REAL-TIME 3-D MEASUREMENT OF COTTON BOLL POSITIONS USING MACHINE VISION UNDER FIELD CONDITIONS

K. G. Fue
G. C. Rains
W. M. Porter
University of Georgia, Tifton Campus
Tifton, GA

Abstract

Cotton harvesting is performed by expensive combine harvesters that hinder small to medium-size cotton farmers Advances in robotics provide an opportunity to harvest cotton using small and robust autonomous rovers that can be deployed in the field as an “army” of harvesters. This paradigm shift in cotton harvesting requires high accuracy 3D measurement of the cotton boll position under field conditions. This in-field high throughput phenotyping of cotton boll position includes real-time image acquisition, depth processing, color segmentation, features extraction and determination of cotton boll position. In this study, a 3D camera system was mounted on a research rover at 82° below the horizontal and took 720p images at the rate of 15 frames per second while the rover was moving over 2-rows of potted defoliated cotton plants. The software development kit provided by the camera manufacturer was installed and used to process and provide a disparity map of cotton bolls. The system was installed with the Robot Operating System (ROS) to provide live image frames to client computers wirelessly and in real time. Cotton boll distances from the ground were determined using a 4-step machine vision algorithm (depth processing, color segmentation, feature extraction and frame matching for position determination). The 3D camera used provided distance of the boll from the left lens and algorithms were developed to provide vertical distance from the ground and horizontal distance from the rover. Comparing the cotton boll distance above the ground with manual measurements, the system achieved an average R^2 value of 99% with 9 mm RMSE when stationary and 95% with 34 mm RMSE when moving at approximately 0.64 km/h. This level of accuracy is favourable for proceeding to the next step of simultaneous localization and mapping of cotton bolls and robotic harvesting.

Introduction

Cotton harvesting has been performed by expensive boll-picking harvesters for many years in the U.S. The cotton-picking harvesters cost more than $600k, putting them out of reach of a large segment of farmers and used less than 4 months a year. In addition, these machines are heavy, and expensive to service and maintain. Breakdowns during harvest can become an issue, making it difficult to harvest cotton in a timely manner for best cotton fiber yield and quality. Therefore, a novel approach is being investigated that would provide simpler, lighter, smaller, more energy efficient and cheaper harvesters. Current advances in computer processing speed, robotics and machine vision technology provide an opportunity to develop automated harvesting machines that can be robust, easy to use, convenient and affordable.

Several harvesting robots have been developed as research tools or in production agriculture; such as harvesting robots for cucumbers (Henten et al., 2003), grapes (Kondo, 2014), apples (Li et al., 2016), tomatoes (Zhao et al., 2016), strawberries (Hayashi et al., 2014) and sweet peppers (Bac et al.,2017). Most of these robots are slow to pick fruit due to the technology and technique used to identify, locate and pick the product. The faster robotic machines use a prescription map and multiple robotic arms to harvest many fruit at once (Zion et al., 2014). This is in part due to the difficulty that arises from trying to control the complex farming environment that has variable lighting, dusty conditions, and machine vibrations which produces “noise” to the imaging system (Bac et al., 2017). These impeding conditions are still challenging even to the current machine vision technologies.

Currently, machine vision techniques using LiDAR have been used to determine height of the cotton plant but not “on the go” (Jiang et al., 2016) which is a main requirement for this project. There are commercial available machine vision systems (e.g. TOMRA sentinel II optical sorter) that work “on the go” to sort good and bad tomatoes and peaches during harvesting at high speed of about 40 - 200 tons/hour (Tomra,2017). It has horizontal rows of kicking bars that are set such that tomatoes that are brought by the conveyor belt and drop over the bars and then the bars will kick out green or bad tomatoes when seen in close enough (Tomra, 2017). Machine vision harvesting robots for cotton picking are not yet available though some research has been conducted by Wang et al. (2007) and Wang et al. (2008). Wang et al., (2008) were able to develop an imaging system for cotton recognition using color segmentation methods and detect the boll at an accuracy of 85%. Furthermore, some work for visual navigation of cotton picking robot has been done using the Otsu method and noise filtering techniques (Xu, 2015). Research in India was attempted to design an automatic cotton-picking robot that used image processing techniques to get features, do modelling and matching but a commercial product was not developed (Rao, 2013). Several studies have approached the harvesting of cotton bolls by using stereoscopic cameras and image processing algorithms (Xu, 2015; Rao, 2013; Kondo, 2014 and Hayashi, 2014).

In this study, the machine imaging systems that can locate and determine 3-D location of the boll was developed and investigated. 3-D location meant the vertical and horizontal distance of the boll from the centre of the camera carrying platform. This location would be predetermined and then signals sent to the robotic controller to move the end effector to pick the boll using computing algorithms or models. However, to achieve this harvesting act, machines need high performing computing and imaging resources. Present computing technologies could provide a quick solution for cotton boll localization and mapping that provided the transformation used to position the robot end effector for harvesting. An x-y Cartesian robotic arm placed vertically could move in two axes: up/down and left/right (Lumelsky, 1986; Zefran, 1996). Hence, an imaging system was required to predetermine the cotton boll position and feed the information to the machine and then, the Cartesian robot arm could plan and move to pick the cotton boll (Lumelsky, 1996; Zefran, 1996).

To reach the primary goal of cotton boll harvesting using a robotic arm, a system of sensors and computer processing algorithms must be able to identify and determine the position of the boll relative to movement and position of the arm end effector. This must be done in real-time so as to provide the robotic arm controller the information necessary to move to the location of the boll and pick it with the end effector. To take the steps of boll identification and 3-D position determination we conducted the following objectives:

1. Acquired images using a ZED 3-D RGB camera and processed images using fast CPU and GPU
2. Classified cotton bolls using machine vision algorithms with the fastest solution convergence
3. Determined position of bolls using depth detection of 3-D camera
4. Experimentally measured the accuracy of classification and position measurement of cotton bolls at static and dynamic (3 rover speeds) conditions.

