Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

In-field citrus detection and localisation based on RGB-D image analysis

Guichao Lin [a], Yunchao Tang [b,**], Xiangjun Zou [a,*], Jinhui Li [a], Juntao Xiong [a]

[a] College of Engineering, South China Agricultural University, 483 Wushan Road, Guangzhou, 510642, China
[b] School of Urban and Rural Construction, Zhongkai University of Agriculture and Engineering, Guang Xin Road, Guangzhou, 510006, China

ARTICLE INFO

Article history:
Received 18 December 2018
Received in revised form
23 May 2019
Accepted 26 June 2019
Published online 11 July 2019

Keywords:
RGB-D image
Bayes classifier
Density clustering
Support vector machine
Citrus-harvesting robot

In-field citrus detection and localisation are highly challenging tasks due to varying illumination conditions, partial occlusion of citrus, and the colour variation of citrus at different stages of maturity. A reliable algorithm based on red-green-blue-depth (RGB-D) images was developed to detect and locate citrus in real, outdoor orchard environments for robotic harvesting. A depth filter and a Bayes-classifier-based image segmentation method were first developed to exclude as many backgrounds as possible. A density clustering method was then used to group adjacent points in the filtered RGB-D images into clusters, where each cluster represents a possible citrus. A colour, gradient, and geometry feature-based support vector machine classifier was trained to remove false positives. To test the method, a dataset with 506 RGB-D images was acquired in a citrus orchard on sunny and cloudy days. Results showed that the proposed algorithm was robust with an F1 score of 0.9197; the positioning errors in the x, y and z directions were 7.0 ± 2.5 mm, −4.0 ± 3.0 mm and 13.0 ± 3.0 mm, respectively, and the sizing error was −1.0 ± 4.0 mm. These excellent performance values demonstrate that the proposed method could be used to guide a citrus-harvesting robot.

1. Introduction

In China in 2016, the citrus planting area was 2.6 million hectares and the citrus yield was 36.2 million tons (Kong, Zeng, Xiong, Zhengliang, & Xia, 2018). Most citrus in China is hand-picked. Growing urbanisation and the ageing of the population has led to a labour shortage, and increased labour costs present a significant problem that may decrease the citrus production. To deal with this situation and improve the economic stability of the citrus industry, one alternative is to develop efficient automatic citrus-harvesting robots. The key problem in establishing a robotic harvesting system is fruit detection and localisation (Gongal, Amatya, Karkee, Zhang, & Lewis, 2015; He et al., 2017; Luo et al., 2016, 2018). In real,

* Corresponding author.
** Corresponding author.
E-mail addresses: ryan.twain@gmail.com (Y. Tang), xjzou1@163.com (X. Zou).
https://doi.org/10.1016/j.biosystemseng.2019.06.019
1537-5110/

Nomenclature

Symbols

(C_x, C_y, f_x, f_y)	Intrinsic parameters of the infrared camera
$\mathbf{C}_i$	RGB values of pixel i, defined as $(r_i, g_i, b_i)^T$
$\boldsymbol{K}$	Intrinsic parameter matrix of the infrared camera
N	Number of pixels in a given RGB-D image
N_k	Number of components of the kth GMM
$\boldsymbol{O}$	A set of cluster centres
$p(\mathbf{C}_i \mid y_k)$	Conditional probability for pixel i given y_k
$p(y_1 \mid C_i)$	Posterior probability of foreground given pixel i
$p(y_1)$, $p(y_2)$	Prior probabilities
$(\mathbf{R}, \mathbf{T})$	Rotation and translation matrixes between CCS and UCS
$\mathbf{S}_k$	The kth cluster
$\boldsymbol{X}_i$	3D coordinate of pixel i, defined as $(x_i, y_i, z_i)^T$
y_k	$k = 1,2$. y_1 and y_2 refer to the foreground and background, respectively
$\boldsymbol{\Omega}$	An RGB-D image
$\boldsymbol{\Omega}_c$	A subset of $\boldsymbol{\Omega}$
$\boldsymbol{\Omega}_d$	A subset of $\boldsymbol{\Omega}_c$
$\{\lambda_{j,k}, \boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k}\}$	Parameters of the jth component of the kth GMM, referring to the weight, mean, and covariance matrix, respectively
$\rho(\mathbf{X}_i)$	Local density of point $\mathbf{X}_i$
$\delta(\mathbf{X}_i)$	Distance of point $\mathbf{X}_i$
$(\delta_{min}, \rho_{min})$	Parameters used to identify cluster centre

Abbreviations

CCS	Camera coordinate system
CHT	Circular Hough transform
CNN	Convolutional neural networks
FPFH	Fast point feature histograms
GMM	Gaussian mixture model
GPU	Graphics processing unit
HOG	Histograms of oriented gradients
MEDE	Median error
MEDAD	Median absolute deviation
SVM	Support vector machine
TOF	Time of flight
UCS	User coordinate system
VA	Viewpoint angle feature
VFH	Viewpoint feature histogram

outdoor orchard environments, citrus detection and localisation is challenging due to varying illumination conditions such as strong sunlight and backlight, fruit variation in colour, and partial occlusion caused by leaves and branches.

Citrus fruit detection and localization have been studied extensively. Analysis of red-green-blue (RGB) images captured from low-cost colour cameras is a common approach (Kusumam, Krajnik, Pearon, Duckett & Cielniak, 2017). Kurtulmus, Lee, and Vardar (2011) slided a multi-scale sub-window along with an "eigenfruit" and a Gabor texture feature on an entire RGB image to detect citrus fruits. However, the detection precision was only 75.3%. Sengupta and Lee (2014) first conducted circular Hough transform (CHT) to detect possible fruits and then used a support vector machine (SVM) trained on texture features to exclude false positives. The algorithm detected only 80.4% of the citrus fruits on trees probably due to partial occlusion. Lu and Sang (2015) segmented image contours into smooth fragments, selected valid fragments based on their lengths, bending degrees and concavities, and then applied circle fitting to combine neighbouring fragments as single citrus fruits. To detect immature citrus, Zhao, Lee, and He (2016) used a red and blue chromatic map and a template matching algorithm to exclude backgrounds, followed by use of an SVM to detect true positives. This algorithm achieved a detection accuracy of 83%. A similar method was established by Zhuang et al. (2018) to recognise orchard citrus with a detection recall of 86%. Our previous work (Wang et al., 2018) used discrete wavelet transform, K-means clustering, and CHT to detect citrus with detection accuracy, false positive and miss rate of 85.6%, 11.8% and 14.4%, respectively. As can be seen from the above studies, the performance of citrus detection through RGB image analysis improved only slightly in the last decade. Recently, convolutional neural networks (CNN) have been successfully applied in some agricultural vision applications. For instance, Bargoti and Underwood (2017) deployed Faster R-CNN for apple and mango yield prediction with an F1 score larger than 0.9. Koirala, Walsh, Wang, and McCarthy (2019) used YOLO to detect mango fruits with an F1 score of 0.89. Although this approach was robust and effective for fruit detection, CNN requires an expensive graphics processing unit (GPU) to accelerate training and inferring, which would increase the cost of a vision system and limit the transfer of the executable codes to an embedded system.

