Available online at www.sciencedirect.com

ScienceDirect

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

Special Issue: Robotic Agriculture

Research Paper

Detection of red and bicoloured apples on tree with an RGB-D camera

Tien Thanh Nguyen, Koenraad Vandevoorde, Niels Wouters, Erdal Kayacan, Josse G. De Baerdemaeker, Wouter Saeys [*]

KU Leuven Department of Biosystems, MeBioS, Kasteelpark Arenberg 30, 3001 Leuven, Belgium

ARTICLE INFO

Article history:
Published online 5 February 2016

Keywords:
Computer vision
RGB-D camera
Fruit detection
Harvesting robot

Recognising and accurately locating fruits on a tree is a critical challenge in developing fruit-by-fruit robotic harvesting. Many researchers have investigated the potential of red, green, blue (RGB) colour imaging for this purpose, but have had limited success due to the occlusion of the target fruits by foliage, branches or other fruits as well as due to the non-uniform and unstructured nature of an orchard environment. Recently, novel, cost-effective camera systems have become available which provide both colour (RGB) and three dimensional (3D) shape information. As these have shown potential for 3D perception for robots operating in unstructured environments, the potential of such an RGB-D camera for the detection and localisation of red and bicoloured apples on tree was investigated in this study. Images were acquired with this camera system in fruit orchards under a light shield blocking direct sunlight, and an algorithm to detect and localise red and bicoloured apples based on colour and shape features was developed. When the algorithm was applied to the data acquired in these orchards, 100% of the fully visible apples and 82% of the partially occluded apples were detected correctly. The location estimation error was below 10 mm in all the coordinate axes of the Cartesian space. This high detection and location accuracy and short processing time (below 1 s for simultaneous detection of 20 apples), makes the developed algorithm suitable for implementation in a robotic harvesting system, and for yield estimation and orchard monitoring.

1. Introduction

The harvest of seasonal fruits requires intensive labour for a short period. As seasonal workers have to be hired for this, this cost constitutes a significant part of the entire production cost. To reduce the labour demand from such labour-intensive tasks, automated and mechanised systems for fruit harvesting have been proposed which remove the fruits from the trees by means of air blast, limb shakers, canopy shakers, trunk shakers or chemical abscission (Li, Lee, & Hsu, 2011). However, when applied to delicate fruits such as apples, pears and peaches, these mass harvesting methods cause bruise damage which makes the fruits unsuitable for fresh consumption. Therefore, these fruits are still harvested manually, often by means of seasonal labourers. To reduce or replace this human labour, a selective

[*] *Corresponding author.*
E-mail address: wouter.saeys@biw.kuleuven.be (W. Saeys).

http://dx.doi.org/10.1016/j.biosystemseng.2016.01.007

Nomenclature

RGB	Red, green and blue colour space
RGB-D	Red, green, blue and depth space
RANSAC	Random sample consensus
VGA	Video graphics array
SIFT	Scale-invariant features transform
CCD	Charge coupled device
SVM	Support vector machine
NIR	Near-infrared
IR	Infrared
r,g	Scaled red and green colour space
d_i	Decision function i
$\mathbf{P}$	Point cloud
p_i	Point i of a point cloud
$\mathbf{P}_d$	Depth filtered point cloud
$\mathbf{P}_r$	Colour filtered point cloud
$\mathbf{P}^i_{apple}$	Point cloud containing apple number i
d_{th}	Distance threshold, mm
$\mathbf{C}$	Cluster
$\mathbf{Q}$	Queue of points to check
r	Radius, mm
CHT	Circular Hough transformation
D_{max}	Diameter of an oversized apple, mm
(x_d,y_d,z_d)	Estimated centre of gravity, mm
d_d	Estimated diameter, mm
$\mathbf{T}$	Size threshold, [−]
ld_g	Longitude diameter, mm
td_g	Transverse diameter, mm
TPR	True positive detection rate [%]
FPR	False positive detection rate [%]
(x_g,y_g,z_g)	Ground truth measured position, mm
Δx	Difference error of estimated and measured position, mm

harvesting method (fruit-by-fruit) is needed which mimics the picking task of a human picker (Jiménez, Ceres, & Pons, 2000a). Thanks to the rapid developments in robotics, automated robot systems consisting of a mobile platform carrying one or more manipulator(s) are now considered to be a feasible solution for this problem.

The research on fruit and vegetable harvesting robots started more than 20 years ago and has resulted in various prototypes such as those for tomatoes, eggplants, lettuces (Kondo & Ting, 1998), melons (Edan, Rogozin, Flash, & Miles, 2000) and apples (Baeten, Donné, Boedrij, Beckers, & Claesen, 2008; De-An, Jidong, Wei, Ying, & Yu, 2011). However, the existing prototypes still cannot meet the challenging requirements for practical applications, requiring a combination of high speed and accuracy in a complex, dynamic and continuously changing environment at an acceptable cost.

Detection of fruits on the tree is the task the robot system has to perform first. Therefore, the success rate of this task determines the maximum success rate of the harvesting process. Several approaches for achieving fruit detection have been reported. Although extensive research has been conducted towards fruit detection for harvesting robots, it still remains a challenge to achieve an accurate integrated fruit sensing (detection, localisation, pose estimation etc.) that performs well under practical circumstances (Kapach et al., 2012). For a complete review of fruit detection research during the last two decades, the reader is referred to the articles of Jiménez et al. (2000a), Li et al. (2011) and Kapach, Barnea, Mairon, Edan, and Ben-Shahar (2012). Kapach et al. (2012) reviewed the fruit detection research with a description of the different sensors, visual cues and algorithms that have been used for this purpose. Jiménez et al. (2000a) classified the fruit detection research based on the different sensors used. A similar division will also be followed in this paper to briefly review the recent state of the art in automatic fruit detection.

Many researchers reported on the use of spectral-based sensors for fruit detection. Tabb, Peterson, and Park (2006) detected apples in RGB images which were acquired with a light cover by modelling the background's colour properties. They used the concept of background modelling using Gaussian mixture colour distributions in RGB images. With this approach, they correctly identified 85–96% of both red and yellow apples at a processing time of 63–71 ms per frame. Zhao, Tow, and Katupitiya (2005) combined texture-based edge detection and redness measurement for the detection of red and green apples on the trees. They reported a true positive detection rate for their detection system of 18 out of 20 clearly visible apples.

Another feature which is commonly used for fruit detection with 2D sensors is the shape of the objects. Spherical fruits such as apples, citrus and peaches have approximately a circular shape in a two dimensional (2D) image. For example, Sengupta and Lee (2014) used a circular Hough transformation and a scale-invariant features transform (SIFT) to detect green citrus with 81.7% correct, 25.6% false positive and 18.3% misclassification. Ji et al. (2012) used a video graphics array (VGA) (640 × 480) colour charge coupled device (CCD) video camera in an uncontrolled environment without light cover and without artificial light source. They proposed a method to recognise apple fruits in real-time based on Support Vector Machine (SVM) classification using both colour and shape features. With this approach, they obtained an average detection rate of 89% with an average processing time of 352 ms per frame.

