Project Documentation

Batpack

A companion for the visually impaired that uses an array of sensors to increase the mobility and spatial awareness of the individual.

University of Central Florida
Department of Electrical & Computer Engineering
Dr. Lei Wei
Senior Design 1 - Group 2

Michael Ferraro (CpE), Connor Roggero (CpE), Lucas Pasqualin (CpE), Christian Jackson (EE)

Table of Contents

1. Executive Summary

Moore's law has ensured that we are surrounded by evidence of always smaller, and always faster devices. However, one area that has remained relatively stagnant has been gadgets to aid the visually impaired. With no recent innovations in this field, visually impaired individuals have been forced to utilize the so-called 'walking stick'. Not only does the walking stick hamper an individual's use of a hand (and usually their dominant hand), but this clunky, obtrusive method of facilitating mobility for visually impaired people has been unchanged for hundreds of years. With recent updates to technology, and especially the small cost of ranging finding sensors, we believe it's time to rethink this device. Using modern components, our aim is to build a device which offers not only the same benefits of the walking stick with none of the disadvantages, but also provides new innovations to aid the visually impaired in their day to day lives.

This report documents the design and research that has gone into the Batpack. It describes our goals and requirements for the system; along with the constraints and standards that the Batpack should abide by. Further in the report we describe the major components, their design, and a way to test the Batpack to ensure it accuracy and functionality.

2. Project Description

The Batpack is a backpack in appearance, but offers an array of sensors to help the visually impaired move around the world without the need of any of their hands to be compromised. The Batpack is accompanied with a pair of goggles that can give more precise feedback on objects in the relative space, with not only a given vibration intensity depending on the distance, but a computer vision system that can help detect everyday surfaces. The user will be able to tell if they are going to be walking on a sidewalk or a grass surface.

2.1. Motivations

It has been nearly 100 years since the walking stick was invented and no improvements have been made to allow the visually impaired improve their quality of life. With the advancements of technology elapsed over this time we believe we can improve on this situation. According to the World Health Organization roughly 39 million people are considered blind and another 246 million have vision that classifies as legally blind. Our team has people in their lives that fall in this classification. One of their biggest difficulties when commuting with a walking stick is that they are unable to use their dominant hand since it is taken up by the stick. To improve on their quality of life we are designing and building the Batpack, allowing them to sense the world around them hands free so they can keep both their hands free for everyday life.

2.2. Goals

For our envisioned device, referred to from here on as Batpack, to truly innovate in this area, it is not enough to simply provide distance information to the user. For Batpack to be truly innovative, it needs to provide the entire package. The device must be light, portable, as unobtrusive to the user as possible, easy to use, and most of all intuitive. While most common electronic devices communicate with their users through sight, Batpack must utilize other methods of communication to provide feedback that feels natural and useful to the visually impaired.

Figure 1. An Illustration of the Batpack's Function

To accomplish these goals, Batpack will come in two separate modules - a haptic feedback backpack with a belt worn around the midriff, and a similar laser/optical feedback system built into a pair of goggles. The haptic feedback belt will be outfitted with both ultrasonic sensors and vibrating units. These two components, when paired together, will be used to indicate the direction and distance of any surrounding objects. This should provide the user with a lightweight method of naturally feeling their surroundings, thus greatly facilitating their mobility without hampering the user or impeding the use of either hand.

The goggles module will provide a method of gauging the distance of objects that are further away with higher resolution. Now instead of pointing a stick to feel objects within close range, users should be able to simply point their heads in a direction and know with high resolution how far away an object is. Additionally, a small camera will be placed on the goggles to aid the detection of street corners and sidewalks. This component will be designed to detect the edges of sidewalks and streets to notify the user if they veer off the path. This additional feature provides the user security in knowing that they are in a safe location while Batpack is being used.

In conclusion, while all the technical and qualitative goals for this device are paramount to its success, ultimately there is one test the device must pass. That test being 'Does Batpack increase the quality of life for visually impaired individuals?' Our goal is to test this assertion with real-life examples. By bringing in individuals that are visually impaired, we will have the benefit of live feedback during and after the design process. Our hope is that continuous real-life testing of the device will ultimately secure its position as a tool for increasing the quality of life for its users.

2.3. Specifications & Requirements

The system will be designed to be low powered, accurate, functional and allow the user to be hands free. This is a huge feature, giving the dominant hand back to the user. As discussed later, price will play a large role in the design. The current design described below will include more expensive pieces to increase performance, but would need to be modified if the item wanted to be successful in the market.

Our functional requirement is to allow a visually impaired user to wear the equipment and maneuver around objects in a room. Additionally, to go outside to use the camera recognition feature, detecting sidewalk from grass. The startup time of the system will be less than fifteen seconds, as calibrations of sensors and camera will need set up. There will be on and off switches to control the belt and glasses, as the vibrators will be on otherwise.

The accuracy of the design should be no more than 10%

The specifications below represent the minimum requirements that need to be met for the project to create a functional and comfortable device to use for a visually impaired person.

- Unit Housing
  - Dimensions : Less than 12” x 12” x 12”
  - Weight : Less than 2lbs

- Micro-Controller
  - I/O : I2C and SPI Communication Protocols
  - Power : Less than 1.5W

- Ultrasonic Sensor
  - Distance : More than 2m

- Laser Sensor
  - Distance : More than 5m

- Vibration Unit(s)
  - Power : Less than 0.1W
  - Signal : PWM

- Camera
  - Resolution : Compatible for Computer Vision Detection
  - Frame Rate : 24

- Battery Supply
  - Capacity : At least 2000mAh

    - Duration : > 24hrs

- The system shall include four ultrasonic sensors
- The system shall include one laser sensor
- The system shall include one camera
- The system shall include five vibrators
- The system shall include a microcontroller and PCB

The goal for this project is to keep the final product near $500. This price could be lowered or increased by changing the types of sensors, microcontrollers, vibrators and camera chosen. We have agreed to set a cap to adhere to, making the most we spend on the project is no more than $1000.

2.4. House of Quality

The House of Quality, found in Figure 1, is a visual guide of relationships for the Batpack between the marketing and engineering requirements.

The marketing requirements are located across the left-hand side of the diagram and are what the customer needs or wants in the system. Requirements such as cost and battery life are very important to the customer and these will ultimately decide if the product is worth purchasing. Next to each of these requirements are the positive or negative polarity in which the customer will view its impact. For the example of cost there is a negative correlation as the customer would want the product to be as cheap as possible.

The engineering requirements of the system are located across the top of the figure. These requirements are more specific to the technical side and are the developers wants and needs for the system. Just like the marketing requirements each one has a negative or positive impact on the product. Since we want the sensor range and computer vision detection to be as far and accurate as possible a positive correlation is associated with them.

	Cost	Weight	Dimensions	Sensor Range	Computer Vision	Point Sensor Laser
Cost				↑	↑	↑
Weight			↑		↑	↑
Dimensions					↑	↑
Sensor Range						↑
Computer Vision						

Legend

↑	Positive Correlation
↑↑	Strong PC
↓	Negative Correlation
+	Positive Polarity
-	Negative Polarity

		Cost	Weight	Dimensions	Sensor Range	Computer Vision	Point Sensor Laser
		-	-	-	+	+	+
Ease of Use	+		↓	↓	↑	↑	↑
Battery Life	+	↑	↑	↑	↓	↓	↓
Portability	+		↓	↓			
Cost	-	↑↑	↑	↑	↑	↑	↑
Sensor Range	+	↑			↑↑		↑
Sensitivity	+	↑			↑	↑	↑
Accuracy	+	↑			↑	↑	↑
Targets for Engineering Requirements		<\$500	<5W	<7"x7"x7"	5m-10m	Sense edge 90% of time	10m range, sense dist.

Figure 2. House of Quality

3. Design Constraints & Standards

3.1. Design Constraints

Hardware and software constraints for this system need to be addressed to properly ensure the design of the project is built with these concerns in mind.

3.1.1. Economical Constraints

Funding is one of the biggest constraints for any project, as it allows for more research to be done. Our team does not have any outside funding from companies and will be providing all the funds for the research, design, and build of this project. This leads our team to fully research the components needed for the system before purchasing any hardware; purchasing additional unneeded hardware will increase the overall cost of the design and will hurt the team's financial limits.

For the end consumer of our product the cost of manufacturing the final product must be kept as low as possible to ensure that it can be marketable. An equilibrium should be accomplished in order to maximize both marketability and performance of the final product.

3.1.2. Health and Safety Constraints

Being that the Batpack's main objective is to allow the visually impaired to better understand their surroundings, while keeping their main hand free, the product must be reliable in order for the user and the people around to be safe. The product must be able to alert the user if the battery needs to be recharged as the product should not leave the individual out of a way to sense their surroundings.

This electronic device will be worn on a person's body and may be in contact with less than desirable weather conditions. The system will need to be able to handle normal weather conditions (i.e. rain, snow, heat, etc…) and be reliable in handling these situations.

When designing the system, we took into consideration how the system will alert the user of objects around them, without any visuals involved. One approach we first thought of was using sound. This would work with a fluctuation in amplitude and a more frequent burst of sounds. However, since the primary customer for this product is visually impaired they must rely heavily on sound, and having the Batpack produce any interference might be obtrusive to the user. Instead we have opted for a haptic feedback to the user. Instead of the of the user receiving sound in different amplitudes and pulses, vibration units will be used to alert the user of surroundings.

3.1.3. Environmental Constraints

Since the main component of our system relies on ultrasonic sensors to relay the distance information about objects around the user, an air temperature monitor may be necessary for accurate data. Ultrasonic sensors use the elapsed time from the time of the pulse to the receiving of the echo, however the time it takes to propagate through the air is effected by the temperature. This can be calculated by the formula below [1]:

$$c(T) = 13044\sqrt{1 + \frac{T}{273}}$$

c(T) is the speed of sound in inches per second
T is the temperature in Celsius

Fortunately, digital temperature sensors can be found which are inexpensive, small, and accurate. If such a component proves to be necessary, we can simply add a module such as the "One-Wire Ambient Temperature Sensor" from Sparkfun.com. This component is priced at only 1.95$, and it is the size of a typical 2N222A transistor. Not only is this price-point fantastic, but the small size would allow us to mount this module wherever it was convenient. The component is accurate to plus or minus 5 degrees Celsius. This is important to our purposes because it would allow us to calculate the velocity of sound in air to a higher resolution. Additionally, the components power draw is negligible, operating at only 3 volts and amperage in the milliamps.

In addition to temperature limitations to the sensors, the batter used to power the system has solar cells that will charge the battery. While walking, users will be exposing the battery to sunlight, extending the batteries life span. If there is overcast weather conditions, however, this recharging ability will be reduced.

3.1.4. Time Constraints

This constraint is what the team needs to constantly check up on. The progress of the project must be constantly maintained in order for the completion of each section to be done in time. The research and final design of the Batpack should be completed by the end of the semester on August 8th 2017. A fully functional Batpack should be completed by the beginning of December for the final presentation. Enough time should be allocated to fix any issues with the design of the project, this will lead to the Batpack being a more reliable and functioning product. A full list of milestones can be found in the "Administration" section.

3.1.5. Manufacturability Constraints

Certain availability of manufacturing will be limited from our team due to the lack of any sort of proper funding. This includes the main enclosure of the device, the backpack. The Batpack's MCU is currently designed to lay inside of a backpack with all the components attached or sewed into it. This included adding a belt around the midriff to house the ultrasonic sensors and the vibrator motors associated with each one. Another additional piece that will have sewn in component is the backpack straps. This part of the Batpack will have to be done by the team.

We will have to also take into consideration the availability of each part in the system and how long they will be supported in produced for. To continue the product development, we want to ensure that the Batpack can be produced in the future without the worry of a component being discontinued.

3.1.6. Sustainability Constraints

The longevity of the Batpack's lifespan is important to ensure a successful product. The product will be going through everyday wear and tear of any other accessory. Usually a person will put on and off their backpack multiple times a day, depending if they are on the move or sitting down. Although the Batpack does not have to go through the harsh treatment of extreme weather conditions, like hurricanes or below freezing temperatures, it does have to survive every day scenarios such as snow and rain. Since this product will likely cost an individual more than $400 it should be able to last at least three years of everyday use, this does not include situations like full water submersion.

3.1.7. Political Constraints

Our team as considered possible political constraints and it was decided that the Batpack does not fall under any relevant political constraints.

3.1.8. Social Constraints

The Batpack is designed for those that visually impaired; this does not contain any information on the height and body type of the users. The Batpack will need to accommodate any body size and height of the individual. Using elastic material will only get the Batpack so far in the universal fit of individuals. This elastic material will be around the waistband of the Batpack, which houses the ultrasonic sensors. For the Batpack to be widely available, multiple sizes may have to be designed to accommodate the difference in size of each individual user.

The goal for the Batpack is to make the product available for anyone who may need assistance in their visual needs. To accomplish this task the Batpack should be designed with the cost of the product in mind. Creating a more cost-efficient product will allow more users to use and benefit from the Batpack's capabilities.

3.2. Related Standards

Design standards are used by many industries to assist in designing products. They help ensure that products that are produced work effectively and are made in a way that any engineer can understand. The following standards were used in order to ensure the system being designed is done so safely and correctly. They are related to power systems, the battery, the printed circuit board, C programming language, and soldering techniques.

3.2.1. Power Standards

The safety standards described by CUI Inc., a technology company focused on on the development and distribution of electronic components, regarding power supply is intended to protect against fires, electric shock, and injury. Their Power Supply Safety Standards will be used to classify the Batpack's class of circuitry and the required safety measures that need to be implemented.

The Power Supply Safety Standards list the following requirements to be called a Safety Extra-Low Voltage (SELV) Circuit:

- Cannot reach a hazardous voltage (42.4 Vac or 60 Vdc peak without a limited current circuit) under normal operation or a single fault condition.

- In case of a single fault condition (insulation or component failure), accessible voltage may not exceed 42.4 Vac or 60 Vdc for longer than 200 ms.

- An absolute limit of 71 Vac peak or 120 Vdc must not be exceeded.

- SELV Circuits require two levels of protection from hazardous voltages I.e. double insulation or a single layer of insulation and an earthed conductive barrier.

SELV Circuits are considered safe for operator access. SELV power supply outputs do not require extensive safety testing or creepage and clearance evaluations.

As the Batpack classifies as a SELV Circuit, being that it will never exceed 10V, it will be a Class III circuit. It will be inherently safe from hazardous shock as it cannot generate a voltage that high.

3.2.2. Battery Standards

Given that this project is a portable system, and that a rechargeable battery would be greatly useful to any operator of this system, ANSI C18.2M was selected. The ANSI is the American National Standards Institute, a private non-profit organization that oversees voluntary consensus standards for products in the United States. Almost all batteries available in the United States will meet the standards set by the ANSI. These standards in particular apply to portable rechargeable batteries of the following compositions: Nickel-cadmium, Nickel-metal hydride, Lithium-ion (including lithium ion polymer).

According to these standards, rechargeable batteries are very safe to use so long as they are not be disassembled, crushed, punctured, opened, or otherwise mutilated. In most circumstances, a person walking to their destination will not experience any conditions that will cause a battery failure according to the ANSI design standards. Additionally, the battery will be stored within the backpack, giving it a layer of padding from shock.

These standards require batteries to be tested at an acceleration of 150gn for a pulse duration of 6 milliseconds. This is well above what the battery will experience in most situations a person might experience when walking, such as a fall, or running into an object. Therefore, even if the battery were to experience most of the force in such an incident it will continue operating.

Temperature-wise, rechargeable batteries are tested with temperatures no human will be casually walking in. They are kept in ovens at 130 degrees centigrade, and stored long-term in -40 degrees centigrade. This makes them suitable in a larger temperature range than the operators themselves.

Finally, the standards strongly recommend that the batteries not be left in their charging receptacle with the wrong terminals connected. This is of significance to this project as operators will be blind. This is relevant to design because it will be necessary for either the battery that is selected for use to be impossible to charge with the incorrect terminal connection, or for the battery and terminal to be marked in a way that will make it readily evident for someone without sight to tell which terminal is which, preferably without touching the conductive surfaces of the battery and charger.

3.2.3. PCB Standards

The Association Connecting Electronics Industries, also commonly called the Institute for Printed Circuits (IPC), regulates standards for commercially and individually produced printed circuit boards. Important, basic documents produced by them include:

- IPC-T-50 Terms and Definitions
- IPC-2615 Printed Board Dimensions and Tolerances
- IPC-D-325 Documentation Requirements for Printed Boards
- IPC-A-31 Flexible Raw Material Test Pattern
- IPC-ET-652 Guidelines and Requirements for Electrical Testing of Unpopulated Printed Boards

Also important to the design of this project is IPC-2221 Generic Standard on Printed Board Design. This document discusses general requirements and proper component mounting when it comes to designing a PCB.

Other documents produced by the IPC are related to material specifications, performance, inspection, and flexible assembly/materials standards. Given that this project is not going into production, and the PCB will be made by a company that meets these standards, non-design related standards are not of concern.

For rugged or hostile environments, indirect cooling methods need to employed ensure the PCB remains cool enough to operate without damage. For the purposes of a PCB stored in a backpack, experiencing a maximum temperature during Summer months, a small heat sink will be suitable.

As a whole, these documents will ensure that PCB designs achieve the goals of the product. The end product will perform to the quality specified in the specifications portion of this document, and do so reliably. Adhering to these standards will make reworking the PCB design easier further down the line, saving time and resources. It will also make it so that all group members stay on the same page with the design, further minimizing time taken to understand the other's work.

3.2.4. Soldering Standards

A freely available set of standards for reliability and safety for soldering is made available by the National Aeronautics and Space Administration (NASA). The document number is NASA-STD-8739.3. This document prescribes NASA's process and end-item requirements for a reliable soldered electrical connection.

Adhering to this standard will ensure that the PCB will work properly and reliably once its parts are soldered on.

The document lists requirements for a suitable environment necessary for proper soldering. A room temperature of 20-30 degrees centigrade is considered suitable for soldering. The humidity affects how far to the edge of that range the room can be. A humidity of 30% requires the temperature not be below 23 degrees centigrade, while a humidity of 70% requires the temperature to not be above 26 degrees centigrade. Furthermore, for the safety for the operator, the room being utilized requires a certain degree of ventilation to prevent the inhalation of toxic fumes. While in the field, and these environmental specifications cannot be met, extra care must be taken to ensure quality soldering.

A couple of additional operator safety standards specify proper safety equipment. Cleaning tools like brushes should be made out of plastic to minimize static shock, which can injure both operators and circuitry alike. Thermal shunts will similarly protect components and operators from being burned. Magnification aids will assist the operator in soldering without getting too close to dangerously hot parts and toxic gasses. During operation gloves and finger cots that do not generate static charge shall be worn to prevent burns and shock.

Proper pre-soldering preparations are described as well. Insulation of components to be soldered needs to be stripped, and afterwards if any damage is displayed beyond minor scuffing and discolorations, the component may not be sued. Additionally, the conductors that have had their insulation stripped must not have suffered more than minor damage. Damage that causes cross-section reduction is especially unacceptable, however smooth imprints are permissible. Any conductors or part leads that are to be soldered need to be tinned and then cleaned prior to soldering. Gold especially needs to be removed from solder areas. Terminals and solder cups need to be examined for damage prior to attachment of conductors, the size of which must be proper for attachment. PCBs must be de-moisturized and cleaned within 8 hours prior to soldering by being baked in a 93-degree centigrade oven for four hours or in a vacuum oven at lower temperatures. They may be stored in moisture controlled areas after this and before soldering.

During soldering it is recommended to try and incorporate stress-relief methods for all leads and conductors leading into soldering terminals. Minimizing forces at these points is important as soldering is not intended to provide a strong mechanical connection, just an electrical connection. It is important to follow the procedures for parts laid out in their engineering documentation and the methods given. If the parts have markings they should be followed. Any hookup wire in excess of one inch of length needs a better mechanical support than the soldering connection.

3.2.5. C Language Standards

It is of paramount importance that the code we write is easy to understand, easy to navigate, and easy to share. For this reason, we are choosing to adopt a coding style standard. For the sake of completeness, we've selected the "Recommended C Style and Coding Standard" produced by Bell Labs. The document was first developed in an industry setting for the Indiana Hill Labs branch of AT&T. According to the standard itself, this standard was shared internally and amongst the industry, and was later updated. The standard selected has a wide range of topics, ranging from File naming, to function declaration, to indentation and commenting style.

While this may seem like overkill, by choosing a document that is all-encompassing we are eliminating any ambiguities in developing code. This ensures that the code is homogenous throughout, professional in presentation, and finally and most importantly, easy to understand.

4. Project Research

The key idea behind this product is to let the user be hands free when walking around. The product will include a belt piece and goggles for the user to wear. Through the interaction of the two, the user will be guided through society with ease.

The belt piece will be equipped with multiple ultrasonic sensors to prevent the user from running into objects. Each of the sensors will be equipped with a vibrator, that will vibrate stronger when objects appear to be closer. Meaning you can detect multiple things around you. For example, if there was a person standing to the right of you one foot away and an ice cream stand standing three feet away from you to the left, the user would feel a stronger vibration on there right side of their body then their left (which has a small vibration to inform the user something is still there). This gives them a full 360 degrees of recognition, almost like eyes in the back of your head!

