Determining Dendrometry Using Drone Scouting, Convolutional Neural Networks and Point Clouds

Kim Jensen§, Oskar Kondrup Krogh§, Marius Willemoes Jørgensen§
Daniel Lehotský§, Anton Bock Andersen§, Ernest Porqueras§
Department of Electronic Systems, Aalborg University, Aalborg, Denmark
komadaau@gmail.com
§ Equal contribution

Jens Aksel S. Søndergaard
TeeJet Technologies
Aabybro, Denmark
JensAkselS@TeeJet.com

Rikke Gade
Section of Media Technology
Aalborg University, Aalborg, Denmark
rg@create.aau.dk

Abstract

This paper presents a solution for mapping the location of trees in an orchard and estimating the dendrometric data of the trees. The combined solution consists of a mapping and navigation algorithm, which allows for autonomous data collection at an orchard with a regular rectangular layout, and data processing for tree detection and dendrometric data estimation. The data collection is done using an Intel RealSense D435i camera, which can obtain both RGB and depth data. The paper presents a comparison between the performance of point cloud processing (PCP) and convolutional neural networks (CNNs) on RGB data for tree detection and dendrometric data estimation. The YOLOv3 CNN achieved a mAP_{50} *of 63.53% with 65.5 FPS and a mean error of 20.6 cm in height estimation. Point cloud processing achieved a precision of 76.72% with 2.1 FPS and a mean error of 20.4 cm in height estimation. In conclusion, this work shows that point cloud processing shows comparable results to convolutional neural networks for height estimation, but trades off processing time for better precision in detection.*

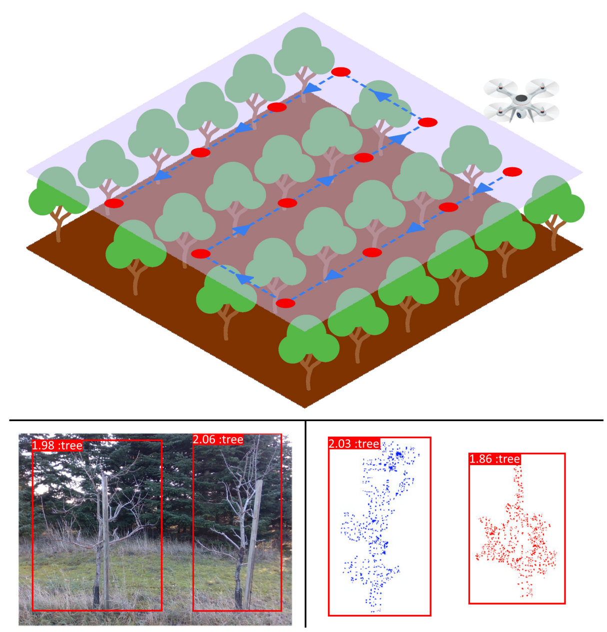


Figure 1: Figure showing the three problems handled in this paper: Automated mapping and data collection, and tree detection and height estimation in RGB images and point clouds, respectively.

1. Introduction

Pesticide spraying is an essential part of agriculture today, however, excessive spraying has both environmental and economic costs [1]. Pesticide usage in citrus orchards in Spain cover 30% of the total cost, while it accounts for 42% in olive orchards [13, 23]. The field of Precision Agriculture (PA) aims to combat this. PA covers many different aspects of farming, such as field-, soil-, and crop-variability [24]. Tracking changes on the farm accurately and in small-scale allows the farmer to vary the amount of fertiliser, pesticide, or herbicide used, to the needs of individual plants or individual areas of the farm. By reducing wasted product, PA can greatly decrease the costs of farming [7]. PA is currently used on 30-50% of corn and soybean acres in the US [21]. One way to automate PA is to use UAVs or drones. There exists multiple solutions using GPS to control a drone, to either survey or spray a field of crops [12]. In apple orchards, the dendrometry of the trees are important to determine how much water and pesticides each tree needs [18]. This paper proposes using a drone to autonomously map the location of apple trees and collect dendrometry data, using neural networks on RGB data and point cloud processing. The proposed solution is intended to increase the level of information gathered during automated PA, potentially further reducing the use of pesticides in agriculture. The main contributions of this paper are: (i) Performance comparison of point cloud processing and convolutional neural networks for both detection and height estimation of young apple trees in an orchard. (ii) Simulated proof of concept for visualisation of the automated mapping and data gathering solution proposed by this paper. Figure 1 illustrates the automated mapping process and the tree detection and height estimation in RGB images and point clouds.

