Garden-Bot

Antonio Cantu III, tony3cantu@tamu.edu. MEEN Student
Carlos Navarro, a112233ffttt@tamu.edu. MEEN Student
Sushmitha Susan Joseph, susie_j@tamu.edu. MEEN student

Abstract
This report introduces a new type of robot prototype (Garden-Bot) that autonomously harvests red peppers in protected environments. The robotic manipulator that was chosen is a lightweight 6-DOF arm that will fit on a mobile platform. The kinematics of the robotic arm are presented using variables of the link lengths. (Garden-Bot proposes vision-based automation done through an RGB-D camera that is mounted on top of the end effector or the robotic arm. Garden-Bot acquires multiple images of the red pepper through this camera and uses it to build a complete 3D model through different techniques. To grasp and cut the red pepper, two actuators (suction cup & thermal cutter) combined into a decoupling mechanism are used. Garden-Bot starts by gathering data through its camera. Then a series of steps occur to process the data and convert it into more meaningful input. Once the input is processed, this one is used to guide the robotic arm to its target. Finally, grasping and cutting operations take place. The cost of building Garden-bot is predicted to go for about $28,900 and it is estimated to take about 5 months , for a group of   engineers to build it. A set of performance metrics are proposed in this paper to test the true performance of this prototype.

Background and Motivation
The Garden-Bot will serve to help alleviate some of the problems inherent to crop harvesting such as reducing labor costs and increasing efficiency of bell peppers in greenhouses. Bell peppers are fairly simple to grow in greenhouses, and must be nurtured manually as they begin to grow. The bell peppers are ready to harvest between 8.5 and 13 weeks after being planted, and are relatively easy to pluck from the stem [1]. An analysis of bell pepper production revealed that non-machine and irrigation laborers were paid an average of $11.98 which included benefits [2]. An average of an 8 hour work day for a month would mean a single harvester would earn $2875 for a month of work, and with an estimation of a minimum of 100 harvesters for a small farm would mean a minimum cost of $287520 to pay the harvesters. In order to help reduce the cost of labor it is proposed to design a robot that will move around the greenhouse to identify and harvest the ripe bell peppers. The robot will work by mounting the manipulator on a moving platform with a section end effector that will have a cutter attached to it.

Research Design and Methods
The robotic manipulator that has been chosen is a 6-DOF manipulator, shown in figure 1(a), being sold by Solvelight Robotics due to its light weight of 5.5 kg [3]. The manipulator is also capable of carrying a payload of up to 3 kg which will be suitable for harvesting and transport of the bell peppers and a range of approximately 0.6 m. The end effector has been chosen and based on the suction model in figure 1(b) [4]. The end effector is intended to work by having it press up against the crop and begin to suction on it which will allow it to grip onto it while the cutter severs the stem of the pepper from the rest of the plant so that it can be removed and placed in a container.

