

Ground Plane Segmentation of Time-of-Flight Images for Asparagus Harvesting

Matthew Peebles[†], Shen Hin Lim, Lee Streeter, Mike Duke, Chi Kit Au
School of Engineering
University of Waikato
Hamilton, New Zealand
mp154@students.waikato.ac.nz

Abstract—A key task in processing time-of-flight images for the detection of asparagus spears is the identification and segmentation of the ground plane. In this paper, we aim to compare the performance of RANSAC with a modified version of an existing deterministic method, namely Hyun’s method. Each method is tested on scenes with varying amounts of field clutter. Additionally, a variety of camera angles are investigated. We find that both RANSAC and the proposed method produce ground plane predictions with a root mean square error of less than 0.05m and execute at a rate of approximately 0.056s per frame. However, RANSAC was shown to be much less reliable in high clutter scenes. The camera mounting angle is found to significantly affect the density and noise of points in time-of-flight images. These factors translate to significantly worse performance for both methods at low camera angles.

Index Terms—asparagus, asparagus harvesting, robotics, robotic harvesting, agricultural automation, point clouds, ground plane segmentation

I. INTRODUCTION

Global demand for agricultural automation has grown in recent years. This growth is fueled by a general decline of available labour, coupled with significant technological growth in the robotics sector. Many crops, such as asparagus, have yet to receive significant benefits from such innovations, mainly due to the unique complexities surrounding each crop’s biology. For asparagus, these complexities include a propensity for quick, sporadic growth and difficulty in controlling the structure of the crop bed, often resulting in large amounts of field clutter from weeds and foreign debris.

Many attempts have been made to automate the asparagus harvest [1]–[6], however, no such device has yet achieved commercial viability. This is partially due to shortcomings of the detection system. Excessive field clutter and the unstructured nature of asparagus beds pose a real challenge for automated detection. Tactile and photoelectric beam sensors, as well as several stereoscopic and monoscopic image processing techniques have been applied to the detection of asparagus spears with varying success [7]–[14].

An investigation [15] of several technologies concluded that time-of-flight imaging is a promising technology for the detection of asparagus spears for robotic harvesting. Leu et al. [16] have developed a harvester which utilizes time-of-flight (TOF) imaging for spear detection. A key aspect of Leu’s detection system is the ground plane segmentation. This process aims to segment points in the TOF images that belong to the asparagus bed, from those that belong to spears and other vegetation. This segmentation reduces the number of points needed to be processed by subsequent feature detection methods and enables the spear bases to be calculated geometrically based on each spears intersection with the resulting plane model. To achieve this segmentation Leu’s system uses random sample consensus (RANSAC).

RANSAC is commonly used for plane detection in scenes with large outliers. However, the non-determinant nature of RANSAC means that it is not often implemented in real-time systems. Hyun et al. [18] proposed a method for plane detection specifically for real time applications. Modifications to Hyun’s original method were made to make it more suited to outdoor, cluttered environments. This paper aims to find a robust method for the real-time identification and segmentation of ground planes in TOF images. Specifically, this paper investigates how both RANSAC and the modified Hyun’s method (MHM) perform when applied to scenes with varying amounts of field clutter. Each method is also tested at varying camera mounting angles. It is intended that the results of this analysis be used to aid in the design of a new automated asparagus harvester.

II. METHOD

A. Random Sample Consensus (RANSAC)

Random Sample Consensus (RANSAC) is an iterative method for estimating the best fitting parameters of a mathematical model to a set of data-points. Although Fishler and Bolles’ [17] original formulation of this method focused primarily on 2D features, the method is easily applied to 3D geometry making it a useful technique for detecting ground planes from point cloud data. This paper focuses on a simple implementation of RANSAC, provided

by *Point Cloud Library v1.8* [20]. The feature model is defined as a plane of the form:

$$ax + by + cz + d = 0 \quad (1)$$

using PCL’s `pcl::SACMODEL_PLANE` feature model. The input point cloud is down-sampled using PCL’s `pcl::VoxelGrid` filter with a 0.01m cubic leaf prior to being segmented using RANSAC. This aims to reduce the total number of points being operated on such that they are comparable in magnitude to the number of points sampled by the MHM.

