Row Following in Pergola Structured Orchards by a Monocular Camera Using a Fully Convolutional Neural Network

Jamie Bell, Bruce A. MacDonald, Ho Seok Ahn
Department of Electrical and Computer Engineering, CARES, University of Auckland
Auckland, New Zealand
jbel060@aucklanduni.ac.nz, {hs.ahn, b.macdonald}@auckland.ac.nz

Abstract

The research presented in this paper is a part of a project that is developing mobile robots, which perform pollination and harvesting in kiwifruit orchards. The navigation system for the mobile robot currently uses a 3D lidar to detect and traverse the rows and headlands of kiwifruit orchards. In order to produce a redundant channel for a more reliable navigation system, a row following method is being developed which uses a camera as the sensor. Kiwifruit orchards have a pergola structure, where a sparse array of posts and trunks holds up a canopy above. This pergola structure presents challenges for row following, especially because the array of posts and trunks is so sparse that it is possible to confuse another direction for the true row direction. This paper tests a novel method for performing row following in pergola structured orchards, using a single camera. A fully convolutional neural network is used to perform semantic segmentation of color images for an abstract class, called "traversable space". Key points from a centreline fitted to the segmented region are used to control the steering angle. Test results indicate that this algorithm may follow rows better than the existing 3D lidar navigation system.

1 Introduction

The research presented here is a part of a wider project that is developing mobile robots to perform harvesting and pollination in kiwifruit orchards. The mobile robots consist of a base platform, which drives autonomously throughout kiwifruit orchards, while carrying either a pollination or harvesting module (Figure 1). The current version of the mobile robot is approximately 3 metres long and 2 metres wide, depending on what modules or items are being carried. A 3D lidar based navigation system has been developed for the mobile robot platforms [Bell *et al.*, 2016a]. However, an analysis of standards for analogous applications has produced a requirement that this navigation system should use multiple independent channels, using different sensors, for redundancy and to perform cross-checking [Bell *et al.*, 2016b]. Hence, it was decided that another channel of the navigation system should be developed. The research presented in this paper describes development and testing of a channel of the navigation system, which uses a camera as its sensor. This channel provides redundancy for the row following behavior of the mobile robots in the kiwifruit orchards.

Figure 1: The mobile orchard robot with the harvesting module.

Figure 2: A pergola structured kiwifruit orchard, with trunks and posts holding up a canopy, which lets patchy sunlight through.

Figure 3: The view of an orchard looking at approximately 45 degrees (blue) to the row direction and orthogonally (red).

The kiwifruit orchards have a pergola structure, with a canopy, consisting of the vines, leaves, flowers, fruit, wires and supporting beams above, which is all supported by trunks and posts as shown in Figure 2. This environment presents some unique challenges compared to other growing systems, because of the sparse nature of the structure-defining features. Figure 3 shows a view in the pergola when looking orthogonally and at an angle of approximately 45 degrees to the true direction of the row. In these directions, feasible paths may appear and may resemble a true row; however, the robot must not drive in these directions because there is less space for the robot and the robot would cross and could damage the irrigation lines. Ensuring that the robot does not accidentally set a course outside of the current row is a key challenge for the navigation system. Another challenge present in the pergola structure is the variation in lighting conditions, which can be observed in Figure 2 and Figure 3. Even when the canopy is in its most fully developed state in early autumn, sunlight comes through the canopy to give high contrast blotted lighting.

1.1 Problem Definition

The overall problem addressed in this paper is to perform row following in pergola structured orchards, using a single camera. The row following problem in pergola structured orchards has previously been described [Bell *et al.*, 2016a]. For some defined mobile robot coordinate system, the linear offset is the distance between the coordinate system origin and the row centreline, which is the line fitted between the tree-lines on both sides of the row (Figure 4). The angular offset is the angle between the robot coordinate system x-axis and the row centreline. The aim of row following is to minimize the linear and angular offsets, simultaneously. The aim for the methods presented here is to perform this task at least as well as the existing 3D lidar navigation system so that these new methods can be used for redundancy and cross-checking.

An interim step that is used in this paper is to find the row centreline in the image. The row centreline in the image is here defined as the line connecting:

- The row centre at the furthest visible row point.
- The row centre closest to the bottom of the image. However, parts of the image, where both sides of the row were not visible, were disregarded since the row centre could not be found in these parts of the image.

