Precision Agriculture (2020) 21:1–17
https://doi.org/10.1007/s11119-019-09654-w

Color-, depth-, and shape-based 3D fruit detection

Guichao Lin[1,2] · **Yunchao Tang**[3] · **Xiangjun Zou**[1] · **Juntao Xiong**[1] · **Yamei Fang**[1]

Published online: 30 March 2019

Abstract
A novel detection algorithm based on color, depth, and shape information is proposed for detecting spherical or cylindrical fruits on plants in natural environments and thus guiding harvesting robots to pick them automatically. A probabilistic image segmentation method is first presented to segment a red–green–blue image as a binary mask. Multiplied by this mask, a filtered depth image is obtained. Region growing, a region-based image segmentation method, is then applied to group the depth image into multiple clusters. Each cluster represents a fruit, leaf, or branch that is later transformed into a point cloud. Next, a 3D shape detection method based on M-estimator sample consensus, a model parameter estimator, is employed to detect potential fruits from each point cloud. Finally, an angle/color/shape-based global point cloud descriptor (GPCD) is developed to extract a feature vector for an entire point cloud, and a support vector machine classifier trained on the GPCD features is used to exclude false positives. Pepper, eggplant, and guava datasets were captured in the field. For the pepper, eggplant, and guava datasets, the detection precision was 0.864, 0.886, and 0.888, and the recall was 0.889, 0.762, and 0.812, respectively. Experiments revealed that the proposed algorithm was universal and robust and hence applicable to an agricultural harvesting robot.

Keywords Fruit detection · Image segmentation · M-estimator sample consensus · Support vector machine · Region growing

✉ Yunchao Tang
ryan.twain@gmail.com

✉ Xiangjun Zou
xjzou1@163.com

[1] Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China

[2] College of Mechanical and Automotive Engineering, Chuzhou University, 1 Huifeng West Road, Chuzhou 239000, China

[3] College of Urban and Rural Construction, Zhongkai University of Agriculture and Engineering, Guangxin Road, Guangzhou 510225, China

Introduction

As China's agricultural labor force continues to decline despite the increasing cultivated areas of fruits and vegetables, agricultural harvesting is facing a potential labor shortage (Xiang et al. 2014; Zou et al. 2012, 2016). To reduce the labor burden, automatic harvesting is necessary. An important aspect of autonomous harvesting robots is how to detect fruits robustly under natural conditions where challenges exist. Such challenges include cluttered backgrounds, varying illumination, low contrast between leaves and fruits, and fruit scale and rotation changes. This study focuses on exploring a general method to detect spherical or cylindrical fruits in the fields, because most fruits are spherical or cylindrical.

In the past few decades, many color-based detection methods have been investigated (Bulanon et al. 2003; Wachs et al. 2010; Luo et al. 2016; Qureshi et al. 2017). Most of them are applicable only to certain fruits, with specialized color spaces carefully constructed to distinguish fruits from their background. Some fruits have similar colors to the leaves (such as peppers, green apples, and immature citrus), and the varying illuminations may alter their colors. Thus, these agricultural objects are difficult to detect with only the use of color. Because object contours are invariant to illumination changes, contour-based methods have attracted some attention. These methods often use a circular Hough transform (CHT) to detect round fruits (Murillo-Bracamontes et al. 2012; Roscher et al. 2014; Lu and Sang 2015; Li et al. 2016). However, these approaches cannot detect cylindrical fruits, such as eggplants, cucumbers, and bitter gourds.

Because a depth image is also invariant to lighting conditions and implicitly contains 3D information, many depth-based detection methods have been proposed. Cupec et al. (2014) established a fruit recognition method by depth image analysis. This approach generates a set of triangles from depth images based on Delaunay triangulation, which is a triangulation method that can create a regular triangular mesh for a given point set. Subsequently, a region growing method was used to merge adjacent triangles to generate convex surfaces, each of which represented a possible fruit. Experiments revealed that a large percentage of surfaces were undersegmented (i.e., the detected surface appeared larger than the ground truth result). Wahabzada et al. (2015) employed a 2D laser sensor to acquire point clouds of wheat, barley, and grapevines and developed a histogram clustering algorithm to segment the point clouds with high accuracy. Color information was not considered in the above approaches.

Fusing color and depth information together brings opportunities and challenges. Monta and Namba (2003) established a 3D tomato sensing system comprising a color camera and a laser scanner, in which image thresholding, a simple image segmentation method to create binary images, and depth filtering were used to detect fruits. Rakun et al. (2011) applied thresholding to the hue image to find areas of interest. The areas were then refined via a support vector machine (SVM) classifier trained on texture-based features. Finally, CHT was used to detect spheres from the point cloud of each area. Both methods required selecting a suitable color space. Font et al. (2014) developed a low-cost stereovision system to estimate the size and position of pears and apples that worked only in uniform illumination conditions. Barnea et al. (2016) first detected highlighted regions from hue–saturation–value (HSV) images and then used a fixed-size 3D sliding window along with an SVM classifier on these regions to detect sweet peppers in varying illumination conditions. This method achieved a precision of only 0.55. Rachmawati et al. (2016) used a *k*-means clustering algorithm on red–green–blue (RGB) and depth images in which fruits were placed on an indoor rotation platform and obtained promising results. To recognize apples on trees, Nguyen

et al. (2016) performed depth and color filtering to exclude unnecessary backgrounds, utilized Euclidean clustering to generate clusters, and used a random sample consensus algorithm (RANSAC), a model parameter estimator, to find the true positives. The true positive rates for Gala and Fuji apples were 0.88 and 0.81, respectively. Another apple detection method was presented by Tao and Zhou (2017), who first used region growing to segment a point cloud into a set of clusters and then built an SVM classifier based on the fast point feature histogram (Rusu et al. 2009) to recognize true positives. The algorithm realized a detection accuracy of 0.923. Stein et al. (2016) utilized a faster region-based convolutional neural network, multiple RGB images, and a light detection and ranging device to detect and locate mango fruits. An error rate of 1.36% was reported; however, an expensive and fully functional unmanned ground vehicle was required. Wang et al. (2017) employed a cascade classifier, the Ostu algorithm, and ellipse fitting to recognize mangoes and utilized an RGB-D depth image to estimate the mango size. Most of these methods implicitly adopted a two-step strategy that first generated a huge pool of candidates and then used an SVM classifier to identify true positives. This two-step strategy is used in this work.