Although these results are promising, several problems were reported when using spectral-based sensors. One of these is the unevenness in the natural lighting which can affect the reflectance of the objects (Bulanon, Kataoka, Ota, & Hiroma, 2002). Moreover, reliance solely on natural lighting limits the usability of a machine since it would have to stop harvesting at night. The logical solution to this problem is to use artificial lighting, as was done by Rajendra et al. (2009) to detect strawberries. Besides providing stable lighting conditions, artificial illumination can also be tuned to increase the contrast between objects and background in regions of interest and suppress surrounding light scatter (Kitamura, Oka, Ikutomo, Kimura, & Taniguchi, 2008).

While accurate fruit detection can be achieved with spectral-based sensors, accurate fruit localisation using only spectral-based sensors is much less straightforward. One way to obtain depth information with spectral-based sensors is by mounting an RGB camera on an end-effector. The distance to the fruit can then be estimated based on the apparent changes in size when tracking the fruit during movement towards it (Baeten et al., 2008; De-An et al., 2011). With this approach, Baeten et al. (2008) were successful in detecting and robotic harvesting 80% of apples with an apple picking cycle of 9 s per apple on average. It was not specified which proportion of the cycle time was required for image processing and which proportion was required for moving the manipulator.

While spectral cameras can provide accurate detection in 2D space and localise targets in 3D space by tracking the fruit in a sequence of 2D spectral images, range-based devices such as laser scanners and stereo vision can provide the 3D location of the fruit immediately after correct detection without moving the camera. Also, background subtraction is simplified if depth information is available. Furthermore, obstacle detection and localisation is possible when depth information is available. However, although it is crucial to avoid collisions, obstacle localisation has received little attention in literature (Bac, Henten, Hemming, & Edan, 2014). Jiménez et al. (2000a) explain how range sensors can provide an interesting alternative for spectral-based sensors as fruit localisation in 3D space is simplified, because the detection is not influenced by shadows nor disturbed by regions with similar spectral features as the fruit of interest. Jiménez, Jain, Ceres, and Pons (1999), Jiménez, Ceres, and Pons (2000b) demonstrated that by use of only a laser range-finder it was possible to detect 80–90% of the visible fruits in one orange tree. Van Henten et al. (2003) used stereo vision and artificial light sources to detect and locate cucumbers. They realised an average harvest success rate of 74.4% for 195 cucumbers. A full harvest cycle at a fixed position of the harvest robot took on average 65.2 s per cucumber with imaging and image analysis accounting for 32% of the execution time. The use of stereo vision is a valid option to obtain 3D information. However, in orchard and greenhouse crops the localisation accuracy is often rather low due to poor feature matching (correspondence problem), since a tree consists of many similar objects. Furthermore, a high degree of occlusion can result in the presence of objects in one image without their corresponding pair being present in the other image (Jiménez et al., 2000a). Wang, Nuske, Bergerman, and Singh (2012) used a two-camera stereo rig with controlled artificial lighting for scanning apple orchards at night-time for yield estimation. They used hue, saturation and value extracted from the RGB cameras to detect red apple pixels and hue, saturation and intensity to detect green apple pixels. In the latter case, they further improved the detection accuracy by detecting specular reflections on the green apples. The detected apple pixels were then converted into individual apples using morphological operations. Finally, apples present in multiple images were merged based on their 3D positions. They reported that the algorithm made some errors in detecting visible apples, but did not provide quantitative results for the detection accuracy. However, they reported that the system obtained a relatively high crop yield estimation accuracy with estimation errors of −3.2% in the red apple block and 1.2% in the green apple block.

Several researchers have proposed combining the spectral information from spectral-based sensors with the range information from range-based sensors in order to benefit from the advantages of both sensor types. Bulanon, Kataoka, Okamoto, and Hata (2004, 2005), Bulanon and Kataoka (2010) used a CCD camera and laser ranging device mounted on an end-effector to detect and localise apples. A feedback loop centred the fruit in the middle of the RGB image by tracking and finally a laser ranging system was used to locate the fruit before automatic harvesting. In this methodology, visual tracking is required to approach the fruit and an accurate distance measurement of the fruit is possible when the end-effector is right in front of the fruit. The image processing time was less than 500 ms and for visual servoing of the end-effector towards the fruit a frame capture rate of 1 frame s^{-1} was used. Tanigaki, Fujiura, Akase, and Imagawa (2008) used a self-made 3D-vision sensor consisting of a light projector, a photo detector and a scanning device. They acquired an infrared, red and range image with a single sensor that enabled detection and localisation of cherries in 3D space. Unfortunately, they did not provide any details on the success rate and processing time. Irie, Taguchi, Horie, and Ishimatsu (2009) developed a 3D-vision sensor with two laser projectors and a sensor that enabled measurements with enough accuracy for asparagus harvesting. Wong and Lim (2012) showed how detection and localisation of spring onions was possible with a commercial time-of-flight camera. They used 2D image processing on the amplitude data together with depth analysis on the point cloud information. Fernández, Salinas, Montes, and Sarria (2014) developed a multisensory system for fruit harvesting robots consisting of a high resolution CCD colour camera, a multispectral imaging system and a Time-of-flight (TOF) 3D camera, and tested this in an apple orchard and a grapes vineyard. After registration of the different images onto each other, each pixel was classified as fruit, stem, leaf or background by means of Support Vector Machines processing based on the Red, Green and Blue channels from the RGB camera and monochrome images acquired with band-pass filters centred at 635 nm and 880 nm. They evaluated the performance of this detection system on masked images from 5 orchard scenes for the Royal Gala cultivar and reported an average precision of 99.8% on the pixel level. The fruit localisation based on the TOF data was, however, less successful with position errors up to 7.6 cm.

Recently, a type of RGB-D cameras has become commercially available that uses structured pattern illumination in the near-infrared (NIR) for the acquisition of a depth image with high spatial resolution. This depth image is consequently registered on an RGB image (Shpunt & Zalevsky, 2013). These cheap devices combine the advantages of spectral-based and range-based devices, providing information on colour, surface and shape. As such, these RGB-D sensors provide an additional approach for coping with the challenges in accurate fruit sensing. Cupec, Filko, Vidović, Nyarko, and Hocenski (2014) already proved the usefulness of an RGB-D camera for fruit sensing by detection of convex properties. However, they only used the depth images for detection of fruit which resulted in low detection accuracies with over- and under-segmentations of fruit. They suggested that additional colour and shape constraints would be necessary to improve detection.

Kapach et al. (2012) recently reviewed the state of the art in computer vision for fruit harvesting robots and emphasised the complexity of fruit detection and localisation as well as the importance of combining many different approaches to develop a robust detection. Bac et al. (2014) also described several possibilities for enhancing harvesting robots to attain better performance. One of the possibilities they suggested, is to improve the sensing aspect by advancing algorithms for novel sensors and by fusing multiple sensors. Therefore, the overall objective of this study was to develop a sensing technique for detecting and localising apple fruits on tree by means of an RGB-D camera and a multiphase algorithm that combines and interprets the recorded colour and depth information. The use of spatial depth information in combination with colour information is expected to provide fast and accurate localisation of fruit in 3D space without the need for continuous tracking. This would allow to simultaneously detect and localise all fruit (and obstacles) in a single RGB-D image of an apple tree.

The outline of this paper is as follows: In Section 2 the materials and methods are described in terms of the sensor system, the detection algorithm and the in-orchard evaluation measurements. In Section 3, the results are presented and discussed. Finally, in Section 4, conclusions are drawn and suggestions are made for further research.

