

Automatic sweet pepper detection based on point cloud images using subtractive clustering

Xiaokang Zhao¹, Hao Li¹, Qibing Zhu^{1*}, Min Huang¹, Ya Guo¹, Jianwei Qin²

(1. Key Laboratory of Advanced Process Control for Light Industry of Ministry of Education, Jiangnan University, Wuxi, 214122, China;

2. USDA/ARS Environmental Microbial and Food Safety Laboratory, Beltsville Agricultural Research Center, Bldg., 303, BARC-East, 10300 Baltimore Ave., MD 20705-2350, USA)

Abstract: Automatic identification and detection of fruit on trees by machine vision is the basis of developing automatic harvesting robots in agriculture. The occlusion of branches, leaves and other fruits in canopy images will affect the accuracy of fruit detection. To provide a scientific and reliable technical guidance for fruit harvesting robots, a method using point cloud images was proposed in this study to detect red fruits to overcome the impact of occlusion on detection. Firstly, the fruit regions were segmented from a tree's point cloud by applying the color threshold of red and green. Then, the noise in fruit point clouds was removed with sparse outlier removal. Finally, the point cloud of each fruit was detected and counted based on the subtractive clustering algorithm. For the sweet pepper dataset, the true positive rate (TPR) is 90.69% and the false positive rate (FPR) is 6.97% for all fruits that are at least partially visible in the scene.

Keywords: sweet pepper detection, point cloud, subtractive clustering, computer vision

DOI: 10.25165/ijabe.20201303.5460

Citation: Zhao X K, Li H, Zhu Q B, Huang M, Guo Y, Qin J W. Automatic sweet pepper detection based on point cloud images using subtractive clustering. *Int J Agric & Biol Eng*, 2020; 13(3): 154–160.

1 Introduction

Fruit harvesting is an important part of the fruit production process, with strong seasonal and labor-intensive characteristics. The quality of fruit harvesting is directly related to the storage, processing, sale of fruit, and ultimately affects the market price and economic benefits. At present, fruit harvesting is still dominated by high-cost, high-intensity and high-risk manual harvesting, in spite of the advances in agricultural robotics^[1]. Since the concept of automatic harvesting machine was put forward in the 1960s, the application of agricultural robots in automatic weeding, field reconnaissance and harvesting has achieved some results. At the same time, object identification, task planning algorithms, digitization and optimization of sensors are some challenges faced by digital agriculture^[2].

With the development of machine vision technology, fruit harvesting robots based on machine vision have been widely studied. The success rate of harvesting robots based on machine vision depends on the accurate detection of fruit from canopy images obtained by their vision detection system. The complexity

and diversity of the natural environment, such as light changes, canopy structure, fruit color, occlusion, bring difficulties to the accurate detection of fruits^[3]. To improve the accuracy of fruit detection, various image sensors and image analysis methods have been used for fruit detection gradually.

Black/White (B/W) camera was used as an image sensor for fruit detection in earlier studies. Edan et al.^[4] applied two B/W cameras to acquire melon images under controlling illumination, the detection accuracy of 82.0%-88.0% was achieved by using geometric and texture features extracted from B/W images. Color is one of the most prominent features to distinguish fruit from the background, B/W camera cannot get color information of an image, and hence, it is difficult to ensure the accuracy of fruit detection in natural scenes. Lu et al.^[5] developed a machine vision system consisting of a color CCD camera and computer to segment fruits and branches and achieved high precision.

Currently, the RGB camera is the most widely used image sensor in the fruit harvesting robot. The RGB camera can simultaneously obtain images at three channels (R/G/B), hence, the color feature, geometry and texture features can extract from RGB images. Tabb et al.^[6] proposed a background modeling method to detect apples in RGB images. Kurtulmus et al.^[7] used color cameras to obtain peach datasets, which were detected by statistical classifiers and neural networks. Khoshroo et al.^[8] developed an algorithm to distinguish red tomatoes using image processing techniques based on color information. Kuang et al.^[9] proposed a novel approach for multi-class fruit detection using effective image region selection. Histogram of oriented gradient features, texture features, and color features were utilized to improve the detection accuracy. Sa et al.^[10] applied deep convolutional neural networks to detect fruits to build an accurate, efficient, and stable fruit detection system. However, the training of the model requires a lot of pictures, and it also takes much time to label these images. In addition, the information provided by the RGB image is limited,

Received data: 2019-10-14 **Accepted data:** 2020-03-18

Biographies: **Xiaokang Zhao**, Master, research interest: nondestructive detecting technologies of agricultural products, Email: zhaoxk0211@163.com; **Hao Li**, Master, research interests: nondestructive detecting technologies of agricultural products, Email: 422079638@qq.com; **Min Huang**, PhD, Professor, research interests: sensing technologies for food quality, Email: huangmzqb@163.com; **Ya Guo**, PhD, Professor, research interests: agricultural information acquisition and utilization, Email: guoya68@163.com; **Jianwei Qin**, PhD, Agricultural Engineer, research interests: sensing technologies for food safety, Email: jianwei.qin@ars.usda.gov.

