Precision Agriculture (2023) 24:764–782
https://doi.org/10.1007/s11119-022-09974-4

Three-dimensional location methods for the vision system of strawberry-harvesting robots: development and comparison

Yuanyue Ge[1] · Ya Xiong[1] · Pål Johan From[1]

Accepted: 20 October 2022 / Published online: 7 November 2022

Abstract
For most fruit-harvesting robots, an essential task of the machine vision system is to provide the manipulator with an accurate three-dimensional location of the target. However, the accuracy of this location can be affected by various factors. This study aimed to develop seven location methods, to investigate their effectiveness, as well as the influences of different camera modes and camera types, and, ultimately, to ascertain which was the optimal method. These methods utilized the pixels of the detected targets in each image, the corresponding depth values, as well as the locations of the 2D bounding boxes extracted from the detection results. These location methods differed in the way that they obtained the position of the 3D bounding box, and in their use of point clustering or colour thresholding. The images were collected via two types of 3D camera, patterned structured light and time-of-flight. Comparative analysis showed that methods using the 2D bounding box and the selected depth value to calculate the 3D bounding box were faster (0.2–8.4 ms compared to 151.9–325.2 ms) and performed better than the 3D clustering methods. In addition, four modes of the structured light camera were tested and compared. The results showed that the high-accuracy mode had fewer noise points but a lower location rate (89.2–89.9%), while the high-density mode created more noise points but a higher location rate (98.9%). Evaluations also indicated that the data from the time-of-flight camera better represented the 3D shape (26.3% more accurate along the camera's depth direction). Therefore, time-of-flight camera was considered better for the applications that required more accurate 3D shape. This paper, thus, provided references in the selection of location methods, cameras and corresponding modes for related work.

Keywords Strawberry-harvesting robots · Machine vision · 3D location · Depth camera

✉ Ya Xiong
caucoexy@hotmail.com

[1] Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

Introduction

Machine vision systems are one of the essential components of fruit-harvesting robots. Recent research on the development of such systems has tended to focus, either, on identification of fruits in 2D images, (Dai et al., 2021; Fu et al., 2020; Gao et al., 2021; Sa et al., 2016; Yu et al., 2019) or on the integration of machine vision systems into picking robots that are required to locate their targets in 3D co-ordinated systems (Silwal et al., 2017; Williams et al., 2020; Xiong et al., 2020b).

Among these studies of detection in 2D images, some focused on the utilization of traditional methods, such as the identification algorithm developed by Silwal et al. (2014) in which circular Hough transform (CHT) was used for the detection of apples, and the detection algorithm developed by Arad et al. (2019) for sweet peppers, based on colour and shape. Other researchers applied and improved deep convolutional neural networks to detect targets, because of the promising results of deep learning methods. For example, Sa et al. (2016) utilized Faster R-CNN for the detection of fruit such as apples, oranges, sweet peppers, strawberries and others. Fu et al. (2020) utilized the same network for apple detection. In addition to the methods that focus primarily on detection of fruits, some studies have presented entire harvesting systems, in which the vision system detects and locates the fruits in 3D and then sends these 3D locations to the arm control system. For example, Williams et al. (2020) designed a kiwifruit-harvesting robot, proposing two versions of its machine vision system for the detection of kiwifruits. The first version (Williams et al., 2019) utilized a fully convolutional neural network (FCN) (Long et al., 2015) to perform semantic segmentation and find the calyx of the fruit. However, this method required a large amount of memory for image processing and caused a significant reduction in the picking speed. Therefore, in a later work, the authors utilized a faster network, Faster-RCNN (Ren et al., 2015), and the model was trained to detect both the kiwifruit and its calyx. In both versions, the centroid points of the calyxes were used to locate the 3D positions of the kiwifruits.

In addition, Silwal et al. (2017) developed a vision system for apple identification, utilizing an identification algorithm (Silwal et al., 2014) that used CHT to detect apples as circular objects, composed mainly of apple pixels. This method was, thus, regarded as a pixel segmentation method. Its vision system setup consisted of an RGB camera placed on top of a time-of-flight 3D camera, so that the 3D point cloud from the 3D camera could be mapped to the RGB image using extrinsic parameters. Following the co-ordinate mapping, the 3D co-ordinates inside the detected apple circles were used to calculate the 3D positions of corresponding apples. The average values of the x, y and z co-ordinates were used to represent the apple position and removed the outliers along depth direction. Since this paper described the entire machine, no evaluation was made of the location accuracy of its vision system. Another apple-harvesting robot, developed by Onishi et al. (2019), utilized a single-shot detection (SSD) network (Liu et al., 2016) to detect the bounding boxes instead of the pixels of apples, in which the centroid of the detected apple's bounding box was matched with the corresponding 3D point from the camera's point cloud for 3D location.

Moreover, Lehnert et al. (2017) presented a sweet pepper harvesting robot with a vision system that used an RGB-D camera. The camera was mounted on the end-effector and was used to capture point clouds from multiple viewpoints so that a single 3D scene could be created, and a colour-based segmentation method was used to classify red sweet peppers. Furthermore, the authors developed methods to select grasping and cutting poses for improved sweet pepper picking. Similarly, Arad et al. (2020) presented

the design of a sweet pepper harvester with a vision system that incorporated a time-of-flight RGB-D camera. Arad et al. (2019) initially developed a colour- and shape-based detection algorithm that segmented the pixel area of the fruit from the background. For 3D location, they transformed the detected region to calculate the exact 3D position of the point of mass. Furthermore, 2D and 3D sizes of the detected targets were used to remove incorrect detections. Similarly, fruits were recorded detected, reached, cut and caught, while the failures that were possibly caused by location were not analysed. The vision system of another sweet pepper harvester, presented by Bac et al. (2017), used a sensing module with two colour cameras and one time-of-flight camera, in which the colour images were used for the detection of fruits and their ripeness, and time-of-flight images were used for location. This detection algorithm segmented the target pixel's 'blob', which then went through several erosion operators to remove noise. The average of the 3D co-ordinates for all pixels inside the blob were used as the 3D position of the target, and the failure cases caused by location error were recorded in the evaluation section.

In conclusions, the vision system of a harvesting robot mostly used colour images to identify the targets and utilized depth information to calculate the 3D locations. Vision systems are used for target detection in a fruit-harvesting robot. The algorithms for identifying the target can be divided into two main categories. One is based on image segmentation to segment target pixels from the background, while the other approach involves the detection of bounding boxes that indicate the positions of targets in the images (Onishi et al., 2019; Williams et al., 2020). Among those systems using segmentation, some utilized traditional methods based on colour and shape, enabling the selection of a centroid point to obtain corresponding 3D locations or the use of the average 3D co-ordinates of all the detected pixels (Arad et al., 2019; Silwal et al., 2017). The development of deep convolutional neural networks for instance segmentation provided an alternative approach to the vision system, enabling the output of both the detected target position and segmented target pixels. The authors' previous strawberry-harvesting system (Ge et al., 2019a; Xiong et al., 2020a) deployed the instance segmentation network Mask R-CNN (He et al., 2017), to extract the pixels of individual detected strawberries. The extracted pixels of each strawberry were transformed into 3D co-ordinates and a clustering method was implemented to filter noise (Ge et al., 2019b).

