VISUAL ROW DETECTION USING PIXEL-BASED ALGORITHM AND STEREO CAMERA FOR COTTON-PICKING ROBOT

K. G. Fue
College of Engineering and Entomology, University of Georgia
Tifton, Georgia
W. M. Porter
Crop and Soil Sciences, University of Georgia
Tifton, Georgia
E. M. Barnes
Cotton Incorporated
Cary, North Carolina
G. C. Rains
Entomology, University of Georgia
Tifton, Georgia

Abstract

Precision farming still depends heavily on RTK-GPS to navigate the rows of crops. However, GPS cannot be the only method to navigate the farm for robots to work as a "swarm" on the same farm; they also require visual systems to navigate and avoid collisions. Also, plant growth and canopy changes are not accommodated. Hence, the visual system remains a complementary method to add to the efficiency of the GPS system. In this study, optical detection of cotton rows is investigated and demonstrated. A stereo camera is used to detect the row depth, and then, a pixel-based algorithm is used to calculate and determine the upper and lower part of the canopy of the cotton rows by assuming the normal distribution of the high and low pixels. The left and right row are detected by using perspective transform and pixel-based sliding window algorithms. Then, the system determines the Bayesian score of the detection and calculates the center of the rows for smooth navigation of the cotton-picking robot. The 92.3% accuracy and F1 score of 0.951 are sufficient to deploy the algorithm for robotic operations. The deployment and testing of the robot navigation will be done in 2019.

Introduction

Introduction of small intelligent machines in farming will be an asset to farmers. Swarms can be scalable for the size of the farm; each machine can be low-cost, multi-functional, and re-programmable for the task at hand. Smaller machines can also reduce the risk of serious injury and fatality from large field equipment (Rains et al., 2015). With modular attachments and selectable programming, these small intelligent machines can be used for multiple tasks such as precision weeding, precision chemical application, precision planting, precision harvesting, and precision scouting (Rains et al., 2014). In this study, we are focusing on intelligent machine navigation in cotton plants to investigate the possibility of robotic cotton harvesting.

Cotton harvesting has been heavily dependent on big machines with human operators. These big machines are costly to maintain and expensive to own. The emergence of modern technologies in robotics provides an opportunity to explore alternative harvesting methods (Hayes (2017) and Fue et al., 2018). Cotton plants are clearly separated in rows when they are young, and the canopy overlaps around 8-10 weeks after planting. For this research, the machine will be deployed in non-defoliated plants as soon as cotton bolls begin to open. Careful navigation in such a situation is required so that the bolls are not knocked off and also easily located and tracked (Fue, 2018). Self-navigation may introduce another problem if not well controlled. For human-driven tractors, RTK GPS is very accurate and can be used to keep following the same course with 1-2 centimeter accuracy. However, for self-navigation, the consciousness of the human being guidance will be missed, if only the GPS will be used. So it is important to return that visual perception to give the rover the alternative view of the environment.

However, the use of RTK-GPS to enhance navigation is challenged by signal loss while in operation due to attenuation around buildings, tree cover and other obstructions (Higuti et al., 2018). It requires the farmers to use a very good and expensive RTK-GPS system. Also, deployment of a swarm of robots that work together may require preprogramming to make sure obstacles and machines avoid a collision. So, it is imperative that the use of camera and simple RTK-GPS should be enough to achieve safe real-time navigation for farm vehicles traveling over the plants. There are other sensors like LIDAR that can be used in this operation, but as we expect the machine to work in daylight and over the plants, then RGB cameras may be enough. Lidar showed good success when small robotics navigate in between large plants (Higuti et al., 2018).

The main objective of this study was to develop a model to detect the rows in a cotton field and test to see the performance of the model. Therefore, a model that uses a stereo camera to guide a cotton-picking robot in row detection is proposed. The camera used to locate the bolls is the same camera used to guide the robot. However, a separate camera may be used to operate other tasks like Simultaneous Localization and Mapping (SLAM) of the robot in the field.

