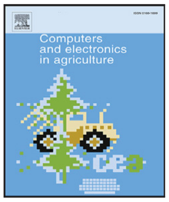
Computers and Electronics in Agriculture 213 (2023) 108248

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Original papers

3D reconstruction of plants using probabilistic voxel carving

Jiale Feng [a], Mojdeh Saadati [a], Talukder Jubery [b], Anushrut Jignasu [b], Aditya Balu [b], Yawei Li [c], Lakshmi Attigala [d], Patrick S. Schnable [c], Soumik Sarkar [a,b], Baskar Ganapathysubramanian [b,*], Adarsh Krishnamurthy [b,*]

[a] *Department of Computer Science, Iowa State University, Ames, 50011, IA, USA*
[b] *Department of Mechanical Engineering, Iowa State University, Ames, 50011, IA, USA*
[c] *Department of Agronomy, Iowa State University, Ames, 50011, IA, USA*
[d] *Environmental Health and Safety, Iowa State University, Ames, 50011, IA, USA*

ARTICLE INFO

Keywords:
3D reconstruction
Probabilistic voxel carving
GPU computing
Plant phenotyping
Computer vision

ABSTRACT

We propose a probabilistic voxel carving algorithm to efficiently reconstruct 3D models of maize plants and extract leaf traits for phenotyping. Traditional voxel carving algorithm is restricted to a limited number of views and usually requires multiple coordinated cameras in the imaging setup. They are also not robust small movements of the object, which introduce noise into the data. These imperfections in data collection can lead to large regions of the object being carved away during the voxel carving process, leading to incomplete and disjoint objects. We have developed a probabilistic voxel carving algorithm to overcome these challenges. In this approach, instead of carving out or keeping a voxel in a binary manner, we associate a probability of a voxel corresponding to it being part of the plant. We then use a user-defined probability cutoff to obtain the final voxelized plant geometry. We optimize the data collection procedure by adopting a rotating base to hold the plant and then capturing videos of the rotating plants, thereby obtaining an arbitrary number of views by extracting the image frames. Additionally, we leverage GPU computing to implement our voxel carving and trait extraction pipeline for a large dataset with over 1000 maize plants with high voxel resolutions (such as 1024^3). Our results demonstrate that our algorithm is robust and can handle an arbitrary number of views, and can automatically extract plant traits such as the number of leaves and leaf angles. Our approach shows that 3D reconstructions of plants from multi-view images can accurately extract multiple phenotypic traits, enabling better plant breeding programs.

1. Introduction

Current trends in global crop yield improvement need to increase to meet the expected demand for crop production, which is projected to double from 2005 to 2050 (Tilman et al., 2011; Ray et al., 2013). Previous studies have established that leaf traits and other features of canopy architecture are closely related to crop yield (Westgate et al., 1997; Hammer et al., 2009). Developing high-yielding crops requires understanding the interplay among various traits across diverse genotypes, environments, and crop management systems to understand the genetic underpinning resulting in this diversity. This information can assist breeders in developing crops with specific architectural characteristics. But the optimal plant architectures under specific environments and crop management systems are unknown. Absent these ideotypes (Mock and Pearce, 1975), breeders must select genotypes based on empirical crop performance data. However, traits such as leaf angle, plant height, total leaf number, etc., can be inputs for process-based crop models (Kimball et al., 2023) that could guide breeding decisions. In addition, accurate 3D reconstructions of the crop canopies of diverse genotypes would enable the modeling of traits such as light interception that could also contribute to breeding decisions, potentially further increasing rates of genetic gain.

Manual trait extraction and analysis can be tedious and time-consuming for plant phenotyping tasks. High-throughput phenotyping utilizes various methods that use computer vision and 3D reconstruction techniques. Computer vision techniques extract traits from multiple 2D images. However, these techniques may not estimate the plant traits correctly due to projecting the 3D structures onto 2D images (McCormick et al., 2016). Alternatively, extracting traits from the 3D reconstruction of the plant is shown to be more accurate. However, obtaining the 3D geometry of the plant is challenging. LIDAR scanning is the most common technique to obtain a 3D point cloud of the plants, which can then be used for trait extraction (Paulus et al.,

[*] Corresponding authors.
E-mail addresses: baskarg@iastate.edu (B. Ganapathysubramanian), adarsh@iastate.edu (A. Krishnamurthy).

https://doi.org/10.1016/j.compag.2023.108248
Received 25 April 2023; Received in revised form 27 August 2023; Accepted 9 September 2023
Available online 28 September 2023

2014; Garrido et al., 2015; Thapa et al., 2018). However, a high-quality LIDAR sensor is significantly more expensive than a digital single-lens reflex (DSLR) camera with high resolution and imaging capabilities. In addition to cost-related concerns, the lack of data and noise in certain regions of plants may cause the reconstruction algorithm to fail (Gaillard et al., 2020). A possible alternative is to collect multiple views of the plant and reconstruct the 3D geometry from these views. For example, the structure from motion (SFM) method (Quan et al., 2006; Lou et al., 2014) uses conventional moving RGB cameras to reconstruct the plant geometry. However, this method has the same reconstruction problems as using LIDAR scanning.

Another example of 3D reconstruction is voxel carving or space carving, a well-developed approach for 3D visual hull reconstruction from multiple views of an object using silhouettes or segmentation masks (Schultze et al., 2012; Kutulakos and Seitz, 2000; Yamazaki et al., 2007). This method has been successfully used in plant phenotyping to reconstruct plants from 2D images (Prakash and Robles-Kelly, 2008; Tabb, 2013; Klodt and Cremers, 2015; Golbach et al., 2016; Cabrera-Bosquet et al., 2016; Roussel et al., 2016; Scharr et al., 2017; Tross et al., 2021). Voxel carving is suitable for a typical plant science lab setting because the camera positions are fixed and can be carefully calibrated. Voxel-carving-based approaches also perform well in reconstructing small features or parts of plants such as seeds (Klodt and Cremers, 2015; Golbach et al., 2016; Roussel et al., 2016). However, while reconstructing larger plants such as maize, voxel carving can only be performed with a limited number of views (Cabrera-Bosquet et al., 2016; Tross et al., 2021). It is not robust for reconstructing large 3D structures from many views, such as 15 or more. This limitation arises from small deviations in the location of the parts of the plant between the views, particularly if the views are not captured using multiple coordinated cameras. Previous researchers have explored statistical methods to improve the carving results, given that this uncertainty exists during the data collection. Prakash and Robles-Kelly (2008) presented a semisupervised approach to space carving with only one manually obtained silhouette. Tabb (2013) used silhouette probability maps as input and proposed a local minimum search algorithm to reconstruct thin objects such as leaf-less plants. In our case, the uncertainty mainly originates due to the vibration of leaves during the video-capturing process and imperfections in the segmentation process.

To overcome this limitation, we have developed a probabilistic voxel carving algorithm. In this algorithm, instead of carving out or keeping a voxel in a binary manner, we associate a probability of a voxel corresponding to it being part of the plant. We then use a user-defined probability cutoff to obtain the final voxelized plant geometry. The probabilistic voxel carving algorithm enables the development of a streamlined data collection pipeline. We can obtain an arbitrary number of views by mounting the plant on a rotating platform and using a fixed camera. This approach significantly reduces the setup cost without limiting the number of views we can obtain. This approach is inspired by previous works using either a rotating platform or a conveyor belt for automatic imaging (Junker et al., 2015; Ge et al., 2016; Cabrera-Bosquet et al., 2016; Thapa et al., 2018).

Even though voxel carving is faster than SFM at low resolutions (Gaillard et al., 2020), it is still challenging to scale it up to the gigavoxel (1024^3) range for larger plants. In recent years, there has been some work on developing an octree data structure (Klodt and Cremers, 2015) for accelerating the process. In another recent study on the 3D reconstruction of plant shoots, Scharr et al. (2017) demonstrated that octrees could speed up the computations by two orders of magnitude compared to less efficient implementations. Simultaneously, GPU computing has emerged as a powerful tool for accelerating geometric operations. The voxel carving algorithm is already embarrassingly parallel, and hence GPU computing can be a promising approach for accelerating the 3D reconstruction without the need for specialized data structures such as octrees. In our work, we use the GPU to compute the transformations for each voxel in parallel, which significantly speeds up the voxel carving process and ultimately improves the efficiency of plant phenotyping studies. However, one of the challenges in developing a GPU algorithm without specialized data structures is that the memory requirements are high, and a large resolution voxel grid (such as 1024^3) may not fit into the GPU memory. To deal with large resolutions, we perform space partition on the voxel grid and perform the voxel carving in batches on the GPU.

After reconstructing the plant model using GPU-accelerated probabilistic voxel carving, we extract morphometric traits (Gaillard et al., 2020), including the number of leaves, the leaf-stalk angles, the distribution of leaves, and inter-node distances. These traits can be used to develop functional, structural models to identify the optimum plant architecture for a specific location and environment (Soualiou et al., 2021). In maize, for example, leaf orientation has been shown to play a role in crop productivity. In an investigation, Pendleton et al. (1968) manipulated the leaf angle of a commercial hybrid and demonstrated that grain yield increased with a more upright leaf orientation. In another study, Pepper et al. (1977) evaluated nine maize inbreds with different leaf orientations, revealing that genotypes with upright leaves had yield advantages only at high leaf area indices. Additionally, apparent photosynthesis measurements on individual corn leaves showed that the relative efficiency of CO_2 fixation per unit of incoming sunlight increased as leaf angle decreased. These researches suggest that leaf orientation and canopy shape are important traits to consider in breeding programs.

To summarize, we developed a streamlined process for reconstructing the 3D plant geometry from a video of a rotating plant using probabilistic voxel carving. The main contributions of this work are:

1. A probabilistic voxel carving method to reconstruct the 3D model of a plant efficiently, leveraging multiple views captured by recording a video of the plant rotating at a constant speed. Compared to earlier approaches, this method can perform robust 3D reconstruction with an arbitrarily large number of views.
2. Robustness of the algorithm to noise in the data. We achieve this by introducing a probability threshold in the algorithm and image processing methods, such as dilation on the binary silhouette masks.
3. Acceleration of the voxel carving and trait extraction pipeline using GPU computing and voxel grid partitioning, which enables gigavoxel carving on a large dataset with over 1000 individual plants.
4. Automatic trait extraction from the reconstructed plants for a large-scale plant dataset.

The source code of the whole pipeline, including data preprocessing, voxel carving, and trait extraction, will be available as open-source software.

The remainder of this paper is structured as follows. In Section 2, we provide a detailed exploration of each step in the 3D maize plant reconstruction pipeline, including data collection (Section 2.1), image projection (Section 2.2), voxel carving using our probabilistic algorithm (Section 2.3) the GPU accelerated version (Section 2.4), as well as trait extraction (Section 2.5). In Section 3, we present the results of validation experiments and performance evaluation. The paper concludes with a summary and discussion of future research directions in Section 4.

2. Methods

Our approach (Fig. 1) employing probabilistic voxel carving to obtain phenotypic traits of maize plants consists of capturing a video of the plant on a rotating base and extracting its frames to obtain the corresponding segmentation masks. Next, we implement our probabilistic voxel carving algorithm to obtain a 3D model of the plant. Finally, we skeletonize the 3D model and extract the traits of the plants. In the following sections, we briefly review the related background, explain the data collection process, and describe our probabilistic voxel carving algorithm in detail.