2. Materials and methods

2.1. Sensor system

The sensor system used in this study was a low-cost RGB-D sensor (Xbox 360° Kinect Model 1414, Microsoft, Redmond, WA USA) consisting of an RGB camera, an infrared (IR) light source and an IR image sensor. The operating principle of the depth sensor was the projection of a known dotted IR pattern on a scene which was then captured by the IR camera. Consequently, the image processing unit of the RGB-D sensor uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image. Each pixel of the RGB camera image was registered with each pixel of the IR camera image to provide a coloured 3D point cloud. This is a set of data points in (x, y, z) coordinates with colour information (R, G, B). According to the specifications, this camera can image objects which are between 0.8 m and maximum 3.5 m away from the camera. The field of view was 57° horizontally and 43° vertically. The image size of the RGB sensor was VGA (640 × 480), while the image size of the IR sensor was 320 × 240. For the average distance to the trees of 1 m used in this research, this corresponded to a field of view of 50 cm by 70 cm with 1 mm × 1.1 mm pixels in the RGB image. According to the analysis of Khoshelham & Elberink (2012) the depth resolution at this distance is 3 mm with a theoretical random error of 1.5 mm.

2.2. Fruit detection algorithm

A custom image analysis algorithm was developed to process and interpret the coloured point clouds captured by the RGB-D sensor in order to count and localise the apples on a tree. This algorithm exploits both colour and shape properties in order to recognise the apples in a given point cloud. The implementation of the detection algorithm was based on the Point Cloud Library (PCL) (Rusu & Cousins, 2011) and was written in C++.

2.2.1. Overview of the detection pipeline

The detection pipeline shown in Fig. 1 represents the detailed procedure of the algorithm schematically. The detection algorithm consists of three phases. The first phase, pre-processing of the point cloud, comprises two filtering steps: a distance filter and a colour filter. The goal was to select all data points with a red colour in the distance range corresponding to the targeted row. These data points have a high probability to belong to apples. The second phase, clustering segmentation, used the Euclidean clustering algorithm to segment the remaining ‘red’ point cloud into multiple clusters of which each cluster is presumed to belong to a single apple. The last phase, post-processing, aimed at resolving the problem of clusters which contain multiple apples by separating these clusters, and the problem of single apples which are distributed over multiple clusters by combining these clusters. The post-processing step was also used to estimate the location and diameter of each fruit. Estimation of apple 3D orientation was not the purpose of this detection pipeline as the data of the RGB-D sensor were not detailed enough for this purpose. In the following paragraphs, each phase of the image analysis algorithm is discussed in more detail.

2.2.2. Pre-processing: pruning of the point cloud

The goal of the pre-processing phase was to preserve only the interesting (apple) data points from the acquired 3D point clouds (Fig. 2a) by removing data points corresponding to leaves, trunks and branches, such that the overall processing time of the detection procedure was significantly reduced. More specifically, the pre-processing of the point clouds relies on two filters: a distance filter and a colour filter. The distance filter was based on the spatial information in (x, y, z) of the point cloud and reduces the point cloud to the data points which belong to the nearest row based on predefined distance boundaries. These boundaries were defined as such that only those fruits are detected which hang on the side of the tree row where the camera was positioned. This was important to avoid double counting of the same apples when imaging from both sides of the row. Furthermore, this also prevented a picking robot from harvesting apples on the other side of the tree row which could result in damaging the supporting system of the trees (metal wires) and the robot manipulator itself. The effect of the distance filtering on the point clouds is illustrated in Fig. 2b.

Redness is one of the properties used by humans to detect a red or bicoloured apple on a tree. So, it is logical to also use this property for automatic apple detection. The colour filter method proposed by Bulanon et al. (2002) for segmenting apple fruit from the background was used in this research. In this approach, two coefficients r and g represent the normalised red and green features, respectively:

$$r = \frac{R}{R+G+B} \tag{1}$$

$$g = \frac{G}{R+G+B} \tag{2}$$

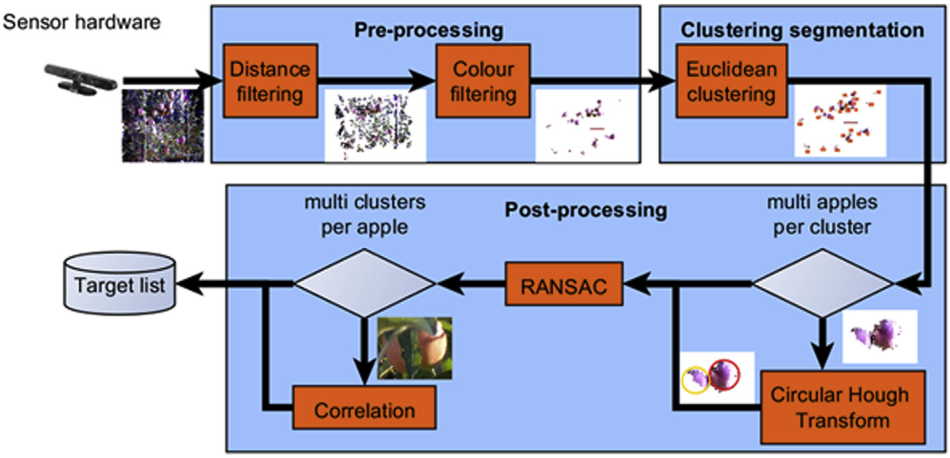

Fig. 1 – Pipeline of the detection algorithm.

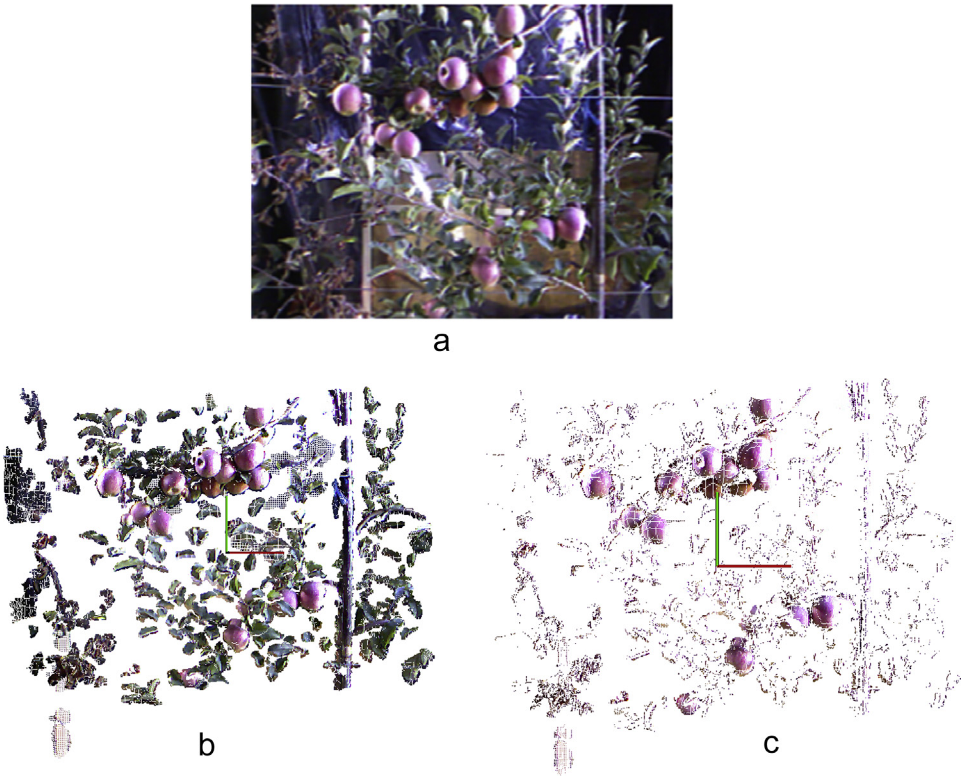

Fig. 2 – Illustration of the effects of distance based filtering and colour filtering on a point cloud acquired in a Fuji apple orchard: (a) acquired point cloud; (b) after distance filtering; (c) after distance and colour filtering.

