Computers in Industry 99 (2018) 130–139

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard

Lufeng Luo[a,c], Yunchao Tang[b,1,*], Qinghua Lu[a], Xiong Chen[c], Po Zhang[c], Xiangjun Zou[c,*]

[a] College of Mechanical and Electrical Engineering, Foshan University, 18 Jiangwan Road, Foshan 528000, China
[b] School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China
[c] Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China

ARTICLE INFO

Article history:
Received 30 August 2017
Received in revised form 7 March 2018
Accepted 16 March 2018
Available online 31 March 2018

Keywords:
Vision detecting
Grape cluster
Cutting point
Harvesting robots
Overlapping

ABSTRACT

Reliable and robust vision algorithms to detect the cutting points on peduncles of overlapping grape clusters in the unstructured vineyard are essential for efficient use of a harvesting robot. In this study, we designed an approach to detect these cutting points in three main steps. First, the areas of pixels representing grape clusters in vineyard images were obtained using a segmentation algorithm based on *k*-means clustering and an effective color component. Next, the edge images of grape clusters were extracted, and then a geometric model was used to obtain the contour intersection points of double overlapping grape clusters. Profile analysis was used to separate the regional pixels of double grape clusters by a line connecting double intersection points. Finally, the region of interest of the peduncle for each grape clusters was determined based on the geometric information of each pixel region, and a computational method was used to determine the appropriate cutting point on the peduncle of each grape cluster by use of a geometric constraint method. Thirty vineyard images that were captured from different perspectives were tested to validate the performance of the presented approach in a complex environment. The average recognition accuracy was 88.33%, and the success rate of visual detection of the cutting point on the peduncle of double overlapping grape clusters was 81.66%. The demonstrated performance of this developed method indicated that it could be used by harvesting robots.

1. Introduction

Grape harvesting is a time-consuming and labour-intensive procedure [1]. With the aging of the population and the decrease of the labour force due to fewer young and middle-aged people in China, the timely harvesting of fruits is becoming a significant challenge for fruit growers. To design strategies to improve picking efficiency, it is important to examine the use of intelligent robots for harvesting fruits. For the past 30 years, scholars from different parts of the world have studied fruit-harvesting robots, including harvesting robots for strawberry [2,3], cucumber [4], apple [5,6], sweet-pepper [7], citrus [8], and litchi [9,10]. However, up to now, commercial automatic harvesting robots are still rarely reported, and one of the main reasons is low location accuracy in unstructured orchards. To realize a practical robotics harvesting, the minimum accuracy required for real harvesting systems need be more than 80%.

For grape harvesting by a vision-based harvesting robot, grape characteristics such as softness, irregular shape, and contour preclude the direct grasping of the fruit, so grasping and cutting the peduncle of the grape is an effective alternative for harvesting [1]. Therefore, the accurate recognition and location of the appropriate cutting point on the peduncle of grape clusters in an unstructured vineyard is crucial for picking precision. However, due to the complexity and uncertainty of the orchard environment and the contour irregularity of fruit clusters, it is difficult to identify and locate the optimal plucking positions, particularly under overlapping conditions [11]. In order to achieve non-destructive harvesting, a vision system that can function in the complex orchard environment is urgently required. To address this need, various vision sensing systems and algorithms have been developed to detect overlapping fruits and determine the appropriate plucking positions.

For fruit detection from images, Luo et al. [12] presented a grape detection approach that fused the advantages of multiple color

* Corresponding authors.
E-mail addresses: luolufeng617@163.com (L. Luo), ryan.twain@gmail.com (Y. Tang).
[1] Co-first author.

https://doi.org/10.1016/j.compind.2018.03.017
0166-3615/

space features (e.g., RGB, HSV, La*b*) based on the Adaboost framework. To detect grape clusters for an automatic selective vineyard sprayer, Berenstein et al. [13] proposed an image segmentation process that utilizes the differences in edge distribution between grape clusters and the foliage. Reis et al. [14] presented a detection algorithm for grape clusters in vineyard images captured at night by defining a region of pixel intensities in the RGB color space based on trial and error. In [15], a grape detection method based on both visual texture and shape was developed to detect the green fruit against a green leaf background. To estimate vineyard yield, Font et al. [16] developed a color-based segmentation method for use with vineyard images captured at night by extracting the H component of the HSV color space. In [17], a grape bunch segmentation algorithm combining color and texture information and the use of a support vector machine (SVM) was proposed. Kondo et al. [18] proposed a grape identification method based on spectral properties. To overcome the challenges of the proper identification of overlapping apples, Xu et al. [19] presented a segmentation method based on a Snake model to obtain the contour of the overlapping apples, and then a corner detection algorithm based on distance measure was used to find the corner points of overlapped apples. A line that connected two of the detected corner points was then used to distinguish the overlapping apples. Xiang et al. [20] presented an algorithm based on binocular stereo-vision to recognize clustered tomatoes. In this approach, the clustered regions were first classified into two types (i.e., adhering and overlapping) based on the depth information obtained by a binocular stereo-vision system, and then the tomatoes were separated using an edge curvature analysis method.

To determine the plucking position, a precision harvesting strategy for sweet pepper was proposed by Bac et al. [7]. In this approach, the support wire that twisted around the stem was used as a visual cue to locate the pepper stem. This was also the first study of stem localization under varying lighting conditions. Yang et al. [21] presented an approach to locate targets for a grape-bagging robot by extracting the external rectangle of the grape contours, but this study did not address the problem of the localization of the overlapping grape clusters. Zhang et al. [22] proposed an image processing procedure to determine the plucking position of strawberry. In this method, the strawberry images were first segmented by color difference, and then the barycenter and the tip position of the strawberry image were extracted by computing the geometric feature of the segmented strawberry image. This information is then used to determine the plucking position. Li et al. [23] a presented a barycenter calculation approach for pineapple based on a monocular vision that employed image processing and morphological technologies to get fruit-eye central points, allowing application of a hierarchical clustering algorithm to calculate the approximate barycenter of the pineapple. Guo et al. [24] proposed an approach based on the Harris algorithm to determine the plucking position of litchi, in which the cutting points of the litchi were calculated by detecting the corner points on the stem of the litchi.

The above studies of vision detection methodology applied in unstructured orchard environments have focused on image segmentation as a strategy to estimate the vineyard yield. Although many fruit detection methods for harvesting robots have been proposed by researchers, an ideal method to accurately determine the plucking position of two overlapped grape clusters has not yet been described.

To ensure picking precision in an unstructured vineyard, an approach to detect the cutting points on the peduncles of double overlapping grape clusters was developed. First, the regions of pixels containing grape clusters in vineyard images were identified using a segmentation algorithm based on *k*-means clustering and the H component of the HSV color space. Next, the edge images of grape clusters were extracted based on the segmented image, and then a geometric model was established to obtain the contour intersection points of double overlapping grape clusters by using profile analysis, such that the regional pixels of double grape clusters were separated by a line connecting double intersection points. Finally, the regions containing the peduncle for each grape cluster were identified based on the geometric information of each pixel region, and a computing method was used to detect the cutting point on the peduncle of each grape cluster by a geometric constraint method. The innovation of this study is the development of a sequential machine vision procedure to determine the plucking position of double overlapped grape clusters based on profile analysis and geometric constraint.