These two points were used in the development of the row following algorithms. An example of a hand labeled row centreline is illustrated in Figure 5.

1.2 Orchard Row Detection in Images Review

He et al. [2011] performed row detection for apple orchards using monocular vision. The preprocessing steps used were extracting the red and blue channels, the image was thresholded, morphological operations were performed, blobs were thresholded based on size and a region of interest was set based on the sum of pixels in each image row. The features that were extracted for row detection were the transition between the bottom of the trunks and the ground, which were found by searching for the lowest corner feature in a vertical column region. The trunk base features were split into containers for the

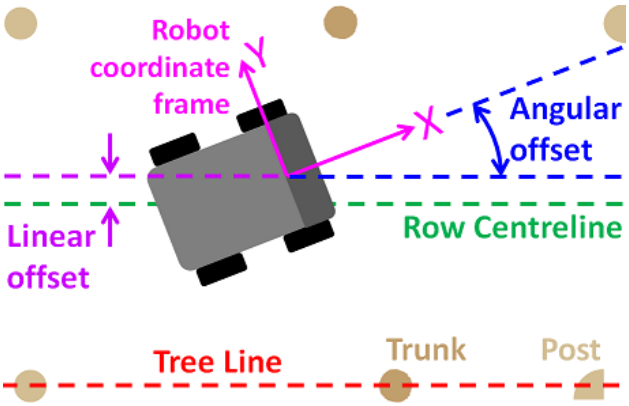

Figure 4: Illustrating the linear and angular offset of the robot coordinate system from the row centreline in a pergola structured orchard.

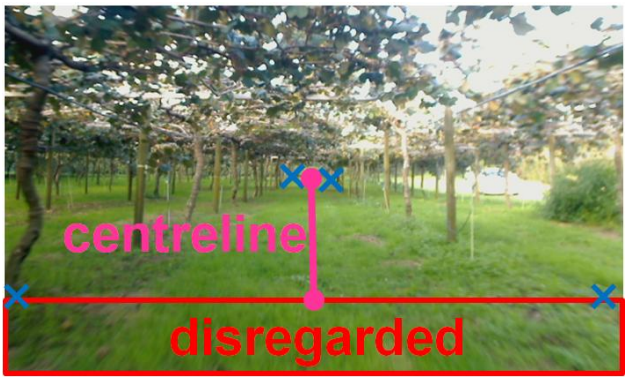

Figure 5: A hand labeled row centreline, where each end point is the centre between points clicked at the edge of the row (blue). Areas in the image, where either side of the row were not visible, were disregarded.

left and right of the image. Lines were fitted to the trunk base features using sum of least squares regression and the row centreline was found as the average of the two tree-lines.

Sharifi and Chen [2015] also performed row detection on colour images of orchard rows. They used mean shift clustering and classified the resulting clusters as sky, ground or trees according to their colour. The ground pixels were extracted and the Hough transform was used to fit two lines for both sides of the row. The row centreline was calculated from the two tree lines.

The methods of He et al. [2011] and Sharifi and Chen [2015] are colour based and hence are vulnerable to different lighting conditions, colour changes when wet, different weeds growing and changes of colour of leaves throughout the year. In addition, these row detection methods are not developed into row following methods and hence are not tested to control a robot, unlike the methods which are presented here.

Torres-Sospedra and Nebot [2011] preprocessed images by performing 2 level Daubechies wavelet transforms. Then the mean and energy of windows in the resulting images were calculated as features. A sample of these features and the coordinates of the window were used as input to an ensemble of fully connected neural networks with 3 layers. The neural networks classified the windows in the image as ground, tree or sky. The resulting output was processed with morphological

operations to reduce noise and edge detection to extract the tree-lines. At different horizontal slices Hough transforms and row centreline calculation were performed.

Zhang et al. [2012] employed a tree-line fitting method in combination with a vanishing point detector. The tree-line fitting method used a Sobel filter and thresholding to extract vertical edges. A subset of these edges was randomly selected and tracked using the Lucas Kanade method. RANSAC was used to find inliers, which were reconstructed into 3D points and the tree-lines were fitted to these points. The vanishing point detection preprocessed the images with a Sobel filter and thresholding to extract pixels on vertical edges, a colour space conversion from RGB to $c_1c_2c_3$ to reduce the effects of lighting variations and a colour filter to extract trunks and branches. The number of resulting points in each column was counted and the centre of a region of minima was taken as the vanishing point column. An Extended Kalman Filter was used to fuse the outputs of the tree-line fitting and vanishing point methods.