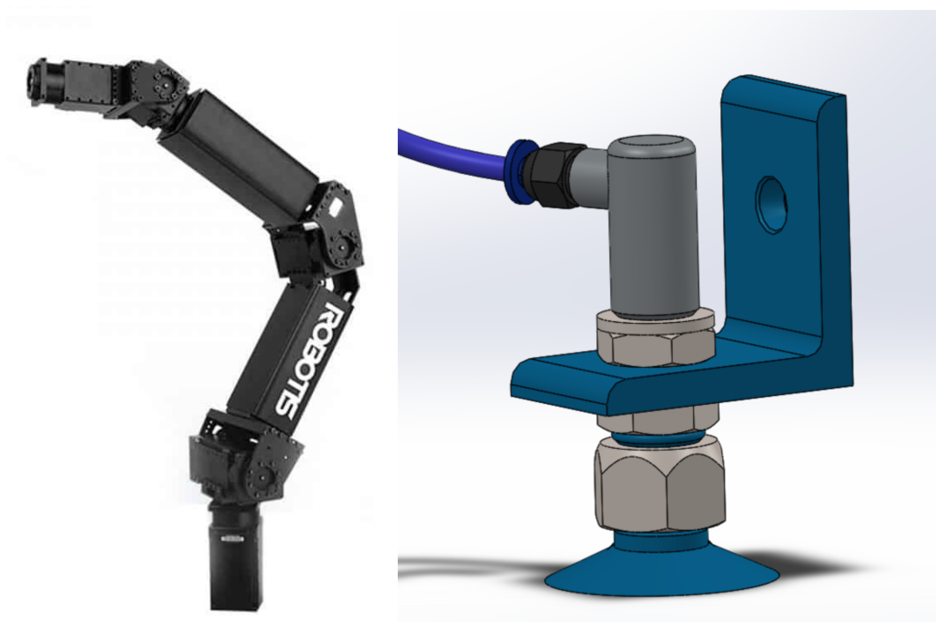

Figure 1: (a) Robot Manipulator (left), (b) End Effector (right) [3, 4]

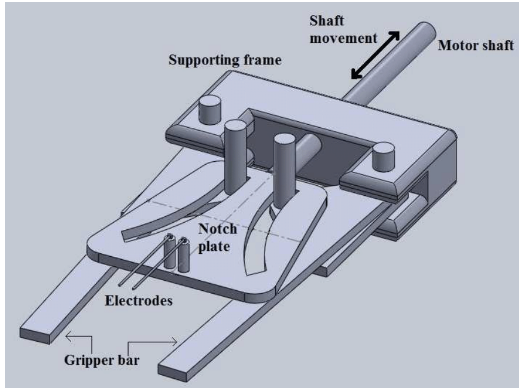

Figure 2:Selected Thermal cutter attached to the end effector of the robotic arm [10]

To better understand Garden bot's system integration sound plan, it is important to first provide some background on the sensors and actuators that are used to perform the harvesting process as well as the rationale behind each of them. It is significant to mention that all sensors and actuators employed in this project are located at the robotic arm's end effector body.

Garden-Bot's vision-based automation is done through an RGB-D (Kinect-style) camera that is mounted on top of the end effector. These cameras are great low-cost sensing systems that capture RGB images along with per-pixel depth information. Garden-Bot acquires multiple images of the red pepper through this camera from multiple viewpoints and uses it to build a complete 3D model through techniques discussed later. To grasp and cut the red pepper, two actuators combined into a decoupling mechanism are used [5]. The first actuator is a suction cup along with a vacuum hose. Its function is to grasp the red pepper firmly and hold it all throughout the cutting phase [5]. After attachment is complete, a thermal cutter removes the fruit from the stem. These two actuators create the need for a pressure sensor to detect successful attachment. (See Appendix Figure 1)

Garden-Bot starts by gathering data through its camera. Then a series of steps occur to process the data and convert it into more meaningful input. Once the input is processed, this one is used to guide the robotic arm to its target. Finally, grasping and cutting operations take place.

For sensors to perform the required perception, multiple processes and algorithms interact with one another (often simultaneously). To control the system, Garden-Bot uses a software called Robot Operating System (ROS) . Garden-bot like all robots runs on a computer. Consequently, there's an application that makes the robot run on that computer, usually interfacing with technology through libraries [7]. The system connects to a controller via USB connection. Controller then makes a digital connection to a driver, and the driver translates a digital signal into an analog signal to make an effect on the real world. Such an effect can be in the form of an actuator like the suction cup & thermal cutter; or to sense something from the real world in the case of our RGB-D camera & pressure sensor. (See Appendix Figure 2)

There are multiple reasons for choosing ROS to control this entire Garden-Bot system. ROS acts as middleware that allows different processes to communicate with each other. Garden-Bot takes advantage of all the rich libraries that ROS provides [7]. There is also a clear advantage in ROS communication ability. Applications or nodes can be taken and spread across different places or into the same computer. This is all done through one set of standard constructs that ROS exposes: nodes, parameters, services, and actions. For the reader to understand

Garden-Bot's harvesting algorithms some basic terminology must be presented. A node is a low-level computational process. Within one node you have three standard interfaces: Topics, services, and actions.