2. Related work

In the field of horticulture there has been introduced a wide variety of robotic and automatic solutions for tasks such as fruit harvesting, yield estimation, and pest control. Zhang et et al. describes how different solutions take advantage of sensors such as LiDARs and RGB cameras, which gather point cloud data and RGB data respectively [25]. Point cloud data is more used for estimation of dendrometric data and RGB data is often used in convolutional neural networks (CNNs) for detection and classification.

Dyrmann et al. proposed a deep neural network capable of classifying 22 different plants and crops in early growing stages with a 86.2% precision [6]. Bodwhani et al. developed a deep neural network based on the ResNet50 framework for classifying and distinguishing between 180 classes of trees with an accuracy of 93.09% [3]. Iman Saedi and Hossein Khosravi created a deep neural network for real-time on-branch fruit detection. They achieved a recognition accuracy of 99.76% with an average time per frame of 8.09 ms [20]. However, while neural networks have been widely integrated for detection and classification in the field of horticulture, it is not commonly used for the estimation of dendrometry information.

Point cloud processing (PCP) on the other hand is not widely used in the field of horticulture, however, research has been done in the area of dendrometry estimation of trees using point cloud processing methods. Bienert et al. developed a method for detecting trees and obtaining dendrometry information such as diameter at breast height, and height of the stem by processing the data obtained from terrestrial laser scanning. They achieved a detection rate of 97.4%, a diameter estimation with a standard deviation of 2.48 cm, and a predicted trunk taper with 1.36 cm standard deviation [2]. Similar work was conducted by Cabo et al. who also proposed an algorithm for terrain smoothing. They achieved a detection rate of 99% after applying terrain smoothing and a root mean square error ranging from 0.8-1.3 cm for the estimation of diameter at breast height and 0.3-0.7 m for tree height estimation [4].

Despite the high detection rates and relatively accurate estimations of dendrometric data using the point cloud processing, these proposed methods are not applicable to widely available spraying solutions, because they are based on expensive terrestrial laser scanners [14]. Additionally, the data gathering procedures described in Bienert et al. and Cabo et al. are based on manually moving the sensor to capture the objects of interest [2, 4]. This is not considered to be an option for all horticulture applications, hence this paper will concern itself with automatic data gathering using a drone. Different models of drones provide the opportunity of mounting external devices, *e.g.*, cameras or infrared sensors, which can be used for data gathering.

Cameras such as the Intel RealSense are cheap alternatives for point cloud gathering and could potentially be used for detection and dendrometric data estimation in a spraying solution [5].

Therefore, the objective of this paper is to explore the possibilities of apple tree detection and dendrometry estimation at an orchard by comparing results on RGB and depth data.

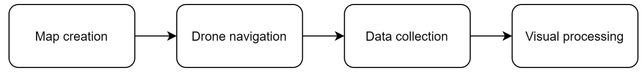


Figure 2: Simple block diagram of the solution structure.

3. General method

The proposed method consists of four different components - mapping, drone navigation, data collection, and processing of the acquired data. A simple block diagram in figure 2 shows how these components are connected. Initially, a map of the orchard is created by flying the drone automatically above the orchard. Using the RGB-D camera on the drone, the placement of the trees are recorded and mapped. Using this map, the drone will be able to plan and fly between the rows of trees automatically, while recording colour and depth images. This will give a one-sided view of the trees to be used for the visual processing. Both point cloud processing and neural networks are applied to separate the trees from the background as well as determine the height of the trees, and the results are compared. The method proposed in this paper could also be expanded to measure other dendrometry parameters, and not only height.

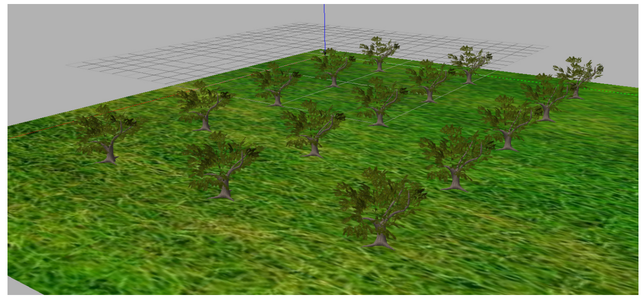
The position of the trees can be determined from the initial mapping of the orchard.

Due to practical reasons, it was not possible to implement the developed navigation onto the drone. Therefore, the navigation was only tested in the simulated environment. Moreover, the navigation and the vision of the developed solution have been tested separately but with real images captured at an orchard.

