Autonomous Fruit Harvester with Machine Vision

Kathleen Anne M. Almendral, Rona Mae G. Babaran, Bryan Jones C. Carzon,
Karl Patrick K. Cu, Jasmine M. Lalanto, and Alexander C. Abad
Electronics and Communications Engineering Department of
De La Salle University, Manila,
2401 Taft Ave., Malate, Manila 1004, Philippines.
kathleen_almendral@dlsu.edu.ph

***Abstract*—This study presents an autonomous fruit harvester with a machine vision capable of detecting and picking or cutting an orange fruit from a tree. The system of is composed of a six-degrees of freedom (6-DOF) robotic arm mounted on a four-wheeled electric kart. The kart uses ZED stereo camera for depth estimation of a target. It can also be used to detect trees using the green detection algorithm. Image processing is done using Microsoft Visual Studio and OpenCV library. The x & y coordinates and distance of the tree are passed on to Arduino microcontroller as inputs to motor control of the wheels. When the kart is less than 65cm to the tree, the kart stops and the robotic arm system takes over to search and harvest orange fruits. The robotic arm has a webcam and ultrasonic sensor attached at its end-effector. The webcam is used for orange fruit detection while ultrasonic sensor is used to provide feedback on the distance of the orange fruit to end-effector. Multiple fruit harvesting is successfully done. The success rate of harvesting and putting fruit into the basket is 80% and 85% for the gripper end-effector and cutter end-effector respectively.**

***Index Terms*—Fruit Harvester; Machine Vision; Robotic Arm; ZED Stereo Camera.**

I. INTRODUCTION

The field of robotics has been continuously growing at an exceptional speed throughout the years. Robots are now being used in agriculture or farming [1-3]. An example of a robotic invention that is being used in the agricultural field is a fruit harvester or fruit picker [4-6]. The fruit harvester's purpose is to be able to identify fruits and pick it without the intervention of humans. This type of machine is very helpful especially in large sized farms that can only employ a handful of people to pick the fruits. One of the vital parts for these types of robots is how to sense and detect the fruits that will be picked or harvested. The recognition system of the most fruit harvester robots rely on computer vision using camera and image processing [7-11]. Image processing uses algorithms for recognizing visual patterns for the fruits especially in using this visual patterns to communicate with the computer by extracting and examining the content of the image to determine what kind of fruit is in front of the camera in different kinds of categories [12]. Visual feedback for robot harvester can be monocular [13] or stereo vision [14]. Stereo vision is a process in which an area or image is seen in two different cameras situated beside each other, from different perspective that can be used to express certain three-dimensional (3D) information. One of the best examples of stereo vision is the human visual system, what is seen on the right eye differs slightly in position from what is seen on the left eye. Using the same principle, the images gathered by the camera, once processed, can give visual depth allowing it to map out the differences between the given points and knowing the exact location of objects in relation to itself. This study uses ZED stereo camera (www.stereolabs.com) and OpenCV library (http://opencv.org/) for image processing. The color detection algorithm used in this study is calibrated when applied for other colored fruits such as red apple. This project can be used not only in fruit harvesting but can also be used in manufacturing industry for pick and place processes.

II. SYSTEM OVERVIEW

Shown in Figure 1 is the autonomous fruit harvester with machine vision. It consists of three major parts: 1) a moving platform called kart in this study, 2) 6-DOF robotic arm, and 3) machine vision. The overall system starts with the stereo vision with green color detection algorithm to detect the leaves of a tree. The position and distance of the tree are sent to Arduino microcontroller for motor control of the kart. When the kart is near the tree, the next process will run orange color detection, send the fruit coordinates to the Arduino which control the motor of the robot arm to pick the orange fruit from the tree and place in the basket in front of the cart.

Figure 1: Autonomous Fruit Harvester with Machine Vision

III. SYSTEM DESIGN

This section discusses design of the different parts of the study namely: kart, robotic arm, end-effector, and the vision system.

A. Kart

The kart is a 4-wheel moving platform that carries the robotic arm. It has fruit basket in front of it. It has a ZED stereo camera that can measure distance (0.5 to 20m). The kart has two battery-powered wiper motors driving the front wheels. The kart is front-wheel drive where the front wheels move forward, backward and steers the kart to left or right direction depending upon the speed of each wheel.