A topic is a simple messaging structure used to receive and send messages. It allows for the creation of a talker on a topic that sends the message out and a subscriber that listens to messages on that topic. That is how communications and data happen back and forth between nodes. Within that there are also services. Nodes can host services that ask a node to do something. These are typically a synchronous request so the service will block until the request is complete. Lastly a node uses actions, asking a robot to achieve a goal, move to a certain place, put yourself in a certain position. As the robot takes time to do that action it will feedback to the user and tell how well it is doing. It will tell us when it is achieved or if it's been blocked.

That is all at the lower level of ROS. At the higher-level standard applications like navigation or movement, allows for a jointed arm to move to a certain location [7]. All those applications have been exposed to ROS, so it is easy to plug in those into Garden-Bot along with the previously mentioned hardware and the system will work. The last reason is client libraries. These are the different programming languages that ROS has been exposed to. Nodes can be created within multiple programming languages. Garden-bot has nodes subscribed to topics to perform actions. ROS allows for multiple programming languages to get combined within Garden-Bot and they all talk through nodes.

Putting everything together, the system integration is based on a series of nodes (processes) interacting with one another using actions, services & topics all integrated using ROS. The following nodes conform to the algorithm.

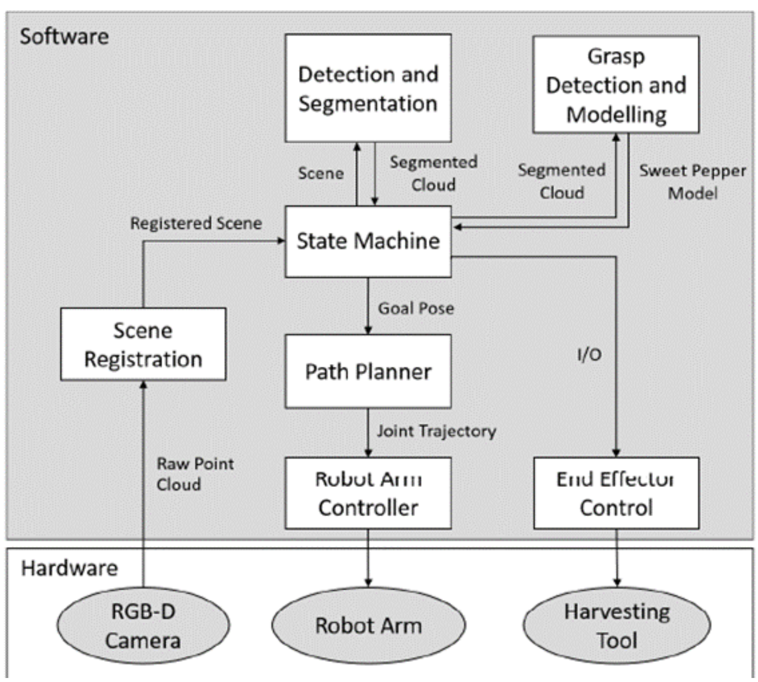

Figure 3: Garden-Bot's Harvesting Algorithm - Node's Diagram [5]

The Scene Registration Node is the first step in the perception process. It consists of moving the robotic arm in a pre-programmed scanning motion. Using the RGB-D sensor to create a 3D model of the scene. The Kinect camera uses a structured light technique to generate real-time depth maps containing discrete range measurements of the physical scene. This data can be reprojected as a set of discrete 3D points (or point cloud) [5] [8]. To create a complete 3D model, different viewpoints of the physical scene must be captured and fused into a single representation. To do such a thing, the Scene Registration node uses a software from Microsoft called KineticFusion. This package allows a user to pick up a standard Kinect camera and move rapidly within a room to reconstruct a high-quality 3D model of the scene. The system continuously tracks the 6DOF pose of the camera and fuses live depth data from the camera into a single global 3D model in real-time. As the user explores the space, new views of the physical scene are revealed, and these are fused into the same model. The reconstruction therefore grows in detail as new depth measurements are added. Holes are filled, and the model becomes complete and more refined over time. (See Appendix Figure 3)