As RGB-D devices that provide additional depth information have become affordable, there has been an ongoing interest in the development of RGB-D-based methods (Sa et al., 2017; Vitzrabin & Edan, 2016). As early as 1992, Benady and Miles (1992, pp. 92–7021) used structured light to obtain depth data for melon recognition. Jiménez, Ceres, and Pons (2000) developed a localisation system using an infrared laser range sensor, where characteristic primitives were extracted from both range and gray images and used to cast votes for the positions of fruits. Barnea, Mairon, and Ben-Shahar (2016) used the reflective effects of smooth-faced fruits to detect regions of interest and then applied a fixed-sized three-dimensional (3D) sliding-window along with an SVM classifier to identify sweet peppers. The detection precision was 55%, and the average running time was 197 s. Nguyen et al. (2016) presented a method for detecting and locating apples simultaneously wherein 3D points with red colour were first selected, then clustered by Euclidean clustering (Rusu, 2009). CHT and RANSAC (Fischler & Bolles, 1981) were used to eliminate the ambiguities in such clusters. A true positive of 88% for Gala trees and 81% for Fuji trees were reported, but unfortunately, this method is applicable only to red fruits. Sa et al. (2017) concatenated hue-saturation-value features and fast point feature histograms (FPFH) (Rusu,

Blodow, & Beetz, 2009) as the point feature, and utilised an SVM classifier to classify each point in the point could as a peduncle or pepper class. The mean average precision was 0.71, but the execution time was a little long (about 40 s). In another study, Kusumam, Krajnik, Pearon, Duckett and Cielniak (2017) utilised Euclidean clustering to group the broccoli point cloud into the broccoli head objects and backgrounds, and then applied a viewpoint feature histogram (VFH) (Rusu et al., 2009)-based SVM classifier to recognise the broccoli heads. A detection accuracy of 94.7% was obtained. Another similar scheme was developed by Tao and Zhou (2017). A region growing algorithm was first employed to cluster apple point clouds. Then for each cluster, an HSI-FPFH feature vector was extracted and classified by an SVM classifier to remove any falsities. A mean average precision of 0.87 was reported. Our research group (Lin, Tang, Zou, Xiong, & Fang, 2019) developed a detection pipeline comprising a colour filter, a region growing-based clustering method, a 3D shape detector and an SVM classifier. This pipeline can detect different types of fruits effectively but was a bit complicated.

As described above, RGB-D based methods can be potentially applied to harvesting robots. Current methods have used distance-based clustering algorithms, such as Euclidean clustering or region growing, to generate regions that may contain fruits. As adjacent citrus may have similar depth values, such distance-based clustering algorithms would group them into a single cluster and hence decrease overall detection performance. Therefore, a new clustering algorithm is required that can address the over-clustering problem of distance-based clustering algorithms. Additionally, to effectively exclude false-positive clusters, a discriminative point cloud descriptor should be investigated.

This work was an extension of our previous study (Lin et al., 2019), and the objective was to detect and locate citrus under real, outdoor orchard conditions to guide a citrus-harvesting robot to pick selectively and automatically. This method includes the following functions: (i) using a depth filter and a Bayes-classifier-based image segmentation method to exclude non-relevant points in RGB-D images, which improves the overall computation efficiency; (ii) using a novel density clustering method to group adjacent points into a set of clusters where each cluster represents a possible fruit; (iii) using an SVM classifier trained on colour, gradient and geometry features to remove false positives; and (iv) estimating citrus position and size.

2. Materials and methods

2.1. Sensor system and image acquisition

A Kinect V2 sensor with an RGB and an infrared (IR) camera (Microsoft Inc.) was used to capture RGB-D images. The IR camera uses time of flight (TOF) technology to obtain depth information with precision below 2 mm reported by Diana and Livio (2015). The Kinect V2 sensor is able to generate an RGB image of 1920 × 1080 pixels and a depth image of 512× 424 pixels simultaneously at a frame rate of 30 fps. Because the resolutions of the RGB and depth images are different, they must be aligned before applications. To do this, we used the *MapDepthFrameToCameraSpace* function of the Kinect software development kit and adjusted the RGB image to match the depth image such that every pixel in the depth image corresponds to an RGB value.

A total of 506 RGB-D images were collected in a citrus orchard (cultivar "emperor citrus") in Guangzhou, China on December 4, 2017 and November 11, 2018. Table 1 lists the details of the image dataset, and Fig. 1 shows our experimental site. The Kinect V2 sensor was placed approximately 600 mm in front of a citrus tree during image acquisition. No artificial light source was used, meaning that the captured images contained different degrees of illumination. To train and test the proposed method, about 80% of the RGB-D images in the dataset were randomly selected as the training set and the remaining images made up the test set.

2.2. Methodology

This section details the proposed fruit detection and localisation method. First, RGB-D images are segmented by a depth filter and a Bayes classifier trained on RGB features. Then, density clustering (Rodriguez & Laio, 2014) is used to group the filtered RGB-D image into a set of clusters. Subsequently, all clusters are evaluated by an SVM classifier. Finally, the 3D position and diameter of the citrus fruits are estimated.

2.2.1. Depth filtering

Let $\boldsymbol{\Omega} = \{(\mathbf{C}_i, \mathbf{X}_i)\}_{i=1}^{N}$ denote an RGB-D image. Here, $\mathbf{C}_i=(r_i, g_i, b_i)^T$ and $\mathbf{X}_i=(x_i, y_i, z_i)^T$ represent the RGB value and 3D coordinate of pixel i, respectively; N is the number of pixels. X_i is calculated as follows:

$$\begin{cases} z_i = I_d(U_i, V_i) \\ x_i = z_i(U_i - C_x)/f_x \\ y_i = z_i(V_i - C_y)/f_y \end{cases} \quad (1)$$

where $(U_i, V_i)^T$ is the image coordinate of pixel i; I_d is the depth image; (C_x, C_y, f_x, f_y) are the intrinsic parameters of the IR camera.