Materials and Methods

Materials

In this research, the red research custom-built articulated rover (West Texas Lee Corp.,) with modifications to meet the field conditions, navigation, and obstacle avoidance requirements of an unstructured (such as open field, end of row) and a structured field like row crop fields (Rains et al., 2015) is used to navigate through cotton rows. This rover is mounted with the stereo camera (Stereo labs Inc, San Francisco, CA, USA) and the rugged development kit, NVIDIA Jetson TX2 (NVIDIA Jetson TX2 development kit, Nvidia Corp., Santa Clara, CA, USA) which has the following features; NVIDIA Pascal 256 CUDA cores, Quad ARM and HMP Dual Denver CPU, 8GB 128-bit LPDDR4 RAM, and 32GB eMMC SATA drive. The ZED camera was mounted at 81.9° below the horizontal and took images and depth maps at the rate of 30 frames per second at the resolution of 1080p while the rover was moving at less than 3 miles per hour. The development kit was installed with Ubuntu operating system, NVIDIA Cuda for GPU acceleration, OpenCV and ROS software. Camera drivers were connected using ZED camera wrapper that connects to ROS and collects images from the ZED camera SDK. ZED camera is initiated from the start using ZED SDK which calculates and computes the image and depth maps using stereo camera techniques.

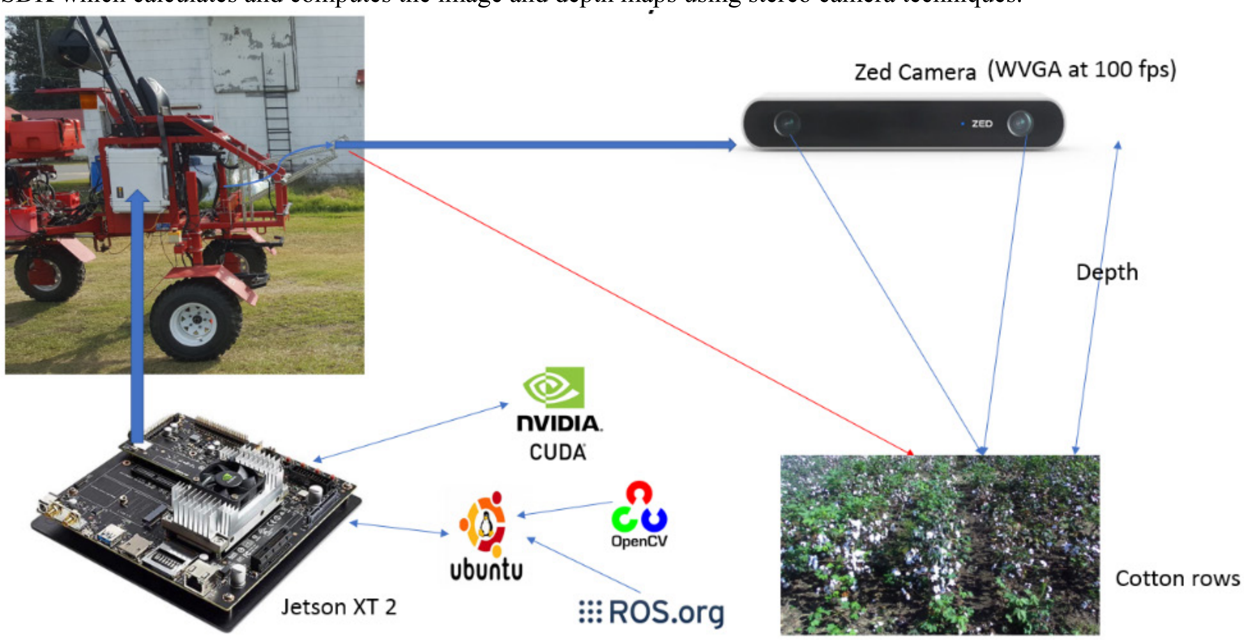


Figure 1. Machine vision components of the research

Data Collection

Field data were collected at the UGA Lang farm (31.521501, -83.545712) along Carpenter Road, Tifton, GA USA. The images were collected near the mid of the day on 28th August 2017. The images and depth map were stored in the internal memory of the development kit. 2546 images were collected, but for the purpose of this research, only 381 images which cover two rows going back and forth were used for analysis and validation of the model proposed in this research.