It is unclear how the existing row detection methods developed for apple and citrus orchards would work in a pergola structured orchard. It seems that the method of Torres-Sospedra and Nebot [2011] would correctly classify the ground in adjacent rows as ground but then would not produce lines for line fitting to the tree-lines. The methods of He et al. [2011] and Sharifi and Chen [2015] might have similar issues. The method of Zhang et al. [2012] might find tree-lines outside of the current row or a vanishing point at an angle to the true row since the vertical edges of posts and trunks in adjacent rows would be extracted by the Sobel filters.

Scarfe [2012] developed a method to detect the posts in a pergola structured orchard. Edge detection was performed by template matching and then straight edges were found in the resulting features using Hough transforms. However, it was found that this method was susceptible to changes in lighting conditions. Further experiments were performed using Sobel filters for the edge detection; however, ultimately it was concluded that the difficult lighting conditions in pergola structured orchards made the use of hand-engineered features very challenging for row detection.

It was decided in the research presented here to not use hand engineered features but rather to leverage state of the art results that have been achieved by Deep Learning [LeCun *et al.*, 2015]. In particular, this paper uses semantic segmentation methods. Recent advances in semantic segmentation using Convolutional Neural Networks (CNN) have allowed more accurate per pixel labeling of many generic classes, including grass and trees [Farabet, 2013].

1.3 Paper Outline

This paper contributes a method for row centreline detection in pergola structured orchards by CNN semantic segmentation. In addition, two driving strategies, to leverage the output of the row centreline detection and control a robot, are contributed. All of these methods are presented in Section 2. An additional contribution is the test methods, which are presented in Section 3, as well as the testing and comparison results of the row following methods, which are presented in Section 4. Discussion of the testing is given in Section 5 and conclusions are made in Section 6.

2 Row Following Methods

2.1 Data Collection

In order to have a dataset of images to develop and test algorithms with, over 10,000 images were collected, while traversing kiwifruit orchard rows, from cameras mounted horizontally at a height of approximately 0.8 of a metre from the ground. The efforts made to create a full range of variability in the data, included collecting images:

- From five different orchards.
- At different times of the day and night with different artificial lighting.
- In different weather conditions and seasons.
- Using five different models of cameras, including Logitech webcams, phone cameras, a camcorder, CCD cameras from Point Grey and CMOS cameras from Basler.
- From a range of locations in orchards and facing different directions.

2.2 Row Segmentation Classes

Initially it was unclear what classes should be used for segmentation of the row. The classes that had been used by previous research included sky, trees, posts and ground. Sky did not seem useful for a pergola structured orchard, since the sky is not visible when traversing a row under the canopy. The classes initially used were:

- Posts big enough to label individually.
- Trunks big enough to label individually.
- The tree-line of the current row, consisting of posts and trunks too small to individually label.
- Ground where the robot can currently drive directly to without collisions.
- The end of the row.

An initial training set of 50 images was hand labeled and used to train a pre-trained FCN-8s fully convolutional neural network [Long *et al.*, 2015], using the Caffe framework [Jia *et al.*, 2014] within the Nvidia Digits toolkit, following steps outlined for existing datasets [Heinrich, 2016] without using data augmentation.

An example of a training image and the corresponding label image are shown in Figure 6. An example inference result is shown in Figure 7. It seemed from these results that the post and tree classes were not as accurate as the tree-line and ground classes. Since the tree-line is made up of posts and trunks, it was hypothesized that an overall more accurate and useable result might be achieved by training the CNN to segment all posts and trunks as the tree-line. Hence, another dataset was hand-labeled without the post and trunk classes and the training was repeated. An example of the inference results for the resulting model is shown in Figure 8. The tree-line exhibits some inaccuracies in the foreground at the extremities. Of the three classes segmented in Figure 8, the ground class seemed to be the most useful class for row following, since it contains information about the row end and the tree-line and is also generic enough to be used outside of the row. An important point to note about the ground class is that it

