**Supplementary Material**

**Appendix A.1**

The specific steps of the plant main axis vector calculation method are shown in Algorithm 1.

|  |
| --- |
| **Algorithm 1** Compute Plant Main Axis Vector |
| **Input:** Point cloud of leaves , Number of main leaves  **Output:** Main axis vector *Dir*  1: // Create an array to store leaf direction vectors.  2: **for** in **do**  3: Get the leaf vein base and vein tip of  4: )  5: **if** [2] 0 **then**  6:  7: **end if**  8: **end for**  9:  10: // Create an array to store angles between leaf direction vectors and the temporary main axis vector  11: **for** in **do**  12: )  13:  14: **end for**  15: // Get the index of sorted angles.  16: // Reorder the array of leaf direction vectors.  17: // Take the mean of the first vectors.  18: **return** |

**Appendix A.2**

The specific steps for calculating the undulation of the leaf margin are shown in Algorithm 2.

|  |
| --- |
| **Algorithm 2** Compute the Undulation Degree of the Leaf Margin |
| **Input:** Flattened leaf point cloud , Maximum length threshold for excessively long triangle edges  **Output:** Mean slope of adjacent convex and concave points on leaf margin  1: // Trim the tail of the leaf.  2: // Apply Delaunay triangulation to the leaf point cloud to obtain indices of points  connected by edges of all triangles.  3: // Create an array to store filtered triangles.  4: **for** in **do**  5: **if** All edges of have length **then**  6:  7: **end if**  8: **end for**  9: // Create a dictionary to store the number of connections between pairs of points.  10: **for** in **do**  11: Use the indices of points connected by each edge of the triangle as in ，increment the corresponding starting from 0  12: **end for**  13: // Filter out in with value equal to 1, forming a contour map.  14: // Find the maximal connected subgraph in the contour map to determine the leaf margin contour map.  15: // Obtain margin points based on the leaf margin contour map.  16: // Compute the center point.  17: // Create an array to store distances from margin points to the center point.  18: **for** in **do**  19:  20: **end for**  21: // Perform simple filtering.  22: // Compute extreme points.  23: // Compute the slope of adjacent extreme points.  24:  25: **return** |

**Appendix A.3**

The corresponding phenotypic calculation methods used in this study are shown in the following Table 1.

Table 1 Phenotype Calculation Methods

|  |  |  |  |
| --- | --- | --- | --- |
|  | Phenotypic indicators | Definition | Calculation Methods |
| Plant phenotype | **Plant height** | The vertical distance from the top of the plant to the root | The difference in the Z-axis after plant correction |
| **Plant width** | The maximum width of the plant | The width is represented by the diameter of the minimum bounding circle |
| **Compactness** | Compactness of the plant | The ratio of plant width to plant height |
| **Projected area** | The vertical projection area of the plant | Based on the Graham's scanning algorithm and the cross product of triangle vectors |
| **Convex hull volume** | Volume of the convex polygon of the plant | Based on the Fast Incremental algorithm |
| **Concave hull volume** | Volume of the concave polygon of the plant | Compute the convex hull for each individual leaf and merge all the convex hulls together |
| **Voxel volume** | The volume of the plant after voxelization | Voxelization with 1mm grid size |
| **Number of leaves** | The number of leaves on a single lettuce plant | Count the number of instances segmented from the lettuce leaf |
| Leaf phenotype | **Leaf length** | The longest length of a leaf | Calculate the distance between adjacent points in the leaf vein point cloud and sum them up |
| **Leaf width** | The maximum width of a leaf | Flatten the leaf point cloud parallel to the X-axis and calculate the difference along the Y-axis |
| **Leaf area** | The surface area of a single leaf | Use the Shoelace formula and multiply the coordinates of adjacent edge contour points in turn |
| **Leaf inclination angle** | The angle between the normal of the leaf surface and the Z-axis | Calculate the angle between the leaf direction vector and the XOY plane |
| **Leaf azimuth angle** | The spatial orientation of the leaf | Calculate the angle between the leaf direction vector and the positive X-axis |
| **Leaf** **margin perimeter** | The length of the leaf margin | Construct a region adjacency graph to obtain complete margin points and calculate the perimeter of the leaf margin |
| **Leaf margin undulation** | The undulation degree of the leaf margin | Search for concave and convex points on the leaf margin and calculate the average slope between adjacent concave and convex points |
| **Number of leaf margin incisions** | Number of incisions of the leaf margin (only for Oakleaf lettuce) | Count the number of incisions of the margin based on the set of concave and convex points along the margin |

**Appendix A.4**

This study integrates the segmentation of individual lettuce point clouds with the phenotype calculation of plant and leaf into a powerful software system named LettuceP3D. The software system is developed based on the PyQt framework, offering excellent cross-platform compatibility, and can run smoothly on both Windows and Linux operating systems. During development and testing, a computer equipped with an Intel i7-13700 CPU, 16GB of RAM, and an NVIDIA GeForce RTX 3060 Ti GPU was used.

The LettuceP3D system contains several key modules that enable comprehensive analysis and phenotyping of lettuce. Point Cloud Data Import and Preprocessing Module is responsible for importing the collected point cloud data of lettuce into the system and preprocessing the data. Point Cloud Segmentation Module implements semantic segmentation of individual plants and instance segmentation of leaves. 3D Phenotype Analysis Module is divided into two sub-modules for calculating plant phenotype and leaf phenotype. Data Visualization Module enables viewing of 3D models of the data, supporting txt, obj, and ply formats; it also allows for viewing results in csv format obtained from phenotype analysis. Running Log Display Module is used to record key information during system operation. Running Progress Display Module displays the progress of the current task through visualization. Processing Progress Estimated Remaining Time Prediction Module predicts the remaining time for processing task completion.

The user interface of LettuceP3D is intuitive and user-friendly, allowing users to easily import data, perform analysis tasks, and monitor the system's runtime status in real-time. The software takes about 2 seconds to analyse phenotypic data for individual plants, demonstrating high processing efficiency. The system is designed to provide efficient and accurate phenotypic parsing of lettuce, aiming to meet the needs of both research and agricultural fields.