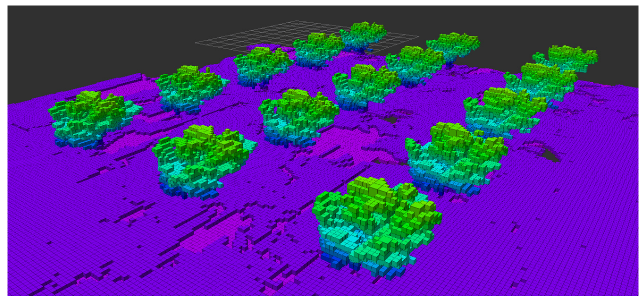
The solution aims to unify processing of the visual data with drone navigation into a complete solution that can be used to collect dendrometry data in the orchard. To do this, the processing blocks of the vision and navigation are combined using ROS framework.

4. Navigation

Simulated navigation takes place in Gazebo 7, ROS Kinetic, with the intention of positioning the drone for proper data collection of trees within an orchard. For this, a simulated environment of 15 trees was developed. Navigation is enabled through the use of the *hector_quadrotor* package, the 3D motion planning framework MoveIt, and the *octomap_mapping* package for developing a 3D occupancy grid map [11, 22, 8]. Two sets of waypoints are determined with separate approaches and for different applications; the first set is scouting waypoints for mapping the environment from above the trees, and the second set is for further data collection between the trees. Waypoints describe a desired position and orientation in the orchard and are passed to the drone. Figure 3 shows both the simulated environment and its representation as a 3D occupancy grid map.



(a) Simulated environment in Gazebo.



(b) Map created by OctoMap.

Figure 3: Visualisation of the simulated environment and the created map.

4.1. Creating scouting waypoints

An algorithm was developed for mapping a regular rectangular orchard, by flying over the trees and covering the orchard in the shortest distance possible. This is done by placing scouting waypoints the drone should follow. The distance between scouting waypoints, or step length (SL), is related to the area that the depth camera can cover in a single frame. The covered area depends on the field of view of the camera, the height of the tallest tree, and the desired height of the drone, which is fixed at 0.5 m above the approximate height of the tallest tree. The value of 0.5 m was chosen as it gives good image quality without sacrificing the safety distance. This calculation is shown in eq. 1. The algorithm for creating scouting waypoints requires the size of the field (MxN) and the calculated step length. With these two parameters, the algorithm can create a grid of scouting waypoints covering the entire orchard. After that, it finds the distance optimal path to visit all scouting waypoints, visualised in figure 4. The drone traverses along one row of points, then it moves to the next row and moves in the opposite direction. This process is repeated until the entire field is covered.

$$SL = 2\tan(\frac{FoV_h}{2})\frac{h_d - h_{tree_{max}}}{\sin(gimbal_{angle})} \tag{1}$$

FoV_h = horizontal field of view angle [deg]
h_d = desired height of the drone [m]
$h_{tree_{max}}$ = height of the tallest tree [m]
$gimbal_{angle}$ = camera angle respect horizontal axis [deg]

4.2. Mapping the environment

For creating a map of the environment, previously determined scouting waypoints are sent to the drone in a go-to-goal method using the */action/pose/goal* topic. The *hector_quadrotor* uses ROS Actions for navigating the drone to the desired location. This mapping is performed at an altitude approx 0.5 m above the tallest tree. This procedure will capture depth data from the attached simulated RGB-D camera, to create a map of voxels and a downsampled point cloud in the form of centres of occupied voxels. This map is created using the OctoMap package and is visualised in figure 3b [8].

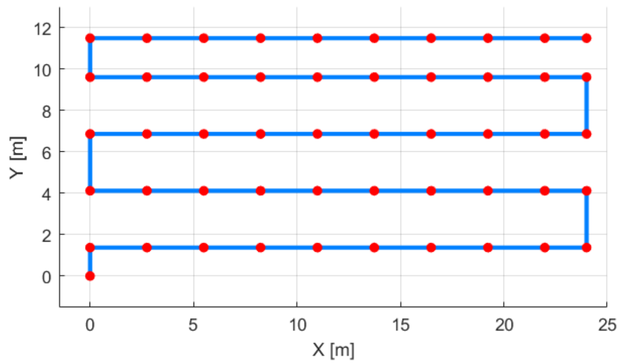


Figure 4: Scouting waypoints and the optimal path used for mapping the orchard.