Lost frames or significant error in the estimation of any given ground plane are undesirable due to the real-time nature of this application. For this reason, the probability threshold was set to 0.99 and the distance threshold was set to be 0.05m (5 times larger than the voxel grid’s leaf size). The maximum number of iterations allowed by RANSAC was determined by evaluating the root mean squared error (RMSE) of planes with respect to a ground truth plane, calculated using RANSAC on a scene with 0 clutter using 10000 iterations. A detailed explanation of the RMSE calculation is provided in section II(D). It was decided that 50 iterations provided an acceptable tradeoff between execution speed and observed errors.

B. Hyun’s Method

Woo et al. [18] developed a method for detecting planes in *organized* point clouds obtained from a Kinect V2 ToF camera (Microsoft Inc., Redmond, Wa, USA). A point cloud \mathbf{P} is considered *organized* if each point \mathbf{p} , in addition to being defined by a vector $\mathbf{p} = (p_x, p_y, p_z)$ in 3D space, can also be uniquely identified by some 2D coordinate (u, v) in the image space of the camera where $u, v \in \mathbb{Z}$. Organization of a point cloud in this way allows for efficient neighbourhood searches as there necessarily exists some set of orthogonal axes on which the relationship between adjacent points in the projection of the point cloud are known. Woo’s method begins by sampling a set of n points $\mathbf{S} = (s_1, \dots, s_n)$ uniformly in the image space of the point cloud such that each $s_i = s(u, v) = \mathbf{p}_i$. For each of these points Hyun’s method calculates the vectors:

$$v_1^i = p(u_i - 1, v_i) - s(u_i, v_i) \quad (2)$$

$$v_2^i = p(u_i, v_i + 1) - s(u_i, v_i) \quad (3)$$

in 3D space between each sampled point s_i and the adjacent points in \mathbf{P} which exist in the $-u$ and $+v$ directions respectively. A local normal vector at the sampled point is then calculated as

$$\mathbf{N}_i = v_1^i \times v_2^i \quad (4)$$

Once all the points have been sampled the normals associated with each point are clustered into groups G based on their angular correspondence using a threshold of 0.3 degrees. This is calculated as:

$$\mathbf{N}_i \cdot \mathbf{N}_j = |\mathbf{N}_i| |\mathbf{N}_j| \cos(\theta) \quad (5)$$

Additionally, the co-planarity of each sample point s is interrogated by checking if a line between each point in G is orthogonal to the normal vector \mathbf{N}_i defined at the point. This is calculated as:

$$(G_p - G_q) \cdot \mathbf{N}_i = 0 \quad (6)$$

where p and q are indices of two separate points in G . Points in G that do not satisfy Equation 6 are separated into a separate group associated with a parallel but non-coplanar plane.

Each group of normal vectors are then filtered based on a minimum member threshold. Finally, for each sampled point s_i in each cluster G a plane is fitted using QR decomposition to solve the linear least squares problem. The resulting model parameters are returned.

C. Modified Hyun’s Method (MHM)

The above formulation is reasonably successful at identifying plane normals when the precision of the point cloud is high. In their paper, Hyun et al. were concerned with detecting planes in point clouds of hull blocks and other small-scale models. This meant that the imaging done in their work was relatively close range which allowed for much more precise range imaging. In our application scenes are often imaged over several meters, and ultimately in outdoor environments where background lighting can be a significant problem. This means that the quality of data is much lower. When this method is applied to such a point cloud the results are wildly inaccurate.

A modification to Hyun’s method was made to mitigate these effects. Instead of calculating v_1 and v_2 using the *immediate* neighbors of each sampled point, the set of points from the point cloud that exist on the u and v axes respectively between each sampled point s_i were found and a linear least squared regression was performed using Eigen’s QR decomposition for each set of points. The result is a v_1 and v_2 that is much more resilient to noisy point clouds but has a lower feature resolution. For our application this is advantageous.

Additionally, the co-planarity check made by the original method was not calculated in the modified method. This is because for the application only the largest plane in the image was required. This plane was assumed to represent the ground plane and was calculated based on fitting a plane to the group of points G with the largest member count following segmentation via Equation 5.

For inlier segmentation the orthogonal distance d is calculated for each point $\mathbf{p} = (p_x, p_y, p_z)$ using:

$$d = \frac{ap_x + bp_y + cp_z - d}{\sqrt{a^2 + b^2 + c^2}} \quad (7)$$

and applying a threshold t of 0.05m.