Materials and Methods

Imaging System

A development kit (NVIDIA Jetson TX2 development kit, Nvidia Corp., Santa Clara, CA, USA) was installed on the research rover and, together with machine vision software, used to extract features of cotton boll images and determine the 3D position of the boll relative to camera and ground. NVIDIA Jetson TX2 (NVDIA Pascal 256 CUDA cores, Quad ARM and HMP Dual Denver CPU, 8GB 128-bit LPDDR4 RAM, 32GB eMMC SATA drive) was used to provide high graphic computing resources for fast image analysis. This system used image transformation algorithms that required fast computing to achieve real-time requirements of the robotic arm. An RGB stereo camera (ZED camera, Stereo labs Inc, San Francisco, CA, USA) was installed and used to acquire images. ZED is 175 x 30 x 33 mm and weighs 159g. ZED has a 4M pixel sensor per lens with large 2-micron pixels. The left and right sensors are 120 cm apart. ZED was chosen due to the nature of the tasks, such as needing to work outdoor and provide depth data in real-time. ZED camera provides a 3D rendering of the scene using the ZED software development kit (SDK) which is compatible with other platforms like Robot operating system (ROS), OpenCV library, MATLAB and Unity. ROS was chosen since it provides all the services required for robot development like device drivers, visualisers, message-passing, package design and management and hardware abstraction (ROS, 2017). ROS was initiated remotely by using a client machine and images were acquired using the ROS topics feature provided by the ZED wrapper. Images were parsed to the processing unit and analysed using OpenCV (version 3.3.0) machine vision algorithms.

The ZED camera system was mounted on a research rover (Rains et al., 2015) at 82° below the horizontal and took images at the rate of 15 frames per second while the rover was moving at less than 1.6 km/h. The research rover is a custom-built articulated vehicle (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 structured row crop field (Rains et al., 2015). However, due to processing speed of the development kit, the camera and machine vision system were able to process approximately 2 frames per second. Distance of the camera to the ground was 1550 cm.

Data acquisition and Boll Image Extraction

Each image frame was acquired using ZED camera and analysed using a 4-step machine vision algorithm (1. depth processing, 2. colour segmentation, 3. feature extraction and 4. frame matching for position determination). These steps are handled by the graphic optimized rugged development kit (NVIDIA Jetson TX2) to achieve improved performance as image calculations required massive graphic computing resources like NVIDIA CUDA cores.

Depth processing is achieved by using the ZED stereo camera which has two lenses with separate image sensor for each lens. This arrangement allowed the camera to have an ability to process 3D images that provides depth of a cotton boll from the camera. This proximity of the cotton boll map can then be used to determine its' distance from the ground as well as the horizontal distance and vertical distance from the center of the camera carrying platform.

Since the ZED is an RGB camera, a passive sensor uses sun light reflection on each pixel to creatge an image. Consequently, varying light illumination can alter image clarity and boll classification frame-to-frame. In addition, bolls visible to the sensor in one frame may become occluded in a subsequent frame of the boll from a different viewpoint of the camera. and the boll detection algorithm must have built-in intelligence to remember the last position of the boll even when it appears undetected in future image frames. Color segmentation was implemented by using machine vision algorithms deployed in the OpenCV library (Gong and Sakauchi,1995). A machine vision algorithm was required to mask/subtract all background environment and leave cotton bolls in the frame. Since cotton bolls are white, the algorithms then needed to mask white objects from the environment. The cotton boll classification task involved four steps (Gong and Sakauchi, 1995):

1. Grab an image
2. Using RGB color threshold, separate each RGB component of the image. For cotton bolls, the white components of the image can be masked.
3. Subtract the image background from original image.
4. Remove all the region where the contours are less than value M. Value M can be determined by estimating the number of pixels defining the smallest boll.

The first step was achieved by applying a threshold to separate the white bolls from the background. For white cotton bolls the color range/threshold was set to 240-255 in the red, green, and blue channel (8-bit color depth map). This makes every boll detectable that gets good illumination in at least one image frame. The second step used feature matching and application of a Boolean AND operation between the mask image and the original image. The output image is then converted to greyscale.

The last step is feature extraction. This is done by finding contours of the continuous points which have the same intensity and are clustered. Color masking of the grey image was performed, then boundary curves were applied to detect and distinguish all white pixels of the image. For each contour, the center (centroid) was calculated and number of pixels that are together were determined. The threshold for number of pixels together that define a boll is called M. Two M values, 5 and 15 pixels were tested.

Frame Feature Extraction, Matching and Tracking

Frame matching was required to track the position of bolls in respective image frames. In some instances, the algorithm may miss the bolls due to illumination problems that impact brightness, contrast, and sharpness of the image. Hence, the system was developed such that it detected and remembered the boll locations in respective image frames. Since, the rover was moving while the bolls were stationary, multiple frames detected bolls with varying depth measurements. This is achieved by calculating projective transformation (homograph) matrix (3x3) that matches the point corresponding to two consecutive image frames.

To achieve this homograph matrix, two consecutive image frames were loaded to the CPU and the ORB feature extraction algorithm applied to get unique features for both frames. ORB was oriented FAST (Rosten et al., 2006) and rotated BRIEF (Calonder, et al.,2010; Rublee et al., 2011). ORB (an open source machine vision algorithm in OpenCV 3.3) was the fastest of all the feature extraction algorithms attempted in this study (Rublee et al., 2011). At least 5 feature extraction algorithms (FAST, ORB, SURF, SIFT and AKAZE) were tested. AKAZE feature extraction was proposed by Alcantarilla and Solutions (2011).

ORB was then adopted as the feature extraction method after showing outstanding performance. The OpenCV Brute force matcher, FLANN (Muja and Lowe, 2011) matcher and find Homography modules were tested to get the homograph transformation matrix. These algorithms were too slow to achieve the required speed as they were taking more than 4 seconds to process two images. C++ bytecode that uses 8 CUDA threads to utilize the GPU cores was written NVIDIA has CUDA cores that deploy fast graphics computing by parallel computing. The C++ bytecode program utilizes 8 out of 256 GPU cores because only two frames are loaded compared to other applications that deploy large number of images and hence may need more cores. Then, a brute force matcher algorithm that uses random sample consensus (RANSAC) was written by modifying code provided by (Nghia, 2011) and applied to the images to get the matching features between the images. The RANSAC algorithm can interpret data containing a lot of gross errors and hence is very useful for error-prone tasks like image processing (Fischler and Bolles, 1981). The algorithm used match scores to determine the best matches and left out the outlier (Fischler and Bolles, 1981). The inliers threshold was determined if at least 5 pixels of the frames matched, otherwise it was discarded. By assuming only 20% of the features to match, the system used RANSAC to iteratively estimate the data set that contained outliers.