The objective of this research is to propose an algorithm that is universal and robust to spherical or cylindrical fruit identification under natural environments. The hypotheses are as follows: (1) a high-efficiency probabilistic image segmentation method can exclude the background, (2) a depth image clustering method can generate a set of clusters for further processing, (3) a 3D shape detection method can identify potential fruits in the clusters, and (4) a discriminative descriptor can identify the true fruit from the output of the 3D detection.

Materials and methods

Image acquisition

A low-cost RGB-D sensor, Kinect V2 (Microsoft Inc.), was used to capture images. Kinect V2 comprises two separate cameras: an RGB and an infrared (IR) camera. The IR camera generated depth images by using a time-of-flight technology (Wang et al. 2017). Because the RGB resolution was 1080×1920 whereas the depth resolution was 424×512, each pair of RGB and depth images was required to be aligned before application. The intrinsic parameters of the RGB and IR cameras and the extrinsic parameters between them were used to perform the alignment operation (Ren et al. 2017). In this way, the RGB image was adjusted to 424×512 pixels.

Three challenging datasets were provided. The first two datasets were captured on December 26, 2017, in the Nansha Base at the Guangzhou Academy of Agricultural Science, Guangdong, China. One of the datasets comprised 100 pairs of RGB and depth images of peppers acquired from the greenhouse, and the other comprised 100 pairs of RGB and depth images of eggplants collected in the field. The third dataset included 80 pairs of RGB and depth images of guavas obtained on July 10, 2018, on a commercial farm in Guangdong, China. Table 1 lists the details of each dataset, and Fig. 1 shows the image acquisition sites. All images were taken from 10:00 a.m. to 3:00 p.m., and the capture distances between the Kinect V2 sensor and plants were approximately 550 mm.

To train, validate, and test the proposed algorithm, 15% of the images in each dataset were randomly selected as the training set, 5% of the images were chosen as the validation set, and the remaining 80% composed the test set. The sizes of the training, validation, and test sets were determined by experience.

Table 1 Number of images and fruits in each dataset

	Pepper dataset	Eggplant dataset	Guava dataset
Number of image pairs	100	100	80
Number of fruits	1083	305	146

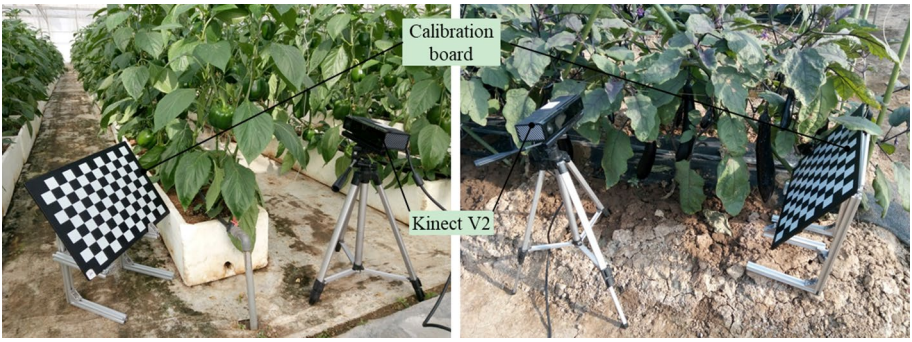


Fig. 1 Image acquisition scene, where the calibration board is used for calibrating the Kinect V2 to obtain its intrinsic and extrinsic parameters

Algorithm overview

The flowchart of the proposed algorithm is shown in Fig. 2. The algorithm uses color, depth, and shape information of target fruits and includes four steps. First, each RGB image is segmented as a binary mask by the proposed probabilistic image segmentation method that aims at excluding only useless pixels, such as sky and soil. By calculating the entrywise product of the mask and depth image, a filtered depth image is obtained. Second, a regional growing-based clustering method is introduced to generate a set of clusters from the filtered depth image. Each cluster is then converted into a point cloud. Third, a novel M-estimator sample consensus (MSAC)-based 3D shape detection algorithm is developed to detect likely fruits from each point cloud. Finally, an SVM classifier trained on angle/color/shape-based features is used to identify true fruits. Note that the first three steps are used to generate a set of potential fruits, and the final step is to identify the true positives. An example of fruit detection is shown in Fig. 3. More details are described in the following sections.

Probabilistic image segmentation

The images captured by the Kinect V2 include not only the target fruits but also a large amount of backgrounds, such as leaves, branches, soils, and sky. Because the colors of some backgrounds are evidently different from those of the fruits, they are easily segmented using color information only, thus reducing the computational burden. Here, a probabilistic image segmentation method (Harrell et al. 1989) is modified and applied.

First, an RGB image is transformed into an HSV space because it is relatively insensitive to illumination changes. The posterior probability of the foreground f given the color attributes of pixel i is then calculated by using the Bayesian formula

$$p\big(f|(h_i,s_i,v_i)\big) = \frac{p\big((h_i,s_i,v_i)|f\big)p(f)}{p\big((h_i,s_i,v_i)|f\big)p(f)+p\big((h_i,s_i,v_i)|b\big)p(b)}, \tag{1}$$

