



**HAL**  
open science

# An Algorithm for the Interactive Calculating of Wheat Plant Surface Point Coordinates Based on Point Cloud Model

Lei Xi, Guang Zheng, Yanna Ren, Xinming Ma

► **To cite this version:**

Lei Xi, Guang Zheng, Yanna Ren, Xinming Ma. An Algorithm for the Interactive Calculating of Wheat Plant Surface Point Coordinates Based on Point Cloud Model. 9th International Conference on Computer and Computing Technologies in Agriculture (CCTA), Sep 2015, Beijing, China. pp.53-63, 10.1007/978-3-319-48357-3\_6. hal-01557846

**HAL Id: hal-01557846**

<https://inria.hal.science/hal-01557846v1>

Submitted on 6 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# An Algorithm for the Interactive Calculating of Wheat Plant Surface Point Coordinates Based on Point Cloud Model

Lei Xi, Guang Zheng, Yanna Ren and Xinming Ma\*

College of Information and Management Science, Agronomy College, Henan Agricultural University, Zhengzhou 450002, China

Employing the 3-point cloud model of the wheat plant as the research object, through the establishment of mapping algorithm from the screen pixel to the coordinate point of the wheat plant body surface, this paper realized the calculating for the point coordinates on the wheat plant surface utilizing the mouse operation. This algorithm was constituted of solving a user view equation of given screen pixels, screening the point cloud near the line of sight, extracting the efficient point sets near the line of sight, surface fitting efficient point set and solving the intersection of the line of sight and the fitting surface five processing flow on the basis of scattered point cloud data preprocessing and rendering. Adopting the mark function of the FastSCAN 3-d digital scanner accessory instrumental software, the comparison validation of the wheat plant body surface coordinates points' selection was conducted. The validation results show that the error less than 2.1 mm and had good precision. Considering the influence of the users' perspective to the calculation of the plant surface points coordinates, the comparison validation was conducted with the  $10^\circ$ ,  $20^\circ$  and  $0^\circ$  perspective respectively. The results show that the error is less than 1.8 mm and had good accuracy. The mapping algorithm between the screen pixel and the object surface coordinate point established in the study could map the action of the mouse on the screen window to the operation of the object surface. Moreover, it also provided the technical reference for the establishment of the geometry measurement of the interactive plants based on point cloud data.