The Detection & Segmentation Node is a service client node that detects sweet peppers within a scanned scene. This node returns a list of point cloud segments as output [5] [8]. The node scans a specific red pepper rather than the entire scene. KinectFusion allows a user to first reconstruct the entire scene, and then accurately segment the desired object by moving it physically. The system continuously monitors the 3D reconstruction and observes changes over time. If an object is physically removed from view or moved within the scene by the user, rapid and large changes in the 3D model are observed. Such changes are detected in real-time, allowing the repositioned red pepper to be cleanly segmented from the background model. This approach allows a user to perform segmentation rapidly. (See Appendix Figure 4)

Before going further into the algorithm, it is important to introduce the concept of State Machines. These “machines” are just groups of nodes, in which each node represents the state of the system [6]. Edges represent transitions in between states, e.g., going from state 3 to state 2. The labels in these edges, represent events and actions. The system goes from state 3 to state 2, e.g., when event 5 happens. When going from state 3 to state 2 the system generates action 4 for example. Segmentation yields two concurrent state machines.

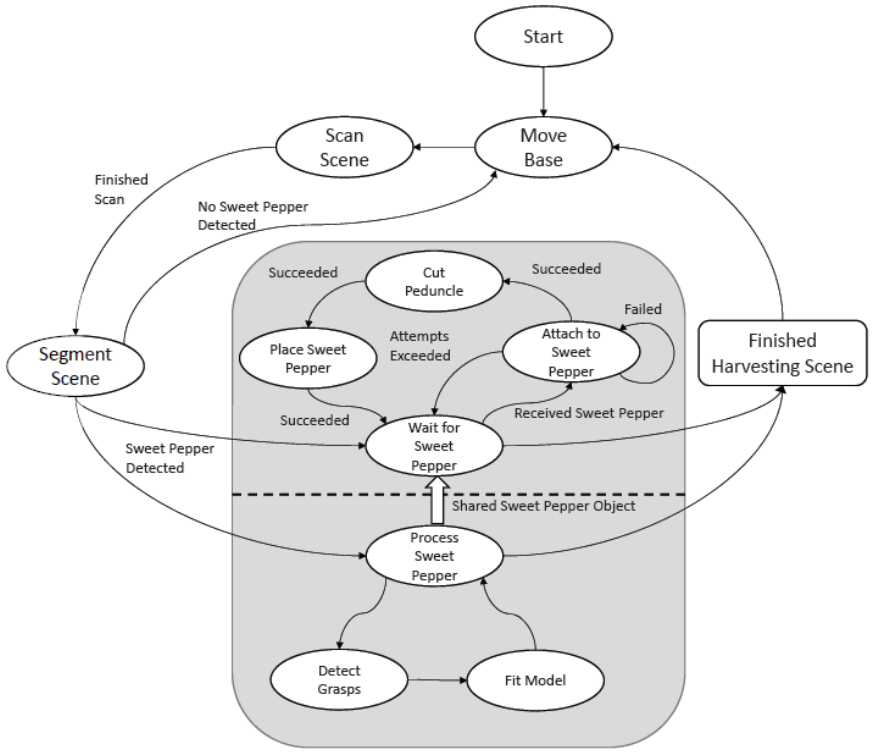

Figure 3: Garden-Bot's State Machines [5]

The first state machine (lower dotted line region Figure 3) makes use of the now segmented red pepper model to calculate candidate grasping and cutting poses [8] [5]. Grasp poses for each sweet pepper are calculated using the segmented 3D point cloud of a sweet pepper. A desirable grasp pose could be placing the suction cup squarely on the planar side of the pepper. A way to do this is by fitting a geometric model (superellipsoid) to the point cloud to estimate the sweet peppers size, position, and orientation [5]. The grasp pose is calculated to be in the center of the front face of the pepper, while the cutting pose is calculated to be offset from the top face of the pepper. (See Appendix Figure 6)