Two decision functions, d_1 and d_2 that separate the fruit from other classes in the feature space were defined by the following equation for which the values have been tuned by trial and error:

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0.09 & -0.11 \\ 0.12 & -0.06 \end{bmatrix} \begin{bmatrix} r \\ g \end{bmatrix} \tag{3}$$

These decision functions were applied to the depth filtered point cloud data $\mathbf{P}_d$ to determine whether each point $p_i \in \mathbf{P}_d$ has a high enough value in red r and a low enough value in green g ($d_1^i > 0$ *and* $d_2^i > 0$) to be allowed as a member of the colour filtered point cloud $\mathbf{P}_r$. The effect of this colour filtering step on the distance filtered point clouds is illustrated in Fig. 2c.

2.2.3. *Point cloud clustering segmentation*

Point cloud clustering segmentation was carried out in the main part of the fruit detection algorithm. In this technique, Euclidean clustering was applied to segment the depth and colour filtered point cloud $\mathbf{P}_r$ into multiple clusters $\mathbf{P}^i_{apple}$ which each were presumed to belong to an apple with number i. More specifically, a point cloud cluster $\mathbf{P}^i_{apple}$ was defined as follows:

Let $\mathbf{P}^i_{apple} = \{p_i \in \mathbf{P}_r\}$ be a distinct point cluster $\mathbf{P}^j_{apple} = \{p_j \in \mathbf{P}_r\}$ if:

$$\min p_i - p_j \geq d_{th} \tag{4}$$

where d_{th} is a maximum imposed distance threshold. The above equation states that if the minimum distance between a set of points $p_i \in \mathbf{P}_r$ and another set $p_j \in \mathbf{P}_r$ is larger than a predefined distance value d_{th}, then the points in p_i are set to belong to a point cluster $\mathbf{P}^i_{apple}$ and the ones in p_j to another distinct point cluster $\mathbf{P}^j_{apple}$ (Rusu, 2009). The estimation of the minimum distance d_{th} between the two sets can be done with an approximate nearest neighbours query via kd-tree representations (Bentley, 1975). A kd-tree is a data structure used for organising points in a space with k dimensions. This type of data structure enables a fast search for neighbouring points. The detailed steps of the algorithm are described by Rusu (2009) as follows:

- Create a kd-tree representation for the colour filtered point cloud data set $\mathbf{P}$
- Set up an empty list of clusters $\mathbf{C}$, and a queue of the points that need to be checked $\mathbf{Q}$
- For every point $p_i \in \mathbf{P}$, perform the following steps:
  - ○ Add p_i to the current queue $\mathbf{Q}$
  - ○ For every $p_i \in \mathbf{Q}$:
    - Search for the set $\mathbf{P}^k_i$ of point neighbours of p_i in a sphere with radius $r < d_{th}$
    - For every neighbour $p^k_i \in \mathbf{P}^k_i$, check if the point has already been processed, and if not add it to $\mathbf{Q}$
  - ○ When the list of all points in $\mathbf{Q}$ has been processed, add $\mathbf{Q}$ to the list of clusters $\mathbf{C}$, and reset $\mathbf{Q}$ to an empty list
- The algorithm terminates when all points $p_i \in \mathbf{P}$ have been processed and are now part of the list of point clusters $\mathbf{C}$.

When this methodology was applied to the colour filtered point cloud $\mathbf{P}_r$ from Fig. 2c, the proposed algorithm constructed a set of separated apple clusters (Fig. 3a) which each were assigned a different label, as illustrated in Fig. 3b. The cluster with the lowest number of points was assigned label 0. The lower the number of points in a cluster the higher the cluster label. The parametric inputs for the Euclidean algorithm were tuned by trial and error on the training images at 3 mm cluster tolerance, d_{th}, and a minimum cluster size of 165 points, because these settings resulted in removal of most of the points not related to an apple which were still present in the data set after performing the colour filtering, while maintaining most apple clusters.

Although the Euclidean clustering algorithm is well able to detect the apples, it cannot deal with a situation in which multiple apples hang close together in a bunch, as shown in Fig. 4. When two or more apples are touching or hanging very close to each other, the minimum distance between the data points belonging to the different apples is less than the distance threshold d_{th}. Therefore, the bunch of apples will be recognised as a single cluster. This leads to an underestimation of the number of apples on a tree. Additionally, this effect produces an error in the localisation of the individual apples which in turn could lead to an unsuccessful picking attempt from a robot. Another situation in which the Euclidean clustering algorithm often fails is in the case of an occlusion of the middle part of the apple which leads to a segmentation of the points belonging to one apple into two different clusters. To overcome these limitations of the clustering algorithm, a post-processing step has been included in the algorithm.

2.2.4. *Post-processing phase*

After the clustering step, the point cloud data have been separated into multiple clusters. Each cluster is presumed to belong to a single apple fruit. The post-processing phase of the detection algorithm has three functions:

1. Resolving the problem of single clusters that include multiple apples by means of the circular Hough transformation (CHT) algorithm
2. Resolving the problem of single apples that are divided over multiple clusters by means of the **ran**dom **s**ample **c**onsensus (RANSAC) algorithm
3. Estimating the location and diameter of each fruit by means of the RANSAC algorithm

To separate multiple apples that are present within a single cluster, a simple apple size estimation procedure was applied to all the detected clusters. The maximum distance between two points in each detected cluster was calculated. If the calculated distance was larger than the diameter of an oversized apple D_{max}, then it was assumed that the cluster contained data points of more than one apple. This algorithm works as follows:

```
for every cluster $P^n_{apple}$ do
    if $\max\|p_i - p_j\| \geq D_{max}$ then
        $P^n_{apple}$ consists more than one apple;
    end
end
```

Clusters selected by this procedure were subsequently converted to a 2D image based on the location of each point in the point cloud. A circle fitting algorithm was then applied to the binary image to detect regions which represent individual apples. Firstly, the allowable range for the radius of the circles in the 2D image was estimated based on a predefined range of expected fruit diameters and the measured distance between the camera and the cluster. The core of the circle fitting algorithm was the circular Hough transformation (CHT) (Laganière, 2011). The effect of this algorithm is illustrated in Fig. 4c where the multi-apple cluster from Fig. 4b has been correctly separated into two separate apple clusters.

