Easy Debris Collector Design Report

Andy Zhang, Hojun Byun, Omar Alinur
Electrical and Computer Engineering, Carnegie Mellon University
{andyz, hojunb, oka} @ andrew.cmu.edu

Abstract
As people get older, their motor skills and their ability to detect objects on the floor become worse and worse. This can lead to potential safety hazards as people can trip on certain items that fall on the floor that either go ignored or cannot be picked up by those with poor motor skills.

Index Terms
OpenCV, Universal Gripper, Rev Robotics,

I. INTRODUCTION

People with dexterity problems have difficulties in picking up items off the floor. If left ignored, the these objects can become problems for the following reasons:

1. People can harm themselves by stepping on or tripping over the items.
2. Picking up these items can be time-consuming, tedious, and labor intensive.
3. Based on different lighting environments, these items can be difficult to detect with the human eye.

Our project aims to solve the problem by creating an easy-to-control robot that would pick up and collect these items that fall on the floor. Named Easy Debris Collector, our robot will eliminate the potential for people to hurt themselves, waste time on picking up after themselves, and miss picking up all the fallen objects due to their poor eyesight.

With Easy Debris Collector (EDC for short) the problem will become nothing but an afterthought as we aim to construct a robot that can quickly and easily collect small objects that have been marked with a distinct color at a rate of 2 objectss per minute while having a 20 minute battery life.

II. DESIGN REQUIREMENTS

In order to achieve reliable small item retrieval in a timely and reasonable manner, we have set these following high-level requirements:

1. Reliable item pickup - The robot must achieve at least a 90% success rate when attempting to pick up items.
2. Item size - The scope of our project is aimed at items that are less than 0.7kg and 10cm in length so that the gripper has enough surface area and strength to effectively grasp the items.
3. Item Collection - The number of items collected should not exceed 10 so that there is enough room in our basket to carry the items without risk of them falling out or causing too much strain on the chassis and motors.
4. Operation time - The robot must have at least 20 minutes of operation time to ensure collection of the dropped objects in a workshop of moderate size (84 square meters) without the robot running out of power
5. Robot weight - The robot must weigh less than 10kg so as to not put too much strain on the wheel motors and cause the run time to be shortened due to the increased torque required by the motors
6. Robot speed - The robot must be able to travel at 0.4 m/s in order to efficiently search the workshop and find the dropped items before the robot base runs out of power.
7. Arm control accuracy - The robot's arm must have an arc length accuracy of at least 5cm so that the arm's position relative to the item and basket can be accurate enough to reliable pick up and drop the item in the basket
8. Image processing latency - The robot must have a maximum of 10ms in latency between each item localization timestep so

as to ensure proper navigation towards the item

9. Image processing reliability - The robot must have a 0% false positive rate and less than 7% false negative rate in order to ensure the correct items are being tracked with some wiggle room in the case of noisy lighting conditions

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The architecture of EDC can be divided into three major parts: the software components to perform computer vision, the hardware components doing the physical act of picking up the objects, and the user input. This idea is represented by figure 2 on page 10.

The software is handled by the Rev Robotics Control Hub, which is responsible for performing the computer vision, and being the hub of the robot that transmits information to the user and obtains user's control inputs.

The process begins with the Logitech C270 camera, where images are captured and sent to an android phone on the robot that acts as a transmitter as well as the computer responsible for image processing. Our image processing algorithm obtains the coordinates of the center of every object that is detected with a neon pink tape. The android phone does this by applying a gaussian blur, then converting the image to an HSV format, then running a color thresholding algorithm, then finally finding the contours and centroids of each object. After that the phone places a bounding box around each detected object and this processed image is then transmitted via the same phone to the user's laptop. This allows the user to see what the robot sees while being able to clearly identify which objects are marked with neon pink tape, indicating that these objects are the ones that need to be collected.

Once the user identifies that they need to steer the robot closer to the detected objects that they see on their user interface on the laptop, they use a logitech controller that sends commands to the robot via another phone that is connected to the controller. This phone transmits user inputs to the phone on the robot so the Rev Robotics Control Hub is able to control the two motors for the wheels, two servos for the arm, and a 2.5 LPM DC air pump for the gripper. The two servos for the arm and the air pump are hardware components necessary to execute the gripping protocol.

To start, the Rev Motor would rotate our single-jointed robot arm downwards to deform the rubber pouch around the object. Afterwards, the Rev Robotics Control Hub would command the air pump to apply suction to the pouch, hardening it around the object and command the Rev Motor to rotate the arm upwards. Then the second servo, named Hitech Servo, would rotate the wrist of the robot which would in-turn rotate the gripper so that the grabbed object is hovering over the storage bin at the back of the robot.