* **Corresponding author:** **Qibing Zhu**, PhD, Professor, research interests: agricultural information acquisition and utilization. School of Internet of Things, Jiangnan University, 1800 Lihu Avenue, Wuxi, Jiangsu Province, 214122, China. Tel: +86-510-85910635, Email: zhuqib@163.com.

color cameras usually cannot achieve better results when the color of the fruit is close to the background.

As the demand for fruit detection continues to increase, the fruit detection method based on RGB image is difficult to meet the requirements. Many scholars have integrated the results of multiple sensors to improve fruit detection. Shamsiri et al.^[11] reviewed the configurable, modular prototype robot system provided by the simulated workspace in the virtual environment, through simple testing and debugging of control algorithms, it can adapt to various field conditions to accelerate the commercialization of real robots. Wang et al.^[12] used a stereovision camera to locate the target in global coordinates for repeated counting of multiple images. Xiang et al.^[13] proposed a clustering tomato recognition method based on depth images, and the images were obtained by a stereo matching technology. However, due to a large amount of computation of stereovision technology, it is often unsuitable for real-time systems. Maturity grading is important for the quality of fruits, Wei et al.^[14] built a multispectral indexes system to identify the maturity with the hyperspectral technique. Because of the influence of light conditions, the sensor is unsuitable for fruit detection in outdoor environments. Underwood et al.^[15] developed a mobile terrestrial scanning system for almond orchards that can efficiently estimate yield and fruit distribution for individual trees, and the visual system was based on lidar and camera sensors. Stein et al.^[16] presented a novel multi-sensor framework that can efficiently identify, localize every piece of fruit in a mango orchard. A novel lidar component automatically generated the image for each canopy and used a multiple viewpoint approach for occlusion problems. In order to detect immature green citrus, Gan et al.^[17] created a new color-thermal combined probability algorithm to fuse information from the color and thermal images to distinguish fruit from the background. However, sensors such as lidar and thermal imaging cameras are more expensive, it is usually uneconomical to apply these sensors to the field of fruit detection.

In recent years, consumer-grade depth cameras have been favored by more and more scholars. This kind of camera can simultaneously acquire color images and depth images of the target scene. Compared with color cameras, we can obtain the position information of the target, and then calculate the three-dimensional geometric features of the object, which are significantly helpful when applied to fruit detection. Gongal et al.^[18] used a depth camera in conjunction with an RGB camera to identify duplicate apples visible in images captured from two opposite sides of the tree canopy. The accuracy of this method was 87.0% in identifying duplicate apples. Nguyen et al.^[1] developed an algorithm for detecting apples on trees using an RGB-D camera. The Euclidean clustering algorithm was applied to the point cloud to extract clusters for each apple. Tao et al.^[19] proposed an apple recognition method based on point cloud data, which is compared with different 3D descriptors and other classical classifiers to obtain recognition results and horizontal comparison results. The results showed that the method had a better performance, but improvements should be made for accurately determining the condition of fruit occlusion. Qureshi et al.^[20] proposed two new mango tree canopy image automatic counting methods for accurately segmenting and detecting fruits in canopy images, one using texture-based dense segmentation and another using shape-based fruit detection. For images collected under the same conditions as the calibration image, the estimated number of fruits

was within 16% of the actual fruit number. However, the results were poor when the model was used to estimate the number of fruits with different canopy shapes and when using different imaging conditions.

Since there are usually occlusions and clustering between fruits in natural environments and the shape of many fruits is irregular, methods based on fruit shape detection in previous literature are not effective in some situations. In order to overcome these problems, a novel method was proposed to detect and count sweet peppers based on subtractive clustering, which stays robust despite the influence of fruit shape and occlusion.

The ultimate goal of this research is to achieve fruit recognition under natural conditions. Specifically, the proposed method is utilized to detect sweet pepper fruits in three-dimensional point cloud scenes.

2 Datasets and methods

2.1 Datasets

An open dataset, called sweet pepper dataset^[21], was used for evaluating the performance of the proposed method. An RGB-D camera (Intel Real Sense F200, Intel company, USA) mounted on the robotic arm was used to collect RGB-D data of sweet pepper. The data collection was conducted over 10 days within a protected farming system. After data collection, the reconstruction of a dense sweet pepper point cloud from multiple views is implemented using the Kinect fusion^[22], and the statistical outlier remover and voxel grid down sampler supported from Point cloud library (PCL)^[23] were used for data de-noising and filtering. Figure 1 shows the 3D models of the point cloud scene. The more detailed dataset information can be found in Reference [22].



Figure 1 3D models of the point cloud scene

2.2 Methods

The color and geometric information of the 3D sweet pepper point cloud were used. According to the algorithm, the fruit regions were segmented by applying the color threshold of R-G first. Then, the noise was removed by an outlier filtering algorithm. Finally, the point cloud of each sweet pepper was processed based on the subtractive clustering algorithm. The automatic method was compared to the visual manual counting, where all the images were counted by human inspectors.