$\boldsymbol{\Omega}$ is filtered by thresholding its z values. Specifically, if the depth value of a pixel is larger than a given threshold, this pixel is excluded. The threshold was set to 1800 mm in experiments as generally a citrus-harvesting robot could not pick the fruits outside such a distance.

Table 1 – Details of the image dataset.

Image acquisition time	Weather	Number of RGB-D images	Number of fruits
9:00 to 15:00, December 4, 2017	Sunny	209	1198
11:00 to 15:00, November 11, 2018	Sunny to cloudy	297	1991

Fig. 1 – Image collection scene.

2.2.2. *Bayes-classifier-based image segmentation*

The intent of image segmentation is to exclude a large number of insignificant points while focussing on the crucial points that are similar to the target fruits in colour. Inspired by Song et al. (2014) and Lin et al. (2019), who used a naïve Bayes classifier to identify points of interest, we generalised their method by dealing with RGB values directly without a problem-dependent colour transformation process and assuming that the RGB values of either the background or foreground are dependent.

To model the conditional probability for pixel i given the foreground or background, we use two Gaussian mixture models (GMM) to describe the colour data distribution (Rother, Kolmogorov, & Blake, 2004):

$$p(\mathbf{C}_i|y_k) = \sum_{j=1}^{N_k} \lambda_{j,k} \frac{1}{\sqrt{(2\pi)^D |\mathbf{\Sigma}_{j,k}|}} \exp\left(\frac{-(\mathbf{C}_i - \boldsymbol{\mu}_{j,k})^T \mathbf{\Sigma}_{j,k}^{-1} (\mathbf{C}_i - \boldsymbol{\mu}_{j,k})}{2}\right) \tag{2}$$

where $k = 1,2$, y_1 and y_2 refer to the foreground and background, respectively, D is the dimension of colour data equal to 3, and N_k is the number of components of the kth GMM. $\{\lambda_{j,k}, \boldsymbol{\mu}_{j,k}, \mathbf{\Sigma}_{j,k}\}$ denotes the parameters of the jth component of the kth GMM, where $\lambda_{j,k}$ is the weight, $\boldsymbol{\mu}_{j,k}$ is the mean, and $\mathbf{\Sigma}_{j,k}$ is the covariance matrix. The covariance matrix can reveal the correlation of the RGB values.

The parameters of two GMMs can be learned by the expectation maximisation algorithm (Dempster, Laird, & Rubin, 1977) from training samples. Note that: (i) N_1 and N_2 were all set to 5 in experiments as suggested by Rother et al. (2004); and (ii) each pixel in the training set was manually labelled as a foreground or background training sample using the *Image Labeler* app in MATLAB.

Using the Bayes formula, the posterior probability for foreground given pixel i is computed as follows:

$$p(y_1|\mathbf{C}_i) = \frac{p(\mathbf{C}_i|y_1)p(y_1)}{(\mathbf{C}_i|y_1)p(y_1) + (\mathbf{C}_i|y_2)p(y_2)} \tag{3}$$

where $p(y_1)$ and $p(y_2)$ learned from training samples are the prior probabilities.

Eq. (3) is applied to every pixel in $\boldsymbol{\Omega}$ to obtain a probability map that can be segmented by a threshold. Larger thresholds may result in over-segmentation, while smaller thresholds may lead to under-segmentation. In the experiments, this threshold was set to 0.2 because such a threshold could remove most background and avoid removing the target fruits. After segmentation, a subset of $\boldsymbol{\Omega}$, denoted as $\boldsymbol{\Omega}_c$, is obtained. Figure 2 shows an example of the Bayes-classifier-based segmentation method.

2.2.3. *Density clustering*

In this step, our purpose is to cluster points in $\boldsymbol{\Omega}_c$ to obtain individual fruits. The density clustering algorithm proposed by Rodriguez and Laio (2014) reveals cluster centres by finding points with both relatively large density and distance. Compared to other methods, such as *K*-means or *K*-medoids, it does not require prior knowledge of the number of clusters that is difficult to estimate. We utilise it for clustering in the proposed scheme.

The local density of point $\mathbf{X}_i$ is defined as follows:

$$\rho(\mathbf{X}_i) = \sum i \neq j,\ j = 1, \ldots, |\boldsymbol{\Omega}_c| \exp\left(-d(\mathbf{X}_i, \mathbf{X}_j)^2 \Big/ 2d_c^2\right) \tag{4}$$

where $|\boldsymbol{\Omega}_c|$ is the number of points in $\boldsymbol{\Omega}_c$, $d(\mathbf{X}_i, \mathbf{X}_j)$ refers to the Euclidean distance between $\mathbf{X}_i$ and $\mathbf{X}_j$, and d_c is the standard variance which is recommended to be 2% of d in an ascending order (Rodriguez & Laio, 2014). The distance of point $\mathbf{X}_i$ is given as:

$$\delta(\mathbf{X}_i) = \min_{\rho(\mathbf{X}_j) \geq \rho(\mathbf{X}_i), i \neq j, j = 1, \ldots, |\boldsymbol{\Omega}_c|} d(\mathbf{X}_i, \mathbf{X}_j) \tag{5}$$

Density and distance are two characteristics of each point. Cluster centres, points with large δ and high ρ, can be easily spotted on the decision graph where δ is a function of ρ, as shown in Fig. 3. Rodriguez and Laio (2014) interactively drew a rectangle (see dotted rectangle box in Fig. 3) on the decision graph inside which all points are cluster centres. Although effective, this strategy is inefficient during harvesting. Here, to enhance computational efficiency, the coordinates of the lower left corner of the rectangle, (δ_{min} ρ_{min}), are fixed and a set of cluster centres is obtained as follows:

$$\mathbf{O} = \{\mathbf{X}_i | \rho(\mathbf{X}_i) \geq \rho_{min} \text{ and } \delta(\mathbf{X}_i) \geq \delta_{min}\} \tag{6}$$

If δ_{min} and ρ_{min} are too small, some fruits may be split into multiple clusters; if δ_{min} and ρ_{min} are too large, fruits and leaves that are close to each other spatially may be merged into a single cluster. In short, selecting the proper δ_{min} and ρ_{min} is very important, which were set to 20 and 3, respectively, in our experiments by trial and error. After obtaining the cluster centres, the next step is to assign each point to the cluster to which the closest point of higher density belongs (Rodriguez & Laio, 2014). The pseudo code of density clustering is given in Algorithm 1.