The advantage of using an instance segmentation network is that it can accurately locate every pixel of a detected target, thus avoiding extra noise points. However, segmentation networks are computationally expensive, which was why, for example, Williams et al. (2020) upgraded their vision system from one using semantic segmentation (Williams et al., 2019) to a bounding box detection method, because the segmentation network was thought to cause a significant reduction in speed.

The aim of this study was to speed up the detection system as much as possible, due to the purpose of developing a closed-loop vison-guided system. The reported speed for Mask R-CNN was five frames per second (He et al., 2017), while it took approximately 0.8 s to process one image when implementing the method in the authors' system via Nvidia GTX 1060 (Ge et al., 2019b). Therefore, the network of the vision system was upgraded to the YOLOv4 object recognition system (Bochkovskiy et al., 2020), which could reach a detection speed of 20 frames per second using the same hardware settings. However, bounding box detection presented some limitations. The target bounding box might contain pixels from other objects, such as leaves, stems and adjacent fruits, that might vary in depth from the camera view and, therefore, these co-ordinates cannot be used to accurately calculate a target's position. The goal of this paper was, thus, to develop different 3D location methods

and compare their performance for strawberry harvesting. The contributions of this work are as follows:

1. Proposition of seven 3D location methods for the machine vision system of strawberry-harvesting robots
2. Comparative analysis of the effectiveness of the proposed methods on 3D location
3. Evaluation of the influence of different cameras on the various 3D location methods

Materials and methods

Preliminaries and overall system

The overall system of the strawberry harvesting robot is shown in Fig. 1. The complete picking system consisted of a platform, an arm, a gripper and an arch, which was used to reduce the influence of sunlight. This vision system used an RGB-D camera (D435, Intel, USA) to acquire RGB and corresponding depth images, while a previous version of the vision system (Ge et al., 2019a, 2019b, 2020) utilized an instance segmentation network, Mask R-CNN, for the detection and segmentation of targets. The obstacle avoidance algorithm (Xiong et al., 2020a) initially used in the strawberry harvesting machine needed 3D bounding boxes of the detected strawberries. However, a faster detection algorithm was required to achieve real-time picking. Therefore, in 2020, the network was changed to a faster detection algorithm, YOLOv4. The instance segmentation network Mask R-CNN outputted pixels of detected targets, which could be transformed into 3D locations accordingly, while the detection network YOLOv4 generated 2D bounding boxes, which could contain pixels from other objects that needed to be removed by further processing. The motivation of this work was to develop an accurate 3D location method using the outputs from the detection network.

Data acquisition

Three-dimensional cameras include various methods for depth estimation, which have different advantages and disadvantages. The RGB-D camera (D435, Intel RealSense Technology, Colorado, USA) uses patterned structured light to calculate depth values, while

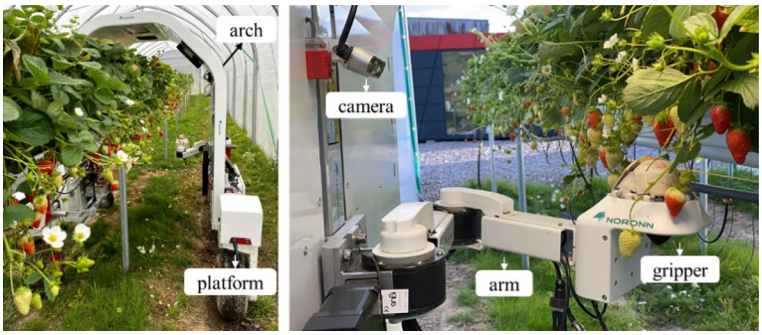


Fig. 1 The strawberry-harvesting robot

the other 3D time-of-flight cameras (Pico Zense DCAM710, Vzense Technology, Qingdao, China) calculate the distance between the camera and the subject by emitting light and recording the time used to receive the returning light. Although neither of these two types of cameras are ideal for outdoor usage because of their sensitivity to sunlight, they are still widely used in agricultural open field applications. In this paper, both cameras were tested in different location methods and their performance was compared for the location of small objects, namely strawberries, in polytunnel conditions. The resolution of the collected images was 640×480 pixels. RGB and depth data were collected simultaneously for each view, and examples of the collected data are shown in Fig. 2a, including the RGB image, the depth values visualized in the colorized depth image, and the cameras' intrinsic parameters. The detection results using YOLOv4 are shown in Fig. 2b.

Location methods

Overall method descriptions

The 2D bounding boxes obtained via the YOLOv4 indicated the positions of the detected ripe and unripe strawberries in the 2D images. Various possible post-processing steps with different location performance and processing speeds could be used to calculate the 3D positions of the berries. Although the clustering method could help to remove noise and improve the accuracy of 3D locations, the application of clustering and coordinating the transformation of all detected pixels would slow down the system. In addition, some noise could be removed by implementing segmentation based on colour within the rectangular bounding boxes. To investigate the influence of these factors on location performance, this paper proposed seven different methods through which to locate targets in a 3D space. The workflows and differences of these methods are illustrated in Fig. 3. The output of the vision system used to indicate the target locations was a 3D bounding box.

The seven methods were grouped into three categories based on the main data that were used to calculate 3D bounding boxes, which were 2D bounding boxes and the depth values, 2D bounding boxes and the clustered transformed 3D points, and 3D transformed points directly, respectively. In the first, the 3D bounding boxes were obtained using the position of the detected 2D bounding box and a calculated depth value. This category of method was subdivided into three sub-methods, based on whether or not colour-based segmentation and clustering were applied. In the second category of methods, all detected pixels

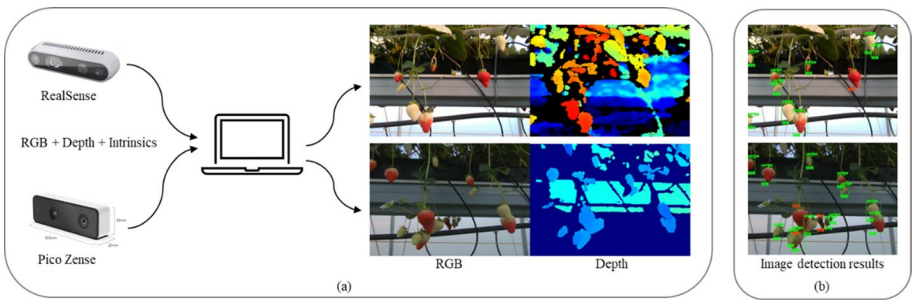


Fig. 2 Data acquired from RealSense D435 and Pico Zense DCAM710 cameras: **a** data capturing process and examples of captured data; **b** image detection results from YOLOv4