D. Experimental Setup

RANSAC and the Modified Hyun’s Method (MHM) were compared in order to gauge their relative applicability for use in asparagus detection. Specifically, the



Fig. 1: laboratory setup showing the sand-bbox and Microsoft Kinect V2 mount

robustness of the methods for various camera mounting angles and field clutter levels were compared. Figure 1 shows the apparatus used to collect the data. The apparatus consists of a MDF sandbox measuring $2.8\text{m} \times 0.6\text{m}$ and a camera stand 0.9m above the sand surface. The Camera mount allows for angles between 25° and 65° to be achieved. These angles are measured as the angle between the optical axis of the camera and the horizontal plane of the sand bed. 3D printed asparagus spears (PLA) and various plastic shrubbery items were used to simulate the types of ground clutter present in real-world scenes. Seven different scenes ranging from no clutter, to extreme clutter (much worse than real-world scenes) were created. These scenes were denoted as clutter level 0-6 and can be seen in Figure 2 (a)-(g). A recording of the cameras data stream was made of each of these scenes for several different camera angles ($25^\circ, 35^\circ, 45^\circ, 55^\circ, 65^\circ$). An example of the point clouds generated using a 45° camera angle can be seen in Figure 2 (h)-(n). Generating a dataset in this way allowed both methods to be tested against the same scenes in real time while eliminating any variability that would result from true real-time imaging.

Point clouds were captured using a Kinect V2 ToF camera. Images were processed with a PC running a 3.5GHz *i5-5700k* processor and 32GB of RAM under an *Ubuntu 16.04_LS* environment. *Freenect2* [19] was used as the hardware driver and a *ROS_Kinetic* environment was used to control the various data streams and generate visualizations. *Point Cloud Library v1.8* (PCL) [20] was used for point cloud manipulation, and its constituent linear algebra library *Eigen* was used for various numerical operations.

Both RANSAC and MHM aim to fit a plane of the form in equation 1 to the input point cloud $\mathbf{P} \in \mathbb{R}^3$. Ground truth parameters of this model were calculated by applying each method to the zero clutter scene for each camera angle. This allowed subsequent clutter levels to be tested

