UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2020

Title: Orchard Row Detection

Author: Leonard Chao Lun Chuang

Project Abbreviation: ORCHARD

Supervisor(s): Assoc. Prof Patrick Marais

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	10
System Development and Implementation	*0*	*20*	15
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	80

Orchard Row Detection from UAV Imagery

Leonard Chao Lun Chuang
CHNLEO008
University of Cape Town
CHNLEO008@myuct.ac.za

1. Abstract

Satellite imagery provides high-resolution images to study the variations and conditions of crop fields, but prohibitive costs and low accessibility suggest that alternative imaging methods should be used. Currently, the use of Unmanned Aerial Systems (UASs), which is a form of Low Altitude Remote Sensing (LARS) systems, have presented a new area of study related to Precision Agriculture (PA). The most popular UASs are Unmanned Aerial Vehicles (UAVs), which are a means for inexpensive data gathering of high spatial and temporal resolution imagery i.e. the images are of good quality and are obtained daily. The images are crucial to initial PA procedures such as field mapping, which includes vegetation detection and row identification. The lack of row detection techniques for orchards from aerial images are attributed to the additional challenges present in orchard data. A modified Hough Transform with post-processing is applied to detect crop rows. However, the algorithm is altered for when orchard data is: (1) highly dense growth areas with overlapping between trees which causes poor initial tree separation and detection, and (2) plantation methods which introduce curved rows to suit the terrain. The algorithm is able to successfully detect rows which are clearly separate with moderate curvature.

Keywords Hough Transform, Row detection

2. Introduction

The increase in crop yield to accommodate for the drastically increasing human population will soon reach a peak [1] which cannot be overcome. Indefinite expansion of agricultural land necessary for crop yield cannot occur as the land will either be occupied or unavailable. Other hindrances to increasing crop yield including pests and disease, can decrease the crop yield up to 40% [2]. A modern solution to control pests and diseases while simultaneously administering optimal resources, measure biomass and monitor crop growth is Precision Agriculture (PA).

PA is the application of geospatial information to identify field variations and conditions with intentions of maximising and monitoring crop yield while minimising resource expenditure through various site-specific farming techniques. The main processes of PA are data collection, field mapping, decision making and management practice [3]. Field mapping is a crucial initial procedure that entails vegetation detection and row identification from field data. Although algorithms able to recreate orchard rows from ground level images have been developed, research is moving towards using aerial images. The use of UAVs to navigate fields and collect aerial data is becoming more common as these inexpensive technologies become more accessible [4].

Figure 1. Three orchard datasets provided. From left to right: simple case with good spacing, extreme case with overlapping trees and dense growth areas and complex case with curved rows and erratic spacing.

The orchard data contains trees detections as polygons in Global Positioning System (GPS) coordinates and a boundary which represents the orchard enclosing the trees. An assumption when dealing with a plantation is that there is some planting technique which the farmer followed. A generally used plantation technique is row planting [5] where the crops are planted in parallel rows with set inter and intra row spacing. Trees require more space and can easily overlap if there is poor spatial planning. The overlapping interferes with the tree identification process which can drastically affect the row identification process as trees are located in the inter row spacing. Other issues include erratic intra row spacing which requires additional parameterisation when solving the problem using a closest distance measure. Unfortunately, tree height information and contour information were not captured which could prove useful when creating point cloud maps or contour boundaries. This limits the possible solutions that are applicable.

Possible solutions include algorithms that directly relate to row identification in tree data such as edge reconstruction and point cloud mapping [6]. Point cloud mapping requires height and contour data which is not available for this project. Other solutions such as The Hough Transform, is a line detection algorithm commonly used on crop imagery to identify crop rows [7-9]. The algorithm can easily identify straight lines, but its performance drastically decreases when the lines are curved and do not belong to a geometric shape [9]. The row detection problem can be viewed as identifying straight or moderately curved lines from a set of points on a 2D plane.

This project shows a modified Hough Transform, comprised of the Probabilistic Hough Transform and a post processing process, can reliably identify rows when there is clear inter row spacing and moderate curvature. The Probabilistic Hough Transform generates initial row segments which forms part of a larger row. These line segments are then expanded along points which are part of the row until a row prediction is made.

