Robotic Programming Assessment 2 CMP9767M

James Tombling (25913882)

January 2022

1 Aims and Objects

The aim of this project was to build a code artefact for navigating a vineyard and identifying and mapping grape bunches using the Thorvald robot, inside of a Gazebo and Rviz simulation using ROS(meoldic) in Ubuntu 18.04 as accurately and quickly as possible. The base code for the simulation was provided from: https://github.com/LCAS/CMP9767M.git .

The rapid expansion of agricultural crop robotics in the recent decade is due to two factors: (a) the convergence of advancing mechatronics technology, which makes such automation technically feasible, and (b) the desire for crop production alternatives to human labour.(Lowenberg-DeBoer et al., 2020, 278) Agricultural employees are tough to find and keep all over the world. Agriculture is being pushed to robotics to use inputs more precisely as environmental and food safety concerns grow (Finger et al. 2019). This is the motivation behind the project and why its is important to explore methods and technologies to estimate the potential yield of crops using state-of-the-art techniques.

This can be achieved by completing these objectives:

- Navigating the environment while avoiding obstacles and reaching certain map coordinates. (navigation).
- Identifying the crop bunches. (perception).
- Calculating and transforming the crop bunch's world coordinates from local relative coordinates. (coordination).
- Mapping the environment in use.
- Plotting crop bunch point clouds for virtual analysis.

2 Related Work

There are many different techniques that could be deployed for the detection of crops(grapes) such as, semantic segmentation, convolutional neural networks (CNN's) , YOLOv3, RGB or HSV colour space thresholding, morphological operations and multi-stage detection algorithms. in most cases it is unlikey that only one of these is used in a system because of their different attributes and faults, it is common to find them combined to improve accuracy and robustness.

A colour image can be expressed in either the RGB (Red, Green, Blue) or HSV (Hue, Saturation, Value) space. An example of a colour image represented in these two spaces is evaluated with many comparisons, including more accurate thresholding for isolating the grapes in the HSV colour space as well as better separation during varying lighting levels. We compute the average and standard deviation values of a pixel's colour in these two spaces for each block of the original image. In terms of colour information, we have a total of 12 values that potentially could be processed. The 2006 paper "Grape Detection By Image Processing" highlights a efficient combination of some of these techniques by using HSV images of grapes and Zernike moments which belong to a class that can be computed straight on an image.(Chamelat et al., 2006, 3697). Zernike moments have been used multiple times in coalition with Canny edge detection algorithms. However modern canny edge detection can be just as or more effective than both techniques combined in certain scenarios. To compute the intensity of the gradients, it employs a filter based on the derivative of a Gaussian. The Gaussian filter decreases the impact of image noise. Then, by deleting non-maximum pixels of the gradient magnitude, possible edges are reduced down to 1-pixel curves. Finally, applying hysteresis thresholding on the gradient magnitude, edge pixels are maintained or eliminated.(Xuan and Hong, 2017, 275).

CNN's are another common approach for crop detection and have been used for a while and the reason for that is that many factors influence crop yield, including crop genotype, environment, and management approaches. Seed firms have greatly improved crop genetics throughout the years. But year-to-year and location-to-location changes in crop yield are greatly influenced by changing environments, both geographically and temporally (Horie et al., 1992). Accurate yield prediction is extremely beneficial to global food production in such conditions. On the basis of precise detection, timely import and export decisions can be made. The production estimate can help farmers make informed management and financial decisions. In fresh and untested areas, the performance of CNN's far outweighed image processing techniques described before. However, due to numerous complicated factors, accurately predicting crop yields is extremely difficult. Genotype and environmental factors, for example, frequently interact with one another and can be detected and analysed using CNN's either statistically or imagery based(Khaki et al., 2020, 1750). The downside for CNN's are the massive amounts of data required to train and build them accurately ,as well as the high amount of GPU power required to process this data. a CNN can also be quite slow if it has to carry out operation such as maxpool. Thus, effecting its real-time relay and speed of crop analysis as specified in the 2018 paper "A Review of the use of CNN's in agriculture". (Kamilaris and Prenafeta-Boldú, 2018, 312).

