PeanutNeRF: 3D Radiance Field for Peanuts

Farah Saeed[1] Jin Sun[1] Peggy Ozias-Akins[4] Ye (Juliet) Chu[4] Changying (Charlie) Li[2,3]

[1]School of Computing, University of Georgia
[2]Department of Agricultural and Biological Engineering, University of Florida
[3]Biosensing, Automation, and Intelligence Laboratory, University of Florida
[4]Institute of Plant Breeding, Genetics and Genomics, Department of Horticulture, University of Georgia
cli2@ufl.edu, {farah.saeed, jinsun, pozias, ychu}@uga.edu

Abstract

Accurate phenotypic analysis can help plant breeders efficiently identify and analyze suitable plant traits to enhance crop yield. While 2D images from RGB cameras are easily accessible, their trait estimation performance is limited due to occlusion and the absence of depth information. On the other hand, 3D data from LiDAR sensors are noisy and limited in their ability to capture very thin plant parts such as peanut plant pegs. To combine the merits of both the 2D and 3D data analysis, the 2D images were used to capture thin parts in peanut plants, and deep learning-based 3D reconstruction using captured 2D images was performed to obtain 3D point clouds with information about the scene from different angles. The neural radiance fields were optimized for implicit 3D representation of the plants. The trained radiance fields were queried for 3D reconstruction to achieve point clouds for a 360-degree view and frontal view of the plant. With frontal-view reconstruction and the corresponding 2D images, we used Frustum PVCNN to perform 3D detection of peanut pods. We showed the effectiveness of PeanutNeRF on peanut plants with and without foliage: it showed negligible noise and a chamfer distance of less than 4×10^{-4} *from a manually cleaned version. The pod detection showed a precision of around 0.7 at the IoU threshold of 0.5 on the validation set. This method can assist in accurate plant phenotypic studies of peanuts and other important crops.*

1. Introduction

Peanut is an important oilseed and food crop grown in over 100 countries. The total world production was around 50 million tons in 2022 [1]. Production of high-yielding peanut varieties involves conventional and molecular plant breeding programs to study and identify desirable traits in peanut plants [12, 13]. The pod yield per plant, number of pods and pegs per plant, and shelling outturn are important yield contributing traits and assist to study different genotypes. Plant phenotyping is key to the study of the physiological and morphological traits of peanut plants contributing to high yield.

While manual phenotyping is time-consuming and labor-intensive, recent advancements in computer vision-based high throughput phenotyping have enabled efficient analysis of plants' physiology using both 2D and 3D sensing. Leveraging 2D images captured using RGB cameras, several studies have performed segmentation of plant parts [3, 5, 26, 34] and skeletonization to characterize plants' architecture [4, 8, 38]. However, 2D RGB images lack depth information and are prone to occlusion. Compared to 2D images, 3D data such as point cloud data allows us to capture depth information directly with less occlusion and more accurate position information. Point cloud data from Kinect and LiDAR sensors of several plants are used to perform skeletonization and segmentation of plants utilizing traditional techniques like region growth, color-based region segmentation [2, 16, 30–32]. In other studies advanced 3D deep learning-based techniques operating on point cloud data were employed.

While 3D data enables a geometrical understanding of the scene and provides more accurate position information, the collection of high-resolution data requires expensive devices such as LiDAR. Less expensive equipment like Intel Realsense have limited resolution. In addition to the cost of the devices, scanning using 3D sensors such as FARO LiDAR is time-consuming, requires human supervision and can take hours for high-resolution scans. Furthermore, high-resolution 3D data requires a significant amount of storage space. In terms of quality, the captured point cloud data is relatively sparse and contains missing regions due to the hardware limitation of 3D sensors. As a result, very fine details like thin peanut plant pegs with less than

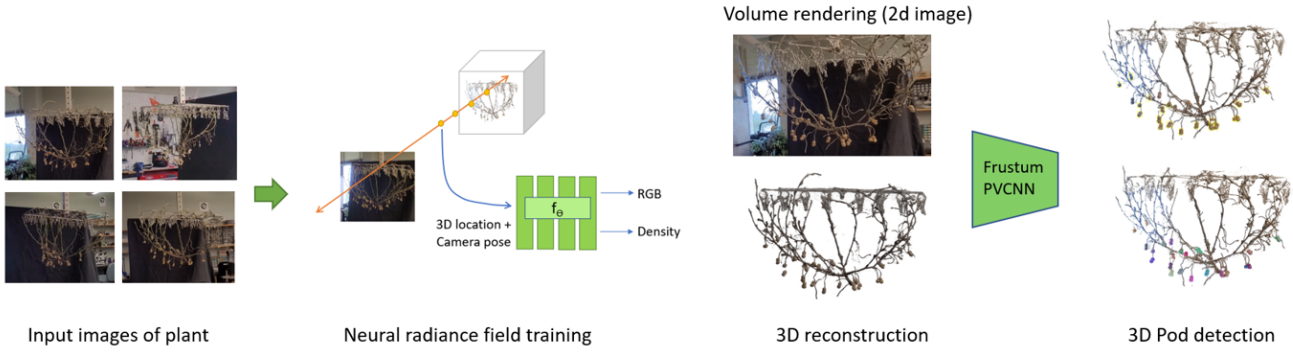


Figure 1. Overview of PeanutNeRF approach: Overlapping images from video clips of a plant are used as input to train a neural radiance field. The radiance field is queried to perform the 3D reconstruction and volume rendering of 2D views. The 3D reconstructed point clouds and 2D views are used as input to Frustum PVCNN for 3D pod detection.