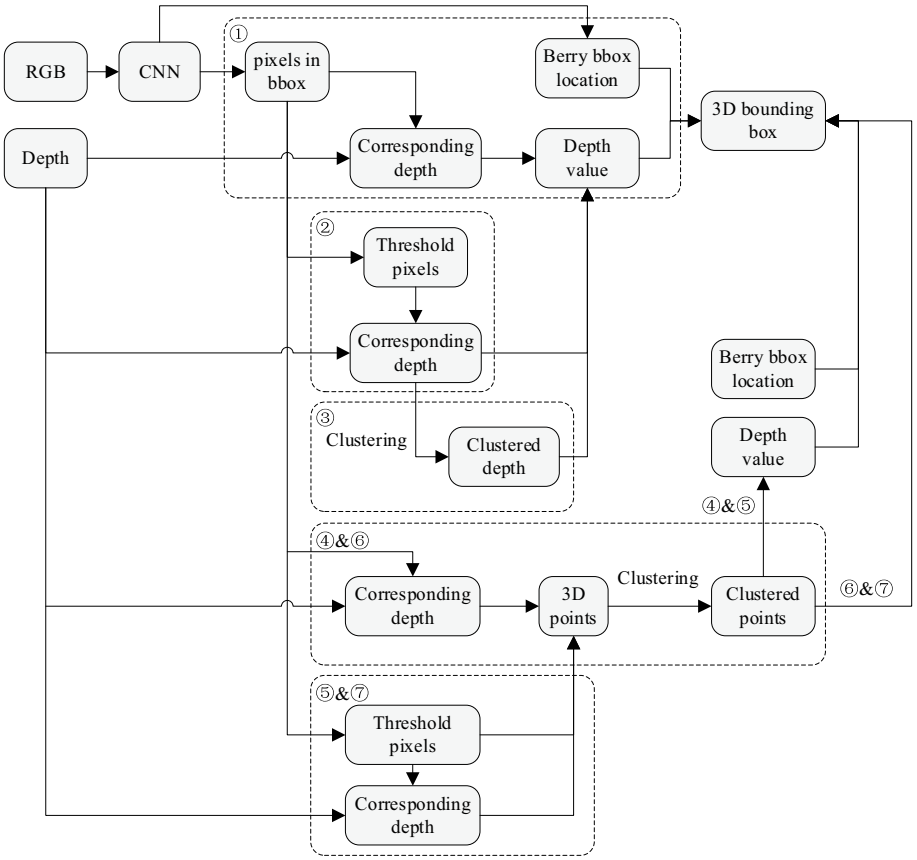


Fig. 3 Flowchart showing seven possible methods through which to detect the 3D positions of targets

were transformed into 3D points and a depth value was calculated based on the transformed and clustered 3D points. This category of method was subdivided into two further methods, based on the use of colour-based segmentation. This category of methods was developed because adjacent noise can be removed by clustering, and it was considered that the location would be more accurate after noise removal. In the third major category of methods, all detected pixels were transformed into 3D points and the positions of the 3D bounding boxes were identified based on the boundary of the transformed points. This category of methods was subdivided into two methods, depending on the use of colour-based segmentation. In this category of methods, the original points from the camera were used to calculate the 3D bounding boxes, thus retaining more information from the 3D points captured by the camera.

Location method first category: 2D bounding box and depth values

As shown in Fig. 4a, the corresponding depth values ($\boldsymbol{V}_d$) of all pixels ($\boldsymbol{P}_b$) inside the detected 2D bounding box were extracted for method one. A median depth value (v_{dm}) of $\boldsymbol{V}_d$ was regarded as the depth location of the detected strawberry. To calculate a 3D

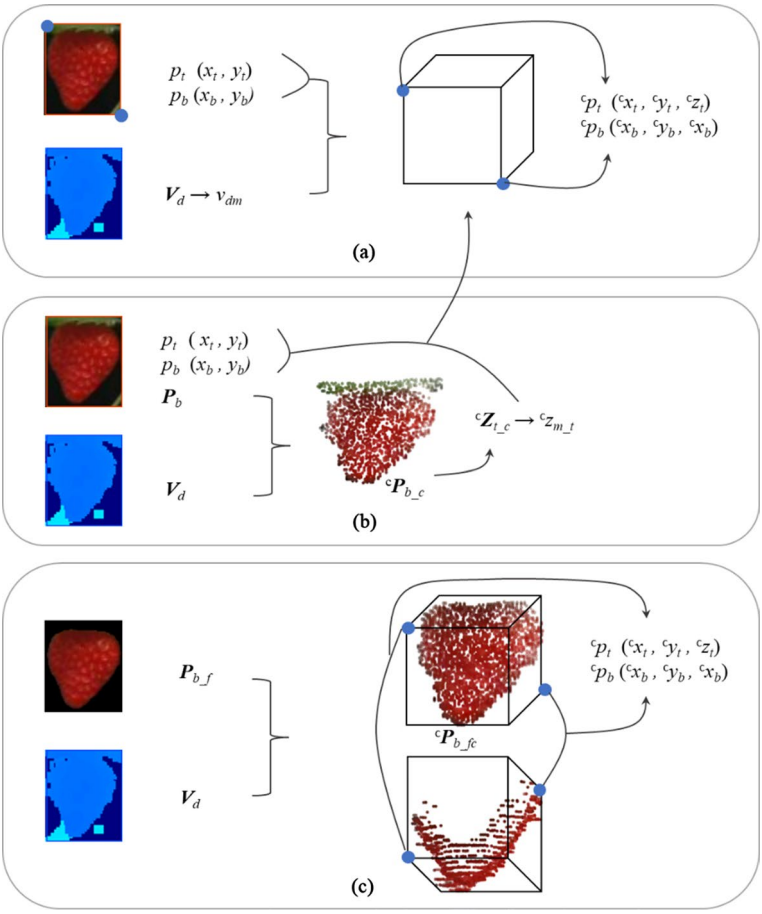


Fig. 4 Illustration of 3D location processes: **a** method one; **b** method four; **c** method seven

bounding box for the position of the berry, the top-left and bottom-right co-ordinates of the detected 2D bounding box, p_t (x_t, y_t) and p_b (x_b, y_b), and the median depth value v_{dm}, were used to calculate a 3D bounding box position ${}^cp_t$ (${}^cx_t$, ${}^cy_t$, ${}^cz_t$) and ${}^cp_b$ (${}^cx_b$, ${}^cy_b$, ${}^cz_b$). Given the intrinsic matrix $\boldsymbol{K}$ of the camera, the 3D location with respect to the camera co-ordinate frame was calculated by ${}^cp_t = v_{dm}\boldsymbol{K}^{-1}p_t$. Additionally, ${}^cp_t$ and ${}^cp_b$ were calculated using the same depth value. A 3D bounding box was then constructed using the width of the 2D bounding as the depth of the 3D box, since strawberries are mostly symmetrical.