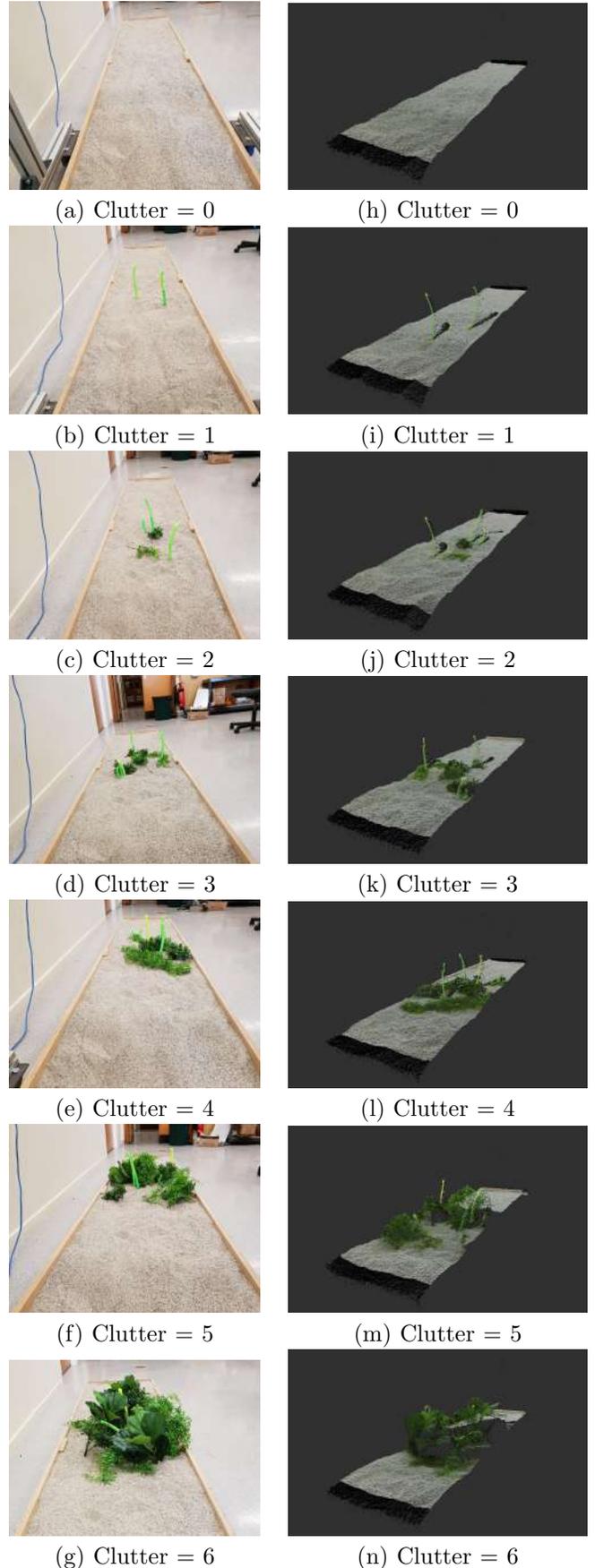


Fig. 2: Images of experimental setup and related point cloud for various clutter levels. Images taken at camera angle of 45°

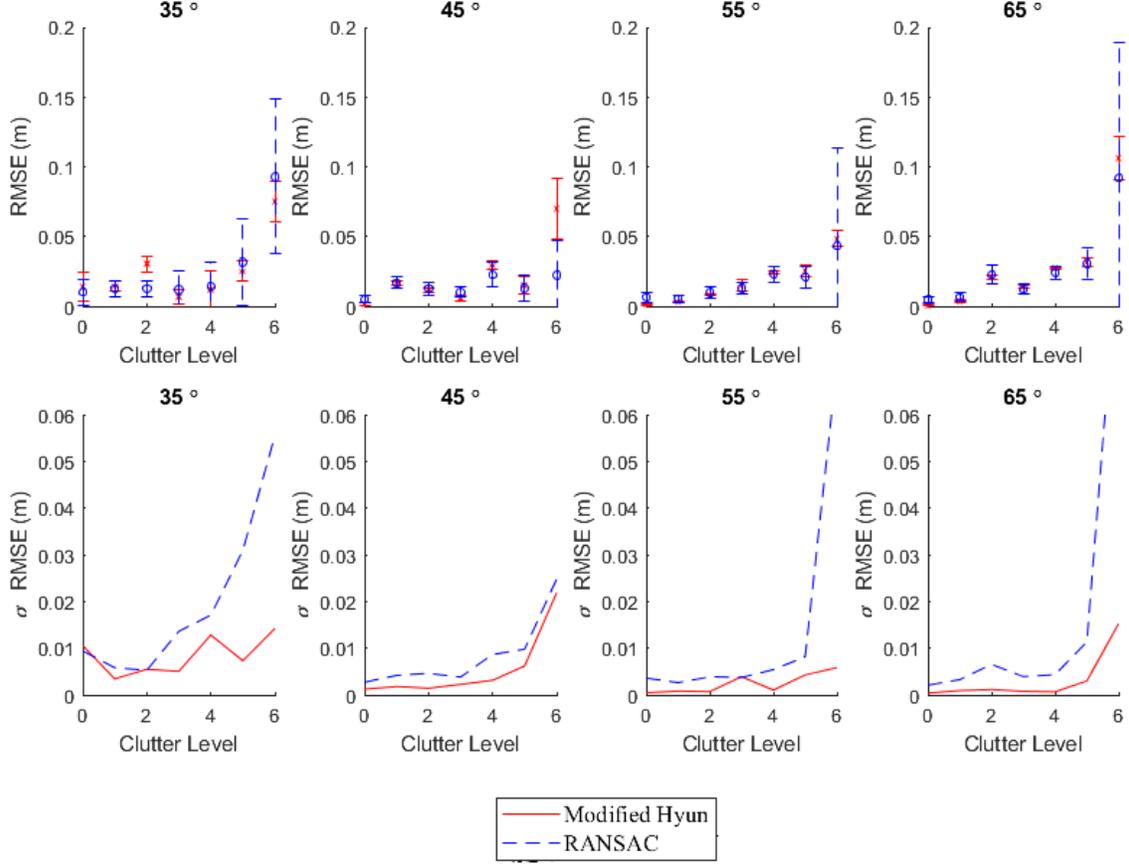


Fig. 3: Average RMSE and standard deviation of RMSE of plane predictions made using RANSAC and MHM at various clutter levels and camera angles.

