50, s ∈ 0, 1, 2, …, 49) with c = 0.002 m. We converted each slice into a binary image with pixel size 0.1 mm.

Then, we applied the closing operation with a disc-shaped structuring element d₁ with radius 20 pixels. We applied contour tracing to each connected component not smaller than tₖ = 250 pixels. After least squares estimation of ellipses for each contour, we performed stem ellipse linking with a linking threshold of 20%. In the cylinder extrusion step, we used tₘ = 4 for the large area threshold and tₙ = 3 for the small area threshold, i.e., the large area
threshold and small area threshold were four times and three times that of the cross-section of the stem represented by ellipse e₃, respectively. An example of the overall procedure is shown in Figure 9.

Figure 9: Stem fitting example: (a) Merged point cloud with estimated ellipses in each slice and estimated stem top and stem bottom. (b) Close up view at the estimated stem top. (c) Estimated stem elliptical cylinder. Ellipses linked to the stem are shown in red, and ellipses not linked are shown in blue. See Section 2 for details.

3.3. Leaf segmentation

For each plant, we performed leaf segmentation as described in Section 2. When projecting points onto the x – y plane to create 2D binary images, we used a pixel size of
0.0001 m. We did a simple test on some pixel sizes and got this is the proper one. For morphological operations, we used a disc-shaped structuring element with radius 20 pixels. An example result is shown in Figure 10.

Figure 10: Leaf segmentation example.

3.4. Phenotype data extraction

After segmenting the leaves of each plant, we performed MLS leaf surface fitting, attached each leaf to the stem cylinder, estimated phenotypes, and generated the visualization according to the methods of Section 2. For MLS leaf surface fitting, we used a radius $t_r = 0.05$ m. An example output of the leaf surface fitting procedure is shown in Figure 11 (c–d). To compute the leaf boundary polygons required for leaf length and leaf area extraction, we used a longitudinal threshold $t_x = 0.002$ m. An example boundary polygon for the same leaf is shown in Figure 11 (e–f).

For leaf angle extraction, we used the MLS leaf points within a radius of 0.05 m from the stem’s vertical center. For the RANSAC, we set $p=0.99$ for probability that the algorithm produces a useful result, $n=3$ for needed points to estimate a model, and max iterations=500.
for allowed iterations for invalid model parameters. As a baseline for leaf angle extraction, we also ran the point cloud skeletonization method of Jalba et al. (2013) and then manually measured the angle of each leaf according to the skeleton. An example of leaf angle measurement with the MLS points and baseline skeletonization method is shown in Figure 12.
Figure 11: Leaf modeling example. (a-b) Two views of original point cloud of one leaf. (c-d) MLS point cloud of the leaf in (a-b) from same viewpoints. (e-f) Boundary of the MLS point cloud in (c-d) from same viewpoints. (g-h) NURBS surface of the leaf in (a-b) from same viewpoints.

Figure 12: Angle measurement example. (a) Angle measurement by plane fitting. (b) Angle measurement from plant skeleton.

<table>
<thead>
<tr>
<th>Plant No.</th>
<th>Major Axis (cm)</th>
<th>Minor Axis (cm)</th>
<th>Height (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ground Truth</td>
<td>System Error</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>1</td>
<td>1.12</td>
<td>1.26</td>
<td>12.50</td>
</tr>
<tr>
<td>2</td>
<td>1.32</td>
<td>1.39</td>
<td>5.30</td>
</tr>
<tr>
<td>3</td>
<td>1.49</td>
<td>1.35</td>
<td>9.40</td>
</tr>
<tr>
<td>4</td>
<td>0.79</td>
<td>0.82</td>
<td>3.80</td>
</tr>
<tr>
<td>5</td>
<td>0.93</td>
<td>0.85</td>
<td>8.60</td>
</tr>
</tbody>
</table>

3.5. 3D holographic visualization

The 3D holographic visualization was generated from stem point cloud and leaf point cloud as explain in Section 2. The NURBS surface calculated for the same leaf is shown in Figure 11 (g-h).
3.5. Results

Figure 13 shows the complete stem and leaf segmentation, and visualization results for all five corn plants used in our experiments. The method was successful at properly segmenting the leaves from the stem and generated an acceptable 3D holographic visualization in every case.