2. Vision equipment and image acquisition

The vision sensor and test platform are as follows: a digital camera (D5200, Nikon, Wuxi, China) and a laptop (Lenovo T430) with 4G RAM and Intel(R)Core(TM)i5–3230 M CPU@2.60 GHz. The software systems include a Windows 7 operating system, OpenCV2.4.13, and a Visual C++ 2013 programming environment. The summer black grape was selected as the test variety. To verify the practicability of the method proposed in this study, we captured test images on July 27th, 2016 (sunny day) and August 6th, 2016 (cloudy day) during the ripe season of the summer black grape. Images were captured by the digital camera in the Tianjin ChaDian Grape Science Park using auto-exposure control mode for shutter speed, and the exposure time was fixed to 1/100s. Thirty images containing double overlapping grape clusters were recorded using the camera at a resolution of 2592×1944 pixels, and most parts of the grape clusters were not covered by other objects. The shooting distance was 600–1000 mm. In order to reduce the calculation time of the algorithm, all of the tested images were resized to 640×480 pixels using a bicubic interpolation algorithm.

3. Detection of a cutting point on the peduncle of a grape cluster

The approach to detect the cutting points on the peduncles of double overlapping grape clusters consists mainly of the following three steps, and the overall procedure of this approach is summarized in Fig. 1.

Step1: the pixel area of grape clusters was obtained using a segmentation algorithm based on the k-means clustering method and the effective color component. The details are presented in Section 3.1.

Step 2: the edge images of grape clusters were extracted, and then a geometric model was established to obtain the contour intersection points of double overlapping grape clusters by profile analysis. Next, double grape clusters were separated and the regional pixels of each grape clusters were obtained by a line connecting double intersection points. The detailed procedures are described in Section 3.2.

Step 3: the barycenter and external rectangular size of each pixel area of the grape cluster were calculated, allowing determination of the region containing the peduncle for each grape cluster. A computational method is then used to detect the cutting point on the peduncle of each grape cluster using a geometric constraint method. The details are presented in Section 3.3.

3.1. Grape image segmentation

The first key step in the procedure for the detection of overlapping grape clusters is image segmentation. Recently described methods to detect fruits in images are mostly color-based or shape-based

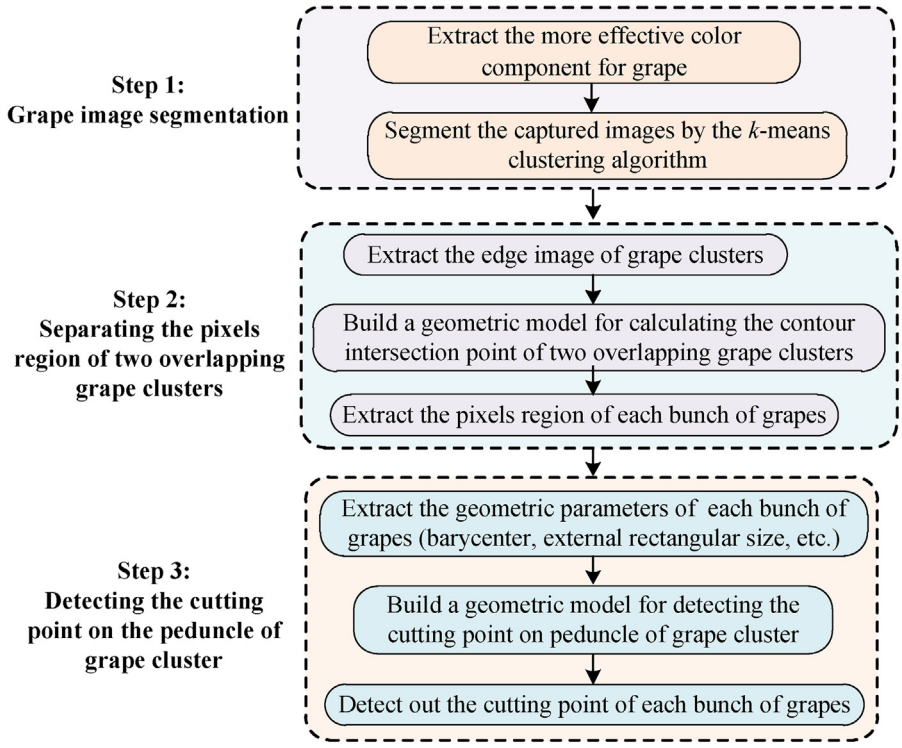


Fig. 1. Schematic illustrating the proposed approach. The three main steps are shown on the left and the detailed sub-steps are presented to the right.

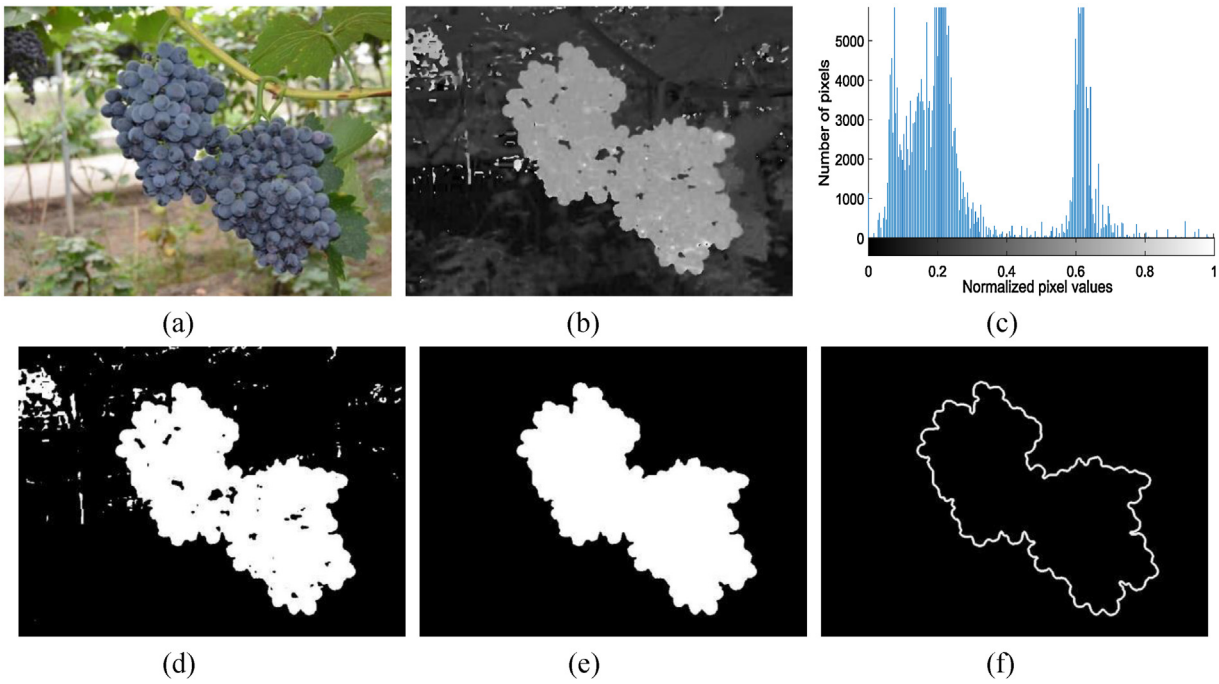


Fig. 2. Procedure of grape image segmentation. (a) Original image; (b) H component image; (c) the histogram of color component H; (d) Obtained binary image after K-means clustering; (e) the segmentation162 image; (f) the edge image of the overlapping grape clusters.

