agronomy

MDPI

Article

Supplementary Material

Yi Gou [1,2], Xin Tan [1,3]*, Mingyu Yang [1]*, Xin Zhang [1,2], Liang Xu [1], Qingbin Jiao [1], Sijia Jiang [1], Ding Ma [1] and Junbo Zang [4]

[1] Changchun Instiute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, Jinlin 130033, China
[2] University of Chinese Academy of Sciences, Beijing 100049,China
[3] Center of Materials Science and Optoelectronics Engineering, Chinese Academy of Sciences, Beijing 100049, China
[4] Jilin Province Product Quality Supervision and Inspection Institute 130051, CHina
* Correspondence: xintan_grating@163.com (X.T.); yangmingyu@ciomp.ac.cn (M.Y.)

S1. 3D Reconstruction Implementation

S1.1. Multi-Resolution Hash Encoding and Scene Representation

The Auto3DPheno-3D reconstruction adopts a multi-resolution hash encoding scheme. The scene is bounded within a unit cube (with coordinate values ranging from 0 to 7) and divided into hierarchical voxel grids. For a vertex point x in the maize seedling image, adjacent voxels are first located at different resolution levels, and then hash indices are assigned to the vertices of these voxels via a hash mapping approach. Given the 3D point coordinate in the scene, it is mapped to hash tables of different resolutions through the hash encoding function $h(x)$, enabling an efficient representation of the volumetric radiance field. Suppose there are L resolution levels in total, and the grid resolution of each level is r_l ($l = 1, 2, \cdots, L$). For grids of different resolutions, a hash table T with a dimension of F is maintained respectively. Based on the generated grid vertex indices, the corresponding F-dimensional feature vectors are retrieved from the respective hash tables. Subsequently, linear interpolation is performed on the obtained feature vectors according to the relative position of x in grids of different resolutions. Through this hash encoding mechanism, the model can capture geometric details spanning from macroscopic to microscopic scales, achieving a multi-scale representation of the scene, which is in line with the core design of Instant-NGP for efficiently encoding scene geometry.

S1.2. Ray Casting and Point Sampling

Rays are cast from the estimated camera poses through each pixel of the input images. Assuming that the origin of a ray in the camera coordinate system is o and the direction is d, the ray equation can be expressed as $r(t) = o + td$, where t represents the propagation distance of the ray. Within the scene bounds, N equally-spaced sample points $x_i = r(t_i)$ ($i = 1, 2, \cdots, N$) are taken along the ray, and t_i is the propagation distance corresponding to the i-th sample point.

S1.3. Network Prediction and Volume Rendering

At the sample point x_i, the network F_Θ predicts the color c_i and density σ_i, where Θ represents the network parameters.

The feature vectors linearly interpolated at each resolution are concatenated with the camera pose information ζ encoded using spherical harmonics to obtain the input information y. Here, the feature dimension of the camera pose information is E, and this

input information serves as the input to the small neural network. The color prediction can be written as $c_i = F_\Theta^c(x_i)$ and the density prediction as $\sigma_i = F_\Theta^\sigma(x_i)$.

Using volume rendering techniques, these predicted values are integrated to synthesize the pixel color. Let the cumulative transmittance of the ray r corresponding to a pixel up to t_i be $T(t_i) = \prod_{j=1}^{i-1}(1 - \sigma_j\delta_j)$, where δ_j is the distance between adjacent sample points. Then the final synthesized color $\hat{C}_r$ of the pixel is given by:

$$\hat{C}_r = \sum_{i=1}^{N} T(t_i)(1 - \exp(-\sigma_i\delta_i))c_i \tag{S1}$$

The loss gradient is back-propagated, and numerical accumulation and encoding training are completed during the search for feature vectors, outputting the RGB color and volume density of the sampling point.

S1.4. Loss Calculation and Network Optimization

The predicted pixel color $\hat{C}_r$ is compared with the pixel color C_r of the ground truth image, and the network parameters Θ are optimized by minimizing the photometric reconstruction loss. A commonly used photometric reconstruction loss is the Mean Squared Error (MSE), which is formulated as:

$$\mathcal{L}(\Theta) = \frac{1}{M}\sum_{r=1}^{M} \|\hat{C}_r - C_r\|^2 \tag{S2}$$

where M is the total number of rays involved in the training. Through the backpropagation algorithm, the network parameters Θ are updated according to the gradient of the loss function, continuously adjusting the network predictions to make them closer to the real images.

During the training process, key hyperparameters (related to l, N_0, N_1, b), feature dimensions F, number of samples per ray, and learning rate need to be carefully adjusted to strike a balance between reconstruction accuracy and computational efficiency.

