

CLUSTERING ALGORITHM BASED STRAIGHT AND CURVED CROP ROW DETECTION USING COLOR BASED SEGMENTATION

^aNazmuzzaman Khan, ^cVeera P. Rajendran, ^bMohammad Al Hasan, ^aSohel Anwar*

^aMechanical and Energy Engineering Department, Indiana University–Purdue University, Indianapolis

^bDepartment of Computer Science, Indiana University–Purdue University, Indianapolis

^cEquipment Technologies, Indianapolis, Indiana

Contact Email: soanwar@iu.edu

ABSTRACT

Autonomous navigation of agricultural robot is an essential task in precision agriculture, and success of this task critically depends on accurate detection of crop rows using computer vision methodologies. This is a challenging task due to substantial natural variations in crop row images due to various factors, including, missing crops in parts of a row, high and irregular weed growth between rows, different crop growth stages, different inter-crop spacing, variation in weather condition, and lighting. The processing time of the detection algorithm also needs to be small so that the desired number of image frames from continuous video can be processed in real-time. To cope with all the above mentioned requirements, we propose a crop row detection algorithm consisting of the following three linked stages: (1) color based segmentation for differentiating crop and weed from background, (2) differentiating crop and weed pixels using clustering algorithm and (3) robust line fitting to detect crop rows. We test the proposed algorithm over a wide variety of scenarios and compare its performance against four different types of existing strategies for crop row detection. Experimental results show that the proposed algorithm perform better than the competing algorithms with reasonable accuracy. We also perform additional experiment to test the robustness of the proposed algorithm over different values of the tuning parameters and over different clustering methods, such as, KMeans, MeanShift, Agglomerative, and HDBSCAN.

INTRODUCTION

Autonomous navigation in agriculture has several advantages, like, reduce operator fatigue, improve profit and efficiency, and enhance operation safety. To support autonomous navigation, a crucial task is to develop the capability of computer vision based detection of crop rows from image/video data. To address this, several methodologies have already been proposed [1–4]; however, our analysis on these existing methodologies reveal that their detection quality deteriorates significantly, when the crop row image is complex. Specifically, the complexity arises due to two reasons: first, missing of crops in some segment of the crop row causing discontinuity in a crop row, and the second, significant weed growth in the area between the crop rows. When an image exhibits both the above reasons simultaneously, distinguishing crop rows from the weed clusters becomes very difficult, and mostly all the existing methods fail to achieve a satisfactory performance in solving the crop row detection task.

The desiderata of a practical crop row detection method that works satisfactorily in real-life deployment are below: (1) It is capable of detecting crop rows even with high weed pressure; (2) It is applicable at different types of crop fields; (3) It is capable to detect crop rows at different crop growth stages; (4) It is capable to detect straight and reasonably curved crop rows; and finally, (5) The processing time of detection on an off-the-shelf computer satisfy real-time requirements. In this work, we proposed a simple, yet robust crop row detection method which satisfy all the above requirements. A short summary of proposed crop row detection method is provided below.

*Address all correspondence to this author.

The basic idea of our proposed method is to use a robust clustering method so that the green weed pixels are detected as noise points, whereas the green crop row pixels are detected as data clusters. We want each crop row to be detected as one distinct cluster so that we can fit a regression line through that cluster which we can return as detected crop row. At the first step, the input image is cropped, segmented and cleared from small noise or weed segments. At this point the image has two channels, black and white pixels. The black pixels contain background and the white pixels contain crop and weed. Now the goal is to separate weed pixels from crop pixels, so that lines can be fitted to the crop rows. A clustering algorithm is used which can cluster each crop row pixels and separate weed pixels from the clusters. If the crop row clusters contain weed pixels, they can hamper the accuracy of line fitting algorithm. As a result, a robust line fitting algorithm is used to fit a line on each crop row cluster which omits the effects of outlier points (weed or noise). Finally the fitted lines are plotted on the image and returned as detected crop rows. Simple flowchart of the proposed algorithm is presented in Fig. 1. Expanded steps of the proposed algorithm is visually presented in Fig. 2a and Fig 2b.

RELATED WORKS

Basso and de Freitas [1] used a filtered Hough transform [2] method and achieved around 30 frame per second(FPS) for 320x240 image resolution on RPi-3B embedded system. Although they showed detailed results about how the algorithm performed for different image sizes and frame rates (max speed used for good detection: 2 m/s), no detailed calculation is showed about the effect of weed pressure and crop missing on row detection accuracy.