Left lens image Corresponding Depth Map

Figure 2. Collected RGB image and corresponding depth map (0-255, 8-bit greyscale image)

Image Analysis and Row detection

The images acquired were then analyzed by matching the left lens image with the depth map. Each of the images was checked and rectified to match the depth map. The images (Figure 2) represent a color and depth image of one part of the farm. In this section, seven steps were carefully taken to establish the model and test it. The steps are;

a. Camera Calibration
b. Calculate depth disparity and rectified left the image of the stereo camera
c. Apply a perspective transform to the depth map
d. Determine threshold binary image of the depth map and detect the rows by sliding window searching
e. Determine best match position at the center and rows position relative to the rover and define and show the detection levels as Success, Not Sure or No Detection

Camera Calibration

Using ZED camera software, the cameras were calibrated to achieve the best estimation of the image coordinates and real-world coordinates. Calibration of the camera is important since the model will estimate the center of the rover and position the wheels along the rows. The parameters cx, cy, fx, fy, k1, and k2 were found. The symbols fx and fy are the focal lengths in pixels while **cx** and **cy** are the optical center coordinates in pixels. **k1** and **k2** are distortion parameters used to rectify the images. The ZED SDK does this work in the background, and the rectified images are supplied when requested using the ZED application programming interface.

cx = 674.221

cy = 374.301

fx = 697.929

fy = 697.929

k1 = -0.173398

k2 = 0.0287331

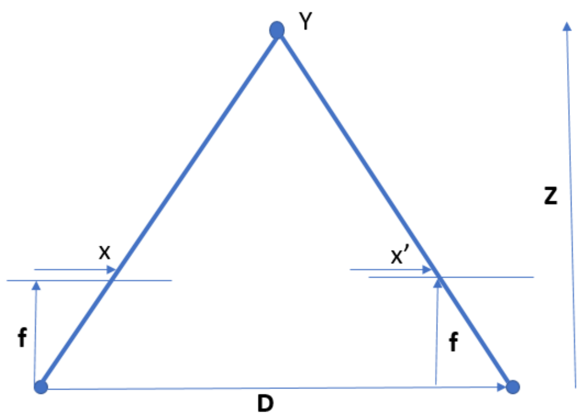
$$\begin{Bmatrix} I_x \\ I_y \\ 1 \end{Bmatrix} = \begin{Bmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{Bmatrix} * \begin{Bmatrix} W_x \\ W_y \\ W_z \\ 1 \end{Bmatrix} \tag{1}$$

Ix and Iy are image coordinates while Wx, Wy, Wz are real world coordinates

Using, calibrated values, the image coordinates can be transformed into real-world coordinates using equation 1 above.

Calculate depth disparity and rectified Left image of the stereo camera

The images obtained from the left lens of the camera are rectified by balancing and removing distortion. Then using the right and left lens image, the disparity is calculated by the law of registration of the distance between the two lenses and the location of the point targeted (Lucas and Kanade, 1981).



$$disparity = x - x' = \frac{fD}{Z}$$

Figure 3. disparity calculation. Where x and x' are the distance between points in image plane corresponding to the 3D scene point and their camera center. D is the distance between two lenses of the stereo camera while f is the focal length of both lenses.

Apply a perspective transform to the depth map

In order to balance the view and determine the row width, it is best to transform the depth map. The transform is done by the birds' eye view model in which it is assumed that the close pixels will present the rows as large while the back ones will be small. Hence, the perspective transform will choose a region of interest and transform. The transformation will be successfully performed on undistorted images. The source image points and destination points are determined as in Table 1. The transformation vertices were determined experimentally by testing several images and come up with figures that suited the camera orientation.

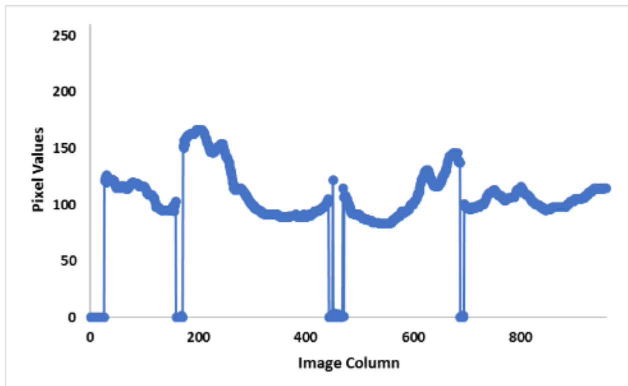
Table 1. The perspective transformation vertices for depth maps

Source Image points (Vertices)	Destination Image Points (Vertices)
0.65*960, 0.65*540	960*0.75,0
960,540	960*0.75, 540
0,540	960*0.25, 540
0.40*960, 0.40*540	960*0.25, 0

Determine threshold binary image of the depth map and Detect the rows by sliding window searching