1mm of radius are rarely captured. Moreover, there is high noise in captured data, which requires an additional denoising operation.

We propose PeanutNeRF, a framework that utilizes 3D radiance fields to conduct phenotypic analysis with a combination of 2D and 3D data. Inspired by the recent advances in 3D scene modeling using implicit 3D models (NeRF) [20], we first capture 2D data of a scene that contains peanut plants. 2D data like RGB images or videos can be easily captured with low-cost cameras and does not take hours to capture high-resolution data of a single plant. As a result, it also requires less human supervision and handling. After the 2D data collection, we train 3D radiance fields using NeRF and obtain an implicit representation of the scene where we could sample 3D point clouds. Our experiment shows that the 3D reconstruction of peanut plants modeled by PeanutNeRF is more accurate, with higher resolution (capturing finely detailed structures such as peanut plant peg) than that achieved from the traditional approach of LiDAR + manual cleaning.

Powered by the 3D scene model from PeanutNeRF, we conduct a preliminary study of plant phenotyping of peanuts to study important traits like node count, and flowering which depends upon thin plant parts like pegs and pods and their locations. We conduct experiments on both defoliated (leafless) and foliated (naturally leafy) plants where peanut pods are more often to be occluded. A comparison is in Figure 2. The pod detection in 3D data format is important to achieve accurate phenotyping and improved yield. We use Frustum PVCNN [18] to perform 3D peanut pods detection and instance segmentation, from the 3D point cloud sampled from the PeanutNeRF model. These initial results show that our approach is able to detect most peanut pods with an IoU of more than 0.5.

Our main contributions are:

Figure 2. Defoliated (left) and foliated (right) plant sample.

1. A novel collection of data for 360-degree and frontal view video clips as well as extraction of overlapping images for each video for 12 plants including defoliated and foliated plants.

2. A neural radiance fields-based 3D analysis pipeline for efficient and reliable reconstruction of peanut plants.

3. 3D peanut pod detection and segmentation from PeanutNeRF point clouds using Frustum PVCNN.

2. Related work

In terms of data acquisition, sensors such as LiDAR, Kinect, and Intel Realsense can capture 3D information. [2, 16, 30–32] but are costly and limited in their ability to capture thin plant parts like pegs. The resulting point cloud is sparse and contains noise. The 3D scans from different angles need to be aligned using a registration algorithm by matching key points in different scans. For capturing thin plant parts, X-Ray technology is used in various studies involving root phenotyping [6, 11]. In contrast to data acquisition through 3D sensors, 2D sensors allow cheap, and less labor-intensive data collection. Moreover, the result contains negligible noise. Various studies in the past utilized 2d multi-view images to perform 3D reconstruction. 3D multi-view stereo reconstruction was used for

cassava tree crown phenotyping and blueberry harvestability trait extraction using Agisoft software [23, 35]. Shape-from-silhouette method was used for the reconstruction of plants [29] through the extraction of silhouettes of objects of interest from multiple views. Simple shape carving was used to reconstruct the roots of rice seedlings but it removed the background. Structure from Motion (SfM) has been used as it outputs sparse point clouds as well as camera poses of input images. [9, 17, 23, 24]. Given the camera poses, multi-view stereo has been adopted to achieve dense point clouds of various plant types [10, 14, 25, 37]. Recent advancements allow representing 3D volume as weights of a neural network. Neural radiance fields [20] allows neural implicit representation of a 3D shape. The density and color of a 3D location are implicitly encoded in the network that takes as input, a 3D coordinate and viewing direction. The objective of NeRF is to perform novel view synthesis. It allows 3D reconstruction, due to the implicit representation of 3D shapes.

Detection of plant parts has been achieved in a variety of ways. Segmentation was used to identify parts using traditional processing methods like cylinder fitting, color-based region growth segmentation, and voxel cloud connectivity segmentation [31, 33]. Hand-crafted features like fast point feature histogram (FPFH) and principal curvature in SVM, Random forest, and decision algorithms [40]. Later studies utilized deep learning to perform both semantic and instance segmentation [15, 19]. However in these studies, the plant parts were at a minimum distance from each other, therefore individual instances were identified through clustering. In peanut plants, the pods are nearby each other. Therefore using the clustering approach can identify more than one nearby pod as a single instance. Therefore we adopt 3D detection to detect each individual pod and segment the pod in the detected bounding box.

3. PeanutNeRF: the approach

In this section, we describe our pipeline of utilizing the NeRF-type implicit scene modeling method for peanut reconstruction and analysis tasks. We first give an overview of NeRF and its Nerfacto [36] variant used for the reconstruction of peanut plants. Then we describe how to perform downstream analysis tasks such as pod detection, from the obtained radiance field model.