Zhang et al. [3] proposed a double thresholding (combining OTSU method with particle swarm optimization) on segmented image with linear regression for line fitting. They said, due to double thresholding their method can separate weed from crop rows. But how this method performs under high weed pressure is not shown in detail. Also, least squares fitting is highly affected by presence of weed in image.

Sainz-Costa et al. [4] developed a strategy based on analysis of video sequences to detect crop rows. They used the lower half part of the image to detect. The image processing experienced five steps: segmentation, morphological opening, horizontal strips dividing, vertical average, row centers extracting and crop rows finding. The above approach can work well under low weed pressure.

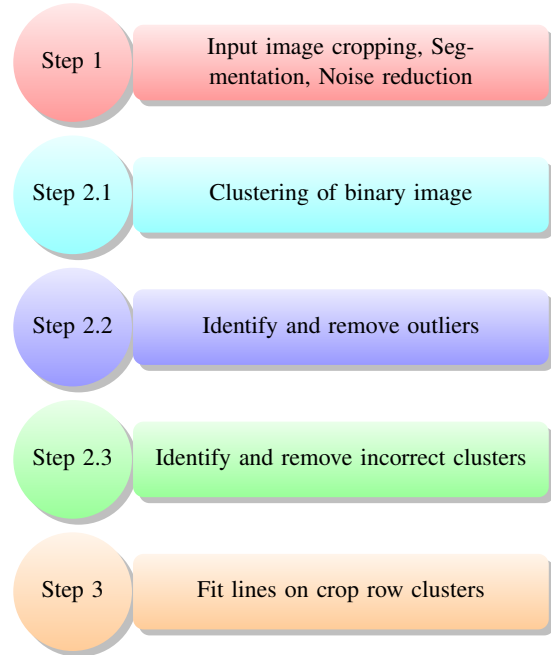


FIGURE 1: Flowchart of proposed algorithm.

Vidovic et al. [5] applied the vanishing point principle to determine that parallel crop rows (straight and curved) are imaged preserving the inter-crop row distances (regular patterns), which allows to apply a matching technique to combine image evidence and prior knowledge about the geometric structure. Optimization, based on dynamic programming, was applied for straight and curved crop row detection for different row crop. But for an image with resolution of 640x480 need around 5 s of processing time.