Table 1 shows the stem phenotype characterization results. The average error levels for the height and the major axis length measurement were below 10%, while it was 15.2% in the case of minor axis length which was driven mainly by a 2 mm. error on plant #4. This occurred because the two bottom-most ellipses in the linkage set for plant #4 were located very close to the first leaf. Since we only used the two bottom-most ellipses in the stem cross-section calculations, proximity to the leaf caused overestimation of the stem width. Figure 14 shows the relationship between the system calculated and the ground truth stem measurements. Although there are only five points, it is clear that the correlations are strong for the major axis length and stem height but weaker for the stem minor axis. We performed statistical tests on the correlations with a null hypothesis that Pearson’s $r = 0$ in each case, with a Type I error rate of 0.05, and found significant correlations for the major axis and stem height measurements. The correlation between system calculated stem minor axis length and the ground truth was not significant, preventing us from rejecting the null hypothesis for stem minor axis.

The leaf length predictions were accurate, with an average error of 10.25%, whereas the leaf area estimates were less accurate, with an average error of 21.89%. In the worst case, the last leaf on plant #2 (second row of Figure 13), the leaf area was underestimated by 51%. As can be seen from the figure, the length of this leaf was underestimated because the stem
height was overestimated, and the leaf area was further underestimated most likely due to the leaf

Figure 13: Segmentation and 3D visualization for all five plants.
Figure 14: Stem phenotype correlation analysis.
rolling over on itself near the tip. Comparison plots are shown in Figure 15. We see that whereas leaf length estimates are quite accurate, the leaf area measurements tend to underestimate the actual leaf area. The correlations for leaf area and leaf angle were both significant in our correlation test.

Finally, per-leaf results for leaf angle estimation with our method (plane fitting) and the baseline method (skeletonization) are shown in Table 2, and comparison graphs are shown in Figure 16. The average angular error was 11% for our method but it was 80% for the skeleton-based method. The reason for the poor performance of skeletonization is clear from Figure 12 (b): the stem tends to bend toward leaves near the junctions, resulting in underestimation of leaf angle. Despite the relatively higher error in leaf angle estimates by the skeletonization method, both the plane fitting and skeletonization methods achieve correlations with the ground truth significantly different from 0.

Overall, referring to the summary in Table 3, we found high correlations, with $R^2 \geq 0.84$, $p < 0.05$, for all phenotypes except for the stem minor axis width ($R^2 = 0.74$, $p = 0.061$). The alternative leaf angle prediction based on the skeleton method was less correlated.
with the ground truth but the correlation was significantly different from 0 ($R^2 = 0.39$, $p = 0.001$).

![Figure 16: Leaf angle results from plane fitting and skeleton methods.](image)

**Figure 16:** Leaf angle results from plane fitting and skeleton methods.

**Table 2: Leaf angle results.**
4. Discussions and Conclusions

In this paper, the results of our methods for characterizing morphological traits of corn plant seedlings through holographic 3D holographic reconstruction clearly demonstrate
the great promise of this automatic plant phenotyping approach. We found the following error rates for different phenotypes: stem major axis 7.92%, stem minor axis 15.20%, stem height 7.45%, leaf area 21.89%, leaf length 10.25%, and leaf angle 11.09%. Correlation analysis revealed that estimates of stem major axis, stem height, leaf area, leaf length, and leaf angle were significantly correlated with the ground truth. Stem minor axis estimates were not significantly correlated with the ground truth but this could be due to the small sample size (only have stem measurements). The visualization of the models appeared realistic, wherein the leaf counts and orientations matched the ground truth, and the stem directions also matched the ground truth (see Figure 13). We found that some outlier points that were not removed in the filtering steps affected leaf length and width estimates, leading to overestimates of leaf area. These estimates could be improved with more discriminative
filtering at leaf boundaries. The leaf area estimates were worse for the topmost leaves of the plant. This was because the topmost leaves were smaller than the other leaves (so that the same level of absolute error led to a larger relative error) and because the topmost leaves typically exhibit strong leaf rolling characteristics, making their full surface reconstruction after cutting and unrolling less accurate. These effects can be seen in Figure 17. If the topmost leaves were left out of the analysis, we would obtain substantially lower overall error rates for leaf area estimates.

We believe that the most important overall factor affecting error rates is the intrinsic noise occurred in the 3D point cloud data sampling process. For example, error in leaf area measurement was found to be more significant, which was largely due to the fact that corn leaves were not exactly flat surfaces, and the ToF 3D sensor did not have sufficient depth resolution to capture the small variations in leaf surface. Similarly, due to sensor noise, when measuring smaller quantities such as the stem minor axis width, the sensor system produced a larger relative error rate.