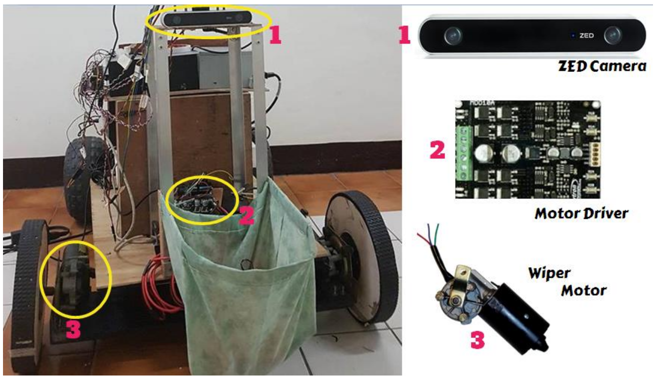


Figure 2: Kart

B. Robotic Arm

The robot arm is based from [15] has six joints at: 1) the base for the left and right, 2) the shoulder and 3) the elbow for the forward and backward, 4) the wrist pitch joint for the upward and downward motion, the 5) wrist roll joint for the rotation of the grip and 6) the joint to grip the fruit. The robotic arm uses Arduino microcontroller and has a different vision system compared to the kart. It uses a simple webcam (A4Tech) mounted on the end-effector for vision system and an ultrasonic sensor (US-100 module) for distance measurement with sensor range of 2cm to 450cm [16]. Shown in Figure 3 is the robotic arm set-up.

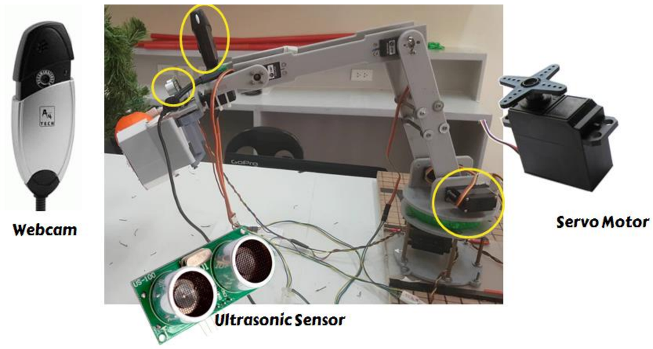


Figure 3: Robotic Arm set-up

C. End-effector

Though the end-effector is part of the robotic arm, a separate section is allotted to highlight the two kinds of end-effectors used in this study.

a. Gripper

The gripper end-effector is used for pick-and-place procedure. It is a 3D printed claw that grips the orange fruit. Shown in Figure 4 is the gripper used in this study. When the gripper is used as the end-effector, the robotic arm is controlled to put the picked fruit to the basket attached the front of the kart.

Figure 4: Gripper

b. Cutter

The cutter is another type of end-effector used in this study. It uses a pruner as the cutting element which is being drive by a DC motor. Shown in Figure 5 is the cutter end-effector used in this study. When cutter is used as an end-effector, a net-hose connected to the fruit basket is used to catch the fruit when cut by the cutter.

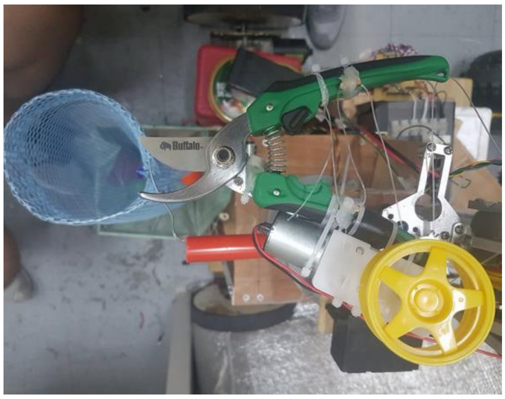


Figure 5: Cutter

D. Vision System

The vision system serves as the feedback mechanism for the whole system. Target distance and coordinates can be deduced from the vision system. In this study, two cameras are being used: ZED Stereo camera and ordinary webcam. The kart uses ZED stereo camera for long range distance measurement and tree detection while the arm uses an ordinary webcam mounted on the end-effector for orange fruit detection. Ultrasonic sensor is also position at the robotic arm end-effector to determine the distance of the fruit to the gripper. An Arduino microcontroller is the one responsible for the communication of the computer, ultrasonic sensor, motors and the robotic arm. A laptop is used to processes all the visual information. X, Y coordinates and depth measurements are sent to the microcontroller, which in turn, signals the motor when to rotate and when to stop.

The overall operation of the robotic arm can be simplified into smaller sub-operations, the tree detection and the fruit detection. Shown in Figure 6 is the flow chart of for the overall system algorithm. First, the tree is being detected using a color detection algorithm. The live video feed from the camera is processed by the algorithm which tracks and captures the shape of the tree. The moving platform moves and follows that object until it reaches a certain distance of 65cm where the ZED camera triggers a function to stop the movement of the kart motors. By using the acquired depth of an object, ZED stereo camera provides an output a distance in centimeters which marks the stopping point of the moving platform. The motors work based on the orientation of the tree. If the object is tracked at the center of the video feed, both motors will rotate making the platform move forward. The motor at the right side of the platform slows-down if the tree is detected at the right side of the video feed, which turns the platform to the right. Finally, if the object is detected at the left side of the video feed, the motor at the left side of the platform slows-down to make the platform turn to the right side. The second part of the system is the detection and picking of the orange fruit. Artificial orange fruits are placed in the artificial tree, once the camera detects the fruit, the robotic arm move the end-effector until the detected fruit is at the center of the screen.