The second state machine (upper dotted line region - Figure 3) makes use of the same segmented red pepper model that was fed to the previous state machine [7] [5]. It uses it to plan and execute the real physical harvesting. The state machine employs a path planner, robot arm controller and an end effector controller subsystem to perform the physical harvesting. Harvesting trajectories are calculated relative to the grasping and cutting poses described previously. The attachment trajectory starts at a fixed offset back from the grasping pose and moves the suction cup along the selected approach axis. Once the attachment trajectory has been executed to attach the suction cup, the end effector is moved vertically from the attachment pose to decouple the suction cup from the cutting tool. Lastly, a cutting trajectory is computed. This trajectory is calculated to keep the end effector aligned with the horizontal world frame. Failure to attach the suction is detected via an in-line pressure sensor on the vacuum system and triggers a reattachment at the next-best grasp candidate. After a fixed number of failed attachment attempts the state machine moves on to harvesting the next sweet pepper.

Garden-Bot takes advantage of a free-space motion planning library developed by ROS, called MoveIt. This tool is used for planning motions between points in space [9]. MoveIt's hand-eye calibration determines the transform between the camera and the end-effector. It uses the segmented red pepper 3D model for optimizing the calibration parameters. The software can quickly choose grasp poses and excels at pick and place applications of different size and orientation.

Evaluation Plans

Forward Kinematics

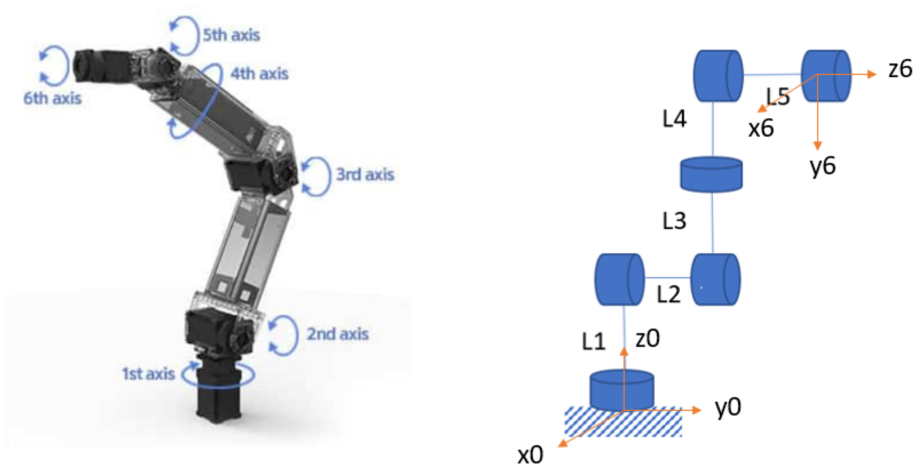

Figure 4: Axes, Joints and links

$\omega1 = \omega4 = [\,0\ 0\ 1\,]^T$

$\omega2 = \omega3 = \omega5 = \omega6 = [\,0\ 1\ 0\,]^T$

For home configuration, $\theta = 0$ for all joints,

$$T_{60} = \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{matrix}$$

$P2 = [\,0\ \ 0\ \ L1\,]^T$

$P3 = [\,0\quad L2\quad L1\,]^T$

$P4 = [\,L3\quad L2\quad L1\,]^T$

$P5 = [\,L3+L4\quad L2\quad L1\,]^T$

$P6 = [\,L3+L4\quad L2+L5\quad L1\,]^T$

Revolute	Joints					
ζ	J1	J2	J3	J4	J5	J6
(-) ω x p	0	(-)L1	L2 - L1	L2	(-)L1	(-)L1
	0	0	0	(-)L3	0	0
	0	0	0	0	L3 + L4	L3 + L4
ω	0	0	0	0	0	0
	0	1	1	0	1	1
	1	0	0	1	0	0

ζ = (-) ω x p ω

0 0

$T_{60}(\theta) = e^{\zeta\theta 1} e^{\zeta\theta 2} e^{\zeta\theta 3} e^{\zeta\theta 4} e^{\zeta\theta 5} e^{\zeta\theta 6}$

The inverse kinematics and dynamics are similarly straightforward in their derivation

Testing & Experimental Results:

To test the actual prototype, the following experiment could give some interesting insights of our prototype's performance. Garden-Bot would be placed in a controlled environment where harvesting could occur under optimal environmental conditions. In such an environment there will be a fixed number of pepper plants, with a fixed known number of peppers. Some ready to get collected, and some not fully red and therefore not ready to be picked by our robot. The experiment would consist in running several harvesting trials. Each trial starts with the same number of fixed initial available peppers.