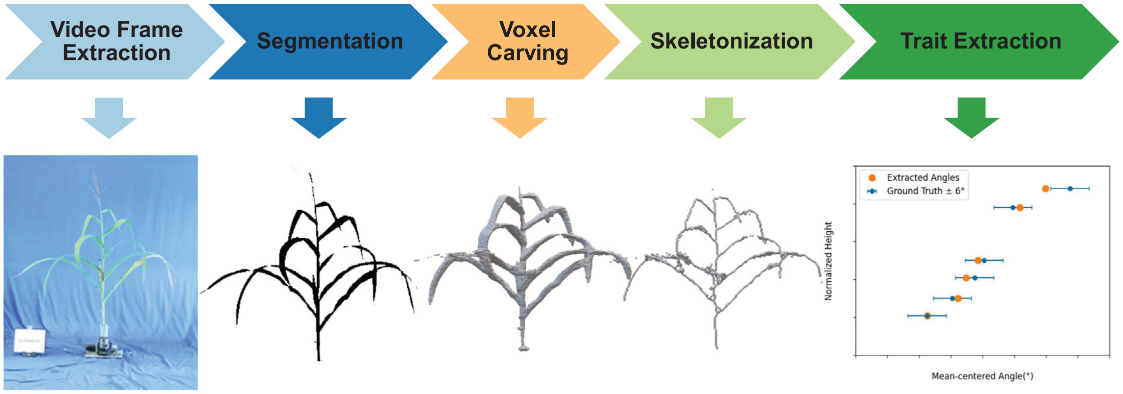
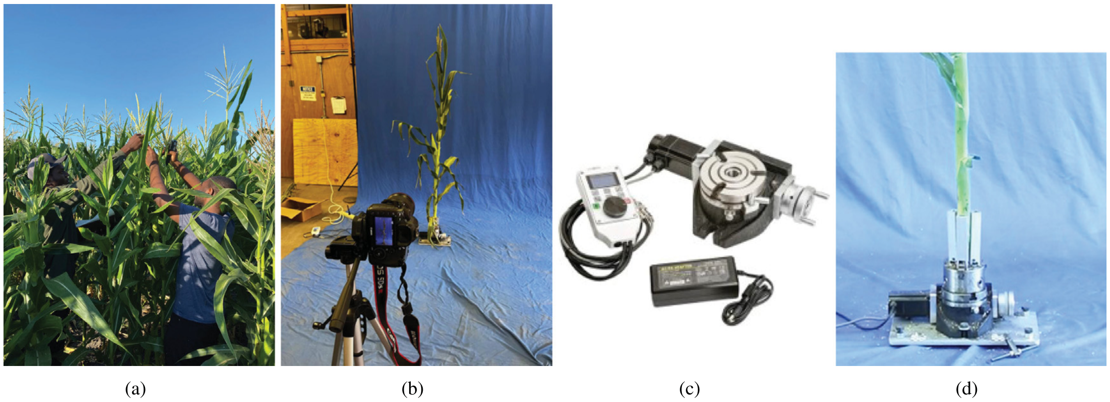


Fig. 1. Overview of our probabilistic voxel carving approach to obtain phenotypic traits. (1) Video Frame Extraction: We capture a video of the plant rotating inside to obtain multiple views based on the desired number of views and the rotation speed of the plant. (2) Segmentation: Segmentation is performed on the extracted frames in the HSV color space to obtain binary silhouette masks. (3) Voxel Carving: We apply our proposed probabilistic voxel carving algorithm to generate the 3D model of the plant. (4) Skeletonization: We post-process the reconstructed 3D model to compute the skeleton. (5) Trait Extraction: We extract the plant's phenotypic traits from the skeletonized model.



(a) (b) (c) (d)

Fig. 2. Maize plants harvesting and imaging station setup. (a) Labeling plants with unique ID tags during the harvesting process. (b) The setup for capturing videos of the maize plants. The imaging station is set up in a barn near the maize field. (c) The rotation base (Grizzly Industrial Inc., 2022). (d) The maize plant is fixed on the rotation base.

2.1. Data collection

The input to our probabilistic voxel carving algorithm is a video of the target maize plant rotating at a constant speed. We capture a complete video of the plant rotating at a steady speed, providing a continuous 360° view of the plant. The imaging studio setup for capturing the video is shown in Fig. 2. A custom fixture was designed and combined with a $4 - 3/8$ inches power rotary table to enable the plants to rotate. We use a digital single-lens reflex (DSLR) camera to capture the video. Our approach of using video capture significantly simplifies the data capture process, unlike other works that capture the different views of the plants from multiple viewpoints. We can extract the frames corresponding to different views from the captured video based on the rotation speed. Using our setup, we can extract one frame for every angle degree (1°). This setup enables us to obtain almost an arbitrary number of distinct views for performing the probabilistic voxel carving. Examples of the extracted frames at different angles are depicted in Fig. 3.

Voxel carving requires a binary mask of the plants from the images. We perform color segmentation on the extracted frames using the HSV color space to obtain the masks. The HSV color space provides a better representation of the base color regardless of the lighting changes in the environment, as it separates color information from the intensity. We start by converting the RGB images to HSV color space and then defining the range of the color we want to segment out, which is green for our maize plants. This segmentation yields binary masks of the images, where the green plant area is labeled as 1 (black color) while the other areas are labeled as 0 (white color). These binary masks are used as inputs for probabilistic voxel carving.

We must identify each leaf and its closest stalk segment to calculate the leaf-stalk angles during trait extraction. However, obtaining the stalk segment directly from the 3D model of the complete plant obtained after voxel carving using geometry-based methods is challenging. This is because the stalk could be bent in 3D space and might require careful segmentation to obtain the geometry. Therefore, we use a deep learning method to generate binary masks for the pseudo-stalk. These masks are then used to reconstruct the pseudo-stalk of the plant. To perform the stalk segmentation, we use a feature pyramid network (FPN, Lin et al. (2017)) architecture with ResNet34 (He et al., 2015) encoder. We use the pre-trained weights from the ImageNet dataset (Russakovsky et al., 2015) and the Dice loss function (Sudre et al., 2017). Please refer to Appendix for additional details on the architecture of the deep network and examples of the stalk segmentation.

To validate the accuracy of our trait extraction results, particularly the stalk-leaf angles, we manually measured the stalk-leaf angles for all leaves on a small set of maize plants selected at random. The leaf angle is defined as the angle between the vectors of the leaf and the stalk in their growing directions. We use a protractor to measure the leaf angle and document the raw data in degrees.

2.2. Image projection for voxel carving

An image is commonly represented as a discrete grid of pixels, whereas a volume can be represented as a grid of volume elements, referred to as voxels. Voxel carving is a 3D reconstruction technique

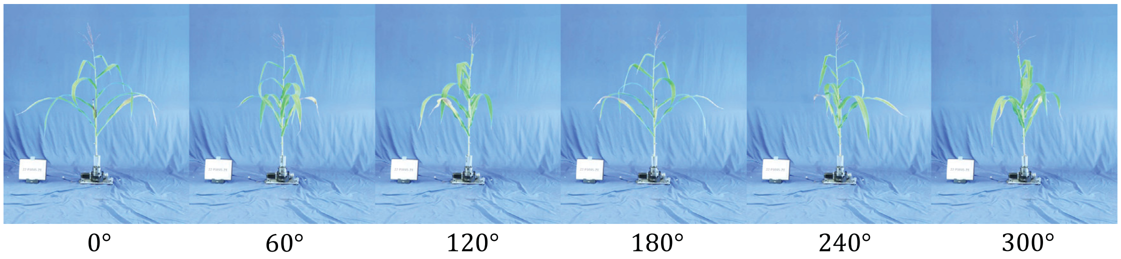


Fig. 3. Extracted frames from the captured video of the maize plant at different view angles.

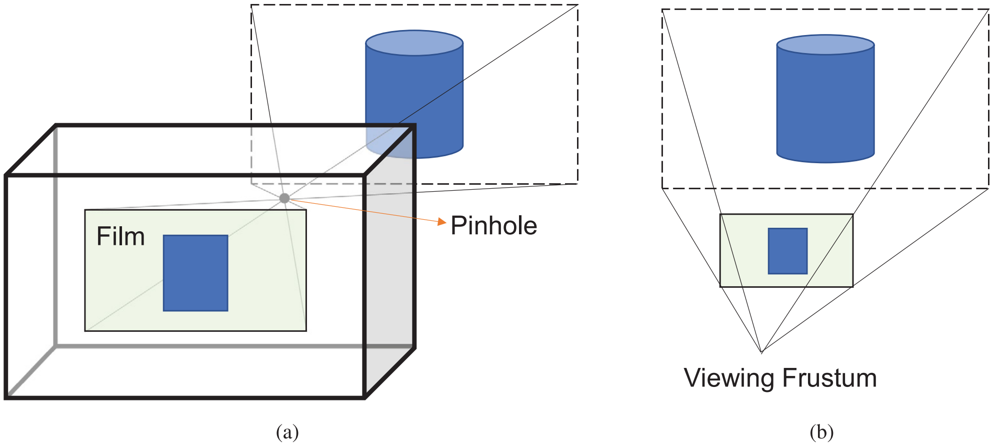


Fig. 4. The pinhole camera model and its corresponding viewing frustum.

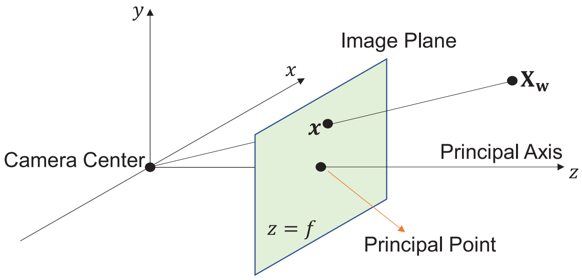


Fig. 5. The mathematical representation of the pinhole camera model.

that involves carving out voxels that do not belong to the object from a 3D voxel grid. In voxel carving, each camera view is processed to generate a segmentation mask, which typically involves thresholding techniques to create a binary mask that separates the object from the background. The segmentation masks generated from images are then projected into the 3D voxel grid to carve out portions of the object. Each voxel is assigned a binary value based on whether it is inside or outside the object, as defined by the mask.

Voxel carving utilizes the 2D projection transformation from 3D. We use the pinhole camera model to formulate the projection transformation that maps 3D points to 2D points. Fig. 4(a) illustrates the pinhole camera model, which can be simplified to a viewing frustum, as shown in Fig. 4(b). We obtain this frustum by taking the pinhole as the reference point and flipping the projected plane (film) to the same side of the target object.

The viewing frustum can be further abstracted and mathematically represented by a model shown in Fig. 5. Let $\mathbf{X_w}$ denote the 3D coordinates of a point in the world coordinate system, and let $\boldsymbol{x}$ denote its corresponding 2D coordinates in the image coordinate system. Using a camera matrix $\mathbf{P}$, we can relate 3D points to 2D image points. The camera matrix can project any 3D points to the image plane according to Eq. (1).

$$\boldsymbol{x} = \mathbf{P}\mathbf{X_w} \tag{1}$$

In general, as shown in Fig. 6, the pinhole camera model consists of three coordinate systems: the world coordinate system, the camera coordinate system, and the image coordinate system. The camera matrix $\mathbf{P}$ describes the transformations between these coordinate systems. Specifically, the camera matrix can be decomposed into the intrinsic matrix $\mathbf{K}$ and the extrinsic matrix $[\mathbf{R}|\mathbf{t}]$.