The elbow motor and wrist motor move to place the tracked object inside the specified area of the x and y-axis. Once the target is centered, the shoulder motors move to pick the fruit and it continuously moves until the ultrasonic signals the robotic arm is close enough to the fruit to pick it off. The ultrasonic sensor signals the gripper part of the robotic arm to pick the fruit, once the fruit is properly secured, the robotic arm returns to its original position, bringing the orange fruit with it and ready to release it to a container.

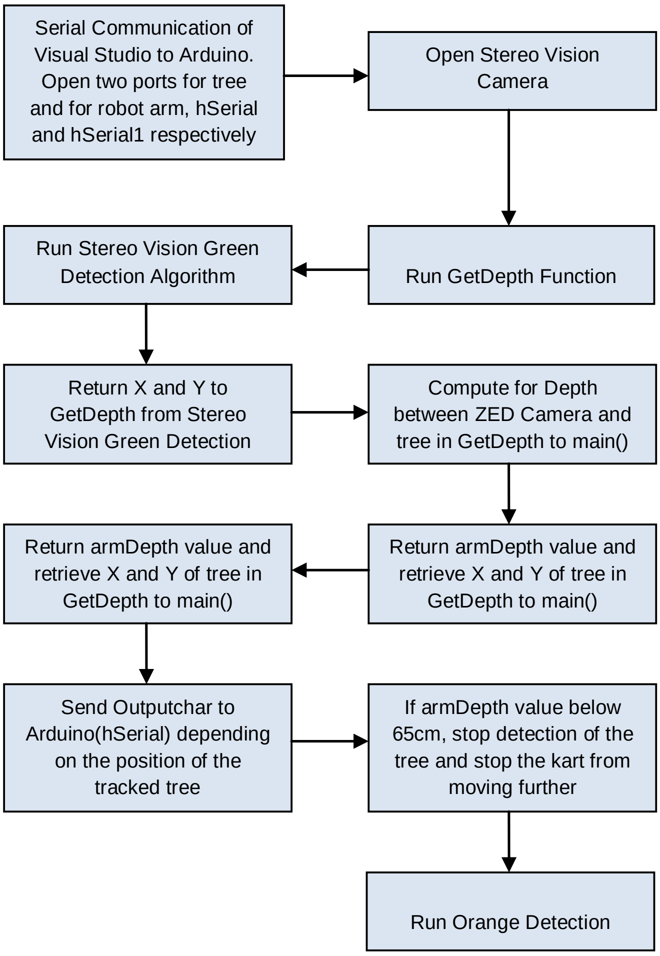


Figure 6: Flow Chart of Overall System Algorithm

a. Kart Vision System

This system is mainly composed of stereo vision camera for green color detection of tree and depth measurement. Generally, the green color detection handles the calibration of YCrCB minimum and maximum values of the necessary color. It also includes enhancement of filtered black and white image of the tree by the use of morphological operations. The tree is detected when the contour area of the tree provides a set of x and y coordinates as the center of the tree area. The kart moves to the detected position of the tree using the value of the center of the contour area. Figure 7 shows the flow chart for tree detection.

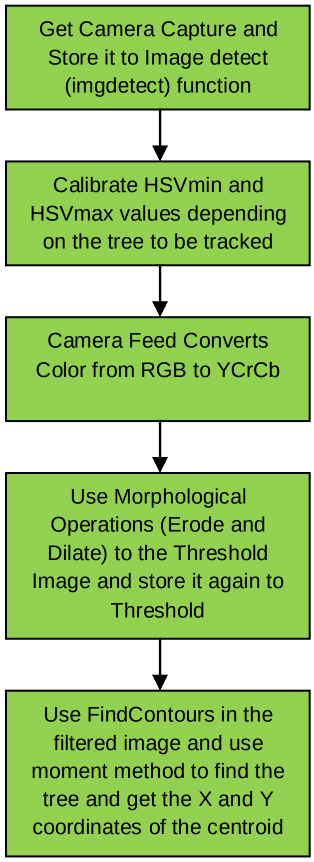


Figure 7: Green Color Detection

It is important to determine the green color of the tree by adjusting values of YCrCB. Green detection is programmed using C++ language in Microsoft Visual Studio 2010. Captured image converted from RGB to YCrCB is stored in camera feed. Mainly, the program provided trackbar to adjust the minimum, 0, and maximum, 255, values of the color of the tree. As the calibration of color is correctly filtered, the filtered tree color is stored in threshold. Morphological operations such as dilation and erosion are used to eliminate the noise background surrounded the tree. Morphological operations [17] apply a certain structure to an input image to generate an output image that still have the original size. Erosion erodes away the boundaries of the forefront pixels while dilation gradually enlarges the boundaries of regions of pixels that are in front.

b. Robot arm vision system

Figure 8 shows the flowchart of the orange detection. Orange detection was coded using C++ including OpenCV in Microsoft Visual studio. The main objective of orange detection is to show the Arduino where and what to move to place the orange in the desired location video capture. This serves as the eye of the Arduino where the orange detection sends characters which have corresponding commands in the Arduino. The orange detection creates a screen of 640x480 resolution which can be used to get the location of the x,y coordinates of the fruit, but before this, orange detection starts with video capture from the video cam which then is processed and in turn be converted from RGB to HSV. Adjusting the HSVmin and HSVmax differentiates the fruit from its surroundings. The 640x480 screen is divided into 5 divisions horizontally and vertically to make the movement of the arm more precise to how the orange is detected.