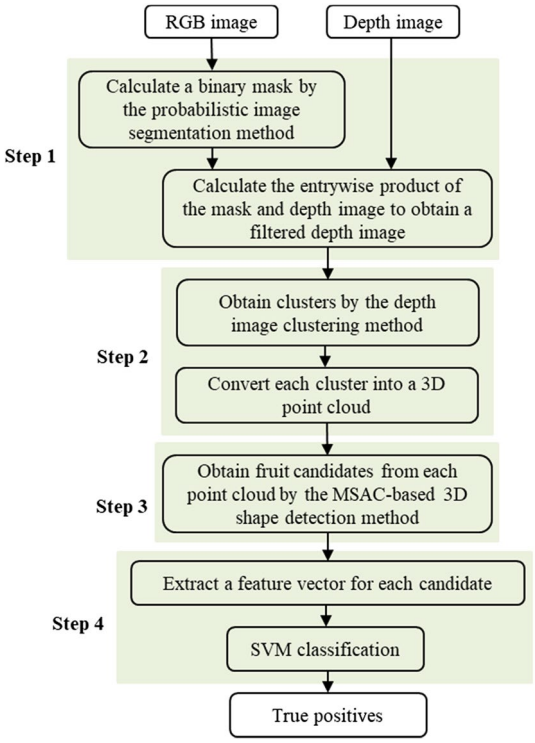


Fig. 2 Flowchart for the proposed algorithm

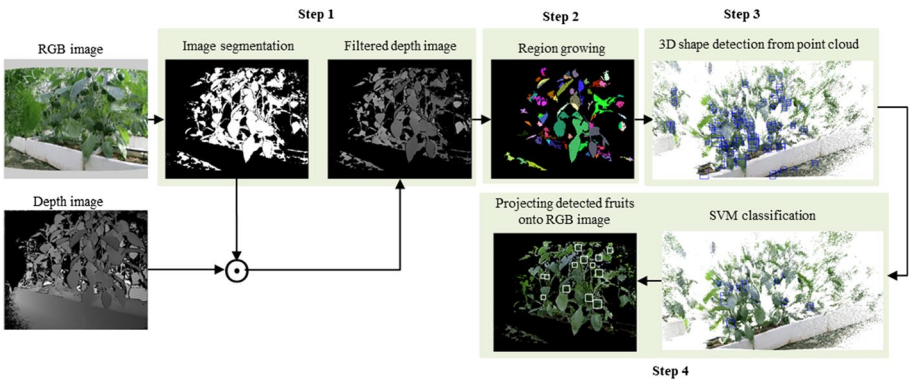


Fig. 3 Example illustration of the proposed detection algorithm where ⊙ means the entrywise product

where (h_i, s_i, v_i) are HSV values of pixel i; f and b refer to the foreground and background, respectively; $p(f)$ and $p(b)$ are prior probabilities learned from the foreground and background training samples, respectively; and $p\big((h_i, s_i, v_i)|f\big)$ and $p\big((h_i, s_i, v_i)|b\big)$ are conditional probability density functions (CPDFs). Based on the naïve Bayes assumption—i.e., each element of the HSV values of pixel i is independent—Eq. 1 can be reduced to

$$p\big(f|(h_i, s_i, v_i)\big) = \frac{p(h_i|f)p(s_i|f)p(v_i|f)p(f)}{p(h_i|f)p(s_i|f)p(v_i|f)p(f) + p(h_i|b)p(s_i|b)p(v_i|b)p(b)}, \tag{2}$$

where $p(h|f)$, $p(s|f)$, and $p(v|f)$ are CPDFs of the H, S, and V components learned from the foreground training samples, respectively, and $p(h|b)$, $p(s|b)$, and $p(v|b)$ are CPDFs learned from the background training samples, respectively. The Gaussian distribution has been widely used to model the bell-shaped CPDF (Harrell et al. 1989; Song et al. 2014). In experiments, the image histograms of the color components of the foreground or background training samples were not found to be strictly bell-shaped that means the Gaussian distribution cannot be used to approximate the CPDF. In this work, the image histograms of the foreground or background training samples are normalized so that the sum over all items in each normalized histogram equals 1, and the resulting normalized image histogram is used to represent the CPDF. To smooth the normalized image histograms, a Parzen density estimator with a Gaussian kernel is used (Duda et al. 2001). Figure 4 shows the pepper CPDFs learned from the foreground and background training samples that were extracted manually from the training set. This method has two advantages: (1) the normalized image histograms are calculated by dividing the image histograms by the total number of pixels in the image, which is more efficient computationally than learning the parameters of a Gaussian distribution that are usually estimated via maximum likelihood estimation, and (2) the normalized image histogram can approximate the complex probability distribution.

By applying Eq. 2 to each pixel of an RGB image, a probability image is generated. In this type of image, pixels with larger values have higher likelihood of belonging to the target fruit. Therefore, threshold segmentation is performed, thus obtaining a binary image as a mask. To avoid oversegmentation, a low threshold (set to 0.1 in the experiments) is recommended. By computing the entrywise product of the depth image I_d and this mask, a filtered depth image is obtained, characterized as I'_d.

Region growing

The filtered depth image contains not only target fruits but also some leaves and branches. These objects occupy various 3D spaces, so it is possible to group them as clusters in 3D space, with each cluster representing a possible fruit. Euclidean clustering (Rusu 2009) was employed by Nguyen et al. (2016) to obtain clusters from point clouds. Its time complexity is O(N log N), which is quite time-consuming, where N is the size of the point cloud; thus, Euclidean clustering is abandoned at this point. Because depth images implicitly contain 3D information on objects, it is used to analyze individual clusters. Here,

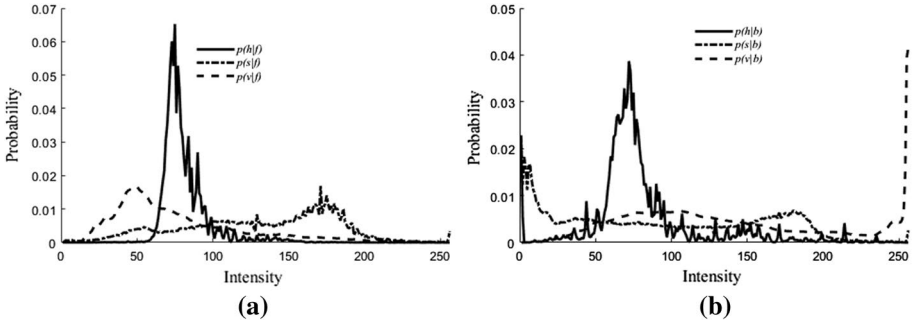


Fig. 4 CPDFs learned from pepper training samples. **a** CPDFs of the foreground; **b** CPDFs of the background

region growing (Tremeau and Borel 1997) is applied to the depth image. This approach repeatedly checks neighboring pixels of seed pixels (i.e., the initial cluster) and determines whether the neighboring pixels should be added to the cluster until every pixel in the image has been processed. Specifically, the first nonzero pixel in I'_d is selected as the first cluster, denoted by $\boldsymbol{O}\{1\}$. The eight-neighboring pixels of every pixel in $\boldsymbol{O}\{1\}$ that are similar to the current pixel are then added to $\boldsymbol{O}\{1\}$ to increase the size of the cluster. The similarity metric used is that the absolute difference between the depth values of two pixels is below a user-defined threshold (which is recommended to be a relatively large value to avoid oversegmentation and set to 5 mm by experience). When $\boldsymbol{O}\{1\}$ stops growing, another nonzero, unprocessed pixel is selected as the second cluster $\boldsymbol{O}\{2\}$, and $\boldsymbol{O}\{2\}$ is augmented in the same way. This growing process is continued until all nonzero pixels have been processed. Finally, a certain number of clusters will be obtained. Any cluster that is less than 150 pixels is removed, because it most likely belongs to the background. The time complexity of region growing is O(N) that is more efficient computationally than Euclidean clustering.