Once location of the grape bunches has been gathered it needs to be turned into global coordinates and displayed. A common way of doing this is using TF transform and then to use point clouds. Previous work on crop plotting point clouds includes the 2015 paper "Open source robotic 3D mapping framework with ROS". Which demonstrates and compares software techniques for gathering and processing point clouds in the ROS environment. The paper explains that LiDAR data can be stored as a XML or point clouds dataset (PCD) metamodels and normally requires the use of an Extended Kalman Filter to get accurate odometry data. (Bedkowski et al., 2015, 644). However, in that paper it states that these files need to be filled from ROS's .BAG files which can record and playback information from certain topics in ROS instead of direct LiDAR data from topics to the XML/PCD file. This .bag files acts a intermediate that is supposed to save computational power. However the 2017 paper "Rosbridge: Ros for non-ros users" demonstrates a faster alternative using a Rosbridge or CVbridge allowing for raw topic data to be transformed into relative coordinate data in coalition with the 'OpenCV library. (Crick et al., 2017, 493). Allowing for quicker data transformation for a more effective point cloud appending using the PointCloud2 from the Point Cloud Library (PCL). (Hsieh, 2012, 729).

Navigation is an important part of this projects ability to function as the two previous processes would not be feasible without the robot's world coordinates. two common ways of localising a robots world coordinates is to use either an Extended Kalman Filter (EKF) or a Adaptive Monte Carlo Localisation (AMCL) algorithms both using 360 degree LiDAR information for SLAM-based pose estimation in a topological navigation system. The 2021 paper "A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles" compares EKF and AMCL and states that AMCL's error is 0.656 percent compared to EKF's at 0.681 percent, however this is for a indoor environment not a outdoor vineyard but because this simulated vineyard has one fixed lighting and no dynamic effects like weather ,wind or terrain. The simulation is comparable to indoor environments. (Zou et al., 2021). Thus AMCL will be the most appropriate technique to use. Data inputs such as RGB camera, infrared, lidar and depth camera can all be used in deep learning techniques such as CNN's have been around for while and are still widely used.

3 Concept

This project took the analysed related work and built the code artefact based around the flow diagram below.