After several trials, the following information would be recorded. The initial number of peppers used in all trials. The number of ready to harvest red peppers that were not picked up due to attachment failure. The number of ready to harvest red peppers that were not picked up due to detachment failure. The total sum of ready to harvest red peppers that were not picked up. The total number of not ready to harvest peppers (not red enough) that were picked up. The total number of red peppers that were picked up and got damaged. Using the previous data, the following metrics could give a good taste of Garden-Bot's Harvesting performance more in depth: overall success rate, attachment success rate, detachment success rate, handling success rate, overall scanning success rate, and scanning the ready-to pick peppers success rate. These metrics indicate the areas where the robot is the strongest, and the ones where more work and research must be conducted to make improvements.

Budget and Timeline
Estimate:
Manipulator cost : $18,900
Design cost - $5,000
Manufacturing cost - $5,000

Deliverable	Description	Timeline
Feasibility Study		2 months
CAD	Modeling of components & Assembly	2 Weeks
Circuit Development & Programming	Hardware and Software	2 Weeks
FEA & Virtual Testing	Test for Balance, Resolution, Feedback Logic	2 Weeks
Bill of Materials	Parts list for Budget	2 Weeks
Production and Testing		3 months
Assembly	Permanent (welding) and Fastening of components	1 month
Physical Testing	Testing with test masses	1 month
Final Adjustments and Tweaking	If necessary	1 month

Timeline does not include part lead times.

References Cited

[1] “Greenhouse Growing Guide: Peppers.” *Www.btlliners.com*, www.btlliners.com/greenhouse-growing-guide-peppers. Accessed 6 Dec. 2022.

[2]*COSTS and PROFITABILITY ANALYSIS for BELL PEPPER PRODUCTION in the OXNARD PLAIN, VENTURA COUNTY, 2012-13 Bell Pepper Production for Processing*.

[3] “ROBOTIS MANIPULATOR-H ROBOTIC MANIPULATOR.” *Solvelight Robotics*, solvelight.com/product/robotis-manipulator-h-robotic-manipulator/. Accessed 6 Dec. 2022.

[4] “Free CAD Designs, Files & 3D Models | the GrabCAD Community Library.” *Grabcad.com*, grabcad.com/library/suction-cup-ventouse-1. Accessed 6 Dec. 2022.

[5] Christopher L., Andrew E., Christopher Mc., Adam W., and Tristan P., “Autonomous Sweet Pepper Harvesting for Protected Cropping Systems” in IEEE Robotics and Automation Letters, 2017.

[6] Udacity. “Finite State Machines - Georgia Tech - Software Development Process.” [Video file]. YouTube, 23 Feb. 2015, https://www.youtube.com/watch?v=2xbWwk4-zTs.

[7] ROS. “An Introduction to ROS, the Robot Operating System: Intro to ROS (2/6)” [Video file]. YouTube, May. 2020, https://www.youtube.com/watch?v=96XsJ7xfsS8.

[8] C. Lehnert, I. Sa, C. Mccool, B. Upcroft, and T. Perez, “Sweet Pepper Pose Detection and Grasping for Automated Crop Harvesting,” in IEEE International Conference on Robotics and Automation (ICRA), 2016.

[9] ROS. “MoveIt Capabilities Overview” [Video file]. YouTube October 13). Retrieved December 4, 2022, from https://www.youtube.com/watch?v=7KvF7Dj7bz0&t=1s

[10] Shivaji B., Koichi O., “Performance Testing of Thermal Cutting Systems for Sweet Pepper Harvesting Robot in Greenhouse Horticulture” Journal of System Design and Dynamics, 2013.