As observed in the experiments, each cluster represented a fruit, leaf, branch, or a combination thereof. Therefore, detecting true positives from each cluster is vital. Because most fruits can be approximated by parameterized shape models, such as a sphere model for a pepper and a cylinder model for an eggplant, a robust 3D shape detection method is investigated in the next subsection. Additionally, because clusters obtained by the region growing method represent certain regions in the depth image that contain only 2D information, they should be transformed into 3D point clouds before shape detection. For each cluster $\boldsymbol{O}\{k\}$, $k=1,\ldots,|\boldsymbol{O}|$ ($|\boldsymbol{O}|$ is the number of clusters), the corresponding point cloud with color information is defined as $\boldsymbol{P}\{k\} = \left\{\left[\boldsymbol{p}_{k,i}, \boldsymbol{c}_{k,i}\right]\right\}_{i=1}^{i=|\boldsymbol{O}\{k\}|}$, where $\boldsymbol{p}_{k,i}$ and $\boldsymbol{c}_{k,i}$ are the 3D coordinates and RGB values of pixel i located at image coordinates $\left[u_{k,i}, v_{k,i}\right]^{\mathrm{T}} \in \boldsymbol{O}\{k\}$, respectively. $\boldsymbol{p}_{k,i}$ is calculated by

$$\boldsymbol{p}_{k,i} = \boldsymbol{K}^{-1} \cdot z_{k,i} \cdot \left[\, u_{k,i} \;\; v_{k,i} \;\; 1 \,\right]^{\mathrm{T}}, \tag{3}$$

where $z_{k,i} = I'_d(u_{k,i}, v_{k,i})$, and $\boldsymbol{K}$ refers to the intrinsic matrix of the IR camera.

MSAC-based shape detection

A 3D shape detection method based on MSAC (Torr and Murray 1997) is presented here. MSAC is a robust, iterative parameter estimator. Specifically, in each round, it randomly selects an adequate number of points from the cluster to estimate the parameters of a specified model, and then it computes a cost function defined by

$$E = \sum_i \rho(e_i) / \sum_i 1, \tag{4}$$

where e_i is the fitting error of the ith element of the cluster to the estimated model, and $\rho(.)$ is

$$\rho(e) = \begin{cases} e & e < T \\ T & e \geq T \end{cases} \tag{5}$$

where T is a constant set to 25 mm in the experiments. After a certain number of iterations, if the cost of the lowest cost model is lower than a given threshold d (set to 2 mm in the experiments), this model is chosen as the optimal model.

Each cluster obtained via the region growing method may contain at least one fruit, and the original MSAC cannot detect multiple models simultaneously from a cluster. Thus, MSAC is revised by repeatedly performing MSAC, and removing inliers from the cluster once an optimal model is detected, until this cluster is empty or no optimal model is detected. Each optimal model corresponds to a potential fruit and contains a set of inliers. The pseudocode of the modified MSAC is outlined in Algorithm 1.

Algorithm 1 MSAC-based shape detection

Input point clouds $\boldsymbol{P}\{k\}$, k=1,…,$|\boldsymbol{P}|$, where $|\boldsymbol{P}|$ is the number of point clouds.

Output fruit candidates $\boldsymbol{\Psi}$ (each candidate is a set of inliers).

Step 1. Initialize cost threshold d = 2.0, maximum iteration L = 500, n = number of parameters in the shape model, $\boldsymbol{\Psi} = \emptyset$, *count* = 0, and k = 1.

Step 2. While $|\boldsymbol{P}\{k\}| > 0$

Step 2.1. Initialize local variables $E_{max} \leftarrow d$, $\boldsymbol{M} \leftarrow \emptyset$ (optimal model), and $\boldsymbol{S} \leftarrow \emptyset$ (inliers).

Step 2.2. For l = 1,…, L

- Randomly select n 3D points from $\boldsymbol{P}\{k\}$, estimate model parameters $\boldsymbol{m}_t$ using these points and calculate cost function E_t.
- If $E_t \leq E_{max}$

  $E_{max} \leftarrow E_t$

  $\boldsymbol{M} \leftarrow \boldsymbol{m}_t$

Step 2.3. If $\boldsymbol{M} \neq \emptyset$

Find all inliers from $\boldsymbol{P}\{k\}$ that fit optimal model $\boldsymbol{M}$ and save them in $\boldsymbol{S}$.

$count \leftarrow count + 1$

$\boldsymbol{\Psi}\{cout\} \leftarrow \boldsymbol{S}$

$\boldsymbol{P}\{k\} \leftarrow \boldsymbol{P}\{k\} - \boldsymbol{S}$

Else

End program.

Step 3. $k \leftarrow k + 1$. If $k \leq |\boldsymbol{P}|$, return to Step 2; else end program.

In step 2.2 of Algorithm 1, the model parameters must be estimated using only n points. Because most fruits look like spheres or cylinders, sphere and cylinder models are considered here. The spherical model can be parameterized via

$$x^2 + y^2 + z^2 - 2ax - 2by - 2cz + d = 0, \tag{6}$$

where a, b, c, and d are parameters that can be solved by algebraic elimination using four points. For the cylinder model, the Schnabel et al. (2010) method is employed. It uses two

points, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and their normals $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. First, the direction of the cylinder axis is computed as $\boldsymbol{a} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$. Two lines $\boldsymbol{p}_1 + t\boldsymbol{n}_1$ and $\boldsymbol{p}_2 + t\boldsymbol{n}_2$ are then projected onto the plane $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{p} = 0$. The intersection of the lines is computed as the cylinder center $\boldsymbol{o}$. Finally, the cylinder radius is set to the distance between $\boldsymbol{p}_1$ and $\boldsymbol{o}$ in the plane $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{p} = 0$.

Several problems were found in the experiments: (1) the leaves with spherical surfaces are prone to be detected as false positives when running sphere detection (Fig. 5a), and (2) the branches or leaves with cylindrical surfaces also tend to be false positives when employing cylinder detection (Fig. 5b). Therefore, a false-positive removal method is investigated in the next subsection.