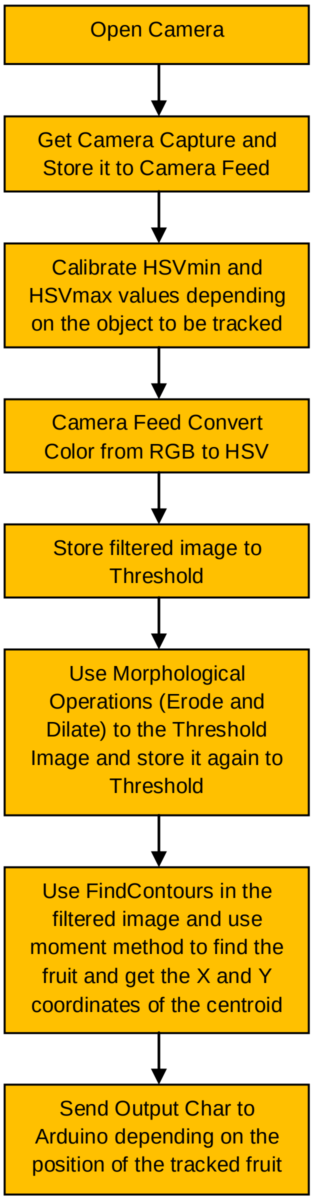


Figure 8: Orange Color Detection

E. Motor controller

a. Kart movement

As shown in Figure 9, the frame width is 1280 pixel in which the x-axis of the frame lies. The frame is divided into 5 vertical positions. The frame is divided so that each vertical position assigned into their respective ranges of x values and to easily determine the exact position of the tree using frame reference. The five vertical positions are left most, left, center, right, and right most.

When the contour area of tree in the form of white object is formed, it is where detection of object started. The contour white area of tree is formed into closely triangular in shape. Its area is needed in order to calculate and determine the center of the area of the tree using moments method. Moments can be considered a characteristic of the computed contour by summing the total pixels inside of the contour. Using the contour area of the tree, the center is determined by a point of x and y values as the position [18]. The moment method calculated the contour area and center of the detected tree.

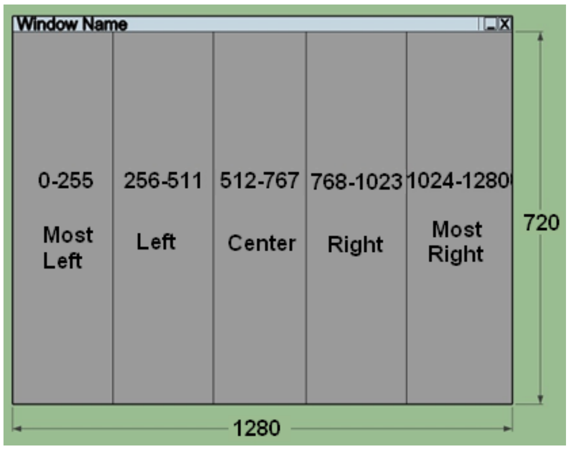


Figure 9: Frame Division using ZED Camera resolution

The movement of the kart depends on the location and position of the tree from the view of the machine vision using stereo vision camera. The direction of the movement of the kart depends on the position of the center of the contour area of the tree. The center is provided by values of x and y coordinates. To set the location of the center of the tree, the frame is divided into 5 vertical positions with its corresponding horizontal distances. The horizontal distance which lies on the x-axis has a certain range to recognize the direction from the other. Table 1 describes the 5 directions of movement of the kart with its corresponding range of x values on the frame.

Table 1
Range of X values on the frame

Position	Range of x-coordinates
Leftmost	x>=0 && x<256
Left	x>=256 && x<512
Center	x>=512 && x<768
Right	x>=768 && x<1024
Rightmost	x>=1024 && x<=1280

For a certain position of the tree on the frame, there is also an equivalent output characters to be recognized and converted to motor signals by the Arduino program. The Arduino recognized the input characters to control the direction of the motor as well as its speed. The motor driver MDD10A [19] is used to drive the wiper motors of the kart. Motor driver PWM pins are connected to analog input pins (3 and 6) of Arduino and DIR pins are connected to digital input pins (4 and 7) of Arduino. The speed of motor is written in analog signals from 0 to 255. The higher the analog signal the faster the motor rotates. In digital inputs of Arduino, HIGH signal means the motor is ON while LOW signals means the motor is OFF. Table 2 shows detected tree positions, assigned characters for each position and speed for each motor.