IV. DESIGN TRADE STUDIES

After going through several design iterations, we have weighed the benefits and detriments of each design choice to be made to come to our final design.

Hardware

A. *Robot Base*

Our early plan for the robot base until the midpoint of the project was to use the iRobot Create 4400. This base would have let the group use an existing base instead of creating one fresh. But we ran into weeks and weeks of trouble trying to get the base to work with no results. So we decided to pivot and create a base of our own. This gave us more mechanical control over the parameters of the robot and created our own control system.

The base is created out of laser cut wood square platform that is reinforced with metal extrudes. This not only allowed us to customize the size and shape of the robot, it allowed for easy installation of the Rev Robotics Control Hub, the Rev servos, the arm/gripper, and all the other components.

B. *Gripping Mechanism*

Originally, we wanted to go with a two finger gripping mechanism but realized that that would require accurate positioning and unobvious point of contact determination. This led us to a universal gripper design which is more robust and simpler. The suction-based universal gripper that we decided on is well tested and can achieve a firm grip on a wide range of shapes and sizes.

The gripping mechanism for our robot needs to be able to grab items of different shapes and sizes. This is why we decided to go with a universal gripper design as shown in Figure 1. The blue component is a flexible rubber pouch that is attached to a funnel and would be filled with finely-ground material. The funnel would have an electric pump/vacuum attached to the back end of it.

In order to grab an object, the gripper will:

1. Deform the rubber pouch around the object.
2. Use an electric pump/vacuum to suck the air out of the pouch, hardening the rubber around the desired object.

After these two steps, the desired object would be firmly embedded into the pouch, allowing easy manipulation for storage.

In order to release a grasped object, the gripper will simply pump air back into the pouch, returning the pouch to its original shape to release its hold on the object.

For construction, we need a rubber pouch, finely ground materials, a funnel, an electric air pump and vacuum, and a piece of cloth to make sure the pouch is sealed.

Figure 1: Side view of gripper and camera

C. *Camera*

For our camera, we decided to use a simple RGB camera as we realized that we only need to distinguish the color objects for our image processing algorithm. We chose the Logitech C270 as it came with an adjustable mounting arm that gave us the option to change the angle of the camera if needed during testing.

D. *Robot Arm*

To rotate the arm that controls the gripper, we decided on an affordable torquey servo rated for 20kg and the Rev Motor which have enough power and accuracy to meet our requirements. The 20kg servo would be responsible for turning the gripper horizontally to enable the gripped object to fall into the bin and the Rev Motor would be responsible for rotating the arm downwards to make the gripper come into contact with the desired object. We also decided to have the motion for gripping and storing the objects be as simple as possible. To accomplish this, the robot's arm is a single joint that would rotate about two axes of motion: one to move straight down to pick up the object in front of the robot and the other to rotate the arm to face the storage bin that is integrated into the base that is located next to the arm to release the object once the arm is rotated.

E. *Computer*

For the robot's computer, we decided to use the Rev Robotics Control Hub since it has the capability to use an android phone to enable a laptop to connect to the robot. This is important as the user interface is on the laptop, so the robot can send the output of its image processing to the user. The image processing is done on the same phone. We initially planned on using the Nvidia Jetson AGX Xavier as it has ample processing power to support the computer vision. But as we had to shift our plans mid-way through the semester, we had to change up our design.

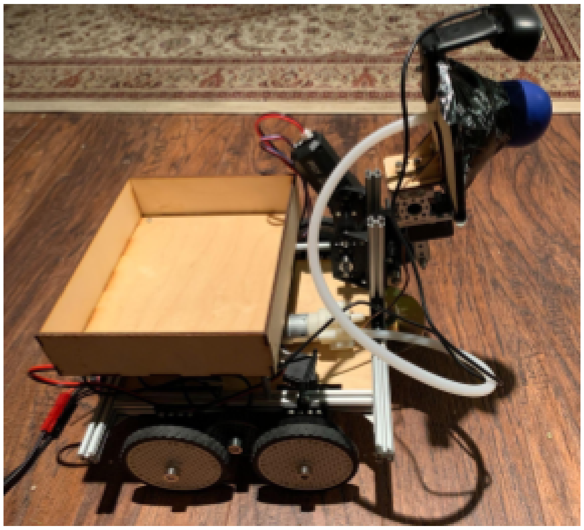

Figure 2: Side view of the completed robot

Software

A. Arm pick-up protocol

Our initial protocol was to have a two-jointed arm that would have a diverse range of motions to pick up different objects of different sizes and shapes. But we realized that this would be very complicated for us to implement with our given time as none of us had programmed 2D reverse kinematics motion for a robot arm before. So, we decided on the simplest design possible of having a single-jointed arm that would rotate about two different axes of rotation, one for picking up objects and one for rotating the arm towards the storage bin.