2.2.1 Color Filtering

In order to achieve accurate recognition and localization of fruits, it is necessary to segment the fruit regions from the background first. The color feature is an important attribute to distinguish fruits from the background. The segmentation method- based color threshold was widely used. In this paper, the R-G color difference segmentation method was chosen to perform background separation on 3D point cloud. The fruits

segmentation method is as follows:

$$\begin{cases} R_s - G_s > \delta & (\text{fruit region}) \\ R_s - G_s \leq \delta & (\text{background region}) \end{cases} \quad (1)$$

where R_s and G_s are the red and green information of the point cloud s , respectively. And δ is the segmentation threshold. When $R_s - G_s$ is greater than δ , the point is recognized as fruit point cloud data, otherwise, it is recognized as background point cloud data. After segmentation, only the fruit point cloud data is retained, and the background point cloud data is removed, which can provide support for the subsequent three-dimensional recognition of fruits.

2.2.2 Sparse outlier removal

In the fruit region point cloud data after color filtering, there are often some outlier noise points that are far from the target object. Additionally, measurement errors lead to sparse outliers which corrupt the results even more. Therefore, the sparse outlier removal module is used to correct these irregularities^[24].

Firstly, for fruit point cloud data, the average distance d_l from each point $p_l(x_l, y_l, z_l)$ ($l=1,2,\dots,n$) to all its neighbors $p_m(x_m, y_m, z_m)$ ($m=1,2,\dots,k$) is calculated as:

$$d_l = \sum_{j=1}^k \frac{1}{k} \sqrt{(x_l - x_m)^2 + (y_l - y_m)^2 + (z_l - z_m)^2} \quad (2)$$

Then, the mean value μ and standard deviation σ of d_l are calculated as:

$$\begin{cases} \mu = \sum_{l=1}^n \frac{d_l}{n} \\ \sigma = \sqrt{\frac{1}{n} \sum_{l=1}^n (d_l - \mu)^2} \end{cases} \quad (3)$$

Finally, the points which fall outside are trimmed as:

$$\begin{cases} d_l \in (\mu - \alpha \cdot \sigma, \mu + \alpha \cdot \sigma) & (\text{remained points}) \\ d_l \in (-\infty, \mu - \alpha \cdot \sigma) \cup (\mu + \alpha \cdot \sigma, +\infty) & (\text{removed points}) \end{cases} \quad (4)$$

where, the value of α depends on the size of the analyzed neighborhood, and it is 0.6 in this paper.

2.2.3 Subtractive clustering

Since there is interference with occlusion and aggregation in the fruit point cloud, methods such as RANSAC algorithm based on shape fitting is not suitable, which easily incorrectly recognizes the occluded adjacent two fruits into one. For these point cloud groups in 3D space, the clustering algorithm can work well.

Data clustering divides the data set into several different groups, and the similarity within the groups is greater than the similarity between the groups. It is generally to know the number of clusters in advance based on such as K -means algorithm. However, the estimation of fruit quantities in advance in a natural scene is unrealistic. For this reason, a subtractive clustering algorithm^[25] is utilized in this research to detect every fruit by finding high-density regions in 3D space. The flow chart is shown in Figure 2.