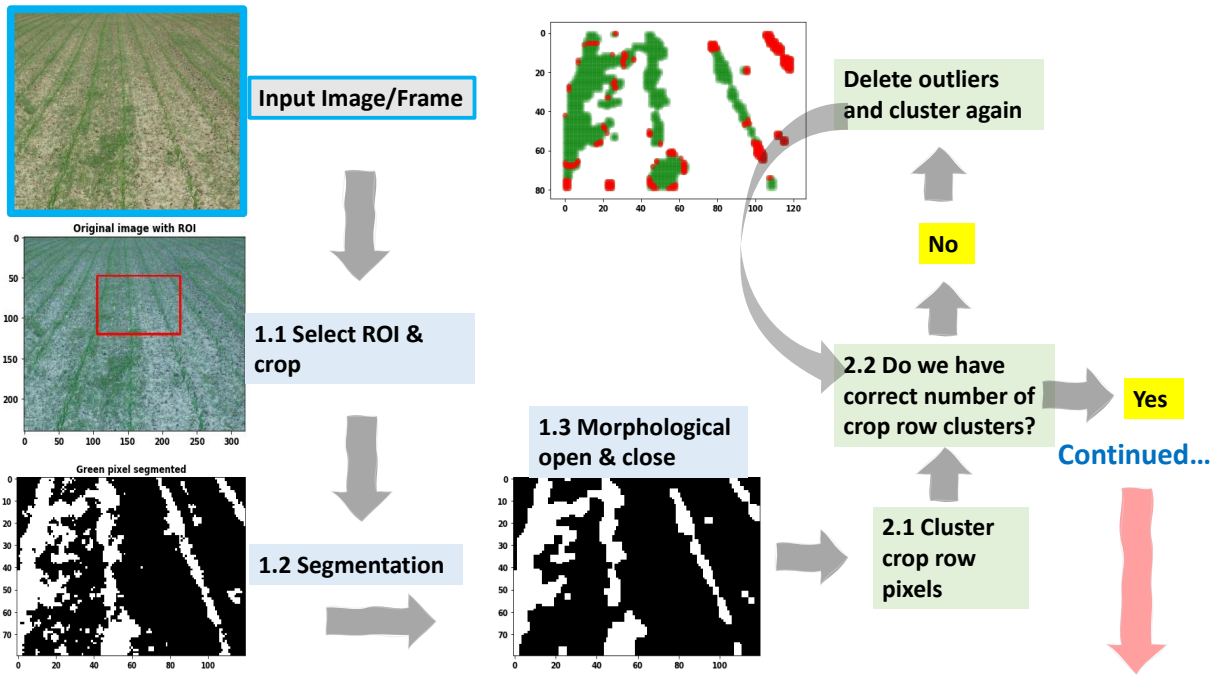
METHODOLOGY

The step-by-step discussion of our proposed crop row detection method in below.

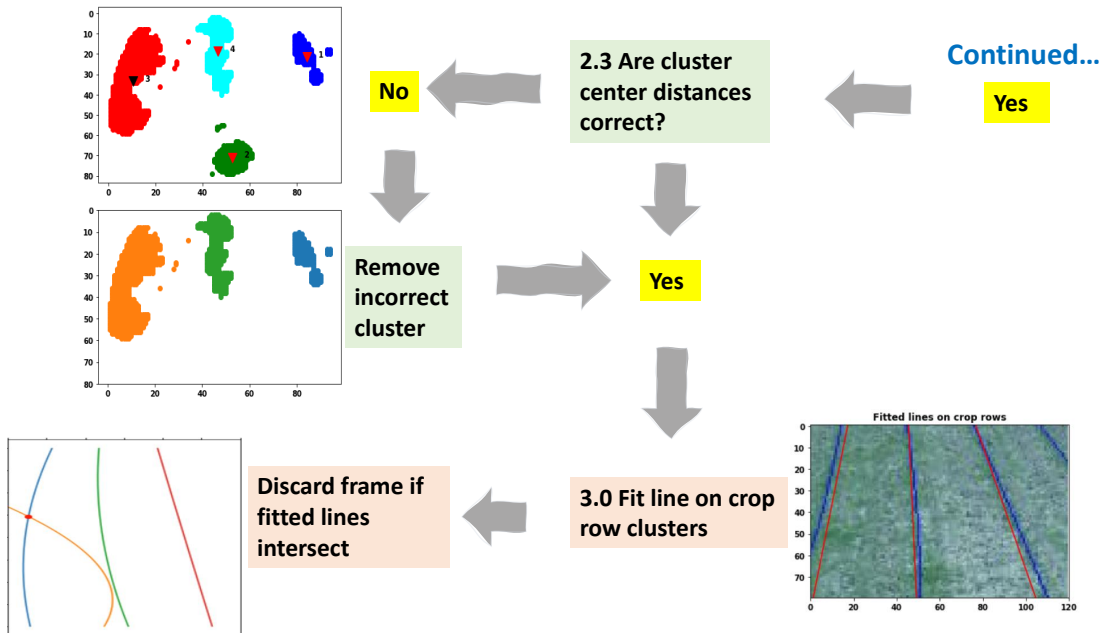
Step 1: Pre-processing (Step 1.1) First a region of interest (ROI) is selected which contains 3 or more crop rows. All the algorithmic procedures are applied within this ROI. Selecting this ROI has three advantages. First, green pixels near horizon are congested and hard to separate. As a result, they increase false detection. Selecting this ROI eliminates the need to process those green pixels. Secondly, this ROI is almost one-fourth the size of the original image. So only operating in this section reduces the computational cost and processing time. And third, curved rows within this small ROI are reasonably straight. As a result, fitting straight lines to these curved rows provide reasonable result.

(Step 1.2) Given an input image in RGB color space, each channel is separated by following system:

R = red channel of input image



(a) Algorithm steps (Step 1 - Step 2.2)



(b) Algorithm steps (Step 2.3 - Step 3)

FIGURE 2: Steps of the proposed crop row detection algorithm with images.