Table 2
Output Characters from Visual Studio for Motor Control

Position	Output Characters from Visual Studio	Speed of Motor (M1) and Motor (M2)
Leftmost	M	M1=110 & M2=145
Left	L	M1=85 & M2=105
Center	F	M1=150 & M2=155
Right	R	M1=100 & M2=90
Rightmost	S	M1=140 & M2=115

b. Robotic arm movement

The webcam adjacent to the end-effector output of the robotic arm has a 640x480 frame resolution. The frame is divided into different sections as shown in Figure 10. The goal is to move the end-effector in such a way that the fruit is at the center location of the frame. Multiple fruits can be detected but the algorithm priority follows the largest area. X and Y coordinates of the centroid and the distance measured from the ultrasonic sensor are being processed by the microcontroller for motor movements.

		Topmost		
		Top		
Leftmost	Left	Center	Right	Rightmost
		Bottom		
		Bottommost		

Figure 10: Frame Division using webcam (640x480 resolution)

The sections in Figure 10 has corresponding numerical values shown in Table 3. The video stream from the camera is being processed using Visual Studio with OpenCV library. Coordinates extracted from the processed image and distance from ultrasound are then pass to Arduino which in turn controls the different motors of the robot arm. The Arduino code consists of multiple cases in switch, which then waits for the characters sent by the orange detection. Each case corresponds to a certain command and has to meet a certain condition for the Arduino to execute. For the upward and downward movement of the robot arm, the Arduino controls the elbow servo motor, and for the left and right movement, the Arduino controls the base, in order for the fruit to be in the desired location in the camera capture. If the fruit is already in the desired position, the Arduino then controls the wrist pitch and the shoulder to be able to reach the fruit, and as soon as the fruit is near the arm, the grip then closes and returns to the original position.

Table 3
Output Characters and Range Coordinates for Robotic Arm Control

Location of the Fruit in the Screen	Output Character	Range of Coordinates
Leftmost	l	x>540
Left	z	x<=540 && x>300
Rightmost	q	x<120
Right	r	x>=120 && x<240
Topmost	f	y<50
Top	w	y>=50 && y<110
Bottom	v	y<=340 && y>170
Bottomost	c	y>340
Center	g	y>=110 && y<=170 and x<=300 && x>=240

IV. Results and Discussion

A. Kart Response and Vision System

In this study, a Christmas tree was used as a prototype tree for testing tree detection algorithm. Green color detection algorithm from Figure 7 is used to detect the leaves of the tree.

Two vision systems are used in this study: one for long range and one for short range. ZED stereo camera is used for long range for it has a useful range of 50cm to 20m. The minimum threshold set in this study is 65cm. This stereo camera is used to get the distance of the tree from the kart. This ZED stereo camera can be used not only for ranging but also for capturing images like an ordinary webcam. These captured images were processed using OpenCV and color detection algorithms. Tree detection and orange fruit detection can be done using ZED stereo camera and ordinary webcam as sensing elements. But unlike using webcam, the ZED stereo camera can output not only x and y coordinates of the target but also the depth or the distance of the target. Shown in Figure 11 is the ZED stereo camera output in the orange fruit detection testing.

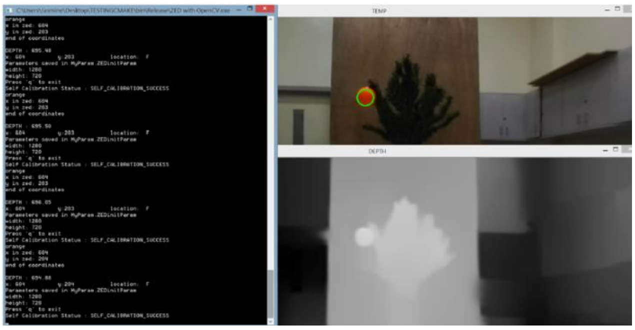


Figure 11: Orange detection and depth estimation

Figure 12 shows the ZED stereo camera output where depth in centimeters and the coordinates of target object are displayed. These data are then pass to Arduino microcontroller to be translated as motor movements.

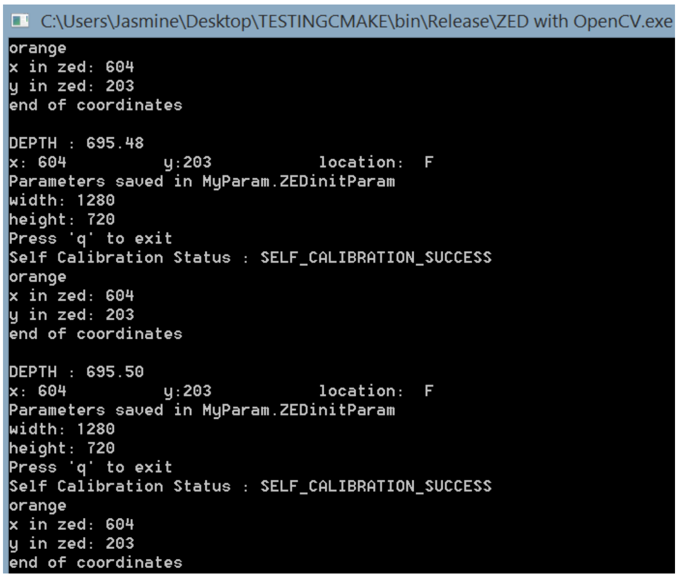


Figure 12: Output for depth estimation and coordinates extraction

It was initially thought that ZED stereo camera is enough as the visual feedback for this study. Unfortunately, it cannot work properly at very short distances of less than 65cm. That is why ZED stereo camera was only used for the kart vision system for tree detection. Eye-in hand feedback or webcam mounted on an end-effector with ultrasonic sensor for ranging is the one used for orange fruit detection and picking once the kart is near the tree (less than 65cm).