S2. Maize Seedling Segmentation Implementation

Auto3DPheno-RoI Segmentation extracts local features from input point cloud data at different hierarchical levels. The network includes three key steps: sampling layer, grouping layer, and feature extraction layer.

Firstly, the network takes point cloud data consisting of multiple three-dimensional coordinate points as input. To efficiently reduce computational load while preserving spatial structural features, the system employs the Farthest Point Sampling (FPS) strategy at each layer to select a subset of points from the original point cloud as centroids. Based on each centroid, the network uses a K-nearest neighbors or radius-based approach to divide the surrounding points into local neighborhoods. For each local region, the network extracts local features using an embedded PointNet structure and further enriches feature representation through the Multi-Scale Grouping (MSG) method, thus capturing spatial and geometric information at different scales. By stacking multiple "sampling-grouping-feature extraction" modules, the local and global features of the point cloud are gradually integrated.

For the segmentation task, the extracted features are propagated back to each original point through feature interpolation and skip connections, ensuring that every point obtains a detailed feature representation that combines both local and global contextual information. These point-wise features are then passed through fully connected layers or shared multi-

layer perceptrons (MLPs), and finally a softmax classifier is used to assign a semantic label to each point in the point cloud.

S3. 3D Phenotypic Parameter Extraction Implementation

S3.1. Laplacian Contraction-Based Skeleton Extraction Implementation

(1) Neighborhood and Euclidean Metric:

For each For each point in the maize seedling point cloud $P = \{p_1, p_2, \cdots, p_n\}$, the local neighbor $N(p_i)$ is determined via an adaptive k-nearest neighbors (KNN) algorithm. Specifically, $N(p_i)$ consists of the k closest points in P that exhibit the smallest Euclidean distances to p_i. The neighbor is defined as:

$$N(p_i) = \{p_j \mid p_j \in P, |N(p_i)| = k\} \tag{S3}$$

where $|N(p_i)|$ denotes the cardinality of $N(p_i)$. The distance $d(p_i, p_j)$ represents the Euclidean metric calculated as:

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{S4}$$

(2)Laplacian Matrix Construction:

After determining the neighborhood for each point, a classic constrained Laplacian matrix L of size $n \times n$ is constructed. The elements of L are defined as follows:

$$L_{ij} = \begin{cases} 1 - \sum_{k \in N(i)} \omega_{ik} & \text{if } i = j \\ -\omega_{ij} & \text{if } j \in N(i) \\ 0 & \text{otherwise} \end{cases} \tag{S5}$$

where the weight ω_{ij} is usually set to 1 (unweighted), or alternatively as $\omega_{ij} = \frac{1}{d(p_i - p_j)}$ to emphasize geometric proximity between points.

(3) Adaptive Point cloud contraction. The numerical values for the adaptive contraction parameters α and ε in Table S1 were determined through systematic empirical testing across various seedling growth stages.

Table S1. Detailed configuration of adaptive contraction parameters across different point cloud scales.

Point Count (N)	Amplification (α)	Termination (ε)
$N < 10^3$	1	0.0100
$10^3 \leq N < 10^4$	2	0.0070
$10^4 \leq N < 10^5$	5	0.0050
$10^5 \leq N < 5 \times 10^5$	5	0.0040
$5 \times 10^5 \leq N < 10^6$	5	0.0030

S3.2. Maize Phenotypic Calculation Implementation

The phenotypic parameters and their definitions are shown in Table S2, and the point cloud structure and characteristics corresponding to these phenotypic parameters are illustrated in Figure S1.

Table S2. Phenotypic Parameters

Phenotypic Parameter	Definition
Plant Height	Vertical distance from the stem-root to the highest point of the plant.
Leaf Length	Linear length from the leaf base to the leaf tip.
Stem-leaf Angle	Angle formed between the stem and the leaf.
Leaf Surface Area	The areas of faces on the leaf surface.
Leaf Number	Total number of fully expanded leaves on the plant.