**Keywords:** wheat plant, point cloud model, point coordinates, interactive

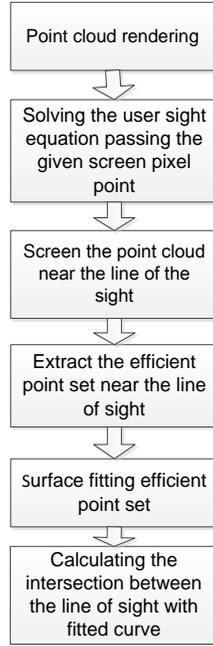
## 1. INTRODUCTION

Virtual plant was always the research hot spot of the interdisciplines such as the digital agriculture and virtual reality. The 3-d geometry measurement of plants was a basic and preliminary work of the virtual-plants modeling. Whether the building of the geometric modeling and the morphogenesis model of plant organs, or the building of the quantitative topology model of the plant, the accurate, complete geometric information was necessary for support. Due to the complex characteristics of the internal morphological structure of the plants, the measurements to the plant geometrical morphology became a time-consuming complex work. With the development of the 3-d information acquisition technology, rapid access to the high-precision point cloud data of the plants became possible. It also provided a new method for obtaining the geometrical morphology data of the plants. The methods of measuring the geometrical morphology data of the plants could be divided into two categories generally: the geometrical morphology measurement to the surface model formed by the reconstruction of point cloud<sup>[1-4]</sup> and the geometrical morphology measurement using the point cloud data directly<sup>[5-6]</sup>. Due to the reconstruction of topological structure between body surface points and establish object surface model in the surface model formed by the reconstruction of point cloud, the design of the measurement algorithm was relatively simple. Moreover, the existing CAD software could be used in the measurement method. However, the shortcomings such as time-consuming and big error (produce redundant triangles) could easily arise during the reconstructing the point cloud data of the whole plant because of the complexity of the plant structure. Therefore, the current study of plant point clouds reconstruction mostly concentrated in the point clouds reconstruction of plant organs<sup>[7]</sup>. The research of the measurement to the plant geometrical morphology using the point cloud data directly was concentrated in certain fixed shape feature extraction<sup>[8]</sup> utilizing the point cloud feature extraction technology<sup>[9-11]</sup>. Although the measurement to the plant geometrical morphology using point cloud feature extraction technology possessed the advantages of the high degree of automation, the interactive measurement to the geometrical morphology of plants such as leaf edge curve, veins curve and space angle with the help of the users' selection to the surface coordinates points had the characteristics of high maneuverability and flexible. Therefore, using the three-dimensional point cloud data of the wheat plant as the object of study, the research in the paper realized the calculation of the

plant surface point coordinates using the mouse operation and provided the technical reference for the interactive plant geometrical morphology measurement based on point cloud data.

## 2. MATERIAL AND METHOD

The calculating of the interactive object surface point coordinates aimed to build the mapping between the screen pixels selected by the user mouse operation to the object surface coordinate points. Using mathematical language, it could be described as the calculating the intersection point of the first time intersection between the object's surface and the sight line of users along the line of sight direction. The intersection point was the mapped coordinate point of the screen pixel on the object body surface. Due to the lack of the concept of the “surface” of the object represented by the point cloud data, the intersecting probability of the user’s line of sight with any point in the point cloud data all



**Fig.1.** The algorithm flow of the mapping from the screen pixel point to the object surface coordinate point

was 0 in theory. Therefore, in this study, the intersection between the line of sight and the object surface was defined as the intersection between the line of sight and the local surface fitted by the point cloud data near the line of sight. Thus, the mapping algorithm from the screen pixel point to the object surface coordinates point on the basis of the preprocessing and rendering to the point cloud data in this paper was constituted by the solving the user sight equation passing the given screen pixel point, screening the point cloud near the line of sight, extracting the efficient point set near the line of sight, surface fitting efficient point set and calculating the intersection between the line of sight and the fitted curve five processing steps and shown in Figure 1.

### 2.1 Solving the user sight equation passing the given screen pixel point

Accounting the coordinate system (object coordinate system) describing the point cloud data was the  $O$  system. The coordinate system with the increased depth values describing the window coordinates was named the  $O'$  system. The transition from the  $O$  system to the  $O'$  system could be expressed by the transformational matrix generated by the point cloud rendering, as shown in the formula (1).

$$M = \prod_{i=n}^1 M_i \quad (1)$$

In the formula, the  $M$  was the transformational matrix from the  $O$  system to the  $O'$  system. The  $M_i$  was the transformational matrix generated by the point cloud rendering following the sequence. Suppose that the window coordinate of the depth value of the pixel point passed by the line of sight was the  $P_w'(w_x, w_y, w_z)$ , the corresponding point of the pixel point  $P_w$  in the coordinate system  $O$  could be expressed by the formula (2).

$$P_w = M \cdot P_w' \quad (2)$$

The  $w_x, w_y$  were expressed by the window coordinate of the mouse during the approach implementation. The  $w_z$  could be determined by the projection matrix in the point cloud data rendering and commonly was 0. Suppose that the location of the user's view was  $P_e$ , the mathematical equation of the line of the sight could be expressed by the formula (3).

$$P(t) = P_e + v_e \cdot t \quad (3)$$

The  $v_e$  was the direction vector of the line of sight and given by the formula (4). The  $P_e$  was set during rendering the point cloud data.

$$v_e = \frac{P_e P_w}{|P_e P_w|} \quad (4)$$

## 2.2 Screening the point cloud near the line of sight

As shown in Figure 2, all the points which the distance to the sight less than the threshold  $r$  were regarded as the points near the line of sight and were defined as the point cloud near the line of sight. In the Figure 2b, the full line was used to express the line of sight of the users. The  $S_1, S_2, S_3, S_4$  was used to the point cloud near the line of sight. Moreover, it also was used to stand for the 4 local surface of the object.

