Detecting Tramways in Crops for Robot Navigation

Peter Nicholls
Department of Mechatronics Engineering
University of Canterbury
Christchurch, New Zealand

Richard Green
Department of Computer Science
University of Canterbury
Christchurch, New Zealand

***Abstract*—A useful tool for agricultural robot developers to have is the ability for the robot to be able to detect crop rows. There have been many methods developed for this purpose, but this paper proposes a real-time method that works in a variety of conditions.**

The proposed method involves four steps: a grayscale transform, skeletonization of crop rows, a Hough transform and then line filtering. The last two steps are iterated until the disired number of crop rows have been found. This was able to find 55.7% of the lines in an average of 32.1ms.

***Index Terms*—Crop Row Detection, Agricultural Robots, Iterative optimization, Skeletonization**

I. Introduction

Agricultural robotics is an emerging field which will improve many aspects of farming. This ranges from increasing the area of crops a single farmer can manage, to increasing the quality of the crops by having robots micro-manage them (such as spraying weeds). One issue with arable crops, which is more evident when crop fields are larger, is that they are vulnerable to being eaten by birds. Approximately 5-10% of cereal crops in New Zealand are destroyed by birds [1]. Although some measures can be taken to deter birds, they are not very effective as birds get used to them. One proposed solution is to have a robotic platform move on tramways between crops and scare birds. This way the bird would not get used to the deterrents and would hence decrease the loss of crops.

September 17, 2017

II. Background

There are many different algorithms which have been tested to detect crops and crop rows, each of which work well in some conditions and fail in others. It seems that all of these algorithms are intended for finding where the crops are so they can be sprayed or differentiated from weeds [2] [3] as opposed to where a robot might be able to drive along. The majority of these algorithms rely on the plants being greener than the soil and are decent when this assumption is correct, but some plants are not, such as wheat. A good summary of different algorithms are found in an article by J. Romeo et al [4], some of which are outlined below:

A. Horizontal Strips

This method converts the image to a binary one, based on enhancing the green values before thresholding [5]. It then slices the image into horizontal strips, and plots the local mid-points. Each strip then has a series of points which can be fitted to a regression line.

B. Hough Transforms

This is one of the most common methods, and uses a Hough transform (HT) to determine where lines are in the image [6]. This can be time-consuming especially for large images, so various optimisations have been done including using a Randomized Hough Transform [7].

C. Vanishing Point Skeleton

This method does a segmentation process to find the sections of crops in the image. The skeleton of the segments is then extracted, and then finds which skeletons are pointing towards a common vanishing point. It then applies a line to each of these skeletons.

D. Blob Analysis

Rather than doing segmentation, this simply looks for blobs of the correct colour and above a certain size. The principle axis of each blob is then said to be the crop row.

E. Stereo or 3D Cameras

Instead of using colour differentiation, the height difference between the crops and the ground can be used. This doesn't always work for seedlings, but the bird-scaring robot should not need to work at this stage of the plants life.

F. Frequency Analysis

Since it is straight lines that are being detected, the rows can be analysed in the frequency domain. Doing this requires more complex maths and is not as intuitive as some of the previous methods [8], which makes it harder to implement or improve on.

G. Trapezium Method

Since rows are always in pairs, the points at the end of each pair of rows can be connected and made into a 'flexible qudrangle', usually forming a trapezium [9]. This is very efficient, as no high-level algorithms (such as Hough transforms) are used, but is quite limited in where it can be applied.

III. Proposed Method

The methods mentioned previously are not mutually exclusive, and the proposed method will use a combination of these. The proposed algorithm was based on one developed by G. Jiang et al [10] (not to be confused with [11]) because it was had the most relevant application and the method was clearly laid out. Instead of having a relatively consistent time to run, the proposed algorithm aims to have a more consistent outcome. This is done by reducing the parameters that need to be optimized manually, so the remaining ones can be optimized iteratively. Because of this different approach, the algorithm varies significantly from G. Jiang's original work. The code used has been made publicly accessible at https://github.com/petern3/crop_row_detection [12].

A. Retrieving the Image

To test the algorithm over a range of data, the Crop-Row Benchmark Dataset (CRBD) was used [13]. This contains images with various types of crops, lighting and viewing angle, which makes it perfect for testing this method. Although the dataset also contains ground truth data, this was not used. The first of the 281 images in the set is seen in Figure 1.