Most of the rows are straight, but the row detection algorithm may calculate a bending row. This is because of the unequal growth of the plants. The upper pixels (which represents the upper part of the canopy) are pixels that are high 8-bit values compared to lower pixels which represent the lower part of the canopy and land and have low 8-bit values. So in each row of the depth map, these values can be determined. The sliding window method calculates for every 10x10 pixels by finding the average 8-bit pixel gray value. For depth maps, the black pixels represent lower pixels while the upper pixels are represented by white pixels. This means the lower pixels represent further objects from the camera while the white pixels closer to the camera. In fact, the color changes from whitish, grayish to blackish is the pixel 8-bit value becomes small. In this essence, the sliding window will group the highest pixels and predict the path. The path is achieved by connecting all the pixels that are the highest pixel values which are determined by choosing the 70th percentile of the pixel values. This technique introduced errors with the plane that has no plants; still, the sliding window may try to get the best upper pixels to detect the rows despite the non-existent rows. Figure 4 presents the depth map that was manipulated at the center (Particularly at row 300 out of 540 pixels available). The map is 960 pixels wide. So, each of the pixel taken aboard is statically manipulated to get the 70^{th} percentile which in this case was 116.5. Then, all the pixels with the value above 116 are formatted to white and given value 255 while all others are brought to black value zero. This binary image obtained is further manipulated. Figure 5 presents a depth map that was changed to a binary image and then transformed to locate the rows and then the sliding window is used to detect the rows. The upper figure presents a raw depth map. The binary map at the center is obtained by applying 70^{th} percentile of the row pixels (Figure 4). Then, the sliding window is now used to group the pixels and detect the rows. It is clearly seen that after pixel number 200 is the left row and after pixel 600 is the right row. If the difference between 90^{th} percentile and 10^{th} percentile is less than 60, then we convert the whole pixels of the row to black or zero.

1


2


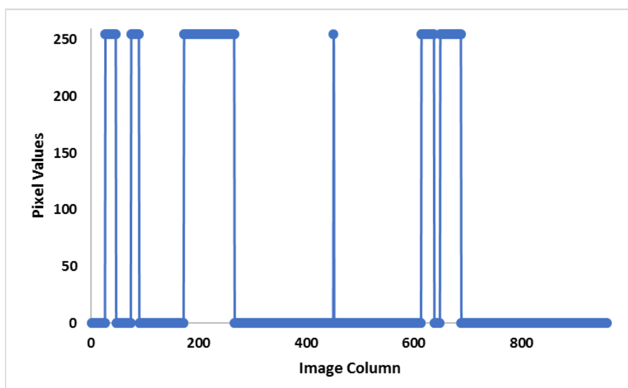
3


Figure 4. Depth map at the row pixel number 300 (each image has 540-row pixels) (1) the pixels are captured in the histogram (2) and then using 70th percentile only pixels with a value above 116 are used to form a binary image histogram (3).

Determine best match position at the center and rows position relative to the rover and define and show the levels of detection from success, not sure and no detection

The detection is always termed as the probability as it heavily depends on the appearance of the depth map. If the depth map is not uniform with many variations, it is difficult to get a 70th percentile of the pixels that show uniform changes in the plant canopy. So, the ranking was done (Figure 6) if the detection was good or no detection. The red means the rows align such that no detection was found while gray if the rows align that it is not easy to detect, but there is some sort of detection. The bottom image in Figure 5 shows the sliding window in green and matching red line for row detection. This means the detection was successful.