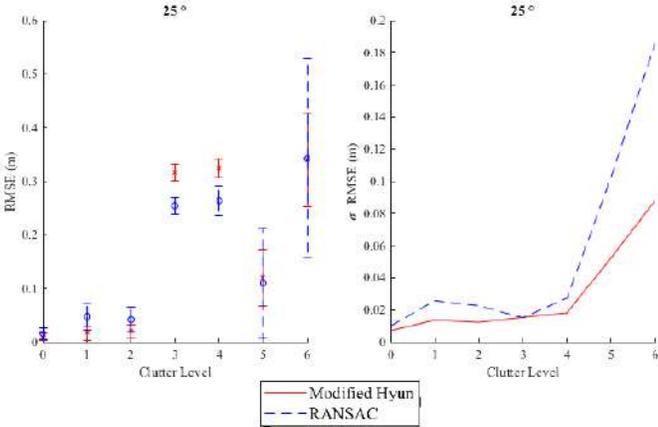


Fig. 4: Average RMSE and standard deviation of RMSE of plane predictions made using RANSAC and MHM at various clutter levels for a 25° camera angle. Note that the y-axis scale is significantly higher than plots for higher camera angles.

for each method with respect to that methods zero clutter performance. In order to quantify this performance a set of sample points, $\mathbf{B} = (b_1, \dots, b_N)$ where:

$$b_i = \{(x_i, y_i) | x_{\min} < x_i < x_{\max}, y_{\min} < y_i < y_{\max}\}$$

and $[x_{\min}, x_{\max}]$ and $[y_{\min}, y_{\max}]$ denotes the ranges of x and y values in \mathbf{P} was selected. For each plane prediction, the root mean squared error between the corresponding z coordinates were calculated as:

$$E = \sqrt{\frac{1}{N} \sum_{i=1}^N (z_i - z_0)^2} \quad (8)$$

where N is the number of points in \mathbf{B} , and z_i and z_0 refer to the z coordinate calculated for a given plane prediction and corresponding ground truth plane respectively.

For each camera angle and clutter level, both methods were used to generate 100 plane predictions. From these predictions, the average RMSE, as well as the standard deviation of RMSE, was recorded. These results can be seen in Figure 3.

TABLE I: Average executions times and standard deviation of execution times for RANSAC and MHM across all clutter levels for various camera angles.

RANSAC		
Angle	Average Execution time (s)	σ_t (s)
25	0.0296	0.0011
35	0.0385	0.0006
45	0.0508	0.0010
55	0.0578	0.0016
65	0.0557	0.0021
MHM		
	Average Execution time (s)	σ_t (s)
25	0.0056	0.0002
35	0.0075	0.0002
45	0.0099	0.0003
55	0.0113	0.0003
65	0.0127	0.0005

III. RESULTS AND DISCUSSION

Figure 3 shows that the RMSE of planes predicted using both RANSAC and MHM generally increases with increasing field clutter. Although this trend is ubiquitous across all camera angles tested, it is most pronounced at the extremes, namely at 25° and 65°. The RMSE of each method at low clutter levels are all consistent and typically show a RMSE of less than 0.05m for all camera angles > 25° which implies that the amount of field clutter is the main limiting factor for both methods.

High RMSE is observed for both methods at extreme camera angles due to the characteristics of the camera. At high camera angles, the field-of-view of the camera is focused on a smaller region of the field than for lower angles. This means that high angle images contain many densely packed and relatively close range points, while low angle images contain fewer, more sparsely distributed, long-range points. Points imaged by the Microsoft Kinect V2 are less precise at longer range. Low angle images are therefore lower quality due to the higher average range of points. This causes large RMSE in plane predictions in low angle images. This is exemplified by the RMSE values found at 25°, shown in Figure 4. RMSE errors for this case are up to 400% larger than for any other camera angles tested.

For high camera angles, large RMSE values are observed for high clutter scenes. This is due to the limited field-of-view resulting from the range of the imaging. Lower field-of-view means that the clutter is more predominant in the scene, resulting in fewer true inliers of the ground plane, understandably increasing the RMSE of both methods at high angles and clutter levels.