analysis [6]. Given the irregular shape of the overlapping grape clusters, color-based analysis was adopted. To find an effective color indicator to recognize the grape clusters from the captured vineyard images, we converted each captured image to the color spaces (e.g., HSV, HIS, YCbCr, and L*a*b), and then compared and analyzed the color differences between the summer black grapes and the background (e.g., leaves, stems, soil, and sky). The H component of the HSV color space was found to be an effective color indicator that can distinguish the summer black grapes from the background. Fig. 2(b) shows the H component image of Fig. 2(a), and the

corresponding histogram can be seen in Fig. 2(c). The H component can be calculated according to Eq. (1) [25]:

$$H = \begin{cases} \theta & B \le G \\ 360 - \theta & B > G \end{cases} \quad (1)$$

In which R, G, and B indicate the primary colors (red, green, and blue), and the θ can be computed by Eq. (2):

$$\theta = \arccos\left\{ \frac{0.5 \times [(R-G)+(R-B)]}{\left[(R-G)^2 + (R-B)(G-B)\right]^{1/2}} \right\} \quad (2)$$

From the data shown in Fig. 2(c), we can see that the histogram of the H component shows good wave peak and trough effect. There are two apparent peaks, one peak corresponds to the grape cluster and the other corresponds to the background. A peak can be defined as a category, and a clustering method can be applied to grape image segmentation. The *k*-means clustering method was adopted in this study, which is an unsupervised learning algorithm. This method aims to partition n observations into k clusters in which each observation is in the cluster with the nearest mean [26]. In this algorithm, the number of clusters and the initial cluster center must be initially set, and their initial values directly affect the efficiency and the results of the algorithm. Because the histogram of the extracted H component images showed two peaks (i.e., background and grape), the number of clusters was set to 2. The values of the two peaks were used as the two corresponding cluster centers to reduce iterations.

After segmenting the captured images by the *k*-means clustering algorithm, some noise and some small areas of background remained in the initial image segmentation, as shown in Fig. 2(d). We used a region threshold method to remove the small discrete regions of background pixels. To do this, the biggest region of neighboring white pixels in the initial segmented image was first extracted, and then all objects of area less than 1/4 of the largest region in the image were deleted as background through debugging and preliminary experiments. The marginal noise in the image was further eliminated by performing morphological opening and closing operations in which an open filtering with a radius of 4 of the disc structure element was implemented, subsequently a close operator was executed to recover the contour of the grape cluster. Finally, hole filling was performed to the target region to obtain a complete and closed grape image region. The final segmentation results are shown in Fig. 2(e).

3.2. Separating the pixel regions of the overlapping grape clusters

Due to the irregular contour and the uncertain location for several overlapping grape clusters in the orchard environment, it was difficult to extract each overlapped grape targets. To find a solution to this difficult problem, we analyzed and compared the contours of double overlapping grape clusters from the captured images. It can be concluded the border which locates between double grape cluster edges with double inflection points near the center area often appears in pairs. Inspired by the above analysis, we proposed a solution to extract the double overlapping grape clusters separately by solving the border contours of the inflection points of the double overlapping grape clusters as described in detail in the following section.

3.2.1. A geometric model to calculate the inflection points of the edges of double grape clusters

After obtaining the region of the grape clusters by the segmentation algorithm described in Section 3.1, the edge image of the overlapping grape clusters was extracted by using the Canny edge detector operator [27], as shown in Fig. 2(f). To obtain the inflection points at the junction of the double overlapping grape clusters, a geometry calculation model to determine the dividing line between double grape clusters was built after analyzing the contour characteristics, as shown in Fig. 3. The horizontal-

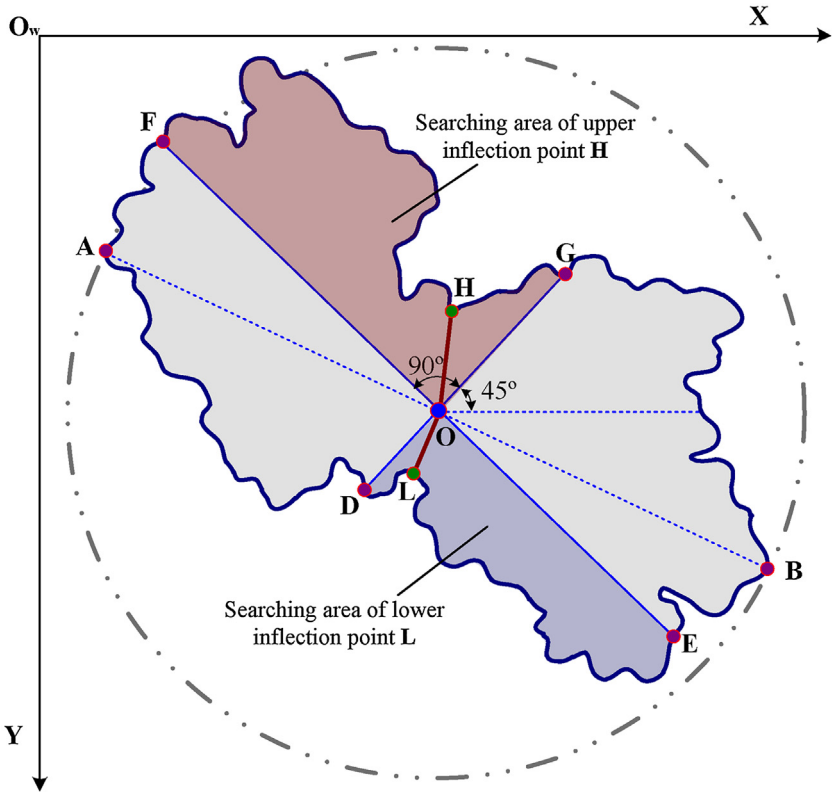


Fig. 3. Geometric calculation model of the intersection point of the edges of double grape clusters.

rightward and vertical-downward were set as the positive direction of the X-axis and Y-axis, respectively. The extreme values of A and B in the obtained grape region image were computed (see Section 3.2.2), and then the midpoint of the line that crossed the extreme points A and B were calculated. The midpoint was then set as the origin point O and the horizontal-right was used as the starting angle. The regions of interest of the upper inflection point **H** and the lower inflection point **L** were determined by defining three constraining rules (described below) and the inflection points were solved by the minimum distance constraint (the detailed procedures are presented in Section 3.2.2). Finally, the dividing line of double grape clusters was obtained by connecting the double intersection points, allowing separate extraction of the double grape clusters.

3.2.2. Extracting the region of pixels for each grape cluster

For grape image segmentation, the edge binary images of the overlapping grape clusters were extracted by using the Canny edge detector operator [27], and the edge contour pixel value was set to 1 (white) and the rest of the region was set to 0 (black). Then, the left and right extreme value points were obtained by searching the maximum and minimum coordinates of the pixels in the X direction in the contour. Finally, the center of the circle around the grape clusters was calculated by solving the midpoint coordinates of the line between the left and right extreme value points, as shown in Fig. 4(a).