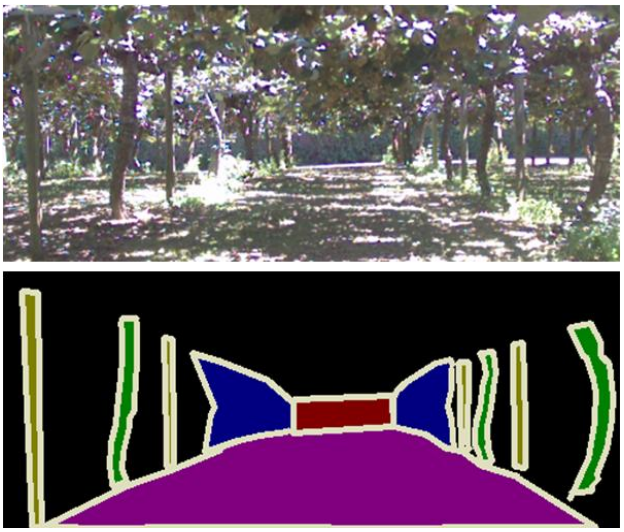

Figure 6: An example of an input image (top) and label image (bottom) pair, using five classes for the posts (tan), tree trunks (green), tree-line further down the row (blue), row end (red) and the ground where the robot can currently drive directly to without collision (purple).

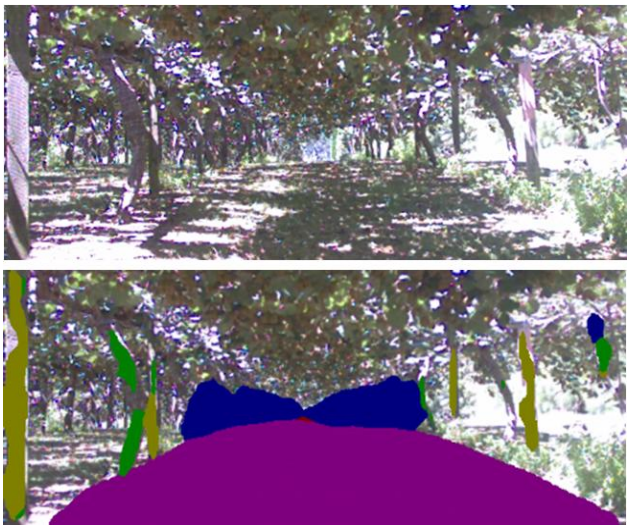

Figure 7: An example input (top) and output (bottom), using an FCN-8s, trained using the five classes, as per Figure 6. There are some errors amongst the posts, trees and tree-line classes, especially in the foreground. However, the row end and current row ground classes appear to be segmented more accurately.

Figure 8: An example inference result from an FCN-8s CNN, trained with three classes for the row end, tree-lines and segments of the ground that can currently be reached without collision. These results appear to be more accurate than those with the five classes from Figure 7.

Figure 9: An example of a test semantic segmentation result of the traversable space.

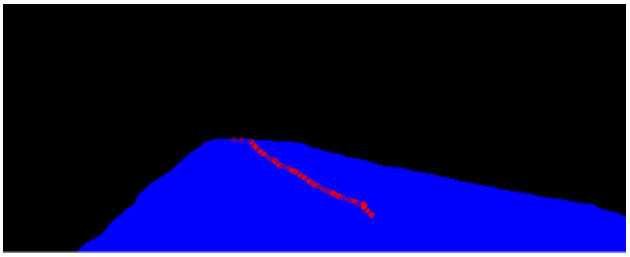

Figure 10: Row centres (red), found in an interim step by the row detection algorithm, in the traversable space (blue).

Table 1: Hyper-parameters used to train the traversable space FCN-8s.

Hyper-Parameter Description	**Value**
Number of Validation Images	5
Number of Training Epochs	50
Batch Size	1
Solver Type	Adam
Initial Learning Rate	0.0001
Learning Rate Multiplier	0.1
Epochs Between Learning Rate Reduction	17

is not just any ground but only the ground that can be directly reached by translation of the robot from the current pose without collision; this class is hereafter referred to as the "traversable space". The results from Figure 7 and Figure 8 suggest that the traversable space can be segmented without the ground from adjacent rows being classified as false positives. It was decided to try to use just the traversable space in order to detect the rows.