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \tag{2}$$

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\mathbf{X_c} = [\mathbf{R}|\mathbf{t}]\mathbf{X_w} \tag{4}$$

$$\boldsymbol{x} = \mathbf{K}\mathbf{X_c} \tag{5}$$

Here, the 3 × 4 extrinsic matrix $[\mathbf{R}|\mathbf{t}]$ specifies the transformation from points in world coordinates $\mathbf{X_w}$ to camera coordinates $\mathbf{X_c}$ (Eq. (4)). It consists of a 3 × 3 rotation matrix $\mathbf{R}$ and a 3 × 1 translation vector $\mathbf{t}$. On the other hand, the 3 × 3 intrinsic matrix $\mathbf{K}$ maps 3D camera coordinates $\mathbf{X_c}$ to 2D homogeneous image coordinates $\boldsymbol{x}$, as defined in Eq. (5). The intrinsic parameters of $\mathbf{K}$ include the focal length f_x and f_y as well as the principal point (p_x, p_y), both in pixels, as seen in Eq. (3).

The intrinsic and extrinsic parameters of the camera can be determined using camera calibration (Zhang, 2000). During camera calibration, a set of known 3D points in the world are imaged by the camera to obtain their corresponding 2D image coordinates. The camera matrix can be calculated by knowing the true 3D positions of these points and their observed 2D positions in the image. In our imaging station setup shown in Fig. 2(b), the locations of the camera and the rotating

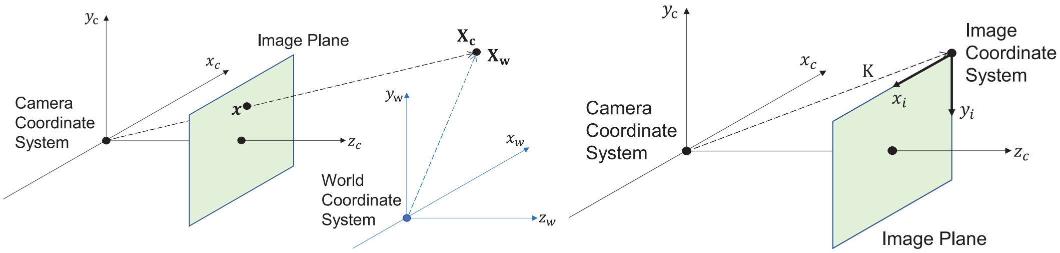


Fig. 6. The transformations from the world coordinate system to the camera coordinate system (left) and the camera coordinate system to the image coordinate system (right).

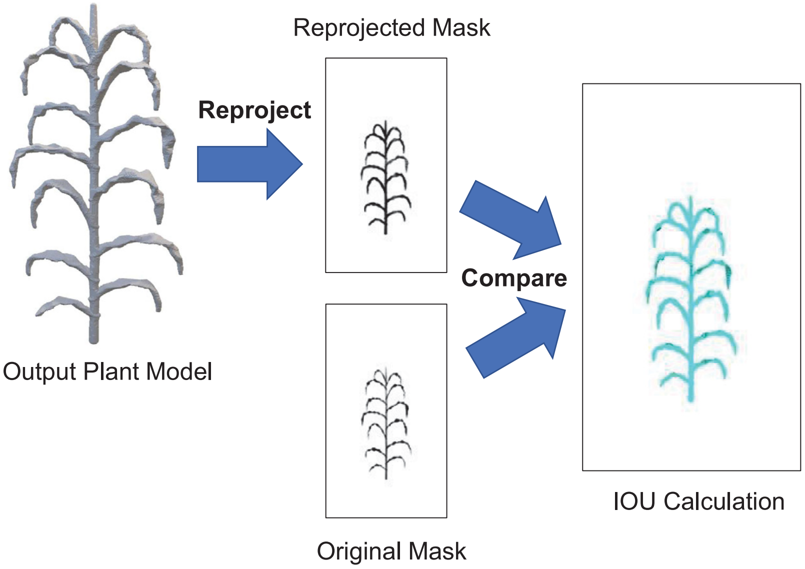


Fig. 7. Calculate IoU to validate the voxel carving result. We perform reprojection of the 3D model onto a 2D plane, with a randomly selected view angle that has not been used in the voxel carving process. Subsequently, we compare the resulting reprojected mask with the corresponding ground truth mask and compute their Intersection over Union (IoU) score.

base are fixed. Therefore, obtaining an accurate camera matrix through calibration is straightforward. For the extrinsic matrix, we can obtain the values based on the rotation angle of the plant relative to the first frame. The rotation matrix **R** is updated based on each view angle. Once we have the camera matrix **P**, we can use it to reconstruct the 3D object from its 2D images.

To validate and evaluate the accuracy of our output models, we reproject our output models back to the 2D plane and calculate the Intersection over Union (IoU) score between the reprojected masks and ground truth masks (Fig. 7). IoU, also known as the Jaccard index, is a common evaluation metric used in object detection and segmentation tasks to measure the overlap between the predicted bounding box/segmentation mask and the ground truth. The IoU score is calculated as the ratio of the intersection of the predicted bounding box/segmentation mask and the ground truth to the union of both. The formula for IoU can be expressed as:

$$\mathrm{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{6}$$

Here, the Area of Intersection is where the predicted bounding box/ segmentation mask overlaps with the ground truth, while the Area of Union is the total area covered by both the predicted bounding box/segmentation mask and the ground truth. Specifically, the Area of Intersection represents the number of object pixels correctly classified. The Area of Union, on the other hand, is the total sum of the number of object pixels correctly classified, the number of empty pixels classified as objects, and the number of object pixels classified as empty. The IoU score ranges from 0 to 1, where 0 indicates no overlap between the predicted and ground truth, and 1 indicates a perfect overlap. A higher IoU score, in our case here, indicates a better-reconstructed 3D model.

2.3. Probabilistic voxel carving algorithm

In this section, we provide a detailed description of our probabilistic voxel carving algorithm. Our probabilistic voxel carving approach addresses uncertainty in the 3D reconstruction process caused by noise, vibrations in the data, and the need to handle a large number of images. In contrast to traditional voxel carving methods (presented in Section 2.2) that rely on binary 0–1 output and may result in incomplete models due to input deformations or noise, our technique assigns a probability value between 0 and 1 to each voxel in the reconstruction. These probability values represent the likelihood of a voxel belonging to the reconstructed object, with higher values indicating greater confidence and lower values implying less certainty.

We begin the probabilistic voxel carving with a data preparation step that involves creating a set of binary segmentation mask images from multiple object views (We provide more information on segmentation in Section 2.1). Once the masks are obtained, we start the voxel carving algorithm by initializing a voxel grid with each voxel value set to a probability 1. Given the viewing angle, each voxel center is projected onto the 2D image plane where the 2D image (binary mask image) lies. The projected point is checked to determine whether it lies inside the object or not.

In the traditional voxel carving method, voxels that project to pixels inside the object contour in all images are classified as object voxels and kept, while others are deleted (or carved out). This is achieved by assigning a true or false label to each voxel (0 or 1 value). In contrast, our probabilistic voxel carving algorithm reduces the voxel probability value by a factor of s, where $0 < s \leq 1$, for voxels not projected inside the object instead of discarding them directly. The parameter s can

Algorithm 1: Probabilistic Voxel Carving (CPU)

```
Input  :  Voxel grid dimensions: $m$, $n$, $l$
          Threshold $\tau$
          Number of views $N$
          Silhouette mask set $\mathbf{I}$
          Dilation kernel size $\delta$ pixels
          Intrinsic parameters of the camera $\mathbf{K}$
Output:   Voxel grid of size $m \times n \times l$
Initialize:
Reduction step $s = 1/N$.
/* $\boldsymbol{u}$ represents the probability density for the voxel
   grid and $\boldsymbol{v}$ represents the coordinates for the
   voxel centers                                              */
Voxel Probability, $\boldsymbol{u} = \mathbf{1}_{m\times n\times l}$
$\boldsymbol{v} = meshgrid(m, n, l)$
Run:
for each view $w = 0 : N$ do
    /* Dilate the silhouette mask $\mathbf{I}_w$ by $\delta$ pixels        */
    $\mathbf{I}'_w = Dilate(\mathbf{I}_w, \delta)$
    /* Calculate the camera matrix $\mathbf{P}_w$ given the view
       angle.                                                 */
    $\mathbf{P}_w = \mathbf{K}[\mathbf{R_w}|t]$ /* The following operations below are
       running using vectorized operations                    */
    /* Project $\boldsymbol{v}$ back to $\mathbf{I}_w$:                             */
    $\boldsymbol{x} = \mathbf{P}_w\boldsymbol{v}$
    if $\boldsymbol{x}$ is not in silhouette $\mathbf{I}_w$ then
        $u = u - s$
    end
end
```

be computed as $s = 1/N$, where N is the number of views used to reconstruct the model. A threshold τ is then set to determine whether a voxel is considered an object voxel after the carving process. If the voxel value is not less than τ, it is regarded as part of the 3D object. The complete algorithm is presented in Algorithm 1. This approach of reducing the value of each voxel based on the probability of $s = 1/N$ enables us to handle inconsistencies between the views caused by factors such as mild wind or vibrations.

2.4. Parallel GPU algorithm

We leverage the graphics processing unit (GPU) to accelerate the probabilistic voxel carving (PVC) process. GPUs have many specialized processing cores that work together in parallel, providing a significant speedup in computations compared to the CPU. This acceleration particularly benefits PVC, which involves repetitive and compute-intensive projection operation for each voxel. Profiling the CPU code revealed that the iteration loop corresponding to the coordinate projection and reducing the probability of voxels (in Algorithm 1, the loop with $w = 0 : N$) is the most compute-intensive step. Hence this step was parallelized using the GPU.

The projection operation involves projecting voxel coordinates ($\boldsymbol{v}_{ijk}$) onto 2D mask images and checking if the projected coordinates fall within the plant area of the mask. If not, the corresponding voxel probability is reduced by a factor of $s = 1/N$, where N is the number of views. This compute-intensive step exploits the concept of Tensor Comprehension to execute on the GPU. Tensor Comprehension operations are amenable to massively parallelized execution where each element in the tensor is broadcast (copied) to several elements of another tensor to perform computations simultaneously (e.g., a batch of matrix pairs, like $\mathbf{P}_w$ and v_{ijk}, can be simultaneously multiplied by the use of Tensor Comprehension). While Tensor Comprehensions are implemented for a vectorized execution in the CPU, the numerous processing cores of the GPU efficiently handle these operations in parallel. Therefore, the projection of all the voxels to the silhouette mask is implemented using Tensor Comprehension, resulting in faster execution on a GPU.