3.1. Neural Radiance Fields (NeRF)

NeRF implicitly represents a 3D scene using learned continuous volumetric radiance field F_θ on a set of images and camera poses to synthesize novel views not in the training set. Specifically, F_θ is modeled using trained MLPs that take a continuous 5D vector input including spatial coordinates $\hat{x} = (x, y, z)$ and viewing direction $d = (\theta, \phi)$ and outputs density $\sigma(\hat{x})$ and view-dependent

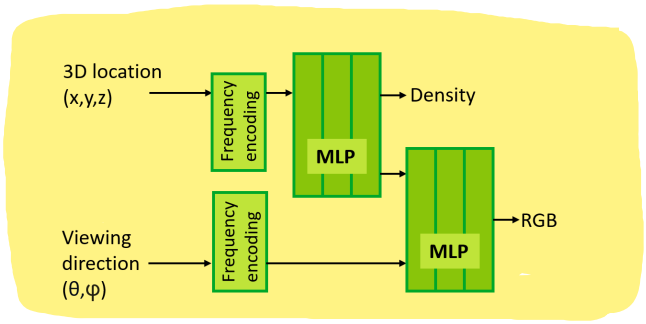


Figure 3. NeRF model architecture. Given a 3D query and a viewing direction, the NeRF predicts the density and color.

radiance $c = (r, g, b)$ at that spatial location. NeRF performs differentiable volume rendering of an implicit 3D scene onto 2D images to optimize the MLPs using a loss function that compares the predicted pixel and ground truth pixel values. To compute the color of a single pixel, Let a ray $[r(t) = o + td]$ be emitted through the center of projection of camera space o through a given pixel on the image plane, traversing between near t_n and far bounds t_f. Uniform sampling is used to select K quadrature points ${t_k}_{k=1}^{K}$ between t_n and t_f, NeRF's approximation of expected color $[\hat{C}(r)]$ for that pixel is given by:

$$\hat{C}(r) = R(r, c, \sigma) = \sum_{k=1}^{K} T(t_k)\alpha(\sigma(t_k)\delta)c(t_k), \quad (1)$$

where

$$T(t_k) = \exp\left(- \sum_{k'=1}^{k-1} \sigma(t_{k'})\delta_{k'} \right), \quad (2)$$

$c(t)$ and $\sigma(t)$ are color and density at point $r(t)$, $\alpha(x) = 1 - \exp(-x)$ and $\delta_k = t_{k+1} - t_k$ is the distance between two quadrature points. Stratified sampling is used to select quadrature points between t_n and t_f, the near and far planes of the camera.

Using MLPs with ReLU, the density $\sigma(t)$ is modeled only as the function of spatial coordinates while the color $c(t)$ is modeled as the function of both spatial coordinates and viewing direction as illustrated in NeRF model architecture in Figure 3. To improve the neural network's performance, the input viewing direction d and spatial position $r(t)$ are encoded in higher dimensions before being fed to NeRF field. The following equation describes the form of MLPs utilized:

$$[\sigma(t), z(t)] = MLP_{\theta 1}(\gamma_x(r(t))) \quad (3)$$

$$c(t) = MLP_{\theta 2}(z(t), \gamma_d(d)) \quad (4)$$

Using NeRF, the point clouds having the desired number of points can be reconstructed. Moreover, it reduces memory consumption by compression of high-resolution point

clouds as they are represented by the trained weights of MLP.

To optimize the MLP weight, the sum of squared error loss is used with respect to RGB image collection. Each image is paired with its intrinsic and extrinsic camera parameters estimated using structure from motion. A set of camera rays are pre-computed corresponding to pixel j in image i with each ray emitting from 3D location oi and through the pixel$_{ij}$ with direction d_{ij}. In addition to using the described radiance field with randomly sampled quadrature points, the weights corresponding to each point result from a volume renderer that correlates the importance of each point in the final rendering of the image. Therefore these weights resulting from the first radiance field (coarse network) are used to guide further sampling of points. The points biased towards regions of higher weights are fed as input to another radiance field called fine network which is similar in structure to the coarse network.

3.2. Nerfacto

While NeRF uses a coarse network (a NeRF field) to guide the sampling of more relevant points, the Nerfacto approach [36] utilizes a proposal sampler to sample the locations of the regions that contribute most to the final render. The proposal sampler consists of multiple density fields to represent the coarse density. Each density field is modeled as a small fused MLP and takes hash encoding of the input. The density function does not need to learn the high-frequency details during the initial passes. It is used to guide the sampling of relevant locations. The sampled locations resulting from the proposal sampler are fed as input to the Nerfacto field to predict the final density and radiance for the rendering. Different from the NeRF field, the Nerafacto field uses hash encoding to encode the spatial position and spherical harmonics encoding to encode the viewing direction as represented in Nerfacto model architecture in Figure 4. The encoded position and direction are fed as input to MLPs in a manner similar to NeRF field as described in Equations (3) and (4) except that the nerfacto also takes as input, appearance embeddings in MLP to output the color as shown in Equation (4). These appearance embeddings are trainable and are optimized alongside MLP parameters (θ). The loss function based on the nerfacto field is:

$$L = \sum_{ij} \left\| C(r_{ij}) - \hat{C}(r_{ij}) \right\|_2^2 \tag{5}$$

3.3. Frustum PVCNN

In this subsection, we demonstrate how to perform important phenotypic analysis using the learned 3D radiance field from the subsections. In particular, we design Frustum PVCNN that is similar to Frustum pointnet, except that it

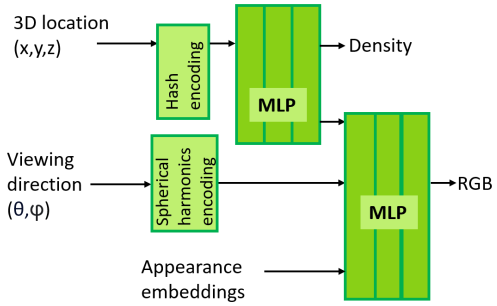


Figure 4. Nerfacto model architecture. Given the 3D position, viewing direction, and appearance embeddings, the Nerfacto model predicts the density and color.

uses PVCNN [18] as the backbone module for feature aggregation. Overall, it consists of three modules: Frustum proposal, mask segmentation, and 3D bounding box estimation.