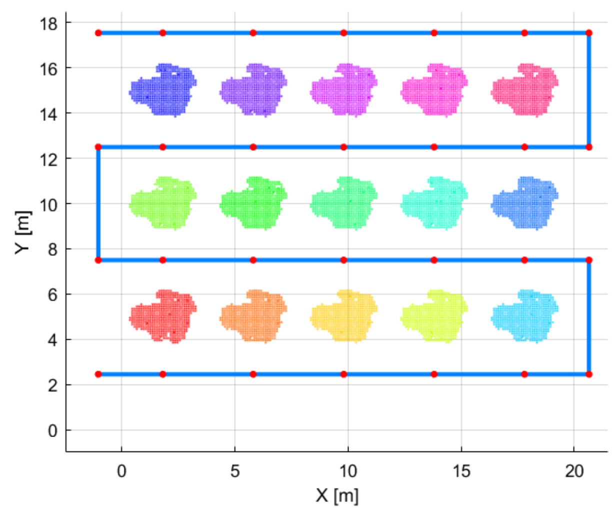


Figure 5: Data collection waypoints used for flying between the rows.

4.3. Estimating data collection waypoints and tree position

From the downsampled point cloud obtained during mapping, an algorithm is created to find each tree and determine its centre position. First, the ground points are removed using a threshold. The remaining points are then projected onto a 2D plane and clustered into trees using an algorithm based on the Euclidean distance between the points. This algorithm operates using a fixed distance threshold of 0.2 m. The centre of the tree is defined as the centre of the bounding box around each tree. The algorithm then clusters the trees into rows by assuming the distance between rows is larger than the distance between neighbouring trees. When the position of each tree and each row is determined, the algorithm creates data collection waypoints as shown in figure 5. This allows the drone to fly in between the rows and gather data of each tree individually.

4.4. Autonomous navigation between rows

As the waypoints for data collection are now determined, the drone will have to fly at a fixed altitude in between the trees. This creates the necessity of obstacles avoidance, to avoid collision and possible hardware disruption. Despite the drone maintaining a constant height when capturing this data, a 3D navigation method is implemented using MoveIt and the created map.

5. Data collection and labeling

For the development of both the neural network and the point cloud algorithm, prerecorded RGB-D data from a real orchard was used. The images were resized to 416x416 pixels and then labeled using an online labeling tool [10]. The labeling of the apple trees is made such that the bounding box encapsulates the entire tree including minor branches.

The dataset for training contains a total of 900 images, with 800 images containing instances of apple trees and 100 negative samples. Negative samples do not contain any part of an apple tree, but instead contains ground vegetation and background forest. The dataset collected for testing consists of 214 images of apple trees and is collected on a different day and of different trees than the training data. All the data which has been gathered originates from the same orchard, but are from three separate occasions, hence there is variation in lighting conditions and the appearance of the apple trees.

The depth data was collected with an Intel RealSense D435i. The resolution of the recorded depth data was set to 1280x720p. The depth data is transformed from a depth map to a point cloud using the RealSense SDK.

6. Point cloud processing (PCP)

The processing of the point cloud is done using the point cloud library [19]. This library provides tools for filtering, model fitting, segmentation, and extraction of point cloud data.

The processing starts with the filtering of Not-A-Number (NaN) points. Removing NaN points ensures the validity of the data for further processing. After the removal, the data is downsampled using a voxelized grid approach. Downsampling using the voxelized grid decreases the number of points in the point cloud while preserving the underlying geometrical information, thus decreasing the processing time without compromising the quality of the data. Following the downsampling, the number of points is further decreased by defining the region of interest (ROI). Because only objects facing the approximate centre of the camera