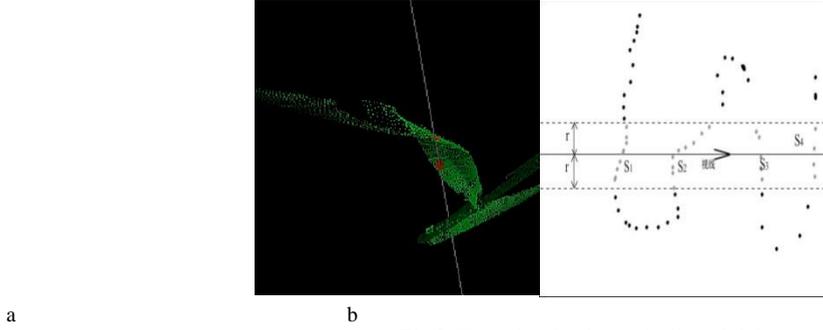


Fig.2. The point cloud near the line of sight

Based on the data equation of the line of sight, for any point in the point cloud, the distance to the line of sight could be given by the formula (5). Any point in the point cloud could be conducted the screening and extracting following the formula (5).

$$d = \sqrt{(P_w P_i \cdot v_e)^2 + P_w P_i^2} \quad (5)$$

## 2.3 Extracting the efficient point set near the line of sight

In general, the point cloud near the line of sight often represented one or more local surfaces. According to the definition of the mapping from the screen pixels to the surface coordinates, only the local surface closest to the viewpoint was used to calculate the surface of the mapping point. That also meant that only the point set which represented the surface was effective. For example, in figure 2 b, the point cloud near the line of sight expressed four local surfaces. However, only the point set  $S_1$  was valid.

In the point cloud data near the line of sight, using the clustering based on density and the WaveCluster method [12-15] both could realize the corresponding requirements of identifying effective point set. However, the clustering based on density generally required the users to set the corresponding parameters. Moreover, both the two clustering methods had the shortage that the calculation speed was slow. Considering the requirement to the procedure response speed for the "interactive" design, this research conducted the dimension reduction process to the point cloud data near the line of sight firstly. After that, a linear algorithm was designed to conduct the method for extracting the effective point set.

In the design of the dimension reduction process for the point cloud data near the line of sight, the point cloud located in the different local surface could be found. The projection of the distance from these point cloud to the point of view in the line of sight direction would have bigger difference, as shown in Figure 3. Hence,  $J$  was set as the result for the dimension reduction process to the point cloud data near the line of sight. It could be defined as the formula (6).

$$J_i = |P_e P_i \cdot v_e| \quad (6)$$

In the formula, the  $P_i$  was the point  $i$  in the point cloud data near the line of sight.

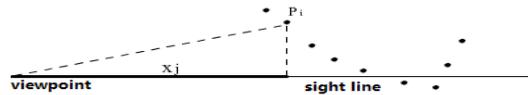


Fig.3 The projection of the distance from the point to the point of view in the line of sight direction

Analysis the point cloud near the line of sight,  $J_i$  would appear the abnormal increasing from a local surface to another local surface along the line of sight direction usually because the threshold value  $r$  was small relatively. Therefore, the efficient point set could be detected through the analysis of the increased abnormal values of adjacent points  $J_i$ . The specific algorithm was designed as follows:

The linear table  $L_p$  was constructed and saved in the point cloud near the line of sight. After that, it was sort following the  $J_i$  of each point. The increment of the  $J_i$  of the two neighboring points was calculated and saved in the new linear table  $L_j$ . The process could be expressed as the  $L_j[i] = L_p[i + 1] \cdot J - L_p[i] \cdot J$ . After that, the abnormal value in the  $L_j$  was detected to judge whether the point could reach the next part surface.