The glasses will be implemented with a extremely precise optical sensor. It will be able to handle a larger distance than the ultrasonic sensor. Additionally, the user will be able to point their head in any direction they want to find the object. Now coming back to the previous example, the user could turn their head to the ice-cream stand to find exactly where it is and walk to it for a delicious treat. Another beneficiary this brings is the distance the user can scan is now much longer than their walking stick. He or she will now have more time to react to the situation ahead. This can be extremely helpful when walking across a street with cars driving around. What if they were crossing the street, when allowed and a car came around the corner running the red light? The user will hear a car approaching and turn their head to see the terrible driver with a distance much longer than their walking stick, possibly saving their life.

As discussed previously, there will be a camera attached to the goggles as well as to act as their eyes. The idea behind the camera is for ground detection. For example, this will differentiate the difference between the sidewalk and grass, keeping the user on path. Normally all of this is done with the walking stick. Continuing with the previous example, say the ice cream stand is actually across a patch of grass and the line starts a few people back, the user would identify the difference in ground detection and walk around it, looking for the back of the line by scanning for opening in the line.

There are thousands of different types of sensors and cameras available, and we plan to experiment with multiple ones to find the right one for this device. We want to meet our specifications, as listed above. The key to all of this is how we connect all the pieces together to the PCB design, which will be handled by Christian and Mike. One important thing we need to keep in mind is the fact that some pieces will not work with others due to compatibility issues. The product we find best for the situation might not be compatible with the other best pieces we found, meaning some trade offs will need to be done. This is why the budget cost could be expected up to $1000 due to buying multiple products for testing, even though they aren't all being used.

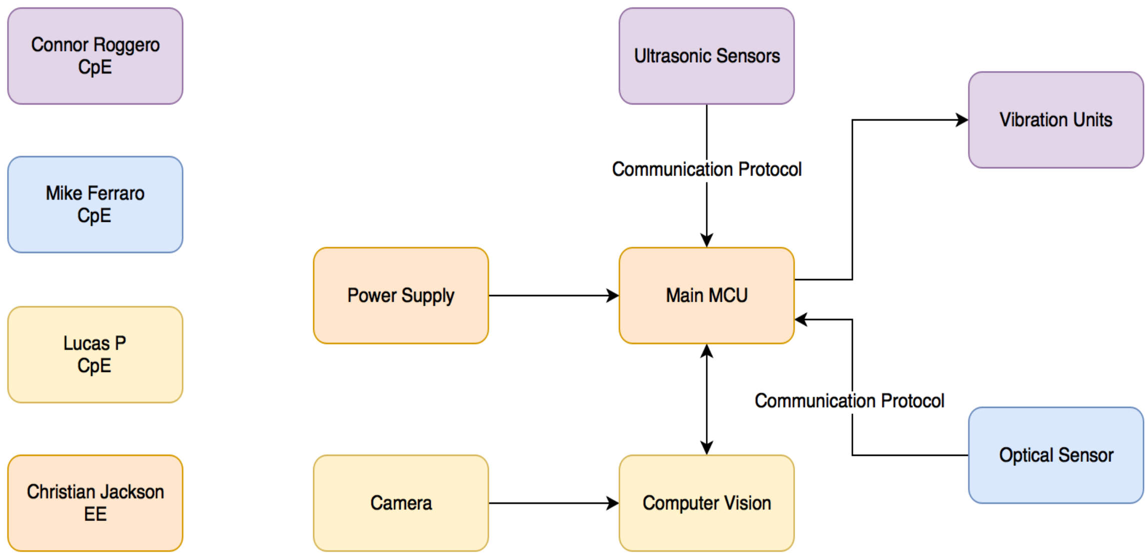


Figure 3. High Level Design Layout

4.1. Existing Products & Technology

With over 285 million visually impaired people around the world, the need for technology to help them has been thought of before. Follow, as we dive deep into the pre-existing technology and discover what has been available.

The following is a selection of a few well-designed products that are currently helping the visually impaired. These products were selected primarily based on their quality and functionality.

Assisted vision smart glasses:

Figure 4. Assisted Vision Smart Glasses [2]

This product, which has similar goals with the Batpack, helps the user walk around and recognize obstacles, allowing them to become more independent. These glasses capitalize even more when the user still has remaining sight, as most visually impaired individuals are this way. “They are constructed using transparent OLED displays, two small cameras, a gyroscope, a compass, a GPS unit and a headphone [2].” They use brightness to show depth, meaning when objects are close, they design it to make it brighter.

Figure 5. Assisted Vision Wearer Perspective [2]

On the one hand, this is much more complicated than our design, but on the other we bring the feature of ultrasonic sensors to help guide the user. An interesting feature they add is the headphone. We opted out of this idea, as noise can be an issue if in a busy area, so we chose to go with a vibration to talk to the user. The gyroscope is also nice feature, as it helps calculate changes in perspective as the user moves around, but we opted out of this as we are using line detection and vibrators to steer the user left and right.

AI Glasses:

Figure 6. Ai Glasses [2]

Designed in Mexico, these glasses come equipped with ultrasound techniques and computational geometry. With this, the data can be used to train the system, also known as artificial intelligence. This thought process is like our design, as we construct many different approaches, but are leaning towards convolutional neural networks. The choice of algorithm for our design is explained in Optical Algorithm chapter. This product combines "glasses with stereo sound sensors and GPS technology attached to a tablet, which can give spoken directions and recognize denominations or currency, read signs, identify colors and other things. [2]" Far more advanced in capabilities than our pair, and includes the ultrasonic sensor. But again, we find the feature of noise, not always the best option, but could be considered in future design, as other products are using it

Laser cane:

This cane uses the idea of laser beams to detect obstacles much like how we do. Equipped with audio and vibrators, you point the device and a distance is calculated. Depending on the distance, one of three audio signals will be played, and if an object gets to close the vibrator will kick in when an object appears in front of the user. "It can detect objects and hazards up to 12 feet" [3]. This idea was thought of as one of our original ideas, but opted out after considerations for the ultrasonic sound sensors were included in the design.

White Cane 2.0:

Dr.Daniela Rus created the White Cane 2.0. The name inspired from a case in 1921, where a man named James Biggs, painted his cane white after losing his sight to help inform others of his disability. It uses a camera that creates three dimensional images to process information to signal the belt which is equipped with vibrators. This concept is like ours as it is two pieces, uses a camera and vibrators to help steer the user. They agree that the blind depend a great deal on their hearing and using sound isn't the best option. This product uses the similar idea, with increasing vibration as an object begins to become closer. This product lacks the laser and ultrasonic sound sensors used in our design, primarily focusing on recognition from the camera to steer the user. This allows for additional features. The White Cane comes equipped with a touchpad with instructions in Braille. One struggle explained by Dr. Daniela Rus is visually impaired students struggle to find empty seats in a crowded lecture theatre. The system is now capable of finding empty chairs and use motors to decide which way to travel and vibrate when an empty chair is spotted [4].

Figure 7. Dr.Daniela Rus [2]

SmartCane:

Professor Meenakshi Balakrishnan, a computer engineer at IIT Delhi, led a team of researchers and students to develop the SmartCane [5]. This product uses ultrasound to detect objects and vibrate when something is in its path. A simple design, but implemented well and at a lost cost of only fifty dollars. Useful for users in other countries whose budgets are much lower and don't have much other than the typing walking stick. The device can help prevent users from running into things lifted off the ground, such as trucks, as a normal walking stick will simply just pass beneath it. This is completely doable, as ultrasonic sound sensors are much cheaper than other types of sensors and still effective, but we feel designs like this is not complex enough to be used daily.

4.1.1. Considerations for Future Technology

As we look at existing products and technology we see small advancements in the technology for the visually impaired. This included well designed glasses that recognize objects using multiple campers, gyroscope, compass and a GPS; glasses that had ultrasonic sound sensors connected with artificial intelligence; a laser cane that used laser beams to detect objects near the user; the white cane, a device strapped around the neck that used a camera and vibrators to steer the user; and the Smartcane that used ultrasonic sensors and vibrators to warn the user. All of these were great ideas to begin the technology for the visually impaired. Our product has incorporated a few of these ideas as you read throughout the paper but the technology can still be improved drastically.

We have high hopes for our product to help the visually impaired and know there could be more advancements to the product, as we hope for others to add on to our idea. One great idea we thought of after the math was to add a java application that would use the camera. The idea behind the application is to have the user point at an object and using machine learning, it would recognize the object. This idea comes from the Google's deep learning. More details on how this deep learning is described in the section 4.3.1.7 Neural Networks. Below is described a more in depth explanation of how recognizing objects with Neural Networks works. Now the visually impaired can identify objects near them and have a voice to tell them what it is.

Another great idea is to incorporate Amazon's Alexa into the product. This would allow the user to make phone calls, control the temperature of the house, get news, play music and much more. You can practically train Alexa to do anything you want. This would take the dependence of the visually impaired to a much lower state, as they can live in ease. We would have to go through getting permission from Amazon before incorporating this piece. For now we will skip on this idea, but think about it for the future.

With these two devices incorporated in the Batpack, not only will it bring it to a new level, but include the missing components the devices currently on the market are lacking.

4.2. Hardware Design Research

There are a few things we must consider when designing Batpack. Because the device relies so heavily on range finding sensors, it is essential we have a strong understanding of how these components work.

Provided below is a list of additional considerations we must keep in mind during the design process for ultrasonic sensors:

- The position of the object we are shooting the signal at. Each sensor comes with specifications of the range it can read from. We must make sure we are in that target range when doing calculations.

- When sending out the signal, if the angle between the sensor and the object is below a certain amount (roughly 45 degrees), the sound will not be reflected back to the receiver. Fortunately, the user can just turn the product easily to fix this issue.

- The object's size. For ultrasonic sensor, the object must be a certain size, or it will not be able to reflect the sound back.

- The material of what we are pointing the sensor at. Some sensors have an issue with reflecting the sound. For example, if the surface is soft, some of the sound is absorbed. Thus, you may have an issue finding your favorite stuffed animal.

- The outside world. This can include rain and outside temperature. These can have an effect on the speed of sound in the air, causing delays or inaccurate readings.

For the glasses, a laser sensor will be used for calculating distance of an object. The considerations can be described below:

- Target distance. Depending on far the object is will play an affect if it recognized, as well as change the accuracy of the readings. As distance increases, the measurement can vary slightly.

- Target size. If the object is too small, the light will be unable to reflect off it

- Orientation to the sensor. Having a bad orientation affects observable cross section, decreasing ability to pick up the return signal

- Reflectivity. There are three different types of reflective surfaces we consider in the daily use.

- Diffuse reflective: Rough surfaces have a diffuse surface that causes reflected energy to disperse uniformly [6]. This is good as it makes the likelihood of the energy to find its way back to the device increase. These are found in everyday objects due to small imperfections in the surfaces of materials. A good example of this is paper. Even though it may look like a smooth surface, paper actually is rough due to the tiny fibers at the microscopic level.

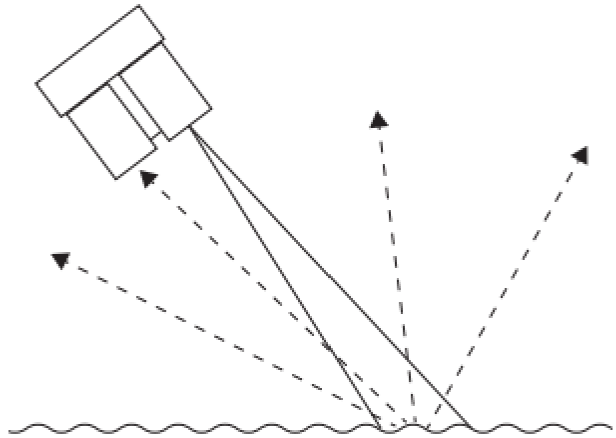


Figure 8. Diffuse reflective

- Specular: Smooth surfaces have specular surfaces which makes it almost impossible for the laser sensor to pick up a reflection that can be used to calculate the distance. This is due to the reflecting have little dispersion, making the beam reflect small. The only way to fix this issue is by taking a measurement with the object directly in front of the sensor. An easy example to picture for these types of objects is a mirror.

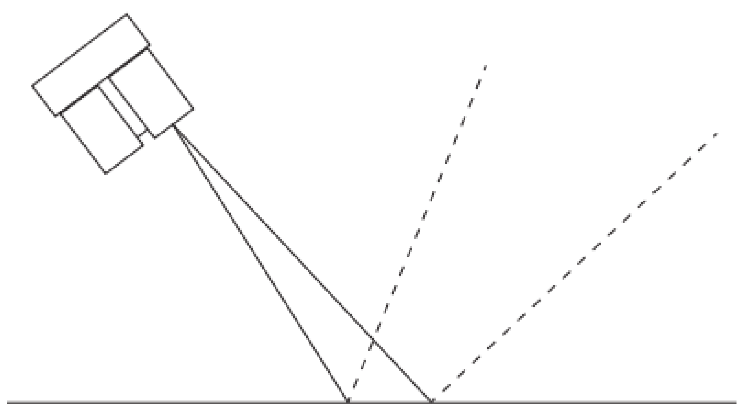


Figure 9. Specular Reflectiveness

- Retro-reflective: Retro surfaces will reflect light with minimal scattering, making these items extremely useful for this product. The reflected energy will run along a vector parallel to the transmitted signal. An easy example for this could be a reflector on the back of a bicycle.

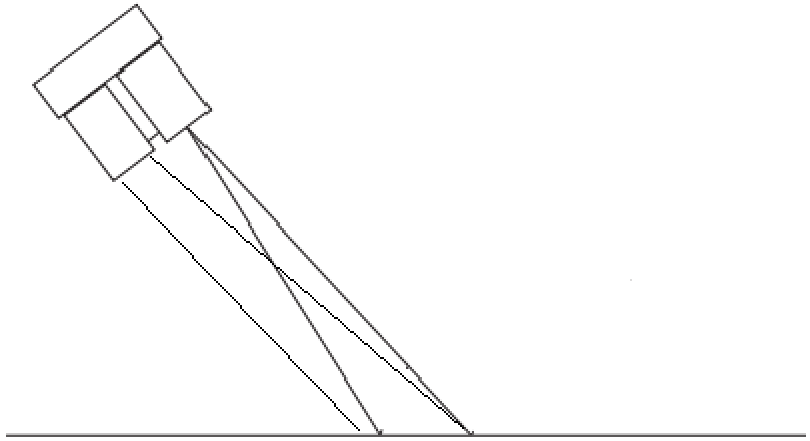


Figure 10. Retro-Reflective

In conclusion, we will be using both sensors to get as much detection as possible of objects nearby. The following diagram represents how you can visualize how each sensor works.

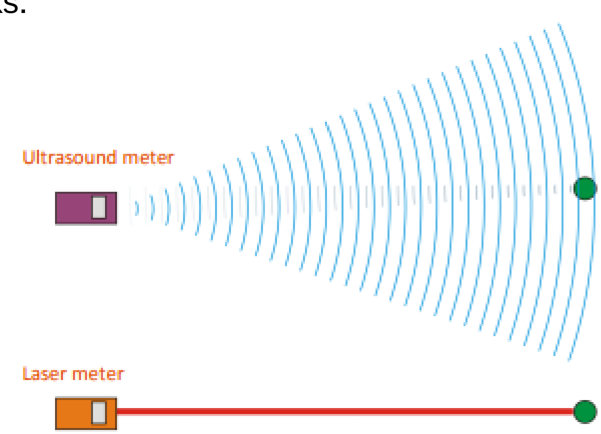


Figure 11. Ultrasonic vs Laser Distance Measuring

4.2.1. Laser Range Finding Sensors

There will be a few considerations for the design of laser sensor that we will look at in order to choose the best product for the best price.

- Size
  - Nothing too large that we won't be able to attach to the glasses
- Weight
  - Not outweighing the glasses, forcing it off the user's face

- Accuracy
    - Compare different lasers for accuracy of product, including when distance gets longer. The size of the object can be a factor. The laser will send the transmission with a spread of roughly .5 degrees up to a meter and then is changed afterwards. This can be estimated by the following formula:
        - Spread angle = distance/100 [6]

- Max range
    - How far the laser will read. It is important to note there is a relationship between distance(D) and the returned signal strength. With increasing the distance, the signal strength is decreased by 1/D^2 [6].

- Update rate
    - The refresh rate for transmitting data, the faster the better

- Interface
    - What protocol is the device using

- Power
    - Want to keep relatively low

- Current consumption
    - Want to keep relatively low, as goal is to make it as low powered as possible

- Operating temperature
    - What weather temperatures the device works in

- Laser wavelength
    - Will denote which part of the spectrum the laser uses

- Optical aperture
    - Different sizes affect cone angle of rays being emitted

The considerations for the laser range finding sensors are as follows:

- LIDAR-Lite v3
- Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YKOF

LIDAR-Lite v3 vs Sharp GP2Y0A02YKOF

Size	.8*1.9*1.6 in	1.2*.5*.85 in
Weight	22g	N/A
Accuracy (error < distance measurement)	+/- 2.5 cm < 5m; +/- 10cm > 5m	N/A
Max Range	40m	1.5m
Update Rate	270 Hz normal, 650 Hz Fast mode	N/A
Interface	I2C or PWM(Pulse width modulation)	N/A
Power	5V	4.5-5.5V
Current Consumption	105 mA idle, 135 mA active	33 mA, max 50 mA
Operating Temperature	-4 to 140 degrees F	14 – 140 degrees F
Laser Wavelength	905 nm	850 +/- 70 nm
Optical aperture	12x2 mm	N/A
Price	$150	$15

Table 1. LIDAR vs Sharp

As we compare from the datasheets, it is clear the LIDAR-Lite v3 wins in most specs but at a much larger cost. Because we want the best readings possible, as we will be only using one, we will be going with the LIDAR-Lite v3. The infrared proximity sensor will also be used and tested to compare accuracy. Results will be discussed later after testing.

The image below will represent an example how to connect the device to the microcontroller. [6]

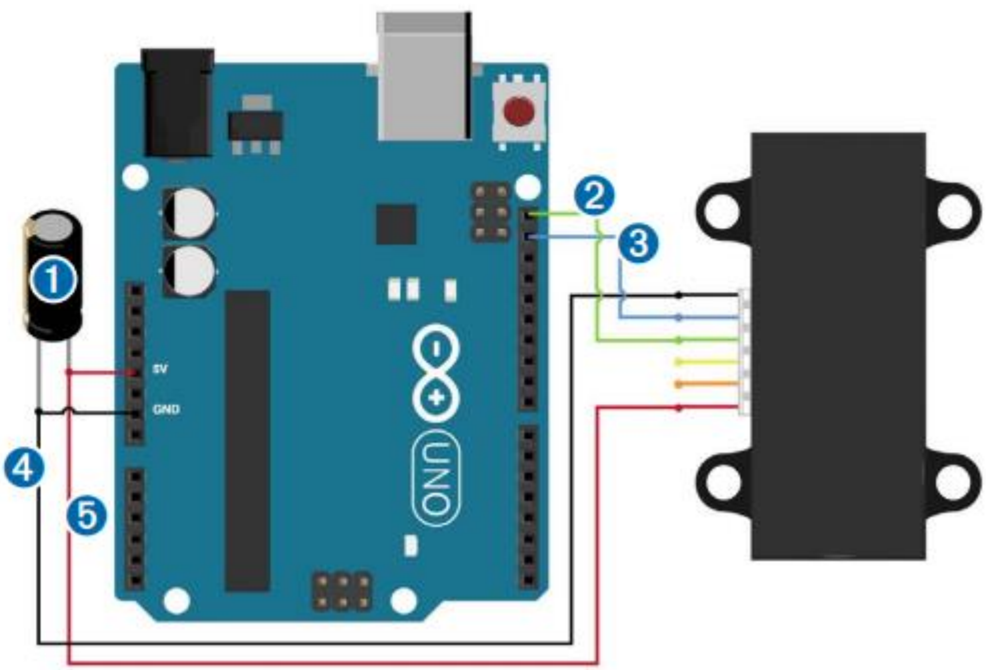


Figure 12. Connections on Microcontroller

So how does it all work?

The laser sensor is more precise for longer distances compared to the ultrasonic range sensors. The component calculates the distance by sending out pulses of near-infrared laser signal. Once the laser beam hits the object, the beam will reflect back to the sensor. The time of flight will give us an idea on how to measure the distance. The distance between two points can be calculated by the following:

$$D = \frac{ct}{2} \qquad t = \frac{\varphi}{\omega}$$

$$D = \frac{1}{2}ct = \frac{1}{2}\frac{c\varphi}{\omega} = \frac{c}{4\pi f}(N\pi + \Delta\varphi) = \frac{\lambda}{4}(N + \Delta N)$$

Key:

D	C	T	φ	ω	$\Delta\varphi$	N	ΔN	λ
Distance	Speed of light	Amount of time round trip	Phase delay	Angular frequency	Phase delay not fulfilling π	Integer number of wave half-cycles	Remaining fractional part	Wavelength

The signal is sent with a coded signature, which it looks for on the reflection. More specifically the device begins by performing a receiver bias correction routine, correcting for changing ambient light levels and allowing maximum sensitivity [6]. Next the device begins the measurement process by sending multiple acquisitions from the transmitter to receive to learn the distance of an object. If there is a match from the signal, the result is stored in memory. The following acquisition is summed with the previous. When multiple signals reflect back to the device after hitting an object, the repeated acquisitions force a peak to show outside the noise. After the peak reaches the maximum value, this value is recorded. This gives the distance of the object. Now that we have the peak value, we take the magnitude of it and will store is as X. There is another factor the still gets calculated in order to determine the signal strength to be a valid reading. This is known as the valid signal threshold, which is calculated from the noise floor. Setting this value to Y. As long as the peak value is larger than the threshold(X>Y), the signal is recorded, otherwise returns 1 cm (an invalid reading). It is important to note that after a distance of 1m, accuracy of the reading changes to +/- 2.5cm. This is a relatively small value, but still important to keep in mind as measuring for accuracy is quite important for the visually impaired.