The upper and lower inflection points were determined by searching the minimum distance between the contour and the center O. The equation is as follows:

$$P(x,y) = min\sqrt{(x-x_0)^2 + (y-y_0)^2} \tag{3}$$

Where (x_o, y_o) is the origin coordinate, (x, y) is the pixel coordinate of the grape contour in the image, and $P(x,y)$ indicates the minimum distance between the contour and the center O. In order to eliminate the interference of some irrelevant contours, it is necessary to determine the regions of interest of the upper inflection point **H** and the lower inflection point **L**. Therefore, the following three constraining rules were set to solve Eq. (3):

Rule 1: For the upper contour inflection point, the pixel coordinates must be located above the center of the circle O. Similarly, the lower contour inflection point must be below the O point.

Rule 2: The upper and lower inflection points must be located on the region of interest of the curves $\tilde{DE}$ and $\tilde{FG}$. To determine the curve starting and ending points, a preliminary experiment was performed to test the performance of the extraction accuracy for each bunch of overlapping grape targets by adjusting the search degree. After testing [28], the curve in the clockwise direction from 45 ° to 135 was determined as the region of interest of the upper inflection point **H**, and the area from 225 ° to 315 ° was set as the region of interest of the lower inflection point **L**. Through this constraint, some redundant edges can be eliminated and the edge contours of interest at the inflection point can be obtained, as shown in Fig. 4(b).

Rule 3: The upper and lower inflection points must be located on the edge of the double grape clusters, i.e., the inflection points must be included in the contour.

$$\begin{cases} 0.707 \le \dfrac{\begin{matrix} y - y_0 < 0 \\ y_0 - y \end{matrix}}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \le 1 \\ (x,y) \in \tilde{FG} \end{cases}$$

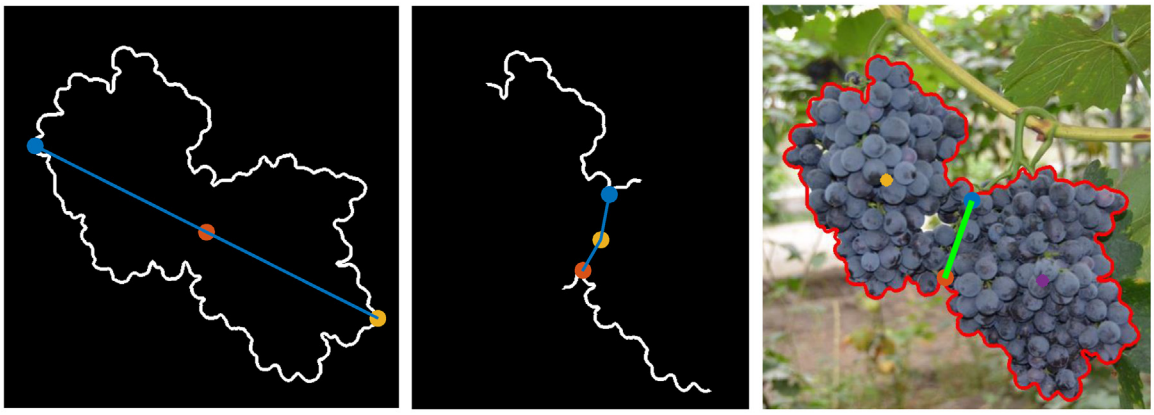
$$\begin{cases} -1 \le \dfrac{\begin{matrix} y - y_0 > 0 \\ y_0 - y \end{matrix}}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \le -0.707 \\ (x,y) \in \tilde{DE} \end{cases}$$

The above three constraining rules can be formulated by Eqs. (4) and (5). The upper inflection point (x_H, y_H) can be obtained by solving Eq. (3) and using the constraint of Eq. (4), Similarly, the lower inflection point (x_L, y_L) can be obtained by using the constraint of Eq. (5) to solve Eq. (1). The obtained inflection points were determined and are shown in Fig. 4(b).

After obtaining the upper and lower inflection points **H** and **L** by the above steps, the dividing line of the double overlapped grape clusters can be drawn by connecting **H** and **L**, as shown in Fig. 4(c). Finally, the region containing the pixels corresponding to each grape cluster can be extracted separately.

3.4. Detecting the cutting point on the peduncle of a grape cluster

After obtaining the section of pixels corresponding to each grape cluster, the geometric information of each region (e.g., barycentre, top point, and external rectangle) was extracted by geometric morphological calculation. On this basis, the region of interest (ROI) of the peduncle was determined. This allowed the establishment of the computing model of the cutting point on the peduncle by using the following three steps.



(a) (b) (c)

Fig. 4. Extraction process of double overlapping grape clusters. (a) left-right extreme value points and centre point, (b) two computed inflection points (c) the dividing line of double overlapping grape clusters.

3.4.1. Determining the ROI of the peduncle

The peduncle is usually located right above the grape cluster due to gravity. To decrease interference due to the background and to increase the rate of detection of the cutting point, the mechanism of visual interest was adopted to reduce the searching range of the peduncle in the images. Thus, the region of interest of the peduncle was set right above the grape cluster, as shown in Fig. 5(a). The size and position of the ROI of the peduncle was determined by the top point coordinates (x_t, y_t), the barycentre (x_c, y_c), and the maximum Euclidean distance between the left and right points of the contour ($L_{\max}$). The length of the ROI was calculated by $aL_{\max}$, and the height of the ROI was determined by $b|y_c - y_t|$. To determine the optimal parameters, the cutting point detection method was tested by adjusting the coefficients a and b between 0∼1. When the length was set as $0.8L_{\max}$ and the height was $0.5|y_c - y_t|$, the cutting point detection was the best. Thus, the sizes and left-up point (x_r, y_r) of the ROI can be determined by Eq. (6).

$$\begin{cases} Roi\_H = 0.5|y_c - y_t| \\ Roi\_L = 0.8L_{max} \\ x_r = x_c - 0.5Roi\_L \\ y_r = y_t - Roi\_H \end{cases} \tag{6}$$

Where *Roi_H* is the height of the ROI, and *Roi_L* is the length of the ROI.

3.4.2. Detecting lines in the ROI

A substantial amount of noise usually exists in images that are captured from an orchard due to shadows and varying illumination. The peduncle of the grape clusters resembled a small circular tube, and their images showed a bilateral character. To reduce noise and enhance the edge images, a bilateral filter was applied to the ROI during pre-processing. This was a non-linear, edge-preserving, and noise-reducing smoothing filter for the images [29]. The bilateral filter was applied by calling the function 'bilateralFilter' in OpenCV, and the size of the filter mask was fixed at 7, the photometric spread in the image range was 20.0, and the geometric spread in the domain was 4.0. Then, the Canny edge detector operator was taken as the edge- detecting operator. Fig. 5(c) shows the edge image of the ROI of the peduncle. The edge of the peduncle was close to a straight line. To detect this edge, Hough line detection was performed based on the cumulative probability [30]. The number of maximum votes of the detection lines was set to sixteen, and the length of the line was at least twenty pixels. All of the lines that fulfilled these conditions were selected, and the pixel coordinates of the endpoints of these lines were recorded. Fig. 5(d) shows the detected lines in the ROI.

3.4.3. Calculating the pixel coordinates of the cutting point

After obtaining the endpoint coordinates of the detected lines, the lines can be described by the following mathematical equation:

$$L_i(x, y) = (x - x_{i1})(y_{i2} - y_{i1}) + (y - y_{i1})(x_{i1} - x_{i2}) \tag{7}$$

In which (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) are the pixel coordinates of the endpoints, and $L_i(x, y)$ is the straight line equation of *i*.