To test the reliability of the kart to move in forward motion near the tree, time traveled is measured with respect to its distance traveled to the tree. The distance of the kart from the tree is tested starting from 1m to 10m. The position of the tree varied and tested from left, right and center positions. Detected tree and its binary image is shown in Figure 13 while detected tree at different positions are shown in Figure 14.

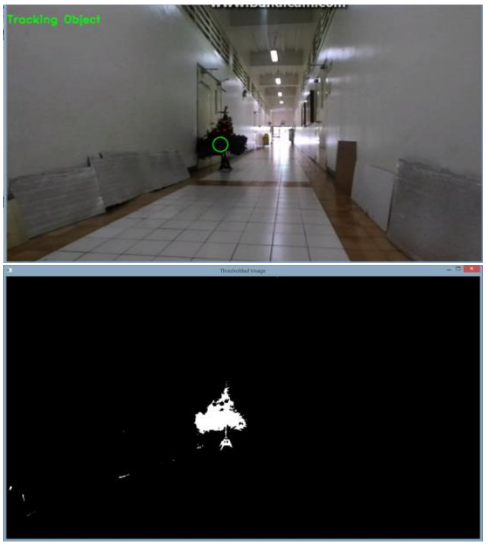
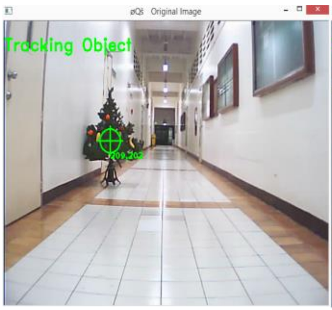
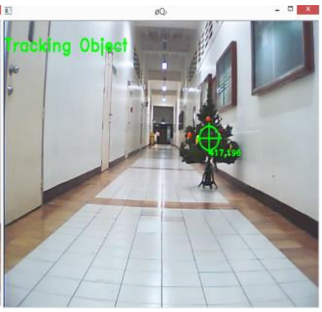


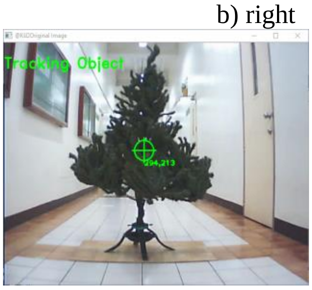
Figure 13: Tree binary image



a) left



b) right



c) center

Figure 14: Detected tree at different positions

Table 4
Summary Of Average Kart Response

Sample Distance (m)	Travel Time (s) Left	Travel Time (s) Center	Travel Time (s) Right
1	5.67	3.28	6.29*
2	8.76	7.35	9.62
3	14.86*	10.32	15.29
4	23.01*	13.71	18.72*
5	18.75	17.12	25.29
6	21.82	20.83	25.09
7	26.59*	24.72	27.49
8	29.69	29.38	34.18*
9	32.91	32.86*	36.78
10	28.86	38.75	35.29*

Table 4 shows the total average of response time for 10 trials per position. The travel time of the kart corresponds nearly in increasing pattern of distance traveled by the kart near the tree. The asterisk beside the average value on the table represents that a failed testing occurred within the 5 trials. Due to environmental factors such as noisy background, the camera was not able to track the position of the tree properly.

Shown in Figure 15 is the relationship between the sample distance versus travel time.

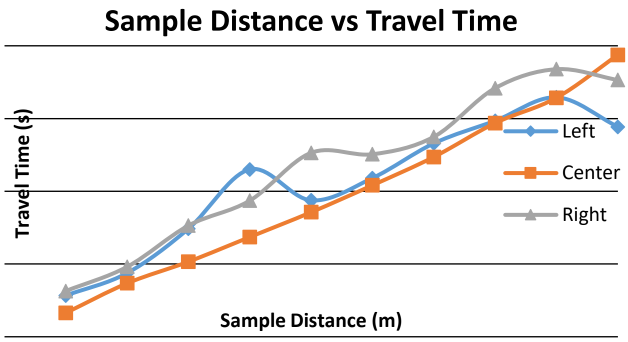


Figure 15: Relationship between sample distance vs. travel time

Shown in Figure 16 is a picture wherein multiple orange fruits are detected by the orange fruit color detection algorithm. Though we can see four orange fruits in the original image, only three have binary image because one orange fruit has a small area not enough to trigger the threshold for tracking in orange fruit detection algorithm. Once the binary image is processed, area of the white blobs, corresponding to the orange fruits, are calculated. The orange fruit largest blob detected will be the first one to be picked up and the loop will continue until all the orange fruits detected were picked up.