G = green channel of input image
 B = blue channel of input image

After splitting the channels, the following normalization scheme is applied, which is common in agronomic image segmentation [6]:

$$\begin{aligned} r &= \frac{R_n}{R_n + G_n + B_n} \\ g &= \frac{G_n}{R_n + G_n + B_n} \\ b &= \frac{B_n}{R_n + G_n + B_n} \end{aligned} \quad (1)$$

where R, G and B are the normalized RGB coordinates ranging from 0 to 1 and are obtained as follows:

$$R_n = R/R_{max}, G_n = G/G_{max}, B_n = B/B_{max} \quad (2)$$

where R_{max} , G_{max} and B_{max} are maximum values of R, G and B channels respectively. A small number is added to the denominator of normalization step to avoid division by zero. Green color (vegetation) can be extracted using the following equations [7]:

$$ExG = 2g - r - b \quad (3)$$

(Step 1.3) Morphological opening and closing operation on the binary image to reduce noise.

Step 2: Cluster An ideal clustering algorithm for crop row detection should have two specific features. First, the clustering algorithm should be able to differentiate between different crop row clusters without any prior knowledge (we don't know exactly how many crop rows may appear inside ROI). Second, should have robust and intuitive tuning parameters to tune the algorithm for wide variety of scenarios.

Four different types of clustering algorithms are tested and a sample output is presented in Fig. 3. Kmeans [8] fails because crop rows are anisotropic data which means they are elongated along a specific axis. Because Kmeans treats every data point equally, it fails to distinguish any local variation within a cluster. The advantage of Meanshift [9] over Kmeans is that we don't have to specify the number of clusters. Meanshift algorithm assumes an underlying probability density function of the data and locates centroids at the maxima of the density function. The default parameter *bandwidth* which dictates the size of the region to search through in scikit-learn [10] shows wrong results. Also it is order of magnitudes slower than other tested methods and not applicable for real-time usage. Agglomerative clustering [10] is a hierarchical method which group data into clusters based on similarity. It starts with treating each data point as a single cluster then merges clusters into a single cluster until some criterion is met. But for this to work, 'number of clusters' has to be known

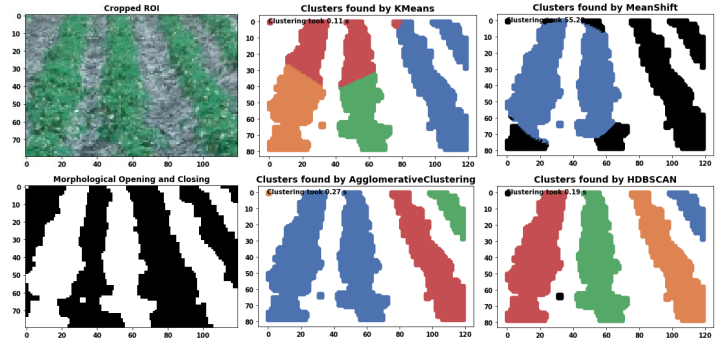


FIGURE 3: Comparison of Kmeans, MeanShift, Agglomerative and HDBSCAN clustering algorithm on crop row detection. Run-time indicated on the top left corner of each image. Different colors indicate different clusters. Black color indicate outliers.

before hand. HDBSCAN [11] is a density based method which extends DBSCAN [12] into a hierarchical clustering algorithm. The only tuning parameter used is *min cluster size* and its effect is explained later. From Fig. 3 it is clear that HDBSCAN successfully clustered the crop rows. It also identifies the outliers (black pixels) which is then omitted from output clusters.

Preventive measures to reduce wrong clustering is implemented into the algorithm. If number of cluster inside ROI is lower than three (step 2.2), the algorithm assumes some of the clusters are merged due to high amount of weed. It then iteratively deletes outliers (weeds) to find crop row clusters. Sometimes due to high weed growth or intermittent crop growth, HDBSCAN creates multiple clusters for a single crop row. This can be controlled by changing the *min cluster size* parameter. But for fixed *min cluster size* parameter, *crop row distance* is used as a threshold parameter (step 2.3). If multiple clusters are close by (determined by the distance of crop rows), then the cluster with smaller data points and smaller height will be deleted. Which is showed in Step 2.3 of Fig. 2b.

Step 3: Line fitting Random Sample Consensus (RANSAC) [13] is used for line fitting on each cluster due to simple implementation and robustness. Because RANSAC iteratively determines the best data points to fit a line in a cluster, it excludes the outliers (possibly weed data points) and shows robust line fitting when weed is present. The algorithm for line fitting on clusters is presented in Algorithm 1.

The full algorithm for crop row detection is presented in Algorithm 2. Some important tuning parameters of the algorithm and their effects on crop row detection is discussed in Table 1.