Fig. 1: First image in the CRBD image set

B. Grayscale Transform

The first step in detecting the crop rows is to determine which pixels are plants. This can be done by subtracting the blue and red channels from twice the green channel [10], i.e.

$$grayscale = 2 \times green - red - blue$$

While other multipliers can be used, this seems to be sufficient. Figure 2 shows the result of this operation when applied to the first CRBD image, Figure 1.

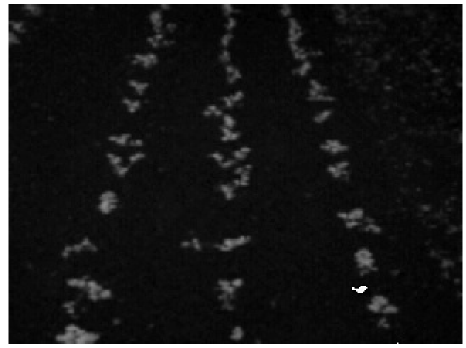

Fig. 2: Enhanced green values

C. Skeletonization

Since a HT will be used later, the image should be conditioned for better results. Finding the skeleton of the grayscale image significantly reduces the size of the image while keeping the important information. Figure 3 demonstrates how the skeleton image contains the crop location information. This step also reduces the number of lines detected by the HT, and makes the HT find lines on the crop rows rather than on it's edges.

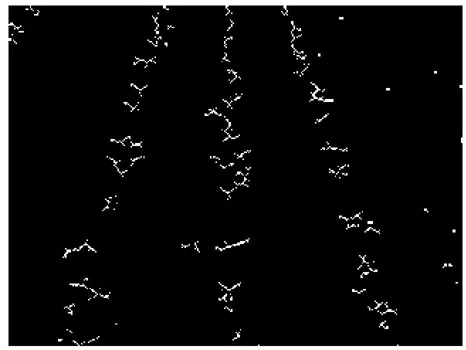

Fig. 3: Skeleton of crop areas

D. Hough Transform

The next part of the proposed method is the part that is iterated. A HT is performed on the skeleton, such that the 'Accumulator Threshold' (AT) is very large. This means that not many lines will have enough 'votes' above the AT, and there will not be many lines. The HT is then re-applied with a lower AT, which means more lines will be found. Figure 4 demonstrates the kind of lines the HT detects. It can be seen that there are many lines which are not crop rows, and can be filtered out.

Fig. 4: All rows detected by the HT

E. Filtering

There are two methods of filtering that are done to get just the crop rows, and are as follows:

1) Checking if the lines are vertical enough.
2) Finding and removing similar lines.

A simple threshold is done to see if the lines are within 30 degrees of vertical. The method also occasionally finds (faulty) vertical lines, so these are also removed, under the assumption that an agricultural robot would not be directly on top of the crops.

To remove similar lines, a value of 6 degrees for angle was used and 8 pixels for linear similarity was used. This left results more like if 5. When the filered number of lines exceeds a user-defined number of crop rows, the algorithm stops iterating. If the AT parameter goes too low, then it assumes there are no crop rows to be found.

Fig. 5: Final detected rows

IV. RESULTS

The method developed by G. Jiang et al [10] (Method 1) was compared since it was what the proposed method (Method 2) was based on. As expected, the proposed method had much more variation in time, but was more reliable. There were four aspects of each method that were tested, two based on timing and two on accuracy. The system used on for these tests was as follows:

- Processor: 2.5 Ghz Intel i5-2520M
- Memory: 2.0 Gb RAM
- Operating System: 64-bit Linux Mint 17
- Implementation: Python 2.7 with OpenCV libraries

A. Timing Comparison

The most time that any single image took to process was 92.3ms, which was using Method 2. Method 1 was very similar, with its maximum time being 91.1ms.

Method 2 had a significantly faster mean time, taking an average of 32.1ms while Method 1 took an average of 79.4ms. This suggests that Method two can process more than twice the images in the same time, or for agriculural robots, can have a higher framerate.

B. Accuracy Comparison

Because Method 1 had several parameters to optimise, when they were calibrated for one image, they didn't work particularly well on the rest. The first 20 images were viewed and manually scrutinized for error for each method. These 20 images did not include the calibrated image which was 100% accurate in both cases. Method 1 found on average 36.5% of the lines, and Method 2 detected 55.7%. It can be seen in Figure 5 that the right-hand line is not quite on the crop row - that is because the method of filtering similar lines always keeps the right-hand row instead of the centre one.