Input: A RBG-D image $\boldsymbol{\Omega}_c$

Output: A set of cluster centres $\boldsymbol{O}$ and a set of clusters $\{\boldsymbol{S}_k\}$, k=1,…,$|\boldsymbol{O}|$

Parameters: Two thresholds for selecting the cluster centre δ_{min} and ρ_{min}

Code:

Compute $d(\boldsymbol{X}_i, \boldsymbol{X}_j)$, i, j=1,…,$|\boldsymbol{\Omega}_c|$, and d_c

Compute $\rho(\boldsymbol{X}_i)$ and $\delta(\boldsymbol{X}_i)$, i=1,…,$|\boldsymbol{\Omega}_c|$, using Eq. (4) and (5)

Find cluster centers $\boldsymbol{O}$ using Eq. (6), and set $\boldsymbol{S}_k$ = null, k=1,…,$|\boldsymbol{O}|$

Sort $\rho(\boldsymbol{X}_i)$ in descending order, and record the subscript number of $\boldsymbol{X}$ as $\{q_i\}_{i=1}^{|\boldsymbol{\Omega}_c|}$

Set array $Label[|\boldsymbol{\Omega}_c|] = \text{null}$

for i:=1 to $|\boldsymbol{\Omega}_c|$

  if X_{q_i} equals the kth element of $\boldsymbol{O}$

    set $Label[q_i] = k$

    add $\boldsymbol{X}_{q_i}$ in $\boldsymbol{S}_k$

  else

    find q_j which satisfies $j < i$, and $d\left(\boldsymbol{X}_{q_j}, \boldsymbol{X}_{q_i}\right) = \delta\left(\boldsymbol{X}_{q_i}\right)$

    set $Label[q_i] = Label[q_j]$

    add $\boldsymbol{X}_{q_i}$ in $\boldsymbol{S}_{Label[q_i]}$

  end if

end for

Algorithm 1. Density clustering

Figure 4 shows an example illustrating the results of density clustering. As can be seen, each cluster represents an independent region in the 3D space, which may be a leaf, a fruit, or other object. Therefore, it is important to exclude false clusters.

2.2.4. *SVM classification*

To remove false clusters, we first extract a feature vector for each cluster and then use an SVM classifier to identify true and false as described below.

2.2.4.1Feature extraction. A colour-, gradient- and geometry-based feature descriptor is investigated. The colour feature is the mean of the RGB values of the cluster. The gradient feature is the histograms of oriented gradients (HOG) (Dalal & Triggs, 2005). Specifically, we scale the RGB component of the cluster to an image of 32×32 pixels, divide the image into 16 cells of size 8×8, and obtain an HOG feature vector of size 324. The geometry feature is the VFH feature. VFH is a combination of the viewpoint angle feature (VA) and FPFH. VA is calculated by first computing the angle between the normal of each point and a viewpoint vector and then binning the angles into a 12-dimensional bin. FPFH is determined by first computing the roll, pitch and yaw angles of each point pair and then encoding all angle triples into a 27-dimensional bin. The use of colour, gradient and geometry features result in a 366-dimensional feature vector for each cluster.

2.2.4.2Classification. To assess whether a cluster represents a true fruit, an SVM classifier is applied to the feature vector. The SVM classifier should be trained before applications. In the training phase, both positive and negative samples were generated from the training set by the proposed method, and the 10-fold cross validation and grid search method (Hsu, Chang, & Lin, 2003) were used to determine the SVM parameters. Figure 5 shows a classification example of the SVM classifier.

2.2.5. *Localisation*

The final step of the proposed method is to estimate the position and size of each true-positive cluster. The citrus position is computed by calculating the mean of the 3D coordinates of the cluster. To determine the citrus size, we utilised the method presented by Kusumam, Krajnik, Pearson, Duckett, and Cielniak (2017) who measured the length of the 3D coordinates of the cluster in the x-axis direction as the fruit size.

3. Results

To evaluate the detection, localisation and sizing performance of the proposed method, quantitative experiments were performed.

3.1. *Evaluation of fruit detection*

The proposed method was evaluated using the test set mentioned in Section 2.1. Two modified versions of the proposed method were compared. The first version replaced density clustering by region growing (Xia, Chen, & Aggarwal, 2011) to obtain a pool of clusters from the RGB-D depth image, a method that is widely used to generate multiple homogeneous regions. The second version used Euclidean clustering (Rusu, 2009) instead of density clustering to