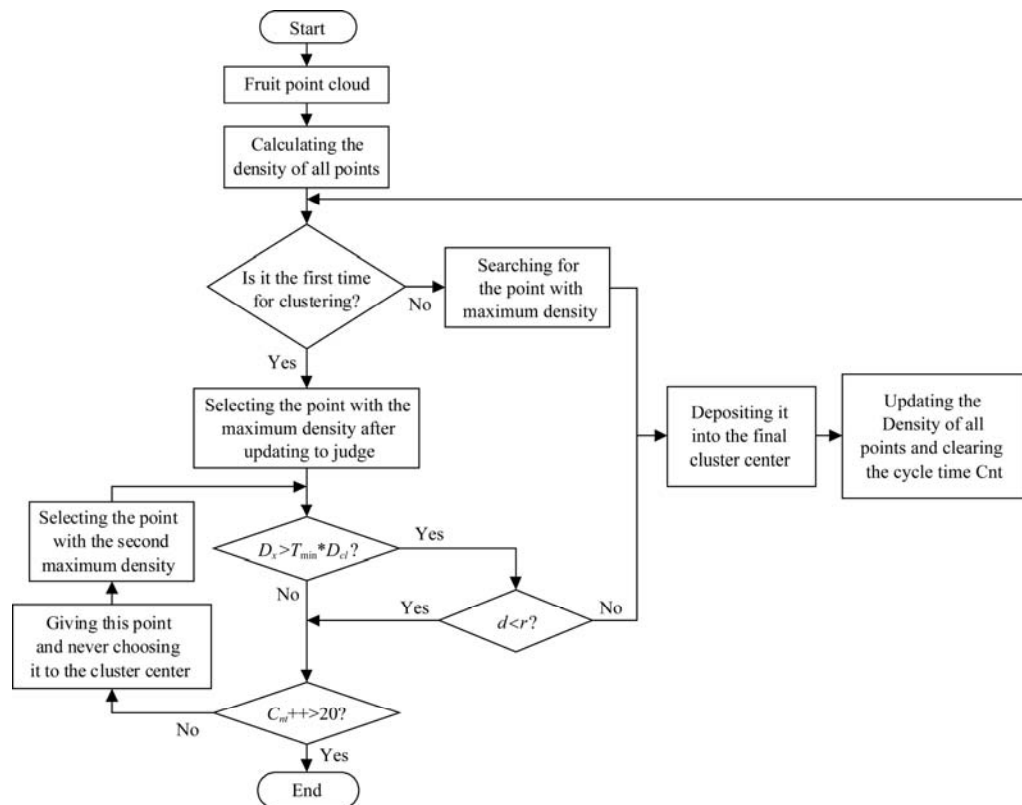


Figure 2 Flow chart of subtractive clustering

Fruit point clouds in the 3D space are roughly modeled by Gaussian function. The 3D coordinates of all point clouds are then considered as candidate clusters centers. Thus, each point p_i with coordinates (x_i, y_i, z_i) is potentially a cluster center whose density D_i is given by the following equation:

$$D_i = \sum_{j=1}^N \exp\left(-\frac{(x_i - x_j)^2}{\left(\frac{R_{ax}}{2}\right)^2} - \frac{(y_i - y_j)^2}{\left(\frac{R_{ay}}{2}\right)^2} - \frac{(z_i - z_j)^2}{\left(\frac{R_{az}}{2}\right)^2}\right) \quad (5)$$

where, N represents the number of 3D points within the neighborhood defined by the radius $R_c = (R_{ax}, R_{ay}, R_{az})$, in this paper, $R_{ax} = R_{ay} = R_{az} = 0.02$ m. The shape of the cluster can then be appropriately adjusted based on the selection of parameters R_{ax}, R_{ay}, R_{az} , which are related to the actual dimensions of 3D. Obviously, candidates p_i surrounded by more points contained in the defined neighborhood will show a high value of D_i . Points at a distance well above the radius defined by R_c hardly affects the value of D_i . Equation (5) is computed for all 3D points measured after sparse

outlier filtering. Where $p_{cl}(x_{cl}, y_{cl}, z_{cl})$ represent the point dominated by D_{cl} that represents the maximum density. This point is selected as the cluster center for the current iteration of the algorithm. The density of all points D_i is then corrected based on p_{cl} and D_{cl} . For this purpose, calculate the subtraction represented in Equation (6) for all points.

$$D_i = D_i - D_{cl} \exp\left(-\frac{(x_i - x_{cl})^2}{\left(\frac{R_{bx}}{2}\right)^2} - \frac{(y_i - y_{cl})^2}{\left(\frac{R_{by}}{2}\right)^2} - \frac{(z_i - z_{cl})^2}{\left(\frac{R_{bz}}{2}\right)^2}\right) \quad (6)$$

where, parameters $R_b=(R_{bx}, R_{by}, R_{bz})$ define the neighborhood. Due to these data points that cannot become the next cluster center, the density of data point close to the first cluster center will be reduced. To prevent clustering centers from being closer, parameters (R_{bx}, R_{by}, R_{bz}) are usually larger than (R_{ax}, R_{ay}, R_{az}) . Typically, let $R_b=1.5R_a$.

After the subtraction process, the density of the point near p_{cl} is correspondingly reduced according to the distance from the point to p_{cl} . After the correction of densities, a new cluster center $p_{cl,new}$ is generated, corresponding to the new density maximum $D_{cl,new}$, and the above process will iterate until Equation (7) is not satisfied.

$$\text{if } \frac{D_{cl,new}}{D_{cl}} > T_{min} \text{ and } d > r \text{ new cluster} \quad (7)$$

where, T_{min} is the experimentally tuned parameter which can be adjusted to define the termination condition based on the

relationship between the previous cluster density and the new one. When the density of the candidate cluster center is greater than T_{min} times of the density of the previous cluster center and the distance between the cluster center and the previous cluster center should be less than r , which prevents secondary recognition of the same fruit. Therefore, the size of r depends on the size of the fruit. For the test of sweet pepper, r is taken as 0.04 m. If the point based on this density satisfies the above two requirements, then the cluster center is accepted. If the density of the candidate cluster center is less than T_{min} times of the density of the previous cluster center, it is considered not to be the cluster center.

3 Results and discussion

3.1 Color filtering and sparse outlier removal results

The detection method was tested on the sweet pepper dataset, but since the amount of data in the point cloud scene in this dataset was too large, the sweet pepper was downsampling before it was detected by the algorithm. Sparse the point cloud scene and the sampling result are shown in Figure 3.

After obtaining the sparse data of the fruit trees, the color cloud threshold segmentation method of R-G was used to perform point cloud segmentation on the fruit and the background. Figure 4 shows the segmentation results of the fruit tree point cloud at different thresholds.