Due to the action of gravity, the detected line that passes through the peduncle should pass through the barycentre in theory. However, in a real situation, errors in image segmentation, irregular contours of the grape clusters, obstructions due to foliage and branches, and other factors could cause the detected line on the edge of the peduncle to show a slight deviation from the barycentre. To determine the line located on the peduncle, a minimum distance restraint between the barycentre and the detected lines was adopted. The resulting line could be selected by solving the minimum distance between the detected lines and the barycentre. The equation to solve the minimum distance is as follows:

$$D_{result\_line} = min|\frac{(x_c - x_{i1})(y_{i2} - y_{i1}) + (y_c - y_{i1})(x_{i1} - x_{i2})}{\sqrt{(y_{i2} - y_{i1})^2 + (x_{i1} - x_{i2})^2}}| \tag{8}$$

Finally, the midpoint of the resulting line was then selected as the cutting point of the grape cluster. The red point in Fig. 6 indicates the detected cutting point for each cluster.

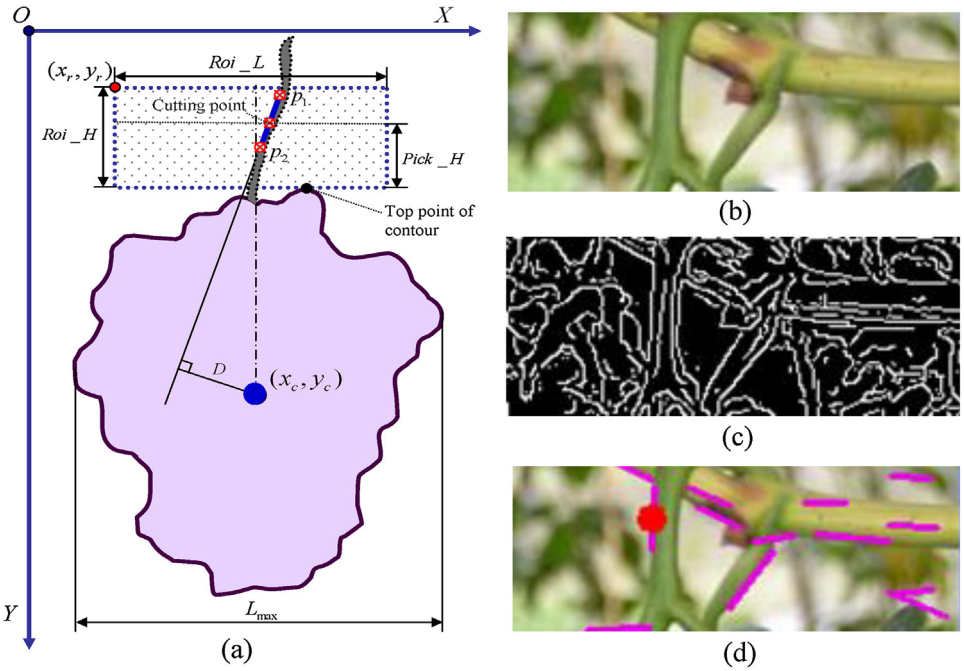


Fig. 5. The computational process to detect the cutting point on the peduncle of the single grape cluster. (a) schematic diagram of the cutting point computational model, (b) the ROI of the peduncle, (c) edge image of the ROI of the peduncle, (d) the detection result image in which the red point indicates the cutting point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

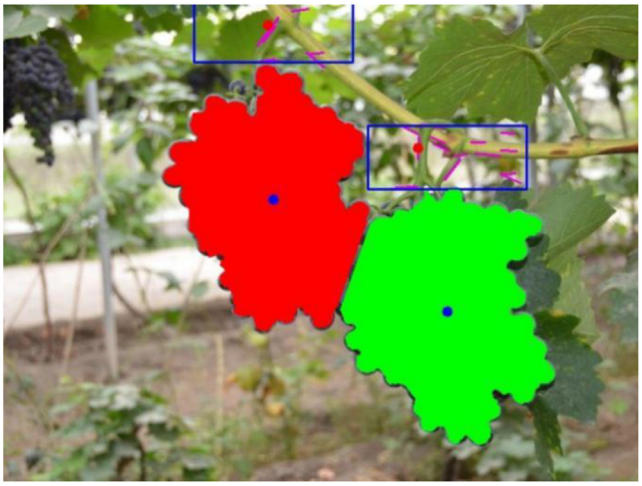


Fig. 6. Detection results of the cutting point in Fig. 2(a).

4. Experiments and results

To validate the performance of the developed approach in a complex orchard, 30 vineyard images captured from different perspectives were tested. Two statistical parameters were determined to assess the robustness of the proposed vision-detecting algorithm. First, the recognition accuracy of the selected section of pixels for each grape cluster was computed by defining two evaluation criteria (described below). Then, the location of the cutting point on the peduncle of double overlapping grape clusters was assessed to obtain the success rate, and the real-time performance of this algorithm was also investigated.

4.1. Recognition accuracy test on double overlapping grape targets

4.1.1. To measure the recognition accuracy for each bunch of overlapping grape targets, a binary image I_1 computed by the proposed algorithm, was compared with a binary ground truth I_2 image

To make this comparison, the true positive rate (*TPR*) and false positive rate (*FPR*) were adopted to evaluate the proposed algorithm. The two metrics TPR and FPR can be computed by $T(I_1, I_2)$ and $F(I_1, I_2)$, respectively, which were defined as Eqs (9) and (10):

$$T(I_1, I_2) = \frac{\sum \left[V_{x,y}(I_1) \times V_{x,y}(I_2)\right]}{\sum V_{x,y}(I_2)} \times 100\% \tag{9}$$

$$F(I_1, I_2) = \frac{\sum \left[V_{x,y}(I_1) \times \overline{V}_{x,y}(I_2)\right]}{\sum V_{x,y}(I_2)} \times 100\% \tag{10}$$

In which $V_{x,y}(I_1)$ corresponds to the value (either 0 or 1) of the pixel at coordinates x, y in image I_1, and $\overline{V}_{x,y}(I_1) = 1 - V_{x,y}(I_1)$. Thus $T(I_1, I_2)$ computes the percentage of pixels in I_2 that belong to the area of interest that was extracted in I_1. $T(I_1, I_2)$ can be considered as the percentage of every grape cluster pixels extracted in I_1. $F(I_1, I_2)$ describes the percentage of pixels that were extracted in I_1 but do not belong to the area of interest in I_2. $F(I_1, I_2)$ is the percentage of background pixels extracted in I_1. In the case of a perfect extraction, $T(I_1, I_2) = 100$ and $F(I_1, I_2) = 0$. Finally, in order to obtain a metric using more than one image to measure the extraction performance, we define:

$$\hat{T}(I_1, I_2) = \frac{\sum_{n=N}^{n=1} T(I_1(n), I_2(n))}{N} \tag{11}$$

$$\hat{F}(I_1, I_2) = \frac{\sum_{n=N}^{n=1} F(I_1(n), I_2(n))}{N} \tag{12}$$

In which N represents the number of tested images, and $I_1(n)$ and $I_2(n)$ are obtained using the *nth* image of the set of images. Thus $\hat{T}(I_1, I_2)$ and $\hat{F}(I_1, I_2)$ provide the mean value of algorithm performance, as evaluated over a set of N images.