B. Robotic arm response

The goal of the robotic arm controller is move the end-effector with mounted camera as its feedback to detected fruit. The end-effector moves up-down and/or left-right in an incremental movements until such time that the detect orange fruit is at the middle of the camera frame. This processing can be called "centering the detected fruit" with respect to the end-effector.

The average time for the robot arm with gripper end-effector to pick and place orange fruit to basket is around 20.33 seconds. The data gathered when the end effector was changed to a cutter improved. The results shown in Table 5 shows that the harvesting rate has increased by using a mechanical cutter as the end effector. The harvesting speed of the gripper is at 20.33 seconds per fruit while 9.41 seconds per fruit using cutter.

The use of a mechanical cutter with net-hose connected to the basked helped a lot in improving the efficiency rate and speed of fruit harvesting. Overall efficiency for multiple fruit harvesting using 30 trials is shown in Figure 17. Gripper has 24/30 or 80% efficiency while cutter has 26/30 or 86.7% efficiency. Efficiency was measured in terms of success rate of putting the harvested fruit into the basket in front of the kart.

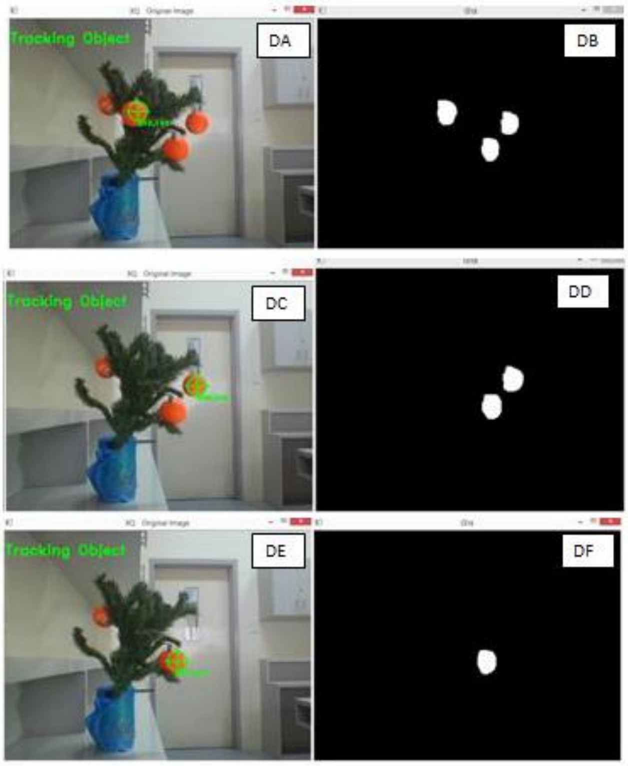


Figure 16: Multiple orange fruits detected tree at different positions

Table 5
Average harvesting time of fruit at center position

Center Position	Time
Gripper	20.33 seconds/fruit
Cutter	9.41 seconds/fruit

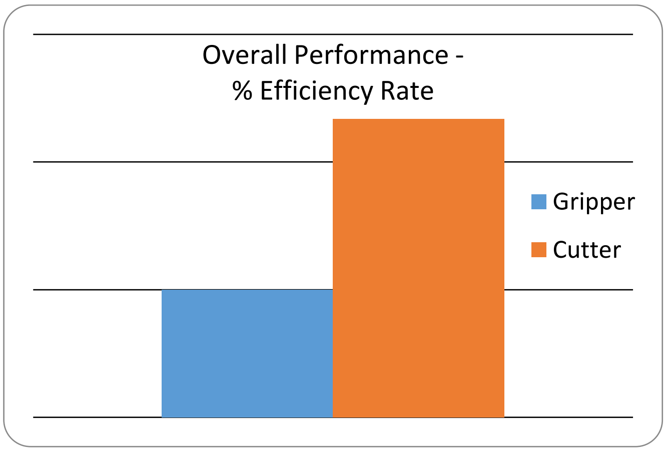


Figure 17: Overall Efficiency

V. Conclusion

This study has successfully implemented an autonomous fruit harvester with machine vision which is capable of detecting and harvesting an orange fruit from a tree. The fruit harvesting procedure was done using gripper or a cutter. The system is composed of a six-degrees of freedom (6-DOF) robotic arm mounted on a four-wheeled electric kart. The kart uses ZED camera for long range distance measurement and to detect the tree using the green detection algorithm. Image processing is done using Microsoft Visual Studio and OpenCV library. The x & y coordinates and distance of the tree are passed on to Arduino microcontroller as inputs to motor control of the wheels. When the kart is less than or equal to 65cm to the tree, the kart stops and the robotic arm with its own vision system algorithm takes over to search orange fruits from the tree. The robotic arm has a webcam and ultrasonic sensor attached at its end-effector. The webcam is used for orange fruit detection while ultrasonic sensor is used for short distance measurement to give feedback how far is the end-effector to the fruit. Multiple fruit harvesting was successfully done. Gripper has 24/30 or 80% efficiency while cutter has 26/30 or 87.5% efficiency. Efficiency was measured in terms of success rate of putting the harvest fruit to the basket in front of the kart. The color detection algorithm used in this study can be calibrated and applied for other colored fruits such as red or green apple. This project can be used not only in fruit picking but also be used in manufacturing industry for pick and place processes.