2.3 Traversable Space Segmentation

From all of the images collected, 600 images were selected for hand labeling. The images were resized and cropped to bring them to the same size. The image size used for training was 400 pixels wide and 160 pixels high. This resolution was used because testing indicated that an image of this size could be processed at 5 Hz, using a laptop with a Nvidia M5000M GPU, and in trials this rate was found to be sufficient for satisfactory row following performance. The images were labeled by marking the pixels corresponding to the "traversable space", which was taken to mean the space that the mobile robot might be able to reach without collision if translated in a straight line from the current pose; the dataset included images inside and outside of the row but all images were labeled according to this definition of traversable space.

The labeled data was used to train a FCN-8s CNN [Long *et al.*, 2015]. Some of the key hyper-parameters used for training are given in Table 1. An example inference result is shown in Figure 9.

2.4 Traversable Space Line Fitting

The inference outputs were used to find the row centreline, as defined in the problem definition in Section 1.1. The steps used to find this row centreline were:

1. For every row of the inference output, an iterator was incremented along the image row pixel by pixel to find the pixels where the segmented region started and ended. If more than one segmented region was found on a row, the start and end points pair with the greatest difference was considered for the next step.
2. If a start point or the end point from step 1 was at the border of the image, both points were disregarded. Also, all subsequent points to the bottom of the image were disregarded. This step allowed for the case where the full width of the orchard row was not visible in the image and so the centre-point in that row of the image could not be found.
3. Using the start and end points of the segmented region for each image row, the centre of the segmented region for each image row was calculated as the average of the start and end points. An example output from this step is shown in Figure 10.
4. If there were more than a threshold number of centre points from step 3, a line was fitted to the centre points.

2.5 Row Following by Tracking Keypoints

Following the row centreline detection in the image, the next step of the row following algorithm was to generate steering commands, which were angular velocities. To use the calculated centreline to control the robot for row following, the basic premise was that the camera was in the centre of the robot and facing directly forwards and so the goal for driving should be to have the centreline intercept the bottom of the image near the horizontal image centre and this centreline should be close to vertical. To develop the algorithm for row following, a sequence of images from a camera on a robot was studied, while a person was driving the robot by remote control and performing a row correction manoeuvre from large linear and angular offsets. The behaviour of the human driver was to:

1. Prioritise correcting the linear offset by driving towards the middle of the row.
2. Correct the angular offset after the linear offset was mostly corrected.

In a similar way, an algorithm was developed to track the row centreline at the bottom of the image, in order to reduce the linear offset. Once the row centreline at the bottom of the image was close to the horizontal centre of the image, the algorithm tracked the row centreline point at the end of the orchard row. The steps in this control algorithm for driving the robot were:

1. The driving error in the foreground (e_f) was calculated according to:

$$e_f = i_w/2 - x_f, \quad (1)$$

where i_w is the width of the image in pixels
x_f is the x-coordinate of the orchard row centreline in the foreground of the image.

2. The steering command (r_z) was calculated by a proportional controller according to:

$$r_z = k_f e_f, \quad (2)$$

where k_f is a constant.

3. However, if the calculated error e_f from Step 1 had been less than a threshold for a set number of program cycles, the steering command was calculated according to:

$$r_z = k_b (i_w/2 - x_b), \quad (3)$$

where k_b is a constant
x_b is the x-coordinate of the orchard row centreline in the background of the image.

An alternative method for row following just used steps 1 and 2. This approach only tracks the row centreline at the bottom or foreground of the image.

3 Testing

3.1 Test Equipment

The robot used for testing was a Clearpath Husky robot. For safety reasons, this robot is used by the project to test navigation algorithms before deployment on the full sized orchard mobile robot shown in Figure 1. One camera was mounted centrally at a height of 0.8 of a metre from the ground. The camera used automatic white-balance, gain and exposure settings. The camera was connected to a laptop with a Nvidia M5000M GPU (Figure 11). The GPU was used to run the CNN.

Figure 11: The hardware test setup with a camera centrally mounted on a Husky robot. The 3D lidar was used for ground truth and comparisons.

A Velodyne VLP-16 lidar was mounted directly above the camera with care taken to physically vertically align the origins of the camera and the lidar. Two other cameras were present on the set-up but these were not used and were only present because the same hardware is being used for other research.

3.2 Row Following Comparison

Row following tests were performed to measure the performance of the semantic segmentation row following methods. Two variants of the row following method were tested:

- Tracking just the row centreline in the foreground of the image.
- Performing two stages of tracking- starting with tracking the row centreline in the foreground of the image and then transitioning to tracking the row centreline at the end of the row.