After defining the above evaluation criteria, 30 vineyard images with double overlapping grape clusters were selected as testing images for the algorithm proposed in Section 2.2.1 and 2.2.2. The contour and boundaries of each grape cluster were individually extracted, allowing the binary image I_1 to be obtained that only included a grape cluster. Next, the binary ground truth images (I_2) of the overlapped grape clusters were manually extracted by using the software Photoshop. Finally, the metrics $T(I_1, I_2)$ and $F(I_1, I_2)$ were computed by using Eqs. (9) and (10), and the mean values of algorithm performance were calculated by Eqs. (11) and (12). Fig. 7 shows the results for the TPR and FPR of target pixel extraction.

The TPR values of the pixel region extraction of the overlapped grape clusters were between 63.45% and 96.12%, and the mean

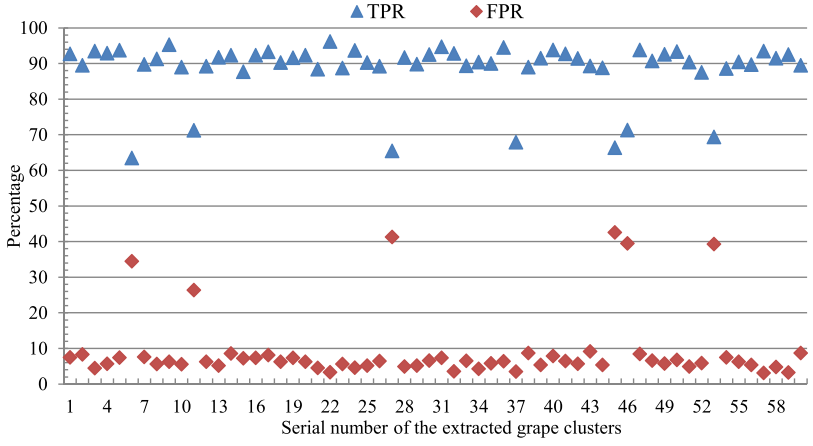


Fig. 7. True positive rate (TPR) and false positive rate (FPR) of target pixel extraction for 30 testing images.

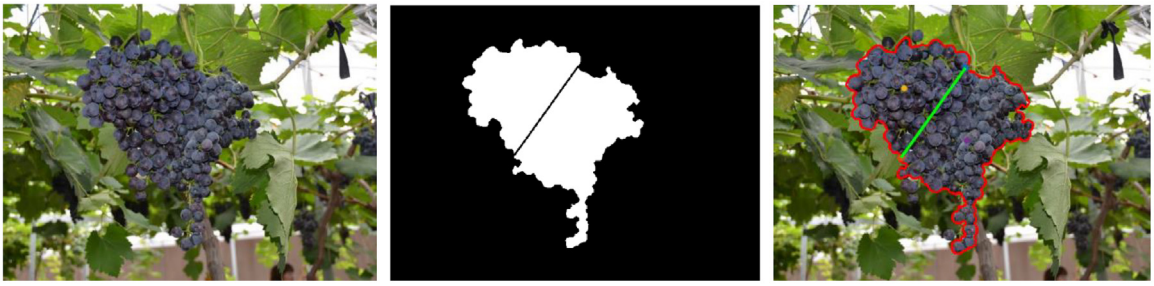


Fig. 8. An inaccurate extraction caused by significantly overlapped double grape clusters.

value of TPR was 88.57%. In all of the tested images, there were 53 grape clusters with calculated TPR values greater than 85%, and others were less than 75%. The FPR values of the overlapped targets were between 3.12% and 42.56%, and the mean value of FPR was 9.21%. There were 7 grape clusters with calculated FPR values greater than 25%, and the rest were less than 10%. From the experimental results, we can see that for clusters with TPR values less than 75%, the corresponding FPR values were greater than 25%. This is because the extracted dividing line of the double grape clusters showed a larger deviation. This deviation was mainly caused by significant overlap between double grape clusters, making the inflection points of the contour not very obvious, giving rise to errors in the calculations of the upper and lower inflection points. For example, the lower inflection points of double grape clusters in Fig. 8 was inaccurately computed, causing part of the region of the left grape cluster to be attributed to part of the right grape cluster. If the metrics of TPR (85%) and FPR (10%) were used as the evaluation criteria, the region extraction accuracy of the double overlapping grape clusters was near 88.33% for the set of tested images.

4.2. Location test of a cutting point on the peduncle of double overlapping grape clusters

The detection accuracy of the cutting point on the peduncle of overlapping grape clusters is an important metric of a machine vision system for a harvesting robot. To determine the success rate of this method, 60 grape clusters in the 30 vineyard images were adopted as testing examples, and the vision location test of the cutting point was evaluated by Eq. (13).

$$SR = \frac{\sum_{k=1}^{k=M} L(k)}{M} \times 100\%, \begin{cases} L(k) = 1, if\, CP(k) \in R_p(k) \\ L(k) = 0, if\, CP(k) \notin \overline{R}_p(k) \end{cases} \quad (13)$$

In which SR is the successful detection rate of the cutting point in M grape clusters, $L(k)$ corresponds to the location value (either 1 or 0), where the value 1 indicates successful location, and the value 0 indicates failed location; $CP(k)$ indicates the pixel coordinate of the cutting point on the peduncle of the kth grape cluster as computed by the proposed algorithm; $R_p(k)$ is the image region of the ground truth peduncle of the kth grape cluster, and M is the number of the tested grape clusters.

Fig. 9(a) and (b) show several successful cases of cutting point detection, and Fig. 9(c) shows a failure. From the images shown in Fig. 9(a) and (b), we can see that the detected cutting point lies on the ground truth peduncle in the testing images. In our approach, the peduncle of the grape cluster was detected by Hough line detection, and in most cases, several lines can be detected. Eventually, the cutting point was mapped to one of the lines by imposing the smallest distance constraint between the line and the barycentre. Mapping the cutting point to any one line of the peduncle was considered a successful detection. However, if the ROI of the peduncle included some branches, leaf stem or other columns, it may be difficult to map the peduncle to the ROI, affecting the accurate detection of the cutting point. The cutting point of the left grape cluster in Fig. 9(c) was incorrectly computed, and the computed cutting point was on the leaf stem. From the detected image, we can see that the peduncle is very short and only a smaller part of the peduncle located to the ROI, resulted in an inability to determine the appropriate cutting point.

To evaluate the location accuracy of the cutting point determination, we asked if the detected cutting point was located on the peduncle. The results of the cutting point detection for the 30 vineyard images were statically analyzed, as shown in Table 1.

From the statistics shown in Table 1, we can see that a cutting point detection of the overlapping grape cluster was about 81.66% successful for the 30 tested vineyard images. This was lower than the success rate of a single grape cluster, 87%, as reported previously [1]. Of the 30 vineyard images with double overlapping grape clusters, there were three images for which the algorithm incorrectly extracted the grape cluster information, causing inaccurate cutting point detection of four grape clusters. One image showed such a large error in target extraction that the cutting point of one grape cluster was falsely detected. Moreover, there were four grape clusters whose peduncle shape were abnormal or too short, causing edge detection failure and leading to an error in cutting point determination.

In the experiments, some of the grape clusters that were partially occluded were also tested, as shown in Fig. 10. When the peduncles of the grape clusters were not blocked by other obstacles, the cutting point could still be calculated using the proposed method.