V. SYSTEM DESCRIPTION

Our system is divided into two major components: the physical system responsible for grabbing desired objects, and the software component that is responsible for item detection and motion planning.

A. Camera

Our team looked for several different cameras during the design and planning process. We initially wanted to use a depth camera to indicate whether a detected object is within reach for the robot to grab. Later on, however, the team realized that a normal RGB camera would suffice and so we decided to use the Logitech C270. This model came with an adjustable mounting arm that made testing easier as we could change up the field of view of the robot easily and did not compromise too heavily on the resolution as the camera came with HD 720p with 30fps.

B. Image Processing

Our computer vision software is powered by OpenCV which has the necessary libraries to recognize and localize the marked items while having plenty of documentation as to how the libraries can be used. The items will be marked with a distinct neon pink tape as we feel that this color does not appear frequently in everyday household items. To localize the marked objects we will perform the following procedure: Gaussian blur the image for smoother image processing, transform the image to HSV to make the markers stand out, perform thresholding to remove everything in the image but the markers, find the contours around the markers, find the centroids of the contours, and finally add the markers' centroid positions to a list to be used for path planning. In order to differentiate the items, we will store their location in the terms of the global frame. All of the described procedures are accomplished through OpenCV functions.

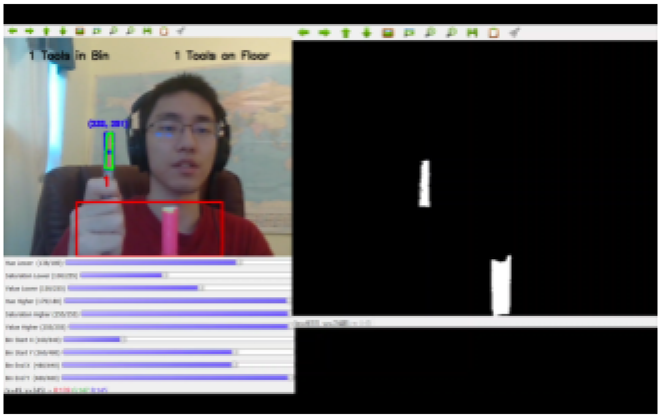

Figure 3: Computer vision component showing which marked objects are in the storage bin

C. User Interface

To allow a user to better operate the robot, track its position, and tune the computer vision, we created a graphical user interface. This interface displays the live camera feed along with a bounding box around the colored objects that the user would like to pick up. This lets the driver better steer towards and grab the desired objects. Using the IMU in the Rev Control Hub, wheel odometry, and the motion model of the robot, we are able to provide the user with the robots x and y coordinates along with its heading relative to its starting state. This was done so that the user can better keep track of the robot's state.

D. Gripping Protocol

When picking up detected objects, the arm would run through the following pick-up protocol everytime:

1. A motor would rotate the arm down so the gripper deforms around the item.
2. Air is suctioned out of the pouch to harden it.
3. Arm rotates back up using the previous motor
4. With the second motor the arm's base rotates towards the storage bin.
5. Air is pumped back into the pouch to reshape it, releasing the object into the storage bin.
6. Using the second motor the arm's base rotates to its default position.

This protocol allows us to build a very simple system for the arm.

VI. TEST AND VALIDATION

We decided to test our debris collector in a home environment, which included carpeted floors and wooden floorboards. The validation of our software and hardware requirements in this environment should give us good estimates of how our collector would perform in a similar real life setting,

- **Hardware Testing and Validation**
  - *Reliable Item Pickup*
    - Due to the large variety of possible items that could be found on the ground in a workshop or home environment, we decided to employ the usage of a air pump gripper as our robot's appendage to collect debris. Therefore, we needed to ensure our universal gripper was reliable in picking up items off of the ground. In our initial design requirements, we stipulated that our gripper should have at least a 90% success rate in picking up averaged small-sized items, or items that were below 0.7 kg in weight and 10 cm in length.
    - We tested our gripper on over 30 different types of items that could be found on the floor, which included pens, discs, and AirPods. We found that our gripper had above a 95% success rate in picking up all these types of items, which met our initial requirement. We also noticed that the gripper mostly had no problems in gripping and picking up solid items with a definite weight and

shape like a pen but had trouble with items that were lighter, larger in surface area, and more flimsy like flat pieces of paper or plastic bags.