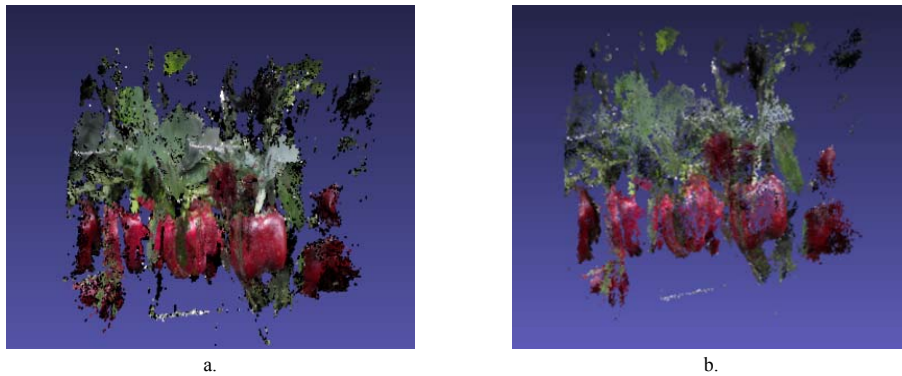


Figure 3 Point cloud original scene and sampling results

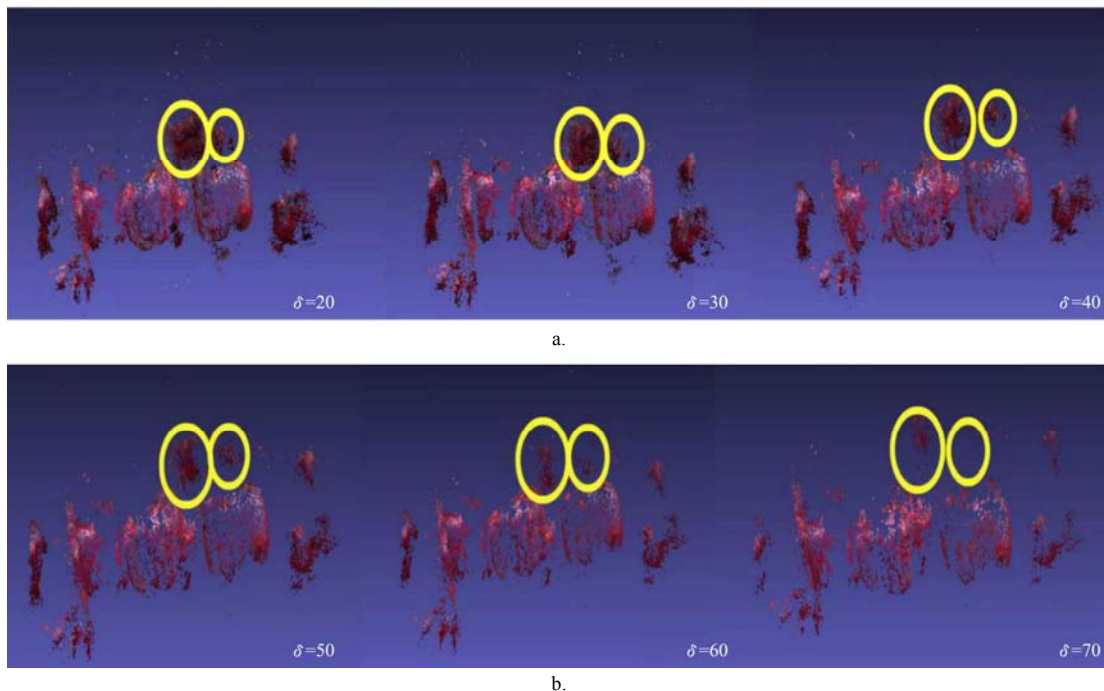


Figure 4 Segmentation results of fruit tree point clouds under different thresholds.

As can be seen from Figure 4, after the color threshold segmentation, the fruit and background were well separated to obtain point cloud data for the fruit region. From the point cloud part circled by the yellow mark, the segmentation effect was slightly different with the change of δ . When δ changed from 20 to 50, the segmentation effect was basically the same. When δ was greater than 50 and approached 60, 70, it can be observed that the number of original point clouds in the yellow circle had a significant decrease trend, indicating that the threshold had passed. The assembly caused the filtration of normal fruit point clouds. Therefore, in this experiment, for the segmentation threshold between the red bell pepper and the green background, it was set between 20 and 50, and when identifying other fruits, different adjustments were made according to experience.

In order to improve the efficiency and accuracy of clustering, the point cloud of the fruit region was first filtered and denoised. Figure 5 shows the point cloud processed by sparse outlier filtering. By filtering out the unstable noise points deviating from the main body, more accurate and effective fruit point cloud data were obtained.