Algorithm 1 RANSAC

- 1: Randomly select two points from cluster to determine the straight line model.
 - 2: Solve for the parameters of the model.
 - 3: Determine how many points from the cluster fit with a pre-defined tolerance.
 - 4: If the fraction of the number of inliers over the total number points in the cluster exceeds a predefined threshold (probability = 0.99), re-estimate the model parameters using all the identified inliers and terminate.
 - 5: Otherwise, repeat steps 1 through 4 (maximum of $N = 100$ times).
-

METHODS USED FOR COMPARISON

Four different methods along with the proposed method is compared in this study. The first one, Hough transform [2] is a feature extraction technique in digital image processing. After step 1, Canny edge detection [14] is applied to extract the edges from the binary image. With the coordinate transformation, the collinear points in edges of the binary image converted to concurrent lines in parameter space by voting. Hough transform detects lines by accumulating the votes.

The second method is named Sliding-window. After step 1, a window of size (20 by 20) slides over the ROI and calculates the center points of the white pixel blobs inside that window. After sliding over the whole ROI, we essentially have the center points of the crop rows. Then least-square straight line is fitted on the crop row center points. Some variant of this method is available is literature [3, 15].

The third method is Template Matching followed by Global Energy Minimization (TMGEM) [5]. It uses dynamic programming for efficient global energy minimization. This method can work without any prior knowledge of crop row number, reasonably insensitive to weed and works with different crop growth stages. The authors of this paper also created Crop row benchmark dataset (CRBD). This evaluation image set includes 281 images of maize, celery, potato, onion, sunflower and soya bean crops. The images are taken at varying yaw, pitch and roll angles; different amount of weed pressure and lighting conditions. This dataset is used in this study for comparison and testing.

The fourth method is named Cluster - Least square. Here after step 2, least square straight line fitting is used. The final method is named Cluster - RANSAC and is the proposed method. Other than TMGEM, all the other methods are built and implemented from scratch by us.

RESULTS

Forty different scenarios from the CRBD dataset is used to test the effectiveness of the proposed algorithm. Under normal

Algorithm 2 Crop Row Detection

- 1: *image* = input color image
 - 2: *ROI size* = variable
 - 3: *ROI image* = center crop *image* of *ROI size*
 - 4: *segmented image* = binary segmentation of *ROI image* using *ExG* and *OTSU* threshold
 - 5: *clean image* = morphological *OPEN* and *CLOSE* operation on *segmented image*
 - 6: *clusters* = cluster *clean image* with *HDBSCAN* (*min cluster size* = variable)
 - 7: *minimum cluster number* = 3
 - 8: *max iteration* = 3
 - 9: *while* number of cluster < *minimum cluster number* **do**
 - 10: **if** *iteration* > *max iteration* **then**
 - 11: *break*
 - 12: **end if**
 - 13: *outliers* = *data points* over 85% outlier score
 - 14: *data points* = *data points* - *outliers*
 - 15: recalculate number of *clusters* on *data points*
 - 16: *iteration* += 1
 - 17: **end while**
 - 18: {*/*check for: one crop row is unusually big due to connected to weed or unusual shape connects to other crop row*/*}
 - 19: *cluster data point threshold* = 1000
 - 20: **if** *data points* in a *cluster* > *cluster data points threshold* **then**
 - 21: *clusters* = cluster using only *good points*
 - 22: **end if**
 - 23: {*/*check for: if multiple clusters are close by, keep only one based on height. Because crop row clusters are vertically oriented and weed clusters are round shaped*/*}
 - 24: *crop row distance* = variable
 - 25: **for** *i* in range(0, number of clusters - 1) **do**
 - 26: **for** *j* in range(*i*+1, number of clusters) **do**
 - 27: *cluster distance* = |*cluster center*[*i*] - *cluster center*[*j*]|
 - 28: **if** *cluster distance* < *crop row distance* **then**
 - 29: **if** *cluster*[*i*] height > *cluster*[*j*] height **then**
 - 30: keep *cluster*[*i*]
 - 31: delete *cluster*[*j*]
 - 32: **else**
 - 33: keep *cluster*[*j*]
 - 34: delete *cluster*[*i*]
 - 35: **end if**
 - 36: **end if**
 - 37: **end for**
 - 38: fit straight line for each cluster with *RANSAC*
 - 39: plot straight lines over *ROI image* and show
-

TABLE 1: TUNING PARAMETERS FOR ALGORITHM 2.