The number of false positives detected was similarly in favour of Method 2. The original method added an average of 6.4 lines each image, and the second had and average of 0.4 false positives per image (one extra line 40% of the time).

When Figures 7 and 8 are compared, these numbers are much more meaningful. Both methods performed fairly poorly on Image 1, not really detecting any rows very well. It is assumed that the reason for this is that the crops are more sparce close the camera, so there are several signicant gaps in the crop rows.

Images 2 and 3 of Figures 7 and 8 are possibly some of the clearest crop rows that would be seen in the real world, and is where Method 2 really stands out. Even though the algorithm was only searching for three crop line, it found additional correct ones. Method 1 was on the right track in 7c, but terrible in 7b - a good demonstration of its lack of reliability.

The final image in the same Figures was almost as bad as Image 1 for each, with Method 2 having a slightly better attempt at finding the rows. In this case it is understandable as even humans would struggle to find the rows in the lower part of the image.

C. CRBD Comparison

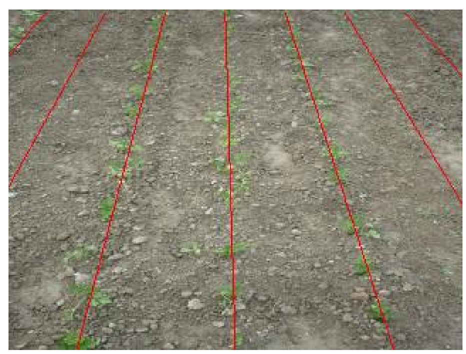

Fig. 6: CRBD ground truth data

Both Methods 1 and 2 detected three rows in the first CRBD image (Figure 1), but the ground truth data for this image in Figure 6 shows an additional four rows. Two of these rows cannot be seen, but make sense if the spacing is consistent, and the other two are only in the corners of the image. Because of the situation each of these lines are in, they have been marked as 100% accurate; it is what the author would have marked.

Many of the crops rows in the CRBD did not have straight crop rows, either because the crops themselves went round a corner, or because they bend up a hill. Fortunately due to perspective projection, the majority of the lines are straighter at the bottom of the image, which means the lines are closer to the actual rows nearer the robot.

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Fig. 7: A sample of detected rows using Method 1

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Fig. 8: A sample of detected rows using Method 2

V. Conclusion

The proposed method of detecting crop rows for agricultural robots works in a range of environments at a high frame-rate making it ideal for real-time systems. Compared to an existing algorithm developed in 2010, Method 2 has many advantages:

- The algorithm has fewer steps, so it is easier to implement and to understand
- It is nearly 2.5 times faster, on average, than Method 1
- The algorithm is 1.5 times as effective at finding crop rows in a variety of images
- As it tries to find a set number of lines, it also has a more consistent output
- It detects approximately 16 times fewer false positives

Once lines have been calculated, they can be converted to the robot's local co-ordinate systems as long as the cameras position relative to the robot is known. With this information, a robot could navigate between crops during most stages of the plants life with better precision than other localisation techniques such as GPS units and IMUs.

VI. Future Work

Although the proposed algorithm is faster and more reliable than Method 1, there is much that could be done to improve Method 2. It is not currently good enough to detect all relevant crops (as of September 17, 2017), but several important steps have been determined in order of the priority of application.

A. Finding a Variable Number of Rows

At the moment the proposed algorithms seems to be the most effecting at finding precisely three crop rows. Firstly, it should be modified such that any number of rows works roughly evenly, and then some criteria made that means that the number of rows to find does not have to be manually set.

B. Iterative Optimization of Skeleton Threshold

One of the first steps of the skeletonization process is to apply a threshold. In order to get the most effective skeleton, the threshold could be repeated until some criteria achieved that gives the best output for the Hough transform. This would give better outputs for both more sparse crops (such as Figure 8d) and more dense crops.

C. Filter for a 'Vanishing Point'

Instead of removing faulty rows based purely on angle, they can be removed if it doesn't go to a similar vanishing point to the rest of the rows. Ideally it wouldn't just be a plain threshold if the top point of the line is in the centre. The vanishing point may be to one side if the camera is not looking down the rows or if the rows are curved.

D. Any Further Required Filtering

After the vanishing point filter has been applied, there may be some filtering that still needs to be done, but cannot be anticipated. One example of this may be to place a crop row in the middle of a group of them instead of only taking one (which is the current method).