Algorithm 2: Probabilistic Voxel Carving (GPU)

```
Input  :  Voxel grid dimensions: $m$, $n$, $l$
          Threshold $\tau$
          Number of views $N$
          Silhouette mask set $\mathbf{I}$
          Dilation kernel size $\delta$ pixels
          Intrinsic parameters of the camera $\mathbf{K}$
Output:   Voxel grid of size $m \times n \times l$
Initialize:
Reduction step $s = 1/N$.
/* $\boldsymbol{u}$ represents the probability density for the voxel
   grid and $\boldsymbol{v}$ represents the coordinates for the
   voxel centers                                              */
Voxel Probability, $\boldsymbol{u} = []$
Pre-compute:
for each view $w = 0 : N$ do
    /* Dilate the silhouette mask $\mathbf{I}_w$ by $\delta$ pixels        */
    $\mathbf{I}'_w = Dilate(\mathbf{I}_w, \delta)$
    /* Calculate the camera matrix $\mathbf{P}_w$ given the view
       angle.                                                 */
    $\mathbf{P}_w = \mathbf{K}[\mathbf{R_w}|t]$
end
Run:
/* First calculate the voxel batches                          */
$(m', n', l') = ceil((m, n, l)/256)$
/* For each voxel batch, perform voxel carving                */
for $i = 0 : m'$ do
    for $j = 0 : n'$ do
        for $k = 0 : l'$ do
            Initialize $\boldsymbol{u}_{ijk} = \mathbf{1}_{256\times256\times256}$, $\boldsymbol{v}_{ijk} = meshgrid(i, j, k)$
            /* Transfer the $\boldsymbol{u}_{ijk}$ from CPU to GPU memory  */
            for $w = 0 : N$ do
                /* Transfer $\mathbf{I}_w$ from CPU to GPU                */
                /* The following operations below are
                   running on GPU                                 */
                /* Project $\boldsymbol{v}_{ijk}$ back to $\mathbf{I}_w$:                    */
                $\boldsymbol{x}_{ijk} = \mathbf{P}_w\boldsymbol{v}_{ijk}$
                if $\boldsymbol{x}_{ijk}$ is not in silhouette $\mathbf{I}_w$ then
                    $u_{ijk} = u_{ijk} - s$
                end
            end
            /* Transfer $\boldsymbol{u}_{ijk}$ from GPU to CPU                */
            Append $\boldsymbol{u}_{ijk}$ to $u$
        end
    end
end
```

In our GPU-based PVC algorithm (Algorithm 2), one of the crucial aspects is the efficient data transfer between CPU and GPU memory. Due to the lower capacity of GPU memory relative to the voxel size, the algorithm implements space partitioning and breaks the voxel grid into smaller batches of size (256^3). The batch size has been chosen based on the biggest chunk that could fit into the GPU memory in one data stream. Transferring each batch ($\boldsymbol{u}_{ijk}$) from CPU to GPU memory allows the voxel probabilities to be processed on the GPU, where the compute-intensive projection operations take place. To create batches based on the resolution and voxel size, a series of points on each dimension are created; we then group these points into 256 size partitions and create a batch mesh grid out of these partitions; For each batch, calculated image masks transferred from CPU to GPU one by one and then carving process occurred. Finally, we stacked up all the carved points and returned them to the CPU again. Fig. 22 shows the processing time of carving, specifying the time allocated for each step.

2.5. Traits extraction

The reconstructed 3D model of the maize plant, along with the model of the stalk created from the pseudo-skeleton of the stalk,

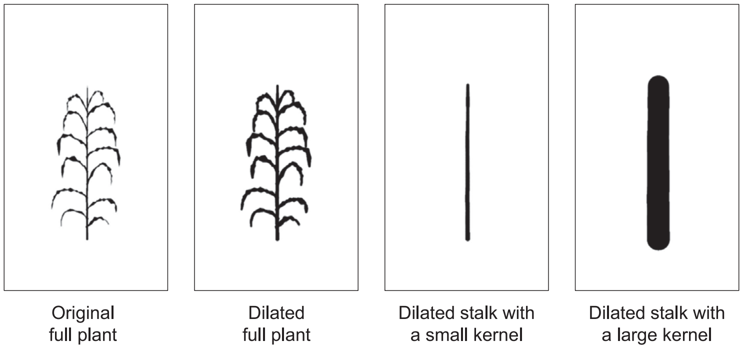


Fig. 8. Create masks for trait extraction. (1) The plants are shaking during the rotation. It is necessary to dilate the original masks to reduce disconnected components in reconstructed models. The traits we need in our application are leaf count, leaf-stalk angle, and inter-node distance. Therefore the negative impact of dilation on our reconstruction is acceptable. (2) For the pseudo-stalk masks, we apply dilation twice to obtain two sets of masks, differing in kernel size.

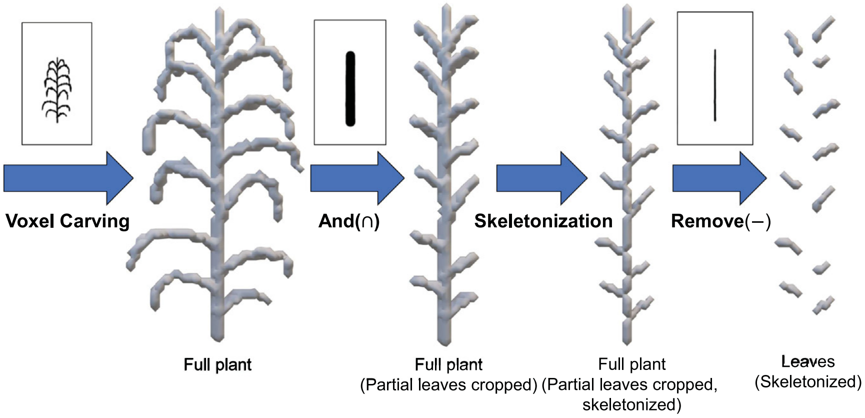


Fig. 9. After obtaining the dilated masks of the full plant, we use them to generate the 3D model of the entire plant. Next, we utilize the dilated stalk masks with a larger kernel to crop the leaves, retaining only the parts connected to the stalk. We then skeletonize the cropped model and remove the stalk from it using the dilated stalk masks with a smaller kernel, resulting in disconnected leaf components.

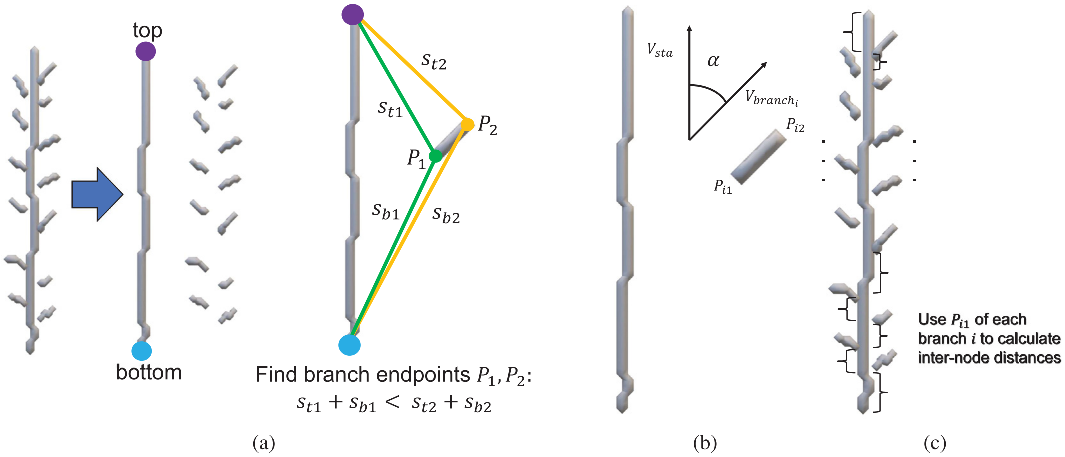

top
bottom
s_{t2}
s_{t1}
P_2
P_1
s_{b1}
s_{b2}
Find branch endpoints P_1, P_2:
$s_{t1} + s_{b1} < s_{t2} + s_{b2}$
V_{sta}
α
V_{branch_i}
P_{i2}
P_{i1}
Use P_{i1} of each branch i to calculate inter-node distances
(a)
(b)
(c)

Fig. 10. Traits extraction and calculation. (a) Find the endpoints of a branch given the stalk's top and bottom points. (b) Calculate the stalk-branch angle. (c) Calculate the inter-node distances.

allows us to automatically estimate morphometric traits, including the position, angle, and count of the leaves. A segment of the leaves close to the stalk is initially cropped out. Subsequently, the orientation of the leaves relative to the stalk is identified, and the traits are estimated. An overview of the trait extraction process is illustrated in Figs. 8–10.

To isolate the leaf segments, we employ a method that involves creating two masks by dilating the reconstructed stalk with two differently sized kernels, as depicted in Fig. 8. The size of the smaller kernels corresponds to the average width of the stalk, while the size of the larger kernel depends on the average length of the leaf segment from the center of the stalk. The dilation step is necessary to prevent gaps or disconnected components from being introduced in the reconstruction. Since the leaves are thin components, certain views might have multiple components of the leaves due to the leaf thickness being smaller than the pixel size. This might lead to multiple plant components in the 3D reconstruction, making trait extraction challenging. Next, we

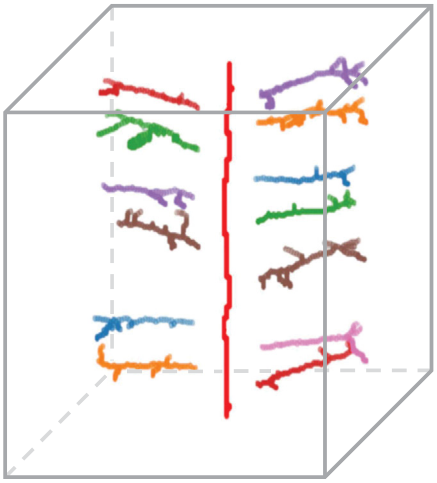


Fig. 11. The stalk and leaves (branches) can be treated as disconnected components. The number of components can then be counted to get the total leaf count.

crop out the portion of the full plant contained within the dilated stalk with the large kernel, as illustrated in Fig. 9. We then skeletonize the cropped-out plant and isolate the individual leaf segments by removing the stalk from the skeleton using the dilated stalk with the small kernel. The number of individual leaf segments (as shown in Fig. 11) is then counted to calculate the number of leaves of the plant.

The orientation of the stalk and leaf segments is identified by labeling the endpoints, as shown in Fig. 10(a). In the case of the stalk, the lowest point in the vertical direction between the endpoints is the bottom point. For the leaf segments, the point closest to the stalk (P_1) is determined by computing the sum of distances from the endpoints to the stalk endpoints and selecting the point with the lowest value. The stalk is oriented from bottom to top, while the leaf segment is oriented from the closest point (P_1) to the furthest endpoint (P_2).

We then proceed to identify the segment of the stalk near the closest endpoint of the leaf and use principal component analysis (PCA) (Jollife and Cadima, 2016) to determine the angle/direction of the leaf segment and stalk. The angle between the two directions is subsequently calculated, as depicted in Fig. 10(b). Finally, we order the leaf segments based on the vertical coordinates of the closest endpoints and assign the leaf number. The closest endpoints are treated as nodes, and the distance between subsequent nodes is computed as the inter-node distance, as shown in Fig. 10(c).

3. Results

In this section, we present the results of performing the voxel carving using a large dataset of more than 1000 plants. For our study, inbred lines from the Shoot Apical Meristem (SAM) diversity panel were cultivated at the Curtiss farm at Iowa State University during the summer of 2022. We considered 350 lines/genotypes grown in a row of 20 plants with two replicates each. Two plants from each row were harvested when they reached vegetative maturity and transported to an indoor room for video capturing.