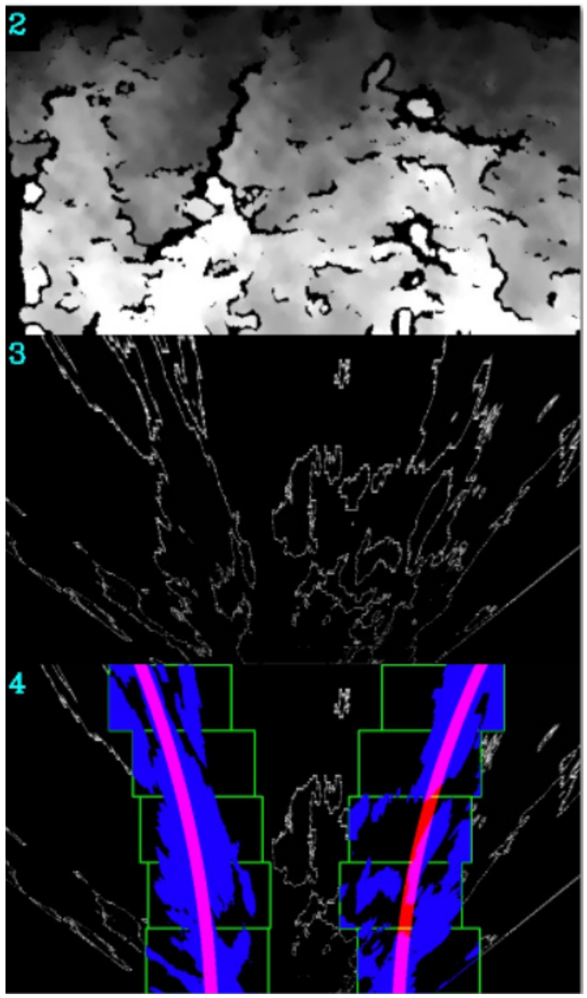


Figure 5. Rows are detected by using depth map (2), transformed binary depth map at the center (3) while lower image (4) represents the sliding window detection of the rows.

Figure 6. Ranking the detection of the rows (Upper image means detection was successful as it shows green and yellow strips, center image detection with gray strip was successful but not sure while the bottom with red strip means no rows were detected)

The binary pixels are detected by placing a 100x50 pixel window along the left and right row. Then, the points were fitted using a polynomial function to detect the rows. If the left and right rows appeared inside the sliding window by less than 40% offset, then this was a successful detection. The distance was achieved by calculating Manhattan distance between the center of the sliding window and the predicted polynomial fit. If one of them was less 100% offset detection, it was concluded as not sure. When the points were 100% offset from the sliding window, it was concluded as no detection.

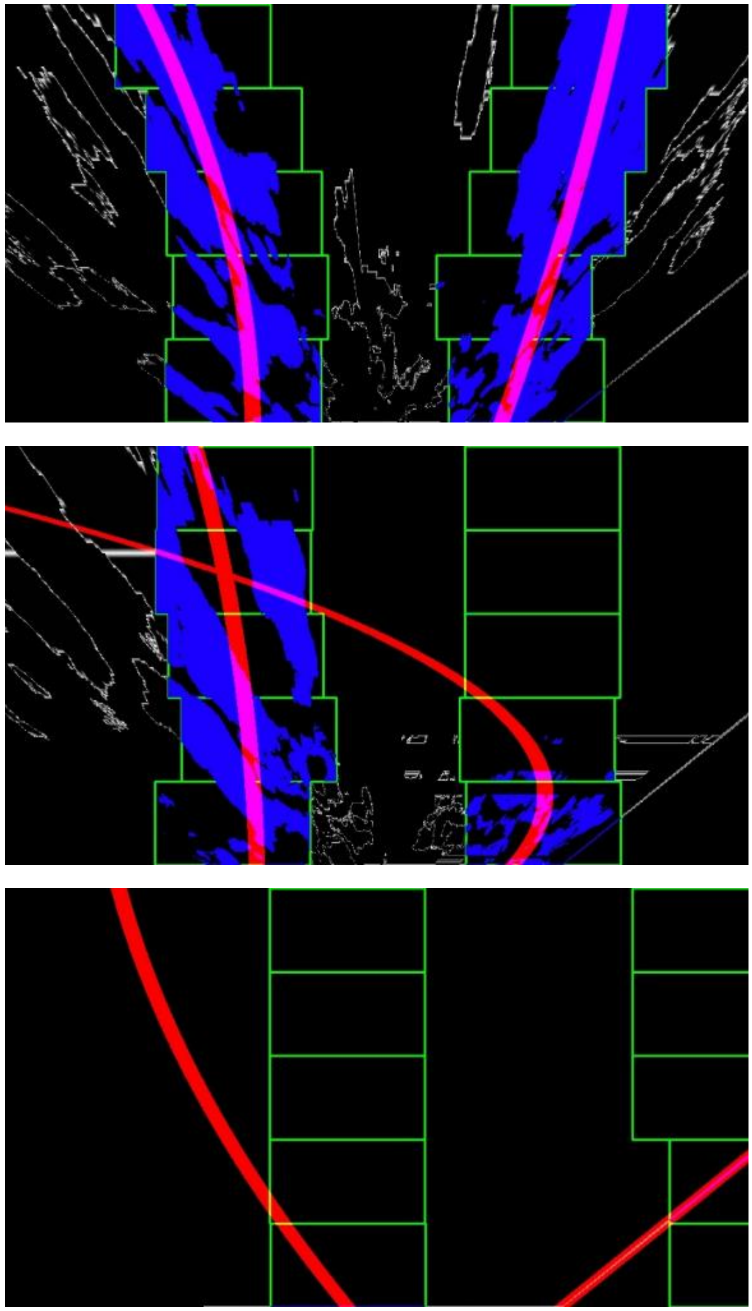


Figure 7. Sliding window comparison with polynomial fit using Manhattan distance. The upper image is the successful detection; center image means the detection is not sure while the lower image means no detection at all