RANSAC was done using the following procedures;

1. Randomly selected subset of data set, in this case 20% of the data set, and then tried to fit the model in the selected subset.
2. The algorithm determined the number of outliers and in this case, the data was tested against the fitted model and some of the points that fit the model were considered inliers of the consensus set.
3. The algorithms did several iterations to achieve the best homograph. The number of iterations was determined by the number of CUDA core blocks and threads. The program established 8 threads per block of the CUDA cores. The homograph was then parsed to the main program for tracking and logging boll positions.

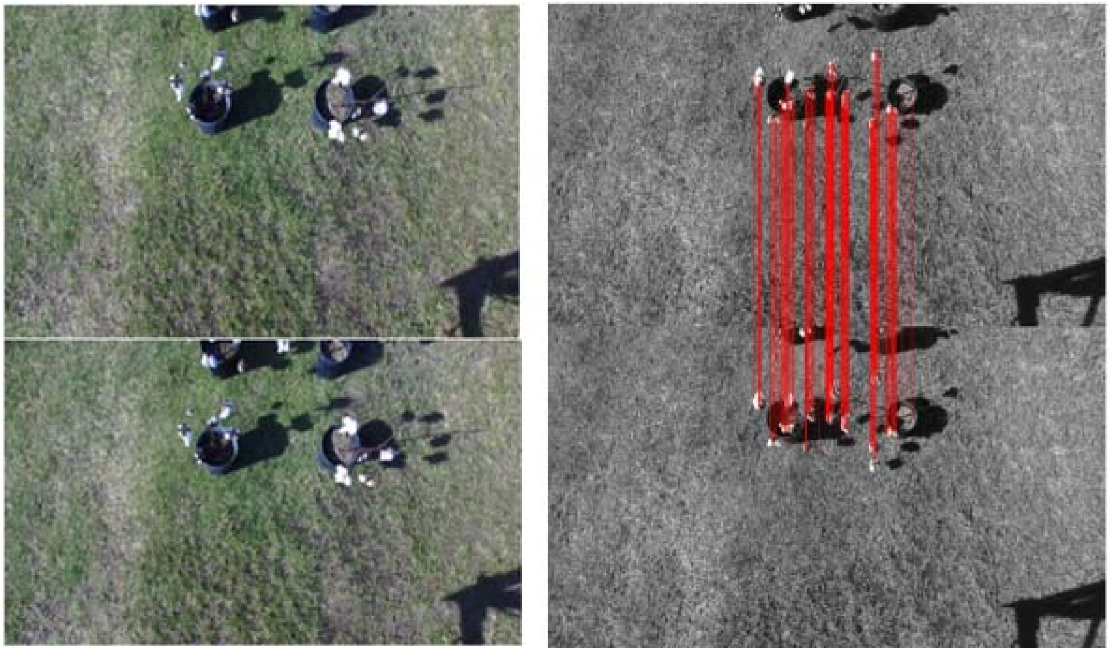

Figure 1. Two consecutive images collected from ZED and moving rover (left images) and matching features obtained using ORB and Homograph RANSAC (right image).

The overall imaging software was written using python, but RANSAC CUDA code was optimized using C++ code. Then, python sub process code was used to access the CUDA bytecode since the main code was written using python. With this CUDA optimized implementation, the system was able to process at least two images per second.

Using the homography transformation matrix, you can project and locate features between two related or consecutive frames as shown in Figure 1. Hence, the new boll position in the next image frame was obtained by multiplying the homography transformation matrix to the initial pixel position. For missing bolls in a new image frame, the system used past stored centroids multiplied by the homograph to get the new position of the bolls.

3-D Cotton Boll Position Measurements and Localization

After the homograph matrix was obtained, matching bolls centroids were determined by using the inverse of the homograph of the current frame and compared to the previous image frame. The boll position for each boll was logged and stored as an array. Figure 2 shows the flowchart of this procedure that was used in this research.

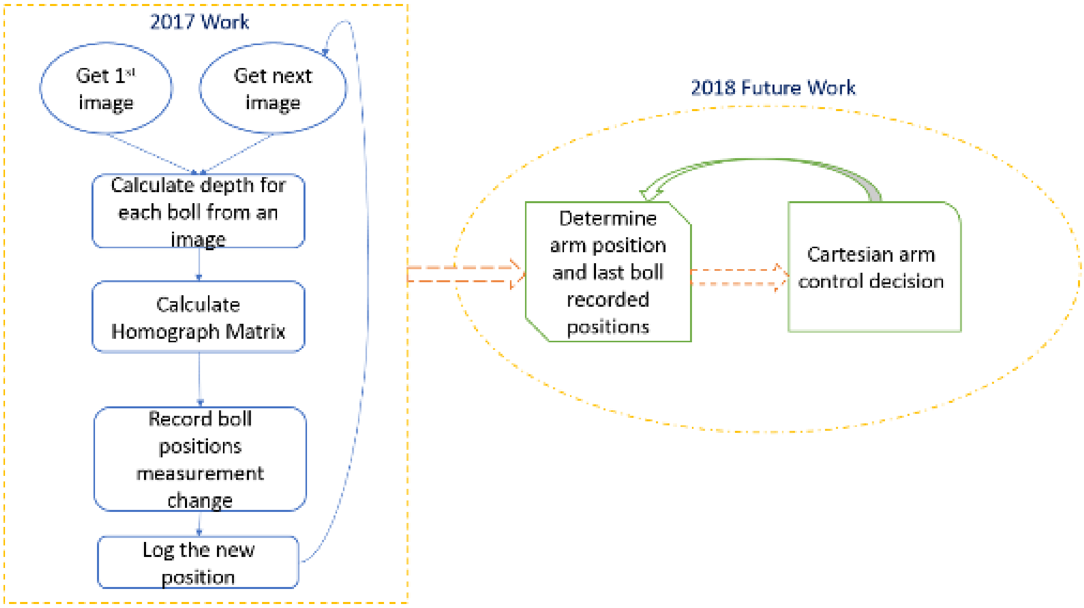
2017 Work
Get 1st image
Get next image
Calculate depth for each boll from an image
Calculate Homograph Matrix
Record boll positions measurement change
Log the new position
2018 Future Work
Determine arm position and last boll recorded positions
Cartesian arm control decision

Figure 2. This is the context of the research to develop cotton boll picking robot.