Parameter	Value	Description
ROI size	(120 by 80), (160 by 80)	This algorithm works better if there are 3-4 crop rows inside the ROI. Based on camera position and camera viewing angle, the ROI size should be fixed.
min cluster size	50, 100, 200	HDBSCAN parameter to affect clustering. Set it to the smallest size grouping that you wish to consider a cluster. For big number (200): at early crop growth stage, combine multiple rows into one cluster. For small number (50): at early intermittent crop growth stage, create multiple clusters for one crop row. As crop grows, shows less effect on clustering. <i>minclustersize</i> = 50 shows reasonable results on CRBD dataset for this study. Should be tuned based on growth stage, ROI size, crop type.
crop row distance	15, 25, 30	This is the pixel value of center distance between two consecutive crop rows. Actual distance converted to pixel. This value should be calculated based on actual crop row distance on field and camera parameters.

condition (crop rows are reasonably separate, none or low weed pressure, crop rows are straight or reasonably curved) the proposed algorithm detects crop rows with 100% accuracy. Fig. 4 qualitatively compares the results of five different methods under challenging scenarios.

In Fig. 4 (Row 1), very early stage of crop growth and crop rows are hard to see with naked eye. Hough transform and Sliding window method fails to correctly detect the third row in the image. TMGEM and the proposed method performs the best. In (Row 2), due to high weed pressure and early crop growth stage, crop row 2 and 3 are connected. This is a hard problem to solve because weeds are at high concentration and any line fitting algorithm fits line through weed pixels. This is what happened for Hough transform and Sliding-window. Our proposed algorithm shows the best results in this scenario. In (Row 3), weeds with unusually big size appears. Any usual row detection algorithm will fail and fit lines over weed pixels due to unusually high concentration. This is what happened for all the

other methods. Our proposed method overcomes this because RANSAC iteratively finds 2 points with 99% probability to be inliers of a cluster and fit lines over those two points. In (Row 4), top half of crop row is missing for the first row. Also at the top left corner there are some green pixels from another crop row. Cluster - least square fails because least square is not robust against outliers. Our proposed method overcomes this problem. In (Row 5) our proposed algorithm shows suboptimal results for the second crop row. This happened because crop row growth is substantially less compared to weed growth. As a result, clustering algorithm deletes the crop row part as outlier. This can be mitigated by tuning the *min cluster size* parameter in HDBSCAN (use a smaller number).

For a real-time algorithm, processing time is important. The average time spent by each step of the algorithm is presented in Fig. 5. It is tested on a personal computer without any parallel-processing or multi-threading techniques (which may reduce the processing time significantly). Also processing time for image acquisition hardware is not included. The algorithm takes 108 ms (around 10 frames per second (FPS)) to process each image. For a slow moving agricultural vehicle, 10 FPS is a reasonable processing time real-time application. More than 90% of the time is spent by the clustering step. Step 2.2 (iteration) only happens for challenging scenarios (high weed pressure or crop rows connected). For normal crop growth and low weed pressure scenario, the algorithm bypasses step 2.2. With the development very powerful embedded hardware like Jetson TX2 which has built-in video processing capabilities, the proposed algorithm is capable of real-time performance even for high speed vehicles.

Comparison of processing time is presented in Table 2. Again it should be mentioned that no parallel-processing or multi-threading techniques are applied which may reduce the processing time significantly. Clustering methods take order of magnitude higher time compared to Hough transform and Sliding window. TMGEM takes order of magnitude higher time than clustering methods which makes it unavailable for real-time crop row detection application.

CONCLUSIONS AND FUTURE WORK

In this research, a new clustering based crop row detection algorithm is presented and tested on CRBD dataset. The proposed algorithm uses clustering technique and prior knowledge of geometric structure of crop rows to differentiate weeds from crop rows. With the use of a smaller ROI processing time is curtailed and straight lines can be accurately fitted to reasonably curved crop rows. The algorithm also uses RANSAC (a robust line fitting technique) to further mitigate the effects of weeds on crop row detection. The proposed algorithm is applicable for real-time usage and shows good crop row detection accuracy at very challenging scenarios. The authors are currently working on a method to quantitatively compare the five different algorithm