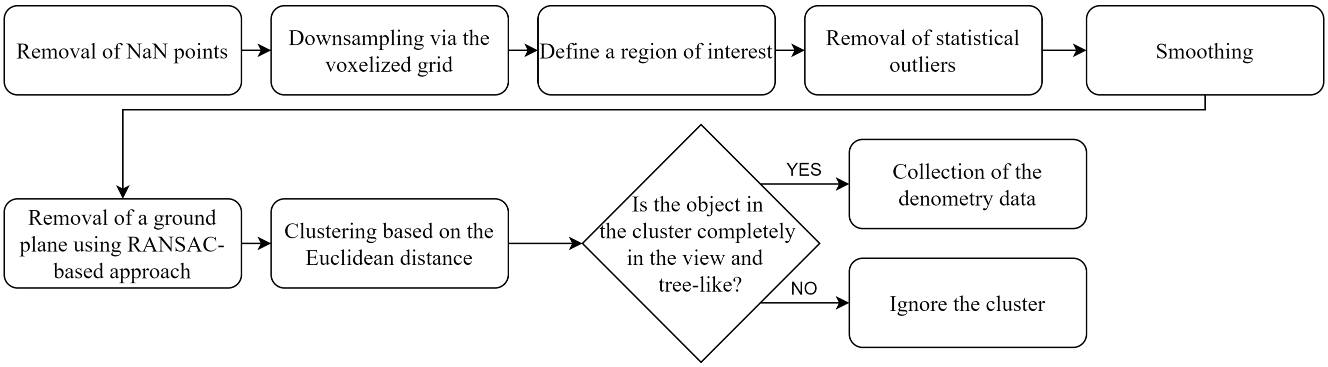


Figure 6: The flowchart of the proposed PCP algorithm.

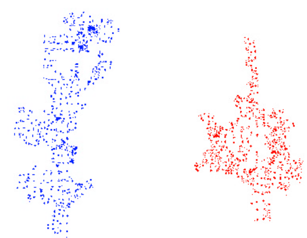
are suitable for the collection of dendrometry data, the ROI is placed around the center of the point cloud. All the points outside of the ROI are removed. Afterwards, the points within the ROI are filtered again, using the filter of statistical outliers.

The statistical outliers are calculated on a local scale. The mean distance from the given point p to all its neighbouring points is calculated. The distance is then assumed to be Gaussian distributed. The points that are more than 1 standard deviation away from the mean are removed. In the special case the point p itself is an outlier, the mean distance becomes large with the small variance. Under these conditions, the point p is not identified as an outlier. However, the filtering process is iterated over all the points in the point cloud. Therefore, the outlier in question is removed in the next iterations. Currently, 50 neighbouring points are used to determine the mean distance. This amount of points was chosen as it was observed during the development that it provides the best filtering.

Finally, the preprocessing of the point cloud ends with the resampling of the data via the Moving Least Square (MLS) method. Resampling is performed to improve the overall smoothness of the data.

In order to detect apple trees in the point cloud, the assumption is made that the point cloud only consists of points that belong to a ground plane or an apple tree. Therefore, removing points of the ground plane should leave only apple trees in the point cloud. Random sample consensus (RANSAC) is used for identifying the biggest plane in the data, which is then removed as it is assumed to be the ground plane. To collect the remaining points into clusters, such that each cluster represents one tree, an algorithm based on the Euclidean distance between points is deployed. This algorithm operates using a fixed distance threshold of 0.1 m. The threshold of 0.1 m was chosen because it can be observed that this value corresponds to the approximate largest distance between points that should be collected into one cluster. The result of the proposed point cloud algorithm is visualised in figure 7.

(a) Point cloud input



(b) Point cloud output

Figure 7: The figure shows the input-output pair of the proposed point cloud algorithm.

Before the dendrometry data is collected, each cluster is analysed to ensure that the given cluster indeed contains a tree. First, only clusters that have at least 250 points are processed further. Clusters with less than 250 points are considered too small to contain a model of an orchard tree. Next, the clusters are investigated if they contain a full or just a partial model of a tree. This is done by iterating over all the points in the cluster and checking the distance of each point from the boundaries of the ROI. If any of the points is closer than 5 cm to the ROI boundary, it is assumed that the object in the cluster continues behind the boundaries. Therefore, the object in the cluster is only partial and thus unsuitable for dendrometry collection.

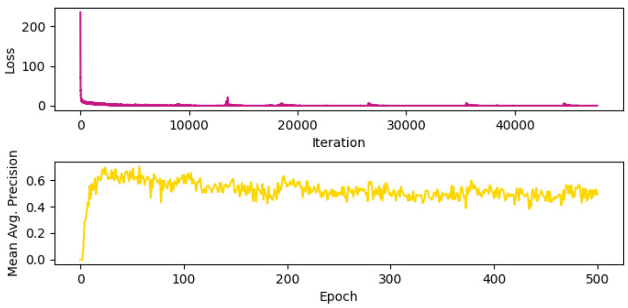


Figure 8: Graph of mean average precision and loss during 500 epochs of training YOLOv3.

The last two checks evaluate the geometrical properties of the cluster. The cluster has to be taller than 1.25 m, while also being taller than it is wide. These two properties were selected as it has become clear during the development of the algorithm that they are commonly present in the clusters containing trees. Finally, to collect the dendrometric data, the height of each cluster is calculated. The height is calculated as the difference between the maximum and minimum Y coordinate of the cluster. Moreover, the height is then lengthened by 30 cm to compensate for the RANSAC procedure and the possible loss created by multiple filtering processes. The flowchart of the proposed algorithm can be seen in figure 6.