Considering the interactive response speed, the algorithm adopted the simple abnormal value detection method and determined whether the element in the  $L_j$  was the abnormal value by detecting whether the deviation of each element in the  $L_j$  was greater than a certain ratio standard deviation.

The two multiple parameters  $n_1, n_2$  were set to be the threshold value for the abnormal value detection in the algorithm. The  $n_1$  was set as the threshold value for the abnormal value-to-be and expressed that the next step detection should to be continue to conduct the abnormal value judgment. The  $n_2$  was the threshold value for the abnormal value and expressed that the abnormal value was detected currently. The algorithm implementation process was shown as follows:

- (1). A set  $S_p$  was established as a valid point set and initialized to the first three elements of the  $L_p$ .
- (2). The linear table  $I_j$  was established to hold the increment of the  $J_i$  in computer and initialized to the first two elements of the  $L_j$ . The loop variable was defined  $i$  as 4.
- (3). At the time of the  $i > n$ , the algorithm was stopped.  $n$  was the length of the  $L_p$ .
- (4). Calculating the mean  $I_{avg}$  and the standard deviation  $I_{std}$  of the elements in the  $I_j$ .
- (5). Executing  $tmp = L_j[i - 1] - L_j[i - 2]$ , if  $tmp - I_{avg} > n_1 \cdot I_{std}$ , go to (8).
- (6). Putting the  $L_p[i]$  in the set  $S_p$ , putting  $L_x[i - 1]$  in the set  $I_x$ ,  $i = i + 1$ , go to (3).
- (7). If  $tmp - I_{avg} > n_2 \cdot I_{std}$ , that meant the next surface had been reached, exiting program.
- (8). While  $i + 1 > n$ , the algorithm was stopped.
- (9). Executing  $tmp = L_j[i] - L_j[i - 2]$ , if  $tmp - I_{avg} > n_2 \cdot I_{std}$ , exiting program.
- (10). Putting  $L_p[i - 1], L_p[i]$  into the set  $S_p$ , putting the  $L_x[i - 1], L_x[i]$  into  $I_x$ , executing  $i = i + 2$ , go to (3).

The set  $S_p$  was the efficient point set near the line of sight at the end of the algorithm. The evaluation of the  $n_1, n_2$  were 4.5 commonly. For the peculiar point cloud data, some small adjustment also could be done. In the Figure 4, the red point set expressed the result of the efficient point set extraction near the line of sight.

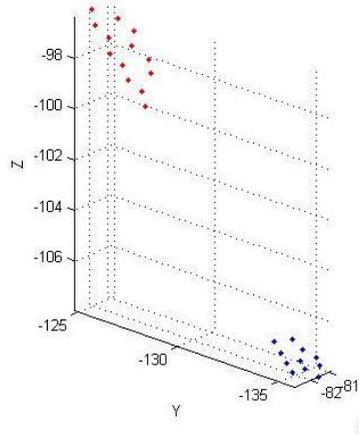


Fig.4 The result of the efficient point set extraction near the line of sight

#### 2.4 Surface fitting efficient point set

According to the interpretation for the intersection of the line of sight and the point cloud, the intersection of the line of sight and the object was actually the intersection of the line of sight and the curved surface expressed by the efficient point set near the line of sight.

The setting of the threshold  $r$  was generally smaller during the extracting the point cloud near to the line of sight. At the same time, according to the demand of the rapid response in the user interaction, the least square plain fitting to the efficient point set was adopted to the express the local surface of object in the method. The method was as follows.