Feature extraction and classification

The aims of this section are to extract a feature vector for each point cloud detected by MSAC and use an SVM classifier to distinguish fruits and nonfruit objects. Each point cloud is generated from only one viewpoint. Thus, part of the surface of the object is missing, and a discriminative descriptor is required. Barnea et al. (2016) and Kusumam et al. (2017) have investigated a shape and an angular descriptor for partial point clouds, respectively. Both descriptors showed reasonable classification results. Because color is inherent to an object, the proposed global point cloud descriptor (GPCD) integrates the shape, angular, and color features of the object of interest. Each component of the proposed GPCD is described as follows.

Angular feature

The point feature histogram (PFH) (Rusu et al. 2009) is a robust local point cloud descriptor that represents angular properties at a point. Here, PFH is extended to be a global descriptor. Specifically, given a fruit candidate $\boldsymbol{\Psi}\{k\}$, the surface normal of each point in $\boldsymbol{\Psi}\{k\}$ is first estimated by principal component analysis (Hoppe et al. 1992). For each pair of points $\boldsymbol{p}_{k,i} \in \boldsymbol{\Psi}\{k\}$ and $\boldsymbol{p}_{k,j} \in \boldsymbol{\Psi}\{k\}\,(i \neq j)$, along with their normals $\boldsymbol{n}_{k,i}$ and $\boldsymbol{n}_{k,j}$, their angular properties are then computed as follows

$$\begin{cases} f_1 = (1 + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{n}_t)/2 \\ f_2 = \left(1 + \boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k,i}\right)/\left\|\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k,i}\right\|\right)/2 \\ f_3 = \left(\pi/2 + \operatorname{atan}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_t, \boldsymbol{u}^{\mathrm{T}}\boldsymbol{n}_t)\right)/\pi \end{cases} \tag{7}$$

where $\boldsymbol{u} = \boldsymbol{n}_{k,i}$, $\boldsymbol{v} = (\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k,i}) \times \boldsymbol{u}$, and $\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}$ ($\boldsymbol{uvw}$ represents a local coordinate system). Each triplet (f_1, f_2, f_3) casts a vote in a 27-bin histogram. The index idx of the bin in which (f_1, f_2, f_3) votes is defined as

$$idx = \operatorname{floor}(3 \cdot f_1) \cdot 3^0 + \operatorname{floor}(3 \cdot f_2) \cdot 3^1 + \operatorname{floor}(3 \cdot f_3) \cdot 3^2, \tag{8}$$

where floor () is a round-down operation. This histogram is normalized to be invariant to the size of $\boldsymbol{\Psi}\{k\}$.

Color feature

The point cloud color is transformed into the HSV space. The mean and covariance of HSV values are calculated such that the mean is a 3D vector and the covariance is a

3×3 symmetric matrix with six different values. By combining the mean and covariance, a nine-dimensional feature is obtained.

Shape feature

A D2 shape function (Osada et al. 2001) is used here. It computes distances between every pair of points from $\boldsymbol{\Psi}\{k\}$ and forms a 30-bin histogram of these distances.

The GPCD feature is a 66-dimensional vector. Figure 6 shows the GPCD features of various objects. The results show that objects of different geometrical surfaces have distinct signatures in the GPCD space. Because GPCD is discriminative, an SVM classifier trained on GPCD features is utilized to identify true positives from the outputs of MSAC (Fig. 7).

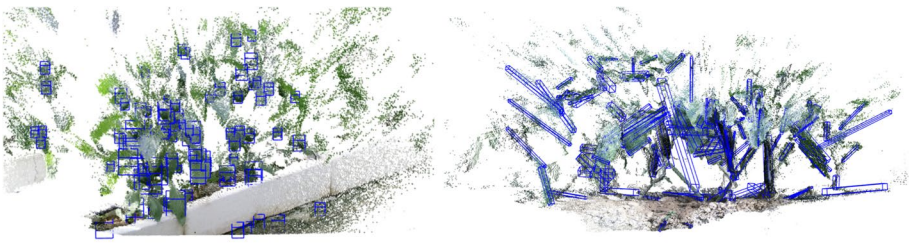
Results and discussion

To test the universality and robustness of the proposed algorithm in natural environments, several quantitative experiments were performed. All codes were implemented in MATLAB (2015a) on a computer with 4 GB of RAM and an Intel Core i3-4150 CPU running at 3.5 GHz.

Performance of the GPCD descriptor

The performance of the GPCD was evaluated by analyzing the classification precision, recall, and accuracy (defined by Eq. 9) of the SVM classifier trained on GPCD features. Two widely used descriptors, histograms of oriented gradients (HOG) (Dalal and Triggs 2005) and local binary pattern (LBP) (Ahonen et al. 2009), were employed as comparison algorithms. The metrics used are defined as follows

$$\begin{cases} \text{precision} = \dfrac{\text{Number of true positves}}{\text{Number of (true positives+false positives)}} \\ \text{recall} = \dfrac{\text{Number of true positves}}{\text{Number of (true positives+false negatives)}} \\ \text{accuracy} = \dfrac{\text{Number of (true positves+true negatives)}}{\text{Number of (true positives+true negatives+false negatives+false positives)}} \end{cases} \tag{9}$$



(a) **(b)**

Fig. 5 Example illustrating shapes detected by MSAC. **a** Spheres detected by MSAC; **b** cylinders detected by MSAC

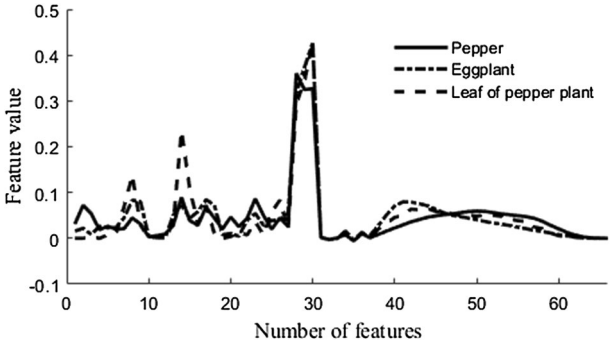


Fig. 6 GPCD features for a pepper, eggplant, and pepper leaf point cloud

(a) **(b)**

Fig. 7 Example illustrating true positives detected by SVM on GPCD features. **a** Detected peppers from Fig. 5a; **b** detected eggplants from Fig. 5b

where true positive refers to a fruit correctly classified, false positive refers to a nonfruit object incorrectly classified as a fruit, true negative is a nonfruit object correctly classified, and false negative is a fruit that is incorrectly classified as a nonfruit object.