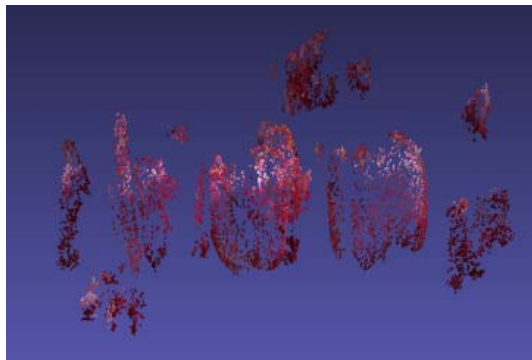


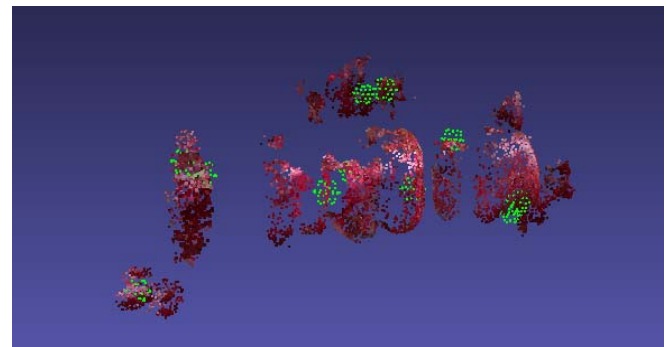
Figure 5 Filtering results after sparse outlier removal

3.2 Subtractive clustering results

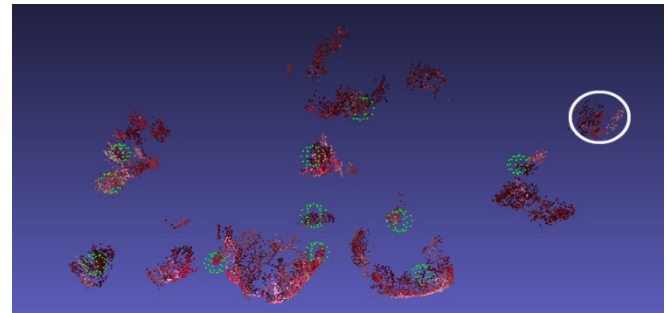
The subtractive clustering algorithm was verified on the processed data and the effectiveness of the algorithm accuracy was identified. The results of the qualitative analysis are shown in Figure 6, where the green point cloud label represents the recognition for sweet pepper. The recognition results based on the filtered point cloud data in the front view are shown in Figure 6a. The recognition results based on the filtered point cloud data in the top view are shown in Figure 6b. Figure 6c shows the effect picture of the green label in the original scene, where some labels are occluded in the point cloud and invisible.

It can be seen from Figure 6 that there are different degrees of occlusion in the sweet pepper, that is, the marked area, for the total of 12 sweet peppers, 8 of which had more than 50% occlusion area of the total area. Of the 12 fruits, 11 fruits were identified, and the cluster center was marked with a green label.

To quantify the performance of our fruit recognition method, true positive rate (TPR), false positive rate (FPR) and the false negative rate (FNR) were used to describe the correct detection accuracy of sweet peppers in the scene.



a.



b.



c.

Figure 6 Results of the qualitative analysis based on the proposed method

If the green clustering label was at the location of sweet pepper, then the detection was considered a true positive; otherwise, the detection was a false positive. Each detection for a fruit was treated as unique: if multiple detections locating the same fruit, one of them was regarded as the true positive and the others were considered false positives. A false negative represented a ground truth position that had no corresponding detection. The specific recognition results are shown in Table 1. The point plot of the recognition results is shown in Figure 7.

In Table 1, the recognition result of the fruit detection algorithm for the sweet pepper data set analyzed in Section 3.2 was summarized. For this data set, the TPR was 90.69% of all sweet pepper that were at least partially visible in the scene. Analysis of unapproved fruits showed that these were highly occluded fruits with fruits at the edges of the image and only a small portion of the surface visible, and the FPR was 6.97%.

Table 1 Recognition performance of the algorithm in sweet pepper datasets

Tree	Total fruits	Occlusion area <50%	Occlusion area >50%	True Positive Rate (TPR)	False Positive Rate (FPR)	False Negative Rate (FNR)
1	12	8	4	11	0	1
2	19	13	6	18	1	1
3	17	10	7	16	2	1
4	19	9	10	15	2	4
5	19	11	8	18	1	1
Total	86	51	35	78(90.69%)	6(6.97%)	8(9.31%)

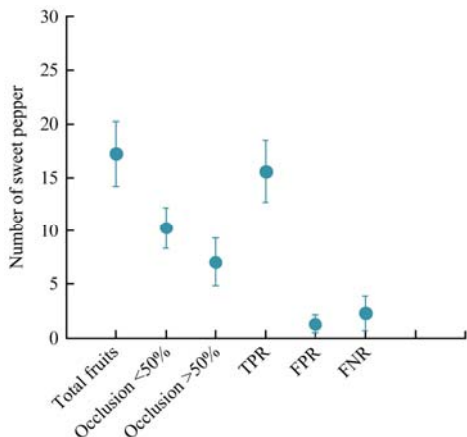


Figure 7 Point plot of the recognition results

The RANSAC^[26] algorithm was also used in the experiment to detect the dataset and compared it with the detection results of the proposed algorithm. RANSAC calculates the mathematical model parameters of the data based on sample data sets containing outliers and obtains an algorithm for valid sample data. The RANSAC algorithm is often used in computer vision. For example, solving the matching point problem of a pair of cameras and the calculation of the basic matrix simultaneously in the field of stereo vision. The detection result is shown in Figure 8 where the original point cloud image is the same as Figure a.