7. RGB tree detection

The CNN approach for detecting apple trees in RGB images is based on the YOLOv3 object detection framework, which is based on the Darknet-53 CNN architecture [16]. YOLOv3 was chosen as it is shown to have very good detection rate and performs well on live recording. The proposed network is developed through transfer learning on the pretrained YOLOv3 weights from Joseph Redmon et al., which has been trained on the COCO dataset [15]. A smaller 23 layer YOLOv3 implementation known as YOLOv3-tiny is also proposed. It should provide faster processing time and can work well when working with few classes [17]. The YOLOv3-tiny is also developed with transfer learning from weights pretrained on the COCO dataset.

8. Experiments and results

8.1. Experiments

The mapping was tested by autonomously mapping the simulated orchard one time using the predetermined scouting waypoints and the navigational system. This is to ensure that the expected map in simulation covers the entire field.

The position of trees within the orchard is estimated based on the point cloud created during the mapping process. This estimated position will be compared to the actual tree position from the simulated environment in Gazebo, to evaluate the deviation from ground truth.

Precision and recall will be used to evaluate the proposed CNN and PCP methods. The YOLOv3 and YOLOv3-tiny framework is implemented using Erik Linder-Norén's GitHub repository [9] in a PyCharm environment, running on a desktop PC with a RTX 3070 GPU. The training is performed using the data as described in section 5 with a 80/20% distribution of training and validation data. For YOLOv3, the batch size is constrained by the RAM of the GPU and cannot be above 8, therefore 8 is chosen as the batch size. For YOLOv3-tiny the batch size of 16 is used for training. The learning rate is set to 0.0001 for both implementations. The training is initialised with the pretrained weights and is trained for 1000 epochs [15]. The loss and mean average precision (mAP) of each epoch is plotted and saved, such that the set of weights achieving the highest mAP can be extracted and used for detection. The loss and mean average precision during training can be seen in figure 8.

During the detection of apple trees a bounding box is created around each instance of the class. The placement of the bounding box can then be used to estimate the proportions of the tree by using the distance to the tree obtained from the drone position and estimated tree position. These metrics were selected so that the point cloud algorithm can be directly compared with the performance of the neural network.

Because no classification is done on the detected clusters in the PCP algorithm, some of the terms related to the precision and recall calculation must be slightly adapted. These terms are true positives, false negatives, and false positives. Clusters containing a single full tree model, that can be used for estimating dendrometry data, will be considered true positives. Full tree models, visible in the input data, that were not segmented into any cluster will be considered false negatives. Finally, any other type of clusters will be considered false positives.

The CNNs which will be tested is YOLOv3 and YOLOv3-tiny with 0.25, 0.50, and 0.75 intersection over union(IoU). The recall and precision of methods are calculated from testing on the testing dataset of 214 RGB-D images. The equations for precision and recall are shown in eq. 2 and eq. 3. The processing time during testing is recorded.

Method	Avg. Precision	Recall	Layers	Parameters	FPS
YOLOv3 IoU_{25}	78.04%	81.42%	53	$6.1 \cdot 10^7$	65.5
YOLOv3 IoU_{50}	63.53%	70.09%	53	$6.1 \cdot 10^7$	65.5
YOLOv3 IoU_{75}	22.53%	29.58%	53	$6.1 \cdot 10^7$	65.5
YOLOv3-tiny IoU_{25}	76.55%	80.39%	23	$8.6 \cdot 10^6$	104.2
YOLOv3-tiny IoU_{50}	52.86%	63.99%	23	$8.6 \cdot 10^6$	104.2
YOLOv3-tiny IoU_{75}	3.31%	14.15%	23	$8.6 \cdot 10^6$	104.2
Point Cloud Processing	76.72%	86.2%	-	-	2.1

Table 1: Table of tree detection results, comparing the CNN based method on RGB data with Point Cloud Processing. YOLOv3 and YOLOv3-tiny were tested with 0.25, 0.50, and 0.75 intersection over union (IoU).

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

TP = True positives
FP = False positives
FN = False negatives

The PCP and CNN test will also be compared by evaluating their ability to extract dendrometry data. 110 RGB-D images containing 25 trees are used for testing. All trees are present in more than one image. If a tree is detected more than once, the average of the predicted height is used to compare to the ground truth, which is measured manually.