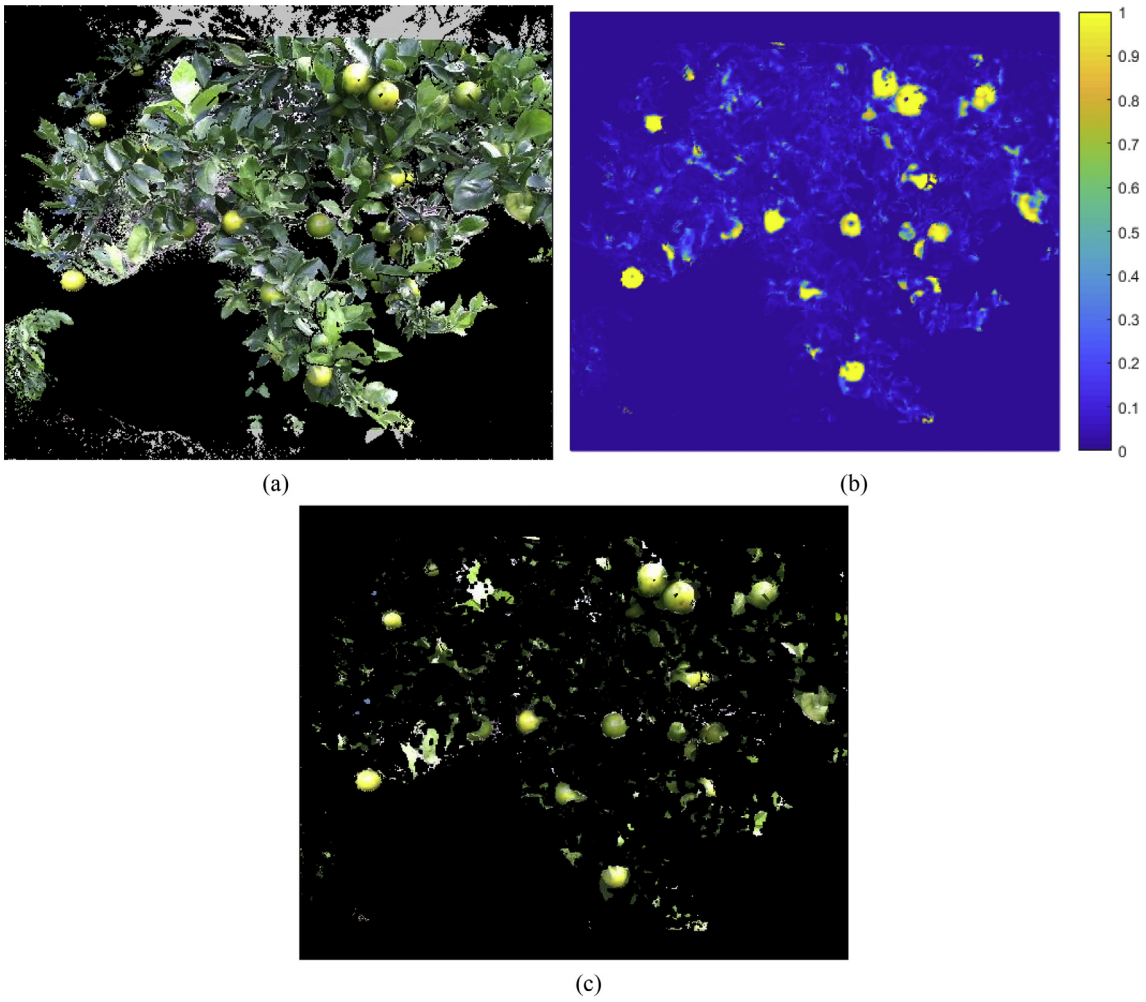

Fig. 2 – Example illustrating the result of the Bayes-classifier-based segmentation method: (a) RGB-D image after depth filtering, (b) probability map and (c) segmentation result.

generate a set of clusters from the point cloud, a method that has been successfully applied to segment apple and broccoli head point clouds (Kusumam, Krajnik, Pearon, Duckett & Cielniak, 2017; Nguyen et al., 2016). These two compared algorithms were termed "RegionGrowing" and "EuclidClustering", respectively, for simplicity. Additionally, two additional algorithms were deployed for comparison purposes: our

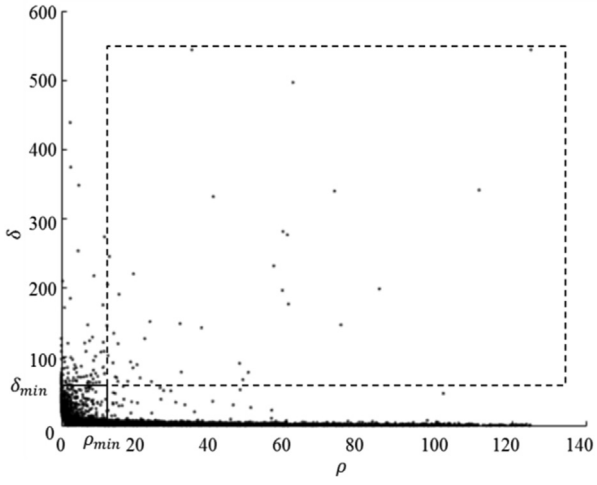

Fig. 3 – Decision graph of Fig. 2c.

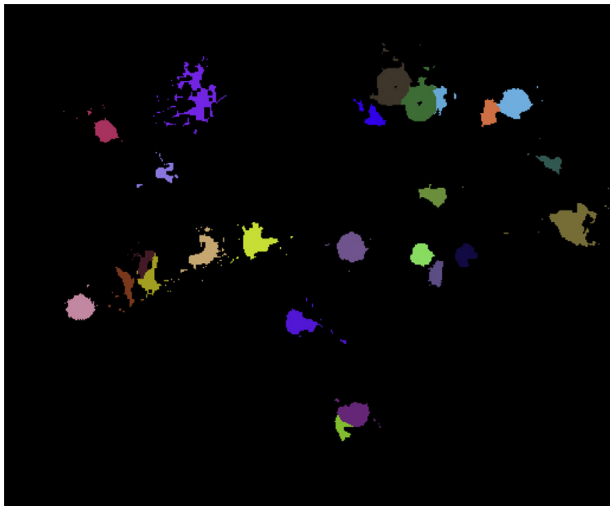

Fig. 4 – Clustering result of the density clustering on Fig. 2c; each cluster is marked with a random colour.

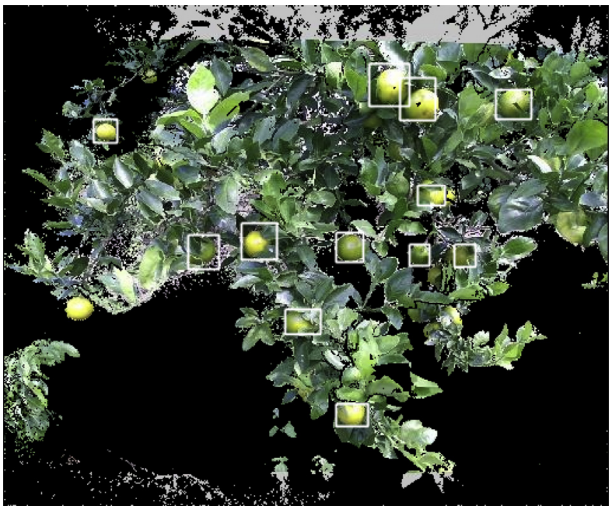

Fig. 5 – Example of the results from applying the SVM classifier to the image in Fig. 4.

previous method (Lin et al., 2019) and a state-of-the-art CNN, YOLOv3 (Redmon, J. & Farhadi, 2018). It should be noted that (i) YOLOv3 was implemented by using a publicly available code (https://github.com/pjreddie/darknet), while the other algorithms were programed in MATLAB 2017b; and (ii) all the codes ran on a computer with Windows 10 system, a 16GB RAM, an Intel i7 CPU, and an NVIDIA GeForce GTX 1060 6GB GPU (only YOLOv3 used GPU).

Precision, recall and F1 score were used to evaluate the performance of the above detectors. Precision is calculated as the ratio of the number of true positives to the number of detections in the images. Recall is calculated as the ratio of the number of true positives to the number of fruits in the images. F1 score is a balance between precision and recall, and is equal to $2 \times (Precision \times Recall)/(Precision + Recall)$. The larger the F1 score, the better the performance of the detector. Table 2 lists the detection results over the test set. The precision, recall and F1 score of the developed pipeline were 0.9839, 0.8634 and 0.9197, respectively, all higher than RegionGrowing, EuclidClustering, or our previous method. The main reasons why the developed method is superior to RegionGrowing, EuclidClustering and our previous method are that (i) density clustering could better group each fruit into a cluster, whereas region growing and Euclidean clustering may classify some adjacent fruits into a single cluster, as shown in Fig. 6; and (ii) our previous method uses an iterative 3D shape detector to extract spherical objects from the output of region growing to avoid grouping neighbouring fruits as a single fruit, which would fail if the output region of region growing contains a large percentage of noise.