The recognition result can be seen in Figure 8, and 10 of the 12 sweet peppers were detected. The most important reason was that the two missing point clouds were fitted into one for the error detection of the adjacent occluded point cloud. In Table 2, the recognition results of the RANSAC algorithm were summarized.

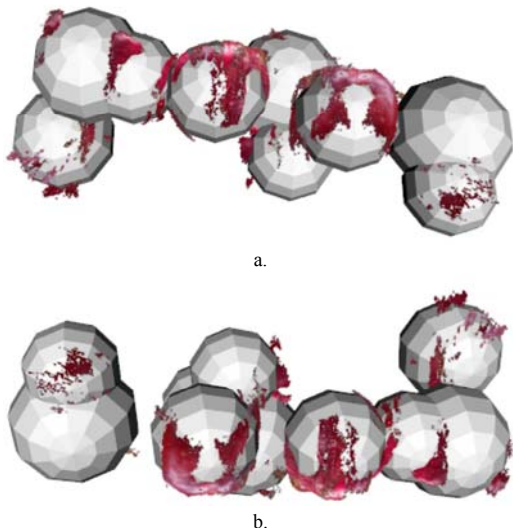


Figure 8 Qualitative results based on RANSAC algorithm from different perspectives

Table 2 Recognition results of the RANSAC algorithm in sweet pepper datasets

Tree	Total fruits	True Positive Rate (TPR)	False Positive Rate (FPR)	False Negative Rate (FNR)
1	12	10	0	2
2	19	16	0	3
3	17	15	1	2
4	19	16	0	3
5	19	17	2	2
Total	86	74 (86.05%)	3 (3.49%)	12 (13.95%)

Based on the RANSAC algorithm, a TPR of 86.05% was obtained, which was worse than the TPR of 90.69% based on the proposed algorithm. The FPR of 3.49% in RANSAC was superior to the one based on subtractive clustering, which was due to the repeated detection of the same fruit. But repeated recognition is also the right recognition of fruit, and has no negative impact on fruit picking. So, a slightly higher FPR had not too much effect on the final identification of fruit. In summary, the proposed algorithm was better than the RANSAC algorithm for a complex environment.

3.3 Discussion

Color filtering, sparse outlier removal and subtractive clustering were used to detect fruits in complex orchard environments. Finally, the detection results of 90.69% TPR and 6.97% FPR were obtained for the sweet pepper data set. After compared with the RANSAC algorithm, the superiority of the proposed algorithm can be verified.

For recognizing the fruit with severe occlusion, the radius parameters are set a little to adapt the size of the small fruit point cloud. Therefore, the results of the algorithm may show two cluster centers for large-sized fruits, which leads to an increase in FPR, so future research will focus on changing the algorithm to parameter adaptation. As for the FPR, the result of 6.97% is slightly higher, but the repeated detection of the fruit does not actually affect on the fruit picking, but only interferes with the yield estimation of the fruit. Therefore, the final test results are generally satisfactory.

Due to the use of the color threshold segmentation, the algorithm proposed is limited to detecting red and bicolored fruits. For detecting green apples, the filtering strategy should be improved. For example, the method of machine learning can classify color regions in a point cloud, and the method of spectral images can identify the target by extracting spectral features.

Because the algorithm is based on the sweet pepper point cloud dataset, it can be applied to larger fruits. However, there are many fruit clusters in the natural scene, especially the bunch of fruits like grapes. It is difficult to detect them for the proposed algorithm.

4 Conclusions

In this study, an algorithm was proposed and verified for sweet pepper detection which provides scientific and reliable technical guidance for sweet pepper harvesting robots. Firstly, the fruit regions were segmented from the original point cloud by applying the color threshold. Then, the noise in sweet pepper point clouds was removed with a sparse outlier removal method. Finally, the point cloud of each sweet pepper was detected and counted based on the subtractive clustering algorithm. For the sweet pepper dataset, the 90.69% TPR and the 6.97% FPR were obtained for all fruits that are at least partially visible in the scene. Undetected sweet peppers showed that these were highly occluded fruits which could be detected more accurately from another angle.

Acknowledgements

This research was financially supported by the National Natural Science Foundation of China (Grant No. 61772240, 61775086), the Fundamental Research Funds for the Central Universities (JUSRP51730A), as well as sponsored by the 111 Project (B12018).