Acknowledgment

The authors would like to thank De La Salle University, their families and friends for helping them to finish this study. They would also like to thank Engr. Carlo Ochotorena and Engr. Cecille Ochotorena for helping them in purchasing the ZED stereo camera.

References

[1] S. Blackmore, W. Stout, M. Wang, B. Runov, "Robotic agriculture – the future of agricultural mechanisation? 5th European Conference on Precision Agriculture. ed. J. Stafford, V. The Netherlands, Wageningen Academic Publishers. 2005. pp: 621-628.

[2] B. Tanuja Priya, B.N.S. Murthy, S. Vasanthu, S. Jaganadh, "ROBOTICS: THE FUTURE FOR FARM MECHANIZATION", International Journal of Research in Engineering and Applied Sciences, VOLUME 5, ISSUE 5 (May, 2015).

[3] B. Sani, "The Role of Robotics at the Future of Modern Farming" , 2012 International Conference on Control, Robotics and Cybernetics (ICCRC2012).

[4] S. Hayashi, K. Ganno, Y.Ishhi and T. Ota, "Robotic harvesting system for egg plants," Japan Agril. Res. Quart, 2002, 36(3): 163 – 168.

[5] S. Hayashi, T. Ota, K. Kubota, K. Ganno and N. Konda, "Robotic harvesting technology for fruit vegetables in protected horticultural production," In: Proc. 5th symposium on information and technology for sustainable fruit and vegetable production, 12-16, Montpellier, France, 2005, Pp. 227-235.

[6] S. S. Mehta, "Vision-based control for autonomous robotic citrus harvesting", Master of Science Thesis, University of Florida, 2007.

[7] S. Bargoti and J. Underwood, "Deep Fruit Detection in Orchards", Online]. Available: https://arxiv.org/abs/1610.03677, [Accessed: April, 17, 2017].

[8] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer Vision for Fruit Harvesting Robots - state of the Art and Challenges Ahead," International Journal of Computational Vision and Robotics, vol. 3, no. 1-2, pp. 4–34, apr 2012.

[9] A. Payne, K. Walsh, P. Subedi, and D. Jarvis, "Estimating mango crop yield using image analysis using fruit at 'stone hardening' stage and

night time imaging," Computers and Electronics in Agriculture, vol. 100, pp. 160–167, jan 2014.

[10] Q. Wang, S. Nuske, M. Bergerman, and S. Singh, "Automated Crop Yield Estimation for Apple Orchards," in Experimental Robotics, ser. Springer Tracts in Advanced Robotics. Springer International Publishing, 2013, vol. 88, pp. 745–758.

[11] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, and S. Singh, "Automated visual yield estimation in vineyards," Journal of Field Robotics, vol. 31, no. 5, pp. 837–860, sep 2014.

[12] W. C. Seng and S. H. Mirisaee, "A new method for fruits recognition system," in 2009 International Conference on Electrical Engineering and Informatics, vol. 1, pp. 130–134, IEEE, 2009.

[13] J. Lv, D. Zhao, W. Ji, Y. Chen, H. Shen, "Design and Research on Vision System of Apple Harvesting Robot", Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics; Zhejiang, China. 2011; pp. 177–180.

[14] D. Font, T. Pallejà, M. Tresanchez, D. Runcan, J. Moreno, D. Martínez, J. Palacín, A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm, Sensors, 14 (2014) 11557-11579.

[15] P. C. Carando, S. A. Dela Cruz, N. J. Lorena, J. P. Zabat, and A. C. Abad, "Wireless Controlled Robotic Arm for Handling Hazardous Chemicals," 4th DLSU Innovation and Technology (DITECH) Fair, DITECH 2016, De La Salle University, Manila, Nov. 24-25, 2016.

[16] "US-100 Ultrasonic Sonar Module", [Online]. Available: http://www.e-gizmo.com/KIT/images/ultrasonicsonar/ultrasonic%20sonar%20module%201r0.pdf, [Accessed: April, 17, 2017].

[17] "Eroding and Dilating, OpenCV Tutorial", [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html [Accessed: April, 17, 2017].

[18] G. Bradski and A. Kaehler, "Learning OpenCV", OReilly Media, Inc, 2008.

[19] G. Chan, "Using MDD10A with ARDUINO UNO", Tutorials by Cytron Technologies April 5, 2015, [Online]. Available: http://tutorial.cytron.com.my/2015/04/05/using-mdd10a-with-arduino-uno/ [Accessed: April, 17, 2017].