- *Sufficient Item Collection Capacity*
  - We also wanted our robot's item collection bin to have enough space to contain the items it collected during one entire operational trip. Our initial design requirements stated that our bin should have enough capacity to hold at least 15 common small-sized items that were below 0.7 kg in weight.
  - During testing, we simply put all our gripper test items into the bin and saw that we were able to comfortably hold 25-30 0.7 kg items reliably and securely without a hitch.
- *Ample Robot Speed*
  - We wanted our robot collector to be fast enough to traverse the house/workshop in a reasonable time and manner. For our initial design, we set our robot's speed goal as being able to cover a 2 meter dash within 5 seconds.
  - During testing, we were able to control the robot and consistently complete a 2 meter dash in 3.05 seconds on average over 10 different tests.
- *Adequate Battery Life*
  - As our robot would be running on a single battery and controlled by a user during its operation, we needed the robot to last long enough in the field to provide a satisfactory experience necessary to complete its tasks. As such, we specified our battery life/duration requirement to be at least 30 minutes of operation time on one full charge.
  - During testing, we fully ran the robot three times on a full charge and we saw that on average our robot was able to continuously operate for around 37 minutes on one full battery charge.

- **Software Testing and Validation**
  - *Fast Image Processing*
    - We wanted our HSV color thresholding algorithm to be fast enough to recognize color-marked items on the ground in real-time for our user to pinpoint on the interface. In our initial design, we wanted our image processing algorithm's runtime to be at most 10 milliseconds per loop.
    - In our testing, we found that over 3000 loops/frames, our image processing algorithm was only able to reach 33.48 milliseconds on average, which was greater than our initial goal. After careful reconsideration, we found that this runtime was still acceptable, as it still permitted able real-time operation even though it exceeded our initial lofty expectations.
  - *Reliable Image Processing*
    - For our actual image processing, we wanted the OpenCV HSV color thresholding to achieve a 0% false positive rate, where the robot should not detect that an object without the neon pink tape marker as an item that it should pick up, and below 7% false negative rate, where the robot should not ignore an object that has the neon pink tape marker when it comes into view of the camera.
    - For these criteria, we tested on two subcases: one where the room is under good lighting conditions and the other where the room is under poor lighting conditions.
    - In the room with bright lighting conditions, we tested on 20 pink taped-marked items scattered over

the floor over 15 times. Each test iteration, we moved the items around, flipped them to catch the light at different angles, and adjusted the lighting. We found that over these 300 item tests (15 iterations of testing with 20 items in each testing iteration), we detected a red item (without tape) 3 out of 300 times and failed to detect a pink tape-wrapped item 2 out of 300 times. With this, we successfully achieved our initial goals with nearly a 0% false positive and 0% false negative rate.

- We simulated a room with poor lighting by changing several environmental conditions. First, we tried adjusting the brightness level of the room by only using 1-4 light bulbs to light up the room instead of the 5 light bulbs that would constitute “brightly lit”. We also experimented with placing items underneath shade, by putting them underneath tables, coverings, and putting objects in front of the light source. We found that in 300 tests (15 iterations with 20 items each), due to the constantly uneven lighting and shade, our algorithm mistakenly marked and detected a red item 23 out of 300 times and failed to detect a pink tape-marked items 11 out of 300 times. This resulted in a sub 8% false positive rate and sub 3% false negative rate, which was much higher than our bright lighting tests. However, despite the false positive rate being higher than our initial goal, we found that it was still very reasonable for our use case in unfavorable conditions. Furthermore, our intended environment is a well-lit area, where we concretely met our false positive and false negative thresholds.

VII. PROJECT MANAGEMENT

A. Schedule

We have set 3 phases for our project:

1. Project Proposal and Planning
2. Learning, Design, and Implementation
3. Performance Testing and Refinement

The first phase is already completed and during it, we brainstormed and honed in on what our project should look like at the end of the term. We are currently in the second phase where we are researching and learning about the specific details needed to complete the project. In the next month and a half, we plan to implement the individual features and integrate them into the robot. For our third phase, we will test and tune our system.

B. Team Member Responsibilities

Each team member is responsible for keeping each other accountable on their individual tasks. We have split up our work based on our previous project experiences as well as who is physically present in Pittsburgh or not, and there are three major components to the project: robot construction and motion control, computer vision, and localization and path planning. These tasks and their subtasks are divided up amongst the three members as shown below.