After detecting and establishing all the features of the frame, the rectified image was used to get the z-coordinate of the boll. The z-coordinate is the vertical distance of the boll from the ground (height from the ground). This distance was considered because it was easier to predetermine and test the system. But in real application, distance of the boll to the camera will be used to remove errors that may be introduced due to topography of the land and twitching of the rover when making left and right turn corrections. The sudden convulsive movements during turning of the rover was an artefact of the method used to turn the rover. A hydraulic on/off directional control valve (DCV) was used to turn the rover and this sudden change in hydraulic pressure caused a "twitch" in the rover and a subsequent side vibration of the camera, introducing errors and bad image frames. The camera obtained image depth from the rectified left image. Then, the depth map (disparity image) was obtained using interactive API provided by the SDK. This depth corresponded to a perpendicular distance between left camera and the boll. Hence, a model was developed to get the vertical distance of the boll from the ground. This measurement was the only coordinate that was determined as it is permanent while other readings were relative to this and change as the rover moved over the plants. The camera mounted on the rover was inclined at 82° from horizontal and obtained 1280 x 720 pixel frames. The field of view was covered at an angle of 54° (ϕ) vertically and 96° horizontally (Rosindustrial, 2016). This z-coordinate was also logged into the system accordingly. The system calculated the moving average as it grabbed images. This average was used to determine the 3D position of the boll relative to the future cotton-picking end effector. Considering Figure 3, we calculated the distance of the boll from the ground (n). By considering, the middle boll (brown arrow), we determined the depth of the boll and distance from the ground. The depth distance that was reported by camera was given by the formula m*atan(μ). By using ZED SDK API, we obtained the depth of each pixel with 16-bit resolution.

$$depth\ of\ an\ object = m * \mathrm{atan}(\mu)$$

Given that $l = m + n$

Since, the camera was inclined at angle μ° from horizontal, the equivalent angle in radians of the object is given by

$$angle = \frac{\pi}{180} * abs\left(\left(\left(\frac{720-y}{720}\right) * \phi\right) - (90 - \mu - \phi/2)\right)$$

Now, distance from the ground is given by

$$n = l - m * \mathrm{atan}(\mu) * \cos(angle)$$

Since, depth of an object is provided by ZED SDK then,

$$n = l - depth * \cos(angle)$$

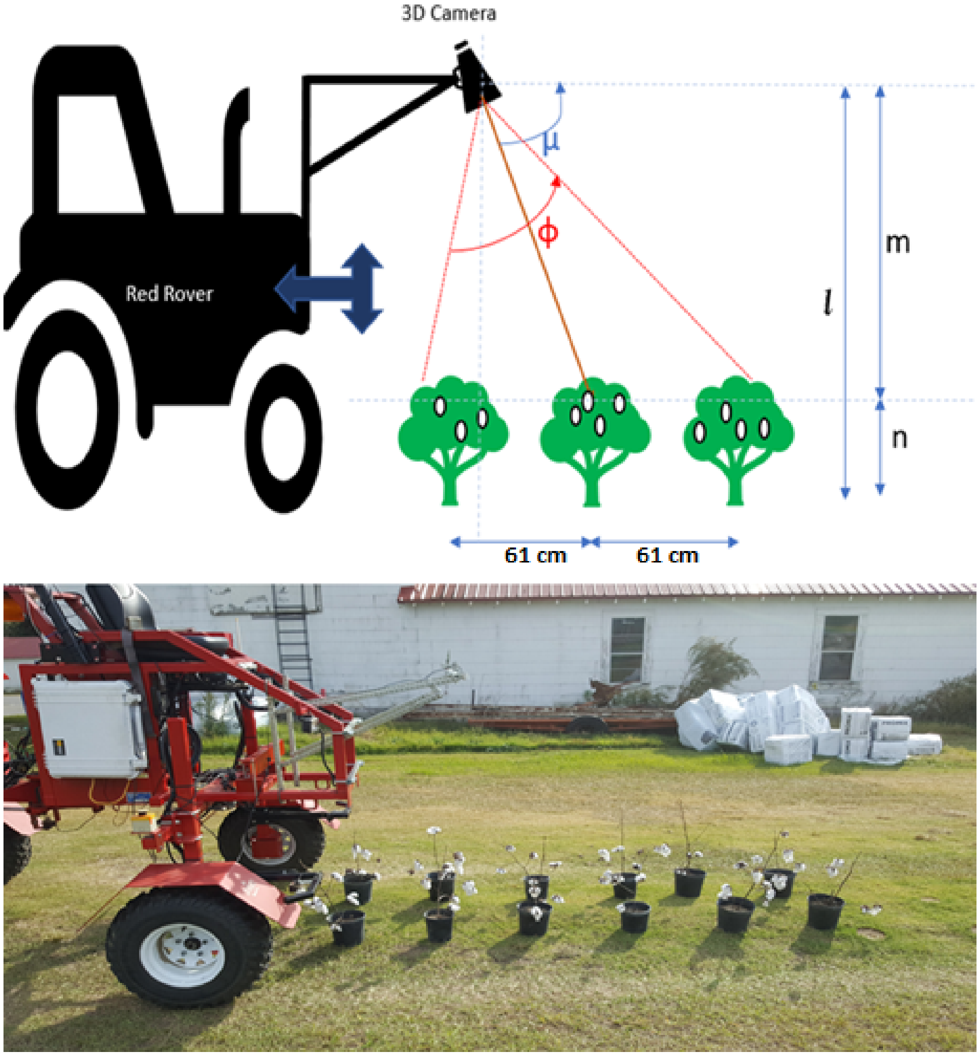

Figure 3. Context diagram that shows cotton boll position measurements

Publishing the Output Frames to External Clients

The system generated images that show how the bolls were tracked and z-coordinate was determined. The system published images using ROS and hence it was possible for clients to get live video of the frames. This video was slower as the system only calculated an average of two frames per second.

Experimental setup

An experiment was setup at UGA campus grounds (N Entomology Dr, Tifton, GA, 31793) at (31°28’N 83°31’W). The location was open to direct sunlight so as to measure under equivalent field conditions. Twelve defoliated cotton plants were taken from a nearby farm and put in pots. The plants were placed in 2 rows of 6, 91.4 cm between the centre of the stalk and each stalk 61 cm from the next. The distances of all bolls were measured manually and the rover driven over the bolls collecting RGB and depth information from the ZED camera. A static test was first conducted with the rover set over the bolls and two consecutive frames were taken. Then a dynamic test was conducted by driving the rover over the plants repeatedly at different speeds on 1st December 2017 as seen in Figure 4. On 4th December 2017, the plants were rearranged, and data collection was repeated 3 times with reduced speed for each. Speed was approximately 1.04 kph, 0.80 kph and 0.64 kph. Comparison of camera and manual measurement of boll locations were conducted. Comparative statistics were calculated to measure standard error, root mean square error, and mean error. Graphs were produced to show results.