The SVM classifier was trained using the positive and negative samples (note that each sample was a point cloud) generated from each training set by performing the proposed algorithm without the SVM classification step (see Table 2). Each sample was projected onto the RGB image plane, and then the minimum enclosing rectangle of the projection was computed as a region of interest where the HOG and LBP features were extracted. A grid search method and tenfold cross-validation were utilized to optimize the SVM parameters during training.

After training, the SVM classifier was validated using the positive and negative samples created from each validation set by following the proposed algorithm without the SVM classification step (see Table 2). The classification precision, recall, and accuracy were then calculated, as listed in Table 3. The GPCD descriptor obtained the two best precisions, two best recalls, and three best accuracies. In this context, the classification performance of GPCD was better than that of HOG or LBP.

Detection performance of the proposed algorithm

The detection performance of the proposed algorithm was validated and compared with a similar algorithm (Nguyen et al. 2016). Nguyen et al. (2016) used a color filter,

Euclidean clustering and an RANSAC-based shape detector to detect apples. Because the color filter developed by Nguyen et al. leveraged the redness property of the apples to exclude the backgrounds, it was not applicable in this case and thus was replaced by the proposed probabilistic image segmentation algorithm. Moreover, an GPCD-based SVM classifier was used to remove false positives. The modified version of their algorithm was termed "MNguyen." In addition, fruits far from the harvesting robot could not be picked, so any depth values larger than 1500 mm in the depth image were set to zero for the two algorithms.

The precision–recall curve visualizes each pair of precision and recall at different thresholds that cut off the SVM outputs. The overall detection performance of the algorithm can be evaluated via the mean average precision (mAP) that measures the area below the curve. Larger mAP values correspond to better detection performance. Figure 8 shows the precision–recall curves of the two algorithms on different test sets. Table 4 lists the mAP values. The mAP values of the proposed algorithm for the pepper, eggplant, and guava test sets were 0.863, 0.741, and 0.807, respectively—all greater than the MNguyen result. Thus, the result showed that the proposed algorithm has superior overall performance to MNguyen.

Table 5 lists the detection precision and recall, at a threshold of 0.5, of the proposed algorithm and MNguyen on three test sets. For the pepper, eggplant, and guava test sets, the precision of the proposed algorithm was 0.864, 0.886, and 0.888, respectively; the recall was 0.889, 0.762, and 0.812, respectively—all larger than the MNguyen result. These performance values confirmed that the proposed algorithm was robust to detecting different types of fruits.

Several detection results are shown in Fig. 9. These examples validated that the proposed algorithm was effective. Some advantages were also revealed: (1) the proposed algorithm was rotation and scale invariant (see Fig. 9d–f), an important property that was helpful for recognizing fruits of different poses and sizes, and (2) the 3D shape detection method could detect fruits that were not perfect spheres or cylinders, hence avoiding missing true positives. In addition, some drawbacks were uncovered: (1) when the color and 3D shape of a leaf resembled the fruit, the GPCD-based SVM classifier could not distinguish them (see Fig. 9a), and (2) owing to occlusion caused by leaves, branches, or adjacent fruits, a single fruit may be split into several parts (see Fig. 9d). Possible solutions include fusing texture features into the developed GPCD descriptor to further improve its discriminability and pruning plants to reduce leaves and branches (Bac et al. 2015).

Table 2 Number of positive and negative samples generated from the training and validation sets

	Training set		Validation set	
	Positive samples	Negative samples	Positive samples	Negative samples
Pepper	190	903	64	302
Eggplant	45	379	16	126
Guava	16	149	13	57

Table 3 Precision, recall, and accuracy for GPCD, HOG, and LBP descriptors on the three different validation sets

Descriptor	Pepper			Eggplant			Guava		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
GPCD	0.846	**0.815**	**0.951**	**1**	0.867	**0.986**	**1**	**0.846**	**0.971**
HOG	**0.921**	0.507	0.898	0.950	**0.904**	0.978	0.889	0.615	0.918
LBP	0.809	0.523	0.892	0.909	0.769	0.971	**1**	0.727	0.959

Bold values indicate the best values

Evaluation of time efficiency

Because a picking cycle for automatic harvesting robots includes detection, trajectory planning, and cutting phases, the detection step was not required to be operated in real time (Nguyen et al. 2016). However, it should not be so time consuming as to slow the whole system. Table 6 lists the average computational time used by the proposed algorithm and MNguyen.

The average computation times of the proposed algorithm for the pepper, eggplant, and guava test sets were 12.93 s, 9.37 s, and 6.96 s, respectively, with an average number of 9.2 peppers, 2.3 eggplants, and 1.48 guavas detected per image; i.e., the detection times per pepper, eggplant, and guava were approximately 1.41 s, 4.07 s, and 4.70 s, respectively. The detection times of MNguyen per pepper, eggplant, and guava were approximately 10.41 s, 44.91 s, and 52.62 s, respectively. These results showed that the proposed algorithm was quite time consuming for robotic harvesting, though it was more efficient than MNguyen. Therefore, future work will focus on improving the overall real-time performance of the proposed algorithm.

Conclusions

Owing to the cluttered backgrounds, occlusion, illumination changes, and low contrast between leaves and fruits, robust fruit detection is highly challenging. In addition, most existing work focuses on the detection of only one type of fruit, thus limiting their application. To resolve these issues, this work investigated a common framework for detecting different types of fruits by using a low-cost RGB-D sensor. Quantitative experiments were carried out to verify the performance of the proposed algorithm, and the following conclusions were obtained:

(1) The probabilistic image segmentation algorithm can remove backgrounds and is efficient computationally.
(2) The depth image clustering algorithm can generate a set of clusters from depth images and has a lower time complexity than Euclidean clustering.
(3) The 3D shape detection algorithm can detect multiple spheres or cylinders in the clusters.
(4) The GPCD descriptor is discriminative, and the SVM classifier trained on GPCD features can remove false positives.