As we know a negative side to laser range signals is it has troubles with objects that are too small and not very reflective. The returned signal strength could be improved by attaching infrared reflectors to the target to increase performance. This would be useful to put on useful objects, such as a fire alarm, fire extinguishers or other important objects that could help the visually impaired.

4.2.1.1. I2C Protocol

Inter-integrated Circuit Protocol is a communication protocol to allow multiple “slave” circuits to communicate to a “master” circuit. The optical range sensor the Batpack will use, LIDAR V3, to detect more precise object detection uses the I2C communication protocol.

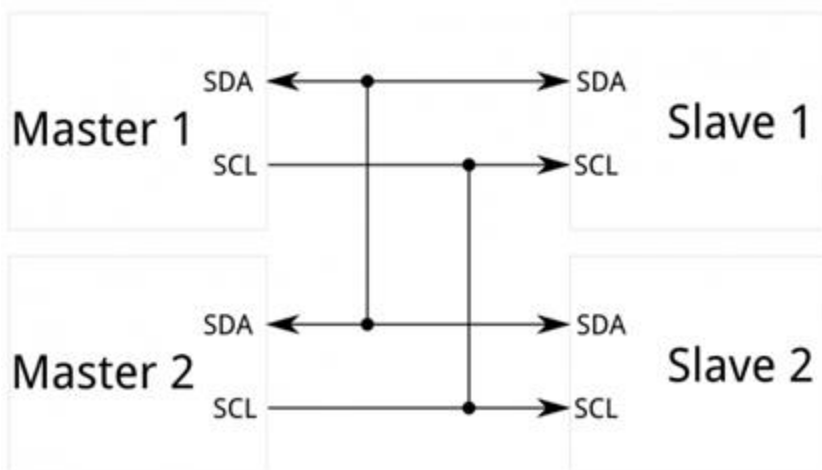


Figure 13. I2C Connection Setup (Courtesy of Sparkfun)

I2C only requires two wires to the master to communicate up to 1008 slave devices. It also allows for multiple master chips to be involved on the same bus, see Figure above. During the communication 8 bits of data are sent between 100kHz or 400kHz, along with an extra bit for the ACKs and NACKs. At the hardware level I2C bus contains only two signals, a data signal, SDA, and a clock signal, SCL. The SCL bus is only controlled by the master, making it easy to synchronize multiple devices. The hardware behind I2C can be seen in the Figure below.

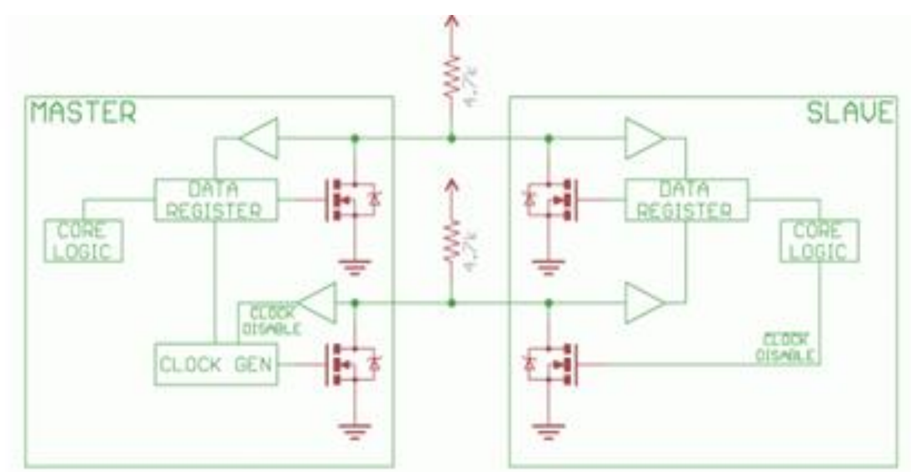


Figure 14. I2C Hardware Level (Courtesy of Sparkfun)

Messages passed between the I2C buses have two types of frames. One is the address frame; this is the frame which tell the slave if it is their message the master is transmitting. The second type of frame is the data frame; more than one can be transmitted but it always is preceded by an address frame. These are transmitted over the SDA bus when SCL is pulled low. Once the message has stopped the master will send a stop condition on the SDA bus and the SCL bus will remain high.

4.2.2. Ultrasonic Range Finding Sensors

For the ultrasonic distance sensor, the sensor sends out an ultrasonic burst, known as a chirp. This transmission will hit an object and reflect the sound back to the receiver, called an echo. The pulse width corresponds with the distance from the object. The speed of sound is roughly 341 m/s. Using the equation below, we can calculate the distance.

Distance = time*speed of sound / 2

Where the time is measured by how long it takes between the chirp and echo.

The beams are sent with a certain beam pattern, which the echo is looking for. The beam angle will change over distance until the full range is met. Using the beam pattern, we can calculate the beam angle for an object at a certain length. Smaller objects will be detected over a narrower beam angle and shorter distance. Larger objects are detected over a wider beam angle and longer range [7].

Pulse-width modulation is considered for this design as it gives us a control of how much power is being supplied to the device, making it easier to handle any power consumptions we need to fix. This is done by switching on the voltage supply on and off a swift rate. The longer the duration between states, will lead to higher power supplied.

One known issue when using multiple ultrasonic sensors, is there can be cross-talk(interference) from the other sensors when using multiple sensors in a design. We can interference by three different methods known as chaining.

- AN output commanded loop: Store all the sensors into an array and begin the loop. The first sensor will process, then trigger the next sensor after done and so on until the end of the array. Then command the first trigger to begin when desired.

- AN output constantly looping: This will run the same as the previous method, but instead automatically restart the process after reaching the end of the array.

- AN output simultaneous operation: Trigger all sensors at same time for range reading. Restart the process whenever desired. This application isn't for most applications, but is considered in the design if the sensors are spaced out enough.

For the purpose of our product, we will be testing the constantly looping and simultaneous operation approaches. As the device will be powered on, we want the sensor to constantly be reading distances on a looping method. An issue with this could be the time it takes for each sensor to calculate the distance of the object nearby, slowing down the chain, delaying the distance calculations for other sensors making objects appear closer to the user faster. We can fix this issue with the simultaneous operation approach, but will need to design the Batpack in a way there is minimal if not no interference between the sensors.

Features we will be paying close attention to are:

- Power
  - Using multiple sensors, we want to target sensors with low power use as the overall design goal is be low powered
- Current Consumption
  - Keep low to keep power low
- Refresh rate
  - How often readings will occur
- Pulse width
  - Considering pulse-width modulation
- Environment
  - What conditions will the device operate in
- Operating frequency
  - Using different frequencies will allow the use of different sensors without interference.

Considerations for the ultrasonic range finder is as follows:

- <u>Ultrasonic Range Finder - LV - MaxSonar - EZ2:</u> This sensor provides very low power ranger, which is useful as we will need multiple sensors. One issue with this sensor is it requires a minimum of 6 inches before it begins recording. When powering on this sensor, it's good practice to have nothing within seven inches of it, as the first cycle is calibrating the device. Ignoring this known issue could cause some objects to be ignored at that range. The following timing diagram shows RX and pulse width set to high. At this time the burst is sent. When the object is detected, the pulse width pin is set to low. The serial data is sent in RS232.

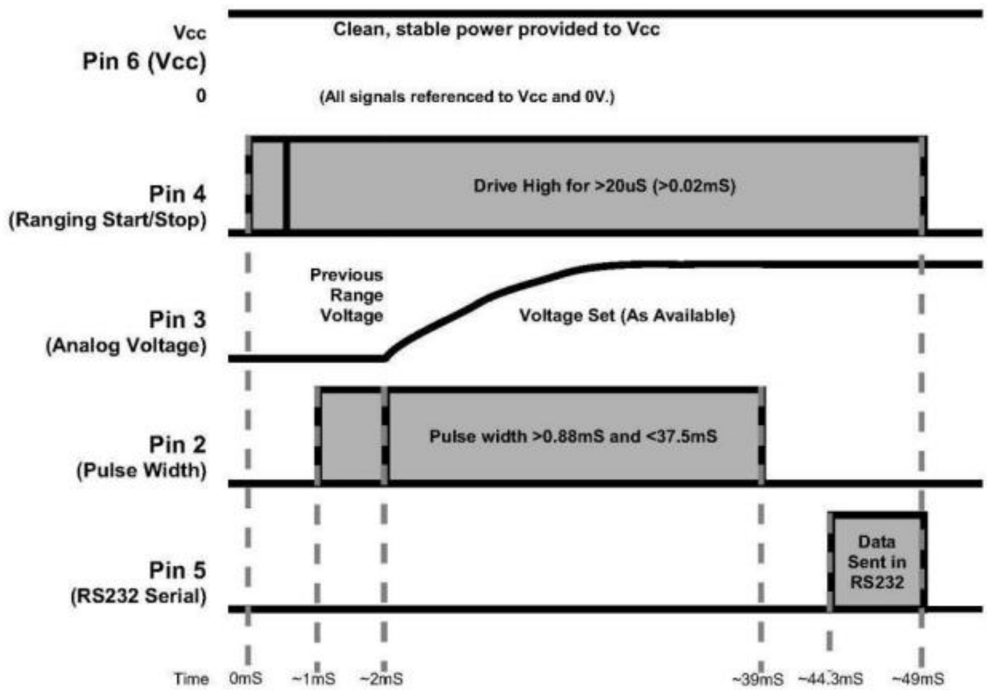


Figure 15. MaxSonar Timing Diagram

- Models:
  - MB1000: Good for people detection, has wide beam, high sensitivity to noise
  - MB1010/MB1020 (chosen model): Balanced for people detection, beam width and sensitivity to noise
  - MB1030/MB1040: Good for large targets, narrow beam and has good noise tolerance

Power	2.5-5.5V
Current consumption	2mA
Refresh rate	20 Hz (50ms)
Pulse width	147 uS/inch
Environment	Indoor
Operating frequency	42KHz
Distance	6.45m

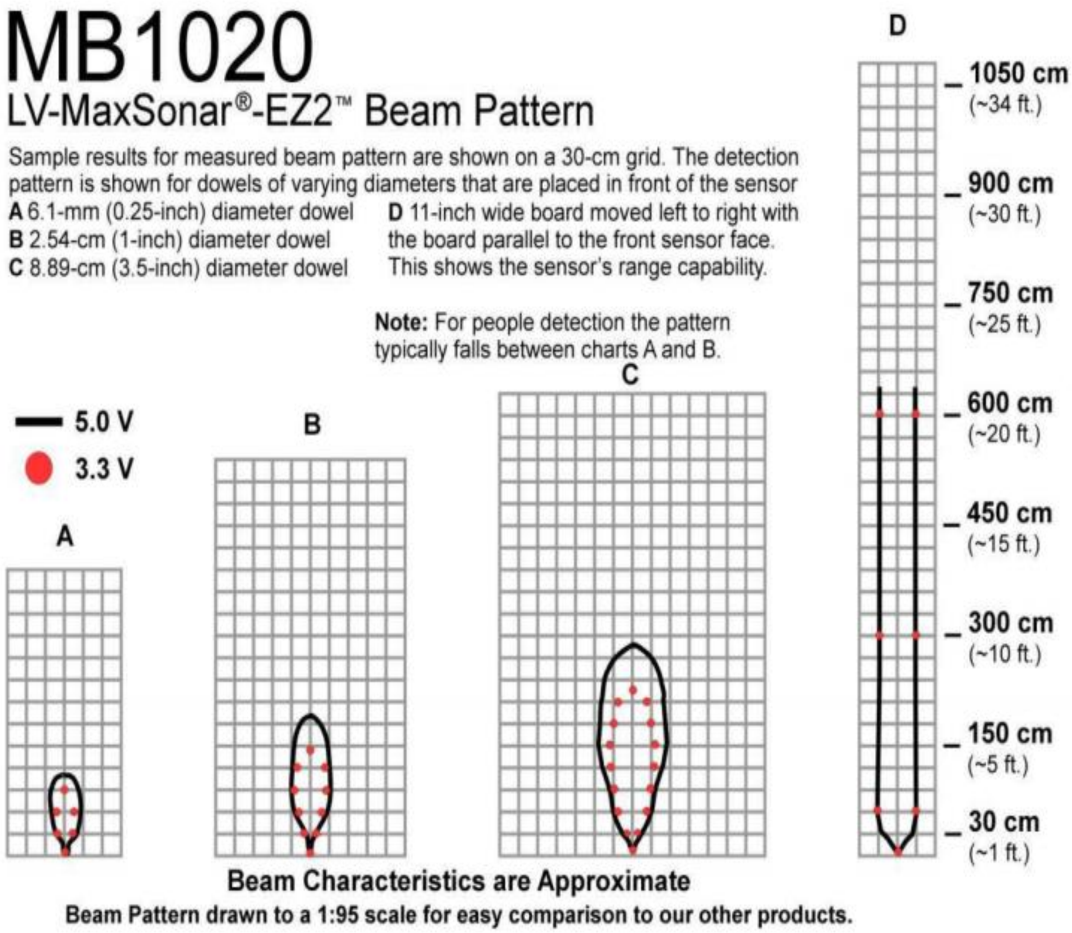


Figure 16. MaxSonar MB1020 Beam Pattern

The beam pattern produced by the MB1020 has a good tradeoff between its sensitivity of picking up objects and side object rejection. MaxBotix's MB1010 is great for general object detection and produces a wider beam. Their MB1040 has the narrowest beam, which makes it good at detecting large object. Finally, our choice, the MB1020 stand in the middle providing the best way to pick up large and smaller objects.

- <u>Ultrasonic Sensor - HC-SR04:</u> This sensor will send out a burst of eight cycle ultrasound and raise its echo to high after receiving in input and output a time in microseconds the sound wave traveled. The following timing diagram from the datasheet represents this in a more visual way.

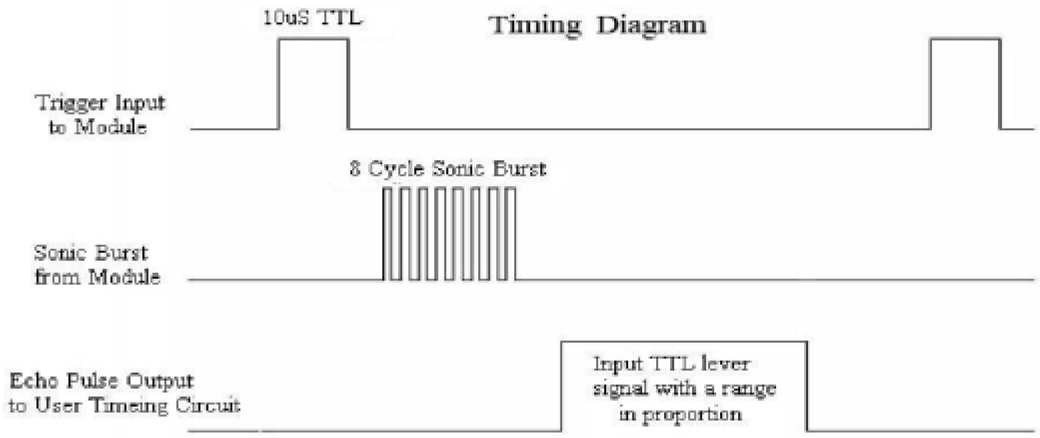


Figure 17. HC - SR04

Power	5V
Current consumption	15mA
Refresh rate	n/a
Measuring Angle	15 degrees
Environment	Indoor
Operating frequency	40Hz
Distance	4m

In our implementation, we will be using both sensors. The operating frequencies are different allowing us to design the system in a way we minimize the possibility of cross-talk. The SR04 will need to be started manually each time a recording wants to be read, so a loop will be needed. One notable thing between the two devices is the current consumption. The SR04 uses seven times more consumption, increases the overall power used by the system.

4.2.3. Vibration Motors

The Batpack will rely heavily on vibrating motors to provide haptic feedback to the user. Because these motors are central to almost all communication occurring to the user, they are a central component of our device, and need to be carefully analyzed and configured. It is extremely important that this component is able to not only vary vibration strength and patterns as quickly as possible, but at the end of the day also communicate clear and intuitive signals to the user.

4.2.3.1. ERM vs LRA

Vibrating motors most often come in two different flavors that we have the option to choose from – ERM and LRA. While both devices essentially accomplish the same thing, they differ in implementation, power draw, and control.

The LRA, which stands for Linear Resonant Actuator, operates on principles similar to those found in loudspeakers. In the component's most basic form, an LRA contains a magnetic mass resting on a spring. The mass is driven to motion by a voice coil -which is essentially another magnet. The voice coil is driven with an AC current, which causes rapid displacement of the magnetic mass resting on the spring. It is this constant displacement which causes the sensation of vibration.

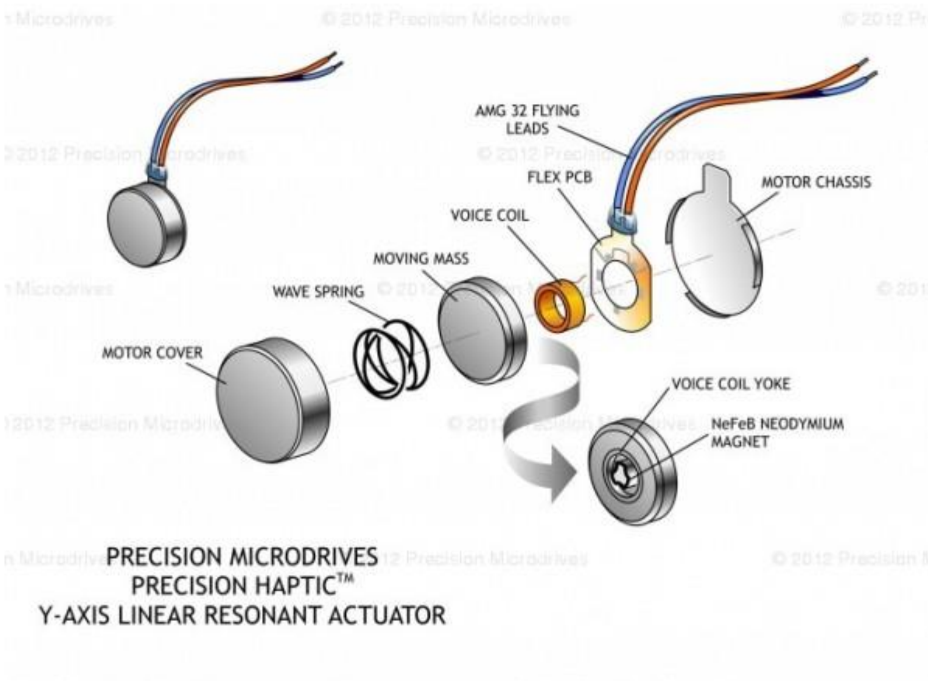


Figure 18. LRA Vibrator

Additionally, because of the mechanics involved in the spring and mass system, this combo is seen as having a natural resonant frequency. This is where the Linear Resonator Actuator gets the 'Resonator' in it's name. This is seen as the main draw of the LRA. Indeed, this is a giant added benefit to the LRA because this resonant frequency can be used to easily modulate the amplitude of the vibration. According to NFP-Motor, a manufacture of electric motors including ERM's and LRA's, this resonant frequency allows the LRA to offer two times the force of traditional ERM's all while drawing half the power. Additionally, because amplitude modulation is so easy in this module, the LRA also provides quicker stopping and starting capabilities (demonstrated in figure below).

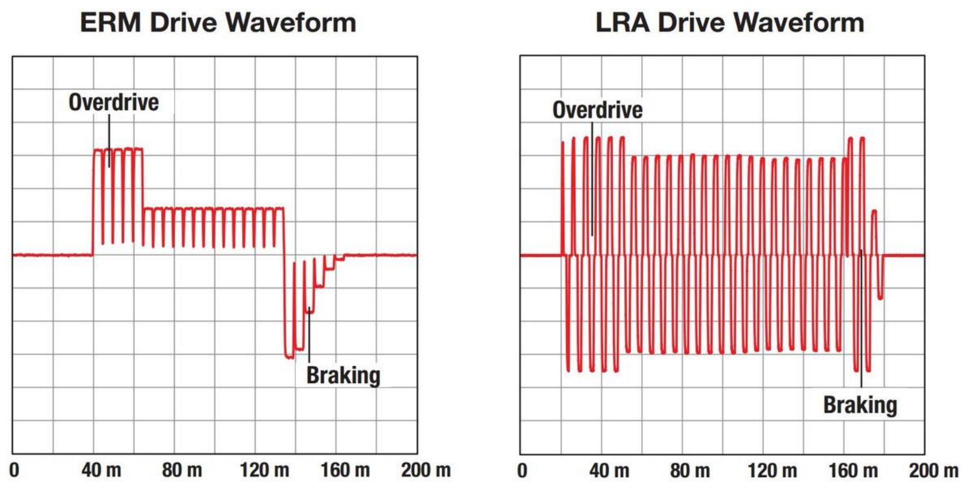


Figure 19. ERM vs. LRA Waveform

However, this added functionality does not come without a cost. To function properly, the LRA must operate around a small band centered around it's resonant frequency. This can be difficult to achieve because the resonant frequency can be affected by a myriad of considerations – including but not limited to acceleration of the body and mounting position. For these reasons it is recommended that LRA's with special software, known as an auto-resonance driver, which automatically adjusts the driving frequency to achieve resonance.