3.3.1 Frustum proposal

In this module, we leverage a 2D detector to detect peanut pods from frontal view images. Specifically, we use yolov5 detector whose weights are trained on ImageNet data before applying transfer learning to detect peanut pods in our data. After detecting the proposed regions for pods in 2D images, we use the corresponding 3d reconstruction from PeanutNeRF. This is done so that the proposed region of pod in 2D image can be lifted to the corresponding 3D frustum. After estimating the 3D frustum, we collect all points in between the frustum to obtain frustum point clouds.

3.3.2 Mask segmentation

The obtained frustum point clouds contain points belonging to the target pod, as well as irrelevant categories of foreground occlusion and background clutter. We train PVCNN as a pointwise binary segmentator, to distinguish between the target object (pod) and distractors (frontal occlusion and background clutter). PVCNN utilizes point-voxel convolution for feature aggregation in point clouds. After performing the binary segmentation, we filter the points belonging to the target pod. Performing binary segmentation for each frustum allows us to achieve the pod instance segmentation task.

3.3.3 3D bounding box estimation

Given the segmented mask of pods from the previous step, we further process the data for 3D bounding box estimation. The 3D bounding box is paramterized by its center (c_x, x_y, c_z) and size (l, w, h). The "residual" approach is adopted for box dimension estimation. We use a template

with pre-defined box dimensions and train the pointnet (referred to as the box estimation net) for regression of residual dimensions. The regression for the box center is achieved in two steps. In the first step, we use pointnet (referred to as T-Net) to regress the center residual. This is performed since in some cases, the center of the bounding box slightly differs from the center of the instance mask. Therefore this slight difference is regressed by the T-net as center residual. Additionally, the box estimation net containing outputs for box dimension also contains outputs for center residual from instance mask. The residual is combined with the center residual predicted from T-net and masked points' centroid to recover an absolute center as represented in Equation (6):

$$C_{\text{pred}} = C_{\text{mask}} + \Delta C_{\text{t-net}} + \Delta C_{\text{box-net}} \tag{6}$$

The three networks involving mask segmentation PVCNN, T-net, and box estimation point net are simultaneously optimized with multi-task losses in Equation (7). L_{mask} is used for instance segmentation by PVCNN, L_{c1-reg} is used for T-net, L_{c2-reg} is used for center regression of box estimation net. L_{s-reg} are for box size. The corner loss represents the sum of distances between the 8 corners of predicted and ground truth boxes. Softmax is used for the classification task and smooth l_1 (Huber) loss is used for regression cases.

$$\begin{aligned} L_{\text{multi-task}} = L_{\text{mask}} + \lambda(L_{c1-\text{reg}} + L_{c2-\text{reg}} \\ L_{s-\text{reg}} + \gamma L_{\text{corner}}) \end{aligned} \tag{7}$$

4. Experiments

In this section, we first evaluate the quality of the 3D reconstruction of PeanutNeRF and compare it against the standard LiDAR approach. To demonstrate the power of PeanutNeRF, we present a preliminary study of using the model learned with PeanutNeRF to help peanut phenotypic analysis–the detection and segmentation of peanut pods.

4.1. Data collection

The data collection was performed for defoliated and foliated peanut plants using a customized setup. Peanut plants are different from many other plants as the plant cannot stand straight by itself being kept in a pot. Moreover, the important parts like pods fruit are below the ground while some important parts like pegs and nodes are above the ground. To allow the scanning of the peanut plant in a way to capture the important parts of the plant along with the whole plant architecture, a cloth dryer was installed with a rod attached to the ceiling and the plant was fixed using the cloth dryer. To set up the plant, the branches of the plants were secured by cloth dryer clips. In this way, the plant's branches are widely spread around so that the plant's architecture along with pods and pegs were mostly detectable by the sensor as shown in Figure 5. To perform the 3D reconstruction of the frontal view of the plant, plain cloths are arranged in the background. The video clips covering a 360-degree view of the plant do not have a plain background.

Figure 5. Data collection setup.

4.2. Experimental setting

The Nerfacto model was trained separately for each plant's data. The overlapping images are extracted and Colmap [27, 28] is used to extract the poses of each image. To perform the 3D reconstruction of video covering the entire plant with a 360-degree view, camera poses were selected from the validation set images and ray tracing was performed. The Nerfacto model was queried in this to predict RGB value and density for each query during the ray tracing process. For frontal view reconstruction, the camera poses were selected manually using NeRF Studio viewer tool [36] so that the plant covered most of the image. In the case of the frontal view plant, the Nerfacto model was used to extract the image as well as the corresponding point cloud for each selected camera pose. Our PeanutNeRF model was trained with an Adam optimizer with a learning rate of 0.01. The training was performed for 30,000 iterations. Each training iteration used a batch of 4096 rays where as the NeRF model was trained with rectified Adam optimizer with a learning rate of 0.0005. The training was performed for 1,000,000 iterations and each iteration used a batch of 1024 rays.

To compare the performance of PeanutNeRF, we use a FARO LiDAR scanner to perform a more standard 3D modeling task of the peanut plant. We performed three LiDAR scans for each selected plant sample. The scans are registered using the FARO Scene software to achieve the 3D model. We normalize both 3D models before comparison so that the result achieved using the LiDAR scans and that from PeanutNeRF are at the same scale.