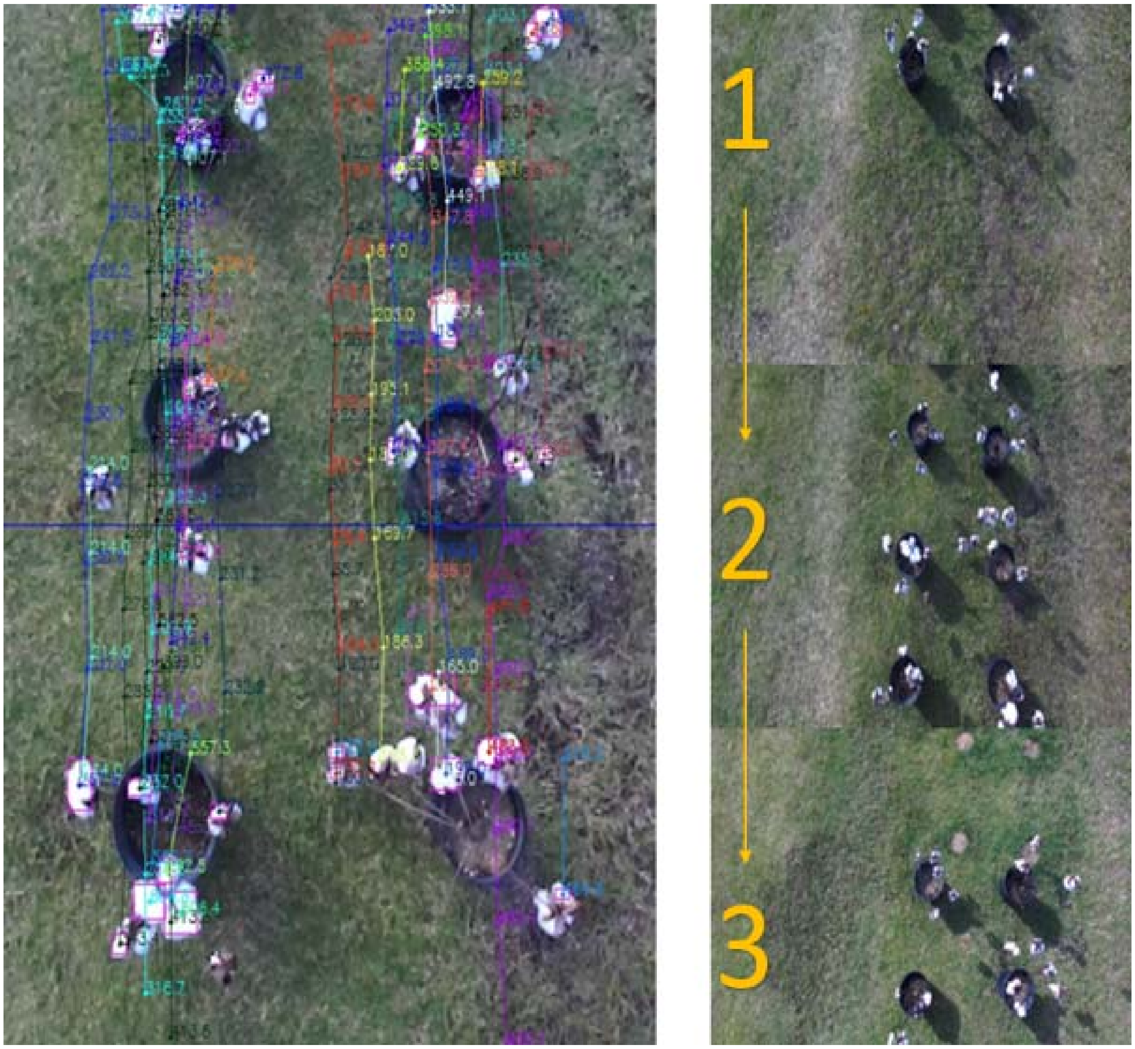

Figure 4. Left image show processed boll position and tracking of the boll while the right images show a series of 3 image frames acquired from the moving camera

Results and Discussion

A static test was conducted to assess the ability of the camera by classifying and locating bolls without dynamics of a moving camera. The red rover was set stationary, but running, and two frames were collected and analysed (Figure 5). Using Excel, the manual measurements were compared with camera measurements. 16 bolls were analysed using these two image frames. The first frame showed high accuracy of the camera by giving regression relationship with R^2 equal to 99% and root mean square of 11 mm. The second frame gave R^2 equal to 98% and root mean square of 16 mm. The mean error was -6 mm and 9.9 mm for first and second frame, respectively. The standard deviation was 9.8 mm and 14.8 mm for first and second frame, respectively. This shows the camera system was able to classify and locate bolls under field conditions with low cotton boll density.