The equations of plane was set as  $z = -Ax - By + D$ . Substituted all the points in the efficient point set into the equation, the least square error formula could be obtained and shown as the formula (7).

$$E = \sum (-Ax_i - By_i + D - z_i) \quad (7)$$

Due to the  $\nabla E = 0$ , the least square solution of the  $A, B, D$  could be obtained and shown as the formula (8).

$$\begin{pmatrix} \sum x_i^2 & \sum x_i y_i & -\sum x_i \\ \sum x_i y_i & \sum y_i^2 & -\sum y_i \\ -\sum x_i & -\sum y_i & n \end{pmatrix} \begin{pmatrix} A \\ B \\ D \end{pmatrix} = \begin{pmatrix} -\sum x_i z_i \\ -\sum y_i z_i \\ \sum z_i \end{pmatrix} \quad (8)$$

In the formula,  $n$  expressed the number of the point sets.  $x_i, y_i, z_i$  was the coordinate value of the point  $i$ . The least square estimation of the local surface could be obtained through the solving of the equation set. For the convenience of the calculating of the intersection of the line of sight and the fitting plane, the vector  $V$  was defined as  $v = (A, B, 1)$ , the vector  $P$  was defined as  $P = (x, y, z)$ . Thus, the plane equation could be expressed as the vector expression and shown in the formula (9).

$$v \cdot P = D \quad (9)$$

#### 2.5 Calculating the intersection between the line of sight and the object surface

Contact the formula (9) and the formula (3), the parameter of the point of intersection between the line of sight of users and the fitting local surface of the object could be determined and shown as the formula (10). After that, as shown in figure 5, the

intersection point could then be found.

$$t = \frac{D - v \cdot P_w}{v \cdot v_e} \quad (10)$$

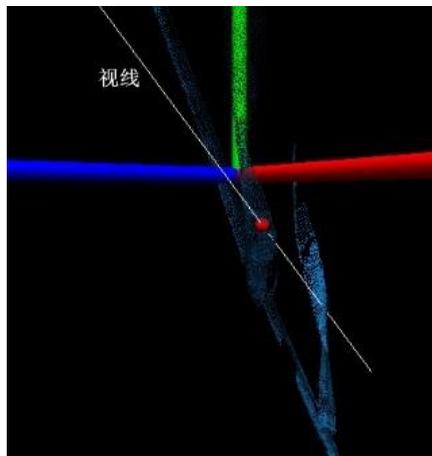


Fig.5 The intersection point of the line of sight and the object

### 3. RESULT AND ANALYSIS

Using the point cloud data of the wheat plant collected in Henan agricultural university science park from 2012 to 2013 as the experimental material, adopting the C++ language, with the help of the software library such as OpenGL, PCL<sup>[16]</sup> and so on, the interactive wheat plant surface pastry coordinates calculating algorithm was realized at the Ubuntu12.04 system environment in the research and shown in Figure 6. As shown in table 1, using the mark function of the supporting tools software of the FastSCAN 3D digital scanner, the mapping from the screen pixels to the object surface coordinates was verified. The results showed that the error of the proposed mapping method in this paper was less than 2.1 mm compared with the FastSCAN and had good accuracy. Because there was no direct intersection between the line of sight and the point cloud in this study, the error of the interactive selection of the points was mainly produced by different user perspective. As shown in table 2 and table 3, this study conducted the comparison validation with the 10°, 20° and the 0° viewing angle respectively. The results showed that the error of the method was less than 1.8 mm in the normal range of angle and had good accuracy.

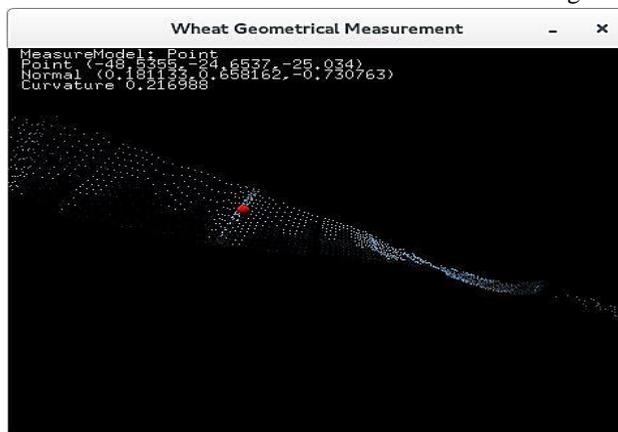


Fig.6 Interactive measurement of point

Table I Experimental verification of screen pixel coordinates point-to-surface point map