To capture the video, a Canon EOS 5D Mark IV (DS126601) camera with a 24 mm lens was mounted on a tripod in portrait mode at 1.37 m. The video was recorded at a resolution of 1080 × 1920 and a frame rate of 30 fps. In total, 1070 plants were recorded, with each video taking approximately 3 min. The video-capturing process took approximately 53.5 h. The parameters of the intrinsic matrix for our imaging setup are listed in Table 1.

Our implementation of the GPU algorithm utilized the PyTorch library version 1.12.1 for an NVIDIA GPU. For all the results in this section, we used an NVIDIA A100 GPU with 80 GB VRAM running on a CPU with Intel Skylake Xeon processors with 512 GB RAM.

Table 1
The intrinsic parameters of the Canon EOS 5D Mark IV (DS126601) camera we used during the experiment.

Camera parameters	Values (pixel)
f_x	1760
f_y	2050
p_x	612
p_y	926

Table 2
The actual dimension of each resolution setup. We use $resX$, the size in the X-axis of the voxel grid to represent each resolution setup.

resX	Actual dimension
32	$32 \times 57 \times 32$
64	$64 \times 114 \times 64$
128	$128 \times 228 \times 128$
256	$256 \times 455 \times 256$
512	$512 \times 910 \times 512$
1024	$1024 \times 1820 \times 1024$

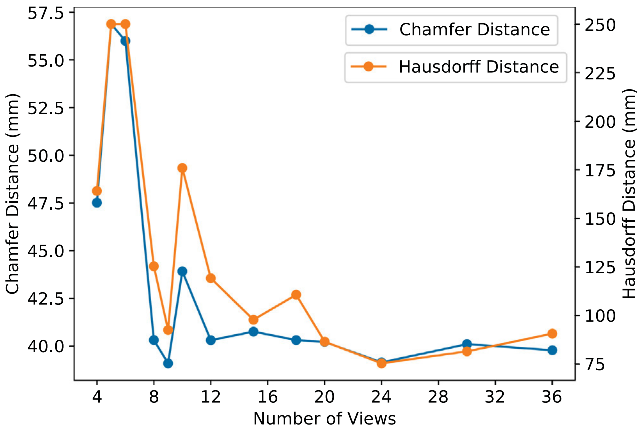


Fig. 12. Accuracy results, i.e., the Chamfer distance and the Hausdorff distance of the output model with respect to the number of views. Here, $resX = 512, \delta = 2, \tau = 0.8$.

As a first step, we need to determine a suitable set of parameters for the algorithm. These parameters comprise the number of views, the resolution of the voxel grid, $resX$ (or grid size), the dilation size, δ employed in binary segmentation mask preprocessing, and the probability threshold τ in the voxel carving algorithm. Our implementation of the voxel carving algorithm can handle different resolutions. To maintain a consistent aspect ratio of the reconstructed plants, we increase the resolution of the voxel grid along the height direction of the plant. This will result in the voxels having a consistent cubical aspect ratio. The exact resolution of the voxel grid for each resolution is shown in Table 2. For ease of discussion, we use the X-axis resolution, denoted as $resX$, to represent each resolution.

We performed a series of parameter-tuning experiments on a virtual plant model to fine-tune and identify the parameters above. To evaluate the effectiveness of our algorithm, we analyzed the Chamfer distance and Hausdorff distance of our output model with respect to the number of views and resolution of the voxel grid, using different combinations of dilation, δ, and threshold, τ.

Chamfer and Hausdorff distances gauge the similarity between two point sets. Chamfer distance (Fan et al., 2017) determines the mean distance between each point in one point set and its nearest neighbor in the other set. It is calculated by identifying the distance between each point in one set and its closest neighbor in the other and averaging the distances. Hausdorff distance (Huttenlocher et al., 1993), conversely, evaluates the maximum distance between each point in one point

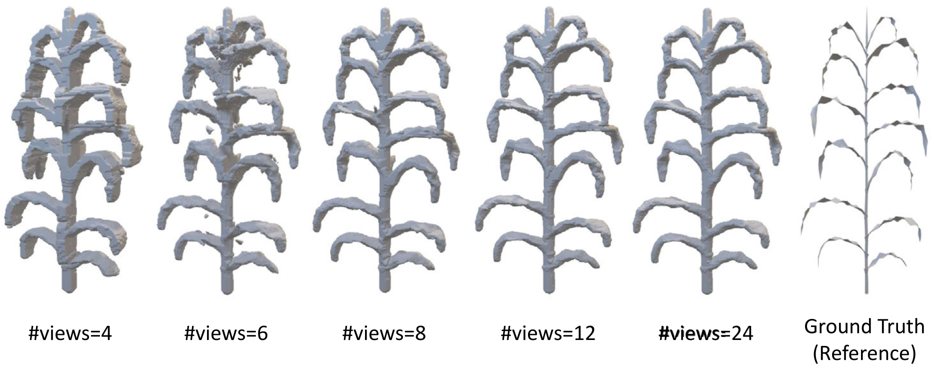


Fig. 13. The reconstructed models of the synthetic plant (Ground Truth/Reference) for the corresponding numbers of views (#$views$) in Fig. 12. Here, $resX = 512, \delta = 2, \tau = 0.8$.

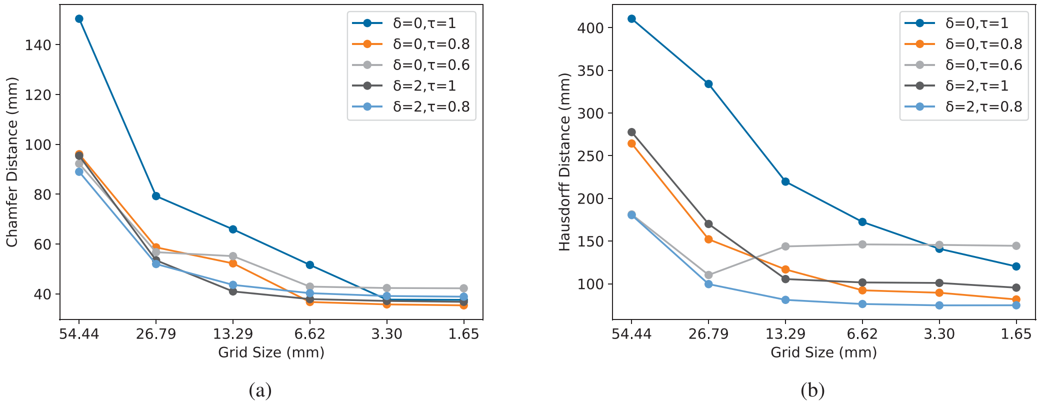


Fig. 14. (a) The Chamfer distances and (b) the Hausdorff distances of the output models for different grid sizes of the voxels. Here the size of the bounding box remains the same, so the resolution controls the grid size. δ is the dilation used in the binary mask preprocessing; τ is the threshold set in the algorithm (See Algorithm 1). The number of views is fixed as 24.

set and its nearest neighbor in the other set. It is determined by ascertaining the distance between a point in one set and its closest neighbor in the other set and then selecting the maximum of these distances. Chamfer and Hausdorff distances are routinely utilized to assess the performance of point set registration algorithms and compare the resemblance between point sets in object recognition and 3D reconstruction tasks.

Fig. 12 demonstrates that augmenting the number of views enhances the precision of the carved model. A greater number of views provides more information, which facilitates the creation of a 3D model of the maize plant with finer details. The graph exhibits some leaps as we increase the number of views. This is possible because when only a limited number of view images are chosen, the angles at which they are viewed may greatly impact the resulting model. Nevertheless, we noticed that when a smaller set of views is included within a larger set, the resulting model generated from the larger set of view images tends to be more precise than that from the smaller one. Some sample reconstructed models for the corresponding numbers of views are shown in Fig. 13.

In Fig. 14(a) and (b), it is evident that carving with a smaller grid size, i.e., a higher resolution of the voxel grid, generally yields better results than carving with a lower resolution. The rationale is that carving with a lower resolution tends to generate a coarser model. The $\delta = 2$ and $\tau = 0.8$ produce reasonably good results among the tested combinations.

After determining the algorithm parameters, we validated the process by reconstructing eight randomly selected maize plants with manually collected ground truth data on leaf numbers and leaf-stalk angles. All eight plants we selected are shown in Fig. 15. We use $\tau = 0.8$ as a threshold and set every value below this to be zero to obtain the final Binary voxelized model of the object for further trait extraction. Table 3 compares the result of the number of leaves extracted by our algorithm with the ground truth. The RMSE of the leaf count for these eight sample plants is 1.84.

We use the marching cubes algorithm (Lorensen and Cline, 1987) to create triangular mesh models of the carved voxel grid for rendering. Fig. 16 compares the output models obtained using the traditional voxel carving algorithm and our proposed probabilistic voxel carving algorithm. In the traditional voxel carving algorithm, the object is represented as a set of 3D voxels, each of which is either inside or outside the object. However, this algorithm can be prone to errors due to noise in segmentation masks and slight deformations in the object, which can lead to disconnected components or holes in the resulting models. As shown in Fig. 16(a), several parts of the plant are carved away due to segmentation errors or vibrations of the plant during video capture. Our probabilistic voxel carving algorithm considers the uncertainty in the voxel labels. As seen in Fig. 16(b), our proposed algorithm exhibits a significant improvement in resolving these reconstruction errors, resulting in a more accurate representation of the plant.

Fig. 17 shows the output models with different combinations of thresholds τ and the number of used view images in a matrix of 4×4. It demonstrates the impact of the parameters on the algorithm's behavior and output's visual quality. When the number of views is fixed, the model tends to be finer as we increase the threshold. However, when the threshold gets close to 1, the same noise issues in traditional voxel carving appear, leading to disconnected components. When the threshold equals 1, our algorithm turns into the traditional voxel carving algorithm. Therefore, the selection of the threshold needs to balance this trade-off.

We used volume rendering (Drebin et al., 1988; Kaufman and Mueller, 2005; Shah et al., 2021; Young and Krishnamurthy, 2018)

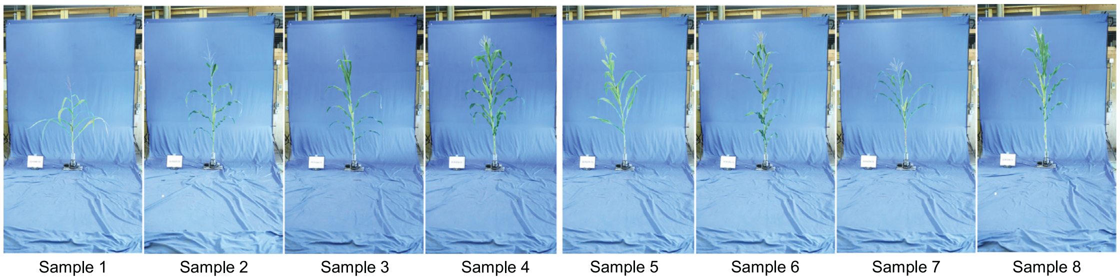


Fig. 15. The different samples of the maize plants used for validating the extracted traits using our framework.