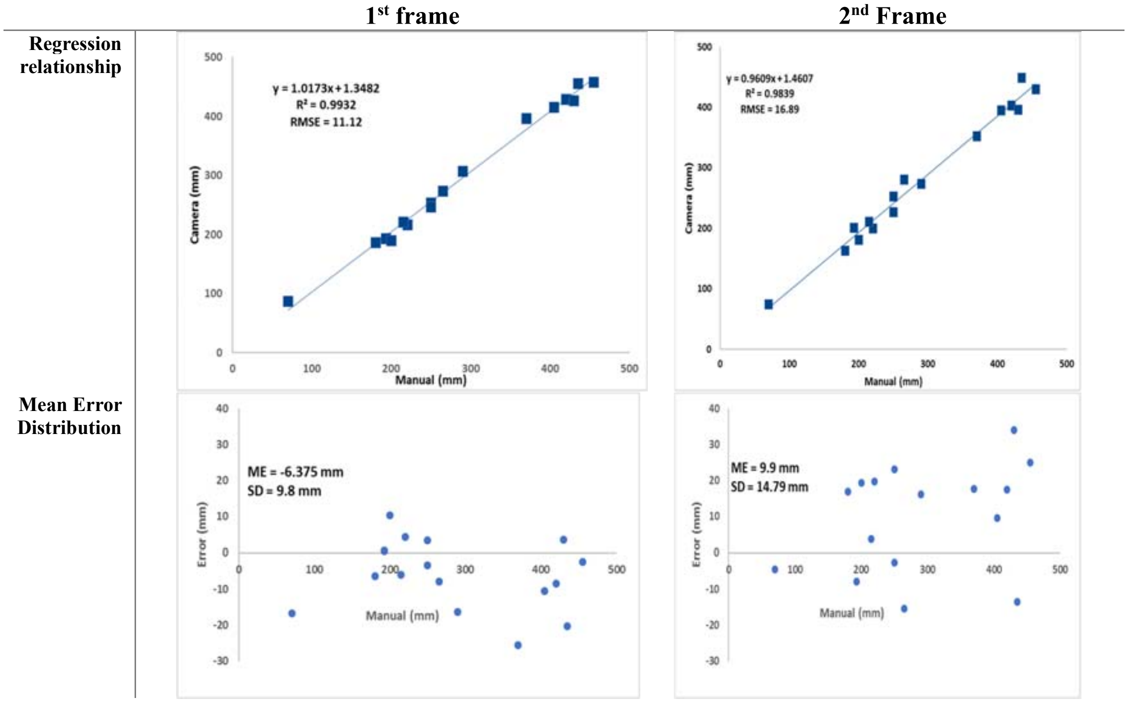

Figure 5. Comparison of the image frames when the rover is stationary

After the static test, a dynamic test of the rover moving continuously of the cotton plants was performed. The rover moved over the cotton plants while taking measurements. During the afternoon, the system was not able to process the image homograph moving uphill due to shadow obstructing the homograph transformation algorithm. The shadow of the rover appeared the same size for all the frames hence the transformation matrix had difficulty processing the image. The homograph RANSAC needs features to match for at least 20% of the features from both images. The rover passed over the cotton at 3 separate speeds, 1.04 km/h, 0.80 km/h and 0.64 km/h. Figure 4 demonstrates position of the boll in an image frame and tracks the changes of the position and determines the location the boll. The relative position of the boll was determined by measuring boll distance from the ground. The data were collected by reducing the M parameter (from 15 to 5) to detect white contours (cotton bolls). With this, the system detected the bolls and recorded multiple depths for the same boll. With 15-pixel boll contour, the system was only able to detect 92.3% of all 65 bolls available while in 5-pixel boll contour, the system was able to detect all the bolls. Figure 6 demonstrates the detection and masking. The white spaces need to form a contour that passed the threshold M-value, in this case 15 pixels or 5 pixels. The color of the boll and obstruction may make one boll detectible in one frame and not the next one. The same boll may also be detected in consecutive frames. By being detected more than once allows the system to obtain more than one depth reading for individual bolls. Multiple values obtained can be averaged to get a more accurate depth value. Figure 7 shows the regression relationship of the experiment for all three speed tests. The system had the worst R^2 of 86% for faster 15-pixel contour (1.04 kph) and 95% for slower 5-pixel contour (0.64 kph). Figure 8 shows the mean error distribution of the experiment. The 15-pixel rounds show unstable spreading of errors compared to 5-pixel rounds. These errors are mainly due to the “twitching” of the rover when adjusting the right and left turn and topography of the land. Rover turning will be converted from an on/off DCV to a proportional control valve over the winter to eliminate this noise issue and reduce the number of bad image frames collected by the camera.

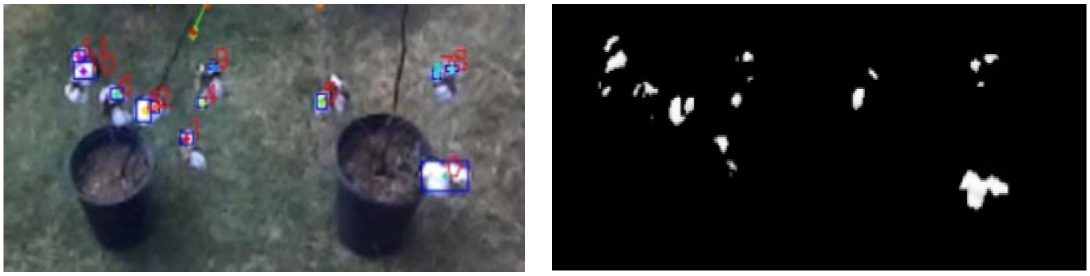

Figure 6. Comparing the contour detection. The larger the contour threshold the lower the number of bolls detected.

