

Jan 03, 2019 Version 4

Plant leaf tooth feature extraction V.4

DOI

dx.doi.org/10.17504/protocols.io.wrefd3e

Wang Hu¹, Li Chu¹, Tian Yan², Zhou Haoyu¹, Tian Di³

¹China Shipbuilding Industry Corporation; ²Huazhong University of Science and Technology; ³Wenhua College



Di Tian

OPEN  ACCESS



DOI: dx.doi.org/10.17504/protocols.io.wrefd3e

Protocol Citation: Wang Hu, Li Chu, Tian Yan, Zhou Haoyu, Tian Di 2019. Plant leaf tooth feature extraction. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.wrefd3e>

Manuscript citation:

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: January 03, 2019

Last Modified: January 03, 2019

Protocol Integer ID: 18950

Keywords: leaf tooth features, curvature, corner detection



Abstract

Previous studies extract features that are not strictly defined in botany; therefore, a uniform standard to compare the accuracies of various feature extraction methods cannot be used. For efficient and automatic retrieval of plant leaves from a leaf database, in this study, we propose an image-based description and measurement of leaf teeth by referring to the leaf structure classification system in botany. First, image preprocessing is carried out to obtain a binary map of plant leaves. Then, corner detection based on the curvature scale-space (CSS) algorithm is used to extract the inflection point from the edges; next, the leaf tooth apex is extracted by screening the convex points; then, according to the definition of the leaf structure, the characteristics of the leaf teeth are described and measured in terms of number of orders of teeth, tooth spacing, number of teeth, sinus shape, and tooth shape.

Preprocess

- 1 Preprocessing is employed to obtain the binary image of the leaf, which includes image graying, image segmentation, and image smoothing.
 - 1.We use the weighted average method to convert the image to grayscale.
 - 2.The maximum interclass variance method (abbreviated as OTSU) is used to binarize the image.
 - 3.Closing operations are performed to fill small holes and gaps in the image, and finally the image is smoothed and denoised.Here, the radius of the disc struct.

Detect the corners

- 2
 1. Following preprocessing of the image, the Canny operator is used to extract the leaf edge. (The setting of the two thresholds (H and L) is explained in [29].)
 2. After single pixelization of the leaf edges, the open-close situation of the contour curve is judged according to the pre-set threshold (Gap_size), then periodic convolution smoothing is used for the closed curve, and the open curve is convolved with the Gaussian function with width controlled by .
 3. The curvature of all points on the curve is then calculated, and the point with the maximum local curvature value is defined as the corner candidates.
 4. The threshold value C of the round corners removal is adaptively calculated according to the average curvature in the ROS region, and corners with curvature less than C are removed [30].
 5. The angle of the remaining corner candidates is calculated based on the recalculated dynamic support region, and corner candidates with angle smaller than the preset parameter T_angled are removed (removing the false corner).
 6. The position of the corner points are compared with the endpoints of the open curve and, if they are far apart, the endpoints are marked as candidate corners as well.
- 3.Extraction of convex and concave points of leaf tooth

Experiment

- 3 With the corner point as the center, set the radius to L, and calculate the number of target points and background points falling in the circle. If the number of target points is less than that of the background points, it is a convex point.

Number of orders of teeth

- 4
 1. Calculate the distance from the convex point to the line formed by adjacent concave points to indicate the length of the teeth;
 2. Normalize the teeth length, that is, divide the distance by the length of the longest tooth in the leaf (after removing abnormal points, which have values greater than twice

the mean of the teeth length);

3. After normalization, the number of teeth for which the tooth length is less than the half of the maximum length are counted. If they are greater than half of the total number of leaf teeth, the leaf is considered to have two orders of teeth, otherwise it is considered to have the order of one;

4. If step 3 shows that the leaf has two orders of teeth, it is necessary to distinguish the first-order teeth and the second-order teeth by comparing the teeth length. The length of the first-order leaf teeth is significantly larger than the length of the two adjacent leaf teeth and the second-order teeth do not have this characteristic, the teeth can be distinguished.

Tooth spacing

- 5
 1. Calculate the tooth spacing, that is, the distance between adjacent teeth;
 2. Compare the minimum intertooth distance d_{min} with the maximum intertooth distance d_{max} . If $d_{min} > (d_{max} * 0.6)$, the tooth spacing characters is regular; otherwise, it is irregular.

Tooth number

- 6 Calculate the total number of teeth of a leaf (i.e., the number of convex points using the abovementioned detection method).

Concave shape

- 7
 1. Calculate the curvature at the concave point of a leaf whose concave shape is rounded;
 2. Calculate the curvature at the concave point of a leaf that has an angular concave shape ;
 3. According to the statistical data in steps 1 and 2, determine the curvature threshold value which distinguishes the angular concave shape and the rounded concave shape.

Tooth shape

- 8
 1. Extract the coordinates of an arc between a concave point and a convex point on the profile curve of the leaf tooth;
 2. Fit the coordinates of the quadratic polynomial to obtain the corresponding polynomial;
 3. Examine the quadratic parameter of the quadratic polynomial; if the parameter is equal to zero, it is straight; If the parameter is nonzero, the subsequent steps are taken to continue processing.
 4. In order to further distinguish the tooth shape of CV, CC, FL, and RT, we use straight lines to connect the concavo-convex points, and count the number of black and white

pixels in the closed area surrounded by the contour of leaf and straight line, and then judge the tooth shape according to the distribution of black and white pixels. As shown in Fig 9, A is the convex point of the leaf teeth, E is the concave point, and is the contour of the leaf, C is the intersection point of line AE and the contour (usually there is only one intersection point at most), B is any point on the curve , and D is any point on the curve .

1) If point C does not exist, count respectively the number of pixels with values of zero (black) and one (white) in the area of the line AC and the arc , and denote them as and respectively; if , the leaf tooth shape is CV; otherwise, it is CC.

2) If point C exists, count respectively the number of pixels with values of zero (black) and one (white) in the area (P) of the line AC and the arc , and denote them respectively as and ; if , the area P is recorded as Black; otherwise, it is recorded as White. Then, count respectively the number of pixels with values of zero (black) and one (white) in the area (Q) of the line AC and the arc , and denote them respectively as and ; if , the area Q is recorded as Black; otherwise, it is recorded as White. If P is Black and Q is White, the tooth shape is RT. If P is White and Q is Black, the tooth shape is FL.

Experiment

9

To verify whether the proposed leaf structure feature description algorithm is scientific and effective, we implemented the algorithm using MATLAB 2017 (MathWorks, Natick, MA, USA) on a standard desktop PC (4.2 GHz CPU, 24 GB RAM). Processing of a single leaf took approximately 1.4 s. This could undoubtedly be improved through further optimization and/or using parallel computing.

Command

Experiment_1 (windows 10)

```
data=toothFeature_finished('D:\ExperimentData_1\leaf_1.mat')
```

Command

Experiment_2 (windows 10)

```
accuracy=Classification_SVM('D:\Experiment_2_data\Our_method_data')
accuracy=Classification_LDA('D:\Experiment_2_data\Our_method_data')
accuracy=Classification_SVM('D:\Experiment_2_data\Proposed_in[20]_data')
accuracy=Classification_LDA('D:\Experiment_2_data\Proposed_in[20]_data')
```

We have uploaded the code and data of this study, please reference to:

https://github.com/hollyprince/Leaf_tooth_feature