In a final step, the RANSAC algorithm was applied to the detected apple clusters to estimate the location of the

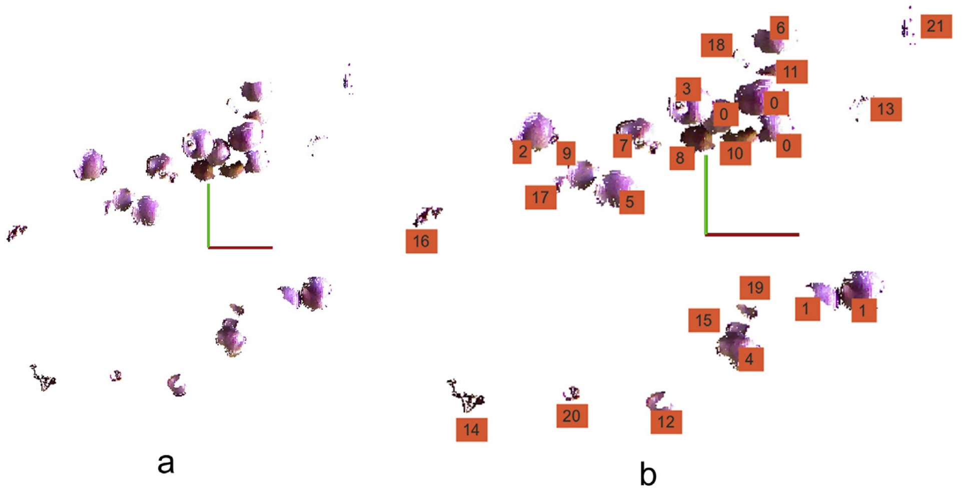

Fig. 3 – Illustration of different steps of Euclidean clustering segmentation of the distance and colour filtered point cloud: a) detected clusters, b) labelled clusters.

centre of gravity and the diameter of each fruit. The estimated centre of gravity of the fruit was defined as (x_d,y_d,z_d), the estimated diameter of the fruit was defined as d_d. RANSAC (Fischler & Bolles, 1981) is an iterative approach for model recognition that is used to estimate parameters of a predefined mathematical model from a data set containing outliers. The RANSAC algorithm assumes that the data set is comprised of both inliers and outliers. Inliers can be explained by a model with a particular set of parameter values, while outliers do not fit that model under any circumstance. The algorithmic steps of the RANSAC procedure are as follows:

1. Randomly select a sample of s data points from $\mathbf{S}$ and instantiate the model from this subset.
2. Select the set of data points $\mathbf{S}_i$ within a distance threshold t_{th} of the model. The set $\mathbf{S}_i$ is the consensus set of the sample and defines the inliers of $\mathbf{S}$.
3. If the size of $\mathbf{S}_i$ is greater than a threshold $\mathbf{T}$, re-estimate the model using all the points in $\mathbf{S}_i$ and terminate.
4. If the size of $\mathbf{S}_i$ is less than $\mathbf{T}$, select a new subset and repeat.
5. After N trials the largest consensus set $\mathbf{S}_i$ is selected, and the model is re-estimated using all the points in the subset $\mathbf{S}_i$.

In the apple detection algorithm presented here, a simple sphere was selected as the model to be fitted to the apple clusters such that the parameters to be estimated were the 3D centre location and the radius of the sphere. The parametric inputs for the RANSAC algorithm were chosen as follows: The maximum number of iterations N was set to 100, while the distance threshold t was set to 1 mm and the threshold $\mathbf{T}$ was not specified. By means of the outputs obtained for the location and radius of the sphere, it was checked whether two or more clusters contain data points originating from the same apple. More specifically, this was done by comparing the total radius of the multi-apple cluster with the distance between the centres of the two spheres. If the total radius was greater than this distance, the clusters were considered to belong to the same apple and were merged.

2.3. *Evaluation measurements*

2.3.1. *Working environment*

Apples are typically grown outdoors in an orchard. In order to facilitate robotic harvesting, suitable fruit varieties and planting methods were studied. This resulted in the selection of a fruit wall of espaliered trees in which the apples are situated close to the trunk or close to the main branches. This type of orchard was planted in Santa Rosa, Chile (−36:528°N, −71:910°E). Four-year-old ‘Gala’ and ‘Fuji’ trees were examined in this orchard (Fig. 5). Single trunk trees were trained to grow in a flat plane and were supported by a trellis system of wires and poles. By using the espaliered tree row with short pruned lateral branches, the occlusion of the fruits on a tree was significantly reduced when compared to other growing systems with a less trained canopy. Apples are defined here as being occluded when more than 50% of the apple surface, observed from one viewpoint, is not visible to an observer. The percentage of occluded fruits was quantified to be 20% in the Chilean trial orchard. As the depth measurement by the Kinect relies on a pattern projection in the NIR and the reliability and performance of many computer vision techniques depends on the lighting conditions, a light shield was constructed to block direct sunlight. This construction consisted of black plastic foil supported by a wooden frame of 3.20 m × 3.20 m × 2.00 m which was placed over the tree rows, as illustrated in Fig. 6. No artificial illumination was added and thus only indirect sunlight was used during the data acquisition process.

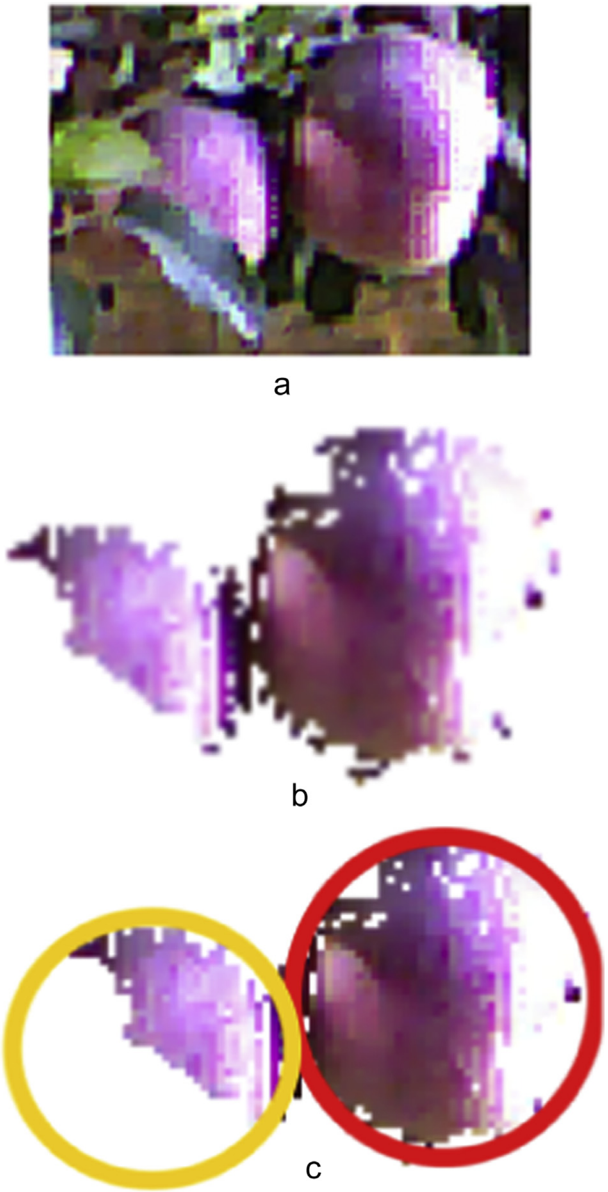

Fig. 4 – Illustration of the post-processing phase of the algorithm for separating a multi-apple cluster identified by the Euclidean clustering algorithm into separate apple clusters (circles with different colour: a) original image, b) multi-apple cluster, c) separation of apple clusters with CHT.