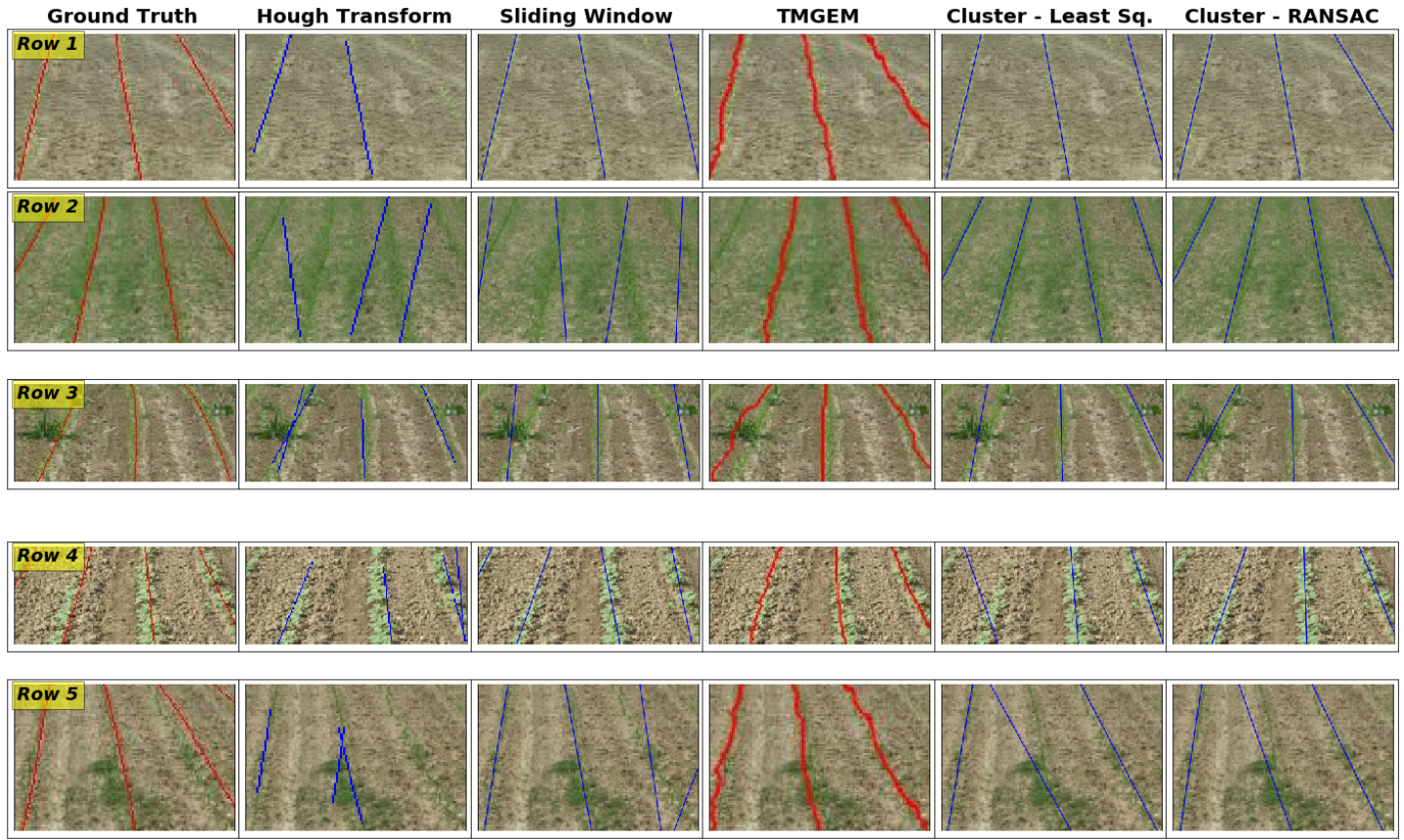


FIGURE 4: Comparison of crop row detection in challenging scenarios for Hough transform, Sliding window, TMGEM, Cluster-Least square and Cluster-RANSAC (proposed) methods with ground truth results.(Row 1) intermittent, very early crop growth, no weed. (Row 2) early crop growth stage, high weed pressure, crop rows are connected, crop rows and weed indistinguishable. (Row 3) early crop growth stage, exceptionally big intermittent weed, curved crop rows. (Row 4) medium crop growth, crop missing from rows, no weed. (Row 5) early crop growth stage - intermittent, concentrated weed growth.