Team Member	Primary Responsibility	Secondary Responsibility
Hojun	Construct the arm/gripper and integrate the GPU/microcontroller with the robot base	Testing of gripper’s reliability, robot’s travel speed, item collection capabilities.
Omar	Motion model and sensor fusion for localization and path planning	Simulation, integration, and testing of each software component on the robot
Andy	Computer vision algorithm to detect objects marked with distinct colors	Help test item/color detection performance and lighting variability

C. Budget

The bill of materials is included in page 11 in the Appendix.

Of the hardware materials we required initially before switching up our design to our own custom robot base using Rev Robotics Control Hub, we have obtained the Nvidia Jetson AGX Xavier and the iRobot Create 4400 robot base for free from the 18-500 course parts list.

D. Risk Management

During the course of our project, we have encountered several potential risks for the design and development of our robot.

Firstly, we considered the wide assortment of items that could be present in one's home or workplace that fall onto the floor. Eating utensils, writing utensils, any home-found objects have very different weights and shapes. Therefore, we needed to ensure that our gripper could tackle all these types of items. To do so, we conducted gripper tests on various objects of varying shapes and sizes and in different positions on the ground. We used an air pump to power our gripper and tested rubber, nylon, and latex as the covering material. We eventually settled on rubber because it was able to pick up all the items we tested it on and did not show any signs of wear or tearing. We intend to conduct more gripper material testing as we continue building our robot and encounter more types of items *Computer*

For the robot's computer, we decided to use the Rev Robotics Control Hub since it has the capability to use an android phone to enable a laptop to connect to the robot. This is important as the user interface is on the laptop, so the robot can send the output of its image processing to the user. The image processing is done on the same phone. We initially planned on using the Nvidia Jetson AGX Xavier as it has ample processing power to support the computer vision. But as we had to shift our plans mid-way through the semester, we had to change up our design. and test cases.

Secondly, we considered the ability of our universal gripper to grab various types of fallen objects in a household. While the gripper was tested to work on more "softer" and smaller objects, there was always the possibility that sharp edged everyday items (small knives, toys with sharp plastic corners, etc.) could potentially penetrate the soft surface of the gripper and render it useless. Additionally, heavier or more slippery objects like TV's remote controllers would be harder to pick up using suction. One solution we considered is attaching a solenoid-head to ensure that all dropped magnetic items are picked up. This meant that we sacrifice some weight on the gripper's top but ensure that most metal items are able to be picked up.

Thirdly, we were also worried about the robot's stability when the arm rotates to pick up objects. Due to the length and extended nature of the arm, the robot's center of mass could shift whenever the arm is swinging around and grabbing objects. To counteract this, we constructed our arm with as less mass as possible to offset the weight balance and potentially added a weighted counterbalance on the underbelly of the robot to ensure that the robot maintains a stable position.

Finally, we considered the challenges and risks associated with the computer vision portion of the project. Since homes have a large variation in lighting both overall and in various places that the robot could travel to. To prevent this risk, we decided to use OpenCV and use auto-calibration to automatically adjust the color threshold every use. Additionally, we set our use case to bright and well-lit environments to avoid any instances of insufficient or distorted lighting.

To combat the challenges of having a remote semester and team members being separated physically, we made sure to set weekly team goals and keep each other updated during the scheduled team meetings and an additional internal meeting every week. That way, we are able to meet our deadlines and help each other whenever obstacles arise.

VIII. ETHICAL ISSUES

There are some ethical problems with our project. One concern that we have is that our robot could be used to covertly steal items on the ground that people would not expect to look downwards. Since our robot has a height of less than 1 foot, it can easily go unnoticed as it drives under a person's line of sight. While this is a valid concern, the nature of our robot's motors makes it difficult to go unnoticed as the servos for both the transport system and the arm rotary system make a lot of noise when activated. This means that unless people do not look down and have hearing problems, the likelihood of stealing anything with this robot is small.

Another ethical issue that this product has is that people could modify our product so that harmful instruments such as make-shift bombs or knives or other effects could be attached to our robot. Since our product is teleoperated, attaching weapons on it and remotely controlling the robot to do harm allows the perpetrator to escape blame more easily. For this problem, there is not much that can be done regarding how people may modify our product at will.

The final concern raised about our product was that since our robot uses live-feed video to aid the user to control it, potential violations of privacy. As the camera feed is what enables the teleoperation of our robot, we cannot see a way to disable the camera and still make our product work. However, this concern becomes trivial considering the fact that our product is meant to be used indoors and at the safety of one's home or workplace.