This paper is laid out as follows: Section 3 introduces the background and related works for agricultural row detection which are used as a basis for the algorithm. Section 4 describes how the data is transformed to a suitable format for the algorithm and steps of the algorithm. Section 5 is a discussion of the results and diagrams depicting the results. Section 6 provides an overview of what conclusions can be drawn from the project and any adaptations or work to be done should the project be extended and improved.

3. Background/related work

The Hough Transform is a popular line detection technique introduced by Hough in 1972 [10]. The Hough Transform is widely used to identify crop and vine rows from aerial imagery with high success rates [11]. This technique works best for straight row detection but modifications to the input data, through segmentation and tiling, have been implemented to accommodate for moderate curvature within the rows. These modified techniques break down a larger input image with curved rows into tiles or segments that contain row segments that are almost straight. The tiles are then combined so that the straight segments join to form a curved row.

The Hough Transform attempts to find the optimal parameters for shapes such as straight lines or circles and ellipses in the parametric space through a voting scheme. The votes are generated from the points in the input image which are represented as tree centroids in this project. A simplified version of the Hough Transform is the Probabilistic Hough Transform[12]. This simplified technique only returns the two end points of line instead of returning parameters in the parametric space. Since the trees do not fall on a straight line exactly, the Probabilistic Hough Transform can be used generated an initial guess of the row layout. An additional post processing is used join rows and expand row in this project.

Some assumptions when the Hough Transform, or various versions of the technique, were applied include perfect parallel rows, equidistant spacing for both inter and intra row spacing and the same row directionality for all rows. These assumptions do not hold for orchard data. From the previous works, it is clear the Hough Transform can identify straight rows with high accuracy but needs an additional post processing step to correct for errors.

Edge and boundary construction are techniques used to find an optimal polynomial which fits through a series of points [13, 14]. This technique is often used for image reconstruction. It applies a moving average least squares regression to a series of points and picks the polynomial with the least error. Although the technique is similar to identifying rows from a set of points, the prediction suffers greatly when there is considerable noise in the plane. Orchard data contains several points with no internal boundaries which means that the technique is not suitable due to excessive noise.

Point cloud mapping of orchard data is another method which can be used to identify tree rows[15]. This method uses height of the trees and the contours lines to determine the directionality and underlying plantation pattern. Unfortunately, the orchard data for this project does not contain any height data or contour lines so contour mapping is not applicable.

4. Methodology

4.1 Overview of the algorithm

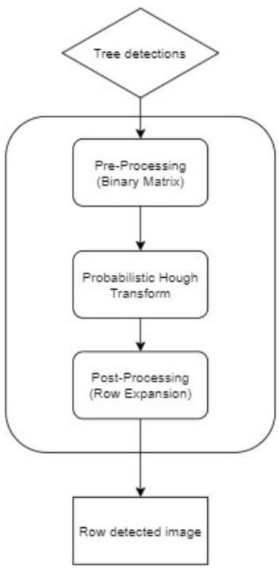


Figure 2. Basic System architecture diagram depicting the transformation of data and flow of inputs and outputs.

Figure 2 shows the main components of the algorithm and where inputs are consumed to produce outputs for the next component.

```
Algorithm Hough Transform with Post Processing
1:  Transform input data into correct format
2:    Apply Probabilistic Hough Transform
3:    For each (line segment detected above) do
4:      While (error less than twenty):
5:        Find the nearest neighbours
6:          While (nearest neighbours exist) do
7:            Perform gradient check
8:            If (gradient correct)
9:              Add point to row and extend row in red
10:             Remove reference point from plane
11:             Update reference point
12:             Update nearest points
13:           Else
14:             Increase error by one
15:             Add point to row and extend row in blue
16:             Remove reference point from plane
17:             Update reference point
18:             Update nearest point
19:         End
20:     End
21: End
```

Figure. 3: High-level representation of the algorithm implemented

Figure 3 is a high-level pseudocode description of the steps implemented in the algorithm. The algorithm consists of two main parts, the application of the Probabilistic Hough Transform listed at step 2, and the Post Processing from step 3 to step 18. Some pre-processing of the data is required to convert the data into a suitable format.