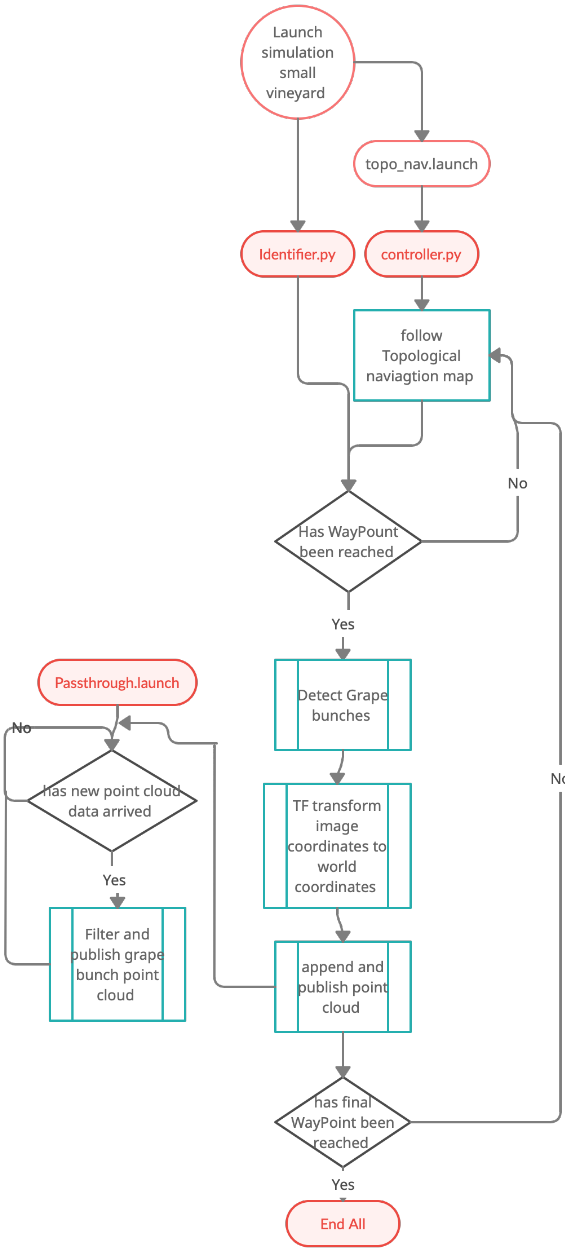

Figure 1 Flowchart Design

The environment chosen was the world simulation of the small vineyard due to the other simulations either lacking grapes or having vineyards to close together for the Thorval Robot to be able to succesfully navigate.

The "topo-nav.launch" launches the environment topological map with waypoints designed for this environment, including set waypoints for the robot to stop and analyse the vineyard. localisation is given as the map was predefined but if a map wasn't present and we didn't have definite localisation the system could have worked with the implementation of an AMCL SLAM-algorithm that had been tested and proven to work. a example of what the topological navigation "topo-nav.rviz' can be seen in Figure 2 Below

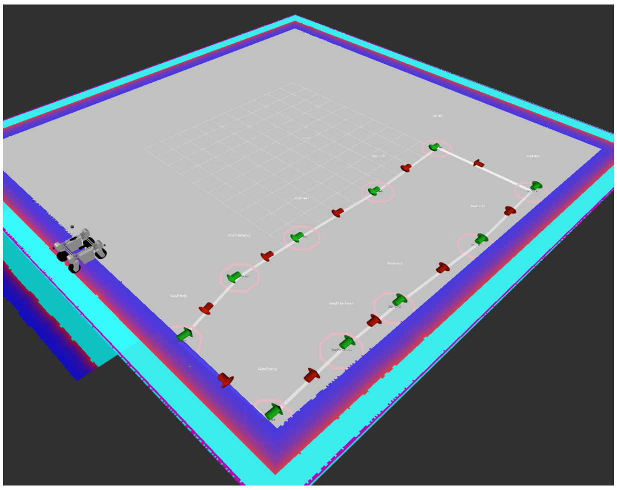

Figure 2 Topological Design

As seen in figure 2 the green arrows indicate waypoints at which the Thorvald robot will stop and process its environment .Using its HD RGB kinect2 camera, the robot will take snapshots of the vineyard and process each one, the first image processing technique the robot must do is to transfer the RGB image to an HSV colour space using the OpenCV library "RGB2HSV" function which allows for easier noticeable differences between the grapes, as seen in figure 3 below

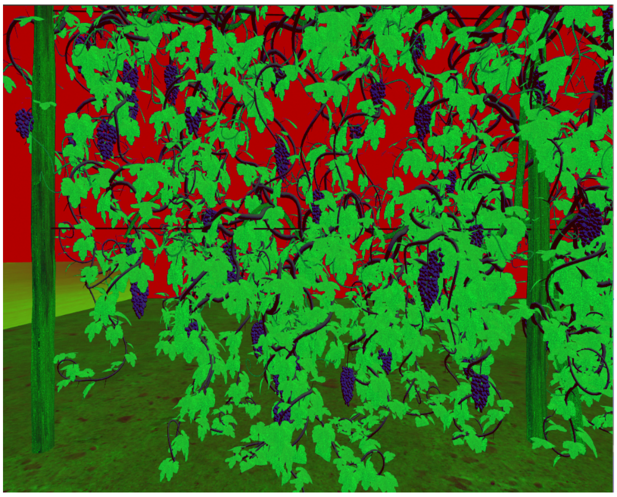

Figure 3 HSV Grape colour space

Once in the HSV colour space, the image can easily be thresholded with the "inrange" function of OpenCV to isolate the specific purple colour of the grapes and make every other non-inrange pixel black or equal to zero. It was found that a Hue range of 100 - 190 was best for isolating the grape colour, while keeping the Saturation and Value range between 25 - 255 (max). Once the grapes are isolated a multi-stage edge detection algorithm can be deployed to identify the grape bunches and transform the image to a binary format. The Canny algorithms was found to be effective for this process. evidence of this can be seen in figure 4.