IX. RELATED WORKS

One project that resembles ours in terms of goal is BallBot, an autonomous tennis-ball collector designed to help the user train their tennis skills more efficiently by allowing them to have the robot pick up the tennis balls. We feel that this project resembles ours in goal because it is designed to help people collect items off of the floor (in this case tennis court) using computer vision to identify and detect a specific set of desired objects.

Similar to our system, Ballbot uses computer vision to determine where the robot should go and collect a set amount of items. It uses motors that are attached to wheels to launch the tennis balls up a ramp to the storage bin at the back of the robot.

The major difference between BallBot and EDC is that our product requires user input for the controls and deciding what the robot would do whereas BallBot is fully autonomous outside of the start and end of its operation.

X. SUMMARY

A. Summary

For the most part, our project has met our design expectations as the key aspects of the project, item collection and the computer vision components, performed well enough for our product to meet our design expectations.

The gripper was able to grasp a wide variety of objects ranging from kitchen utensils to metal gears. With a functioning universal gripper, our robot was able to collect 25 everyday household items, well over our initial goal of 15 items, in its storage bin.

Our computer vision system was able to perform adequately for the final product, but could not meet our expectations set at the beginning of the project. This, however, does not mean that our computer vision component was a failure. We could not meet our expectations because our expectations in the first place were unrealistic for our given expertise and the limitations on our system regarding budget. We expected our computer vision component to be able to detect the marked objects in both good and bad lighting conditions and at the same time run extremely fast. These two factors in conjunction to one another set unrealistic expectations regarding our computer vision component.

Overall, we feel that our project was a success in delivering an easy-to-control robot that is designed to pick up small household objects with the aid of computer vision.

B. Lessons Learned

From our experiences, we learned that coordinating and distributing work on a physical system is extremely challenging. We wish that we had laid out more concrete plans at the start of the semester. Had we planned out better and delineated our roles so that each member would have to report more frequently and accurately on their progress, we feel that our project would not have had to go under such a dramatic change at the midpoint of the semester.

GLOSSARY OF ACRONYMS

EDC - Easy Debris Collector
CV - computer vision

REFERENCES

1. "Adding a Trackbar to Our Applications!" *OpenCV*, docs.opencv.org/3.4/da/d6a/tutorial_trackbar.html.

2. “Contour Features.” *OpenCV*, docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html.
3. “Contours : Getting Started¶.” *OpenCV*, opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html.
4. “Eroding and Dilating.” *OpenCV*, docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html.
5. “Getting Started.” *OpenKinect*, openkinect.org/wiki/Getting_Started.
6. Jun, et al. “OpenCV and Python Color Detection.” *PyImageSearch*, 16 Apr. 2021, www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/.
7. “OpenCV Modules.” *OpenCV*, docs.opencv.org/4.5.2/.