4.1 Pre-processing and altering input data

The initial data is a set of points in Global Positioning System (GPS) coordinates where each point, representing a tree, is given as a polygon with a degree of confidence in the detection. Enclosing the set of points is a boundary which represents the Orchard enclosure. An aerial image of the Region of Interest (ROI) is supplied in the Tagged Image File (TIF) format which can be used to help visualise the ROI, but includes unwanted structures such as housing, roads and additional trees or foliage.

The set of points is converted from GPS coordinates to a cartesian coordinate system so the data can be read and depicted on a 2D plot. A binary image of the ROI is required as input for the Hough Transform. Since the TIF contains unwanted obstructions, it is not suitable to convert to a binary image, hence a conversion of the points into a binary matrix the size of the orchard is used. This matrix conversion is performed by creating a zero matrix the size of the orchard boundary and changing inputs to indicate the presence of a tree. However, a matrix coordinate system cannot reference cells with floating point indices, so the point coordinates are converted into integers. The scaling process of the points result in some information loss, but the scaled points still reflect the orchard area with the correct trees. Each tree coordinate is then matched to the corresponding matrix coordinate and the matrix entry is changed from a zero to indicate the presence of a tree.

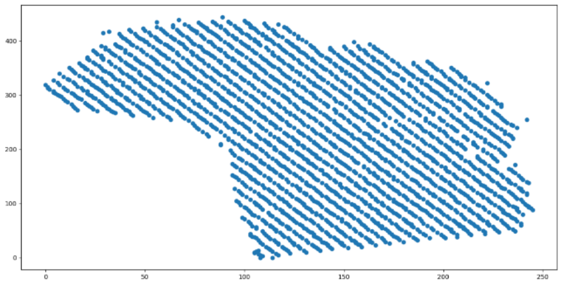


Figure 4. Extracting the only the tree points from the data which is converted to a binary matrix

4.2 Probabilistic Hough Transform

The algorithm consists of two parts: The Probabilistic Hough Transform and post-processing. The row detection problem can be viewed as identifying lines from a set of points. It begins by predicting initial line segments by applying the Probabilistic Hough Transform with the binary matrix as input. These line segments serve as the basis for the post-processing which expands and corrects the initial guess. Full row predictions cannot be made since the points do not necessarily lie on a straight line and do not have a set intra and inter row spacing. This introduces the need for a post processing step to determine the full row.

4.3 Post processing

```
Algorithm Post Processing
1.  For each (line segment detected above) do:
2.    While (error less than twenty):
3.      Find the nearest neighbours
4.        While (nearest neighbours exist) do:
5.          Perform gradient check
6.          If (gradient correct):
7.            Add point to row and extend row in red
8.            Remove reference point from plane
9.            Update reference point
10.           Update nearest point
11.         Else:
12.           Increase error by one
13.           Add point to row and extend row in blue
14.           Remove reference point from plane
15.           Update reference point
16.           Update nearest point
17.       End
18.   End
19. End
```

Figure 5. Pseudocode for the post processing.

The post-processing is applied to each line segment predicted by the Hough Transform. For each line segment predicted, the algorithm extracts one of the end points and expands the row to the nearest point. This is listed from step 4 to step 18 of the pseudocode in figure 5.

The nearest neighbour is limited to a square of size a x a. The size of the square is a parameter that can be changed. A square that is too small will result in neighbouring points not being found leading to an early termination of the row expansion process. A square that is too big will find neighbouring points that are from adjacent rows and cause row merging leading to incorrect row identification. A reference point, starting with the initial end point, is used as the centroid of the square. This reference point updates to the neighbouring points as the row expands.

When a nearest neighbouring point is found, a gradient check is performed. The gradient is calculated between the two most recent points in the row and is used as a basic check for the directionality of the row. Additionally, an interval is calculated from and added on to the gradient. This interval is set at on either side of the previous gradient. The size of the interval is another parameter which can be altered. One concern when searching for the optimal interval size is the volatility of the gradient. Gradients of lines nearing vertical lines are the most volatile as these tend toward being undefined.

Neighbouring points that fall within the gradient bound are added to the row with a red line segment which indicates correct prediction. Neighbouring points that do not fall within the bound are still added to the row but with a blue line segment instead, which indicates possible incorrect predication. Alongside a blue line segment, an error counter is used which is incremented whenever a possible incorrect prediction is made. This error counter can be set for step 4 of the algorithm so that the line expansion process terminates when a certain number of errors are reached. The original point is removed from the set of points to prevent it from being used in multiple rows.