2.3.2. *Data collection*

To evaluate the accuracy of the detection algorithm, 2 datasets of coloured point clouds were acquired from the trees in the testing orchard in Chile. The RGB-D sensor was placed about 1 m away from the tree row and pointed straight to the tree row.

- The first dataset contained 5 point clouds captured from 5 different fruit trees with red-coloured Gala apples. On every tree, 6 apples were labelled manually and made completely visible by removing all leaves and branches in front of them. The pose and diameter ground truth was measured for each labelled apple. The explanation of the ground truth measurements is given at the end of this section. All other apples were left untouched. In total, this dataset contained 86 apples and was used to evaluate the recognition accuracy for both completely visible apples and partially occluded apples.
- The second dataset contained 10 point clouds of 10 different Fuji apple trees. No changes were made to the trees. 6 apples per tree were labelled manually and the pose and diameter ground truth was measured for these labelled apples. This data set contains 225 apples and was used for evaluating the recognition and the localisation accuracy in an occluded environment.

For measuring the pose ground truth data, a 3D Guidance trakSTAR (Ascension Technology Corporation, Milton, VT, USA) was used. This is an electromagnetic tracker designed for short-range motion tracking with small, lightweight sensors. According to the user manual, the static accuracy is 1.4 mm RMS for position and 0.5° for orientation. By attaching these sensors on objects, it was possible to track the full 3D pose of investigated objects (6 degrees of freedom: x_g,y_g,z_g, azimuth, elevation and roll). The pose of a trakSTAR sensor is measured relative to a fixed transmitter.

The ground truth pose measurements included reference measurements of the pose of both the six labelled apples and the RGB-D camera for all 15-point cloud measurements.

In Fig. 7 the experimental setup in the orchard is illustrated. The left part of Fig. 7 illustrates how the trakSTAR reference transmitter was placed in the middle between the RGB-D camera and a target tree in order to receive clear signals from the trakSTAR sensors. The centre of Fig. 7 shows how one trakSTAR sensor was attached to the RGB-D camera to measure the pose of the camera. The x-direction of the sensor faced in the same direction as the RGB-D camera. The right part of Fig. 7 illustrates how a trakSTAR sensor was manually aligned with the orientation of the apple. The pose of the apple was measured by manually holding trakSTAR sensors on two opposite sides of the apple (once at the front of the apple and once at the back). The mean of the two position measurements from the opposite sides of the apple resulted in the position of the centre of the apple. The orientation of an apple was measured by aligning the x-axis of the trakSTAR sensor with the stem-calyx axis of the apple. Because orientation detection was not implemented in the detection algorithm, the orientation ground truth data was not used for evaluation in this study.

The diameter ground truth measurements were performed with a calliper. As apples are not perfectly spherical, both the longitude and transverse diameters were measured. The longitude diameter ld_g was defined as the diameter along the main stem-calyx axis of the apple, the transverse diameter td_g was defined as the diameter perpendicular to this axis.

2.4. *Performance evaluation*

The fruit recognition and localisation algorithm was evaluated on the collected datasets. The performance of the algorithm was quantified in terms of the recognition accuracy, the localisation

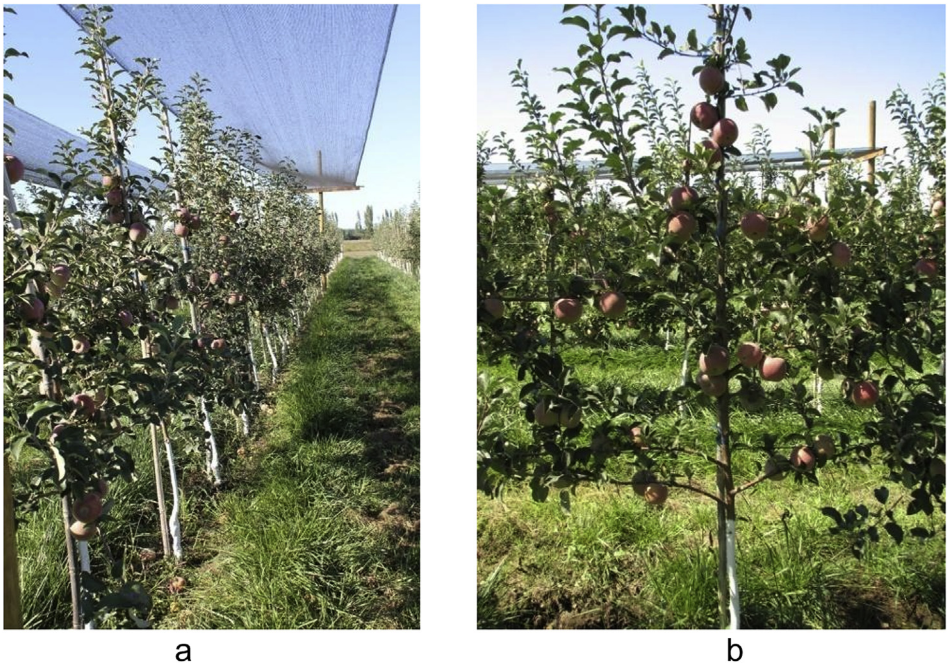

Fig. 5 – Test orchard with espaliered apple trees.

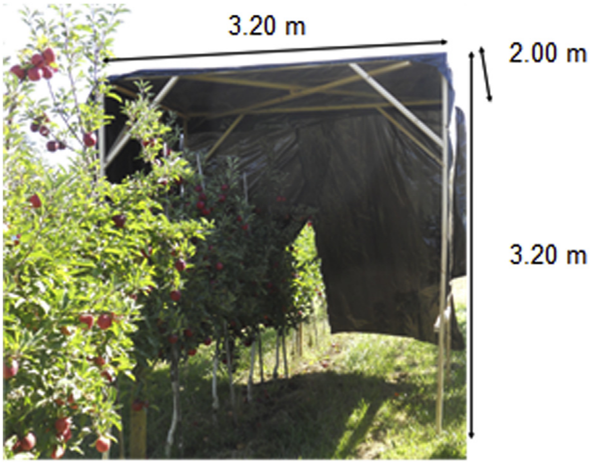

Fig. 6 – Light shield construction in the testing orchard.

accuracy, the diameter estimation accuracy and the processing time. The recognition accuracy was quantified both in terms of the true positive detection rate (TPR), defined as the percentage of correctly detected apples over the number of apples in the scene, and the false positive detection rate (FPR), defined as the percentage of wrongly detected apple clusters over the total number of counted apples in a scene. The localisation accuracy was quantified as the mean (μ_x, μ_y, μ_z) and corrected sample standard deviation (s_x, s_y, s_z) of difference errors between the position (x_d, y_d, z_d) estimated by the detection algorithm and the ground truth position (x_g,y_g,z_g) measured with the trakSTAR sensor. For instance μ_x and s_x are defined as:

$$\mu_x = \frac{\sum_{i=1}^{N}(x_{d,i} - x_{g,i})}{N} \tag{5}$$

$$s_x = \sqrt{\frac{\sum_{i=1}^{N}\left(\left(x_{d,i} - x_{g,i}\right) - \mu_x\right)^2}{N-1}} \tag{6}$$