The standard deviation of the RMSE (σ_{RMSE}) for each method was also investigated. Figure 3 shows that σ_{RMSE} for both RANSAC and MHM are both less than 0.005m for low clutter levels and camera angles higher than 45°,

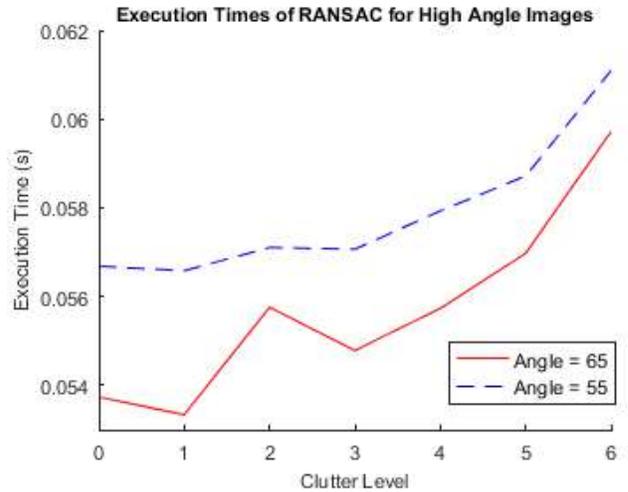


Fig. 5: Execution times of ransac at 55° and 65°. The plot shows that at high clutter RANSAC’s execution time increases

but rapidly increase at higher clutter levels. This value represents a σ_{RMSE} of approximately 10%. The rapid increase in σ_{RMSE} can be seen to be much faster for RANSAC than MHM which means that plane predictions made by RANSAC are much more unreliable. In high clutter cases, particularly at extreme camera angles, the σ_{RMSE} of RANSAC is upwards of 100%, signifying very little confidence in the plane prediction. The σ_{RMSE} for the 25° case was approximately 200% higher for most clutter levels. This indicates that low angle images, in addition to being more error prone, are also more unreliable in general. This is likely due to a reduction in point cloud quality at low angles.

The execution times of both methods were found to be constant over all clutter levels for low to moderate camera angles, however, execution times for RANSAC at camera angles of 55° and 65° seemed to increase at higher clutter levels. The average execution times over all clutter levels can be seen in Table I.

It should be noted that the execution times between methods cannot be directly compared as both RANSAC and MHM operated on different numbers of points due to their respective sampling schemes. Trends within each algorithm can still be observed for various clutter levels and camera angles. Both methods were able to perform at a speed of 0.056s per frame (18fps) which was deemed fast enough for real-time applications. This suggests that, using the algorithm parameters outlined in this paper, the execution time is not a limiting factor for the application of these methods to a real-time system.

MHM exhibited a constant execution time over all clutter levels for each camera angle. In general lower camera angles resulted in faster execution times, however, this effect is minimal, with a maximum difference of 0.007 seconds between the 25° and 65° cases. This is because

MHM samples each scene with the same grid of points. Low angle images are more sparse, resulting in slightly fewer points being sampled by MHM. This leads to slightly faster execution times for low angle images.

RANSAC utilised all 50 iterations for all predictions in this paper due to the high probability factor of 0.99. Differences in observed execution times are therefore primarily due to the varying number of input points in each scene. Low camera angles were observed to execute the fastest and to be invariant to various clutter levels. However, at high camera angles, RANSAC exhibited longer execution times for high clutter scenes. This trend can be seen in Figure 5. It can be seen from the figure that at high clutter RANSAC's execution time exceeds 0.06s per frame (approximately 16 fps) which was considered to slow for real-time applications. This is because points in low angle scenes tend to be sparse and span far in comparison to the height of clutter, resulting in fewer points being passed through the voxel filter. In contrast, high angle images are dense and span much smaller distances. This means that the observed perturbations of the points due to clutter are quite large, resulting in a larger number of points being selected by the voxel filter.

IV. CONCLUSION

Ground plane predictions were made by applying RANSAC and MHM to point clouds taken for a variety of different camera angles and field clutter levels. The RMSE of plane predictions made by both methods are similar, and for clutter level 0 to 5 are generally below 0.05m. At lower camera angles, and in the presence of extreme clutter, both methods produce unacceptably large RMSE.

The σ_{RMSE} of planes predicted using RANSAC is larger in general than planes predicted using MHM at the same camera angle and clutter level. At high clutter levels, σ_{RMSE} of planes predicted using RANSAC become large, often exceeding the magnitude of the RMSE. This indicates low confidence in the plane prediction, which is undesirable for real-time applications.

The σ_{RMSE} of planes predicted using MHM was shown to be generally smaller for the 45° and 55° camera angles, particularly at high clutter levels. This indicates that at moderate camera angles MHM is more robust to clutter than RANSAC.

The execution times of both methods were shown to be generally within the real-time range of 0.05s per frame for each scene using the algorithm parameters outlined in this paper. It was shown that high angle images generally result in higher execution times due to higher point density resulting in a higher number of input points.