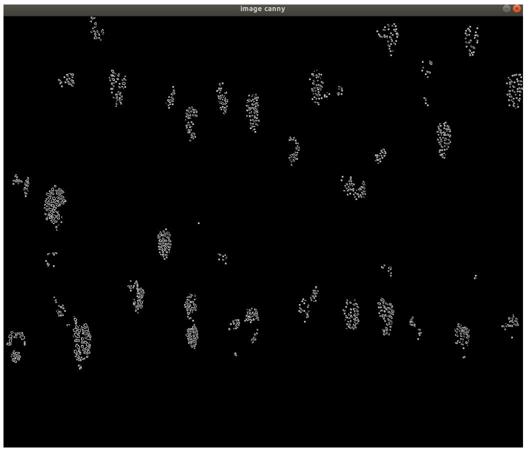

Figure 4 Canny algorithm

A series of morphological operations are carried out on the Canny filtered image to fill , separate and denoise the image to make it as accurate as possible. Some of these operations included dilation, erosion and pixel size filtering. This creates solid White blobs for the grape bunches, as seen in figure 5 below.

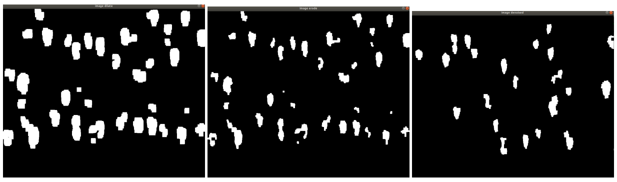

Figure 5 Dilated (left),Eroded (middle) and denoised (right)

The use of OpenCV's "findCountours" function which is another edge detection algorithm highlights each blob which is then sorted from left to right and counted. The centre point image coordinates is then found of each contour and a two stage TF transform process is carried out using the depth camera of the same frame and location of the kinect2 camera. The first transform includes finding the coordinates of each bunch from its image X and Y coordinates and height using the depth camera data for the location of the bunch relative to the robots specific location. The second transform includes taking the grape bunch location relative to the robots frame and making it relevant to the world/map frame. This data is then displayed in a overlay of the RGB image, circling, numbering and publishing the coordinates, as seen in figure 6 below.

Figure 6 Circled grape bunches

The coordinates of each grape are then appended to a point cloud every iteration and then passed through a filter to remove anomalous data (Passthrough.launch) and then Density-Based Spatial Clustering of Applications with Noise (DBscan) function which clusters together coordinates within a set range of each other (epsilon). This process removes duplicates and errors and can be counted to get the estimated yield of the grape vine once all scans are completed. the final point cloud of the clusters as well as the LiDAR data of the environment is mapped, see figure 7.

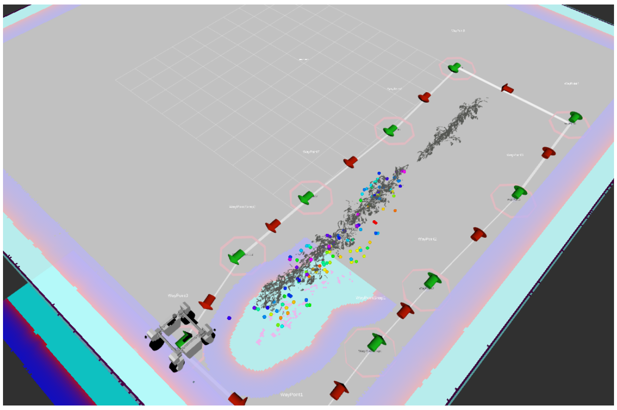

Figure 7 point cloud grape bunches

4 Evaluation and Discussion

Overall the project was a success however only perception was explored fully, the next steps would be to include the implementation of the robots own localisation using AMCL and EKF mentioned in the related work section. Thus to assist a roam algorithm, so the robot doesn't rely on waypoints and can traverse any terrain. Other maps would have be incorporated in the project such as the multi row vineyard however the gaps between the rows was to small for the Thorvald robot and meant that the vines where out of the field of view of the camera and would have not accurately detected all grape bunches. a solution to this would be to redesign the vineyard or Thorvald robot. Due to computational limitations of the hardware used to develop the code not all of the surrounding environment could be mapped, but if run on a system with a better GPU it would have had been able to plot all of the surroundings.