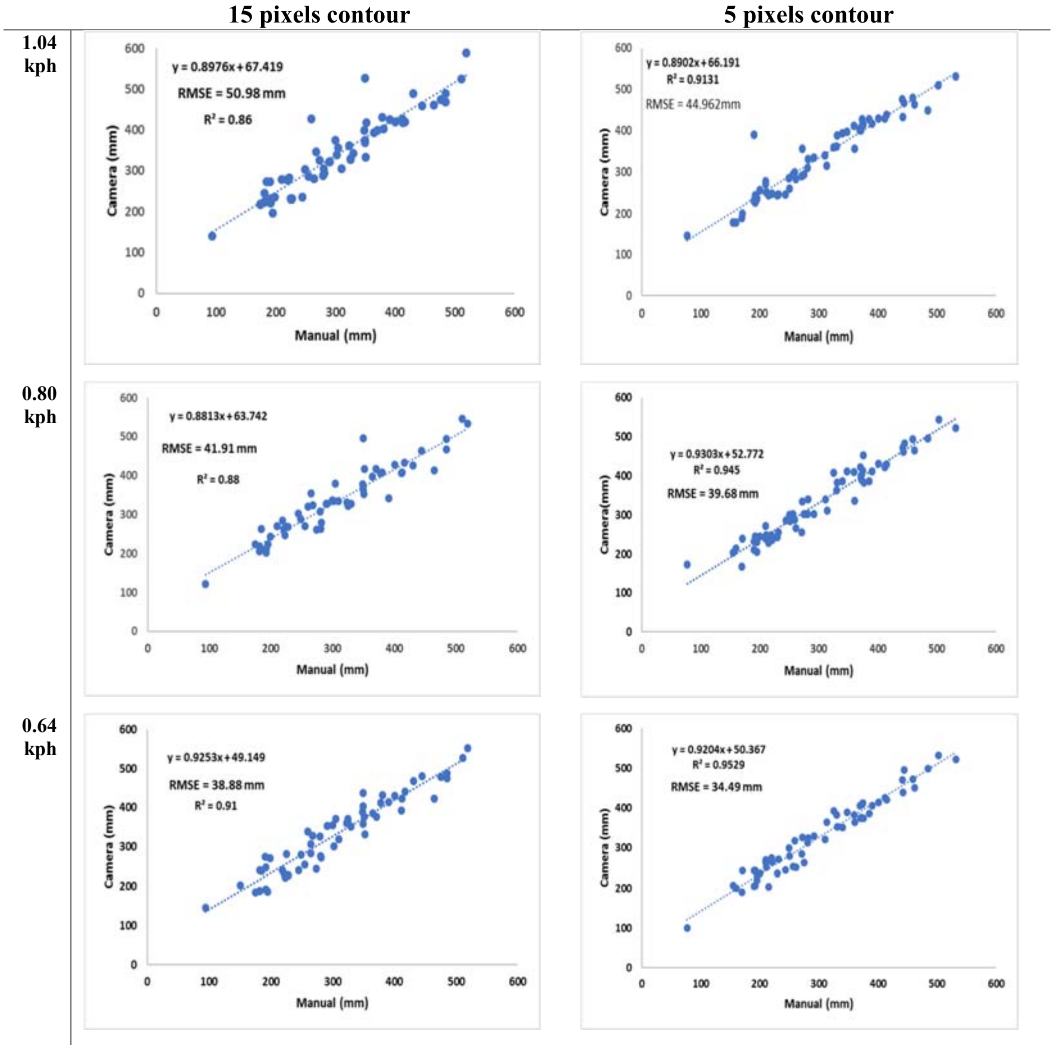

Figure 7. Comparison of 15 pixels contour and 5 pixels contour for 1.04 kph (fast speed), 0.80 kph (slow speed), 0.64 kph (very slow speed) rep of cotton boll position measurements

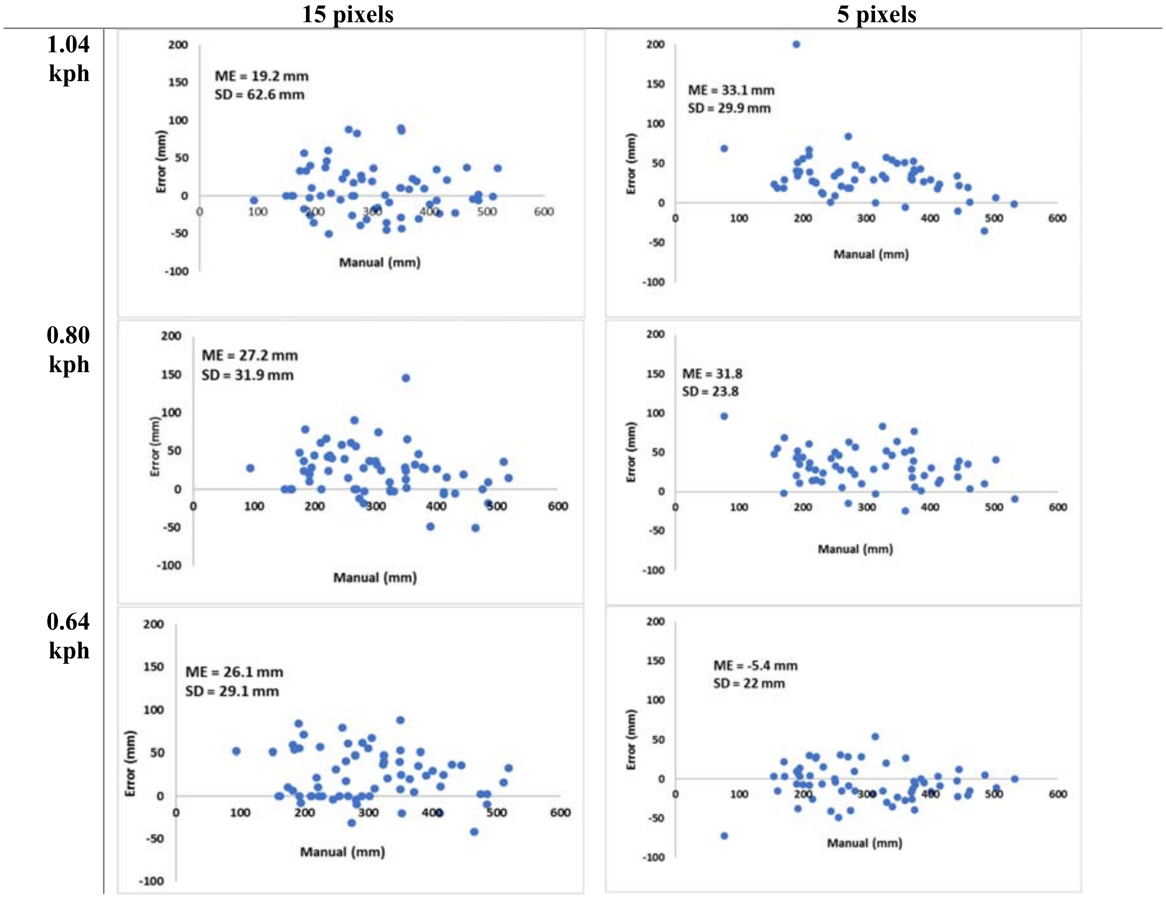

Figure 8. The Mean error distribution of the experiment (15 pixels contour and 5 pixels contour for 1.04 kph (fast speed), 0.80 kph (slow speed), 0.64 mph (very slow speed) rep of cotton boll position measurements).

Conclusion

The imaging system to determine 3D boll location in real time was developed in this study. This system was able to acquire images and process them using GPU resources of the system. The images were classified using a four-step machine vision algorithm with the fastest solution convergence. Then, distance of the bolls from the ground was determined to be the reference point for distance of the boll from camera. The distance of the boll from ground was not varying when the rover was stationary, hence can be measured and tested to be a reference. The boll changing coordinate along the image plane was determined and demonstrated. The experiment was carried out to determine the accuracy of classification and position measurement of cotton bolls at static and dynamic (different rover speeds) conditions. When comparing the boll detection and localizing system to manual measurements, the system achieved an average R^2 value of 99% with RMSE of 9.8 mm when stationary and 95% with RMSE of 34.49 mm when moving at less than 1.6 kph. The system performance was highly correlated to the speed of the rover and contour threshold (M) used to detect bolls. The lower rover speed showed better performance. This level of accuracy is favourable for proceeding to the next step of simultaneous localization and mapping of cotton bolls and robotic harvesting. The system processed an average of 2 image frames per second to perform boll classification and position. With the 5-pixel contour, the system may detect multiple contours compared to 15-pixel contour setting. The system will be modified to detect the nearby contours that form a real boll instead of multiple contours for one boll. This will help the end-effector to spend less time on one boll instead of processing several boll centroids due to multiple contours.