In the second method in this category, the above process (Fig. 4a) was taken one step further with the filtration of noisy pixels from the 2D image. In addition to the target berry, a detected 2D bounding box might contain pixels from other objects with distinct depth values, such as background leaves and other unripe berries. Therefore, a colour-threshold processing step was carried out to remove such noise pixels, thereby potentially improving the accuracy of the 3D location. This image processing method based on colour was shown to be simple and fast. However, it would not be stable in an outdoor environment. Therefore, a wide range of hue saturation values (HSV) was applied to ensure the inclusion of most strawberry pixels. The pixels ($\boldsymbol{P}_{b\_f}$) that had been filtered were then used to find

the corresponding depth values ($\boldsymbol{V}_{d\_f}$) in the depth image. A median depth value v_{dm_f} was selected from $\boldsymbol{V}_{d\_f}$ to perform the subsequent steps to calculate the 3D bounding boxes.

Depth values can contain some noise due to the deformation of point clouds and, therefore, a clustering method was added in method three to filter any possible noise in the depth values. A one-dimensional k-mean clustering method was used to apply on $\boldsymbol{V}_{d\_f}$ to obtain the clustered depth values $\boldsymbol{V}_{d\_fc}$. Therefore, in this method, a median depth value v_{dm_fc} was selected from $\boldsymbol{V}_{d\_fc}$ to perform the subsequent steps to calculate the 3D bounding box.

Location method second category: 2D bounding box and clustered 3D points

In methods one to three, described above, one-dimensional depth values were used directly to select a depth value and calculate the 3D bounding boxes without completing co-ordinate transformations from the 2D image pixels to the 3D co-ordinates. However, completely transformed co-ordinates can more precisely represent the point cloud distribution in a 3D space, enabling more accurate performance of the clustering method. Therefore, in methods four and five, the pixels $\boldsymbol{P}_b$ of the detected 2D bounding box were transformed from 2D pixels to 3D points ${}^{c}\boldsymbol{P}_b$, before the clustering of 3D points to remove noise. The 3D clustering method used here was the density based DBSCAN algorithm (Ester et al., 1996). As before, a median depth value was selected to calculate the 3D location of the detected berry. In method five, the pixels in the detected berry bounding boxes went through a thresholding process based on their HSV values, as described in method two; this process was not included in method four. The process used in method four can be seen in Fig. 4b. $\boldsymbol{P}_b$ and $\boldsymbol{V}_d$ were used to calculate the complete transformed points ${}^{c}\boldsymbol{P}_b$ and then ${}^{c}\boldsymbol{P}_{b\_c}$ was obtained by applying the clustering process. A median depth ${}^{c}z_{m\_t}$ from ${}^{c}\boldsymbol{Z}_{t\_c}$ obtained from ${}^{c}\boldsymbol{P}_{b\_c}$ and the position of 2D bounding box p_t and p_b were used to calculate the 3D bounding box. Note, the values of ${}^{c}\boldsymbol{Z}_{t\_c}$ equal to the corresponding $\boldsymbol{V}_d$.

Location method third category: transformed 3D points

The 3D bounding box location, determined in methods one to five, was calculated from a selected depth value and the 2D bounding box co-ordinates. In this way, much of the extraneous noise can be removed from the deformed point cloud. However, the final 3D bounding box obtained cannot represent the actual point distribution of the primary 3D points. Thus, to investigate how the original 3D points work, methods six and seven were proposed to calculate the 3D bounding box using the original transformed 3D points (Fig. 4c). As with methods four and five, methods six and seven transformed the detected 2D strawberry pixels $\boldsymbol{P}_b$ *or* $\boldsymbol{P}_{b\_f}$ to 3D points ${}^{c}\boldsymbol{P}_b$ and a 3D clustering method was used to obtain ${}^{c}\boldsymbol{P}_{b\_c}$ and ${}^{c}\boldsymbol{P}_{b\_fc}$, respectively. However, methods six and seven used the clustered points ${}^{c}\boldsymbol{P}_{b\_c}$ or ${}^{c}\boldsymbol{P}_{b\_fc}$ directly to outline the 3D bounding boxes, rather than using a specified depth value to calculate them. As can be seen in Fig. 4c, which illustrates the process used in method seven, the 3D bounding box was obtained based on the outer contour of the original points. In this example, $\boldsymbol{P}_{b\_f}$ were pixels in the 2D box that have been filtered and the 3D points were ${}^{c}\boldsymbol{P}_{b\_fc}$.

Location examples: third category of methods

In Fig. 5, two examples, namely e1 and e2, are presented to show the co-ordinate transformation process and the effects of colour filtering and clustering on the transformed

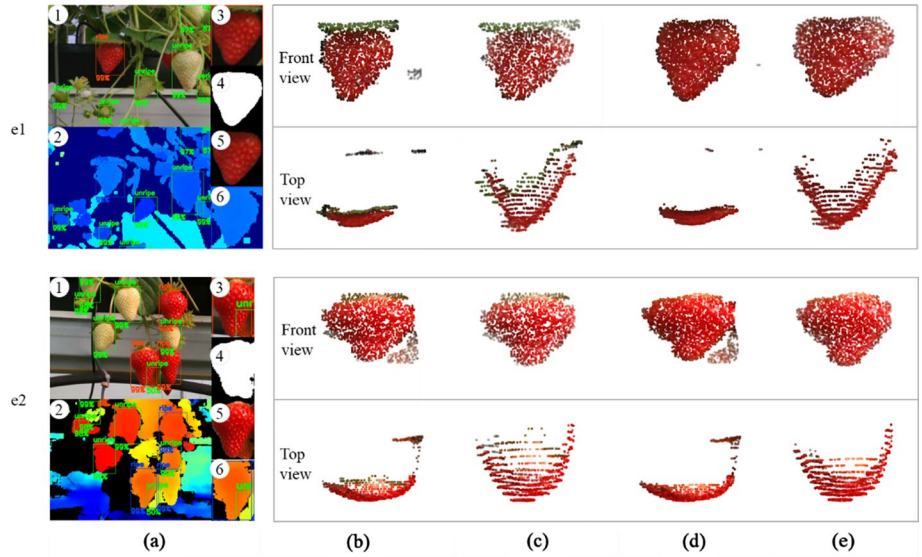


Fig. 5 Two examples of the transformation from image to 3D points, both with and without filtering and clustering: **a**-(1) original RGB image with detection results, **a**-(2) colorized depth image with detection results, **a**-(3) detected berry target in the detected 2D bounding box, **a**-(4) thresholded binary image of (3), **a**-(5) remaining pixels after colour filtering from (3), and **a**-(6) corresponding depth image of detected berry; **b** transformed 3D points without any filtering or clustering; **c** transformed 3D points with clustering; **d** transformed 3D points with filtering; **e** transformed 3D points with both filtering and clustering

3D berry points. The examples from the third category were introduced here because they used the transformed 3D points directly and could reflect the effects more intuitively. Figure 5a-(1) and a-(2) are the detection results of the RGB image and corresponding colorized depth image, respectively. Figure 5a-(3) is the bounding box area of the RGB image, and Fig. 5a-(4) shows the binary results applying the colour filtering. Figure 5a-(5) shows the remaining pixels $\boldsymbol{P}_{b\_f}$ after thresholding, while Fig. 5a-(6) shows the visualized corresponding depth values $\boldsymbol{V}_d$ of $\boldsymbol{P}_b$. However, some RGB pixels $\boldsymbol{P}_b$ do not have valid depth values $\boldsymbol{V}_d$ due to the camera's inaccurate sensing, as can be seen from the white 3D points in Fig. 5b. Figure 5c, d and e are the points that were transformed using only clustering (${}^{c}\boldsymbol{P}_{b\_c}$), only colour filtering (${}^{c}\boldsymbol{P}_{b\_f}$), and both colour filtering and clustering (${}^{c}\boldsymbol{P}_{b\_fc}$), respectively.