This post processing process repeats itself for each of the initial line predations by starting with an end point and expanding to the nearest neighbour until there are no more nearest neighbouring points.

5. Results and Discussion

The algorithm was developed in Python and used the scikit-image for the Probabilistic Hough Transform. Its performance has been tested on three different input datasets of varying difficulties. The tests were run on a laptop running windows 10 with a core i5 processor and Nvidia GTX 960M GPU. To ensure the reliability of the algorithm, it was run three times on each dataset to check if the output was the same each time.

As validation, rows were manually labelled using geographic information software (QGIS) as the ground truth. Since evaluated row layouts were not provided, we hand-labelled a possible row layout for the datasets. The results are expressed as a percentage of row length with respect to manual detections.

5.1 Evaluation metrics

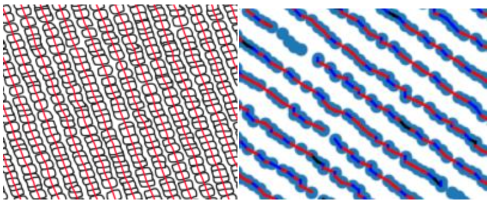
Table 1

Evaluation Criteria of Row detection results

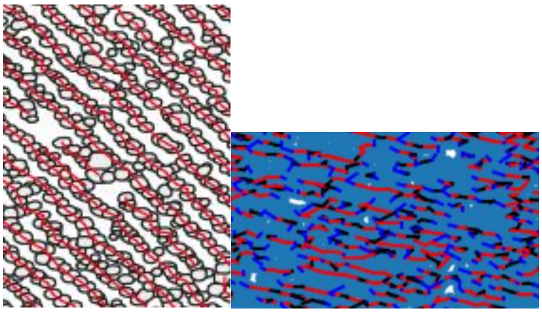
Case		Meaning
1.	Good detection	Correct detection of row, matching manual evaluation
2.	Over-detection	Length of detected row greater than manual evaluation
3.	Under-detection	Length of detected row smaller than manual evaluation
4.	Missed Detection	Row completely not detected

The five classifications listed in table 1 are used to evaluate the performance of the algorithm. The experiment aims to evaluate how robustly and accurately the algorithm can identify rows. The robustness refers to how general the algorithm is and is measured by testing the algorithm on different input data. It also measures the algorithm's ability to overcome different features such as different row layouts, both curved and straight, and densely populated orchards with erratic tree space and no clear row separation. The accuracy refers to how correct the length of the detected row is when compared to the ground truth. This is used to check if the algorithm combines different rows but also if the algorithm terminates row detection too early.

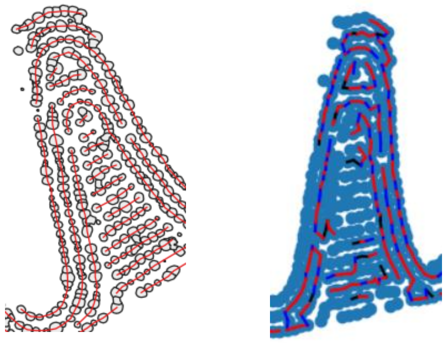
5.3 Results



(a)



(b)



(c)

Fig 6. Manually detected rows on the left column and algorithmic detections on the right column. (a) depicts a simple case of straight rows and consistent spacing. (b) depicts an extreme case of poor spacing and (c) depicts curved rows with erratic spacing.

Figure 6 shows diagrams containing the manual row detections (left) and the algorithmic detections (right). Only segments of the orchard are shown, each highlighting specific features which are goals the algorithm was built to achieve. The full detections are in the appendix. Figure 6 (a) shows a segment of a simple input because of straight rows with consistent inter and intra row spacing and the inter row spacing is greater than the intra row spacing. The results show the algorithm can correctly detect rows with little under or over classification. Figure 6 (b) shows a densely populated area where limited manual detection was possible but was not the case for the entire input. The results reflect very poor attempts of row detection. Figure 6 (c) shows a combination of curved and straight rows with erratic row spacing and drastically different row lengths. It can be seen that the algorithm can detect more than just moderate curves but can also detect quadratic curves. Noticeably, the differing row lengths affect the detection process as only either long or short rows are detected but not both. Below in Table 2 are the results the algorithm achieved on each of the different datasets.