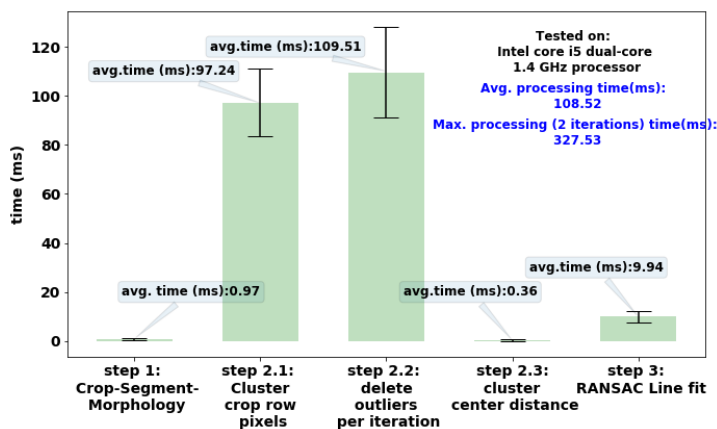


FIGURE 5: Processing time of each step of the proposed crop row detection algorithm. ROI size (120 by 80) pixels.

on CRBD dataset. The final goal of this work is to implement this algorithm on an off the shelf embedded system (Jetson TX2) for real-time detection of crop rows from video input.

REFERENCES

- [1] Basso, M., and de Freitas, E. P., 2019. "A uav guidance system using crop row detection and line follower algorithms". *Journal of Intelligent & Robotic Systems*, pp. 1–17.
- [2] Hough, P. V., 1962. Method and means for recognizing complex patterns, Dec. 18. US Patent 3,069,654.
- [3] Zhang, X., Li, X., Zhang, B., Zhou, J., Tian, G., Xiong, Y., and Gu, B., 2018. "Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method". *Computers and electronics in agriculture*, **154**, pp. 165–175.
- [4] Sainz-Costa, N., Ribeiro, A., Burgos-Artizzu, X. P., Gui-

TABLE 2: PROCESSING TIME OF DIFFERENT ALGORITHMS.

Algorithm	Processing time (ms)	Avg. FPS
Hough transform*	2.02	495
Sliding window*	2.88	347
Cluster - least sq.*	100.12	9.9
Proposed*	108.52	9.2
TMGEM**	1750	0.5

* core i5 dual-core 1.4GHz processor. (120 by 80) ROI size.

** core i5 quad-core 3.3GHz processor. (320 by 240) ROI size.

- jarro, M., and Pajares, G., 2011. "Mapping wide row crops with video sequences acquired from a tractor moving at treatment speed". *Sensors*, **11**(7), pp. 7095–7109.
- [5] Vidović, I., Cupec, R., and Hocenski, Ž., 2016. "Crop row detection by global energy minimization". *Pattern Recognition*, **55**, pp. 68–86.
- [6] Gée, C., Bossu, J., Jones, G., and Truchetet, F., 2008. "Crop/weed discrimination in perspective agronomic images". *Computers and Electronics in Agriculture*, **60**(1), pp. 49–59.
- [7] Ribeiro, A., Fernández-Quintanilla, C., Barroso, J., García-Alegre, M., Stafford, J., et al., 2005. "Development of an image analysis system for estimation of weed pressure". *Precision agriculture*, **5**, pp. 169–174.
- [8] Arthur, D., and Vassilvitskii, S., 2006. k-means++: The advantages of careful seeding. Tech. rep., Stanford.
- [9] Comaniciu, D., and Meer, P., 2002. "Mean shift: A robust approach toward feature space analysis". *IEEE Transactions on pattern analysis and machine intelligence*, **24**(5), pp. 603–619.
- [10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*, **12**, pp. 2825–2830.
- [11] McInnes, L., Healy, J., and Astels, S., 2017. "hdbscan: Hierarchical density based clustering". *The Journal of Open Source Software*, **2**(11), mar.
- [12] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., 1996. "A density-based algorithm for discovering clusters in large spatial databases with noise." In *Kdd*, Vol. 96, pp. 226–231.
- [13] Fischler, M. A., and Bolles, R. C., 1981. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communi-*

cations of the ACM, **24**(6), pp. 381–395.

- [14] Canny, J., 1986. "A computational approach to edge detection". *IEEE Transactions on pattern analysis and machine intelligence*(6), pp. 679–698.
- [15] García-Santillán, I. D., Montalvo, M., Guerrero, J. M., and Pajares, G., 2017. "Automatic detection of curved and straight crop rows from images in maize fields". *Biosystems Engineering*, **156**, pp. 61–79.