To train the frustum PVCNN, we prepared a dataset using a selected defoliated and foliated plant. Reconstruction was performed for 7 frontal views from different camera

poses for each plant. The reconstructed point clouds and their corresponding 2D images are collected. The dataset is annotated by first labeling the bounding boxes in the 2D image and then extracting the frustum point cloud corresponding to the 2D bounding box. We have a total of 172 pod samples (train/val split of 150/22) for defoliated and 129 pod samples (train/val split of 109/20) for the foliated plant from 7 views of each plant. As a preprocessing step, all pod sample coordinates were translated by subtracting all coordinates in a pod from the maximum coordinates in that pod sample. We did not normalize as normalization can affect the size as well as the network's regression output. The frustum PVCNN is trained for 500 epochs with a batch size of 4. Adam optimizer was used with a learning rate of 0.01 and a cosine annealing learning rate scheduler was used. The network was trained separately for defoliated and foliated categories. During validation, we evaluated the performance of Frustum PVCNN on frustums in the validation set which are extracted from manually labeled 2D bounding boxes and precision refers to the percentage of predictions above a certain threshold. For inference on the test set, we first used yolov5 detector for performing the detection. The 2D detected pods are used to extract corresponding frustum point clouds which were input to Frustum PVCNN for mask segmentation and bounding box prediction.

4.3. Evaluation metrics

To evaluate the 3D reconstruction performance of PeanutNeRF, we measure two types of metrics.

First, we evaluate the image synthesis results from camera poses in the validation set and compare them with ground truth images. Peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS) are used.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{8}$$

where

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{9}$$

and MAX_i is the maximum pixel value of the image.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma x^2 + \sigma y^2 + c_2)} \tag{10}$$

where μ_x is the pixel sample mean of x; μy is pixel sample mean of y; σ^{x^2} the variance of x; σ^{y^2} the variance of y; σ^{xy} the covariance of x and y/; c_1 and c_2 are two variables to stabilize the division with weak denominators. L is the dynamic range of the pixel values. $k_1 = 0.01, k_P = 0.03$.

We also evaluate the 3D point cloud obtained from PeanutNeRF. To compare the level of noise, chamfer distance is used as shown in Equation (11) to estimate the distance of a noisy point cloud from its cleaned version.

$$\begin{aligned} CD(S_1, S_2) = &\frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 \\ &+\frac{1}{|S_2|}\sum_{y\in S_2}\min_{x\in S_1}\|x-y\|_2^2 \end{aligned} \tag{11}$$

where S_1 and S_2 represents the two point clouds.

The LPIPS [39] is used to estimate the similarity between the activations of two image patches. The high value of LPIPS corresponds to more distance and dissimilarity between the image patches while the low distance shows that the image patches are similar. For pod detection evaluation, we report the precision of pod samples with IoU thresholds of 0.5, 0.6, and 0.7 to determine the true positives.

4.4. Results and discussion

Our experiments show that the performance of Nerfacto is significantly higher than that of NeRF as shown in Table 1. The PSNR for Nerfacto exceeded 25 while that for NeRF was less than 18 for both foliated and defoliated plants. Similarly, Nerfacto showed a higher SSIM (more than 0.8). The LPSIS obtained using Nerfacto was less than half of that obtained using NeRF.

Type	Method	LPIPS	PSNR	SSIM
Defoliated	NeRF	0.4407	16.26	0.7587
	Nerfacto	0.1067	25.61	0.8642
Foliated	NeRF	0.4568	17.32	0.7194
	Nerfacto	0.1432	25.99	0.8089

Table 1. Performance of NeRF and Nerfacto on the defoliated and foliated plant.

Comparison of noise among the point cloud registered from LiDAR scans and that reconstructed using nerfacto show that LiDAR-based plant model contains more noise when measured in terms of the Chamfer distance. The noise in the foliated and defoliated plants are compared with the manually-cleaned versions. In addition, denoised versions were formed by applying the statistical outlier removal method with a nearest neighbor value of 6 and standard deviation multiplier (nSigma) value of 1, 5, and 10. Chamfer distance of denoised and raw version with the manually cleaned version is significantly lower in the case of nerfacto-based reconstruction compared to the model from LiDAR scans as illustrated in Table 2. The Chamfer distance in the case of LiDAR-based model is more than 5 times that of nerfacto-based model in both foliated and defoliated plants.