In the results with clustering, shown in Fig. 5c, some extraneous noise was removed. However, some points that were close to the target berry, such as the leaves, could not be removed. Some of those points could, thus, be additionally filtered by applying the colour thresholding method on the RGB image, as shown in Fig. 5d and e, in which the points of the green leaves on the top of the berry have been removed. Additionally, the points transformed using only colour filtering included other noise, such as the random points in the background in Fig. 5e1-(d) and other adjacent ripe berry points that could not be filtered, like in Fig. 5e2-(d). Clustering can remove some of these kinds of noise. The transformed points combining both clustering and filtering, as shown in Fig. 5e, were most likely to have the least noisy points. Thus, tests should be designed to investigate the effects of these methods on location results.

Evaluation results

Results from different location methods

A total of 267 berries detected in the default mode were used to test the proposed seven location methods. The number of located berries, time consumed, and mean and standard deviations of lengths along the x, y and z axes for each method are summarized in Table 1. M_x, M_y and M_z are the average lengths in the three directions, and Std_x, Std_y and Std_z are the corresponding standard deviations. The mean and standard deviations of the lengths along different axes indicate the sizes of the detected targets in 3D and provides information about the distribution of the 3D points of the detected target.

The located berries were those with valid 3D points that could be used to generate the 3D bounding boxes, while the unlocated berries were those that lost some valid 3D points, either because the berries were not perceived by the camera, or because the points were filtered out during processing. The number of located berries decreased from method one to method three because of the additional processing steps, which not only filtered noisy points but also sometimes incorrectly filtered out some berries lacking complete depth values. Figure 6a–c show three examples of incomplete depth values, in each example, image No. 6 shows the colorized depth values of the target detected. Figure 6a, b are examples of incomplete depth values caused by the occlusion of front stems and adjacent objects, while Fig. 6c is an example in which the target was small and far from the camera. A target cannot be located if all depth values are lost, such as the case shown in Fig. 6c. While the berries in these incomplete depth values were still located using method one, their locations could be inaccurate because the depth values were mostly from surrounding objects. For example, the depth values in Fig. 6b were mainly obtained from berries that were in front of the target berry and its own depth values were lost because of the occlusion. Examples of 3D points of a target without complete points can be seen in Fig. 7, in which most of the berry points were lost because of adjacent occlusions. In the cases of methods two and three, the depth values were indexed by the filtered colour image Fig. 6—(5), and the filtered depth values corresponding to the berry pixels could be used to locate the berry targets. These examples illustrate that occluded targets such as these can still be located, providing there are some valid depth values remaining after thresholding. The same applied in methods four to seven, in which the berries could only be located if there were some valid 3D points available after processing.

Table 1 Test results of location methods one to seven using a RealSense camera

Method	Berries located	Location rate (%)	M_x (mm)	M_y (mm)	M_z (mm)	Std_x (mm)	Std_y (mm)	Std_z (mm)	Time (ms)
1	265	99.3	42.6	55.9	42.6	10.9	12.2	10.9	0.2
2	263	98.5	42.9	56.3	42.9	10.9	11.9	10.9	0.2
3	263	98.5	42.4	56.8	43.4	10.6	11.6	10.6	8.4
4	265	99.3	42.7	55.9	42.7	10.8	11.9	10.8	140.5
5	263	98.5	43.1	56.5	43.1	10.8	11.8	10.8	228.0
6	265	99.3	41.6	52.9	20.8	12.0	14.8	10.2	325.2
7	263	98.5	37.7	47.9	17.6	12.6	15.9	10.0	151.9

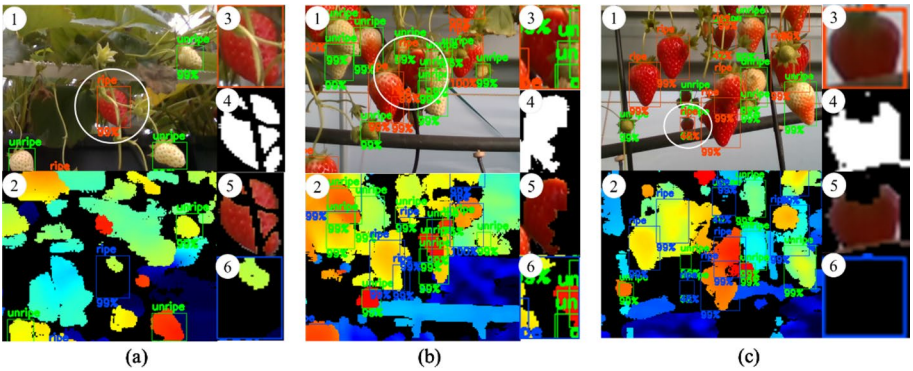


Fig. 6 Examples of target berries with incomplete depth values (marked by a white circle): **a** and **b** are example of berries that are occluded by adjacent objects; **c** is an example in which the berry was far from the camera, consequently losing depth values; (1) RGB image with detected bounding boxes; (2) corresponding colorized depth image; (3) detected target in RGB image; (4) thresholded binary image; (5) thresholded RGB image; (6) depth image of the detected target, showing the depth values captured from the depth camera

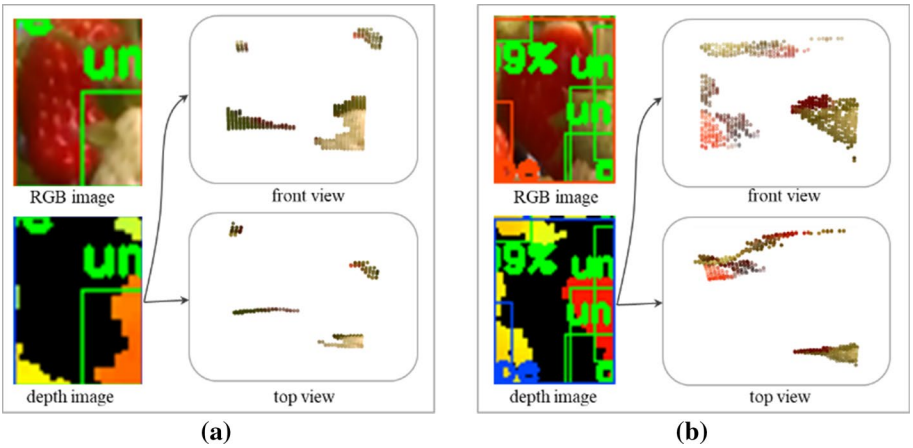


Fig. 7 Examples of detected targets with incomplete 3D points: **a** and **b** are similar examples that the 3D points of the target berry were lost because of adjacent occlusions, in which the left column of each example shows the RGB image of detected target and corresponding colorized depth image, and the right column of each example shows the front and top view of the 3D point cloud

Results from different RealSense camera modes

The RealSense camera supports different modes for depth capturing. Examples of colorized depth images captured in four modes, namely default (DF), hand (HN), high accuracy (HA) and high density (HD), are shown in Fig. 8b, in which the order of point density from high to low is HD, DF, HN, then HA. The location methods were tested in these four camera modes to explore the effects of depth values in different modes on the results.