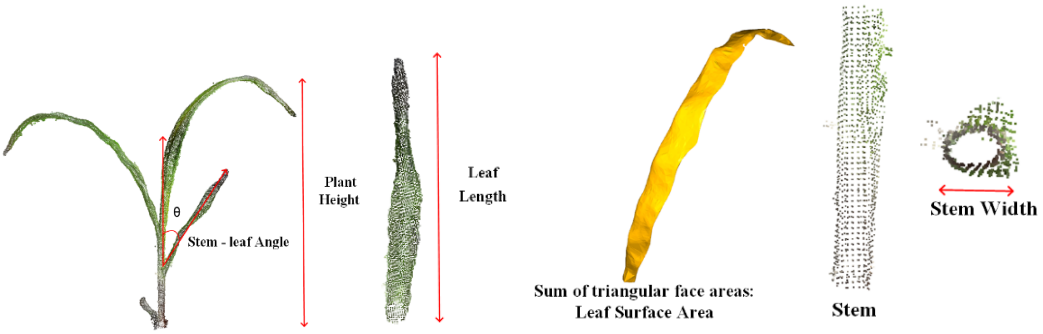


Figure S1. Phenotypic parameters point cloud structure and characteristics. (a) Stem-leaf angle and Plant height; (b) Leaf length; (c) Leaf surface area; (d) stem width.

Parameter Sensitivity Analysis for Leaf Clustering:
The clustering parameter ε significantly impacts the accuracy of leaf counting. As shown in Figure S2, a smaller radius ($\varepsilon = 0.05$) causes over-segmentation, where a single leaf is incorrectly split into multiple fragments. Conversely, a larger radius ($\varepsilon = 0.15$) leads to under-segmentation, causing adjacent leaves to merge into a single cluster. The optimized value of $\varepsilon = 0.1$ provides the highest stability, ensuring each leaf is correctly identified as an independent unit while effectively filtering noise through the $MinPts = 500$ constraint.

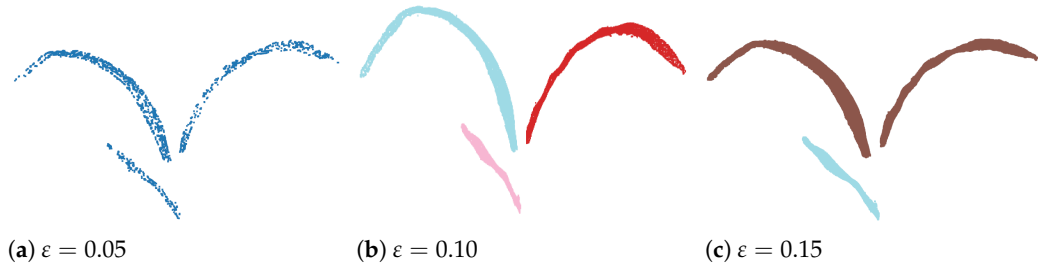


Figure S2. Sensitivity analysis of the DBSCAN clustering parameter ε for individual leaf segmentation.

Leaf Surface Area Physical Unit Conversion:
To convert the mesh surface area into real-world units, the following formula is applied:

$$\text{Leaf Area}_{\text{physical}} = \lambda^2 \times \text{Leaf Area}_{\text{point cloud}} \tag{S6}$$

Leaf Length Fitting and Physical Conversion:
We evaluated polynomial orders of 4th, 6th, 8th, 10th and 12th to determine the optimal model complexity for leaf skeleton fitting. As shown in Table S3, the fitting accuracy increases with higher polynomial orders. When the order is further increased to 12th, the MSE shows no significant decrease compared to the 10th-order result. Therefore, the

10th-order polynomial was selected to maintain the best balance between geometric fidelity and structural smoothness.

Following the curve generation, uniform sampling is performed to calculate the total length by accumulating Euclidean distances between adjacent points. The result is then converted to the physical length using the scale factor λ:

$$\text{Leaf Length}_{\text{physical}} = \lambda \times \text{Leaf Length}_{\text{point cloud}} \tag{S7}$$

Table S3. Fitting error of different polynomial orders for leaf skeleton point clouds.

Polynomial Order	MSE
4th	4.81×10^{-5}
6th	6.26×10^{-6}
8th	6.92×10^{-7}
10th (Selected)	$\mathbf{9.35 \times 10^{-8}}$
12th	8.12×10^{-8}

Stem Width Calculation and Physical Conversion:

The stem width S_w is determined by analyzing the stem point cloud P_{stem}. For each subset $P_{\text{stem}}(x, y, z)$ consisting of points with the same y-axis coordinate, the maximum Euclidean distance between any two points within the xoz plane is computed. This process is repeated across all y-axis heights, and the stem width is defined as the largest value among these maximum distances, formulated as:

$$S_w = \max_{y} \left\{ \max_{\substack{(x_1,z_1)\in P_{\text{stem}}(x,y,z) \\ (x_2,z_2)\in P_{\text{stem}}(x,y,z)}} \sqrt{(x_1 - x_2)^2 + (z_1 - z_2)^2} \right\} \tag{S8}$$