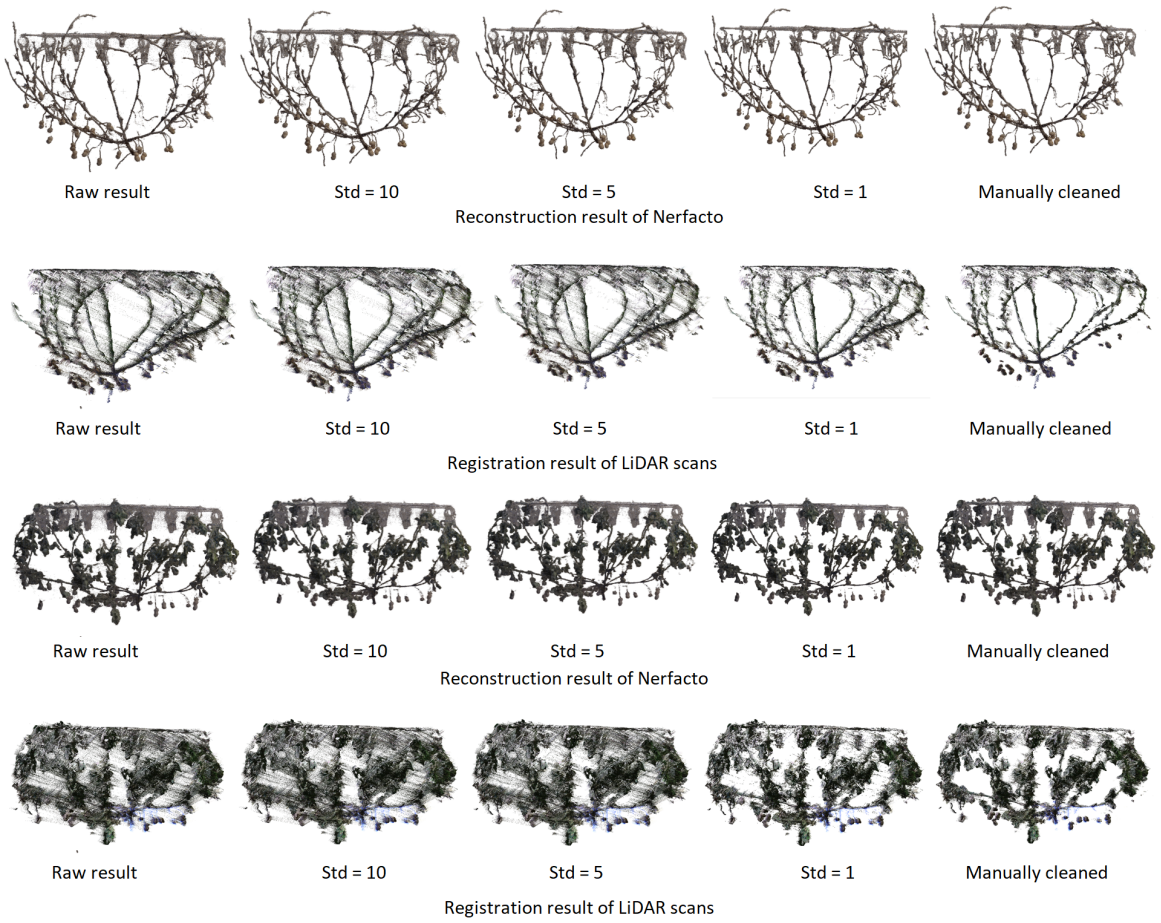


Figure 6. Comparison of registration results from LiDAR scans and Nerfacto for a defoliated plant (top two rows) and a foliated plant (bottom two rows). Samples contain raw data and results denoised using statistical outlier removal at std dev multiplier (Std/nSigma) value of 1, 5, and 10.

It was observed that by decreasing the nSigma value, the denoising becomes more aggressive, and the Chamfer distance from the manually cleaned version is the lowest. While with the highest nSigma value (10), the denoising is less aggressive and the distance is closer to the raw result. These quantitative results were also supported by the visualization of 3D models as illustrated in Figure 6. In addition, we observed that the thin parts including pegs in the defoliated plant were missed in the LiDAR capture while they were mostly captured in nerfacto based reconstructed as shown in Figure 6.

We also analyzed performance using four positional encodings in Nerfacto. Frequency encoding from the original study [20] was used which encodes input coordinates as sinusoidal of various frequencies. Hash grid encoding [21] was used that adapts to the training data distribution and inherits the benefits of the Hash table for efficient performance. In one blob encoding [22], a kernel is used to

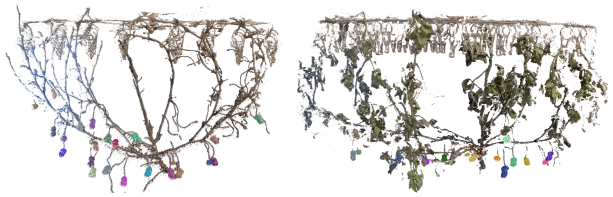
Type	Denoising level	Nerfacto	LiDAR
Defoliated	nSigma=1	0.97	28.13
	nSigma=5	2.11	42.81
	nSigma=10	2.82	44.39
	Raw result	3.39	44.55
Foliated	nSigma=1	1.81	14.86
	nSigma=5	2.24	30.17
	nSigma=10	3.2	32.78
	Raw result	3.66	33.18

Table 2. Chamfer distance (CD) between the manually cleaned point cloud and the raw and denoised data from Nerfactor and LiDAR. Denoising is performed using statistical outlier removal with standard deviation multiplier (nSigma) values of 1, 5, and 10. CD is multiplied by 10^4.

activate multiple adjacent entries instead of a single one. Spherical harmonics (SH) [7] was used to apply the encoding of high-frequency functions on manifolds. It was observed that the choice of positional encoding significantly impacts the performance metrics for both foliated and defoliated plants. (Table 3). We observed that using Hash grid-based positional encoding can achieve better performance. The PSNR achieved using Hash grid encoding was more than 25 while it was less than 23 using other encodings. The LPSIS using spherical harmonics, frequency, and one blob encoding was more than 3 times that achieved using hash encoding. Similarly, the SSIM using Hash encoding exceeded 0.8 and was higher than other encodings.

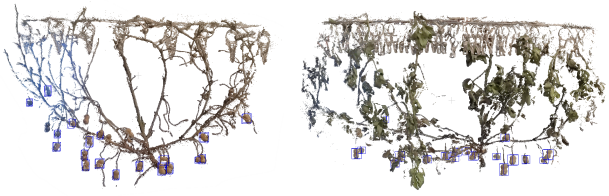
Type	PE	LPIPS	PSNR	SSIM
Defoliated	Hash grid	0.1067	25.61	0.8642
	SH	0.4263	19.52	0.7808
	Frequency	0.4313	20.88	0.7795
	One blob	0.3964	22.27	0.7847
Foliated	Hash grid	0.1432	25.99	0.8089
	SH	0.4609	20.22	0.7288
	Frequency	0.4635	20.45	0.7283
	One blob	0.4815	22.07	0.7417

Table 3. Nerfacto performance at different Positional (PE) encodings. (SH represents spherical harmonics encoding.)