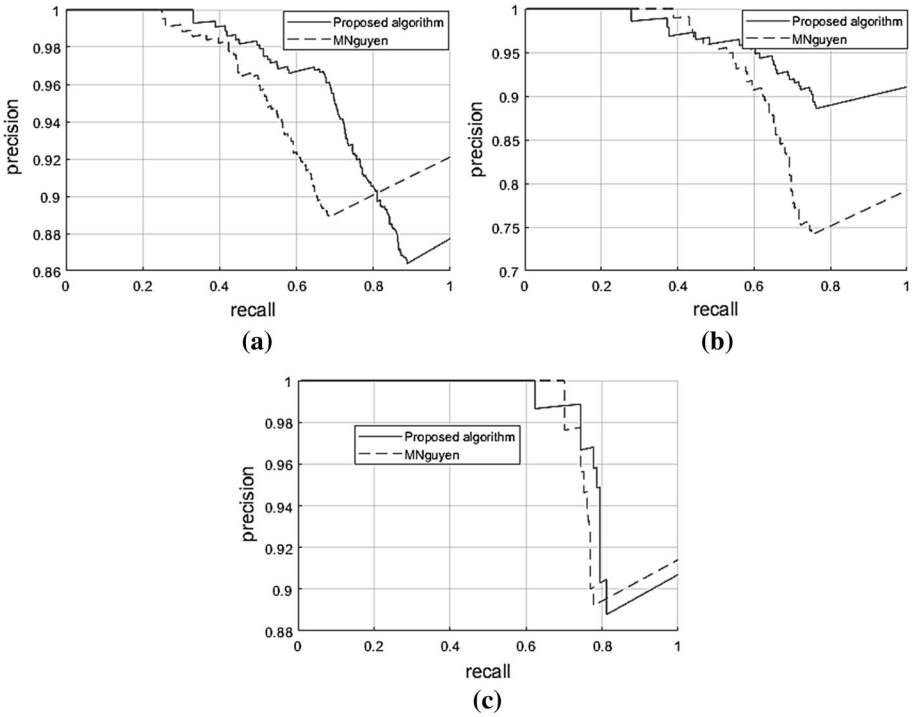


Fig. 8 Precision–recall curves of the proposed algorithm and MNguyen on **a** pepper test set, **b** eggplant test set, and **c** guava test set

Table 4 mAP values of the proposed algorithm and MNguyen on pepper, eggplant, and guava test sets

	mAP	
	Proposed algorithm	MNguyen
Pepper	0.863	0.666
Eggplant	0.741	0.719
Guava	0.807	0.774

Table 5 Precision and recall at threshold 0.5 that binarizes the SVM outputs on pepper, eggplant, and guava test sets

Method	Test set	Total fruits	True positives	False positives	False negatives	Precision	Recall
Proposed algorithm	Pepper	829	737	116	92	0.864	0.889
	Eggplant	244	186	24	58	0.886	0.762
	Guava	117	95	12	22	0.888	0.812
MNguyen	Pepper	829	567	71	262	0.888	0.684
	Eggplant	244	184	64	60	0.742	0.754
	Guava	117	91	11	26	0.892	0.779

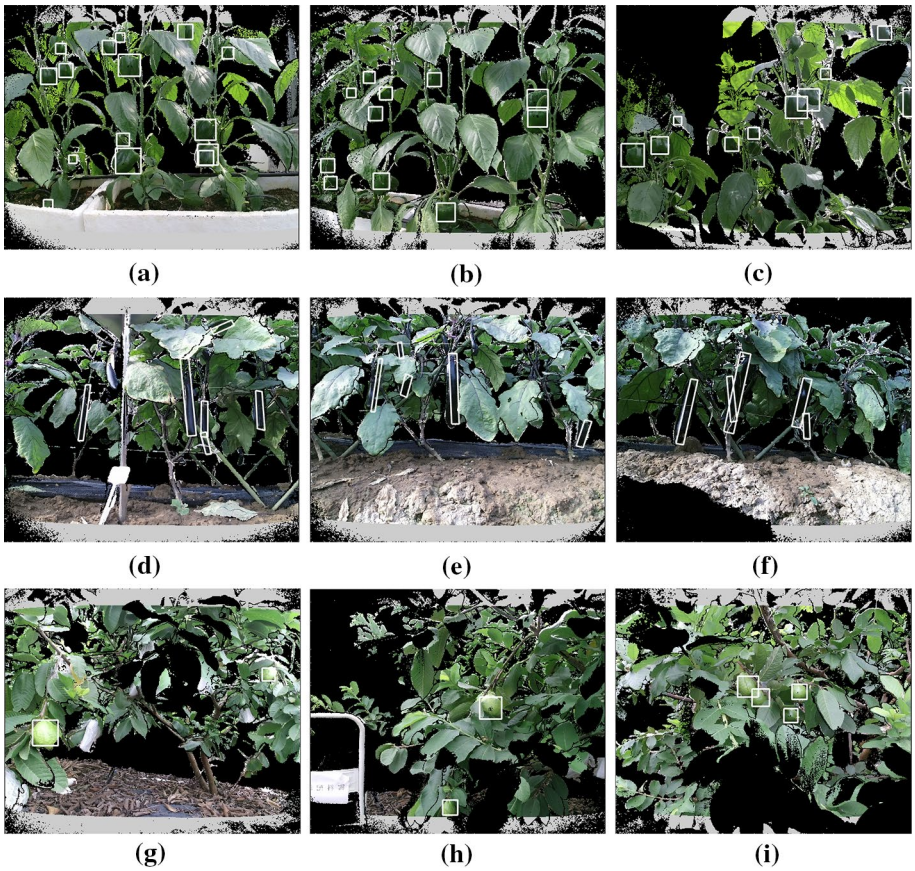


Fig. 9 Example illustrating fruits detected by the proposed algorithm. **a–c** are detected peppers; **d–f** are detected eggplants; and **g–i** are detected guavas. Note that pixels with depth values greater than 1500 mm were set to black

(5) For the pepper, eggplant, and guava test sets, the detection precision is 0.864, 0.886, and 0.888, respectively; the detection recall is 0.889, 0.762, and 0.812, respectively; and the average detection time per fruit is 1.41 s, 4.07 s, and 4.70 s, respectively. These performance values confirm that the proposed algorithm is capable of detecting different types of fruits in the fields, though it is somewhat time consuming.

Although the proposed algorithm is effective, it has some shortcomings to be resolved in future works:

(1) In extreme cases, the color and shape of the leaf or branch may resemble those of the fruit. This issue makes the proposed GPCD descriptor that represents the color and shape properties of the object unable to distinguish between the leaf/branch and the fruit. Future work will focus on integrating other features, such as texture in GPCD.