ERM, on the other hand, stands for 'Eccentric Rotating Mass'. These motors are usually found to be much simpler. Essentially ERM vibrators are DC motors which rotate an eccentric mass around a drive shaft. The mass is said to be 'eccentric' because it is asymmetric, causing a net centrifugal force while the mass is rotating. Because the mass is being spun very quickly, most likely several rotations per minute, this constant displacement is felt as a vibration.

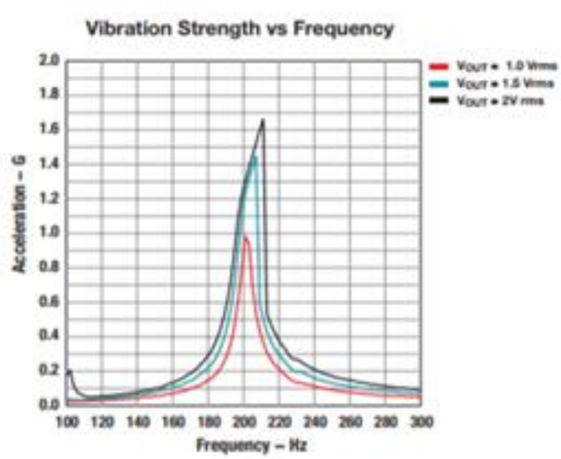


Figure 20. Vibration Strength for LRA Vibrator

For our purposes, it is sufficiently clear that the device best fitted to satisfy our needs are the vibrators in the ERM group. Admittedly, the LRA offers better power consumption, force, and control. However, these added benefits are outweighed by the complexity of the implementation. The additional software needed to run each vibrating module in our opinion is too computationally expensive to run on the embedded microprocessors we have selected – especially when considering not only multiple vibrators but also all the other components running the on microprocessor. Additionally, because these vibrators take as input an AC voltage, using LRA vibrators would require a completely separate hardware module to convert DC voltage from the battery to AC voltage for the LRA. While devices do exist which accomplish exactly that, these devices would represent even more investments in time, space, weight; all things which increase the complexity of the system and make it harder to work within our design constraints. In conclusion, while LRA should be kept in mind as a potential upgrade to future designs, we believe the ERM vibrators are versatile enough to efficiently provide communications to the user.

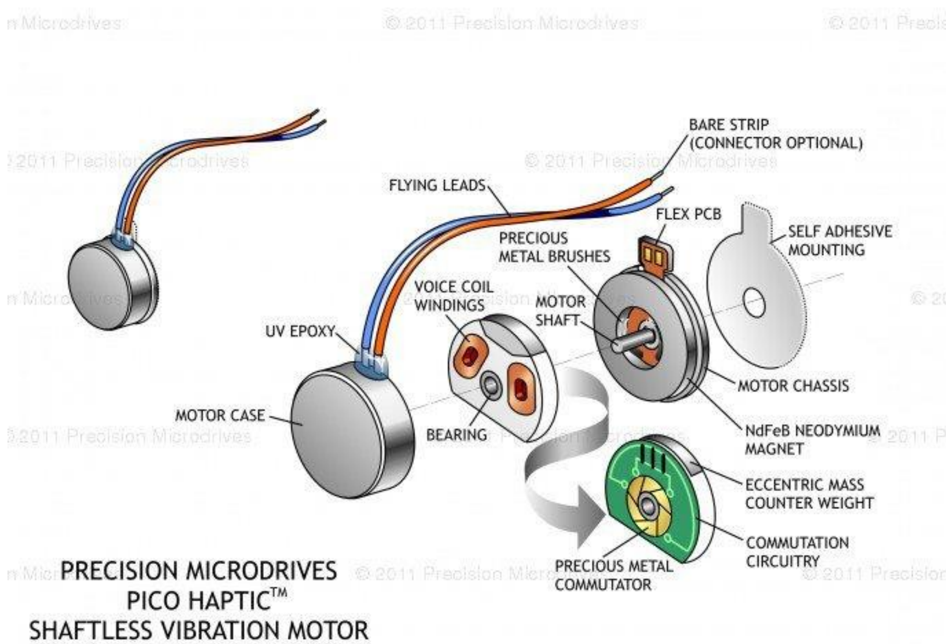


Figure 21. ERM Vibrator

4.2.3.2. Component Specifications

While ERM motors can vary in size and shape, the size and enclosure constraints of the Batpack narrow options considerably. The two smallest and most easily mounted designs our team found are the 'coin' vibrator and the 'pager' vibrator. Functionally, these two designs are very similar. However, the 'coin' vibrator has the added benefit of enclosing all of its moving parts, while the 'pager' vibrator exposes it's eccentric mass. For the sake of simplicity, and to avoid the added obstacles that come with exposed moving parts, we have selected the ‘coin’ design. Provided in the graphic below are the technical specifications for the specific vibrators in our design, and an image of the component.

Item Weight	0.2 ounces
Product Dimensions	4.1 x 0.4 x 0.1 inches
Power Draw	DC3V/0.1A 1.5V/0.05A
Size	10x2.7mm 5 Pcs
Material	Plastic, Metal

Figure 22. Vibrator Specifications

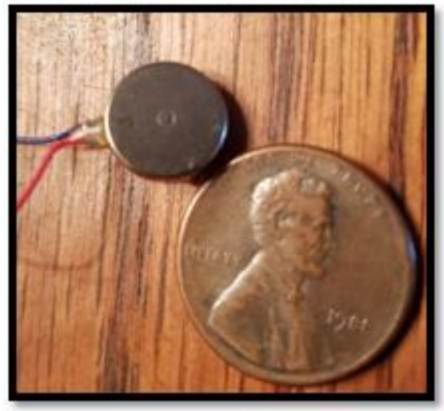


Figure 23. Vibrator with Penny

4.2.3.3. Pulse Width Modulation

Different vibrational intensities and patterns will be used to communicate different messages to the user. Because the vibrational force is a function of the voltage (and power) supplied, it is necessary that we can modulate the voltage to produce different vibrational strengths. Using a microprocessor such as the one on the Arduino, the industry standard way of accomplishing this is with Pulse-Width Modulation.

Pulse Width Modulation, or PWM for short, is a method of modulating a digital signal such that it behaves much more like an analog signal. The goal of this modulation, as previously mentioned, is ultimately to control the power provided to a component.

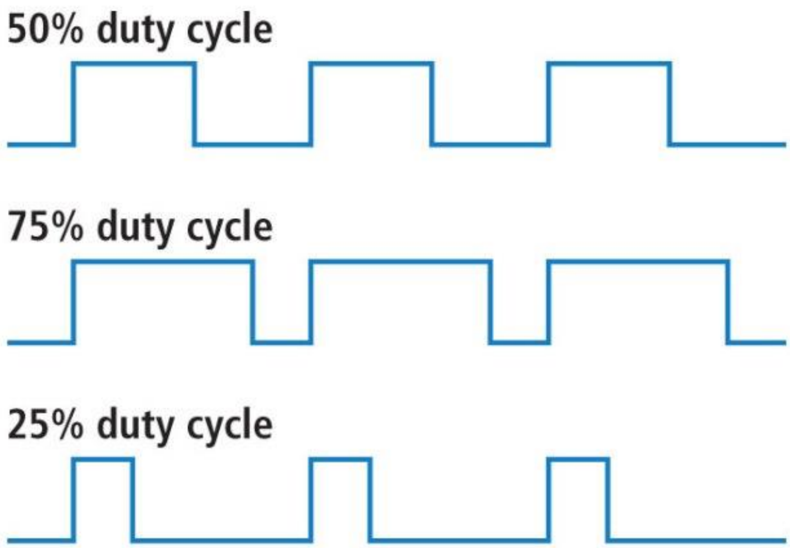


Figure 24. Pulse Width Modulation

To understand PWM, first let us consider a simple digital signal, which is either ‘on/high’ or ‘off/low’. In the specific example of the vibrators, high would be considered 3 Volts, and low would be considered 0 Volts. PMW drivers turn the signal on and off very quickly at different frequencies. When the signal is high, that is considered a pulse. The duration for which the signal is high is considered it’s width. By varying the frequency and duration of the pulse, the PMW driver is effectively modulating it’s width; hence why it is named Pulse-Width Modulation. Note that when speaking of PWM, duty cycle is typically referred to as the percentage of the time the signal is high for a constant time. In this context, duty cycle is analogous to pulse width. For a visual explanation, refer to the above figure.

While duty-cycle and pulse width are very similar measures, it is much easier to talk about the average delivered voltage in terms of duty cycle. In fact, this relation yields a linear relation in the form of:

$$V_{OUT} = DutyCycle \times V_{IN}$$

It is also important to note that once the signal is applied to the motor, it is filtered by the motor’s own internal low pass analog filter. This filter is actually a natural result of the DC resistivity. Precision Microdrives, a manufacturer of DC motors similar to those used in the Batpack, provides the following graphic to help demonstrate this effect.

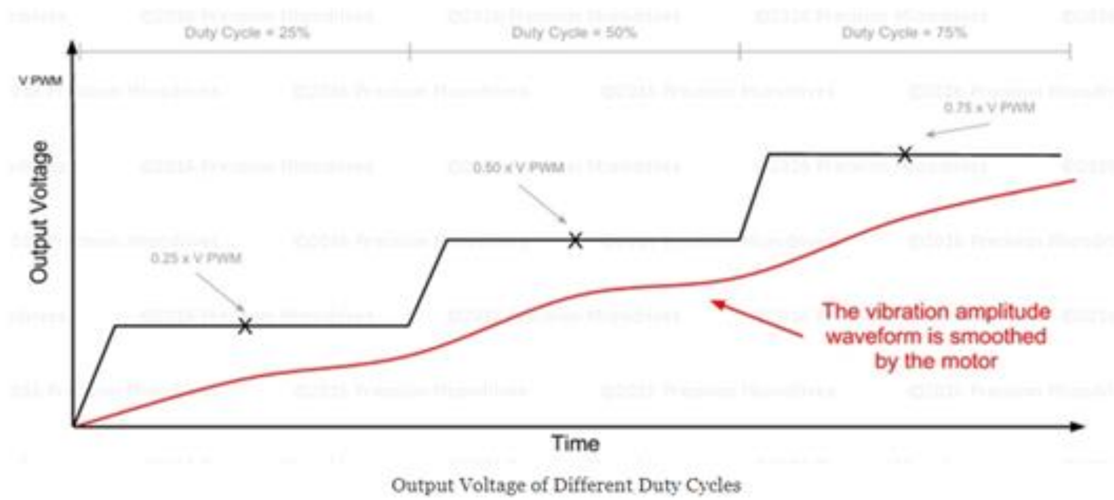


Figure 25. Pulse Width Modulation Voltage Graph

4.2.4. Microcontrollers

In short, the microcontroller(MCU) is a small computer. It will be used to power the PBC, which is supplying power for the vibrators and sensors and as well as camera recognition. There are a few considerations to make when choosing the right MCU.

- Low power vs performance
- Number of I/O required
- Memory required
- Development environment and support
- Computation power
- Cost

Considerations:

- Arduino Uno - One of the most common MCU's on the market. The Arduino comes with a large community and development environment ready to go. It holds a 32KB flash memory with a clock speed of 16MHz.

- MSP430FR5994 LaunchPad - Texas Instruments offers some very well designed boards, with low power rates. This board runs on a 16-bit RISC architecture with 16MHz systems clock. Including a 256KB of FRAM and 8KB of SRAM.

- Raspberry PI 3 Model B - This microcontroller has built in Wi-Fi and Bluetooth. Additionally, it consists of a CSI camera port, 40 pins, 1GB of RAM and a quad core 1.2GHz 64-bit CPU.

- Raspberry PI Zero W - This is the smallest of all the microcontroller considered. Built in Wi-Fi and Bluetooth allows the project to wireless later on, as it is not part of the specifications of the current project. It comes equipped with a CSI camera connector, 1 GHz single core CPU, 40 pin header and 512MB of RAM.

The Arduino Uno and MSP430 both offer low power, enough I/O pins, but have a small memory capacity and slow clock rates. Camera recognition and as well as powering other devices will rely on these two cons, so we will eliminate the use of them in the design. Testing will be completed with the Raspberry 3 and Zero. Both microcontrollers offer Wi-Fi, Bluetooth, CSI camera connector port, 40 pins. They differ greatly in memory and computing speed. The Raspberry 3 has about double the RAM, slightly faster clock rate but includes four cores instead of only one. This will increase performance dramatically. The main point of focus when testing the two is the response speed of the device, checking if the Zero will be strong enough to handle all the tasks, and how it ranks up against the Raspberry 3's performance. The following table will allow us to look more closely at the differences in the two MCUs.

	Raspberry Pi 3	Raspberry Pi Zero W
Processor	Quad core 1.2 GHz 64 bit	1 core 1 GHz 32 bit
RAM	1GB	512MB
Bluetooth	Yes	Yes
Wi-Fi	Yes	Yes
Pins	40	40
USB ports	4	2
CSI camera port	Yes	Yes
Micro SD port	Yes	Yes

Table 2. Comparison of Pi3 and Pi Zero

4.2.5. Camera Part Selection

The camera attached to the Batpack will be used for the computer vision aspect of the project. When comparing different cameras the Batpack could use we were looking at the following specifications: chipset, resolution, and frames per second. Our research lead us to comparing the Raspberry Pi Camera Module V2, Arducam 5 Megapixels 1080p Camera, and Waveshare RPi Camera (I).

The Raspberry Pi Camera V2 comes with excellent support when using it on a Raspberry Pi board, since it was made by the same manufacturer. Raspberry Pi upgraded the chipset on the V2 to a IMX219, offering an 8-megapixel sensor with the same. This in tandem with shooting 1080p at 30fps will allow the computer vision algorithm to properly poll and processes the images for user. This is priced at $29.99.

The Arducam 5 Megapixel camera uses the OV5647 sensor and has a 5-megapixel camera and can shoot up to 30 frames per second at 1080p. This camera is priced a little lower than the Raspberry Pi Camera V2 at $16.99.

Lastly on the list of the Waveshare RPi Camera (I), which uses the same sensor as the Arducam. The Waveshare cameras main difference is it's field of view at 170 degrees, where the Arducam only provides roughly a 54 degree angle.

The following pictures show the difference in quality each of the three cameras provide.

-

Figure 26. Camera Comparisons (Courtesy of Semifluid.com)

4.3. Software Design Research

This section will cover the algorithms and software needed to have the Batpack meet the requirements we set out to satisfy. Each aspect of the software design is covered for the components that allow the Batpack to function and detect any obstacles the user might run into in normal environments.

4.3.1. Operating System

Using computer vision and the OpenCV libraries to convey the world around the user will be processed on a Raspberry Pi v3. A main component that needs to be considered when running the code on the Raspberry Pi is the operating system.
A real-time operating system (ROS) was taken into consideration. This would increase the overall performance of OpenCV. However most of the ROS for the Raspberry Pi were not documented well and have little to no tutorials on how they operate. Narrowing down our choices we were between Arch Linux and Raspbian. Both operating systems are a Linux kernel and will be able to use the OpenCV libraries. Considerations that we took in while researching the best OS to use were: Boot time, background processes, and support.

- Boot time is important to the user as the faster the product boots and starts operating the better the overall user experience will be. Arch Linux is able to boot in less than 10 seconds in most cases. Where Raspbian usually boots in less than 20 seconds.

- Unneeded background processes will allow the CPU to use most of its power to run the processes important to the user experiences. You are able to get as barebones as possible with Arch Linux. However, Raspbian also provides a “Lite” version of their OS. This “Lite” version does not provide any GUI to the user, which the Batpack does not need.

- Product support is one of the biggest considerations. Our team does not have a lot of experience with programming computer vision devices and will need all the help we can find when designing this subsystem. Arch Linux has a lot of support for desktop applications and general designing of the OS; but provides little support on running it on a Raspberry Pi. Raspbian provides a lot of support to run on the Raspberry Pi, as it was specifically built for the hardware. They also have support on how to get OpenCV up and running on the Raspberry Pi.

Our team has decided on using the Raspbian Lite operating system. We do not need a GUI for the Batpack, this will decrease our boot time of the device. The support that Raspbian offers will contribute immensely when designing the Batpack.

4.3.2. Optical Algorithm

The purpose of the optical algorithm is to use computer vision to facilitate mobility on sidewalks for the visually impaired. More specifically, the algorithm seeks to detect the edges of the sidewalk, and notify the user if they veer too much from the path. However, while there have been rapid advances in the field of computer vision in recent years, there is still no (open source) foolproof algorithm to do this. As with any Computer Vision algorithm, there are several considerations which may hinder the algorithms efficacy, including but not limited to:

- Reflective surfaces in the camera view.
- Time of day, which may affect the lighting condition.
- Different patterns which may appear in different sidewalk constructions.
- Reflective materials or other obstructions in the camera view.
- Camera lens specifications, which may alter the image.

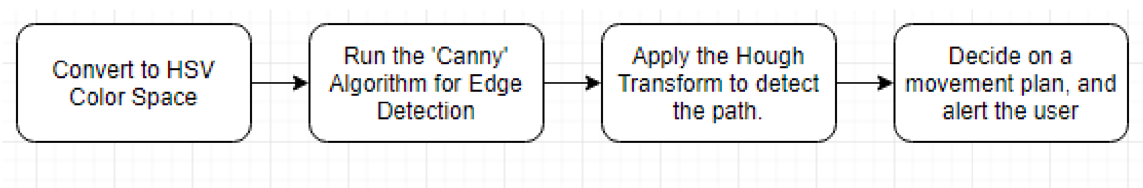


Figure 27. Optical Sensor Vibrator

To combat these hindrances, the algorithm will make use of the most modern computer vision filters and algorithms available to the public. An estimated flowchart of these filters and algorithms is provided to the right. As can be seen, the first step will including converting to the HSV color space and generating masks based on color. This will help reduce the effect that lighting saturation has on the image, and will allow us to filter based only on color and not level of brightness. Next, the Canny algorithm will be run to find large changes of gradient in the image. If the image was masked correctly in the previous step, then it should be easy to detect where the sidewalk ends and the grass begins. To filter out other noise and detect the both straight edges of the sidewalk, the Hough transform is applied to the output from the Canny Algorithm, and this final step will allow us to see the angle and direction of the sidewalk. With this information, we are confident that the device will be able to make it's best attempt at providing the user with a walkable path.

The following is a more detailed explanation for the aforementioned algorithms, complete with prototype screenshots and explanations.

4.3.2.1. HSV Color-space

Modern computers most often store pixels in images as a series of three values representing the amount of Red, Blue, and Green, in that pixel. Together, these values add up to a specific color. The HSV colorspace takes this idea and expands upon it. From the values of Red, Blue, and Green, (RGB) values are calculated which represent Hue, Saturation, and Value (HSV). To understand what this means, it's important to first define these terms. These definitions were taken directly from "Introduction to Color Theory" a web publication presented by the New Mexico Institute of Mining and Technology [8]:

- The hue (H) of a color refers to which pure color it resembles. All tints, tones and shades of red have the same hue.
- Hues are described by a number that specifies the position of the corresponding pure color on the color wheel, as a fraction between 0 and 1. Value 0 refers to red; 1/6 is yellow; 1/3 is green; and so forth around the color wheel.
- The saturation (S) of a color describes how white the color is. A pure red is fully saturated, with a saturation of 1; tints of red have saturations less than 1; and white has a saturation of 0.
- The value (V) of a color, also called its lightness, describes how dark the color is. A value of 0 is black, with increasing lightness moving away from black.

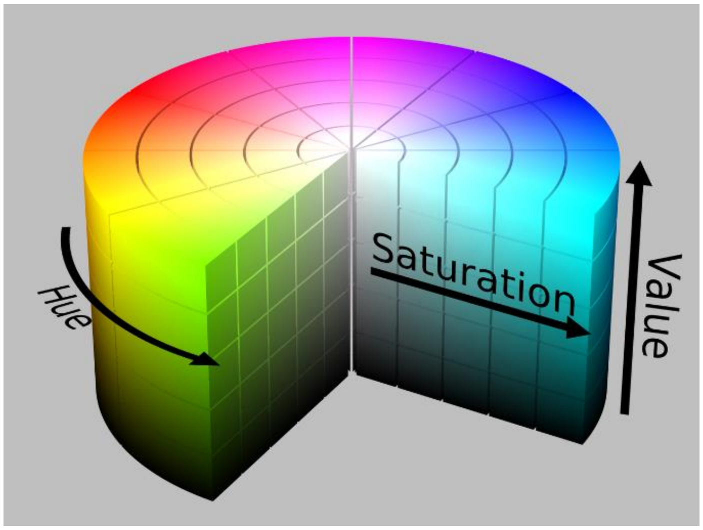


Figure 28. HSV Colorspace

Graphically, the HSV color space can be thought of as a cylinder. In this cylinder, the angle around the top is representative of the Hue, the distance out from the center the saturation, and finally the height represent value. For a visual explanation, see the image above. Additionally, while many modern computer vision libraries contain code which convert RGB to HSV and vice-versa, it may be useful to calculate these values by hand. From the Wikipedia article on HSV color space, we find the following formulas [8].