E. Testing of Other Colour Spaces

Fig. 9: A crop with no visible crop rows [14]

The first step in this and many other algorithms is to find the green values in the image in one way or another. This works for many of the images in the CRBD, but there are three cases that this method has been untested and may not function:

1) The crop is not green
2) The plants are too small (see Figure 7d)
3) The plants are too large and there are no gaps between rows such as the wheat field in Figure 9

In these cases, it could be worth investigating other colour spaces for whichever consistently has the most contrast between where the crop rows are and aren't. Such colour spaces that could be compared are YUV and HSV [15], with the RGB that the proposed method uses.

If none of these methods are able to detect rows when there are no gaps, it would also be worth investigating if the robot is able to drive between them at all without damaging the crops. It could be that no detecting rows is a beneficial feature.

F. Detection of Curved Crop Rows

Many crop rows are not linear due to circular irrigators or various landscapes, as mentioned in the results section. In these cases, a simple set of lines may not be enough. There are two ways one could go about this: Either use a Hough transform to find circles treating rows as circles of various radii, or aligning the lines to the bottom part of the crop rows, as that is the important part for a mobile robot.

REFERENCES

[1] Foundation for Arable Research. "Pest Birds Result in $30 Million Loss to New Zealand Arable Growers." Internet: https://www.far.org.nz/assets/files/uploads/080303_Falcons.pdf, Mar. 3, 2008 [Mar. 27, 2016].

[2] G. Ruiz-Ruiz, L. M. Navaz-Gracia, J. Gómez-Gil. "Camera kinetic model applied to crop row tracking and weed detection." Dept. of Agricultural and Forest Engineering, University of Valladolid, Spain, before 2006.

[3] V. Fontaine, T. G. Crowe. "Development of line-detection algorithms for local positioning in densely seeded crops." Dept. of Agricultural and Bioresource Engineering, University of Saskatchewan, Canade, 2006.

[4] J. Romeo, G. Pajares, M. Montalvo, J. M. Guerrero, M. Guijarro and A. Ribeiro. "Crop Row Detection in Maize Fields Inspired on Human Visual Perception." *The Scientific World Journal*, 2012.

[5] L. Zheng and J. Xu. "Multi-Crop-Row Detection Based on Strip Analysis." *2014 International Conference on Machine Learning and Cybernetics (ICMLC)*, Lanzhou, 2014.

[6] R. Fisher, S. Perkins, A. Walker and E. Wolfart. "Hough Transform." Internet: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm, 2003 [May. 6, 2016].

[7] R. Ji and L. Qi. "Crop-row detection algorithm based on Random Hough Transformation." Elsevier, vol. 54, no. 3-4, p. 10161020, August 2011.

[8] V. Kumar, M. Nanda. "Image Processing in Frequency Domain Using Matlab: A Study for Beginners" Internet: https://hal.archives-ouvertes.fr/file/index/docid/321613/filename/IMAGE_PROCESSING_IN_FREQUENCY_DOMAIN_USING_MATLAB_A_STUDY_FOR_BEGINNERS.pdf, 2008 [May 12, 2016].

[9] C. Tu, B. J. van Wyk, K. Djouani, Y. Hamam and S. Du. "An efficient crop row detection method for agriculture robots." *Image and Signal Processing (CISP)*, Dalian, 2014.

[10] G. Jiang, C. Zhao and Y. Si. "A Machine Vision Based Crop Rows Detection For Agricultural Robots." *2010 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Qingdao, 2010.

[11] G. Jiang, C. Zhao. "Baseline Detection and Matching to Vision-Based Navigation of Agricultural Robot." *2010 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Qingdao, 2010.

[12] P. Nicholls. "crop_row_detection." Internet: https://github.com/petern3/crop_row_detection, May. 6, 2016 [May. 6, 2016].

[13] Robotics and 3D vision Group. "Crop Row Benchmark Dataset." Internet: http://www.etfos.unios.hr/r3dvgroup/index.php?id=crd_dataset, Nov. 13, 2015 [Apr. 5, 2016].

[14] Myrabella. "File:Champ de blé Côte-d'Or Bourgogne avril 2014.jpg." Internet: https://commons.wikimedia.org/wiki/File:Champ_de_bl%C3%A9_C%C3%B4te-d%27Or_Bourgogne_avril_2014.jpg, Apr. 22, 2014 [May 7, 2016].

[15] A. Ford, A. Roberts. "Colour Space Conversions" Internet: http://www.poynton.com/PDFs/coloureq.pdf, Aug. 11, 1998 [May 12, 2016].