In addition, these variants of the semantic segmentation row following methods were compared to the previously developed 3D lidar row following method [Bell *et al.*, 2016a]. The premise of the row following tests was to perform repeat runs over the same stretch of kiwifruit orchard row with all row following methods. The procedure used for performing the tests was:

1. The robot was positioned in a kiwifruit row with a deliberate linear and angular offset.
2. The pose of the robot was marked using four pegs, driven into the ground, next to all of the wheels.
3. The row following algorithm was started along with data collection from a Velodyne VLP-16 3D lidar, which was mounted above the camera on the robot.
4. The robot was allowed to traverse 50 m of the row at a speed of 1 ms^{-1}.
5. The robot was repositioned at the starting position, using the pegs to ensure that the robot was in the same starting pose as previously used.
6. Steps 3-5 were repeated for the different row following methods.
7. Steps 1-6 were repeated six times for different poses in the orchard.

As defined in the problem definition (Section 1.1), the metric used to assess the row following behavior was the linear and angular offsets of the centre of the robot from the row centreline. For the purposes of this testing, the centre of the robot was defined as the centre of the lidar sensor.

Ground truth linear and angular offsets were found by hand labeling the lidar data for each row following run of the robot. The choice of using hand-labeling for this ground truth was dictated by the cost of using some robotic surveying equipment or inertial systems and the loss of signal under the kiwifruit canopy for expensive GPS systems. The linear and angular offsets from the row centreline for each program cycle of the test runs were found by:

1. Reading data from the 3D lidar for a given run of the robot during testing.
2. Viewing the data from a birds-eye view in a purpose built application.
3. Clicking on points in the data that were the posts and trunks on one side of the current row and then doing the same for the other side of the row.
4. The coordinates of the points clicked on were used to fit lines to the tree-lines on both sides of the row and the row centreline was found as halfway between the two tree-lines. The angle of this centreline was the angular offset.
5. The linear offset was calculated as the distance from the origin (of the lidar) to the row centreline.

4 Results

The angular offsets used during testing were large enough that free paths through to the adjacent rows were visible in the camera images. Nevertheless, the robot always corrected this angular offset and followed the true direction of the rows for every test run and for multiple rows of orchard driving performed since the testing runs. This result was somewhat surprising. It seemed that the CNN was able to conditionally classify the ground correctly; the ground in adjacent rows was traversable but only if the image was taken from that row and the CNN correctly learned this condition. An example result of classifying the current row somewhat correctly without false positives from clearly visible ground in adjacent rows is shown by the red region in Figure 12.

Figure 12: Segmentation with little of the current row visible.

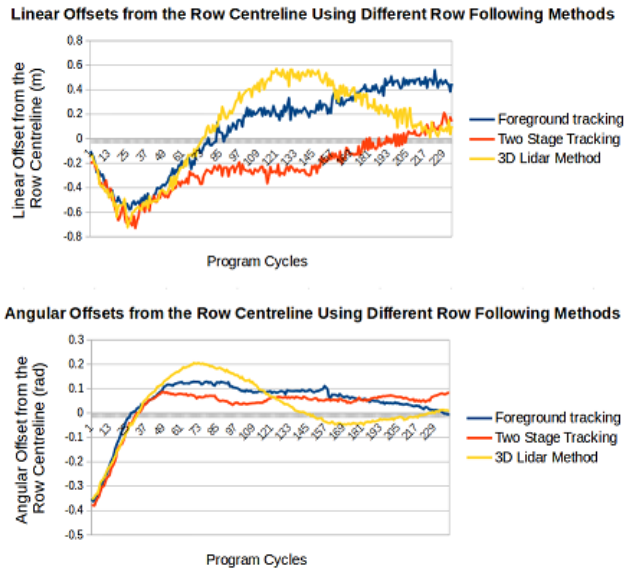

Figure 13: The linear and angular offsets measured by the hand labelled lidar data for the three row following methods and one of the starting poses.

Table 2: Mean linear and angular offsets from test runs for the three row following methods tested.

