**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 Computational Methods

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