Table 2
Results of algorithm (expressed as a percentage of row length with respect to manual detections) applied to three different inputs

		Input A	Input B	Input C
1.	**Good detection**	83.62%	N/A	40.01%
2.	**Over detection**	4.91%	N/A	21.90%
3.	**Under detection**	6.56%	N/A	13.33%
4.	**Missed detection**	4.91%	N/A	24.76%

The results were limited to the number of datasets provided by the time of testing. The results for input B are discarded since the manual labelling process could not be completed as rows could not be identified without professional input. Rather the detection image is supplied as supplement to the results. Additionally, a mean column per classification would not reflect the true performance as the datasets differed too drastically in complexity and features.

For input A, a high 83.62% of good detections was achieved. Presence of over and under detections as well as missed detections exist but are within acceptable ranges. Input C had a low 40.01% good detections with a high missed detection of 24.76%. The over and under detections are also quite high at 21.90% and 13.33%. On a surface level these error rates are considered too high.

5.4 Discussion of results

The algorithm was developed to detect rows curved rows and rows within dense growth areas. Only the ability to detect curved rows was achieved. It was able to detect rows with moderate curvature and also rows with quadratic curvature.

The inability to correctly identify full rows within dense growth areas is likely due to the high volatility of the gradient parameter. Since the points within the rows can have their gradients change quite drastically between any two points, it becomes very difficult to find the optimal gradient bound to use as a constraint.

Besides the gradient parameter, the inter and intra row spacing can be quite erratic which also affects the size of the square in which to locate the nearest points. Erratic spacing increases the number of over and under detections which affects the presentation of the results. It can be seen in Figure 6 (c) that longer rows were identify instead of shorter rows and not a combination of long and short rows. This is because of the erratic row spacing which can easily be fixed with additional iterations of the algorithm.

6. Conclusions and Future Works

Introducing more iterations to the algorithm with different parameters can help the algorithm when a combination of short and long rows is present. This will decrease both the under and over detection classifications. A more robust parameter such as an angle of change can be used to replace the gradient parameter which is too volatile.

Future works for the algorithm include optimisation of the parameters through parameter auto-tuning and exposure to more datasets. Parallelisation can be included in the post processing steps of the algorithm, so the expansion of each initial segment is run independently. Inclusion of more features such as height data or contour can help build smaller internal boundaries to aid the directionality of the row expansion.

In conclusion, the algorithm can detect straight and curved rows, but the accuracy of the algorithm suffers under dense areas with erratic spacing. This can be improved through better parameter tuning and auto-tuning.

7. Acknowledgements

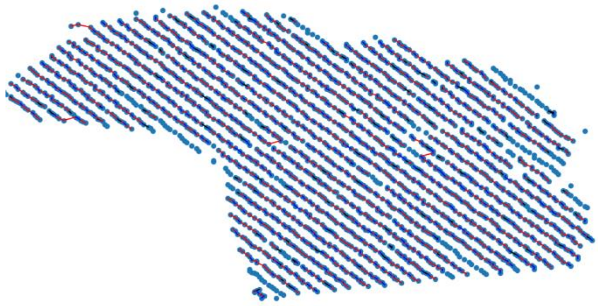
Acknowledgements to the National Research Foundation (NRF) for funding and Aerobotics for providing the orchard datasets.

8. References

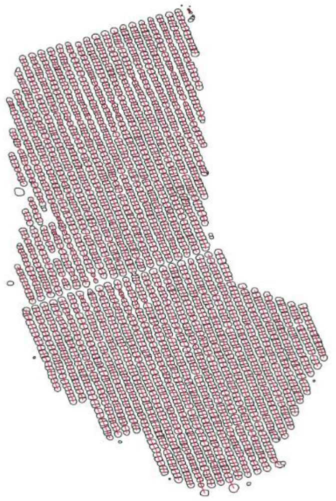
[1] Oerke, E. Crop losses to pests. *The Journal of Agricultural Science*, 144 (2006), 31.