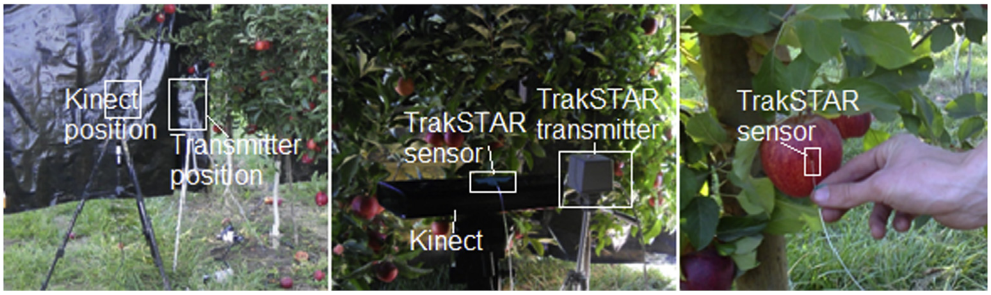

Fig. 7 – Illustration of the setup used for ground truth data acquisition of the pose of the RGB-D camera and of the pose of the labelled apples.

with i the number of the measured apple and N the number of apples in the point cloud dataset. The diameter estimation accuracy was determined by comparing the estimated diameter given by the detection algorithm (d_d) and the (ground truth) measured transversal and longitudinal diameters (td_g,ld_g) of the labelled apples. Quantification of the accuracy was done by calculating the mean (μ_{td}, μ_{ld}) and corrected sample standard deviation (s_{td}, s_{ld}) of the difference errors of estimated and measured transversal and longitudinal diameters. For instance μ_{td} and s_{td} are defined as:

$$\mu_{td} = \frac{\sum_{i=1}^{N}\left(d_{d,i} - td_{g,i}\right)}{N} \tag{7}$$

$$s_{td} = \sqrt{\frac{\sum_{i=1}^{N}\left(\left(d_{d,i} - td_{g,i}\right) - \mu_{td}\right)}{N-1}} \tag{8}$$

The processing time for the algorithm was approximated by calculating the average running times of the different steps of the algorithm for the 15 point clouds and summing these. An Intel®Core™ i5 2.5 GHz × 4 with 4 GB memory was used in the time efficiency analysis.

3. Results and discussion

3.1. *Apple recognition results*

In Table 1, the recognition performance of the apple detection algorithm is summarised for the Gala dataset discussed in Section 2.3. For this dataset, the TPR was 88% over all fruits in the scene which were at least partially visible. For the completely visible fruits, the TPR was 100%, while the TPR for the partially occluded fruits was only 82%. An analysis of the un-recognised fruits revealed that these were the highly occluded fruits and the fruits located at the edge of an image of which only a small area of the surface was visible. It is expected that these fruits could be detected more accurately from another viewpoint.

The recognition performance of the algorithm on the Fuji dataset is summarised in Table 2. On this dataset, a TPR of 81% was obtained. Only two false detections were observed resulting from objects with a similar colour as the apples. The lower detection performance on the Fuji dataset might be explained by the fact that there were more occluded apples in the Fuji orchard (22%) than in the Gala orchard (16%). Moreover, at the moment of the data acquisition the Gala apples were more ripe and had more red colour than the Fuji apples.

Table 1 – Recognition performance of the algorithm in Gala trees.

Tree	Total fruits	True positive (TPR)	False positive (FPR)
1	26	23	0
2	9	8	0
3	16	13	0
4	16	16	0
5	19	18	0
Total:	*86 (100%)*	*76 (88%)*	*0 (0%)*

Table 2 – Recognition performance of the algorithm in Fuji trees.

Tree	Total fruits	True positive (TPR)	False positive (FPR)
1	20	17	1
2	28	23	1
3	24	18	0
4	25	19	0
5	24	21	0
6	25	20	0
7	24	19	0
8	26	21	0
9	7	6	0
10	22	18	0
Total:	*225 (100%)*	*182 (81%)*	*2 (0%)*

3.2. *Localisation and diameter estimation results*

For the Gala dataset, the localisation ground truth was measured for 30 apples for which all occlusions had been removed. For the Fuji dataset the ground truth was measured for 60 apples where no occlusions had been removed. The accuracies for localisation and diameter estimation (transversal and longitudinal) are summarised in Table 3. The accuracies in Table 3 have been calculated as described in Section 2.4 as mean (μ_x, μ_y, μ_z, μ_{td}, μ_{ld}) and corrected sample standard deviation (s_x, s_y, s_z, s_{td}, s_{ld}) of the difference errors of estimated and measured positions (Δx,Δy,Δz) and diameters (Δtd,Δld). The difference errors Δx, Δy and Δz correspond to ($x_d - x_g$), ($y_d - y_g$) and ($z_d - z_g$), while the difference errors Δtd, Δld correspond to ($td_d - td_g$) and ($ld_d - ld_g$). As shown in Table 3, the location errors for the detected apples were slightly larger in the depth direction and less than 10 mm in all directions. This low location error indicates that the detection system will be well suitable for robotic fruit harvesting if the end-effector has at least 10 mm tolerance.

3.3. *Time efficiency analysis*

As a harvesting robot senses and plans before acting, detection does not have to be performed in real-time. However, as it contributes additively to the total cycle time for picking a fruit, apple detection should not take more than a second. In Table 4, the average execution times are summarised for the different processing steps.

By combining the average processing time of 1 s per scene with the average number of 20.7 apples per scene, the detection time per apple was around 50 ms. As harvesting one apple with a robotic system will at least take a few seconds, this

Table 3 – Mean (μ_x, μ_y, μ_z, μ_{td}, μ_{ld}) and corrected sample standard deviation (s_x, s_y, s_z, s_{td}, s_{ld}) of the difference errors of measured and estimated positions (Δx,Δy,Δz) and diameters (Δtd,Δld) in mm.

		Δx	Δy	Δz	Δtd	Δld
Fully visible (N = 30)	μ	−4	−4	7	−6	13
	s	4	4	6	5	8
Partially visible (N = 60)	μ	−5	−5	8	−8	−15
	s	6	8	9	6	8

Table 4 – Average processing time for the fruit detection algorithm calculated over the 15 point clouds used for the performance evaluation in this study.

Phases	Functions	Processing time (ms)
Pre-processing	Distance filtering	13
	Colour filtering	142
Segmentation	Euclidean clustering	375
Post-processing	CHT	136
	RANSAC	298
Total		964

means that the detection should constitute only a very small part of the total picking cycle. Therefore, it can be concluded that the proposed algorithm is suitable for integration in a harvesting robot.

3.4. *Discussion*

While the detection accuracy and processing time of the proposed algorithm were suitable for use in an apple harvesting robot, there is still room for further improvements. It would be interesting to retrieve additional information from the point clouds such as the 3D orientation of the fruit, position of the stalk and the pose of leaves. These additional parameters would be useful to optimise grasping and picking operations (Kapach et al., 2012) or finding occlusion-free viewpoints of the fruit. However, the RGB-D camera used in this study requires a minimal distance of 0.8 m from the object. Therefore, the point clouds were not detailed enough to retrieve the fruit orientation or the position of the stalk. Leaf pose estimation was also not investigated for the obtained point clouds. A close-range camera would certainly be a useful addition for the RGB-D camera in order to retrieve additional close-range sensing information about target fruit. An example of orientation estimation and stem detection for strawberry harvesting was presented by Feng, Qixin, and Masateru (2008). It is recommended to perform similar work in apples to provide pose input to a harvesting robot. However, this is not an easy task as apples are almost spherical and the stalk of an apple on a tree is for most of the time not visible. One possibility for pose estimation could be to use the apple calyx which is often visible. When the position of the apple calyx and apple centre could be correctly estimated, it would be possible to calculate the apple orientation and estimate the stem location.