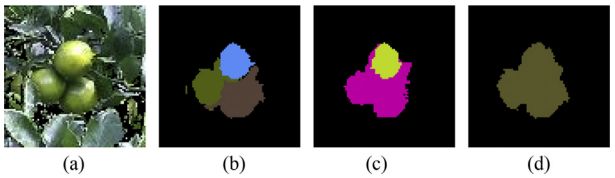

Fig. 6 – Visual example showing the clustering results of different clustering methods: (a) input image, (b) density clustering, (c) region growing, and (d) Euclidean clustering. Each region is marked with a random colour.

Interestingly, the proposed algorithm only slightly underperformed YOLOv3, and this performance difference may be negligible for agricultural applications. Additionally, the proposed algorithm was easier to implement and did not require use of a GPU to accelerate training and inferring. Therefore, the proposed algorithm was competitive to YOLOv3 and could be used by a citrus harvesting robot.

Some detection examples of the proposed method on the test set are shown in Fig. 7. These results indicate that the proposed method can successfully detect citrus fruits under variable lighting conditions in the field.

3.2. *Evaluation of fruit localisation*

The precision of positioning and sizing is important for a harvesting robot, as the harvesting task may fail if the precision is too poor or unstable. An in-field experiment was conducted to ascertain the precision of the proposed method.

To measure the ground-truth position of a detected citrus, a right-hand user coordinate system (UCS) was defined on a calibration board (see Fig. 8a). Obviously, the ground-truth citrus position on UCS can be easily measured using tools such as a Vernier caliper (see Fig. 8b–d). Given a rotation matrix $\boldsymbol{R}$ and a translation matrix $\boldsymbol{T}$ between the camera coordinate system (CCS) and UCS (see the Appendix for details), an estimated position can be transformed to UCS by application of the equation $\boldsymbol{C}^{UCS} = \boldsymbol{R}\boldsymbol{C}^{CCS} + \boldsymbol{T}$. In experiments, the ground-truth position and size of 80 fruits were measured. The errors between the ground-truth and estimated positions (or sizes) were calculated. The median error (MEDE) and median absolute deviation (MEDAD) were computed and used to represent the positioning (or sizing) accuracy and precision, respectively, using the following equations (Kusumam et al., 2017):

$$\mathrm{MEDE} = median(\{e_i\}) \tag{8}$$

$$\mathrm{MEDAD} = median(\{|e_i - \mathrm{MEDE}|\}) \tag{9}$$

where *median*() is the median operation.

Table 2 – Precision, recall and F1 score of different detection methods over the test set.

Method	# Images	# Fruits	# True positives	# False positives	Precision	Recall	F1
Proposed	101	637	550	9	0.9839	0.8634	0.9197
RegionGrowing	101	637	351	22	0.9400	0.5510	0.6950
EuclidClustering	101	637	331	30	0.9169	0.5196	0.6633
Our previous method	101	637	347	11	0.9693	0.5447	0.6975
YOLOv3	101	637	559	2	0.9964	0.8776	0.9332

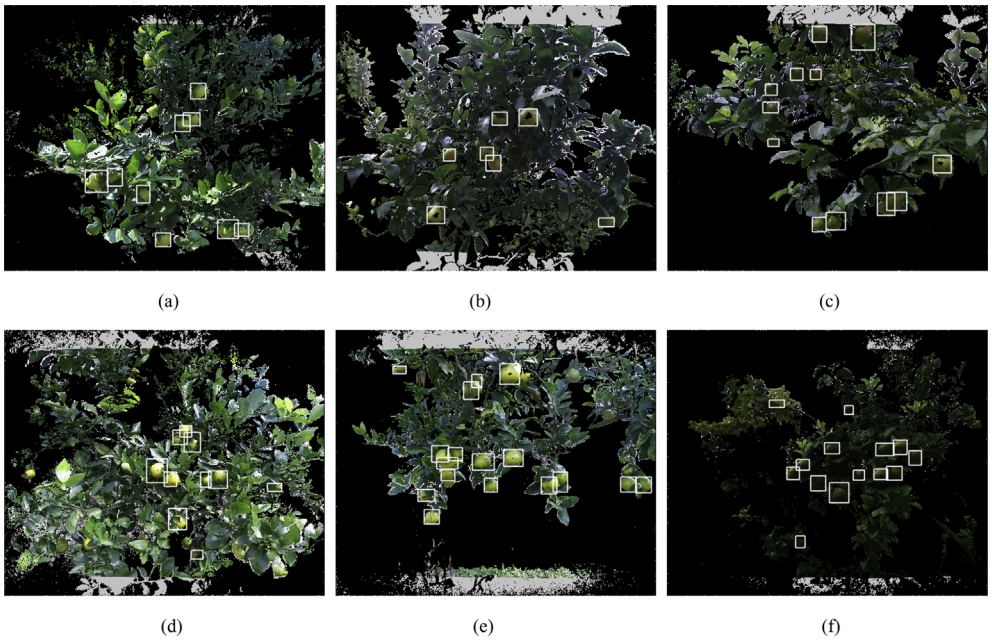

Fig. 7 – Detection examples of the proposed method from the test set.