Table 6 Average computation time (s) per image for the proposed algorithm and MNguyen

Functions	Proposed algorithm (s)			Functions	MNguyen (s)		
	Pepper	Eggplant	Guava		Pepper	Eggplant	Guava
Image segmentation	0.12	0.11	0.10	Image segmentation	0.11	0.11	0.10
Region growing	0.54	0.56	0.21	Euclidean clustering	57.99	93.45	66.23
MSAC	2.45	3.47	2.16	RANSAC	3.06	3.27	2.33
GPCD + SVM	9.82	5.23	4.49	GPCD + SVM	12.57	6.47	8.69
Total	12.93	9.37	6.96	Total	73.73	103.3	77.35

(2) Owing to occlusion by leaves or branches, a single fruit may be split into several parts by the region growing method. Consequently, multiple partial fruits will be detected as false positives. This is a challenging problem to be overcome and will be considered in future work.

Acknowledgements This work was funded by a grant from the National Natural Science Foundation of China (No. 31571568) and a grant from the National Key Research and Development Program of China (No. 2017YFD0700103).

References

Ahonen, T., Matas, J., He, C., & Pietikäinen, M. (2009). Rotation invariant image description with local binary pattern histogram fourier features. In *Proceedings of the 16th Scandinavian Conference on Image Analysis* (pp. 61–70).

Bac, C. W., Henten, E. J., Hemming, J., & Edan, Y. (2015). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888–911.

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

Bulanon, D. M., Kataoka, T., Ota, Y., & Hiroma, T. (2003). A segmentation algorithm for the automatic recognition of fuji apples at harvest. *Biosystems Engineering, 83*(4), 405–412.

Cupec, R., Filko, D., Vidović, I., Nyarko, E. K., & Željko Hocenski. (2014). Point cloud segmentation to approximately convex surfaces for fruit recognition. In *Proceedings of the Croatian Computer Vision Workshop* (pp. 56–61).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 886–893).

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. New York: Wiley.

Font, D., Pallejà, T., Tresanchez, M., Runcan, D., Moreno, J., Martínez, D., et al. (2014). A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm. *Sensors, 14*(7), 11557.

Harrell, R. C., Slaughter, D. C., & Adsit, P. D. (1989). A fruit-tracking system for robotic harvesting. *Machine Vision and Applications, 2*(2), 69–80.

Hoppe, H., Derose, T., Duchamp, T., Mcdonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. *ACM SIGGRAPH Computer Graphics, 26*(26), 71–78.

Kusumam, K., Krajník, 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*(8), 1505–1518.

Li, H., Lee, W. S., & Wang, K. (2016). Immature green citrus fruit detection and counting based on fast normalized cross correlation (fncc) using natural outdoor colour images. *Precision Agriculture, 17*(6), 678–697.

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

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.
Monta, M., & Namba, K. (2003). Three-dimensional sensing system for agricultural robots. In *Proceedings of 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 1216–1221).
Murillo-Bracamontes, E. A., Martinez-Rosas, M. E., Miranda-Velasco, M. M., Martinez-Reyes, H. L., Martinez-Sandoval, J. R., & Cervantes-De-Avila, H. (2012). Implementation of hough transform for fruit image segmentation. *Procedia Engineering, 35*(12), 230–239.
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.
Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2001). Matching 3D models with shape distributions. In *Proceedings International Conference on Shape Modeling and Applications*, (pp. 154–166).
Qureshi, W. S., Payne, A., Walsh, K. B., Linker, R., Cohen, O., & Dailey, M. N. (2017). Machine vision for counting fruit on mango tree canopies. *Precision Agriculture, 18*(2), 224–244.
Rachmawati, E., Khodra, M. L., & Supriana, I. (2016). Fruit image segmentation by combining color and depth data. *International Conference on Information System & Applied Mathematics, 1746*(1), 651–666.
Rakun, J., Stajnko, D., & Zazula, D. (2011). Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. *Computers & Electronics in Agriculture, 76*(1), 80–88.
Ren, C. Y., Prisacariu, V. A., Reid, I. D., & Murray, D. W. (2017). Real-time tracking of single and multiple objects from depth-colour imagery using 3d signed distance functions. *International Journal of Computer Vision, 124*(1), 80–95.
Roscher, R., Herzog, K., Kunkel, A., & Kicherer, A. (2014). Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Computers & Electronics in Agriculture, 100*(1), 148–158.
Rusu, R. B. (2009). Semantic 3D object maps for everyday manipulation in human living environment. PhD thesis. Germany: Computer Science Department, Technische Universit€at Mu¨ 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).
Schnabel, R., Wahl, R., & Klein, R. (2010). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum, 26*(2), 214–226.
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.
Stein, M., Bargoti, S., & Underwood, J. (2016). Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors, 16*(11), 1915.
Tao, Y., & Zhou, J. (2017). Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers & Electronics in Agriculture, 142,* 388–396.
Torr, P. H. S., & Murray, D. W. (1997). The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision, 24*(3), 271–300.
Tremeau, A., & Borel, N. (1997). A region growing and merging algorithm to color segmentation. *Pattern Recognition, 30*(7), 1191–1203.
Wachs, J. P., Stern, H. I., Burks, T., & Alchanatis, V. (2010). Low and high-level visual feature-based apple detection from multi-modal images. *Precision Agriculture, 11*(6), 717–735.
Wahabzada, M., Paulus, S., Kersting, K., & Mahlein, A. K. (2015). Automated interpretation of 3D laser-scanned point clouds for plant organ segmentation. *BMC Bioinformatics, 16*(1), 1–11.
Wang, Z., Walsh, K. B., & Verma, B. (2017). On-tree mango fruit size estimation using RGB-D images. *Sensors, 17*(12), 20170154.
Xiang, R., Jiang, H., & Ying, Y. (2014). Recognition of clustered tomatoes based on binocular stereo vision. *Computers & Electronics in Agriculture, 106,* 75–90.
Zou, X., Ye, M., Luo, C., Xiong, J., Luo, L., Wang, H., & Chen, Y. (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.
Zou, X., Zou, H., & Lu, J. (2012). Virtual manipulator-based binocular stereo vision positioning system and errors modelling. *Machine Vision and Applications, 23*(1), 43–63.