The authors would like to thank Inkyu Sa for providing the sweet-pepper point cloud dataset.

[References]

- [1] Shamshiri R R, Hameed I A, Karkee M, Weltzien C. Robotic harvesting of fruiting vegetables: A simulation approach in V-REP, ROS and MATLAB. In: Kulshreshtha S N (Ed.). *Automation in Agriculture-Securing Food Supplies for Future Generations*, London: InTechOpen, 2018; pp.81–105.
- [2] Shamshiri R R, Weltzien C, Hameed I A, Yule I J, Grift T E, Balasundram S K, et al. Research and development in agricultural robotics: A perspective of digital farming. *Int J Agric & Biol Eng*, 2018; 11(4): 1–14.
- [3] Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K. Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, 2015; 116: 8–19.
- [4] Edan Y, Rogozin D, Flash T, Miles G E. Robotic melon harvesting. *IEEE Transactions on Robotics and Automation*, 2000; 16(6): 831–835.
- [5] Lu Q, Cai J R, Liu B, Deng L, Zhang Y J. Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine. *Int J Agric & Biol Eng*, 2014; 7(2): 115–121.
- [6] Tabb A L, Peterson D L, Parker J. Segmentation of Apple fruit from video via background modeling. In: *ASABE Annual International Meeting Presentation*. St. Joseph: ASABE, 2006; pp.60–63.
- [7] Kurtulmus F, Lee W S, Vardar A. Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. *Precision Agriculture*, 2014; 15: 57–79.
- [8] Khoshroo A, Arefi A, Khodaei J. Detection of red tomato on plants using image processing techniques. *Agricultural Communications*, 2014; 2(4): 9–15.
- [9] Kuang H L, Liu C R, Chan L L H, Yan H. Multi-class fruit detection based on image region selection and improved object proposals. *Neurocomputing*, 2018; 283: 241–255.
- [10] Sa I, Ge Z Y, Dayoub F, Upcroft B, Perez T, McCool C. Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 2016; 16(8): 1222.
- [11] Shamshiri R R, Hameed I A, Pitonakova L, Weltzien C, Balasundram S K, Yule I J. Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison. *Int J Agric & Biol Eng*, 2018; 11(4): 15–31.
- [12] Wang Q, Nuske S, Bergerman M, Singh S. Automated crop yield estimation for apple orchards. *Experimental Robotics*, 2013; 88: 745–758.
- [13] Xiang R, Jiang H, Ying Y. Recognition of clustered tomatoes based on binocular stereo vision. *Computers and Electronics in Agriculture*, 2014; 106: 75–90.
- [14] Wei X, He J C, Ye D P, Jie D F. Navel orange maturity classification by multispectral indexes based on hyperspectral diffuse transmittance imaging. *Journal of Food Quality*, 2017; 7: 1–7.
- [15] Underwood J P, Hung C, Whelan B, Sukkarieh S. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. *Computers and Electronics in Agriculture*, 2016; 130: 83–96.
- [16] Stein M, Bargoti S, Underwood J. Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors*, 2016; 16(11): 1915.
- [17] Gan H, Lee W S, Alchanatis V, Ehsani R, Schueller J K. Immature green citrus fruit detection using color and thermal images. *Computers and Electronics in Agriculture*, 2018; 152: 117–125.
- [18] Gongal A, Silwal A, Amatya S, Karkee M, Zhang Q, Lewis K. Apple crop-load estimation with over-the-row machine vision system. *Computers and Electronics in Agriculture*, 2016; 120: 26–35.
- [19] Tao Y T, Zhou J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers and Electronics in Agriculture*, 2017; 142(A): 388–396.
- [20] Qureshi W S, Payne A, Walsh K B, Linker R, Cohen O, Dailey M N. Machine vision for counting fruit on mango tree canopies. *Precision Agriculture*, 2017; 18: 224–244.
- [21] Sa I, Lehnert C, English A, McCool C, Dayoub F, Upcroft B, et al. Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. *IEEE Robotics and Automation Letters*, 2017; 2(2): 765–772.
- [22] Lehnert C, Sa I, McCool C, Upcroft B, Perez T. Sweet pepper pose detection and grasping for automated crop harvesting. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm: IEEE, 2016; pp.2428–2434.
- [23] Rusu R B, Cousins S. 3D is here: Point cloud library (PCL). In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai: IEEE, 2011; pp.1–4.
- [24] Rusu R B, Marton Z C, Blodow N, Dolha M, Beetz M. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 2008; 56(11): 927–941.
- [25] Fernandez D, Parra I, Sotelo M A, Revenga P, Gavilan M. 3D candidate selection method for pedestrian detection on non-planar roads. In: *2007 IEEE Intelligent Vehicles Symposium*. Istanbul: IEEE, 2007; pp.1162–1167.
- [26] Wu G, Zhu Q B, Huang M, Guo Y, Qin J W. Automatic recognition of juicy peaches on trees based on 3D contour features and colour data. *Biosystems Engineering*, 2019; 188: 1–13.