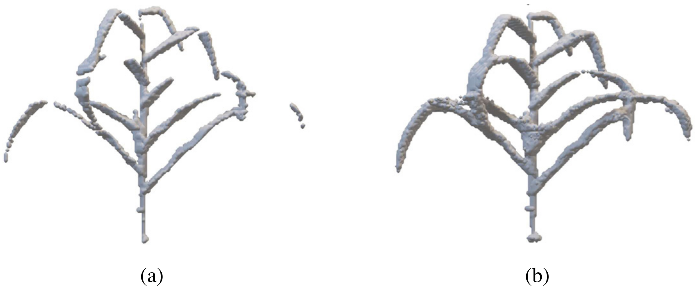


Fig. 16. A comparison between results of traditional voxel carving and probabilistic voxel carving algorithms. The resolution is set as $resX = 256$ and the dilation $\delta = 2$. The plant we show here is Sample 1. (a) The result of the traditional voxel carving algorithm. A lot of disconnections can be spotted in the output model. (b) The result of the probabilistic voxel carving algorithm with the threshold $\tau = 0.8$.

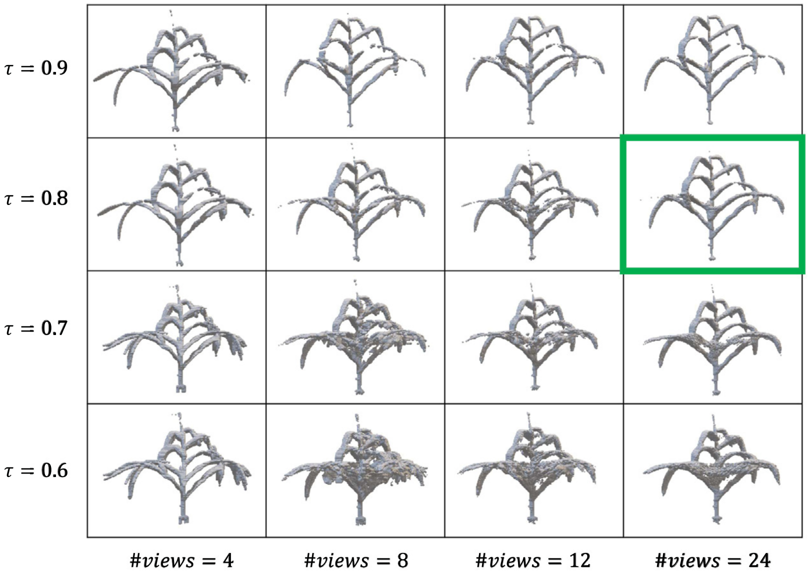


Fig. 17. A 4 × 4 matrix of output models with different combinations of thresholds (τ) and the number of used view images ($\#views$). The resolution is set as $resX = 256$ and the dilation $\delta = 2$. The plant we show here is Sample 1. The green box marks the combination setting for the best output model we observed.

of the predicted probabilities for 3 sample plant geometries (Fig. 18) to further demonstrate the effectiveness of our approach. The visualization shows that areas with multiple leaves nearby have a lower probability than other areas of the plants. It also demonstrates regions that would be carved off using the traditional voxel carving approach, for example, the tips of the leaves or leaves farther away from the stalk, which are prone to vibrations during data collection.

Another trait that we evaluated is the leaf-stalk angle. Table 4 shows the RMSE results of the leaf-stalk angle measurements. We noticed the presence of a constant bias in angle measurement, as the extracted angles were typically larger than the ground truth values. This can be attributed to some consistent bias in the ground truth measurements since the leaf angles are very sensitive to the measurement plane. However, the relative trend of the leaf angles with the plant height is maintained. To remove this bias, we use mean-centered angles. The mean-centered leaf angles as a function of normalized height are shown in Fig. 19, which shows good agreement with the measurements.

We analyzed the impact of voxel grid resolution on the accuracy of leaf-stalk angle estimation. Fig. 20 shows the mean-centered angle error of eight sample plants at different voxel grid resolutions. Our algorithms were executed across different resolution settings (Table 2), including $resX = 128$, $resX = 256$, and $resX = 512$. Our results suggest that a resolution setting of $resX = 512$ provides better angle estimation than $resX = 128$ and $resX = 256$. This is because output models with higher voxel resolution contain finer details. The skeletonization algorithm (Lee et al., 1994) we used tends to remove branches with

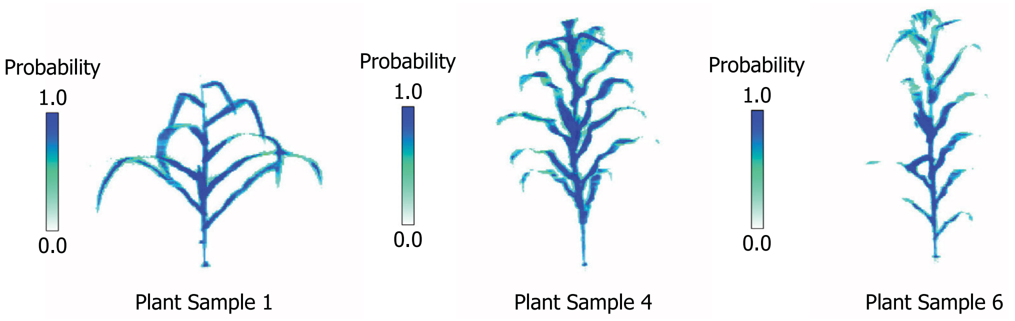
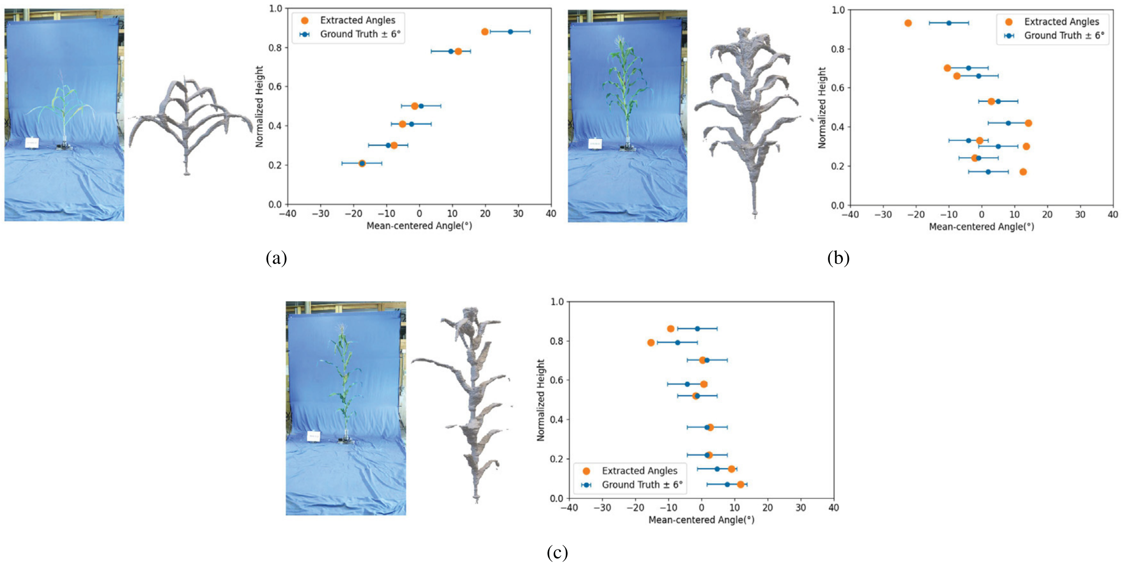


Fig. 18. Volume rendering of predicted probability of voxels for plant Sample 1, plant Sample 4, and plant Sample 6. Here, $resX = 512, \delta = 2, \tau = 0.8$.



(a)

(b)

(c)

Fig. 19. Mean-centered leaf angle distribution as a function of normalized height for (a) plant Sample 1, (b) plant Sample 4, and (c) plant Sample 6, along with their RGB images and reconstructed 3D models. Here, $resX = 512, \delta = 2, \tau = 0.8$.

Table 3
Leaf counting results of 8 randomly-selected maize plants. The RMSE of the number of leaves is 1.84. ($resX = 512, \delta = 2, \tau = 0.8$.)

Plant ID	Genoytpe	Ground truth	Result
Sample 1	Ki43	8	7
Sample 2	78551S	8	8
Sample 3	A679	9	8
Sample 4	Tzi18	13	12
Sample 5	CML333	13	9
Sample 6	T8	12	12
Sample 7	CI90C	9	11
Sample 8	NC33	11	9

Table 4
The RMSE results of the leaf-stalk angles for each tested maize plant. ($resX = 512, \delta = 2, \tau = 0.8$.)

Plant ID	RMSE of leaf-stalk angles (°)
Sample 1	14.50
Sample 2	15.55
Sample 3	15.40
Sample 4	14.96
Sample 5	8.04
Sample 6	14.42
Sample 7	13.61
Sample 8	9.61

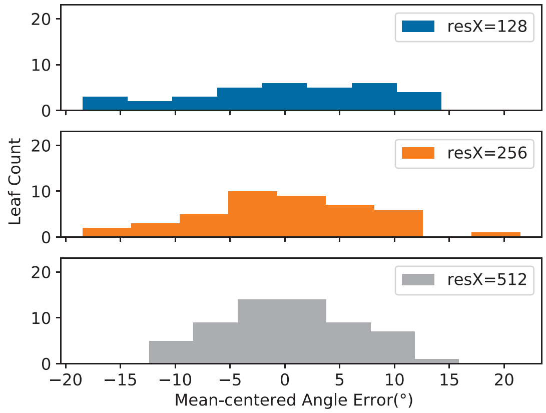


Fig. 20. The mean-centered angle error histograms for eight sample plants with varying voxel grid resolution.

very few voxels, which could remove leaves in the lower-resolution models by mistake. Considering the accuracy-cost trade-off, the figure also indicates that a resolution setup of $resX = 512$ is sufficient for angle extraction.

To evaluate the performance of our voxel carving and trait extraction algorithm, we performed automated runs on more than 1000 plants and assessed the total computational time. Performing the voxel

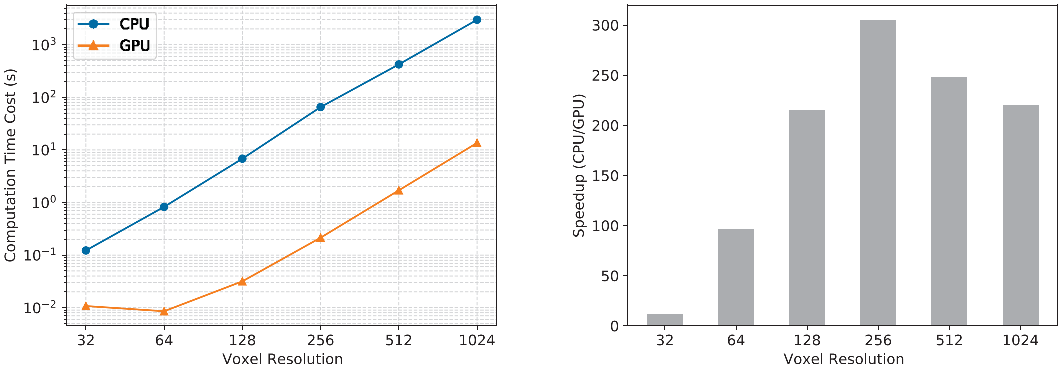


Fig. 21. Time performance comparison between CPU and GPU computing (left) and the speedup using the GPU (right) with respect to voxel resolution $resX$.