For calculating Chroma (an intermediate value):

$$M = \max(R, G, B)$$
$$m = \min(R, G, B)$$
$$C = M - m$$

For calculating Hue:

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \frac{G-B}{C} \mod 6, & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases}$$
$$H = 60^\circ \times H'$$

For calculating saturation:

$$S_{HSV} = \begin{cases} 0, & \text{if } V = 0 \\ \frac{C}{V}, & \text{otherwise} \end{cases}$$

And finally, for calculating Value:

$$V = M$$

For our purposes the main benefit of the HSV color space comes from it's ability to represent color independently of brightness. As mentioned before, reflective lights and varying levels of brightness can cause discrepancies which make it very difficult to filter the image based on color. However, since the brightness of a pixel has been isolated to the Value parameter in HSV, we can create masks which are functions of only *hue and saturation.* To demonstrate this concept, a prototype of the final algorithm was written. Images from this prototype are provided below starting from the left is the original image, then the HSV representation, the generated mask, and finally the masked image.

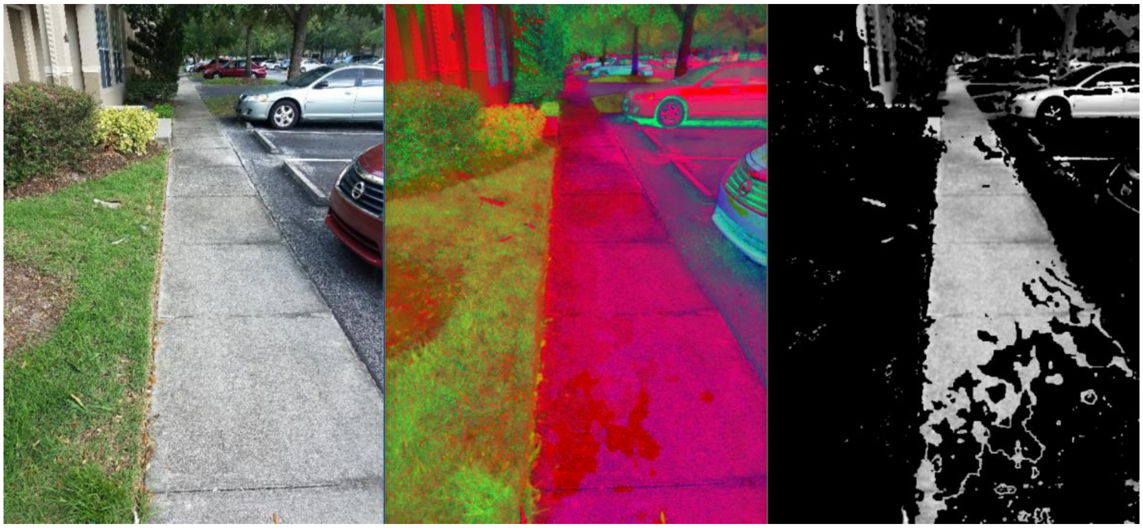


Figure 29. HSV color space masking

This particular mask was designed to only pass values which are the same shade as the sidewalk. While this does provide a somewhat noisy mask, it does do a decent job of isolating the sidewalk and removing extraneous objects. However, in the future, this kind of masking may be better applied to different objects. For example, if we know that there will always be grass on the left side, then perhaps the resulting mask would be less noisy if we filtered based on green instead of grey. This kind of design consideration requires more research into the domain of this problem, including the specific images properties produced by the camera component.

4.3.2.2. Canny Algorithm

According to the OpenCV documentation on the Canny algorithm, the Canny Edge Detector Algorithm was first developed by John F. Canny in 1986 [9]. Since then, the Canny Edge Detector algorithm has gained infamy as the *optimal detector.* In fact, the algorithm's main motivations were to achieve optimization with these three criteria:

- Low error rate: The algorithm seeks to detect only existent edges, and minimize the number of false positives.
- Good Localization: This means that the algorithm is accurately locating the edges - meaning that reported edges are as close as possible to where the real edges lay.
- Minimal response: The algorithm having minimal response essentially means to use that a single edge is not reported as two or more distinct edges.

These qualities are essential to any edge detection algorithm we would use. A high error rate or bad localization would cause our users to vear of the path - if the algorithm were to even find a path.

To accomplish these criteria, the Canny operator functions by essentially generating a map of the intensity gradient of the image. The following is a brief, undetailed explanation of this process [9]. First, each pixel has a convolution matrix applied of the form:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Then, the direction and strength of the gradient is calculated using the following formulas:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

After this mapping has been generated, a set of simple rules decide whether a pixel constitutes an edge or not. First, the gradient strength must be within two threshold parameters which are supplied to the algorithm. The two threshold parameters are known as the lower and upper thresholds. Finally, for the values within the threshold parameters, a pixel is considered an edge if and only if it is connected to a pixel which is above the upper threshold.

To demonstrate the efficacy of the Canny Edge Detector Algorithm, our team has run the OpenCV on the masked image produced in the step above. The resultant edges are displayed in the below graphic.

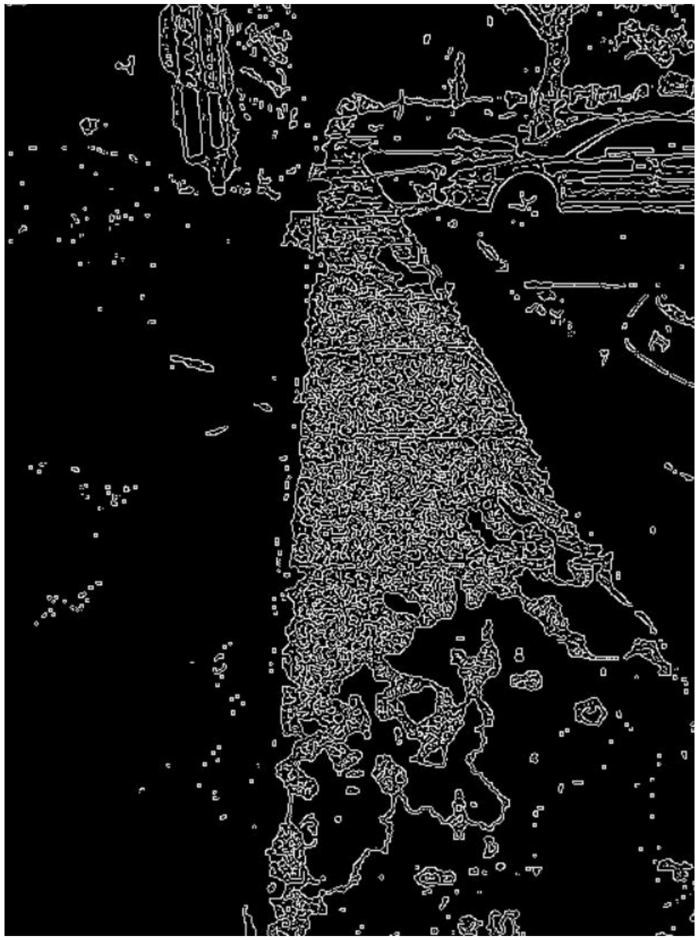


Figure 30. Canny Edge Algorithm

Admittedly, this is quite the noisy image - even after removing extraneous objects with the mask. Moving forward, our team will research other methods of smoothing out the image. For example, the OpenCV documentation also recommends running images through a Gaussian Blur before the Canny operator. This is done to smooth out noise and help reject false positives. However, the Gaussian Blur represents even more additional parameters that will need tuning.

4.3.2.3. Hough Transform

Finally, the last step in our algorithm is the Hough Line Transform. This algorithm takes as input a list of points, and as output reports lines which may exist inside the provided list of points. The algorithm takes advantage of some very basic mathematical concepts to achieve this. Essentially the Hough Line Transform generates a complete set of all candidate lines for each point, and then finds lines for which the most amount of points exists [10]. A more detailed explanation is provided below:

1. First, a complete set of candidate lines is generated for each point. While at first this may seem like a daunting, or even impossible task, the Hough Transform makes short work of this task by utilizing the polar coordinate system. Simply put the x and y coordinates into the following formula:

$$r_\theta = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta$$

Plotting all values for theta (between 0 and 2π) yields a sinusoidal curve that represents the family of all lines that pass through the point 'x,y'. This is pictured in the below image:

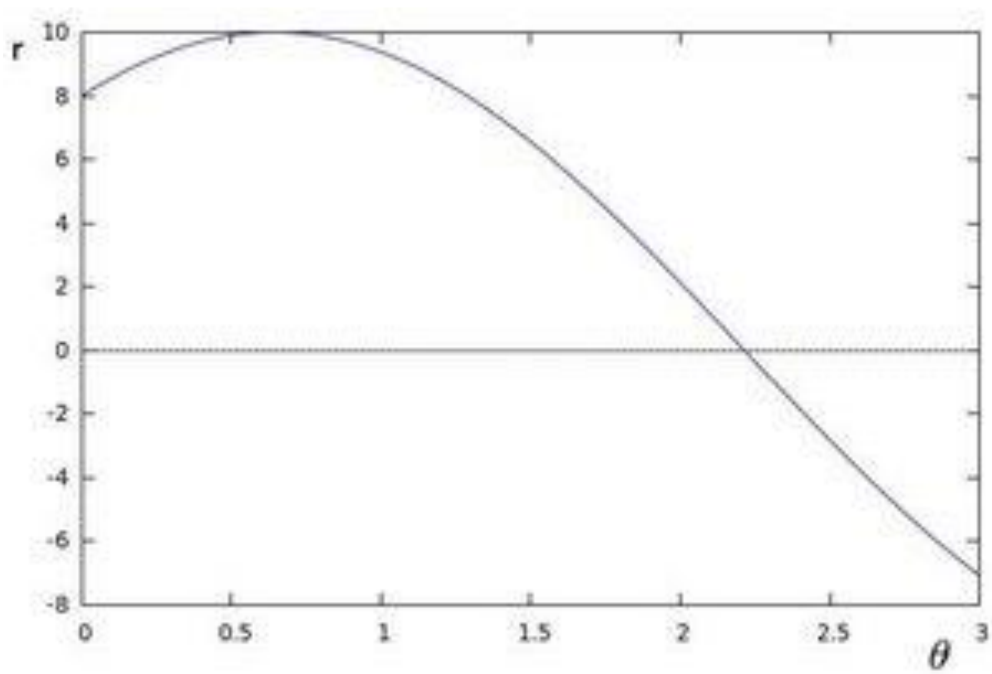


Figure 31. Family of Lines Sinusoid

2. When this process is done for all lines, the sinusoidal curves can be analyzed to find intersections. Each intersection between two curves represent a single line that those points can share (also pictured below).

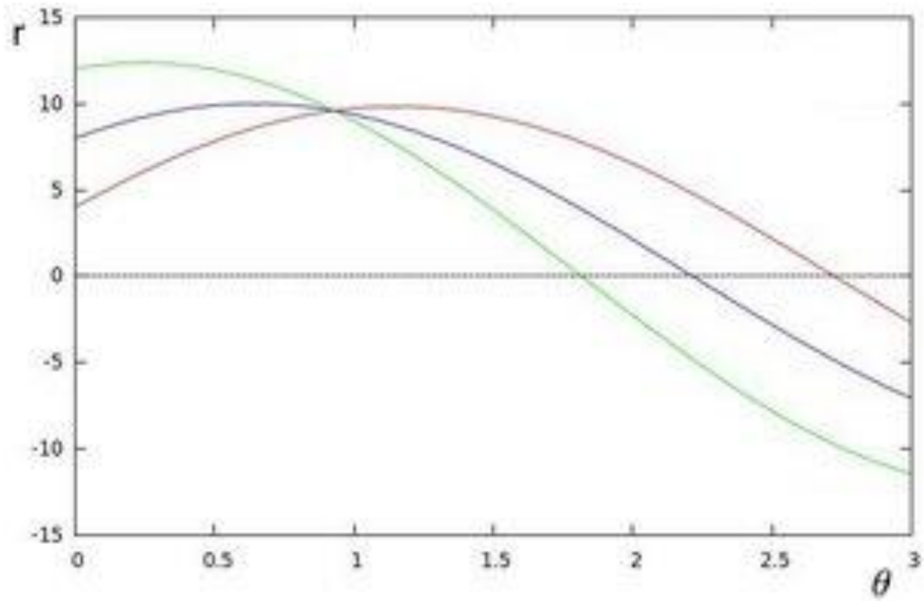


Figure 32. Intersection of between Families of Lines

3. Finally, the last step is to simply count these intersections. The idea is that the larger the amount of intersections centered around a point on the r-theta plane, the more points that exist on a single line. A threshold is provided to the algorithm which filters out lines below a certain amount of intersections.

Pictured below is the result of running the Hough Line Transform on the points provided from the Canny algorithm.

Figure 33. Hough Line Transform

In this image, we see many green lines pointing in the direction of the sidewalk. This prototype image, while nowhere near performant enough for the final product, definitely showcases the robustness of this approach. Because the results of the Canny operation on the previous image were so noisy, this algorithm had very limited data to work with. However, even with the large amount of noise, and scarce amount of 'real' edges to work with, the algorithm was able to find a set of lines that traversed the sidewalk. Further tuning will be necessary to find the correct parameter values that will reject false positives while still finding the proper lines. Once all of these lines have been found, navigating with the user will be as simple as finding the direction of the line and communicating this information to the user.

4.3.2.4. Otsu-thresholding

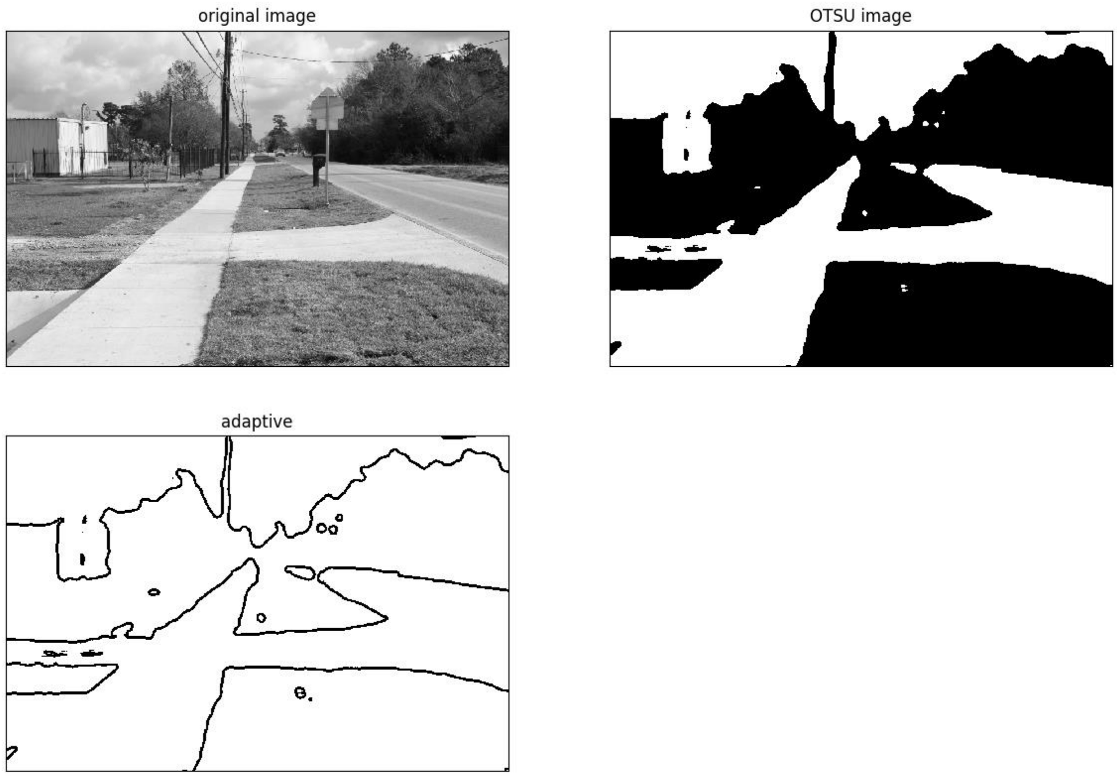


Figure 34. Otsu-thresholding on sidewalk

Otsu's thresholding method is a technique of image segmentation. The basis behind this method is to find the threshold value, 0 to 255, of an image that minimized the the weighted within-class variance [30]. This technique must be done on a grayscale image since the Otus threshold is computed with a histogram. The histogram must be bimodal for the threshold value to be computed with accurate and workable results. Any type of smoothing techniques will allow the noise of an image to go away and will give the Otsu's thresholding method a proper bimodal histogram. In the Figure above a median blur filter was applied in order to eliminate this type of noise.

To find the threshold value Otsu's method runs through the threshold values, 't', from 0-255 and choosing the value that minimizes the result. [30]

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Class probabilities are estimates are defined as

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

Class means are defined as:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

This helps calculate the individual class variances.

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

4.3.2.5. Camera Calibration

Motivations

If it were not for the rapid advances in optical technology, this project would not be possible. Indeed, recent times have seen an ever-increasing explosion in availability for ever-smaller and ever-cheaper cameras. Most notable amongst these cameras is the pin-hole camera whose small size and cost has made it a perfect candidate for modern electronic devices – including a wide array of phones, laptops, security cameras, and even the Batpack. As previously discussed, the Raspberry Pi Camera, one such pinhole camera, is our weapon of choice for attacking the Computer Vision aspects of this project.

However, while small in cost and size, the Pi Camera unfortunately suffers from the same disadvantages as most pinhole cameras. The most relevant disadvantage in the context of this project is the large optical distortion caused by the camera. Wikipedia defines optical distortion as “a deviation from rectilinear projection”. Put into layman’s terms, this means that some areas of the image coming from the camera will be magnified, while some areas of the image will be compressed. The term ‘rectilinear’ comes from the fact that lines which are straight in the real world will appear curved in cameras which suffer from optical distortion.

As one can imagine, because our optical algorithm is relying heavily on the camera to calculate position and heading of the sidewalk, this presents a giant problem for the Batpack. Fortunately, the field of optical distortion is a very active field which has been heavily studied and documented. Experts in the field have pinpointed not only what causes optical distortion, but have also detailed the most efficient manners in dealing with this problem. In fact, open source software exists to calculate and undistort images coming from any camera that has been properly calibrated. The following section will provide a short summary of different types optical distortion, along with industry standard solutions.

Different Kinds of Distortion

The major types of optical distortion are tangential distortion and radial distortion. Tangential distortion occurs because the plane of the alignment between the image taking lens and the image taking sensor (pictured below). This kind of distortion causes some objects to appear closer in the image, and simultaneously causes others to appear further [11].

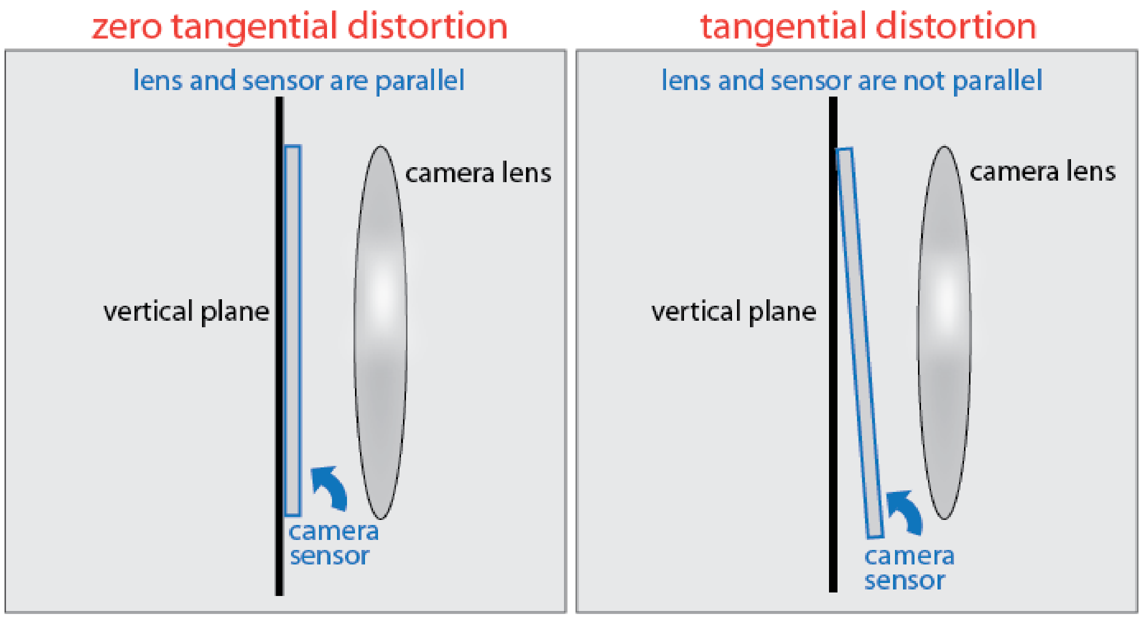


Figure 35. Tangential Distortion

Radial distortion, perhaps the most noticeable of the two types of distortion, is a direct cause of the focal length and general shape of the camera lens. In particular, the symmetric shape of most modern camera lenses has been shown to produce a quadratic relation which increases the distortion in direct correlation to the square of the distance from the center of the image – hence the ‘radial’ in ‘radial distortion. Most people have experienced this phenomenon while using binoculars or cameras with high-zoom lenses. Furthermore, this kind of distortion usually manifests itself either in what is known as Barrel distortion, Pincushion distortion, and Mustache Distortion.