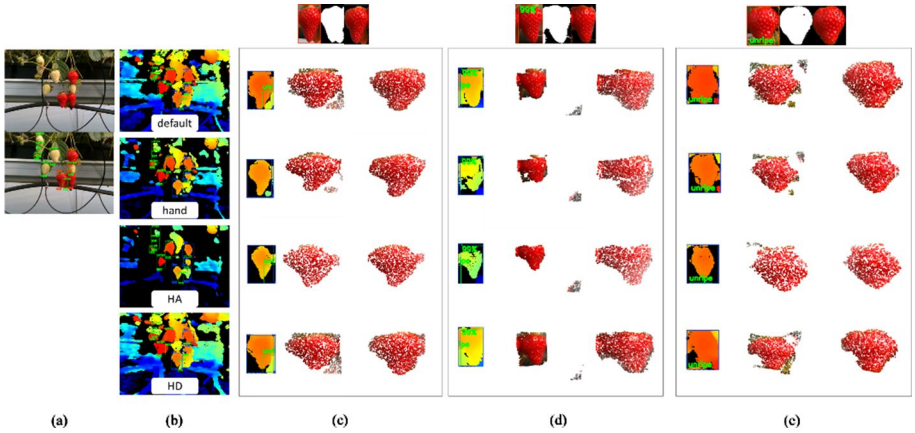


Fig. 8 Images and 3D data from four camera modes: **a** RGB images; **b** correponding colorized depth images in four modes; **c**, **d** and **e** the data of three ripe strawberries in the images, in which the first column is the colorized depth image of the berry; the second column is the original points from the camera; the third column is the filtered and clustered points of the berry; and the three images on the top of each example are the RGB image, thresholded binary image and thresholded color image of the target berry, respectively

Depth values in HD mode were found to have the maximum density, while the density of the depth values in HA mode was much lower but had fewer noisy points and the points were more precise. The point densities in the DF and HN modes were in the middle of the point densities of the other two modes. Figure 8c, d and e show three sets of strawberry 3D points in the four modes. The strawberry points on the left column (second column of examples (c), (d) and (e)) are the complete transformed points from the depth image, and the strawberry points on the right column are the points after filtering and clustering. It can be seen in the figure that the point set in HD mode had the most noise, while that in the HA mode had the least noise and, thus, represented the berry shape more precisely.

The results of methods two and seven using the four modes are shown in Table 2, including the detected and located berries, and mean and standard deviations along three directions. Methods two and seven were considered representative methods, with one using

Table 2 Results of four camera modes using method two and method seven

	Berries Detected	Berries Located	Location Rate (%)	M_x (mm)	M_y (mm)	M_z (mm)	Std_x (mm)	Std_y (mm)	Std_z (mm)
DF M2	267	263	98.5	42.9	56.3	42.9	10.9	11.9	10.9
DF M7	267	263	98.5	**37.7**	**47.9**	**17.6**	**12.6**	**15.9**	**10.0**
HN M2	275	268	97.5	43.7	56.6	43.7	9.93	12.2	9.9
HN M7	275	268	97.5	**34.7**	**44.3**	**15.9**	**11.8**	**16.4**	**8.9**
HA M2	288	259	89.9	43.4	56.0	43.4	9.36	11.6	9.4
HA M7	288	257	89.2	**28.3**	**38.6**	**12.9**	**12.4**	**17.4**	**7.3**
HD M2	275	272	98.9	43.6	56.3	43.6	10.4	11.6	10.4
HD M7	275	272	98.9	**38.5**	**47.6**	**20.5**	**12.4**	**15.5**	**10.1**

DF default, *HN* hand, *HA* high accuracy, *HD* high density, *M2* method two, *M7* method seven

the 2D bounding box to calculate the 3D bounding boxes and the other using the original points to calculate the 3D bounding boxes.

The results showed that the rate of location was highest in HD mode, followed by the rates in the DF, HN and HA modes. This was mainly due to the density of the 3D points. As can be seen in Fig. 8, the density of depth points in the four modes, from lowest to highest, was found in the HA, HN, DF, then HD modes. These results indicate that the smaller the density, the lower the rate of location using the proposed location methods. As mentioned above, the berry could be located as long as there were valid depth values. When the depth density was low, the possibility of having valid depth points after thresholding and clustering was also low, which is why the HA mode, with lowest depth density, delivered the lowest rate of location.

The mean lengths and standard deviations using method seven are shown in bold in Table 2. The lengths in three dimensions were shortest in the HA mode, followed by those in the HN and DF modes, with those in HD mode the longest. Therefore, these results indicate that the lower the density, the shorter the length. The lengths of x, y and z in the different modes using method 2 were similar, because this method used 2D bounding boxes to extract 3D bounding boxes and so the influence of the point density was much smaller. Similarly, with regard to deviations in length, no obvious regularity was found for the results using method two. Same regularity can be seen from the deviations in the length in z direction, in that the lower the density, the smaller the standard deviation in the z direction.

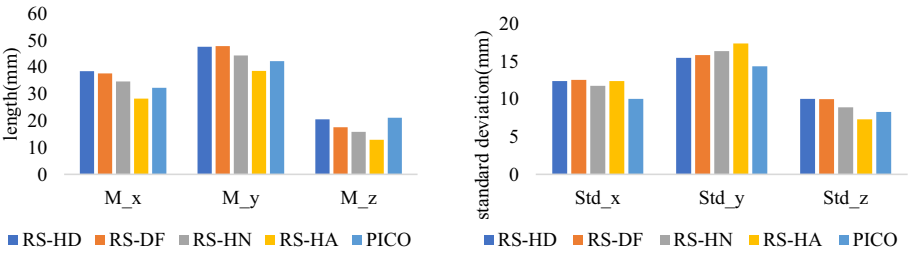
Results from two types of cameras

In this study, different cameras were found to affect the quality of the captured 3D points, which could consequently affect the location performance. Therefore, the results of the structured light camera and time-of-flight camera were compared. The results from the RealSense camera have been listed in the above two sections, while the results of the Pico Zense camera are shown in below Table 3. The ground truth number of berries from the images was 151. The setting of the Pico Zense camera was similar to the HA mode of the RealSense camera, so their location rates were close.