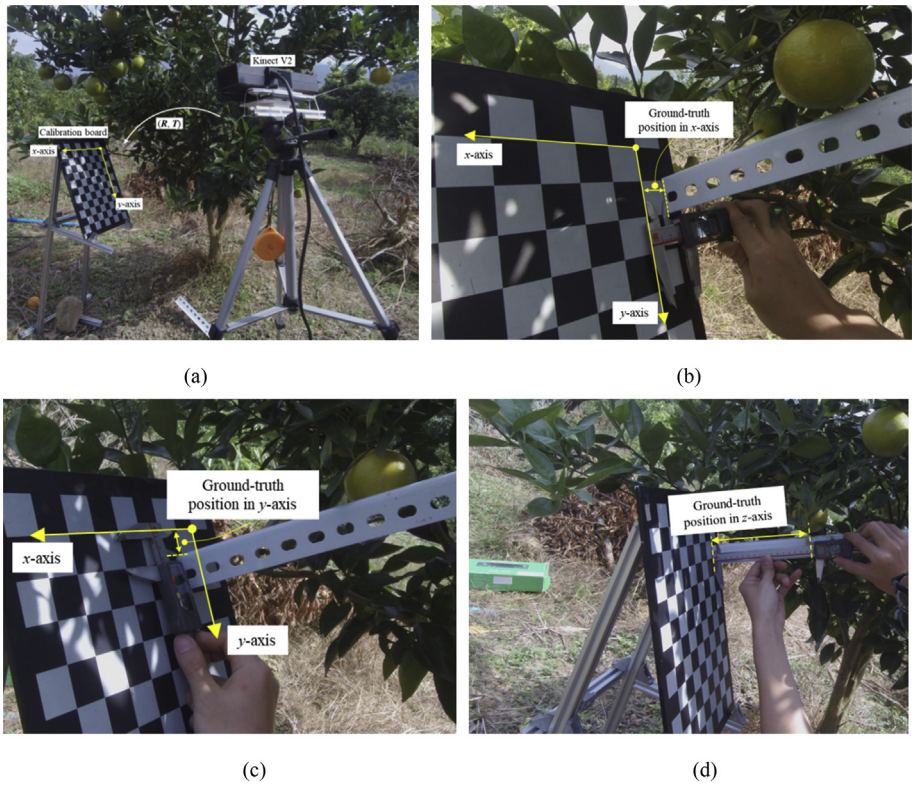

Fig. 8 – Illustration of coordinate transformation (a) and the measurement of the ground-truth position of a detected fruit in the *x*-axis (b), *y*-aix (c), and *z*-axis (d) directions.

The results are shown in Table 3 and reveal that (1) the accuracies (i.e. biases): in the *x*-, *y*- and *z*-axis were 7.0 mm, −4.0 mm and 13.0 mm, respectively, probably resulting from the systematic error of the Kinect V2 sensor and the calculation error of the pose between CCS and UCS; if the biases in the *x*-, *y*- and *z*-axis directions were compensated, the positioning precision in fact was reasonable (Zou et al., 2016), and thus the proposed method could guide robotic

Table 3 – Statistics of the positioning and sizing errors of 80 fruits.

	Error in x-axis	Error in y-axis	Error in z-axis	Diameter error
MEDE (mm)	7.0	−4.0	13.0	−1.0
MEDAD (mm)	2.5	3.0	3.0	4.0

harvesting in practical applications; (2) both the MEDE and MEDAD values of citrus diameter were small, indicating that our method allowed precise size estimation.

3.3. *Evaluation of algorithm running time*

The algorithm's running time can affect the picking cycle of a harvesting robot. To determine the running time, the clock function in MATLAB 2017b was used. Experiments over the test set showed an average execution time for the proposed algorithm equal to 7.92 s per image, with 6.30 fruits detected on average in an image; hence, detecting and locating a citrus required 1.25 s, on average. Clearly, the proposed method is time-consuming and should be optimised further.

4. Discussion

This paper investigates a citrus detection and localisation algorithm using RGB-D image analysis. Experiments show that the F1 score of the proposed algorithm was 0.9197, meeting the robustness requirement of a citrus-harvesting robot. Nevertheless, the proposed detection pipeline can still be improved. First, as observed in experiments, adjacent fruits were sometimes grouped together, reducing detection recall. One possible solution is to incorporate curvature information in density clustering, as the curvature of the area where two adjacent fruits touch each other is obviously larger than other areas. Second, under strong lighting conditions, some true-positive clusters were classified as false positives by the SVM classifier. A possible explanation is that the TOF-based sensor may suffer some interference in strong sunlight, lowering the depth precision (Wang, Walsh, & Verma, 2017); consequently, the geometry component of the extracted features is noisy and thus indiscriminative. One strategy to counter this would be to use a light shield to block strong sunlight (Nguyen et al., 2016), but this would complicate the harvesting task.

The positioning errors in the x, y and z directions were 7.0 ± 2.5 mm, −4.0 ± 3.0 mm and 13.0 ± 3.0 mm, respectively, quite small values that can meet the requirement of a citrus-harvesting robot. However, these deviations can be further reduced. One potential solution is to apply the least squares spherical fitting method to fit a sphere model for each true-positive cluster and use the sphere centre position as the fruit position. This approach can precisely determine the centre position of a partial point cloud, though it requires more computations. Another approach is to use a calibration board with relatively high precision to reduce the coordinate transformation error.

The proposed method needed 1.25 s to locate a single citrus fruit, which may be too slow for efficient harvest. Uniform sampling can be used to downsample the RGB-D images to reduce computational costs. This would lower the local density of each point, so the density parameter (i.e., ρ_{min}) of density clustering would need to be adjusted accordingly.

5. Conclusions

To detect and locate citrus fruits in real, outdoor orchard environments, we developed an algorithm framework that consisted of the following steps: (i) depth filtering, (ii) image segmentation, (iii) point cloud clustering, (iv) point cloud classification and (v) fruit localisation. Field tests were performed to evaluate the performance of the proposed method, and the following conclusions were obtained:

(i) Under changing lighting conditions, the precision, recall and F1 score of the proposed algorithm were 0.9839, 0.8634 and 0.9197, respectively, indicating that the proposed algorithm is robust to detect in-field citrus.
(ii) The positioning accuracies in the x, y and z directions were 7.0 mm, −4.0 mm and 13.0 mm, respectively, and the positioning precisions were 2.5 mm, 3.0 mm and 3.0 mm, respectively. This is sufficient positioning accuracy and precision for guidance of a citrus-harvesting robot to pick fruit.
(iii) Detecting and locating a citrus fruit required 1.25 s, on average.

Overall, the proposed method showed robust ability for the detection of citrus fruits in outdoor field conditions. The positioning and sizing precision values of the proposed method were reasonable for robotic harvesting. However, the execution time was unsatisfactory. Therefore, future work should focus on improving the efficiency of the proposed method for practical use.

Acknowledgments

We want to thank Jie Wang and Kuangyu Huang for their assistance in collecting RGB-D images and conducting the localisation experiment. This work was funded by a grant from the National Natural Science Foundation of China (No.31571568).

Appendix. Algorithm for estimating the pose between CCS and UCS

To estimate the pose (i.e., a rotation matrix **R** and a translation matrix **T**) between CCS and UCS, a set of pixel coordinates of the corners on the calibration board, defined as $\boldsymbol{p}$, is first extracted from an aligned RGB image by implementing the *detectCheckerboardPoints* function in MATLAB. Then, the corresponding 3D coordinates of $\boldsymbol{p}$, defined as $\boldsymbol{q}$, in UCS are extracted manually by counting the number of the uniform-

size checkerboard patterns. On this basis, a perspective-*n*-point problem (P*n*P) is defined as follows:

$$cost = \sum_i \left| s_i \begin{bmatrix} \boldsymbol{p}_i \\ 1 \end{bmatrix} - \boldsymbol{K}[\boldsymbol{R}\ \boldsymbol{T}] \begin{bmatrix} \boldsymbol{q}_i \\ 1 \end{bmatrix} \right|^2$$

where $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ are the i'th point in $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively; s_i is a scale factor; $\boldsymbol{K}$ is the intrinsic parameter matrix of the IR camera. This P*n*P problem could be solved using the method proposed by Lepetit, Moreno-Noguer, and Fua (2009), thus obtaining an optimal ($\boldsymbol{R}$, $\boldsymbol{T}$).