Results and Discussions

The images collected were each evaluated. Results were based on the ranking of the software to decide whether the detection was successful. The model can only be used for navigating the rover if it shows promising results. True positive meant the detection was successful. Easy detection meant the software was sure with detection and all the rows aligned with a sliding window. Difficult meant some of the row pixels were out of the sliding window, but still, detection was a success. False Positive meant the detection was found in a place where there were no visible rows.

This appeared if there were plants other than cotton or the depth image obtained was blurred. True negative meant the rows were not seen and the software successfully detected that situation. Easy meant there are no differences between upper pixels and lowers pixels confirming no plant canopies and the system was 100% sure there were no rows while the difficult meant the situation was obstructed by having some small plant canopies. False Negative meant the system was detecting no rows because of the skips in the row where cotton was not growing. Consequently, the algorithm only detects one row but cannot comprehend the other. The row that was detected was also not clearly seen. This investigation of the images was manually done. 381 images were investigated thoroughly, and data collected for analysis of the model.

Table 2. Results of the manual inspection of the images

	Easy	**Difficult**	**Total**
True Positive	207	76	283
False Positive	00	01	01
True Negative	54	15	69
False Negative	05	23	28
Total	**266**	**115**	**381**

$$precision = TP/(TP + FP) = 283/284 = 0.996$$

$$recall = TP/(TP + FN) = 283/(283 + 28) = 0.909$$

$$F1\,Score = 2 * \frac{Precision * recall}{precision + recall} = 0.951$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{FN} + \text{TN}) = (283 + 69)/381 = 0.923$$

The model was evaluated from the data collected (Table 2), and the model was found to perform at 92.3% accuracy with F1-score of 0.951. The algorithm was accurate, but it had many difficulties in predicting the rows that had short or taller plants. The situation has led the algorithm to predict many false negatives.

Conclusion

The performance of the algorithm developed in this research showed promise as an ensemble method with RTK GPS to navigate an intelligent rover for harvesting cotton bolls. Adding a visual system to navigation provides the increased perception of the environment to avoid obstacles and to clearly see the actual row path for the vehicle. Also, in case of failures of the RTK-GPS then the visual systems can help the robot to navigate. The 92.3% accuracy and F1 score of 0.951 were quite sufficient to deploy the algorithm for real-time robot operations.

Acknowledgements

The authors are very thankful for project support from Cotton Incorporated, Agency Assignment #17-038. We also thank Mr. Ricky Fletcher and Mr. Gary Burnham for their helpfulness and technical support in building camera platform and collection of data.

References

Fue, K., Porter, W., and Rains, G. (2018). Real-Time 3D Measurement of Cotton Boll Positions Using Machine Vision Under Field Conditions. BWCC Paper No. 18385. Cordova, TN: NCC

Fue, K. G., Porter, W. M., & Rains, G. C. (2018). Deep Learning based Real-time GPU-accelerated Tracking and Counting of Cotton Bolls under Field Conditions using a Moving Camera. In *2018 ASABE Annual International Meeting*. ASABE Paper No 1800831. St. Joseph, MI: ASABE. doi:10.13031/aim.201800831

Hayes, L. (2017). Those Cotton Picking Robots. Available at http://georgia.growingamerica.com/features/2017/08/those-cotton-picking-robots/

Higuti, V. A., Velasquez, A. E., Magalhaes, D. V., Becker, M., & Chowdhary, G. (2018). Under canopy light detection and ranging based autonomous navigation. *Journal of Field Robotics*.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

Rains, G. C., Faircloth, A. G., Thai, C., & Raper, R. L. (2014). Evaluation of a simple pure pursuit path-following algorithm for an autonomous, articulated-steer vehicle. *Applied engineering in agriculture*, *30*(3), 367-374.

Rains, G., B. Bazemore, K. Ahlin, A. Hu, N. Sadegh, & McMurray, G. (2015). Steps towards an Autonomous Field Scout and Sampling System. ASABE Paper No. 15219077. St. Joseph, MI: ASABE.