The main objective of this research was to showcase the methodology for plant reconstruction and to evaluate its accuracy. To fully characterize the morphological traits of a plant species, plants at different growth stages would have to be considered. Nevertheless, the methodology presented here will be readily applicable, with minor adjustment, to plants at a variety of sizes and growth stages. In future work, we will experiment with a larger data set including a wider variety of plant growth stages. To accommodate plants at different growth stages (V2--V10), we will investigate more adaptive methods for setting parameters, integrate point clouds acquired at multiple heights for taller plants, and generalize the image processing algorithms as necessary.
We understand that the current study only considered healthy plants. In plant phenotyping, besides measuring trait characteristics of healthy plants, it is also common to measure plants under various kinds of stresses. Characterizing stressed plants would likely require different sensors such as color and spectral sensors and correspondingly different classification algorithms.

Another possible direction for future work would be to adapt the approach to different plants, such as rice, barley, and wheat, which may have similar phenotyping requirements.

Figure 17: Visible, rolled, and hidden parts of the top leaf of plant #1. The appearance of the topmost leaf on the plant (left) is very different from its appearance after cutting it off of the stem and spreading it flat (right).

Appendix A. Libraries and functions used

We implemented the methods described in this paper in C/C++, making use of two open source libraries: PCL 1.7.0 and OpenCV 2.4.5. Table A.5 lists the functions imported from the PCL and OpenCV libraries and the functions we built from scratch.
Acknowledgments

S. Chaivivatrakul was supported by the Thailand National Science and Technology Development Agency (NSTDA). The lab facility used in this project at Iowa State University (ISU) was partly supported by the project “Automated Phenotyping for Biomass Crops” sponsored by the Grow Iowa Values program at ISU.

<table>
<thead>
<tr>
<th>Libraries</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCL 1.7.0</td>
<td>FILTER</td>
</tr>
<tr>
<td></td>
<td>ITERATIVE-CLOSEST-POINT</td>
</tr>
<tr>
<td></td>
<td>LEAF-SURFACE-ESTIMATION</td>
</tr>
<tr>
<td></td>
<td>NURBS</td>
</tr>
<tr>
<td></td>
<td>PLANE-FITTING</td>
</tr>
<tr>
<td>OpenCV 2.4.5</td>
<td>BLACK-LINE-DRAWING</td>
</tr>
<tr>
<td></td>
<td>CLOSING</td>
</tr>
<tr>
<td></td>
<td>CONNECTED-COMPONENT</td>
</tr>
<tr>
<td></td>
<td>CONTOUR</td>
</tr>
<tr>
<td></td>
<td>CONVEXITY</td>
</tr>
<tr>
<td></td>
<td>LEAST-SQUARES-ELLIPSE-FITTING</td>
</tr>
<tr>
<td></td>
<td>LEAST-SQUARES-LINE-FITTING</td>
</tr>
<tr>
<td></td>
<td>OPENING</td>
</tr>
<tr>
<td>Our</td>
<td>ANGLE</td>
</tr>
<tr>
<td>Implementation</td>
<td>ATTACH-LEAF-TO-CYLINDER</td>
</tr>
<tr>
<td></td>
<td>BOUNDING-BOX</td>
</tr>
<tr>
<td></td>
<td>DISTANCE</td>
</tr>
<tr>
<td></td>
<td>ELLIPSE-SIZE</td>
</tr>
<tr>
<td></td>
<td>ELLIPTICAL-CYLINDER-POINT-CLOUD</td>
</tr>
<tr>
<td></td>
<td>LEAF-CUTTING</td>
</tr>
<tr>
<td></td>
<td>LEAF-LENGTH-AND-LEAF-AREA-EXTRACTION</td>
</tr>
<tr>
<td></td>
<td>LINKING</td>
</tr>
<tr>
<td></td>
<td>MAXIMUM-Z</td>
</tr>
<tr>
<td></td>
<td>MINIMUM-Y-ASCENDING-SORT</td>
</tr>
<tr>
<td></td>
<td>MINIMUM-Y-VALUE</td>
</tr>
<tr>
<td></td>
<td>MINIMUM-Z-Z</td>
</tr>
<tr>
<td></td>
<td>POLYGON AREA</td>
</tr>
<tr>
<td></td>
<td>PROJECTION-TO-XY-PLANE</td>
</tr>
<tr>
<td></td>
<td>PROJECTION-TO-YZ-PLANE</td>
</tr>
<tr>
<td></td>
<td>REPROJECTION-TO-3D-SPACE</td>
</tr>
<tr>
<td></td>
<td>ROTATION</td>
</tr>
<tr>
<td></td>
<td>SAMPLE</td>
</tr>
<tr>
<td></td>
<td>TRANSFORM</td>
</tr>
<tr>
<td></td>
<td>UPWARD-CONCAVITY</td>
</tr>
<tr>
<td></td>
<td>VERTICAL-STEM-CENTER</td>
</tr>
</tbody>
</table>
References