(a) (b)

Figure 7. Visualization of mask instance segmentation. The predicted mask in each frustum is represented by a different color.



(a) (b)

Figure 8. Pod detection results on the validation set. The bounding boxes are represented in blue outline.

In the task of 3D pod detection, the network showed almost similar performance for both foliated and defoliated categories and the precision achieved at the IoU threshold of 0.5 is around 70% and it is more than 45% as the IoU threshold is increased to 0.7 (Table 4). The mask instance segmentation shows a mean IoU of more than 0.8 and most masks for pods were correctly predicted as shown in Figure 7. In some cases, mask segmentation showed errors. Therefore, the bounding box dimension and center regression showed high error from ground truth and the IoU of the predicted box with the ground truth box was less than 0.4. These bounding boxes were considered as missed detection in the validation dataset. The detected pods on the validation set showed mostly detected pods and few missed pods as illustrated in Figure 8. The proposed method has some limitations. Using frustum PVCNN, the pods are detected only in frontal-view point clouds. Additionally, the nerfacto model has to be trained per plant.

IoU threshold	Defoliated	Foliated
0.5	0.72	0.7
0.6	0.63	0.65
0.7	0.45	0.5

Table 4. Precision on the validation set at different IoU thresholds.

5. Conclusion

We presented the use of NeRF for 3D reconstruction of peanut plants. The Nerfacto method showed better results than the original NeRF and is able to recover most of the thin plant parts. We trained Frustum PVCNN on the frontal view reconstructed point clouds and corresponding 2D images of the selected defoliated plant to achieve 3D pod detection. In future studies, we aim to perform detection from reconstruction covering a 360-degree view of the plant. In addition to pod detection, we aim to detect the thin parts including pegs recovered from the 3D reconstruction.

Acknowledgements

We thank Jacob Brannon, Javier Rodriguez-Sanchez, Orr Shalev, and all Biosensing Automation and Intelligence lab members for useful discussions and help in data collection, and experiments.Additionally, we gratefully thank for the computing resources and technical expertise from the Georgia Advanced Computing Resource Center (GACRC). The first author is supported by Higher Education Commission under US-Pakistan Knowledge Corridor.

References

[1] Fas usda (https://ipad.fas.usda.gov). 1

[2] Zurui Ao, Fangfang Wu, Saihan Hu, Ying Sun, Yanjun Su, Qinghua Guo, and Qinchuan Xin. Automatic segmentation of stem and leaf components and individual maize plants

in field terrestrial lidar data using convolutional neural networks. *The Crop Journal*, 10(5):1239–1250, 2022. 1, 2

[3] Suchet Bargoti and James P Underwood. Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039–1060, 2017. 1

[4] Ayan Chaudhury and Christophe Godin. Skeletonization of plant point cloud data using stochastic optimization framework. *Frontiers in Plant Science*, 11:773, 2020. 1

[5] Sruti Das Choudhury, Saptarsi Goswami, Srinidhi Bashyam, Ashok Samal, and Tala Awada. Automated stem angle determination for temporal plant phenotyping analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2022–2029, 2017. 1

[6] Keith E Duncan and Christopher N Topp. Phenotyping complex plant structures with a large format industrial scale high-resolution x-ray tomography instrument. In *High-Throughput Plant Phenotyping: Methods and Protocols*, pages 119–132. Springer, 2022. 2

[7] Carlos Esteves, Tianjian Lu, Mohammed Suhail, Yi-fan Chen, and Ameesh Makadia. Generalized fourier features for coordinate-based learning of functions on manifolds. 8

[8] Mathieu Gaillard, Chenyong Miao, James Schnable, and Bedrich Benes. Sorghum segmentation by skeleton extraction. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 296–311. Springer, 2021. 1

[9] Jordi Gené-Mola, Ricardo Sanz-Cortiella, Joan R Rosell-Polo, Josep-Ramon Morros, Javier Ruiz-Hidalgo, Verónica Vilaplana, and Eduard Gregorio. Fruit detection and 3d location using instance segmentation neural networks and structure-from-motion photogrammetry. *Computers and Electronics in Agriculture*, 169:105165, 2020. 3

[10] Jingwei Guo and Lihong Xu. Automatic segmentation for plant leaves via multiview stereo reconstruction. *Mathematical Problems in Engineering*, 2017, 2017. 3

[11] Tsung-Han Han and Yan-Fu Kuo. Developing a system for three-dimensional quantification of root traits of rice seedlings. *Computers and Electronics in Agriculture*, 152:90–100, 2018. 2

[12] Pasupuleti Janila, SN Nigam, Manish K Pandey, P Nagesh, and Rajeev K Varshney. Groundnut improvement: use of genetic and genomic tools. *Frontiers in plant science*, 4:23, 2013. 1

[13] Pasupuleti Janila, Murali T Variath, Manish K Pandey, Haile Desmae, Babu N Motagi, Patrick Okori, Surendra S Manohar, AL Rathnakumar, T Radhakrishnan, Boshou Liao, et al. Genomic tools in groundnut breeding program: status and perspectives. *Frontiers in Plant Science*, 7:289, 2016. 1