To validate the real-time performance of the proposed algorithm, the elapsed time of the whole process from image segmentation to detection of the cutting points for every tested image was calculated using the clock function in C language. The amount of time required to test each of the 30 images was determined and these data are presented in Fig. 11. As can be seen from the figure, the running time of each tested image was different, but the time was between 0.53 and 0.69s, and thus the real-time performance of the proposed approach can meet the demand of a grape harvesting robot.

5. Discussion

The current implementation of the procedure for detection of the appropriate cutting point on the peduncle of double overlapping grape clusters can meet the requirements for a harvesting robot. In the 30 tested vineyard images, the region extraction accuracy for double overlapping grape clusters was near 88.33%, the success rate of cutting point detection of overlapping grape clusters was about 81.66%, and the elapsed time for the whole algorithm for each image was between 0.53 and 0.69s.

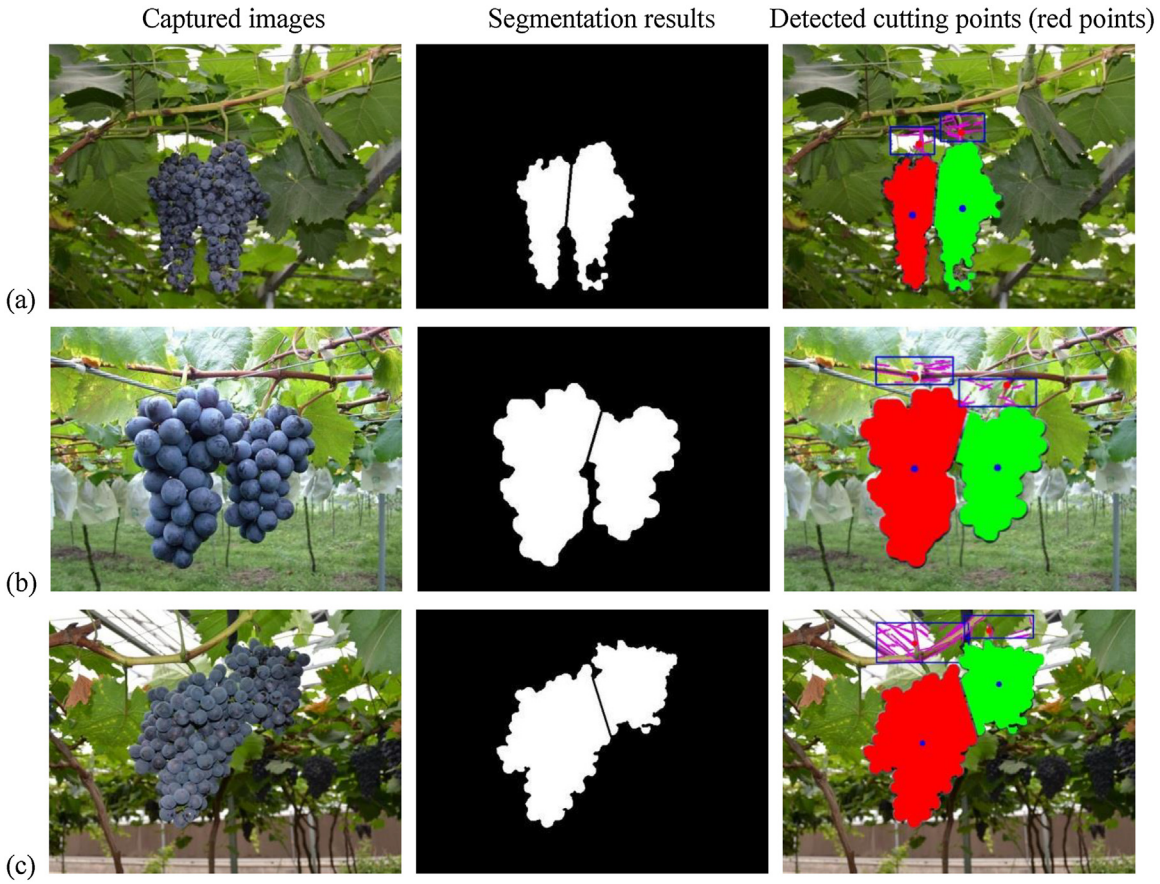


Fig. 9. Detection of the cutting point of double adjacent and overlapping grape clusters. (a) – (b) are successful detections, and (c) shows an inaccurate detection.

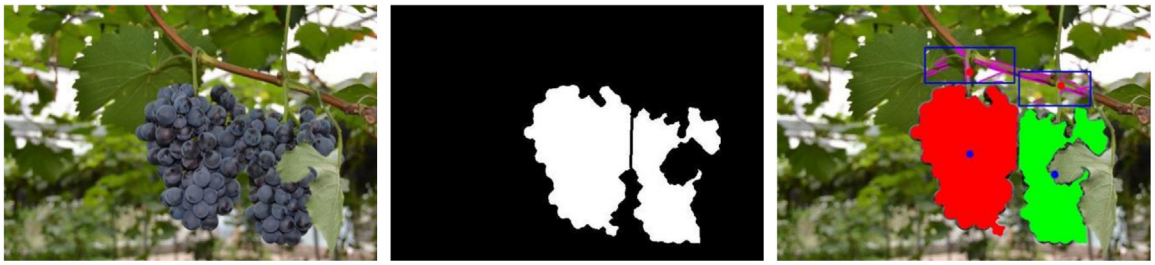


Fig. 10. Detection of the cutting point of double adjacent and overlapping grape clusters.

From the experimental results, we know that the proposed approach can accurately detect the cutting points on the peduncle of double overlapping grape clusters. However, this approach depends strongly on the pixel region extraction accuracy of double overlapping grape clusters and reasonable line detection of the peduncle's edge. Thus, there are still some factors that affect the performance of the proposed algorithm. The reasons for inaccurate cutting point detection include, (1) unstructured factors such as changing illumination, grape clusters obscured by leaves, and unripe berries mixed with ripe grape clusters that reduce the robustness of the segmentation algorithm, causing poor extraction of the appropriate section of pixels. A significant error of the extracted pixel region of grape clusters caused the corresponding parameters in the calculation model of the cutting point to be inaccurate, potentially causing detection failure. (2) The degree of overlap of double grape clusters and the degree occluded by leaves can significantly influence this algorithm. When a high degree of overlap

Table 1
Statistics of cutting points located on double overlapping and adjacent grape clusters.

Number of tested images	Number of grape clusters	$CP(k) \in R_p(k)$	SR/%
30	60	49	81.66

Fig. 11. The analysis time required for the 30 tested images.

or occlusion occurs between double grape clusters, the inflection points of their contour may not be easily detected, resulting in errors in the calculation of the cutting points. (3) The peduncle was detected by the line detection method, and the detected lines usually lie on one edge of the peduncle. Which line includes the cutting point was determined by using the minimum distance constraint between the detected lines and the barycenter. However, when the shape of the peduncle was out of normal range, such as too short in length or with too big of a curvature, the line detection will be inaccurate, leading to failure in cutting point detection.

To reduce the influence of the complex and uncentain vineyard environment, future research efforts should test other algorithms and sensors. For example, deep learning methods and depth cameras could be used to build a combined vision system [31]. Although training a robust depth neural network requires a large number of vineyard image samples, deep learning is a promising approach for accuracy in an unstructured and complex outdoor environment. Additionally, use of a depth camera can obtain depth information, and integrating the information from depth cameras and color cameras may improve the detection of overlapping targets.