	Mean Offset Magnitude	
Algorithm Description	**Linear (m)**	**Angular (rad)**
Foreground Centreline Tracking	0.3	0.07
Two Stage Centreline Tracking	0.2	0.06
3D Lidar Row Following	0.3	0.07

An example of the linear and angular offsets, while performing row following, from one of the robot starting poses for all three methods tested is shown in Figure 13. The summary statistics from all of the test runs are presented in Table 2. It seems that the semantic segmentation row following methods presented here provide at least similar performance to the existing 3D lidar row following method. In addition, it seems that the two stage tracking method using semantic segmentation may provide some improvements in performance over the existing 3D lidar row following method.

5 Discussion

The method used to determine the ground truth pose has some weaknesses, including the assumption that a person labels the data accurately and the time consuming nature of labeling the data by hand means that larger test runs becomes prohibitively expensive. Since the lidar is the ground truth sensor and the navigation sensor for the 3D lidar row following method, the lidar row following may have some advantage in terms of the results since the vertical axis of the camera may have some offset from the vertical axis of the lidar. An effort was made to manually reduce this offset by carefully mounting the sensors; nevertheless, some residual offset would have been present. In any case, this offset favours the incumbent lidar method and hence the positive performance of the camera based methods presented here is still valid.

The tests performed here were conducted during the day. However, testing of the row detection method has been performed using night time images, collected with artificial lighting. Initial results indicate that the methods presented here could also be used at night.

6 Conclusions

A FCN-8s Convolutional Neural Network was used to segment the features of the current row in a pergola structured orchard; in particular, the segmentation worked best for the tree-line, end of row and the ground in the current row. The ground in the current row was segmented and differentiated from the ground in adjacent rows, even though the ground in adjacent rows was visible and had a similar appearance. This segmented region was used to find the row centerline in an image. Using the assumption that the camera was centrally mounted and facing forwards on the robot, the approach used to control the robot was to drive so that the row centerline appeared central and vertical in the image. This was achieved by tracking just the orchard row centerline in the foreground and also by using a two stage tracking method, which had an additional step of tracking the orchard row centreline in the background also.

The key contributions of this paper are the row detection and following methods for pergola structured orchards, the comparison of algorithms and the method for determining ground truth. From the results presented here, it seemed that the two stage tracking method using semantic segmentation may provide some improvements in performance over an existing 3D lidar row following method. Even equal performance would have been sufficient for the goal of the vision navigation system providing redundancy and cross-checking; hence it was decided to use the two stage tracking method of the row centreline- which was calculated from semantic segmentation output- as an independent channel in the navigation control system for redundancy and cross-checking, along with the existing 3D lidar row following method.

Acknowledgements

We would like to thanks our friends from the University of Waikato and Robotics Plus Limited for their excellent contributions to the orchard robotics project and, in particular, for their help in facilitating the testing of the research, presented in this paper.

References

[Bell *et al.*, 2016a] Jamie Bell, Bruce A. MacDonald, and Ho Seok Ahn. Row following in pergola structured orchards. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 9-14, 2016.

[Bell *et al.*, 2016b] Jamie Bell, Bruce A. MacDonald, Ho Seok Ahn, and Alistair J. Scarfe. An analysis of automated guided vehicle standards to inform the development of mobile orchard robots. *IFAC-PapersOnLine*, 49(16):475--480, 2016.

[Farabet *et al.*, 2013] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915--1929, 2013.

[He *et al.*, 2011] Bei He, Gang Liu, Ying Ji, Yongsheng Si, and Rui Gao. Auto recognition of navigation path for harvest robot based on machine vision. In *Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Selected Papers, Part I*, pages 138--148, 2011.

[Heinrich, 2016] Greg Heinrich. *Image Segmentation Using DIGITS 5*. 2016. [Online]. Available: https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5/ [Accessed: 04-Aug-2017].

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *CoRR*, abs/1408.5093, 2014.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521:436--444, May 2015.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431--3440, 2015.

[Scarfe, 2012] Alistair J. Scarfe. *Development of an Autonomous Kiwifruit Harvester: A thesis presented in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Automation at Massey University, Manawatu, New Zealand*, 2012.

[Sharifi and Chen, 2015] Mostafa Sharifi and XiaoQi Chen. A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 251--255, 2015.

[Torres-Sospedra and Nebot, 2011] Joaquin Torres-Sospedra and Patricio Nebot. A new approach to visual-based sensory system for navigation into orange groves. *Sensors*, 11(4):4086--4103, 2011.

[Zhang et al., 2012] Ji Zhang, George Kantor, Marcel Bergerman, and Sanjiv Singh. Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3659--3666, 2012.