8.2. Results

The mapping test shows that navigating the drone between the predetermined scouting waypoints in the simulation enables the OctoMap package to create a map which captures the entire field. This evaluation is based on visual inspection of the map which was created during this process. The map has some gaps in the ground plane, but the trees appear to be mapped in full detail, including both the canopy and the stem.

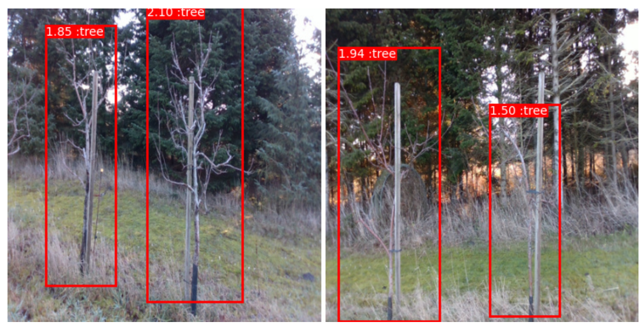
The results when comparing the estimated and actual values of the (X,Y) tree centre position in the simulated environment show a maximum deviation in estimated tree centre position in both X and Y coordinates of 0.05m.

Results of the tree detection test can be reviewed in table 1. Examples of good and bad detection results are shown in figure 9. The number written in the corner of the bounding box indicates the estimated tree height.

The compared results for tree height estimation can be seen in table 2. All 25 trees where detected at least once, with all three methods during the test.

Method	Mean error[cm]	σ Tree height [cm]
YOLOv3	20.6	15.16
YOLOv3-tiny	28.2	17.74
Point Cloud Processing	20.4	18.02

Table 2: Table showing the error of the height estimation and standard deviation of the calculated tree height compared to ground truth tree height.



(a) Good detections

(b) The three images show false positives, badly sized bounding box, and false negative, respectively

Figure 9: The figure shows examples of good and bad detection results. The number written in the corner of the bounding box indicates the estimated tree height.

9. Discussion

The obtained results of tree detection and dendrometry are influenced by the data gathered for the training and testing of both the RGB approach and the PCP approach. The RGB data has a significant amount of noise from both background trees and ground vegetation, which influences the results of the CNN.

The issue with the proposed PCP algorithm is the chosen clustering method. Using the Euclidean distance between points is an effective condition for grouping, but it fails once the tree branches become mixed or the ground vegetation grows too close to the tree's stem. Ground vegetation did not present problems for the CNNs. Moreover, the PCP algorithm was developed to be used on the trees with leaves. However, due to the change of season that occurred during the development, all the apple trees in the orchard had dropped their leaves by the time of testing, leaving the testing dataset of poor quality. If the testing was to be performed on the dataset of trees with leaves, the mean error in height estimation is expected to drop.

Looking at the raw data alone, the trees are much clearer when captured with 3D data. This is mainly due to the depth data being limited to 3 meters, where the RGB can have several trees in the background. A problem observed with the CNNs during testing was how the bounding boxes were placed compared to the actual trees. A bounding box was often too large due to including some parts from the forest in the background. This gives the PCP an advantage when segmenting, because the trees in the background are not a factor as they are for the RGB detection. However, the frame rate achieved by the neural network is considerably higher than that achieved by the point cloud algorithm. Furthermore, the same tree will appear in the data stream several times, so a lower computation time will allow for the algorithm to run detection more times and thereby increase the chance of detection.

The mapping and estimation of tree position is accurate when performed in simulation and on data gathered from here. This indicates that both the developed navigational system and the tree position algorithm are theoretically applicable when tested on simulated data. However, further testing, under real-life conditions, is required to evaluate said systems properly.

10. Conclusion

This paper examined two methods of detecting apple trees at an orchard, and determining their dendrometry; CNN on RGB images, and point cloud processing. Furthermore, the paper examined the possibility of recording RGB-D data automatically, using a drone to navigate around an orchard based on the calculated waypoints. The paper shows that it is possible to navigate a drone and collect data in a simulated orchard environment based on the navigational capabilities of the hector quadrotor, MoveIt, and the 3D occupancy grid map created using OctoMap. From the downsampled point cloud it is possible to accurately estimate the position of the trees. In addition, it is shown that the apple trees can be detected in RGB images with 63.53% precision and 70.09% recall using YOLOv3 IoU_{50} and 76.72% precision and 86.2% recall using PCP. Using YOLOv3, it was possible to calculate tree height with a 20.6 cm mean error, while PCP estimated tree height with a 20.4 cm mean error. Both methods show that they are viable options for automatic collection of dendrometry data from trees as they detected all 25 trees at least once during dendrometry testing. PCP shows better results compared to YOLOv3 for detection precision, but trades off processing time. Further research could be done on developing an approach that combines YOLOv3 and the proposed PCP method. If a high FPS is needed, YOLOv3 can be used for fast detection of trees, while PCP can perform more precise detection at lower FPS. Additionally, dendrometry estimation could be expanded to include other parameters like the diameter of the stem and crown or the tree's shape.