6. Conclusion

The accurate detection and location of the cutting point on the peduncle of overlapping grape clusters is a significant challenge for picking precision by a harvesting robot. This study focused on detecting the cutting points on the peduncles of double overlapping grape clusters from vineyard images. The pixel region of the overlapping grape clusters was first obtained, and then the edge images of grape clusters were extracted. The areas of pixels for double overlapping grape clusters were separated into two independent regions. We then determined the region of interest of the peduncle for each grape cluster and detected the cutting point on the peduncle of each grape cluster by a geometric constraint method. Thirty images were captured from different perspectives in a vineyard. They were used to be testing samples for testing the ability of the proposed algorithm. The recognition accuracy for overlapping grape clusters was 88.33%. The success detection rate of the cutting point on the peduncle of double overlapping grape clusters was 81.66%. The demonstrated performance indicated the developed approach can effectively detect the cutting points for double overlapping grape clusters for a harvesting robot in a complex vineyard environment. However, the results of the proposed method depend strongly on the accuracy of the extraction of the pixel region for each grape cluster. When the extraction accuracy is not ideal, the method may fail in some situations. Overall, the detection accuracy of multiple overlapped grape clusters remains an issue that needs to be solved in further research. Future steps may include the use of deep learning method by capturing and processing a large number of learning samples of vineyard images.

Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China (No. 51705365, 31571568), the Project of Province Science and Technology of Guangdong (2015A020209111), and the Project of Science and Technology of Huizhou (2014B040008006).

References

[1] L. Luo, Y. Tang, X. Zou, M. Ye, X. Feng, G. Li, Vision-based extraction of spatial information in grape clusters for harvesting robots, Biosyst. Eng. 151 (2016) 90–104.
[2] N. Kondo, M. Monta, K. Hisaeda, Harvesting robot for strawberry grown on annual hill top (Part 2), J. Soc. High Technol. Agric. 13 (4) (2001) 231–236.
[3] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, M. Kurita, Evaluation of a strawberry-harvesting robot in a field test, Biosyst. Eng. 105 (2010) 160–171.
[4] E.J. Van Henten, B.A.J. Van Tuijl, J. Hemming, J.G. Kornet, J. Bontsema, E.A. Van Os, Field test of an autonomous cucumber picking robot, Biosyst. Eng. 86 (3) (2003) 305–313.
[5] D. Zhao, J. Lv, W. Ji, Y. Zhang, Y. Chen, Design and control of an apple harvesting robot, Biosyst. Eng. 110 (2011) 112–122.
[6] Y. Si, G. Liu, J. Feng, Location of apples in trees using stereoscopic vision, Comput. Electron. Agric. 112 (2015) 68–74.
[7] C.W. Bac, J. Hemming, E.J. van Henten, Stem localization of sweet-pepper plants using the support wire as a visual cue, Comput. Electron. Agric. 105 (2014) 111–120.
[8] S.S. Mehta, T.F. Burks, Vision-based control of robotic manipulator for citrus harvesting, Comput. Electron. Agric. 102 (2014) 146–158.
[9] X. Zou, M. Ye, C. Luo, J. Xiong, L. Luo, H. Wang, Y. Chen, Fault-tolerant design of a limited universal fruit-picking end-effector based on visoin positioning error, Appl. Eng. Agric. 32 (1) (2016) 5–18.
[10] C. Wang, X. Zou, Y. Tang, L. Luo, W. Feng, Localisation of litchi in an unstructured environment using binocular stereo vision, Biosyst. Eng. 145 (2016) 39–51.
[11] C.P. Julio, R. Thomas, Novel image processing approach for solving the overlapping problem in agriculture, Biosyst. Eng. 115 (2013) 106–115.
[12] L. Luo, Y. Tang, X. Zou, C. Wang, P. Zhang, W. Feng, Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components, Sensors 16 (12) (2016) 2098.
[13] R. Berenstein, O. Ben Shahar, A. Shapiro, Y. Edan, Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer, Intell. Serv. Rob. 3 (2010) 233–243.
[14] M.J.C.S. Reis, R. Morais, E. Peres, C. Pereira, O. Contente, S. Soares, A. Valente, J. Baptista, P.J.S.G. Ferreira, J. Bulas Cruz, Automatic detection of bunches of grapes in natural environment from color images, J. Appl. Logic 10 (2012) 285–290.
[15] S. Nuske, S. Achar, T. Bates, S. Narasimhan, S. Singh, Yield estimation in vineyards by visual grape detection, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30, 2011, pp. 2352–2358.
[16] D. Font, M. Tresanchez, D. Martínez, J. Moreno, E. Clotet, J. Palacín, Vineyard yield estimation based on the analysis of high resolution images obtained with artificial illumination at night, Sensors 15 (2015) 8284–8301.
[17] S. Liu, M. Whitty, Automatic grape bunch detection in vineyards with an SVM classifier, J. Appl. Logic 13 (2015) 643–653.
[18] N. Kondo, Y. Shibano, K. Mohri, Basic studies on robot to work in vineyard (part 2), J. Jpn. Soc. Agric. Mach. 56 (1) (1994) 45–53.
[19] Y. Xu, Y. Li, H. Song, et al., Segmentation method of overlapped double apples based on Snake model and corner detectors, Trans. Chin. Soc. Agric. Eng. 31 (1) (2015) 196–203.
[20] R. Xiang, H. Jiang, Y. Ying, Recognition of clustered tomatoes based on binocular stereo vision, Comput. Electron. Agric. 106 (2014) 75–90.
[21] Q. Yang, C. Liu, Y. Xun, et al., Target recognition for grape bagging robot, Trans. Chin. Soc. Agric. Mach. 44 (8) (2013) 234–239.
[22] T. Zhang, L. Chen, J. Song, Study on strawberry harvesting robot: images based identifications of strawberry barycenter and plucking position, J. China Agric. Univ. 10 (1) (2005) 48–51.
[23] B. Li, N. Wang, M. Wang, et al., In-field pineapple recognition based on monocular vision, Trans. Chin. Soc. Agric. Eng. 26 (10) (2010) 345–349.
[24] A. Guo, J. Xiong, D. Xiao, X. Zou, Computation of picking point of Litchi and its binocular stereo matching based on combined algorithms of Harris and SIFT, Trans. Chin. Soc. Agric. Mach. 46 (12) (2015) 11–17.
[25] C.G. Rafael, E.W. Richard, L.E. Steven, Digital Image Processing Using Matlab, Publish House of Ecectronics Industry, Beijing, 2012, pp. 152–158.
[26] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[27] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[28] L. Luo, X. Zou, C. Wang, X. Chen, Z. Yang, W. Situ, Recognition method for two overlaping and adjacent grape clusters based on image contour analysis, Trans. Chin. Soc. Agric. Mach. 48 (6) (2017) 15–22.
[29] C. Tomasi, R. Manduchi, Bilateral filtering for gray and colour images, Proceedings of the IEEE International Conference on Computer Vision, Bombay, India, 1998, pp. 839–846.
[30] C.G. Matas, J. Kittler, Robust detection of lines using the progressive probabilistic hough transform, Comput. Vision Image Understanding 78 (2000) 119–137.
[31] A. Gongal, S. Amatya, M. Karkee, et al., Sensors and systems for fruit detection and localization: a review, Comput. Electron. Agric. 116 (2015) 8–19.