Serial number	Fastscan			Mapping method in this paper			Error(mm)
	x(mm)	y(mm)	z(mm)	x(mm)	y(mm)	z(mm)	
1	-93.03	41.06	208.84	-92.9382	42.48706	209.3463	1.51698
2	-72.27	27.55	168.54	-72.2284	27.91212	169.2688	0.814823

3	-52.98	18.29	136.58	-52.4159	19.09847	137.0976	1.113435
4	-49.19	12.52	116.09	-48.1335	13.99432	116.9974	2.028112
5	-37.67	7.37	101.52	-37.2543	7.99218	101.8187	0.805715
6	-24.03	5.63	81.85	-23.5856	6.558495	82.29905	1.123049
7	-33.52	3.68	83.44	-32.4752	4.144433	84.60859	1.634903
8	-13.76	1.7	45.37	-12.7095	1.78187	45.72393	1.111563
9	-0.53	1.58	8.44	0.342988	2.501845	9.016809	1.394495
10	-16.4	-1.71	44.72	-16.0489	-0.42301	45.43838	1.515156

**Table II** The verification of interactive point selection with angle of slight is 0 degrees and 20 degrees

Serial number	0°			10°			Error(mm)
	x(mm)	y(mm)	z(mm)	x(mm)	y(mm)	z(mm)	
1	-92.938	42.487	209.346	-92.083	42.793	209.082	0.948
2	-72.228	27.912	169.269	-72.062	28.466	169.009	0.634
3	-52.416	19.099	137.098	-52.542	18.906	137.786	0.725
4	-48.134	13.994	116.997	-47.982	13.460	116.648	0.656
5	-37.254	7.992	101.819	-36.953	7.862	101.332	0.587
6	-23.586	6.558	82.299	-23.459	6.893	83.078	0.857
7	-32.475	4.144	84.600	-31.813	4.848	83.938	1.176
8	-12.7095	1.782	45.724	-12.049	2.732	44.968	1.382
9	0.343	2.5018	9.0168	-0.240	2.286	8.540	0.784
10	-16.049	-0.423	45.438	-15.319	-0.668	45.156	0.820

**Table III** The verification of interactive point selection with angle of slight is 0 degrees and 20 degrees

Serial number	0°			20°			Error(mm)
	x(mm)	y(mm)	z(mm)	x(mm)	y(mm)	z(mm)	
1	-92.9382	42.48706	209.3463	-93.1475	42.12267	210.6366	1.356964
2	-72.2284	27.91212	169.2688	-72.3269	28.22446	169.8735	0.687665
3	-52.4159	19.09847	137.0976	-51.0175	20.17961	136.6038	1.835253
4	-48.1335	13.99432	116.9974	-48.4386	13.57301	117.768	0.929734
5	-37.2543	7.99218	101.8187	-36.2038	8.447684	100.6524	1.634433
6	-23.5856	6.558495	82.29905	-23.504	6.636894	82.53445	0.261196
7	-32.4752	4.144433	84.60859	-32.5588	3.288491	83.80317	1.178269
8	-12.7095	1.78187	45.72393	-13.1347	2.014106	44.55533	1.26504
9	0.342988	2.501845	9.016809	0.962779	2.533851	9.861952	1.048539
10	-16.0489	-0.42301	45.43838	-15.8339	-0.59462	45.10651	0.43108

## 4. CONCLUSION

Using the scattered three-dimensional point cloud data of the wheat plant as the research object, through the procedure including "Solving the user sight equation passing the given screen pixel point--Screening the point cloud near the line of sight-- Extracting efficient point set in the point cloud near the line of sight-- Surface fitting efficient point set--Calculating the intersection between the line of sight and the object surface", the mapping from the screen pixel point to the object surface coordinate point was formed. It could map the window coordinate of the user's mouse to the object surface which expressed by the point cloud and realize the calculation for the wheat plant surface point coordinate. This method had certain universality and could to provide technical reference for the construction of the interactive plant to geometry measurement based on point cloud data.