Other ways of identifying the grape bunch's would have been to use a deep learning neural network such a classification CNN or semantic segmentation network, however these processes take a lot of computational power and due to the non dynamic environment , would not have been necessary. But if this project was to implemented into a real would scenario they would be crucial.

A key aspect to the yield prediction is the DBscan, so to evaluate the best epsilon value for the system a series of tests running different epsilon values (x-axis) 10 time was carried out and a mean yield (y-axis) gathered, a gaph showing this can be seen in figure 8.

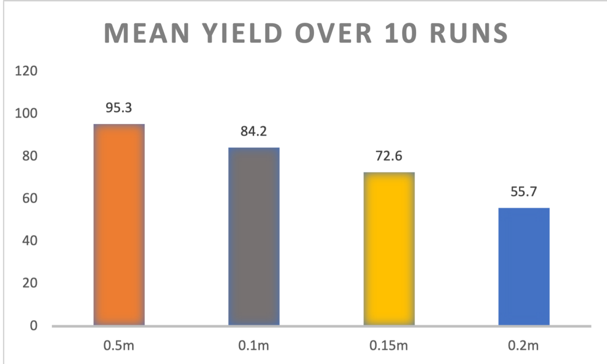

Figure 8

5 References

Lowenberg-DeBoer, J., Huang, I.Y., Grigoriadis, V. and Blackmore, S. (2020) Economics of robots and automation in field crop production. Precision Agriculture, 21 (2) 278-299.

Finger, R., Swinton, S.M., El Benni, N. and Walter, A. (2019) Precision farming at the nexus of agricultural production and the environment. Annual Review of Resource Economics, 11 313-335.

Chamelat, R., Rosso, E., Choksuriwong, A., Rosenberger, C., Laurent, H. and Bro, P. (2006) Grape detection by image processing. In: Anonymous IECON 2006-32nd Annual Conference on IEEE Industrial Electronics IEEE. 3697-3702.

Bharadwaj, K. and Gaur, S. A Zernike Moment based Modified CBIR System with Canny Edge Detector. International Journal of Computer Applications, 975 8887.

Xuan, L. and Hong, Z. (2017) An improved canny edge detection algorithm. In: Anonymous 2017 8th IEEE international conference on software engineering and service science (ICSESS) IEEE. 275-278.

Horie, T., Nakagawa, H., Centeno, H. and Kropff, M.J. (1995) The rice crop simulation model SIMRIW and its testing. Modeling the impact of climate change on rice production in Asia, 51-66.

Khaki, S., Wang, L. and Archontoulis, S.V. (2020) A cnn-rnn framework for crop yield prediction. Frontiers in Plant Science, 10 1750.

Kamilaris, A. and Prenafeta-Boldú, F.X. (2018) A review of the use of convolutional neural networks in agriculture. The Journal of Agricultural Science, 156 (3) 312-322.

Bedkowski, J., Pełka, M., Majek, K., Fitri, T. and Naruniec, J. (2015) Open source robotic 3D mapping framework with ROS—robot operating system, PCL—point cloud library and cloud compare. In: Anonymous 2015 International Conference on Electrical Engineering and Informatics (ICEEI) IEEE. 644-649

Crick, C., Jay, G., Osentoski, S., Pitzer, B. and Jenkins, O.C. (2017) Rosbridge: Ros for non-ros users. In: Anonymous Robotics Research Springer. 493-504.

Zou, Q., Sun, Q., Chen, L., Nie, B. and Li, Q. (2021) A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems,.