[14] Dawei Li, Guoliang Shi, Weijian Kong, Sifan Wang, and Yang Chen. A leaf segmentation and phenotypic feature extraction framework for multiview stereo plant point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:2321–2336, 2020. 3

[15] Dawei Li, Guoliang Shi, Jinsheng Li, Yingliang Chen, Songyin Zhang, Shiyu Xiang, and Shichao Jin. Plantnet: A dual-function point cloud segmentation network for multiple plant species. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:243–263, 2022. 3

[16] Yinglun Li, Weiliang Wen, Teng Miao, Sheng Wu, Zetao Yu, Xiaodong Wang, Xinyu Guo, and Chunjiang Zhao. Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning. *Computers and Electronics in Agriculture*, 193:106702, 2022. 1, 2

[17] Xu Liu, Steven W Chen, Shreyas Aditya, Nivedha Sivakumar, Sandeep Dcunha, Chao Qu, Camillo J Taylor, Jnaneshwar Das, and Vijay Kumar. Robust fruit counting: Combining deep learning, tracking, and structure from motion. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1045–1052. IEEE, 2018. 3

[18] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2, 4

[19] Liyi Luo, Xintong Jiang, Yu Yang, Eugene Roy Antony Samy, Mark Lefsrud, Valerio Hoyos-Villegas, and Shangpeng Sun. Eff-3dpseg: 3d organ-level plant shoot segmentation using annotation-efficient point clouds. *arXiv preprint arXiv:2212.10263*, 2022. 3

[20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3, 7

[21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 7

[22] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019. 7

[23] Xueping Ni, Changying Li, Huanyu Jiang, and Fumiomi Takeda. Three-dimensional photogrammetry with deep learning instance segmentation to extract berry fruit harvestability traits. *ISPRS Journal of Photogrammetry and Remote Sensing*, 171:297–309, 2021. 3

[24] Yeping Peng, Mingbin Yang, Genping Zhao, and Guangzhong Cao. Binocular-vision-based structure from motion for 3-d reconstruction of plants. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021. 3

[25] Johann Christian Rose, Stefan Paulus, and Heiner Kuhlmann. Accuracy analysis of a multi-view stereo approach for phenotyping of tomato plants at the organ level. *Sensors*, 15(5):9651–9665, 2015. 3

[26] Hanno Scharr, Massimo Minervini, Andrew P French, Christian Klukas, David M Kramer, Xiaoming Liu, Imanol Luengo, Jean-Michel Pape, Gerrit Polder, Danijela Vukadinovic, et al. Leaf segmentation in plant phenotyping: a collation study. *Machine vision and applications*, 27:585–606, 2016. 1

[27] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[28] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for un-

structured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[29] Weinan Shi, Rick van de Zedde, Huanyu Jiang, and Gert Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019. 3

[30] Guoxiang Sun and Xiaochan Wang. Three-dimensional point cloud reconstruction and morphology measurement method for greenhouse plants based on the kinect sensor self-calibration. *Agronomy*, 9(10):596, 2019. 1, 2

[31] Shangpeng Sun, Changying Li, Peng W Chee, Andrew H Paterson, Yu Jiang, Rui Xu, Jon S Robertson, Jeevan Adhikari, and Tariq Shehzad. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160:195–207, 2020. 1, 2, 3

[32] Shangpeng Sun, Changying Li, Peng W Chee, Andrew H Paterson, Cheng Meng, Jingyi Zhang, Ping Ma, Jon S Robertson, and Jeevan Adhikari. High resolution 3d terrestrial lidar for cotton plant main stalk and node detection. *Computers and electronics in agriculture*, 187:106276, 2021. 1, 2

[33] Shangpeng Sun, Changying Li, Andrew Paterson, and Peng Chee. Three-dimensional cotton plant shoot architecture segmentation and phenotypic trait characterization using terrestrial lidar point cloud data. In *2020 ASABE Annual International Virtual Meeting*, page 1. American Society of Agricultural and Biological Engineers, 2020. 3

[34] Shangpeng Sun, Changying Li, Andrew H Paterson, Peng W Chee, and Jon S Robertson. Image processing algorithms for infield single cotton boll counting and yield prediction. *Computers and electronics in agriculture*, 166:104976, 2019. 1

[35] Pongsakorn Sunvittayakul, Piya Kittipadakul, Passorn Wonnapinij, Pornchanan Chanchay, Pitchaporn Wannitikul, Sukhita Sathitnaitham, Phongnapha Phanthanong, Kanokphu Changwitchukarn, Anongpat Suttangkakul, Hernan Ceballos, et al. Cassava root crown phenotyping using three-dimension (3d) multi-view stereo reconstruction. *Scientific Reports*, 12(1):10030, 2022. 3

[36] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, et al. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023. 3, 4, 5

[37] Sheng Wu, Weiliang Wen, Yongjian Wang, Jiangchuan Fan, Chuanyu Wang, Wenbo Gou, and Xinyu Guo. Mvs-pheno: a portable and low-cost phenotyping platform for maize shoots using multiview stereo 3d reconstruction. *Plant Phenomics*, 2020, 2020. 3

[38] Sheng Wu, Weiliang Wen, Boxiang Xiao, Xinyu Guo, Jianjun Du, Chuanyu Wang, and Yongjian Wang. An accurate skeleton extraction approach from 3d point clouds of maize plants. *Frontiers in plant science*, 10:248, 2019. 1

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[40] Illia Ziamtsov and Saket Navlakha. Machine learning approaches to improve three basic plant phenotyping tasks using three-dimensional point clouds. *Plant physiology*, 181(4):1425–1440, 2019. 3