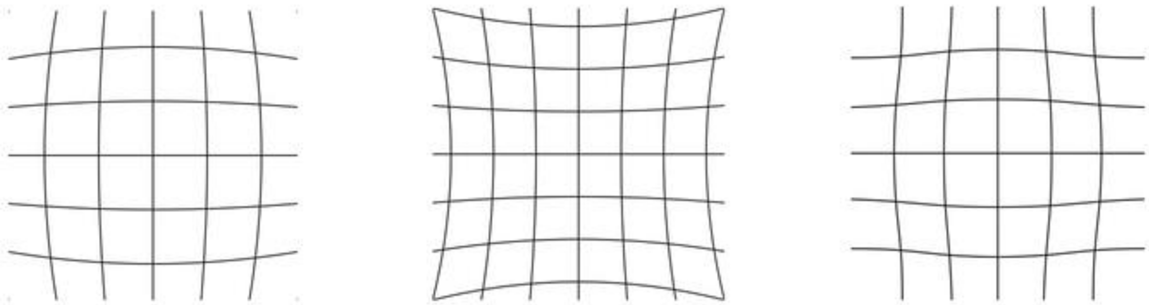


Figure 36. Barrel, Pincushion, and Mustache Distortion

Solution

One of the most common and widely implemented methods of undistorting an image is the Brown-Conrady model [12]. The Brown-Conrady model takes as inputs measured distortion coefficients, which represent the distortion caused by the camera lens, and uses this information while iterating through each pixel to apply a correction transformation. For the radial distortion, the model is described as follows:

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

For the tangential distortion, the model is described in similar fashion:

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

In this model, ‘x’ and ‘y’ represent the pixels coordinate in the image. The ‘k’ parameters are the radial distortion coefficients, while the ‘p’ parameters are the tangential distortion coefficients. Lastly, ‘r’ is calculated as such:

$$r = \sqrt{(x_{\mathrm{d}} - x_{\mathrm{c}})^2 + (y_{\mathrm{d}} - y_{\mathrm{c}})^2}$$

Implementation

Fortunately, the distortion coefficient and other intrinsic and extrinsic properties of the camera can be calculated and saved for future use. In fact, to make matters even better, OpenCV provides an implementation of the Brown-Conrady model in its Camera Calibration Module. The following demonstrates the steps taken to calculate the distortion matrix, and demonstrates the effects of undistorting the image [13].

The first step in running the algorithm is finding the distortion coefficients. Using OpenCv, the preferred method for accomplishing this is as follows:

1. Print a checkboard and measure a checkerboard grid. These measurements will be later used to calculate distortion coefficients.

2. Place the checkerboard against a hard surface and ensure that it is straight.

Next, OpenCv uses the relation between the real-world measurements of the squares to calculate the distortion coefficients. Pictured left is an image of a successful run of the algorithm, with the corners found. With distortion coefficients found, we can begin to undistorted images. Below is pictured an undistorted image using the metrics calculated.

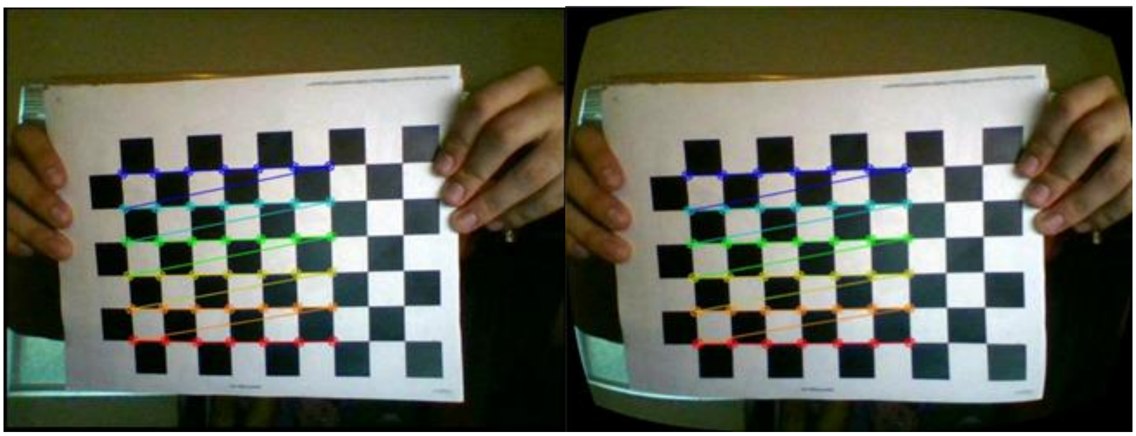


Figure 37. Distortion Correction

4.3.2.6 Comparison of Different Sobel Methodologies

While the Canny Edge detector is a very robust algorithm for edge detection, it is definitely not the only way to accomplish edge detection, and for our usage, it may be too general purpose. As mentioned previously Canny relies on the calculation of gradient to detect edges. This process is also known as the Sobel detection, and can be accomplished in many different ways. This is especially true when one considers all the different methodologies for filtering and expanding upon the output of Sobel detection. This section takes advantage of code readily available online to explore these different methodologies for accomplishing edge detection, and discusses each one as a viable alternative to Canny.

Split Axis Filtering on Magnitude

This Edge Detection method differentiates itself from Canny and other edge detection methods in the way it calculates the gradient of each edge. As mentioned before, Canny runs two convolution matrices at each pixel – one for the x axis and one for the y axis. At each pixel this yields two values which both colloquially represent the strength of the gradient in the x axis and the strength of the y axis. Then, again at each pixel, a gradient for x and y axis are calculated simultaneously by finding the magnitude of the vector created by both strengths.

However, in this method, the gradient is calculated using only one axis at a time! The convolution matrix is run at each pixel similar to Canny, however the gradient is then calculated simply by taking the absolute value of the convolution matrix. After the gradient has been calculated at each location, pixels which fall within a provided threshold are considered edges [14]. This methodology has many benefits over Canny. The first is that because we are only filtering on the magnitude of the gradient, it is much easier to tune this algorithm because the only parameters needed are the upper and lower bounds for the magnitude filter.

Additionally, because of the nature of the images we receive from the camera, we are most interested in large gradients which occur across the x axis. This is mostly likely due to the fact that the sidewalk will cause a vertical line across the image. This vertical line will undoubtedly cause high changes in gradient as the texture and color of the grass contrasts with the texture and color of the sidewalk. To test that this assertion is indeed accurate, the algorithm was run on the same image which was previously run on the Canny algorithm.

Additionally, pictured below is the result of running a probabilistic Hough Line Transform on each set of detected edges. As is shown in image, this method of edge detection causes much less noise in the detected edges. And as can be expected from a less noisy image, we receive much better output from the Hough Line Transform.

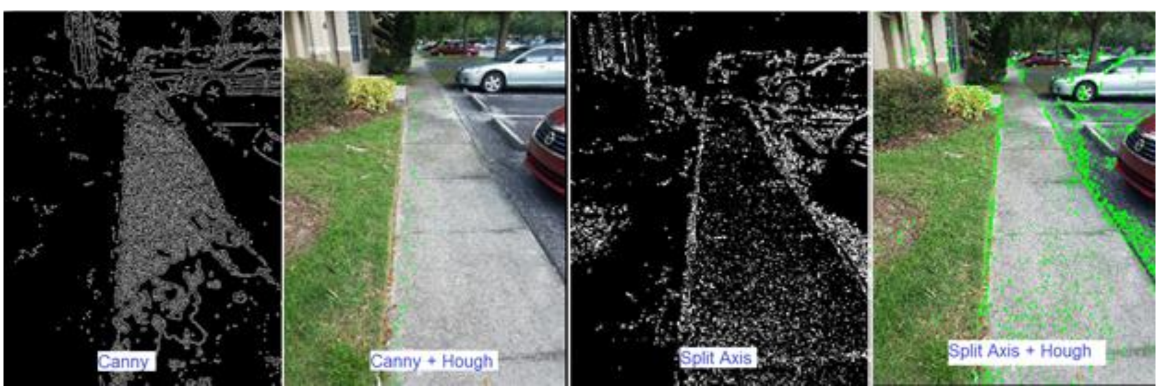


Figure 38. Canny Vs. Split Axis

Filtering on Gradient Direction

An alternative to using the magnitude of the gradient is generating a filter based on the direction. This method produces a much noisier result – however it has the added benefit of capturing the direction of edges in the image. While the result of this filter may be too noisy to use directly, it's advantage in finding direction could potentially be used in conjunction with other methods [14].

Below is provided a comparison of Canny and the direct output of this filter. Additionally, the results of running the Hough transform on this image are added to showcase the strength of each algorithm in the context of this application. As can expected, the result of running the Hough Transform on the output of this edge detector is quite chaotic. However, even with the immense amount of noise, there is still more information about the direction of the sidewalk when compared to the same output on the Hough Transform. While this result alone is still not good enough to find the heading, it could potentially provide an improvement when combined with other edge detection methods.

Figure 39. Canny Vs. Gradient Direction

Combination

In this approach, we combine the results of the last two edge detection methodologies. Admittedly, as we combine different methodologies for edge detection we begin getting closer and closer to the implementation of Sobel Detection done by Canny. However, the combination method still has many differences which have proven very advantageous as compared to Canny. To name one such advantage, not relying on the Canny algorithm increases the amount of control we have over each step in this process.

This makes it much easier to fine tune parameter values, and additionally allows a much higher level of customization. In this combination approach, we combine the last two methodologies in the hopes that this will yield a stronger result. Again, the results are displayed below in conjunction with the output of the Hough Line Transform. So far, that has proved the best method for edge detection.

Figure 40. Canny vs. Combination approach.

As can be seen above, this produced a much cleaner, less noisy result as compared to Canny – while still managing to retain the sides of the sidewalk! As predicted, the Hough Transform was also much more successful in defining lines which outlined the sidewalk. The lines on the right side of the image appear very strong and carry on for almost the entire length of the image. Admittedly, the algorithm is much less successful on the left, where the lines appear much weaker, and additionally stop closer to the halfway mark of the image. However, once again if we compare these results to the results of the Canny algorithm, it becomes plain that this is much more performant.

Conclusion

In the future, it may be advantageous to create a hybrid approach by utilizing the combination method and then applying the same filtering techniques on the output which are employed by Canny. However, a cost benefit analysis would have to be done to determine if the added complexity justifies these methods. As can be seen here, even without Canny’s filtering techniques, we still receive relatively much less noise using this approach. While employing those filtering techniques would hypothetically decrease the noise even further in the image, there is such little noise to begin with that at this point that it may be completely unnecessary.

Moving forward our team will begin utilizing the combination method of edge detection. It is our hypothesis that this method will be sufficient for finding the direction of the sidewalk, and in turn ensuring the user stays within its bounds.

4.3.3. Neural Networks

4.3.3.1. Motivations

As the cost of processing power continues to shrink, and especially with the wide availability of GPU’s, Neural Networks have become a recent hot topic in the field of artificial intelligence. Indeed, Neural Networks have grown so much in popularity that they are being employed in such varied areas as Speech Recognition, Natural Language Processing, and even Computer Vision. In fact, it is their success in Computer Vision applications that may make Neural Networks an ideal candidate for the Optical Algorithm. However, while Neural Networks definitely do excel in certain applications, there are many considerations to be made if they are to be used in the Batpack. Provided below is an explanation of what Neural Networks are, and how they could potentially be used in this project – especially in contrast to other Computer Vision Methods.

4.3.3.2. How Neural Networks Work

As their name imply Neural Networks are inspired by the basic structure of the brain. At it's most basic form one can think of the brain as a collection of neurons connected across billions of synapses. An analogy for the brain, Neural Networks contain collections of artificial neurons connected to each other in similar fashion. The most basic unit for the Neural Network, the neuron so to speak, is the perceptron. The perceptron was introduced by Dr. Frank Rosenblatt in 1958, in his famous paper "The Perceptron: A Probabilistic Model for Information Storage And Organization In The Brain", where he first described perceptrons as circuits which could be modeled by certain probabilistic functions [15].

While modern times have seen a huge increase in the study and development of perceptrons and the relation between networks of perceptrons, the same basic model developed in 1958 still applies today. The following provides an overlook of the perceptron algorithm [16].

1. Begin with a set of training examples. Perceptrons need to be trained, which means they must first be fed example data points with the correct desired output.

2. A set of weights, equal to the number of inputs for each training example, is initialized to some standard value. The initial value of the weights is actually not very important, so the weights are very commonly initialized with a random value, or simply one.

3. One additional parameter, called the bias, is also initialized to an arbitrary value.

4. A Net Input Function is used to calculate a numerical value from all inputs and weights. Most often, this is done simply by summing the product of each input with each weight. This is described mathematically below:

$$S = w_1 x_1 + \cdots + w_n x_n + b$$

In this context, signifies the weights of the data point. Similarly, denotes all the inputs with n equal to the number of inputs. ‘b’ is the value of the bias.

5. For classification problems, an activation function is run with the output of the net input function. The activation function simply decides whether or not the perceptron is ‘firing’. Very often, if is positive, the activation function will cause the perceptron to fire, and inversely if is negative, the perceptron will not fire. This step is slightly different in regression problems, where instead of a binary response, the activation method provides a continuous value as the result.

6. This step is where the perceptron accomplishes it’s ‘learning’. Because in the training phase the perceptron is given input values with the correct output values, it is able to calculate an error. A ‘learning rule’ is applied with this error to each of the weights. Provided below is a common representation of this ‘learning rule’.

$$\Delta w_j = \eta \left(\text{target}^{(i)} - \text{output}^{(i)}\right) x_j^{(i)}$$

7. Finally, the perceptron begins again at step four with it’s newly updated weights. This allows the perceptron to fine tune it’s weights until, hopefully, a high level of accuracy is reached. This process is repeated either for a certain number of iterations, known as epochs, or alternatively it can be stopped once a certain level of accuracy or precision is reached.

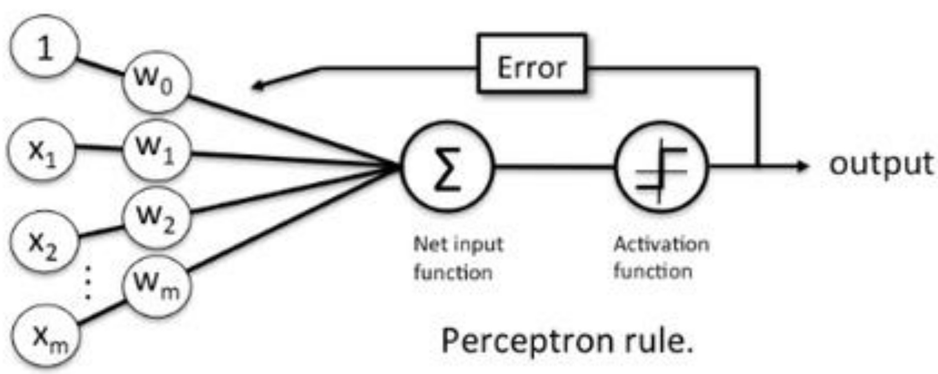


Figure 41. Perceptron function

From the perceptron, the jump to a neural network seems like a natural conclusion. A neural network is simply a collection of perceptrons, which lead into and take as inputs a different collection of perceptrons. A similar feedback loop is also put into place at the network layer to ensure all models are learning (updating their weights). This feedback loops begins at the output perceptrons, the perceptrons whose outputs are directly correlated to the outputs of the data points, and pass the error backwards to the perceptrons connected as inputs until there are no more perceptrons left. This process is known as backward propagation of error.

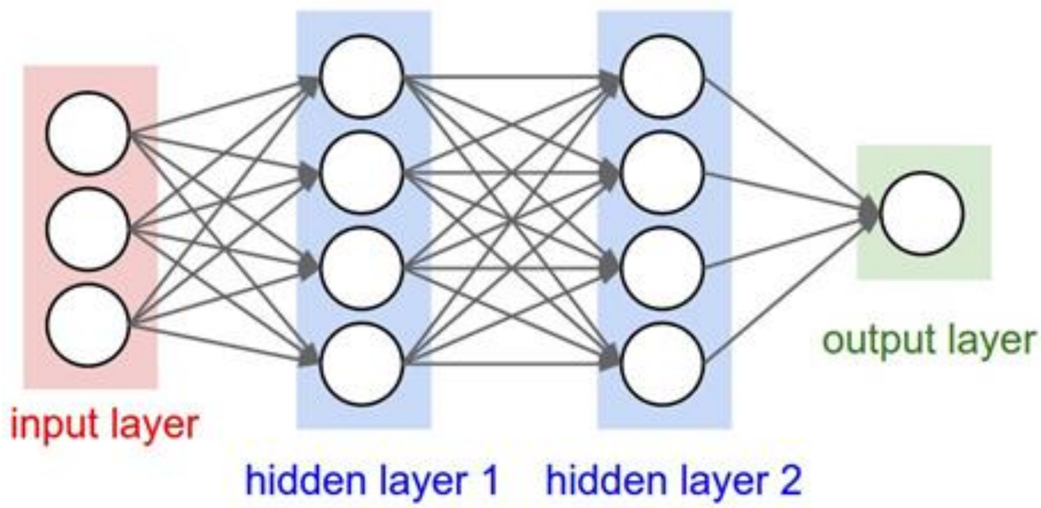


Figure 42. Neural Network

The large challenge of neural networks comes in designing the connections and relations between perceptrons. Pictured above is a very basic neural network, provided by the University of Stanford on an online learning page. The input layer, as it's name implies, represents each input. In this architecture, the inputs are fed into the first layer of perceptrons, whose outputs in turn are fed into the second hidden layer. Finally, the output from the second hidden layer is used to generate an output.

Neural Networks and Image Processing

As previously mentioned, we solve problems by connecting basic neurons. Machine learning works best when you have a lot of data to work with, as we can train it better. Pictures are much more complex than analyzing a single number, or is it? "An image is actually just a grid of numbers that represent how dark each pixel is. [17]" We can convert this to an array of numbers and treat each element of the array as an input node. On the output side of things, we want to have two different ones, either it is or it is not to help classify objects into groups. Now that we have that established, we simply just train the system for the specific image with images that "are" or "are not" it.

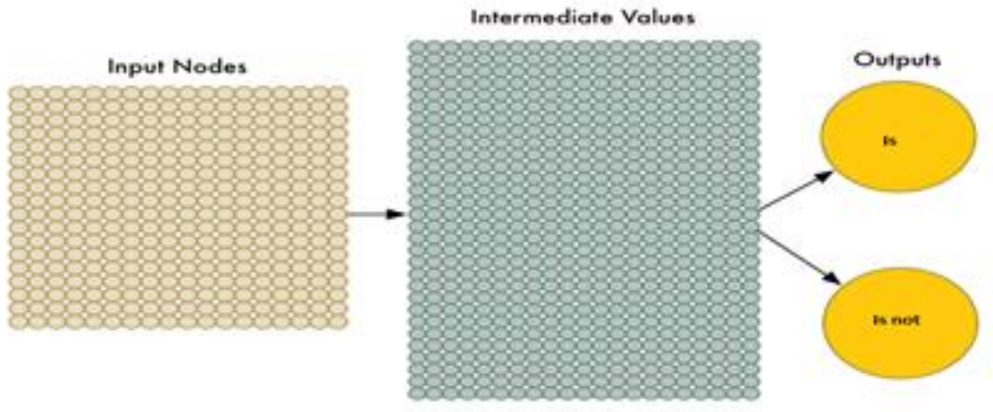


Figure 43. Neural Network Image Processing [17]

There are a few implications that arise though, as it is not this simple. Firstly, we must take into consideration of the position of the object in the image.

Method number one consists of searching with a sliding window, aka brute force. We simply scan through the image looking for the specific part we want to recognize. This method is inefficient, as we constantly check for objects of different sizes.