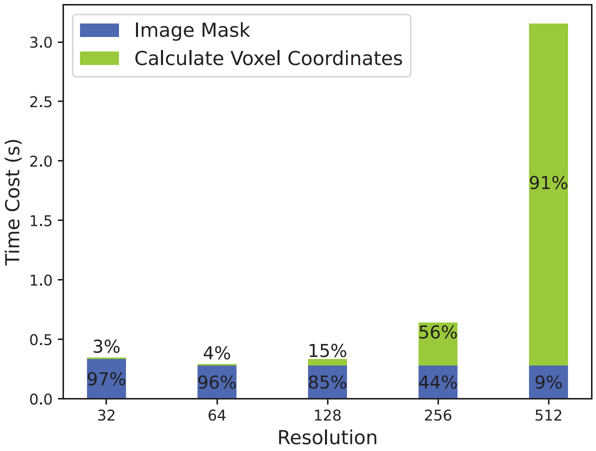


Fig. 22. Time for performing each step of voxel carving on the GPU. Calculating the voxel coordinates at higher resolutions dominates the total time, emphasizing the need for GPU parallel computing. The calculation time for image masks remains the same for different resolutions. Transferring voxel coordinates data from GPU to CPU took (<0.005) for all resolutions.

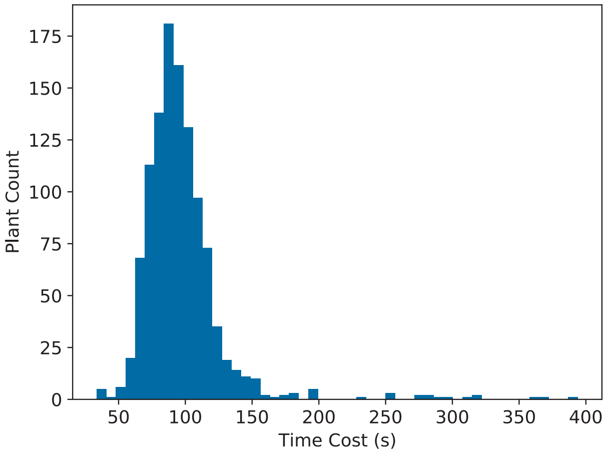


Fig. 23. The histogram of total time for voxel carving and trait extraction of all 1070 maize plants.

carving computation on the GPU resulted in a marked improvement in computation time. To measure the improvement, we measured the average computational time for the 3D reconstruction of ten plants over six voxel resolutions, as shown in Fig. 21. Our observations indicate that the logarithm of CPU computation time increases linearly, which is not the case for the GPU at lower resolutions. This is because transferring the data from the CPU to GPU at low resolutions dominates the computational time. The greatest speed improvement is associated with the resolution setup of $resX = 256$. However, for higher resolutions, where the voxel size exceeds the GPU size, the data must be sliced into smaller chunks and processed separately. This process requires additional processing time, resulting in relatively lower speed-up for $resX = 512$ and $resX = 1024$ resolution configurations.

The distribution of the total computational time for 1070 plants at $resX = 512$ is shown in Fig. 23. As the resolution of the voxel grid is the same for all plants, the time for voxel carving remains constant. The primary factor affecting the computational time for each plant is the number of leaves since the trait extraction step is directly dependent on this factor. However, te median time cost is less than 100 s, making the process computationally tractable for a large data set.

4. Discussion and conclusions

In summary, our probabilistic voxel carving algorithm accurately reconstructs the voxelized 3D model of the plants, which can then be further used to extract morphometric traits. As discussed in Section 3, we undertook a series of experiments to assess and validate our proposed approach. Firstly, we conducted parameter-tuning experiments on a synthetic plant to identify suitable parameters for our algorithm. Our results indicate that increasing the number of views enhances the accuracy of voxel carving. We then validated our approach by applying it to ten randomly selected plants. Our results demonstrate that our approach effectively captures the trends of leaf growth in the plants, except for outlier plants where leaves are close to the stalk. Finally, we evaluated the computational performance of our approach by running it on over one thousand plants. Our results reveal that, with the aid of GPU acceleration, the entire pipeline is computationally feasible on a large-scale dataset.

One main limitation of our 3D reconstruction approach is that the leaves of the 3D models are thicker than actual plants. This is due to two reasons: the threshold value used in our probabilistic voxel carving algorithm; and the dilation applied to the 2D masks extracted from the plant images to prevent disconnected components in the 3D reconstruction. Hence, the threshold value needs to be chosen very carefully; if the value is too high, it will lead to some leaves getting split into multiple components, while a lower value might lead to multiple leaves fusing together. Both will lead to incorrect traits being extracted from the 3D reconstruction. Since the main goal of this work is to automatically extract the morphometric traits of plants, we chose a slightly conservative threshold value, which allows us to accurately extract the traits, specifically for the leaf angle and number of leaves. However, this threshold has to be carefully selected for accurate 3D reconstruction, based on the final application.

Some of the parameters used in the current algorithm require further experimentation to better understand the reconstruction process.

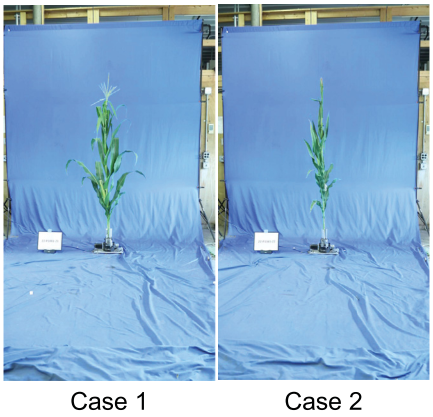


Fig. 24. Two failure sample cases for trait extraction.

For instance, the threshold used for the voxel probability could be a dynamic value rather than a fixed one. Additionally, identifying the endpoints to calculate the leaf-stalk angle in trait extraction may not work for overlapping leaves as well as small leaves that could be discarded.

Further investigations are required to reconstruct plants with leaves close to the stalk, as predicting their pseudo-stalks with the current deep-learning model proves challenging. This could lead to incorrect values for the extracted traits. Two failure sample cases are shown in Fig. 24. One way we can mitigate this is to have a filter on the extracted traits. If the number of leaves extracted using our approach is significantly small or large (<5 or >20), we could flag these plants for further manual review.

Finally, it should be noted that we are using a deep learning-based approach for segmenting the stalk, while we use the voxel carving approach for segmenting the leaves. We need to identify each leaf and its closest stalk segment to calculate the leaf-stalk angle. It was challenging to obtain a clean stalk by voxel carving; therefore, we used a deep learning method to segment the stalk. It is worth exploring if we can simplify the pipeline to use the voxel-carving approach for the complete trait extraction pipeline.

Overall, our results suggest that the algorithm is robust to varying numbers of views and can efficiently handle large datasets due to GPU acceleration. Leveraging this algorithm provides a powerful tool for plant phenotyping, enabling the rapid generation of plant and pseudo-stalk models and the extraction of various traits such as leaf number, stalk-leaf angles, inter-node distances, and more. Our work contributes to developing automated plant phenotyping tools to aid plant breeding and crop improvement efforts.

CRediT authorship contribution statement

Jiale Feng: Software, Writing – original draft, Formal analysis, Investigation, Data curation. **Mojdeh Saadati:** Software, Writing – original draft, Visualization, Investigation. **Talukder Jubery:** Software, Writing – original draft, Visualization, Investigation. **Anushrut Jignasu:** Software, Writing – original draft, Visualization, Investigation. **Aditya Balu:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Yawei Li:** Formal analysis, Investigation, Data curation. **Lakshmi Attigala:** Formal analysis, Investigation, Data curation. **Patrick S. Schnable:** Conceptualization, Supervision, Writing – review & editing. **Soumik Sarkar:** Conceptualization, Supervision, Writing – review & editing. **Baskar Ganapathysubramanian:** Conceptualization, Supervision, Writing – review & editing. **Adarsh Krishnamurthy:** Conceptualization, Methodology, Supervision, Writing – review & editing.

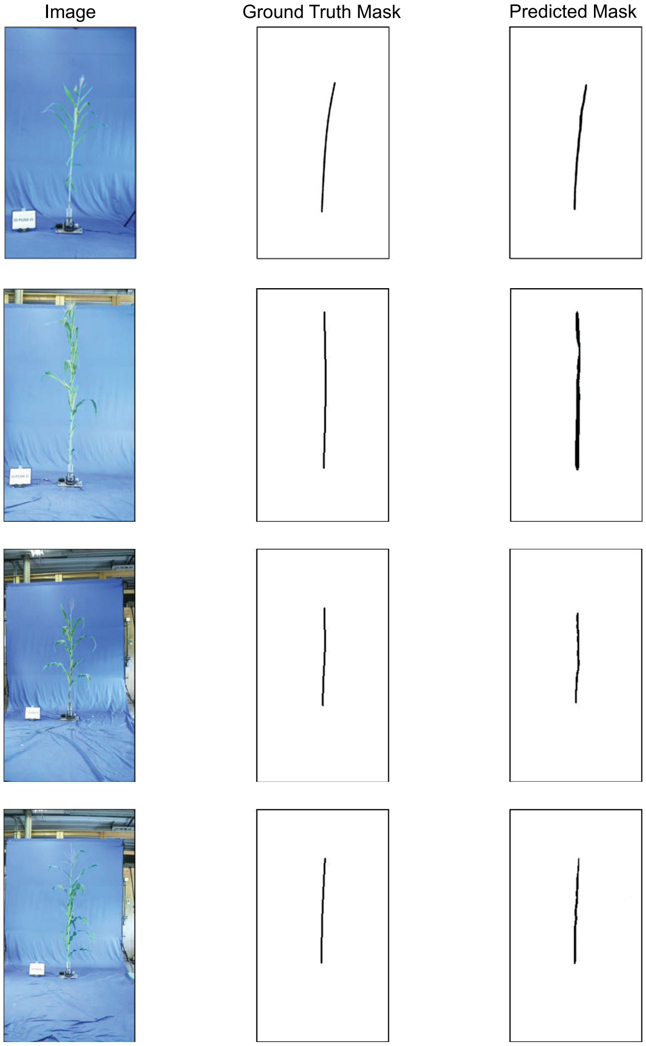


Fig. A.1. Examples of pseudo-stalk detection.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the Plant Science Institute at Iowa State University, USA, the National Science Foundation, USA under grant number OAC:1750865, and the National Institute of Food and Agriculture (USDA-NIFA), USA as part of the AI Institute for Resilient Agriculture (AIIRA), grant number 2021-67021-35329. We would like to thank James Afful, Timilehin Ayanlade, Nirmal Baishnab, Shivani Chiranjeevi, Dhruv Gamdha, Mozhgan Hadadi, Shambhavi Joshi, Mahsa Khosravi, Ankush Mishra, Ajith Reddy Moola, Suriya Moorthy, Anirudha Powadi, Nasla Saleem, Mehdi Shadkhah, and Ronak Tali for helping in the data collection process.

Appendix. Stalk detection

In order to predict the stalks of plants, a deep neural network model is employed, utilizing the FPN architecture (Lin et al., 2017), ResNet34 encoder (He et al., 2015), and sigmoid activation. The dataset consists of images from 100 randomly-selected plants, with each plant having one image that has been manually labeled with its corresponding pseudo-stalk. This dataset is divided into three parts: training, validation, and test sets, with 68 images being allocated for training, 16 images for validation, and 16 images for testing.

During training, the weights from pre-training on the ImageNet dataset (Russakovsky et al., 2015) are employed, along with the Adam optimizer, for 100 epochs. The Dice loss function (Sudre et al., 2017) is used to optimize the model, which is commonly used for image segmentation tasks. This function measures the degree of overlap between the predicted and ground truth masks, indicating their similarity. In Fig. A.1, we show some results of our pseudo-stalk detection.