The current imaging system would have to stop in some locations to get best measurements of the 3D position of the boll in the field for boll picking with a robotic arm. We will reduce processing speed and image outliers with a combination of noise reduction to the camera system, upgrade to a faster on-board processing system and use more efficient algorithm coding.

References

Alcantarilla, P., and T. Solutions. 2011. Fast explicit diffusion for accelerated features in nonlinear scale spaces. IEEE Trans. Patt. Anal. Mach. Intell, 34(7), 1281-1298.

Bac, C., J. Hemming, B. Tuijl, R. Barth, E. Wais, and E. Henten. 2017. Performance Evaluation of a Harvesting Robot for Sweet Pepper. Journal of Field Robotics.

Calonder, M., V. Lepetit, C. Strecha, and P. Fua. 2010. Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010, pp. 778-792.

Dubrofsky, E. 2009. Homography estimation. Diplomová práce. Vancouver: Univerzita Britské Kolumbie.

Fischler, M. A., and R.C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), pp. 381-395.

Gong, Y., and M. Sakauchi. 1995. Detection of regions matching specified chromatic features. Computer vision and image understanding, 61(2), pp. 263-269.

Hayashi, S., S. Yamamoto, S. Saito, Y. Ochiai,J. Kamata, M. Kurita, and K. Yamamoto. 2014. Field operation of a movable strawberry-harvesting robot using a travel platform. Japan Agricultural Research Quarterly: JARQ, 48(3), pp. 307-316.

Jiang, Y., C. Li, and A. Paterson. 2016. High throughput phenotyping of cotton plant height using depth images under field conditions. Computers and Electronics in Agriculture, 130, pp. 57-68.

Kondo, N. 2014. Study on grape harvesting robot. Mathematical and control applications in agriculture and horticulture, 243-251.

Li, J., M. Karkee, Q. Zhang, K. Xiao, and T. Feng. 2016. Characterizing apple picking patterns for robotic harvesting. Computers and Electronics in Agriculture, 127, pp. 633-640.

Lumelsky, V. 1986. Continuous motion planning in unknown environment for a 3D cartesian robot arm. In Robotics and Automation. Proceedings. 1986 IEEE International Conference on (Vol. 3, pp. 1050-1055). IEEE.

Muja, M., and D. Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11), pp. 2227-2240.

Nghia, H. 2011. http://nghiaho.com/?p=490 retrieved on 10/23/2017

Rains, G., B. Bazemore, K. Ahlin, A. Hu, N. Sadegh, and G. McMurray. 2015. Steps towards an Autonomous Field Scout and Sampling System. In 2015 ASABE Annual International Meeting (p. 1). American Society of Agricultural and Biological Engineers.

Rao, U. S. N. 2013. Design of automatic cotton picking robot with Machine vision using Image Processing algorithms. In Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on (pp. 1-5). IEEE.

ROS. 2017. http://wiki.ros.org/ retrieve on 12/01/2017

Rosindustrial. 2016. http://rosindustrial.org/news/2016/1/13/3d-camera-survey retrieved on 12/01/2017

Rosten, E., and T. Drummond. 2006. Machine learning for high-speed corner detection. Computer Vision–ECCV 2006, pp. 430-443.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In Computer Vision (ICCV), 2011 IEEE international conference on (pp. 2564-2571). IEEE.

StereoLabs. 2017. https://www.stereolabs.com/documentation/overview/depth-sensing/advanced-settings.html retrieved on 12/01/2017

Tomra. 2017. https://www.tomra.com/en/sorting/food/sorting-equipment/sentinel/ [retrieved on 12/01/2017]

USDA-NASS. 2016. Crop production 2015 summary: January 2016. Washington D.C.: USDA National Agricultural Statistics Service. [www.usda.gov/nass/PUBS/TODAYRPT/cropan16.pdf Retrieved on 12/01/2017]

Van Henten, E., B. Van Tuijl, J. Hemming, J. Kornet, J. Bontsema, and E. Van Os. 2003. Field test of an autonomous cucumber picking robot. Biosystems engineering, 86(3), pp. 305-313.

Wang, M., J. Wei, J. Yuan, and K. Xu. 2008. A research for intelligent cotton picking robot based on machine vision. In Information and Automation, 2008. ICIA 2008. International Conference on (pp. 800-803). IEEE.

Wang, Y., X. Zhu, and C. Ji. 2007. Machine vision based cotton recognition for cotton harvesting robot. In International Conference on Computer and Computing Technologies in Agriculture (pp. 1421-1425). Springer, Boston, MA.

Whitaker, J., S. Culpepper, M. Freeman, G. Harris, B. Kemerait, C. Perry, W. Porter, P. Roberts, D. Shurley, and A. Smith.2017. Georgia cotton production guide. Cooperative Extension Service/The University of Georgia, College of Agricultural and Environmental Sciences Bulletin, Tifton. [http://www.ugacotton.com/vault/file/2017-Georgia-Cotton-Production-Guide.pdf retrieved on 12/01/2017]

Xu, S., J. Wu, L. Zhu, W. Li, Y. Wang, and N. Wang, 2015. A novel monocular visual navigation method for cotton-picking robot based on horizontal spline segmentation. In MIPPR 2015: Automatic Target Recognition and Navigation (Vol. 9812, p. 98121B). International Society for Optics and Photonics.

Zefran, M. 1996. Continuous methods for motion planning. IRCS Technical Reports Series, 111.
Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). Robust tomato recognition for robotic harvesting using feature images fusion. Sensors, 16(2), 173.

Zion, B., M. Mann, D. Levin, A. Shilo, D. Rubinstein, and I. Shmulevich. 2014. Harvest-order planning for a multiarm robotic harvester. Computers and Electronics in Agriculture, 103, pp. 75-81.