We conclude that although the RMSE of planes predicted by both methods is comparable for similar camera angles and clutter levels, the improved resilience to high clutter offered by MHM makes it more applicable to the detection of ground planes for asparagus harvesting.

A. Future Work

A holistic understanding of plane segmentation is required for developing a robust plane segmentation strategy. Future work will be concerned with generating similar comparisons with other methods. In particular, Hough Transform, and region growing based methods will be investigated.

REFERENCES

- [1] Haws, S.K., Stalk selective harvesting machine. 1977, Google Patents.
- [2] Humburg, D.S. and J.F. Reid. Field performance of machine vision for the selective harvest of asparagus. in International Off-Highway and Powerplant Congress and Exposition, September 9, 1991 - September 12, 1991. 1991. Milwaukee, WI, United States: SAE International.
- [3] Strauß, J., Development of an Automatic harvesting system for green asparagus with stalk detection in Ambient Light. 2014. p. 1-14.
- [4] Lund, W.J., Asparagus harvester. 1985, Google Patents.
- [5] Turkington, J.O., Asparagus harvesting machine. 1956, Google Patents.
- [6] Matteoli, A.J., Asparagus harvester. 1952, Google Patents.
- [7] Haws, S.K.R., WA, US, Selective Harvester. 2010: United States.
- [8] Arndt, G., R. Rudziejewski, and V.A. Stewart, On the future of automated selective asparagus harvesting technology. *Computers and Electronics in Agriculture*, 1997. 16(2): p. 137-145.
- [9] Du, C., Z. Qin, and W. Shumao, Current Status and Technical Challenges of Asparagus Mechanical Harvesting. 2010 Pittsburgh, Pennsylvania, June 20 - June 23, 2010, 2010.
- [10] Baylou, P., Amor, B.E.H, Monsion, M., Bouvet, C., Boussau, G., Detection and three-dimensional localization by stereoscopic visual sensor and its application to a robot for picking asparagus. *Pattern Recognition*, 1984. 17(4): p. 377-384.
- [11] Humburg, D.S. and J.F. Reid, A machine vision algorithm for identification of harvestable spears of asparagus. 1990. p. 13pp.
- [12] Irie, N., Taguchi, N., Horie, T., Ishimatsu, T., Asparagus Harvesting Robot Coordinated with 3-D Vision Sensor. 2009 Ieee International Conference on Industrial Technology, Vols 1-3, 2009: p. 408-413.
- [13] Grattoni, P., Cumani, A., Guiducci, A., Pettiti, G., Automatic harvesting of asparagus: an application of robot vision to agriculture. 1994.
- [14] Clary, C.D., Ball, T., Ward, E., Fuchs, J., Durfey, J.E., Cacialieri, P., Folwell, J., Performance and economic analysis of a selective asparagus harvester. *Applied Engineering in Agriculture*, 2007. 23(5): p. 571-577.
- [15] Peebles, M, Lim, S.H., Duke, M, and C. K. Au. (2017) Overview of sensor technologies used for 3D localization of asparagus spears for robotic harvesting. *Applied mechanics and Materials*, vol 884. pg 77-85.
- [16] Leu, A., Razavi, M., Langstädtler, L., Ristic-Durrant, D., Raffel, H., Schenck, C., Gräser, A., Kuhfuss, B., "Robotic Green Asparagus Selective Harvesting," in *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2401-2410, Dec. 2017. doi: 10.1109/TMECH.2017.2735861
- [17] M. A. Fischler and R. C. Bolles. (June 1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, vol 24, no 6.
- [18] H. W. Yoo, W. H. Kim, J. W. Park, W. H. Lee and M. J. Chung. Real-Time Plane Detection Based on Depth Map from Kinect. *IEEE ISR 2013*, Seoul, 2013, pp. 1-4. doi: 10.1109/ISR.2013.6695732
- [19] Lingzhu Xiang, Echtler, F., Kerl, C., Wiedemeyer, T., Lars , Hanyazou, Alistair . (2016, April 28). Libfreenect2: Release 0.2 (Version v0.2). Zenodo. <https://doi.org/10.5281/zenodo.50641>
- [20] R. B. Rusu and S. Cousins, 3d is here: Point cloud library (PCL), in *Proc. 2011 IEEE Int. Conf. Robot. Automat.*, 2011, pp. 14.