The aim of this section was to compare the performances of the two cameras. The results from method seven, which used the original points from the camera to locate the berry, reflected the most information and, therefore, these are shown in Fig. 9a. It can be seen that the most obvious difference from the average results of all methods was

Table 3 Results of data from the Pico camera

Method	Berries located	Location rate (%)	M_x (mm)	M_y (mm)	M_z (mm)	Std_x (mm)	Std_y (mm)	Std_z (mm)
1	136	90.07	41.6	54.7	41.6	9.1	10.9	9.1
2	134	88.74	42.1	55.2	42.1	8.7	10.7	8.7
3	134	88.74	42.2	55.3	42.2	8.6	10.2	8.6
4	136	90.07	41.5	54.7	41.5	9.0	10.9	9.0
5	134	88.74	42.2	55.2	42.2	8.6	10.7	8.6
6	136	90.07	33.7	46.1	24.0	10.5	13.4	10.8
7	134	88.74	32.3	42.3	21.1	10.0	14.4	8.3



(a) Mean length **(b)** Standard deviation

Fig. 9 Results of two cameras using method seven. RS means RealSense

that the mean length in the z direction of the Pico camera was greater than the z length of all modes of the RealSense camera. The average z length of the RealSense camera in four modes was 16.7 mm, while the average z length of the Pico camera was 21.1 mm, almost half of the diameter of the strawberry, which was closer to the real 3D shape, since only half of the target could be captured by a single-view camera along the depth direction. This meant that the Pico camera could provide 26.3% more accurate shape information along the depth direction. A reasonable explanation for this was that data from the Pico camera would more precisely present the 3D shape of the strawberry, while that of the RealSense camera may flatten the shape. While the mean lengths from the Pico camera were relatively larger, the standard deviations were smallest, except for those of the z direction, which were the second smallest, as can be seen in Fig. 9b. Since methods six and seven used the original 3D points, the smaller deviation indicates that the fluctuation of the data was smaller. The reasons for the larger deviation in the RealSense camera could be that the size of the 3D points was more easily changed in the different captured frames and also that there might be a large amount of data when there was more noise.

To ascertain the size changes of the 3D points from these two cameras, abnormal data were defined by checking the size of the detected 3D bounding boxes. Two ranges were used to extract the data with large sizes and to collect the corresponding number of cases. The first range setting was $x > 60$ mm, $y > 80$ mm and $z > 60$ mm, and the second range setting was x in [50–60 mm], y in [70–80 mm], and z in [50–60 mm]. The results from methods two, six and seven are shown in Table 4. As there were no significant differences among methods one to five, only the results from method two, six and seven are listed in the table.

As can be seen from the table, the rate of outliers from the RealSense camera was larger than that from the Pico camera for all methods and modes. For example, in the first range setting, the rate of outliers for method two was 14.9%, while the rate was 20.2% for method two in the RealSense default mode. Furthermore, there were no outliers from the Pico camera in the second range setting, while there were quite a few from the RealSense camera.

To more fully investigate the reasons for these outliers, the data from the second range setting were collected, and the results are summarized in Table 5. Reasons a to c in the table were found to lead to insufficient 3D points, resulting in the failure of the clustering method. In this case, the noise points could not be filtered, and the noise made

Table 4 Length outliers along *x*, *y* and z axes: the number outside the brackets indicates quantity, and the number inside the brackets is percentage of the outliers to the total number of detected targets

Camera	Method	*x* outliers (50–60 mm)	*y* outliers (70–80 mm)	*z* outliers (50–60 mm)	*x* outliers (>60 mm)	*y* outliers (>80 mm)	*z* outliers (>60 mm)
Pico Zense	2	20(14.9%)	8(6.0%)	20(14.9%)	0	0	0
	6	3(2.2%)	1(0.7%)	4(2.9%)	0	0	0
	7	1(0.8%)	0	0	0	0	0
RealSense Default mode	2	53(20.2%)	26(9.9%)	53(20.2%)	9(3.4%)	3(1.1%)	9(3.4%)
	6	45(17.1%)	17(6.4%)	2(0.8%)	22(8.3%)	15(5.7%)	15(5.7%)
	7	22(8.4%)	13(5.0%)	2(0.8%)	12(4.6%)	8(3.0%)	11(4.2%)
RealSense HA mode	2	53(20.5%)	22(8.5%)	53(20.5%)	7(2.7%)	6(2.3%)	7(2.7%)
	6	14(5.2%)	3(1.1%)	1(0.4%)	4(1.4%)	4(1.5%)	2(0.7%)
	7	6(2.3%)	4(1.6%)	0	1(0.4%)	0	0

Table 5 Reasons for outliers of the first range setting ($x > 60$ mm, $y > 80$ mm and $z > 60$ mm)

Mode	x outliers (> 60 mm)	y outliers (> 80 mm)	z outliers (> 60 mm)
DF	a(7), b(2), c(0), d(3)	a(4), b(2), c(0), d(2)	a(7), b(2), c(1), d(1)
HN	a(4), b(1), c(1), d(2)	a(3), b(0), c(1), d(0)	a(4), b(2), c(2), d(0)
HA	a(0), b(0), c(0), d(1)	a(0), b(0), c(0), d(0)	a(0), b(0), c(0), d(0)
HD	a(5), b(1), c(0), d(3)	a(5), b(1), c(0), d(3)	a(6), b(1), c(2), d(1)
Total	a(15), b(4), c(1), d(9)	a(12), b(3), c(1), d(5)	a(17), b(5), c(5), d(2)

Total occurrence: a(44), b(12), c(7), d(16)

Reasons: a insufficient 3D points were left because strawberry was too far from the camera; b insufficient 3D points were left because only a small part of the berry was detected; c insufficient 3D points were left because strawberry was occluded by other objects; d highly deformed points

the size of the detected bounding box abnormally large. Figure 7. Among the deformed data, the largest outlier was 17.3 mm larger than the limit value, and the deformations were mostly due to occlusion by stems or adjacent berries.

The total occurrence results showed that 55.7% of the cases were due to reason a. These were cases in which strawberries were on the other side of the table, which could still be captured by the RealSense camera, as can be seen in Fig. 2, but were not captured by the Pico camera. Therefore, insufficient 3D points did not occur with the Pico camera, while cases b, c and d did also occur on the Pico camera. For reasons b to d, the larger number of outliers indicated that the 3D point shapes of RealSense camera were not as stable as those of the Pico camera.

Discussions

Location methods

In this study, the results from methods one to five were found to be comparatively uniform because they utilized the position of the 2D bounding box to calculate the 3D bounding box, while methods six and seven used original points to calculate the 3D bounding boxes, and these may have contained deformed 3D points.