Appendix

Figure 1: Robot's end effector with RGB-D camera mounted [8]

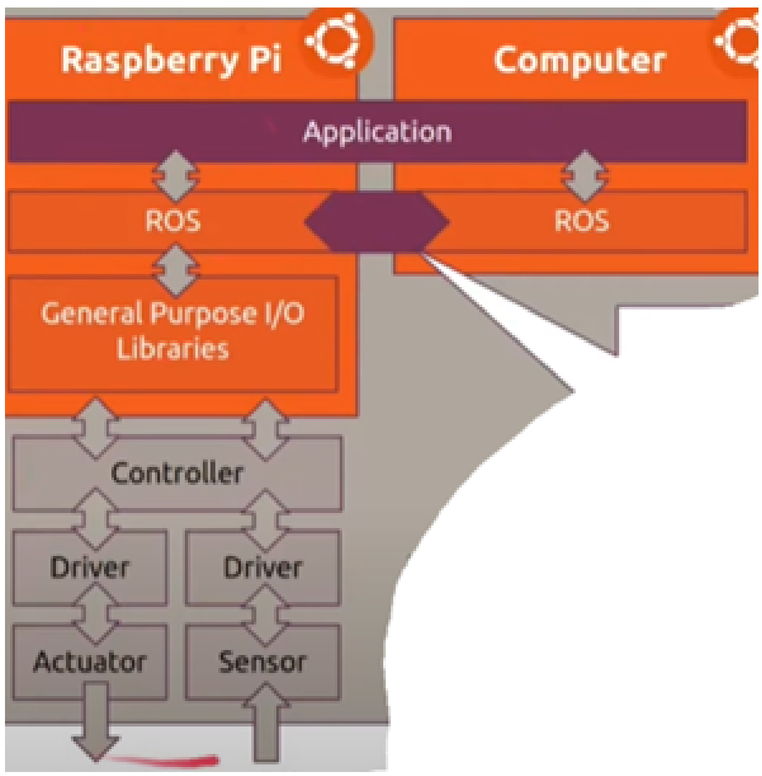

Figure 2: ROS Implementation Diagram [7]

Figure 3: Scanning & Unsegmented 3D Model [5]

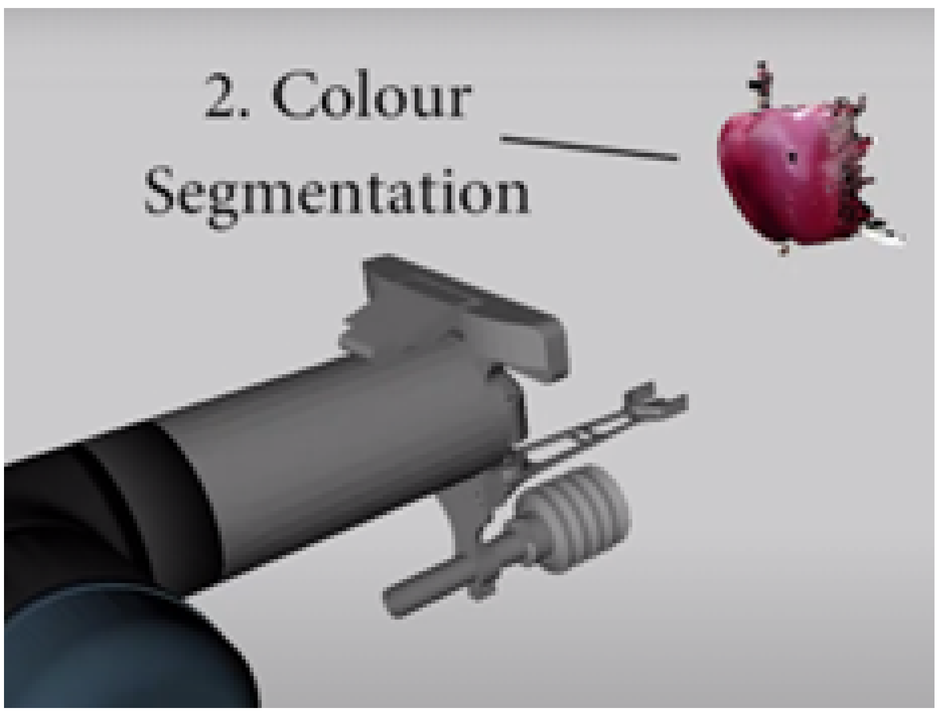

Figure 4: Segmentation of the 3D Model [5]