Method two is another brute force idea, a deep neural net with tons of data. Consider taking the object you want to recognize in an image and write a program to relocate it multiple times in different positions. Using this method, we could in theory create an infinite amount of training data, and we know, the more training data we have, the potential for recognizing the image increases drastically. But this also comes with a downfall, as more data makes it more difficult for our neural network to solve, but we can simply work around this by increasing the size of our neural network, allowing us to solve more complex patterns! Looking at the picture below, we take the input nodes and begin creating variations, do this multiple times and create multiple layers. *

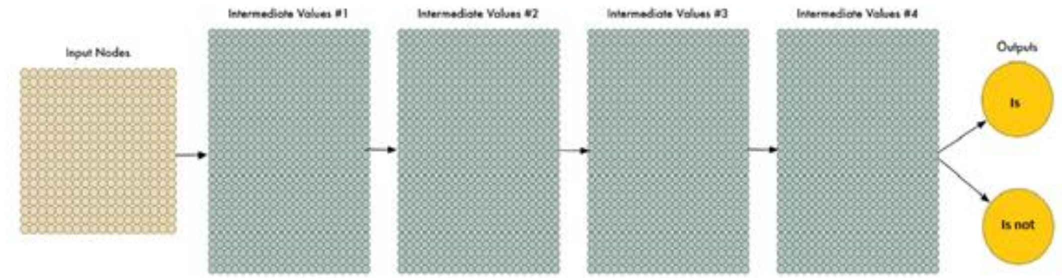


Figure 44. Neural Networks Convolution Layers [17]

This idea was originated back in the late 1960's [17] but wasn't very practical. This was because the neural networks get so large that basic computer processors can't work them in an efficient time basis. Thanks to improvements in 3d graphic cards, the game has completely changed. The graphics cards allow for training of the neural networks to be extremely snappy, bringing new advancements in machine learning. Though looking back on method two, it still isn't the best way to handle this process. We don't want to scan the image for the object we want based on location, but rather just recognize that the object is in the image as it this method will take loads of training to be efficient.

The answer is convolution. This takes the object and recognizes it for what it is, also known as translation invariance. This removes the dependency of the object based on location. Pictures have a "hierarchy or conceptual structure [17]" that we can pick up on right away. For example, take a look at this picture of Mike slaloming.

Figure 45. Mike Slaloming

We see a few things:

- The ground is the water he is skiing on
- There is a person in this picture
- The person is on one ski
- The ski is on top of the water “ground”
- There is trees in the background above the water

We intuitively know that a person is a person, regardless of what surface he or she is on, meaning we don’t have to relearn the idea of this person for all surfaces. But computers aren’t people (not yet at least), so how do we train them?

Firstly, take the image and split it into overlapping image tiles [17]. Using method one of sliding image, we break the image into small equal size image tiles. We then take each tile and put it into a small neural network with equal weights, making each tile equal. When we notice something different in the tile, we will mark the tile.

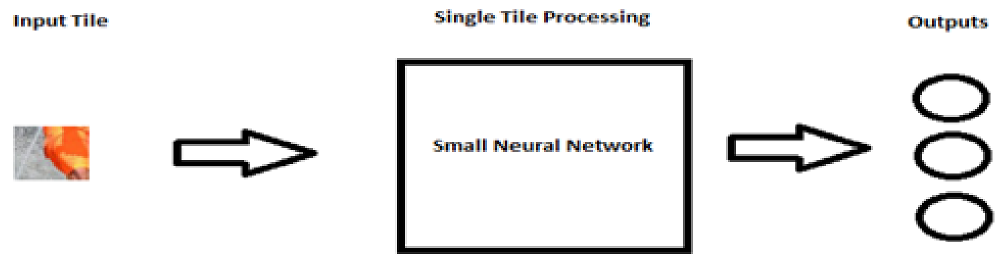


Figure 46. Processing Individual Tiles

Next, we need to save the results from each image tile into a new array [17]. It is important to keep the same arrangement of tiles from the original image when putting it into the new array. This array will be a bit too large for our liking, so we will simply “down sample” it. This is done by an algorithm known as max pooling. “The idea here is that if we found something interesting in any of the four input tiles that makes up each 2x2 grid square, we’ll just keep the most interesting bit [17]”.

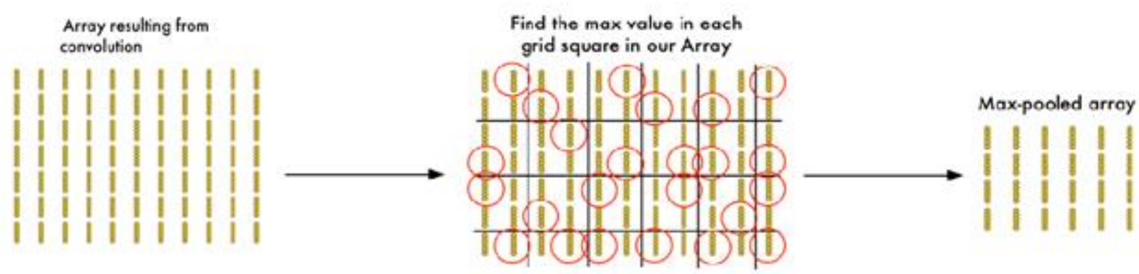


Figure 47. Maxpooling Algorithm [10]

Now the array has shrunk in size, while maintaining the valuable bits. This array is just a bunch of values, so can you guess what’s going to happen now? Just like how we stated in the beginning of this topic, we had an array of numbers and treated each element of the array as an input node. So now we can use this new array as the input into another neural network and have it decide if the image is correct or not, also known as “fully connected [17]” network. The image below wraps up the five-step process of the convolution process.

The beautiful thing about this process is we can combine these steps as many times as you want, having multiple convolution layers. The more layers, the better the system will be at recognizing an object in an image. For example, we can have one convolution for recognizing round edges for the skis, one convolution for a person, and so on.

Though this is not implemented in our design yet, we hope to down the road in the future to improve the Batpack, or have others join in and help implement the best implementation of this, as we know there will be a lot of trial and error. Follow along in the following section for a more comprehensive review of available neural network implementations.

4.3.3.3. Architecture Variations

As you may have noticed by now, utilizing a neural network for our optical algorithm is not as simple as finding code for a neural network online, and copying and pasting that code into our project. Instead, we have learned that neural networks themselves represent an entire class of algorithms, and as mentioned before, these algorithms can be tuned, combined, trained, implemented, and executed in countless ways. Indeed, different neural architectures could be implemented or designed from scratch, different preprocessing steps could be taken on the images themselves, and actually, the only shortage of implementations when it comes to neural networks in this domain would be one caused by a lack of imagination. And this does not even take into consideration the fact that even equivalent implementations can be written using different languages and machine learning tool kits, and most likely, these different implementations would yield different results!

Both fortunately and unfortunately for our purposes, object recognition using neural networks and their derivations happen to be a very active area of research. This means there is a very wide availability of solutions and even code online to implement neural networks. This is extremely fortunate because it keeps us from having to reinvent the wheel when it comes to object detection using neural networks. Instead of wandering around in the dark and making guesses as to what will work the best, we can take stock of solutions which already exist to learn what the best design decisions are. On the other hand, ironically enough the large number of available implementations also make it very hard to truly test and understand all options. Instead of this impossible task, our team will manually select a handful of approaches that we feel would be the most useful to us.

This section will discuss our understanding of these selected implementations, as well as a pragmatic review of the algorithms feasibility in our problem domain. Included in this pragmatic review will be a short and curt cost-benefit analysis of each algorithm. Some of the things our team will consider as a cost in this context are research time, implementation time, and computational complexity.

On the other hand, the benefits of our analysis will focus heavily on the accuracy and precision of each algorithm. Keep in mind as we discuss these algorithms that in this context accuracy and precision are measures on the correct classification of the sidewalk, as well as the correct localization of the sidewalk.

Convolutional Neural Networks

As mentioned before, in traditional neural networks, perceptrons are fully connected. What this means for image processing is that each perceptron ascribes a weight to each input point. For small images, this proves not to be a large problem, however, in high resolution images, this creates an explosion of data which renders the model computationally complex to the point of uselessness [18]. To give a simple example, a 32 by 32-pixel image will have 1024 pixels, and with 3 channels each (for RGB values) this leads to 3072 separate inputs for a single perceptron. With modern computation, this is no big deal, and even allows for larger increases in dimensionality with the introduction of hidden layers. However, the same logic does not hold true for high resolution images [18]. Consider for a moment a 1024 by 720-pixel image. This same calculation (1024*720*3) yields 2,211,840 million inputs for each perceptron! Even for highly performant computers, nothing like the embedded platforms we are using for the Batpack, this amount severely restricts the amount of layers possible in the neural network, but also is highly inefficient.

Convolution Neural Networks are a variant of classical Neural Networks designed as a solution to this problem. Analogous to how neural networks are modeled after the basic structure of the brain, convolution neural networks are said to be modeled after the animal visual cortex. Essentially convolution neural networks are an attempt to use insights we have learned from animal biology to construct more efficient architectures for analyzing images [19].

In fact, arguably one of the first building blocks for this type of neural network was put into place with the publication of Hubel and Wiesel's paper on the animal visual cortex [19]. As summarized on Wikipedia, these researchers were able to show that "cat and monkey visual cortexes contain neurons that individually respond to small regions of the visual field [18]." In addition to this, the paper also showed that similar and neighboring cells/neurons overlap on the visual fields they process. Emulating this characteristic is one of the fundamental pillars of Convolution Neural Networks.

Obviously, this begets the question of how exactly this is accomplished. According to Wikipedia, Convolution Neural Networks achieve their claim to fame with these three distinguishing features:

3D volumes of neurons

Images are thought of as volumes with three dimensions – namely depth, width, and height. In a convolution neural network, this translates to a perceptron layer with the same dimensionality. In direct contrast to traditional neural networks, the perceptrons in this layer are only connected to a small region of surrounding layers. This thought mimics the idea of receptive fields in the animal visual cortex (described by Hubel and Wiesel). Pictured below is an illustration of this concept [18].

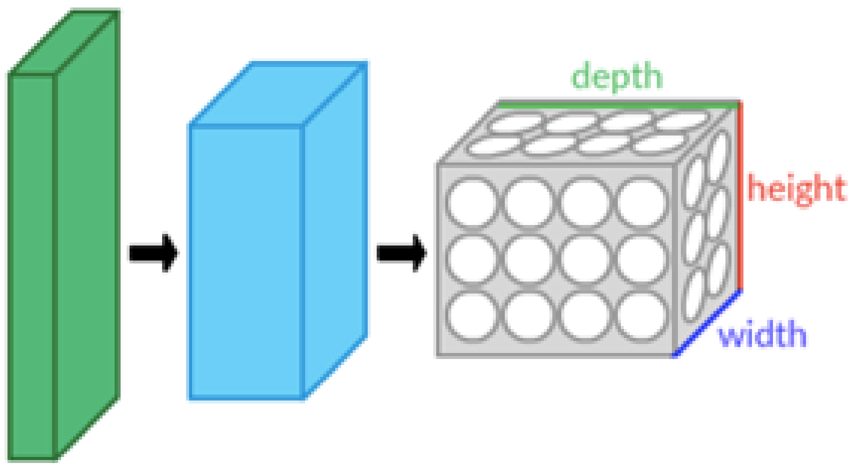


Figure 48. *Convolution Neural Networks Dimensionality*

Local Connectivity

Convolution Neural Networks also take advantage of the strong spatial correlation which occurs in objects in images. Intuitively, we know that pixels in an image which are closer together are more likely to be related because usually they are more likely to be part of one object – think of a single pixel on an image of a balloon. The pixels in the balloon are all (hopefully) related by things such as color and gradient, while the pictures further away will be spatially correlated to different objects. The same is true for computer vision. This concept emulates the previously following the concept of receptive fields which not only makes it much less computationally complex to analyze images, but it also ensures that strong responses are generated for local patterns. Taken from a tutorial on CNN's provided online, the following is an illustration of this concept [18].

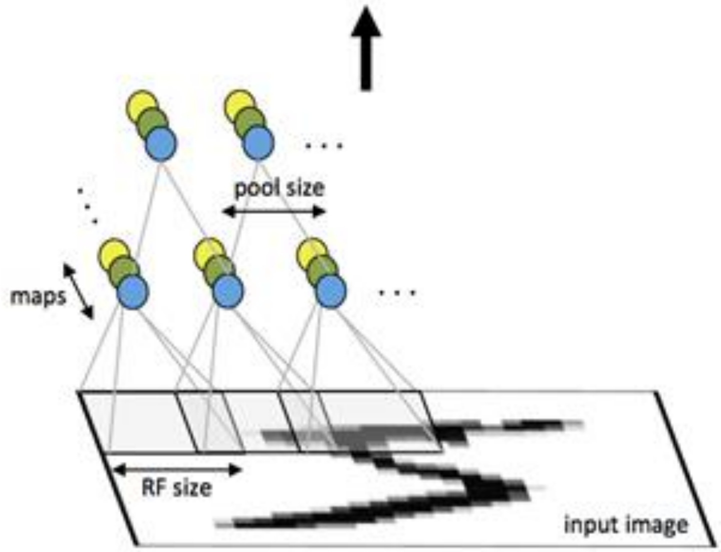


Figure 49. Local Connectivity for Neural Networks

Shared Weights

Finally, there is the concept from shared weights. This step takes the layers generated in the previously discussed steps, and clones them across the entire image – ergo where the shared weights come from. While at first this may seem pointless, this is actually one of the strongest benefits to using convolution neural networks, because it causes the classifier to be spatially independent. Even if objects are in locations never seen before in the training set, they will still be classified correctly by the convolution neural network because the layer parameters will be the same regardless of where the receptive field is located in the image [18].

Localization

As mentioned before, one large downside of using Convolution Neural Networks on image processing is that CNN's, just like traditional Neural Networks, are classifiers. This means that while this deep learning model can classify whether or not an object is in an image, it provides to explicit information as to where that object is in the image. For our purposes, this localization is almost as important as the accurate classification!

Sliding Window for Localization

Koustubh Sinhal, an editor for the Deep Learning company iLenze, describes a method for object localization which is, albeit simple, very popular in the deep learning community. In this method, a sliding window is used to test one area of the image at a time [20]. One very common configuration of this technique is provided below.

1. First, initialize a window of fixed size pointing to the top left of the image.
2. Run the image through the classification algorithm.
3. Iteratively move the window to the right one pixel at a time, repeating step 2, until the window is at the end (the right side) of the image.
4. Initialize the window at the left side of the image, but one pixel down, and repeat this process starting at step two until you have reached the bottom of the image.

This process yields a collection of windows which have each generated a classification metric based on the classifier. This classification metric, which is hopefully non-binary, is a representation of how ‘sure’ the model is that it’s classification is correct. Then, as an optional last step, these windows and their strengths can be combined to create a sort of ‘heat map’ where the ‘hotter’ regions represent higher classification scores. Now localization is as simple as assuming that objects will be located in the ‘hottest’ regions of the image! This process is illustrated below [21].

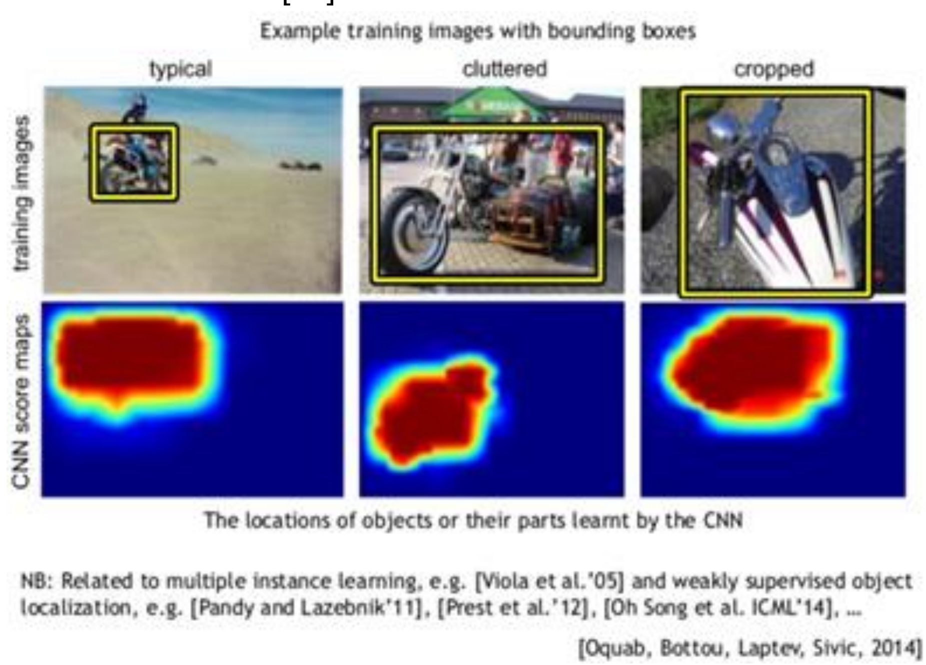


Figure 50. Sliding Window Heatmaps

The largest downside to this approach is the high computational cost associated with this approach. Even with very simple ballpark estimations, because each window must be individually run through the convolution neural network, it is very apparent that it would be extremely difficult to run this algorithm within the constraints of an embedded platform such as our own. However, in the same publication in which Koustubh Sinhal describes the sliding window approach, he also provides us with a much faster method – Regional Convolution Networks.

Regional Convolution Networks

Regional Convolution Networks are not generally seen as one specific localization technique. Instead, the basic idea is to use less fancy, much quicker methods, for locating potential objects, and then only running the Convolution Neural Networks on those regions. Some pipelines that come to mind include the edge detection techniques mentioned earlier in this paper. For a review of those, feel free to check to associated Optical Algorithms Section. In his online publication, however, Sinhal names the Selective Search method, which uses similar features (RGB, HSV, gradient, etc.) to locate objects [20].

In the previous method, the classifier was run the same amount of times as the number of pixels in the image. With the finely tuned selective search algorithm, we now have instead only several thousand regions to test. Because most of the computational complexity of this algorithm comes from the Convolution Neural Network Classifier, this exponentially increases the speed of the algorithm! This is process is illustrated below [20].

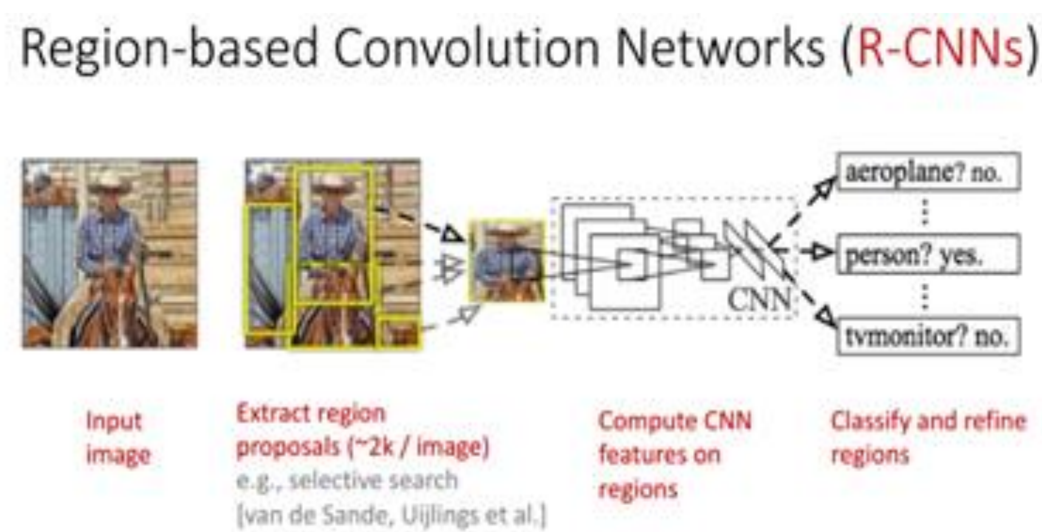


Figure 51. Regional Convolution Networks

Spatial Pyramid Pooling

Unfortunately, for our purposes, even at a conservative estimate this still may not be quick enough. All the research from this area still seems to rely heavily on GPU's and large power-hungry CPU's. However, for applications where speed is very important Spatial Pyramid Pooling seem to be the go-to method. In fact, in his editorial Sinhal says about Spatial Pyramid Pooling that "is one of the turning points for making a highly accurate RCNN pipeline feasible in run-time [20]." Ultimately – Spatial Pyramid Pooling seems to be beating out other methods when it comes to speed due to one key feature. In Spatial Pyramid Pooling, the entire image is run through the convolution layers all at once!

To put the process in layman's terms, this impressive feat is accomplished by running the entire image through the convolution layers, and then calculating independent features on the regions found by the Selective Search algorithm. These features in turn, are used to generate the bounding objects for localized objects. This process is extremely clever in my opinion, because it sidesteps the entire issue of processing many sub-images, and instead 'pools' recourses to decrease the computational complexity! The entire pipeline, including steps with the convolution classifier were provided graphically (shown below), in Sibhal's publication, and this publication does a fantastic job of illustrating this concept [20].