APPENDIX

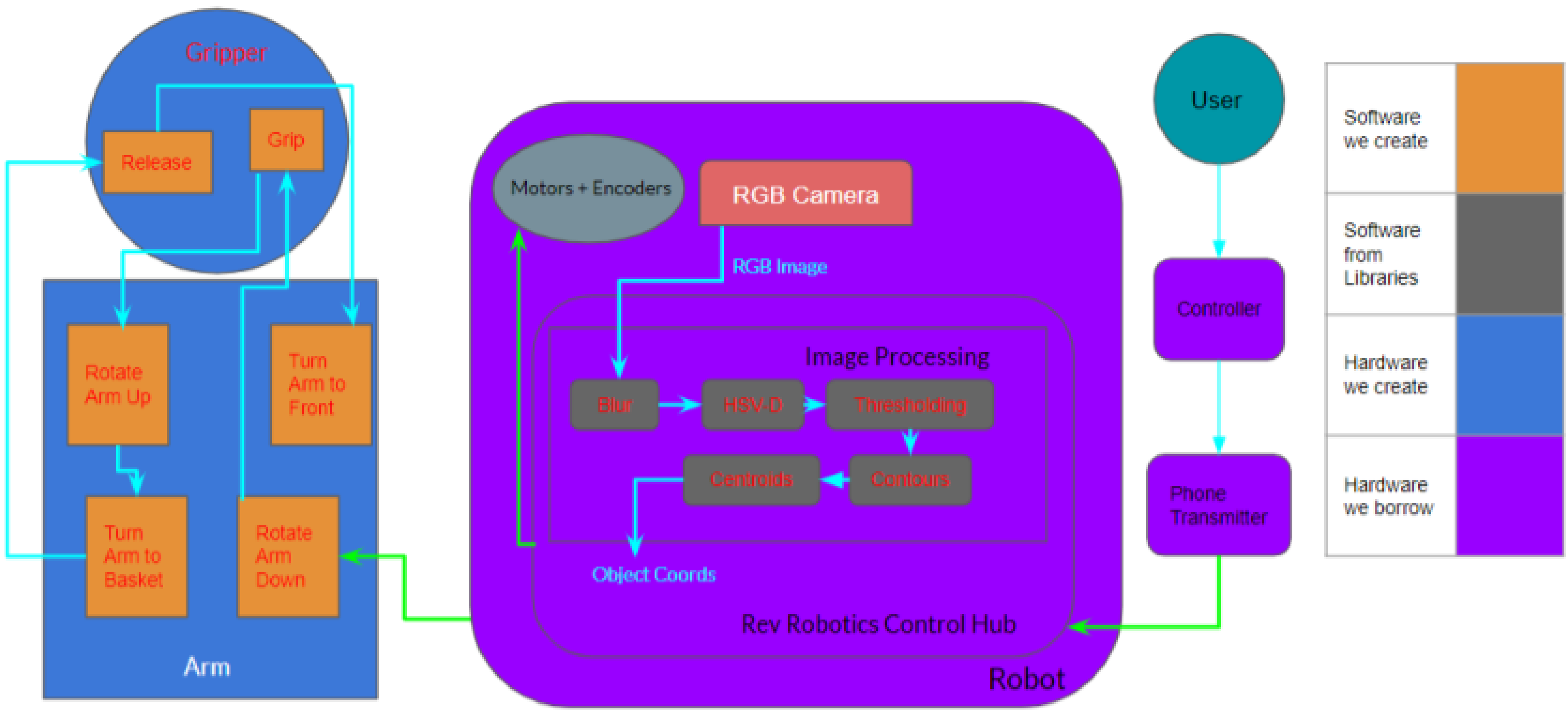

Figure 4: System Specifications

Bill of Materials

Part	Website	Notes	Shipping Cost	Base Cost + Tax (per quantity)
Depth Camera	azon.com/Intel-RealSense-Depth-Camera-D435i/dp/		0	210.52
iRobot Create 4400		Robot Base	0	*Taken from Course Parts list
Nvidia Jetson AGX Xavier		GPU to handle computer vision and motion planning	0	*Taken from Course Parts list
-8 Stainless Steel Flush-Mount aptive Nut, 4-40 Thread Size, .062" min Panel Thick	https://www.mcmaster.com/catalog/127/3470/	Can let us screw stuff into the roomba. 25 per pack	0	8.56
8 Stainless Steel Male-Female eaded Hex Standoff, 1/4" Hex, 2" Length	https://www.mcmaster.com/catalog/127/3536	Standoff from the case so we can have clearance between the case and box for the pump and other devices. Might need larger standoffs depending on pump size etc.	0	2.39
ow-Profile Button Head Torx achine Screw, 18-8 Stainless eel, 4-40 Thread, 3/8" Length	er.com/machine-screws/18-8-stainless-steel-button-	Screws into threads. 25 screws per pack.	0	9.08
arge Round Latex Balloons	om/AnnoDeel-Balloons-Birthday-Decorations-Assor	Flexible rubber for the gripping mechanism	0	18.99
Pump and Vacuum DC Motor - 4.5V and 1.8 LPM	eBBhDiARIsADyBcE6Yh_PC1nYLoGNTc4jP3RusT	Two airpump system for vacuuming and pumping air into the rubber pouch	0	6.95
20kg Digital Servo	6CNKQX4&pd_rd_r=595e0a29-e931-4b50-a2fc-1b2	Two servos for two-servo system for moving the arm	0	16.99
Silicon Tubing (1m)	4661?utm_source=youtube&utm_medium=videodes	Tubing for the gripping mechanism	0	2.5
-Connector for Silicon Tubing	4662?utm_source=youtube&utm_medium=videodes	flexible rubber for the gripping mechanism for the gripper	0	4.5
ir Pump and Vacuum Valve	4663?utm_source=youtube&utm_medium=videodes	Valve for controlling airflow for the gripper	0	2.95