**Acknowledgements:** This study has been funded by China National 863 Plans Projects (Contract Number: 2008AA10Z220) and key scientific and technological project of Henan Province (Contract Number: 082102140004).

## References and Notes

1. Boissionnat J D. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4), 266-286 (1984).
2. Dey T K, Goswami S. Tight cocone: a watertight surface reconstructor. *Journal of Computing and Information Science in Engineering*, 3(4), 302-307 (2003).
3. Dey T K, Goswami S. Provable surface reconstruction from noisy samples. *Computational Geometry*, 25(1), 124-141 (2005).
4. Gong Yinxi, He Cheng, Feng Zhongke, Li Wenzhao, Yan Fei. Amended Delaunay Algorithm for Single Tree Factor Extraction Using 3-D Crown Modeling. *Transactions of the Chinese Society for Agricultural Machinery China*, 44(2), 192-199 (2013).
5. He Cheng, Feng Zhongke, Yuan Jinjun, Wang Jia, DongZhihai, GongYinxi. Three-dimensional volume measurement of trees based on digital elevation model. *Transactions of the Chinese Society of Agricultural Engineering, China*, 28(8), 195-199 (2012).
6. Paulus S, Dupuis J, Mahlein AK, Kuhlmann H. Surface feature based classification of plant organs from 3D laserscanned point clouds for plant phenotyping. *BMC Bioinformatics*. 14, 238(2013).
7. LU Shenglian, GUO Xinyu, LI Changfeng. Research on Techniques for Accurate Modeling and Rendering 3D Plant Leaf. *Journal of Image and Graphics, China*, 14(4), 731~737. (2009).

8. Zhao Yuandi, Wen Weiliang, GuoXinyu, Xiao Boxiang, Lu Shenglian, Sun Zihui. Midvein Extraction for 3-D Corn Leaf Model Based on Parameterization. Transactions of the Chinese Society of Agricultural Machinery, China, 42(4), 183-187(2012).
9. Satio T, Takahashi .Comperhensible Rendering of 3-D Shapes. Computer Graphics, 24(4), 197-206(1990).
10. Zou Dong and Pang Mingyong. Algorithm for Extracting Sharp Features from Point Cloud Models. Transactions of the Chinese Society for Agricultural Machinery, China, 42(11), 222-227(2011).
11. Liu Qian, GengGuohua, Zhou Mingquan, Zhao Lulu, Li Jijunnan. Algorithm for feature line extraction based on 3D point cloud model. Application Research of Computer, China, 30(3), 933-937(2013).
12. Birant D, Kut A. ST-DBSCAN. An algorithm for clustering spatial-temp oral data. Data&Knowledge Engineering, 60(1), 208-221(2007).
13. Elsner G. A new sequential cluster algorithm for optical lens design. Journal of Optimization Theory and Applications, 59(2), 165-172(1988)
14. Alexander Hinneburg and Hans-Henning Gabriel. DENCLUE 2.0: Fast Clustering Based on Kernel Density Estimation. Lecture Notes in Computer Science, 4723, 70-80(2007).
15. Zhou Weizhong, FengXinhai and SunGuoji. Classification Based on Feature Extraction from Cluster of Wavelet Coefficients. Journal of Computer Research and Development, 1-12, 982-988 (2001).
16. Point Cloud Library. <http://www.pclcn.org/study/shownews.php?lang=cn&id=29>.( 2014)