References

Cabrera-Bosquet, L., Fournier, C., Brichet, N., Welcker, C., t Suard, B., Tardieu, F., 2016. High-throughput estimation of incident light, light interception and radiation-use efficiency of thousands of plants in a phenotyping platform. New Phytol. 212, 269–281. http://dx.doi.org/10.1111/nph.14027.

Drebin, R.A., Carpenter, L., Hanrahan, P., 1988. Volume rendering. ACM SIGGRAPH Comput. Graph. 22, 65–74. http://dx.doi.org/10.1145/378456.378484.

Fan, H., Su, H., Guibas, L., 2017. A point set generation network for 3D object reconstruction from a single image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017-January, IEEE, pp. 2463–2471. http://dx.doi.org/10.1109/CVPR.2017.264.

Gaillard, M., Miao, C., Schnable, J.C., Benes, B., 2020. Voxel carving-based 3D reconstruction of sorghum identifies genetic determinants of light interception efficiency. Plant Direct 4, http://dx.doi.org/10.1002/pld3.255.

Garrido, M., Paraforos, D., Reiser, D., Arellano, M.V., Griepentrog, H., Valero, C., 2015. 3D Maize plant reconstruction based on georeferenced overlapping LiDAR point clouds. Remote Sens. 7, 17077–17096. http://dx.doi.org/10.3390/rs71215870.

Ge, Y., Bai, G., Stoerger, V., Schnable, J.C., 2016. Temporal dynamics of maize plant growth, water use, and leaf water content using automated high throughput RGB and hyperspectral imaging. Comput. Electron. Agric. 127, 625–632. http://dx.doi.org/10.1016/j.compag.2016.07.028.

Golbach, F., Kootstra, G., Damjanovic, S., Otten, G., van de Zedde, R., 2016. Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. Mach. Vis. Appl. 27, 663–680. http://dx.doi.org/10.1007/s00138-015-0727-5.

Grizzly Industrial Inc., 2022. Product page for 4-3/8" power rotary table with DRO control at grizzly.com. https://www.grizzly.com/products/grizzly-4-3-8-power-rotary-table-with-dro-control/t1191.

Hammer, G.L., Dong, Z., McLean, G., Doherty, A., Messina, C., Schussler, J., Zinselmeier, C., Paszkiewicz, S., Cooper, M., 2009. Can changes in canopy and/or root system architecture explain historical Maize yield trends in the US Corn belt? Crop Sci. 49, 299–312. http://dx.doi.org/10.2135/cropsci2008.03.0152.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. 2016-December, 770–778. http://dx.doi.org/10.48550/arxiv.1512.03385.

Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1993. Comparing images using the Hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 15, 850–863. http://dx.doi.org/10.1109/34.232073.

Jollife, I.T., Cadima, J., 2016. Principal component analysis: A review and recent developments. Phil. Trans. R. Soc. A 374, http://dx.doi.org/10.1098/RSTA.2015.0202.

Junker, A., Muraya, M.M., Weigelt-Fischer, K., Arana-Ceballos, F., Klukas, C., Melchinger, A.E., Meyer, R.C., Riewe, D., Altmann, T., 2015. Optimizing experimental procedures for quantitative evaluation of crop plant performance in high throughput phenotyping systems. Front. Plant Sci. 5, 1–21. http://dx.doi.org/10.3389/fpls.2014.00770.

Kaufman, A., Mueller, K., 2005. Overview of volume rendering. Vis. Handb. 127–174. http://dx.doi.org/10.1016/B978-012387582-2/50009-5.

Kimball, B.A., Thorp, K.R., Boote, K.J., Stockle, C., Suyker, A.E., Evett, S.R., Brauer, D.K., Coyle, G.G., Copeland, K.S., Marek, G.W., Colaizzi, P.D., Acutis, M., Alimagham, S., Archontoulis, S., Babacar, F., Barcza, Z., Basso, B., Bertuzzi, P., Constantin, J., De Antoni Migliorati, M., Dumont, B., Durand, J.-L., Fodor, N., Gaiser, T., Garofalo, P., Gayler, S., Giglio, L., Grant, R., Guan, K., Hoogenboom, G., Jiang, Q., Kim, S.-H., Kisekka, I., Lizaso, J., Masia, S., Meng, H., Mereu, V., Mukhtar, A., Perego, A., Peng, B., Priesack, E., Qi, Z., Shelia, V., Snyder, R., Soltani, A., Spano, D., Srivastava, A., Thomson, A., Timlin, D., Trabucco, A., Webber, H., Weber, T., Willaume, M., Williams, K., van der Laan, M., Ventrella, D., Viswanathan, M., Xu, X., Zhou, W., 2023. Simulation of evapotranspiration and yield of Maize: An inter-comparison among 41 Maize models. Agricult. Forest Meteorol. 333, 109396.

Klodt, M., Cremers, D., 2015. High-resolution plant shape measurements from multi-view stereo reconstruction. Lecture Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8928, 174–184. http://dx.doi.org/10.1007/978-3-319-16220-1_13.

Kutulakos, K.N., Seitz, S.M., 2000. A theory of shape by space carving. Int. J. Comput. Vis. 38, 199–218. http://dx.doi.org/10.1023/A:1008191222954.

Lee, T.C., Kashyap, R.L., Chu, C.N., 1994. Building skeleton models via 3D medial surface axis thinning algorithms. CVGIP: Graph. Models Image Process. 56, 462–478. http://dx.doi.org/10.1006/cgip.1994.1042.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Lorensen, W.E., Cline, H.E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. ACM SIGGRAPH Comput. Graph. 21, 163–169. http://dx.doi.org/10.1145/37402.37422.

Lou, L., Liu, Y., Sheng, M., Han, J., Doonan, J.H., 2014. A cost-effective automatic 3D reconstruction pipeline for plants using multi-view images. Lecture Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8717 LNAI, 221–230. http://dx.doi.org/10.1007/978-3-319-10401-0_20.

McCormick, R.F., Truong, S.K., Mullet, J.E., 2016. 3D Sorghum reconstructions from depth images identify qtl regulating shoot architecture. Plant Physiol. 172, 823–834. http://dx.doi.org/10.1104/pp.16.00948.

Mock, J.J., Pearce, R.B., 1975. An ideotype of Maize. Euphytica 24 (3), 613–623.

Paulus, S., Schumann, H., Kuhlmann, H., Léon, J., 2014. High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants. Biosyst. Eng. 121, 1–11. http://dx.doi.org/10.1016/j.biosystemseng.2014.01.010.

Pendleton, J.W., Smith, G.E., Winter, S.R., Johnston, T.J., 1968. Field investigations of the relationships of leaf angle in corn (*Zea mays* l.) to grain yield and apparent photosynthesis. Agron. J. 60, 422–424. http://dx.doi.org/10.2134/agronj1968.00021962006000040027x.

Pepper, G.E., Pearce, R.B., Mock, J.J., 1977. Leaf orientation and yield of Maize. Crop Sci. 17, 883–886. http://dx.doi.org/10.2135/cropsci1977.0011183X001700060017x.

Prakash, S., Robles-Kelly, A., 2008. A semisupervised approach to space carving. In: 19th International Conference on Pattern Recognition. IEEE, pp. 1–4. http://dx.doi.org/10.1109/ICPR.2008.4761306.

Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B., 2006. Image-based plant modeling. ACM Trans. Graph. 25, 599–604. http://dx.doi.org/10.1145/1141911.1141929.

Ray, D.K., Mueller, N.D., West, P.C., Foley, J.A., 2013. Yield trends are insufficient to double global crop production by 2050. In: Hart, J.P. (Ed.), PLoS One 8, e66428. http://dx.doi.org/10.1371/journal.pone.0066428.

Roussel, J., Geiger, F., Fischbach, A., Jahnke, S., Scharr, H., 2016. 3D surface reconstruction of plant seeds by volume carving: Performance and accuracies. Front. Plant Sci. 7, 745. http://dx.doi.org/10.3389/fpls.2016.00745.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Scharr, H., Briese, C., Embgenbroich, P., Fischbach, A., Fiorani, F., Müller-Linow, M., 2017. Fast high resolution volume carving for 3D plant shoot reconstruction. Front. Plant Sci. 8, 1680. http://dx.doi.org/10.3389/fpls.2017.01680.

Schultze, B., Witt, M., Schubert, K.E., Hurley, R.F., Bashkirov, V., Schulte, R.W., Gomez, E., 2012. Space carving and filtered back-projection as preconditioners for proton computed tomography reconstruction. In: IEEE Nuclear Science Symposium. pp. 4335–4340. http://dx.doi.org/10.1109/NSSMIC.2012.6551987.

Shah, H., Huang, X., Bingol, O., Rajanna, M., Krishnamurthy, A., 2021. GPU-accelerated post-processing and animated volume rendering of isogeometric analysis results. Comput.-Aided Des. Appl. 19, 779–796. http://dx.doi.org/10.14733/cadaps.2022.779-796.

Soualiou, S., Wang, Z., Sun, W., de Reffye, P., Collins, B., Louarn, G., Song, Y., 2021. Functional–structural plant models mission in advancing crop science: Opportunities and prospects. Front. Plant Sci. 12, 2955. http://dx.doi.org/10.3389/FPLS.2021.747142/BIBTEX.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J., 2017. Generalised DICE overlap as a deep learning loss function for highly unbalanced segmentations. Lecture Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10553 LNCS, 240–248. http://dx.doi.org/10.1007/978-3-319-67558-9-28.

Tabb, A., 2013. Shape from silhouette probability maps: Reconstruction of thin objects in the presence of silhouette extraction and calibration error. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 161–168. http://dx.doi.org/10.1109/CVPR.2013.28.

Thapa, S., Zhu, F., Walia, H., Yu, H., Ge, Y., 2018. A novel LiDAR-based instrument for high-throughput, 3D measurement of morphological traits in Maize and sorghum. Sensors 18, 1187. http://dx.doi.org/10.3390/s18041187.

Tilman, D., Balzer, C., Hill, J., Befort, B.L., 2011. Global food demand and the sustainable intensification of agriculture. Proc. Natl. Acad. Sci. 108, 20260–20264. http://dx.doi.org/10.1073/pnas.1116437108.

Tross, M.C., Gaillard, M., Zwiener, M., Miao, C., Grove, R.J., Li, B., Benes, B., Schnable, J.C., 2021. 3D reconstruction identifies loci linked to variation in angle of individual sorghum leaves. PeerJ 9, e12628. http://dx.doi.org/10.7717/peerj.12628.

Westgate, M.E., Forcella, F., Reicosky, D.C., Somsen, J., 1997. Rapid canopy closure for maize production in the northern US corn belt: Radiation-use efficiency and grain yield. Field Crops Res. 49, 249–258. http://dx.doi.org/10.1016/S0378-4290(96)01055-6.

Yamazaki, S., Narasimhan, S., Baker, S., Kanade, T., 2007. Coplanar shadowgrams for acquiring visual hulls of intricate objects. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1–8. http://dx.doi.org/10.1109/ICCV.2007.4408902.

Young, G., Krishnamurthy, A., 2018. GPU-accelerated generation and rendering of multi-level voxel representations of solid models. Comput. Graph. 75, 11–24.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1330–1334. http://dx.doi.org/10.1109/34.888718.