4. Conclusions and future perspectives

In this study, an algorithm has been proposed and evaluated for detection and localisation of red and bicoloured apples on tree in the orchard based on colour and range data captured with an RGB-D camera system under a light shield blocking direct sunlight. The algorithm achieved 100% correct detection of all fully visible apples and more than 80% correct detection for partially occluded apples with respectively 0% and 1% false detections. The errors for the localisation were less than 10 mm in all directions. With an average image processing time of 50 ms per apple, the algorithm proved to be very efficient. This combination of high detection and localisation accuracy with a short processing time makes this algorithm very suitable to be used on a harvesting robot. However, additional closer sensing will be necessary to provide information on the 3D pose of the fruit and the position of the stalk in order to optimise grasping and to optimise the picking motion for individual fruit.

Due to the use of the red colour filter the presented algorithm is limited to the detection of red and bicoloured apples. To be able to also detect green apples, the colour filtering should be improved or replaced by an alternative filtering strategy. This might be achieved by using a machine learning method which can classify colour regions in the point cloud rather than individual pixels and/or uses other features like more detailed spectral information, object textures or surface smoothness.

However, occlusion of the fruits by leaves, branches, or other fruits remains the main challenge for detecting apples on tree in the natural environment. Therefore, future studies should focus on the combination of multiple viewpoints as well as other options to reduce the effects of occlusions.

Acknowledgements

This research was funded by the European Commission in the 7th Framework Programme (CROPS GA no.246252). The opinions expressed in this document do by no means reflect the official opinion of the European Union or its representatives.

The authors gratefully acknowledge the support of the colleagues of INIA (Chile): Dr. Stanley Best, Lorenzo Leon and Fabiola Flores.

REFERENCES

Bac, C. W., Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888–911.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., & Claesen, E. (2008). Autonomous fruit picking machine: a robotic apple harvester. In *Field and service robotics* (pp. 531–539). Springer Berlin Heidelberg.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM, 18*(9), 509–517.

Bulanon, D. M., & Kataoka, T. (2010). Fruit detection system and an end effector for robotic harvesting of Fuji apples. *Agricultural Engineering International: CIGR Journal, 12*(1).

Bulanon, D. M., Kataoka, T., Okamoto, H., & Hata, S. (2004). Development of a real-time machine vision system for the apple harvesting robot. In *SICE 2004 Annual Conference* (Vol. 1, pp. 595–598). IEEE.

Bulanon, D. M., Kataoka, T., Okamoto, H., & Hata, S. (2005). *Feedback control of manipulator using machine vision for robotic apple harvesting*. ASAE Paper, 53114.

Bulanon, D. M., Kataoka, T., Ota, Y., & Hiroma, T. (2002). A color model for recognition of apples by a robotic harvesting system. *Journal of the Japanese Society of Agricultural Machinery (JSAM), 64*(5), 123–133.

Cupec, R., Filko, D., Vidović, I., Nyarko, E. K., & Hocenski, Ž. (2014). Point cloud segmentation to approximately convex surfaces for fruit recognition. In *Croatian Computer Vision Workshop, Year 2*.

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems Engineering, 110*(2), 112–122.

Edan, Y., Rogozin, D., Flash, T., & Miles, G. E. (2000). Robotic melon harvesting. *Robotics and Automation, IEEE Transactions on, 16*(6), 831–835.

Feng, G., Qixin, C., & Masateru, N. (2008). Fruit detachment and classification method for strawberry harvesting robot. *International Journal of Advanced Robotic Systems, 5*(1), 41–48.

Fernández, R., Salinas, C., Montes, H., & Sarria, J. (2014). Multisensory system for fruit harvesting robots. Experimental testing in natural scenarios and with different kinds of crops. *Sensors, 14*, 23885–23904.

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

Irie, N., Taguchi, N., Horie, T., & Ishimatsu, T. (2009). Asparagus harvesting robot coordinated with 3-D vision sensor. In *Industrial Technology, 2009. ICIT 2009. IEEE International Conference on* (pp. 1–6). IEEE.

Jiménez, A. R., Ceres, R., & Pons, J. L. (2000a). A survey of computer vision methods for locating fruit on trees. *Transactions of the ASAE-American Society of Agricultural Engineers, 43*(6), 1911–1920.

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

Jiménez, A. R., Jain, A. K., Ceres, R., & Pons, J. L. (1999). Automatic fruit recognition: a survey and new results using range/attenuation images. *Pattern Recognition, 32*(10), 1719–1736.

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Computers & Electrical Engineering, 38*(5), 1186–1195.

Kapach, K., Barnea, E., Mairon, R., Edan, Y., & Ben-Shahar, O. (2012). Computer vision for fruit harvesting robots–state of the art and challenges ahead. *International Journal of Computational Vision and Robotics, 3*(1), 4–34.

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors, 12*(2), 1437–1454. http://dx.doi.org/10.3390/s120201437.

Kitamura, S., Oka, K., Ikutomo, K., Kimura, Y., & Taniguchi, Y. (2008). A distinction method for fruit of sweet pepper using reflection of LED light. In *SICE Annual Conference, 2008* (pp. 491–494). IEEE.

Kondo, N., & Ting, K. C. (1998). *Robotics for bioproduction systems*. American Society of Agricultural Engineers (ASAE).

Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision*. Packt Publishing Ltd.

Li, P., Lee, S. H., & Hsu, H. Y. (2011). Review on fruit harvesting method for potential use of automatic fruit harvesting systems. *Procedia Engineering, 23*, 351–366.

Rajendra, P., Kondo, N., Ninomiya, K., Kamata, J., Kurita, M., Shiigi, T., et al. (2009). Machine vision algorithm for robots to harvest strawberries in tabletop culture greenhouses. *Engineering in Agriculture, Environment and Food, 2*(1), 24–30.

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

Rusu, R. B., & Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 1–4). IEEE.

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

Shpunt, A., & Zalevsky, Z.. (2013). U.S. Patent No. 8,390,821. Washington, DC: U.S. Patent and Trademark Office.

Tabb, A. L., Peterson, D. L., & Park, J. (2006). *Segmentation of apple fruit from video via background modeling*. ASABE paper, (063060).

Tanigaki, K., Fujiura, T., Akase, A., & Imagawa, J. (2008). Cherry-harvesting robot. *Computers and Electronics in Agriculture, 63*(1), 65–72.

Van Henten, E. J., Van Tuijl, B. V., Hemming, J., Kornet, J. G., Bontsema, J., & Van Os, E. A. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering, 86*(3), 305–313.

Wang, Q., Nuske, S., Bergerman, M., & Singh, S. (2012). Automated crop yield estimation for apple orchards. In *Proc. International Symposium on Experimental Robotics, June 2012, Quebec City*.

Wong, C. K., & Lim, P. P. (2012). Processing of point cloud data from ToF camera for the localisation of ground-based crop. In *Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference* (pp. 184–189). IEEE.

Zhao, J., Tow, J., & Katupitiya, J. (2005). On-tree fruit recognition using texture properties and color data. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on* (pp. 263–268). IEEE.