REFERENCES

Bargoti, S., & Underwood, J. P. (2017). Deep fruit detection in orchards. In *In 2017 IEEE international conference on robotics and automation (ICRA)* (pp. 3626–3633).

Barnea, E., Mairon, R., & Ben-Shahar, O. (2016). Colour-agnostic shape-based 3d fruit detection for crop harvesting robots. *Biosystems Engineering, 146*, 57–70.

Benady, M., & Miles, G. E. (1992). Locating melons for robotic harvesting using structured light. *American Society of Agricultural Engineers Society for Engineering in Agricultural, Food, and Biology*, 92–7021.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition* (pp. 886–893). Los Alamitos, USA: IEEE Computer Society Press.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society. Series B (Methodological), 39*(1), 1–38.

Diana, P., & Livio, P. (2015). Calibration of Kinect for Xbox one and comparison between the two generations of Microsoft sensors. *Sensors, 15*(11), 27569–27589.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Gongal, A., Amatya, A., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture, 116*(C), 8–19.

He, Z. L., Xiong, J. T., Lin, R., Zou, X., Tang, L. Y., Yang, Z. G., et al. (2017). A method of green litchi recognition in natural environment based on improved LDA classifier. *Computers and Electronics in Agriculture, 140*, 159–167.

Hsu, C., Chang, C., & Lin, C. (2003). *A practical guide to support vector classification*. Technical report. Department of Computer Science, National Taiwan University http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Jiménez, A. R., Ceres, R., & Pons, J. L. (2000). A vision system based on a laser range-finder applied to robotic fruit harvesting. *Machine Vision and Applications, 11*(6), 321–329.

Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. *Precision Agriculture*. https://doi.org/10.1007/s11119-019-09642-0.

Kong, W., Zeng, Z., Xiong, W., Zhengliang, W. U., & Xia, R. (2018). Production forecast of citrus in China and production and marketing situation of citrus in Chongqing in 2016 production season. *Asian Agricultural Research, 10*(02), 16–19+31.

Kurtulmus, F., Lee, W. S., & Vardar, A. (2011). Green citrus detection using 'eigenfruit', color and circular Gabor texture features under natural outdoor conditions. *Computers and Electronics in Agriculture, 78*(2), 140–149.

Kusumam, K., Krajnik, T., Pearson, S., Duckett, T., & Cielniak, G. (2017). 3D-vision based detection, localization, and sizing of broccoli heads in the field. *Journal of Field Robotics, 34*, 1505–1518.

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision, 81*(2), 155–166.

Lin, G., Tang, Y., Zou, X., Xiong, J., & Fang, Y. (2019). Color, depth, and shape based 3D fruit detection. *Precision Agriculture*. https://doi.org/10.1007/s11119-019-09662-w.

Luo, L., Tang, Y., Lu, Q., Chen, X., Zhang, P., & Zou, X. (2018). A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard. *Computers in Industry, 99*, 130–139.

Luo, L., Tang, Y., Zou, X., Wang, C., Zhang, P., & Feng, W. (2016). Robust grape cluster detection in a vineyard by combining the Adaboost framework and multiple color components. *Sensors, 16*(12), 2098.

Lu, J., & Sang, N. (2015). Detecting citrus fruits and occlusion recovery under natural illumination conditions. *Computers and Electronics in Agriculture, 110*, 121–130.

Nguyen, T. T., Vandevoorde, K., Wouters, N., Kayacan, E., Baerdemaeker, J. G. D., & Saeys, W. (2016). Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosystems Engineering, 146*, 33–44.

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An incremental improvement*. ArXiv Preprint ArXiv:1804.02767.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science, 344*(6191), 1492–1496.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG), 23*(3), 309–314.

Rusu, R. B. (2009). *Semantic 3d object maps for everyday manipulation in human living environment*. PhD thesis. Germany: Computer Science Department, Technische Universit€at Mü nchen.

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 3212–3217).

Sa, I., Lehnert, C., English, A., Mccool, C., Dayoub, F., Upcroft, B., et al. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting - combined colour and 3D information. *IEEE Robotics & Automation Letters, 3*(99), 588–595.

Sengupta, S., & Lee, W. S. (2014). Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. *Biosystems Engineering, 117*(1), 51–61.

Song, Y., Glasbey, C. A., Horgan, G. W., Polder, G., Dieleman, J. A., & van der Heijden, G. W. A. M. (2014). Automatic fruit recognition and counting from multiple images. *Biosystems Engineering, 118*(1), 203–215.

Tao, Y., & Zhou, J. (2017). Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers and Electronics in Agriculture, 142*, 388–396.

Vitzrabin, E., & Edan, Y. (2016). Adaptive thresholding with fusion using a RGBD sensor for red sweet-pepper detection. *Biosystems Engineering, 146*, 45–56.

Wang, C., Lee, W. S., Zou, X., Choi, D., Gan, H., & Diamond, J. (2018). Detection and counting of immature green citrus fruit based on the local binary patterns (LBP) feature using illumination-normalized images. *Precision Agriculture, 19*(6), 1062–1083.

Wang, Z., Walsh, K., & Verma, B. (2017). On-tree mango fruit size estimation using rgb-d images. *Sensors, 17*(12), 20170154.

Xia, L., Chen, C. C., & Aggarwal, J. K. (2011). Human detection using depth information by Kinect. *Computer Vision and Pattern Recognition, 85*, 15–22.

Zhao, C. Y., Lee, W. S., & He, D. (2016). Immature green citrus detection based on color feature and sum of absolute transformed difference (SATD) using color images in the citrus grove. *Computers and Electronics in Agriculture, 124*, 243–253.

Zhuang, J., Luo, S., Hou, C., Tang, Y., He, Y., & Xue, X. (2018). Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. *Computers and Electronics in Agriculture, 152*, 64–73.

Zou, X., Ye, M., Luo, C., Xiong, J., Luo, L., Wang, H., et al. (2016). Fault-tolerant design of a limited universal fruit-picking end-effector based on vision positioning error. *Applied Engineering in Agriculture, 32*(1), 5–18.