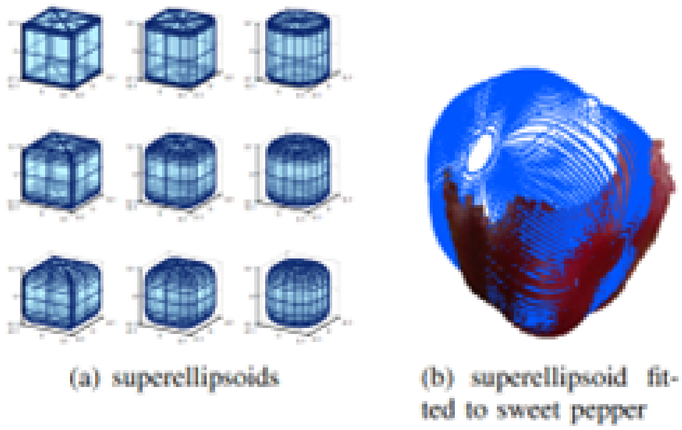

Figure 5: The range of super ellipsoids which are used to fit to sweet peppers [8]

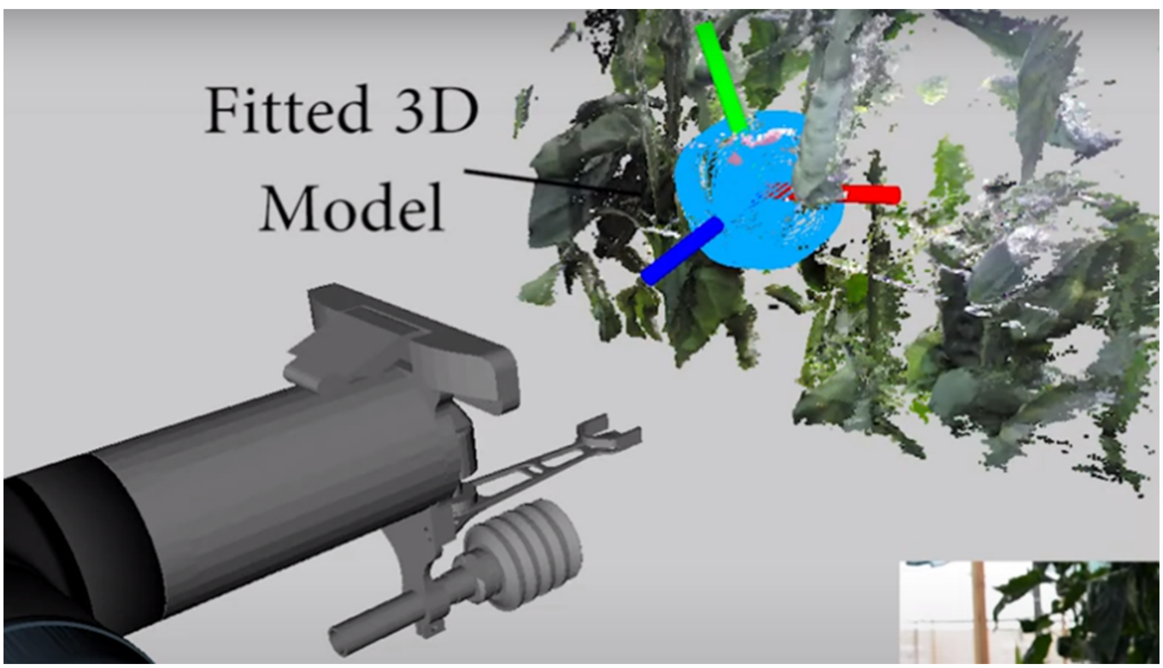

Figure 7: Super ellipsoid fitted to the 3D model and cutting and grasping poses identified [5]