The decrease in the standard deviation from method one to method three could be attributed either to noise in the 2D image or to the removal of 3D points by the addition of thresholding and clustering methods. The same phenomenon was observed in methods four to five, and from methods six to seven, because methods five and seven include an additional thresholding step to filter out noise, while methods four and seven do not. The average lengths of x, y and z for methods one to five were found to be similar, because the lengths of x and y were calculated based on the size of the 2D detected bounding boxes and, consequently, were less affected by point cloud deformation.

The time (t1–t7) consumed by methods one to seven could be categorized into three groups, with group one comprising t1 and t2, group two comprising t3, and group three comprising t4–t7. The main difference among these three groups was their clustering method, with group one having no clustering method, group two using a k mean clustering on the depth values, and group three using a DBSCAN clustering method on the 3D points.

Moreover, methods four to seven needed co-ordinate transformations, from 2 to 3D, for all the detected and selected pixels in the image, which consumed additional time.

In conclusion, the methods using 2D boxes and a selected depth value to calculate 3D bounding boxes were found to be significantly faster, while methods that included thresholding benefited from the removal of some noisy points. Among methods one to five, method two was considered the most ideal in terms of both time consumption and effectiveness. Original 3D points from the camera provided direct information about the target locations. However, their accuracy was restricted to the performance of the camera. The unlocated cases were due to incomplete point clouds, but these cases did not always impact negatively on the final picking rate, because even if some berries with incomplete points could be located, their locations might be inaccurate if the target was occluded.

RealSense camera modes

By evaluating these two types of methods using the four camera modes, it could be concluded that the rates of location were affected by the different camera modes. In terms of the sizes and deviations of the detected 3D bounding boxes, the effects of the different modes were not obvious when using method two, while for method seven, as the density of the points in different modes increased, so the size and deviation of the detected 3D bounding box rose accordingly.

Types of cameras

The analysis results of the two types of cameras showed that the 3D points from the Pico camera had less noise and, thus, could more precisely present the 3D shape of the strawberry. This makes it a better option when the goal is to process raw point cloud data, such as when using the original points to locate the target or in 3D reconstruction.

Conclusions

This paper proposed seven methods for the location of targets in 3D locations for strawberry-harvesting robots. These methods used the detection results of a convolutional neural network but differed in their usage of colour filtering and point clustering, and in their extraction of the 3D bounding boxes. The methods were tested on data collected in the four different modes on a RealSense camera and a Pico Zense camera. Evaluations showed: (1) methods using a detected 2D bounding box and single selected depth value to calculate the 3D bounding box were found to be faster and could somewhat avoid the noise from deformed 3D points; the clustering algorithm required more computational resources, while the thresholding algorithm took relatively less time, and the second method (2D bounding box with a median depth value after thresholding) was found to be the optimal solution for the strawberry harvester; (2) among the four modes, the HA mode had less noise but lower located rates, while the HD mode had more noisy points but obtained a higher rate of location; the DF mode could be a compromise choice if the end effector has sufficient tolerance to positional errors; (3) the 3D points from the Pico camera had fewer noisy points and could more precisely present the 3D shape of the strawberry, thus indicating that the Pico camera was a better option for applications that require full information of the 3D points, including shape.

Funding Open access funding provided by Norwegian University of Life Sciences. This work was supported by the Research Council of Norway, Project Title: Strawberry Harvester for Polytunnels and Open Fields, Grant Number: 303607.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., & van Tuijl, B. (2020). Development of a sweet pepper harvesting robot. *Journal of Field Robotics, 37*(6), 1027–1039.

Arad, B., Kurtser, P., Barnea, E., Harel, B., Edan, Y., & Ben-Shahar, O. (2019). Controlled lighting and illumination-independent target detection for real-time cost-efficient applications. The case study of sweet pepper robotic harvesting. *Sensors, 19*(6), 1390.

Bac, C. W., Hemming, J., Van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics, 34*(6), 1123–1139.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv:200410934

Dai, D., Gao, J., Parsons, S., & Sklar, E. (2021). Small datasets for fruit detection with transfer learning. In *UKRAS21 Conference: "Robotics at home" proceedings* (pp. 5–6). London, United Kingdom: UK-RAS.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (Vol. 96, No. 34, pp. 226–231). Palo Alto, USA: AAAI.

Fu, L., Majeed, Y., Zhang, X., Karkee, M., & Zhang, Q. (2020). Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosystems Engineering, 197*, 245–256.

Gao, J., Westergaard, J. C., Sundmark, E. H. R., Bagge, M., Liljeroth, E., & Alexandersson, E. (2021). Automatic late blight lesion recognition and severity quantification based on field imagery of diverse potato genotypes by deep learning. *Knowledge-Based Systems, 214*, 106723.

Ge, Y., Xiong, Y., & From, P. J. (2019a). Instance segmentation and localization of strawberries in farm conditions for automatic fruit harvesting. *IFAC-PapersOnLine, 52*(30), 294–299.

Ge, Y., Xiong, Y., & From, P. J. (2020). Symmetry-based 3D shape completion for fruit localisation for harvesting robots. *Biosystems Engineering, 197*, 188–202.

Ge, Y., Xiong, Y., Tenorio, G. L., & From, P. J. (2019b). Fruit localization and environment perception for strawberry harvesting robots. *IEEE Access, 7*, 147642–147652.

He, K., Gkioxari, G., Dollˊar, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969). Piscataway, USA: IEEE.

Lehnert, C., English, A., McCool, C., Tow, A. W., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters, 2*(2), 872–879.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37). Berlin, Germany: Springer.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440). Piscataway, USA: IEEE.

Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., & Iwai, A. (2019). An automated fruit harvesting robot by using deep learning. *ROBOMECH Journal, 6*(1), 13.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (Vol. 28, pp. 91–99.) Long Beach, USA: Neural Information Processing Systems Foundation Inc. (NeurIPS).

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors, 16*(8), 1222.

Silwal, A., Davidson, J. R., Karkee, M., et al. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics, 34*(6), 1140–1159.

Silwal, A., Gongal, A., & Karkee, M. (2014). Apple identification in field environment with over the row machine vision system. *Agricultural Engineering International: CIGR Journal, 16*(4), 66–75.

Williams, H. A., Jones, M. H., Nejati, M., Seabright, M. J., Bell, J., Penhall, N. D., et al. (2019). Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosystems Engineering, 181*, 140–156.

Williams, H., Ting, C., Nejati, M., Jones, M. H., Penhall, N., Lim, J., et al. (2020). Improvements to and large-scale evaluation of a robotic kiwifruit harvester. *Journal of Field Robotics, 37*(2), 187–201.

Xiong, Y., Ge, Y., & From, P. J. (2020a). An obstacle separation method for robotic picking of fruits in clusters. *Computers and Electronics in Agriculture, 175*, 105397.

Xiong, Y., Ge, Y., Grimstad, L., & From, P. J. (2020b). An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *Journal of Field Robotics, 37*(2), 202–224.

Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. *Computers and Electronics in Agriculture, 163*, 104846.