Figure 5: Bill of Materials

Schedule

	Tasks	Start	End	Team Member	Status
		Andy		Entire Team	
		Hojun		Completed	
		Omar		In Progress	
1	Phase 1 : Project Proposal and Planning				
1.1	Identify Project Idea	2/15/2021	2/22/2021	Entire Team	Completed
1.2	Project Abstract	2/15/2021	2/22/2021	Entire Team	Completed
1.3	Setup WordPress Blog	2/15/2021	2/22/2021	Entire Team	Completed
1.4	Project Proposal Presentation	2/15/2021	2/22/2021	Entire Team	Completed
1.5	Research and Finalize Parts List	2/15/2021	2/22/2021	Entire Team	Completed
1.6	Finalize Mechanical Design	2/15/2021	2/22/2021	Entire Team	Completed
2	Phase 2 : Learning, Design, and Implementation				
	Milestone 1 : Proof of Concept & Hardware Obtainment				
2.1.1	Order Hardware and Mech. Parts	2/22/2021	2/22/2021	Entire Team	Completed
2.1.2	Learn how to use OpenCV	2/22/2021	2/26/2021	Andy	Completed
2.1.3	Learn how to program iRobot	2/22/2021	2/26/2021	Hojun	Completed
2.1.4	Determine motion model	2/22/2021	2/26/2021	Omar	Completed
2.1.5	Assemble gripping mechanism	3/1/2021	3/5/2021	Hojun	Completed
2.1.6	CAD robot	3/5/2021	3/12/2021	Hojun	Completed
2.1.7	Reserach color detection algorithm	3/1/2021	3/12/2021	Andy	Completed
2.1.8	Design path planning algorithm	3/1/2021	3/15/2021	Omar	Completed
	Milestone 2 : Basic Integration				
2.2.1	Assemble arm and attach to iRobot	3/15/2021	3/19/2021	Hojun	Completed
2.2.2	Jetson can read images from camera	3/15/2021	3/19/2021	Omar	Completed
2.2.3	Gmapping on simulated robot	3/15/2021	3/31/2021	Omar	Completed
2.2.4	RTABMAP SLAM working on simulated robot	3/15/2021	3/31/2021	Omar	Completed
2.2.5	RTABMAP SLAM working with real camera	3/6/2021	4/5/2021	Omar	Completed
2.2.6	Create a testing robot that can move	3/2/2021	4/9/2021	Omar	Completed
2.2.7	RTABMAP working on testing robot	3/3/2021	4/9/2021	Omar	Completed
2.2.8	ORB SLAM	4/7/2021	4/16/2021	Omar	Completed
2.2.9	Identify colors from camera images	3/15/2021	4/11/2021	Andy	Completed
2.2.10	Track colors during live camera footage	3/15/2021	4/11/2021	Andy	Completed
2.2.11	Motor control with Arduino + Jetson Xavier	3/19/2021	3/31/2021	Hojun	Completed
	Milestone 3 : Full Implementation				
2.3.1	Eliminate contour fragmentation and false positives	3/24/2021	4/21/2021	Andy	Completed
2.3.2	Detection of marked tools on floor during live camera footage	3/24/2021	4/21/0201	Andy	Completed
2.3.3	Arm Movement + Suction Procedure	4/2/2021	4/9/2021	Hojun	Completed
2.3.4	Test and adjust path planning algorithm	3/29/2021	4/14/2021	Omar	Completed
3	Phase 3 : Performance Testing and Refinement				
3.1	Color Recognition Fine Tuning	4/11/2021	4/21/2021	Andy	Completed
3.2	Testing on single tool on floor	4/9/2021	4/21/2021	Omar	Completed
3.3	Multiple types tools on floor	4/9/2021	4/21/2021	Omar	Completed
3.4	Test movement speed	4/9/2021	4/21/2021	Omar	Completed
3.5	Test with different lighting	4/11/2021	4/21/2021	Andy	Completed
3.6	Test grabbing speed	4/12/2021	4/21/2021	Hojun	Completed
3.7	Adjust suction power of gripping mechanism	4/12/2021	4/21/2021	Hojun	Completed
3.8	Real simulation (dropping tools)	4/12/2021	4/21/2021	Entire Team	Completed
3.9	Test on tools without identifiers	4/16/2021	4/21/2021	Entire Team	Completed
4	Phase 4 : Final Report and Presentation				
4.1	Record Final Project Video	4/23/2021	5/30/2021	Entire Team	In Progress
4.2	Final Presentation	4/23/2021	5/30/2021	Entire Team	In Progress
4.3	Finalize Project Video	4/23/2021	5/30/2021	Entire Team	In Progress
4.4	Turn Project in	???	???	Entire Team	In Progress

February: M 22, W 24, F 26 (Hojun); March: M 1, W 3, F 5 (Omar); M 8, W 10, F 12 (Andy); M 15, W 17, F 19 (Hojun); M 22, W 24, F 26 (Omar); M 29, W 31, F 2 (Andy); April: M 5, W 7, F 9 (Hojun); M 12, W 14, F 16 (Omar); M 19, W 21, F 23 (Andy); M 26, W 28, F 30 (Hojun); May: M 3, W 5 (Omar)