11. Acknowledgments

We would like to thank Charlotte Mølgaard for providing access to the apple orchard Egebakken.

References

[1] Michael C R Alavanja. Introduction: pesticides use and exposure extensive worldwide. *Reviews on environmental health*, 24(4):303–309, 2009.

[2] A Bienert, S Scheller, E Keane, F Mohan, and C Nugent. Tree detection and diameter estimations by analysis of forest terrestrial laserscanner point clouds. In *ISPRS workshop on laser scanning*, volume 36, pages 50–55, 2007.

[3] Vinit Bodhwani, DP Acharjya, and Umesh Bodhwani. Deep residual networks for plant identification. *Procedia Computer Science*, 152:186–194, 2019.

[4] Carlos Cabo, Celestino Ordóñez, Carlos A López-Sánchez, and Julia Armesto. Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning. *International journal of applied earth observation and geoinformation*, 69:164–174, 2018.

[5] Sumit Chakraborty. Intel realsense d415, d435 depth cameras launched. https://gadgets.ndtv.com/cameras/news/intel-realsense-d415-d435-camera-price-features-specifications-1802305, (visited on 17-11-2020).

[6] Mads Dyrmann, Henrik Karstoft, and Henrik Skov Midtiby. Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151:72–80, 2016.

[7] Markus Gandorfer and Andreas Meyer-Aurich. *Economic Potential of Site-Specific Fertiliser Application and Harvest Management*, pages 79–92. Springer International Publishing, Cham, 2017.

[8] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees, 2013. Software available at http://octomap.github.com.

[9] Erik Linder-Norén. Yolov3 repository. http://github.com/eriklindernoren/PyTorch-YOLOv3, (visited on 16-11-2020).

[10] MakeSense. Image labeling tool. https://www.makesense.ai, (visited on 16-11-2020).

[11] Johannes Meyer and Stefan Kohlbrecher. hector_quadrotor. http://wiki.ros.org/hector_quadrotor, (visited on 17-11-2020).

[12] UM Rao Mogili and BBVL Deepak. Review on application of drone systems in precision agriculture. *Procedia computer science*, 133:502–509, 2018.

[13] E Moltó, B MartıN, and A Gutiérrez. Pm—power and machinery: Pesticide loss reduction by automatic adaptation of spraying on globular trees. *Journal of agricultural engineering research*, 78(1):35–41, 2001.

[14] Benedict O'Neil. Terrestrial laser scanners (TLS): guide and product selection. https://www.aniwaa.com/buyers-guide/3d-scanners/terrestrial-laser-scanners-long-range, (visited on 17-11-2020).

[15] Joseph Redmon. Yolov3 weights (download). https://pjreddie.com/media/files/darknet53.conv.74, (visited on 16-11-2020).

[16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[17] Joseph Redmon and Ali Farhadi. Darknet github. https://github.com/AlexeyAB/darknet, (visited on 16-11-2020).

[18] J.R. Rosell and R. Sanz. A review of methods and applications of the geometric characterization of tree crops in agricultural activities. *Computers and Electronics in Agriculture*, 81:124–141, feb 2012.

[19] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[20] Seyed Iman Saedi and Hossein Khosravi. A deep neural network approach towards real-time on-branch fruit recognition for precision horticulture. *Expert Systems with Applications*, page 113594, 2020.

[21] David Schimmelpfennig. Farm profits and adoption of precision agriculture. (1477-2016-121190):46, 2016.

[22] Ioan A. Sucan and Sachin Chitta. Moveit. https://moveit.ros.org, (visited on 02-12-2020).

[23] Hayo MG van der Werf. Assessing the impact of pesticides on the environment. *Agriculture, Ecosystems & Environment*, 60(2-3):81–96, 1996.

[24] Naiqian Zhang, Maohua Wang, and Ning Wang. Precision agriculture—a worldwide overview. *Computers and Electronics in Agriculture*, 36(2):113 – 132, 2002.

[25] Qin Zhang, Manoj Karkee, and Amy Tabb. The use of agricultural robots in orchard management. *arXiv preprint arXiv:1907.13114*, 2019.