[2] FAO-FOOD and ORGANIZATION, A. *The state of food and agriculture*. Livestock in the balance Rome, City, 2009.
[3] Zhang, C. and Kovacs, J. M. The application of small unmanned aerial systems for precision agriculture: a review. *Precision Agriculture*, 13, 6 (2012/12/01 2012), 693-712.
[4] Stefas, N., Bayram, H. and Isler, V. Vision-based UAV navigation in orchards. *IFAC-PapersOnLine*, 49, 16 (2016), 10-15.
[5] Hassanein, M., Khedr, M. and El-Sheimy, N. CROP ROW DETECTION PROCEDURE USING LOW-COST UAV IMAGERY SYSTEM. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).
[6] Comba, L., Biglia, A., Aimonino, D. R. and Gay, P. Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. *Computers and Electronics in Agriculture*, 155 (2018), 84-95.
[7] Comba, L., Gay, P., Primicerio, J. and Ricauda Aimonino, D. Vineyard detection from unmanned aerial systems images. *Computers and Electronics in Agriculture*, 114 (2015/06/01/ 2015), 78-87.
[8] Pérez-Ortiz, M., Peña, J., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C. and López-Granados, F. A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method. *Applied Soft Computing*, 37 (2015), 533-544.
[9] Soares, G. A., Abdala, D. D. and Escarpinati, M. *Plantation Rows Identification by Means of Image Tiling and Hough Transform*. City, 2018.
[10] Duda, R. O. and Hart, P. E. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15, 1 (1972), 11–15.
[11] Rovira-Más, F., Zhang, Q., Reid, J. and Will, J. Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 219, 8 (2005), 999-1010.
[12] Kiryati, N., Eldar, Y. and Bruckstein, A. M. A probabilistic Hough transform. *Pattern recognition*, 24, 4 (1991), 303-316.
[13] Xie, H., Wang, J., Hua, J., Qin, H. and Kaufman, A. *Piecewise C/sup 1/continuous surface reconstruction of noisy point clouds via local implicit quadric regression*. IEEE, City, 2003.
[14] Gordon, W. J. and Wixom, J. A. Shepard's method of "metric interpolation" to bivariate and multivariate interpolation. *Mathematics of computation*, 32, 141 (1978), 253-264.
[15] Weiss, M. and Baret, F. Using 3D point clouds derived from UAV RGB imagery to describe vineyard 3D macro-structure. *Remote Sensing*, 9, 2 (2017), 111.

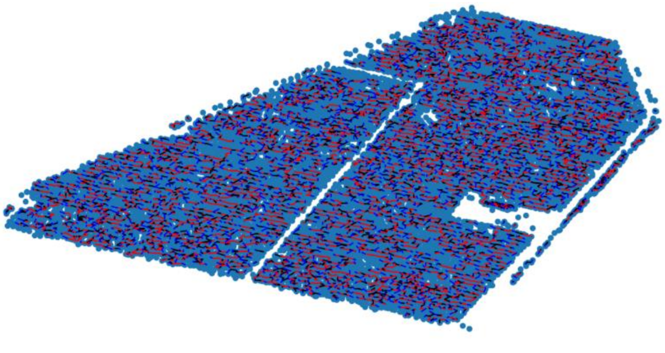
Appendix A



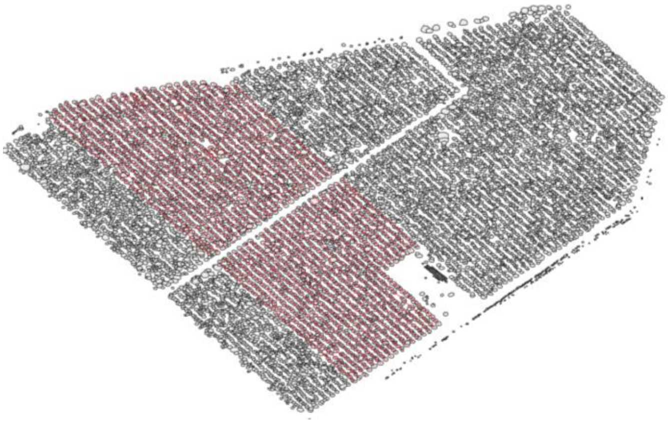
Results for input A



Ground Truth for input A



Results for input B



Ground Truth for input B

Results for input C

Ground Truth for input C