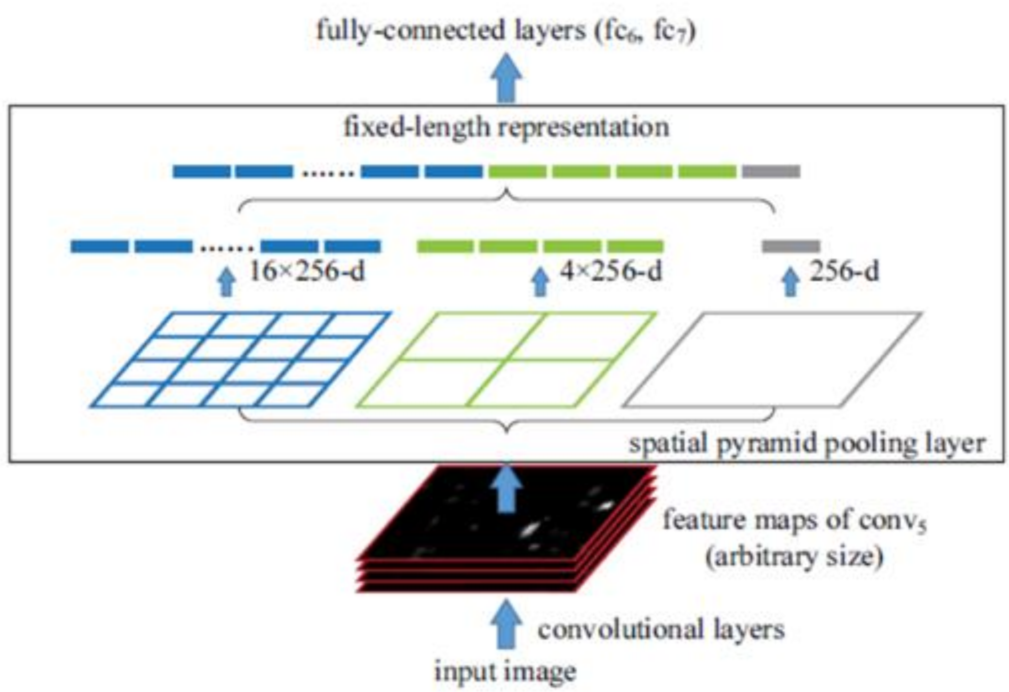


Figure 52. Spatial Pyramid Pooling for CNN

4.3.3.4. Review of Available Tools

One large benefit of using a Neural Network for the optical algorithm is that there is a large abundance of open source and free-to-use Deep Learning Api's. Some of the most popular modern toolkits, especially when it comes to neural networks, include TensorFlow, Theano, Torch, and Keras. Unfortunately, it would take much time and recourses to evaluate and test each deep learning library. Instead, without going into too much detail, this section conducts a high-level review of each of the aforementioned technologies.

TensorFlow and Theano

TensorFlow and Theano are both Deep Learning Frameworks. This means that both libraries provide it's users with a syntax for describing, in the words of one noteable deep learning expert, "general-purpose computation graphs" [22]. Each library essentially provides a language to facilitate describing mathematical models - including Neural Networks. TensorFlow is a Deep Learning Framework developed by google. It enjoys wide usage, with a large and growing community. Theano is the predecessor of TensorFlow, and is where TensorFlow received many of it's original ideas.

Torch

In stark contrast to TensorFlow and Theano, Torch is a high level library designed to make it easy to generate machine learning models. The goal of this library is to provide users with a quick and easy way of mixing and matching different machine learning models with high level of abstraction and immediate results [22]. Torch is written in a language called Lua. According to the homepage for Lua, Lua is a language developed in ANSI C, which facilitates writing machine learning algorithms for embedded applications.

Keras

Finally, there is Keras. Keras, another tool developed my google, is built to run on top of either TensorFlow or Theano. Keras is similar in philosophy to Torch. It seeks to provide a simple interface for users wishing to jump in and get their hands dirty. Keras also has the added benefit of being written in python. This means that keras plays nicely with a very large suite of statistical and machine learning libraries. Additionally, Keras enjoys a booming community, and a large horde of tutorials and documentation [22].

4.3.4. Filtering Sensor Data

The Batpack relies heavily on data coming from ultrasonic sensors and laser range finders. While this data is usually fairly accurate, there are many things that can cause 'noise' in the system. In the case of the ultrasonic range finders, this noise can come from anywhere ranging from the material a surface is made out of, the angle which the ultrasonic ping hits, velocity of the sensor, and other various acoustic signals that may also interfere with the ultrasonic sensor's receivers. In the case of the laser range finders, there are different considerations such as reflective surfaces which nonetheless cause similar noise in the range data.

This poses a significant challenge to our engineering team. The user expects a smooth response from the Batpack, and may become confused if the signal seems erratic and/or unresponsive. To solve this obstacle, before the range data is used to make calculations on vibration strength, it will first be filtered. However, in this situation, it is necessary that we pair the correct filter and implementation with the correct use case. The following describes some Software Filters that are common in the industry.

4.3.4.1. Moving Average Filter

The moving average filter is not only known as one of the easiest digital filters to understand, but it is one of the most commonly used as well [23]. This filter is also very intuitive in that it simply averages the last 'N' data points, and presents that as the current output.

Pros:

- Easy to understand, and operates on the same basis that the mathematical average operates.
- Smooths data very well.
- Good for digital signal processing in the time domain.

Cons:

- May smooth the data to well.
- The mean is not resistant to outliers, which may cause a shift in the average for that window.

- Window size may make this filter less responsive to change

Additional Considerations:

- The manufactures of the Maxbotix are seen on electronic forums as saying this is the worst possible filter to use with their ultrasonic sensors [24]. Maxbotix has been contacted to confirm these claims, but we are still awaiting comment from them.

4.3.4.2. Moving Median Filter

This filter is almost identical to the Moving Average Filter, with the only difference being it uses the Median of the data instead of the average. For our applications, this may be the most suitable filter, not only because it is so simple to implement, but also because it is resistant to outliers [23].

Pros:

- Easy to understand, and operates on the same basis that the mathematical median operates.
- Smooths data very well.
- Good for digital signal processing in the time domain.
- Resistant to outliers.

Cons:

- Window size may make this filter less responsive to change.

See the image below for a visual explanation of the Pro's and Con's listed here for the median and mean filters [23].

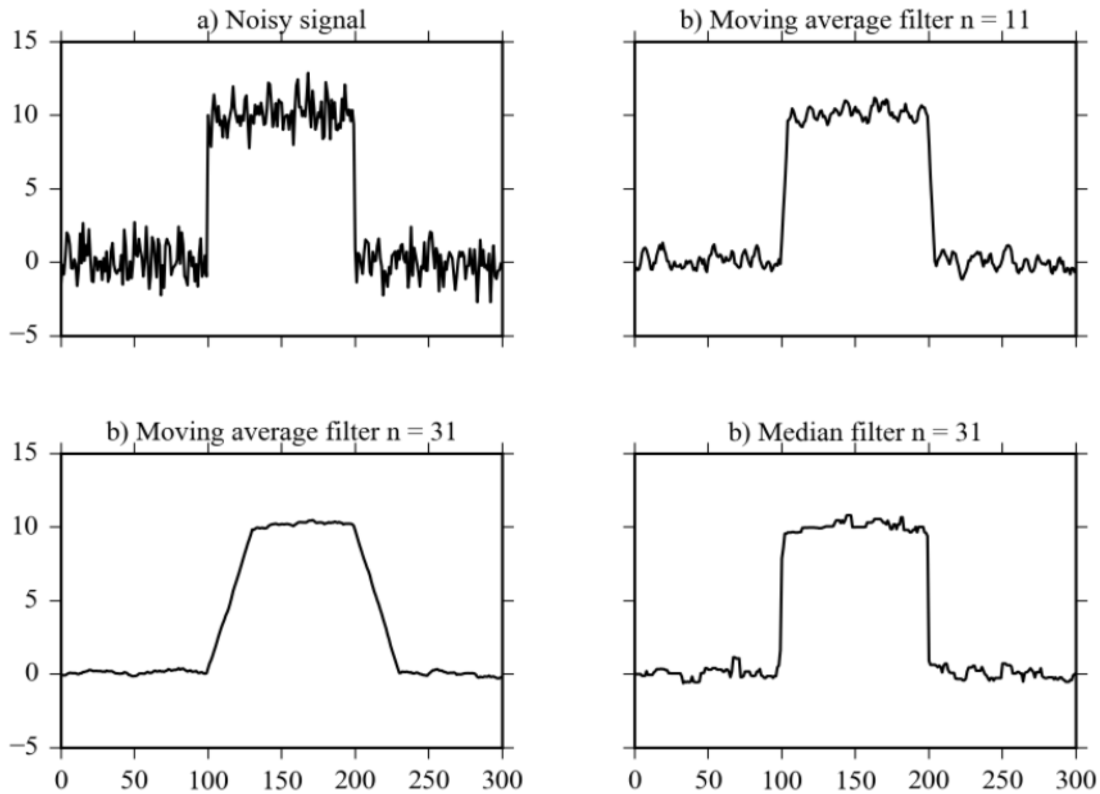


Figure 53. Filters Comparisons

4.3.4.3. Kalman Filter

This filter is the most complicated of the three filters presented, but is also seen as being much robust. The filter has been used for decades in the industry, especially in aerospace and navigational applications. In essence, this filter takes as inputs models of the physical world along with models which measure the error found in the inputs, and uses a statistical basis to make inferences of the most likely actual values. This filter is also constantly updating its internal estimators with the latest data to increase its model’s accuracy. This feedback loop is central to the Kalman Filter’s ability to accurately estimate actual data from sensors that are filled with error. For an example Kalman filter, see Greg Czerniak’s example in the graphic below [25]:

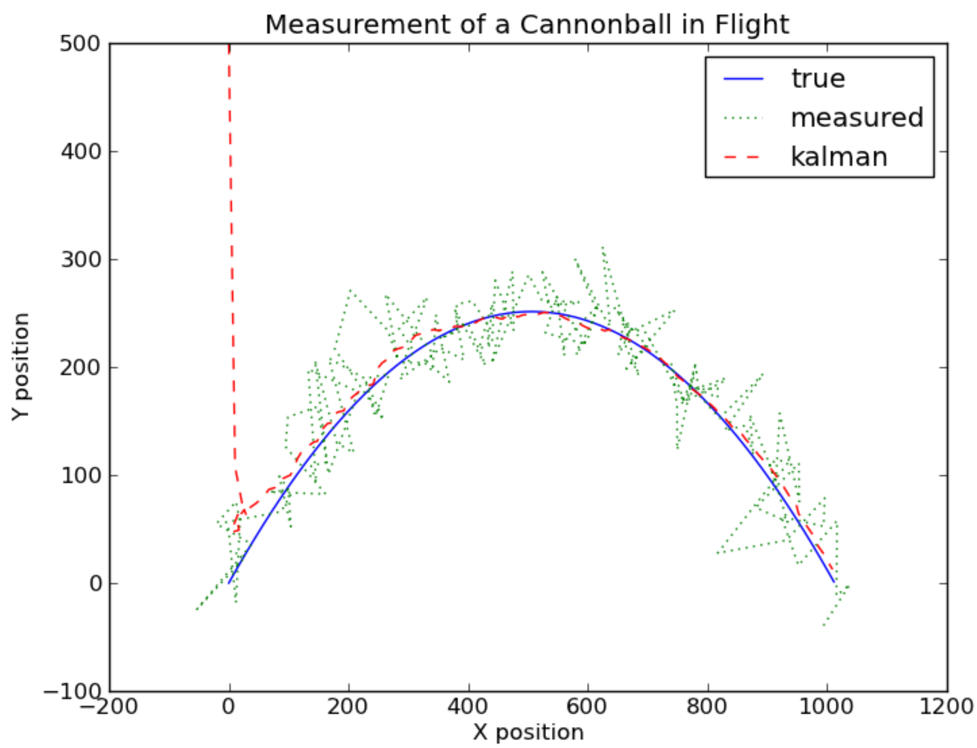


Figure 54. Example of Kalman Filter

Pros:

- Extremely robust.
- Smooths data exceptionally well.
- Very responsive to change.

Cons:

- Hard to implement. Especially in embedded settings where we are limited by the technologies and software libraries we can employ; this filter represents a serious investment in development and implementation time.
- Computationally expensive. This filter requires many more calculations than the previous two filters, and we must keep in mind the hardware and timing constraints which come from the microprocessors.

4.3.4.4. Filter Conclusions

To minimize speculation and increase the efficiency of our sensors, our team will not make decisions on filters until we have thoroughly tested each filter with our physical components in the Build phase. During this phase, we will test each filter with each component, and analyze the data to make a decision on the best fit. Additional considerations during this phase will include measuring the latency of the filter, and the computational expense on the microprocessor.

4.3.5. Multiple Ultrasonic Sensor Array

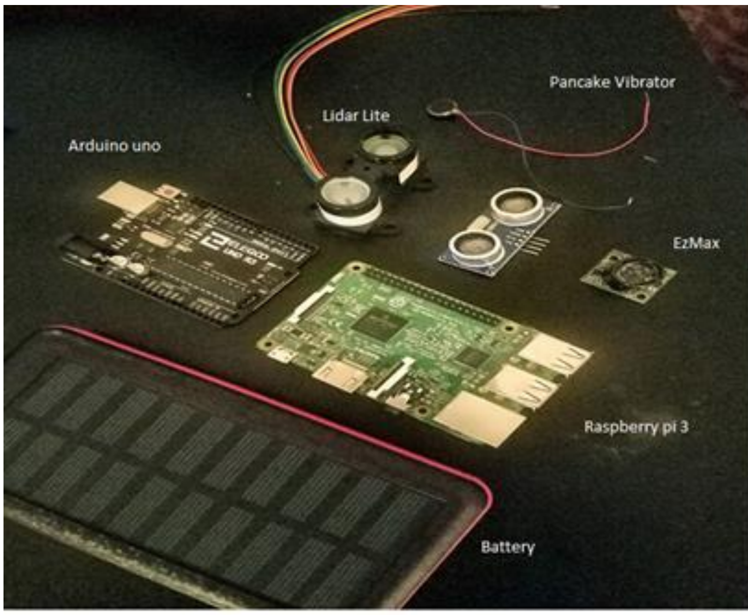
An ultrasonic sensor has two pins that control the input and output of the device, the trigger pin and the echo pin. Where the trigger pin starts, or stops the transmission of the ultrasonic beam and the echo pin reads if any data was captured, being either HI or LO. More information on this is described above, in section 5.1.2. Ultrasonic Range Finding Sensors. The Batpack is designed with multiple ultrasonic sensors. The echoes of one ultrasonic sensor should not interfere with that of any others. Our team discussed the difference in frequencies that could be used to ensure that no cross communication between sensors occurs, however this approach is cumbersome and not needed.

The approach our team has taken in order to prevent any cross-signal interference is to have a delay on the Trigger pins after each cycle on the corresponding sensors. Since ultrasonic sensors deal with the milliseconds, delaying the trigger for each pin will not be noticeable to the end user. The ultrasonic sensors will be triggered in a loop after each iteration a new ultrasonic sensor will be triggered.

To start the loop off the trigger pin will be set to HI and the echo pin will listen to see if it picks up an object. If there is nothing to detect, the loop will automatically timeout after the max time for the distance allowed; this would work as after the time expires for the allowed distance the loop would move to the next sensor gathering the same information. If an object was detected the Echo pin would be set to the HI signal and the time from the Trigger to the echo would be captured. This then would be converted into a distance based off the current air temperature and sent to the step function to determine the correct duty cycle to use for the corresponding vibration motor. This motor will continue to vibrate until the loop gets to its turn next, in which case the vibration motor will be switched off if no current object is found in it’s path.

5. Project Design

Under this section the design aspect of this project will be discussed. This will be further split into two sections: hardware design and software design. The hardware design section will focus on the physical designs of the project, and the software section will focus on the code being utilized to run the sensors, the vibrators, the Raspberry Pi, and the microcontroller. The following images shows all the different devices that will be incorporated into the design.



5.1. Hardware Design

This part of the report includes any physical designs. It includes the circuit design for the vibration motors and the laser rangefinder. It will also discuss the power system and microcontroller. There is no need for circuitry related to the ultrasonic sensors as they plug directly into the microcontroller.

5.1.1. Vibration Motor Circuit Design

The schematic for a single vibrating motor is provided below, along with an explanation for each component.

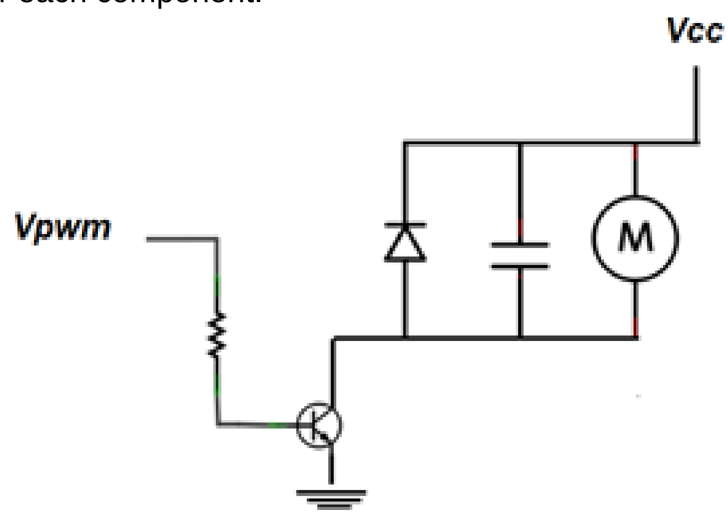


Figure 55. Vibrator Motor Circuit

- Vcc: Vcc is the constant voltage supplied to the system. On the prototype board this wire is connected to the 3 volt pin. This is the DC voltage that powers the system.

- Vpwm: This is the voltage which provides the Pulse-Width-Modulation. The main function of this voltage is to provide current to the base of the transistor, allowing it to modulate the voltage coming from Vcc.

- Transistor: The transistor pictured is a simple Bipolar Junction Transistor. This transistor takes input from Vpwm on it's base. When the voltage is high on it's band, it allows current to flow from Vcc to ground, effectively turning on the system and driving the motor.

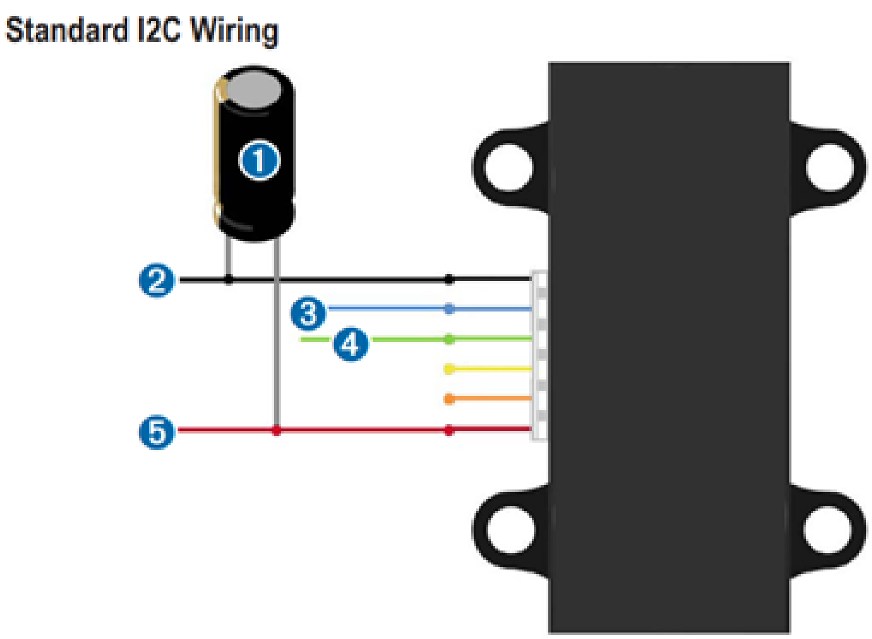
- Resistor: The resistor in the image serves to limit the current entering the transistors. Resistance values will be around 1kΩ

- <u>Diode:</u> The diode pictured is a Schottky diode. This diode is a safety precaution to prevent any inductive kickback from the motor for quiescent currents.

- <u>Capacitor:</u> The capacitor serves to reduce electromagnetic interference coming from the DC motor. This component sits as closely as possible to the leads on the vibrator.

5.1.2. Laser Rangefinder Circuit Design

The circuitry for the LIDAR Lite v3 is relatively simple. It only requires a single capacitor with 680 μF of an electrolytic capacitor. It is important to observe the correct polarity of the capacitor when soldering it on, otherwise it will break, requiring replacement which is difficult with soldering.



Item	Description	Notes
1	680μF electrolytic capacitor	You must observe the correct polarity when installing the capacitor.
2	Power ground (-) connection	Black wire
3	I2C SDA connection	Blue wire
4	I2C SCA connection	Green wire
5	5 Vdc power (+) connection	Red wire The sensor operates at 4.75 through 5.5 Vdc, with a max. of 6 Vdc.

Figure 56. Wiring for LIDAR Lite v3 (Courtesy of Sparkfun)

The black wire is ground, the red wire is 5 Volt DC power, the blue and green wires are digital connections relating to telling the laser when to pulse and to report distances.

As shown, all that is required is the capacitor between the ground and power lines. This capacitor is used to prevent inrush current, which is defined as the maximum instantaneous input current which gets drawn by the device when it is initially turned on. preventing this will increase the longevity of the LIDAR Lite v3.

5.1.3. Power Systems Design

Of great consideration for the design of the power systems in this project, is the required voltage being fed into the microcontrollers in our system. There are two of these that will need to be powered. The first being the ATmega328, which will be a part of the printed circuit board that will be designed specifically for this project, and will control the ultrasonic sensors, the laser rangefinder, and the vibrators corresponding to those sensors. The other will be the Raspberry Pi 3 which will be handling the computer vision aspect of this project.

The ATmega328 microcontroller requires 5 volts to be delivered to its VCC pin, AVCC pin, and AREF pins. The Raspberry Pi 3 will also require 5 volts delivered to it ports, the current draw will depend on the components that are connected to the Pi. This is very convenient for us, as the battery selected to power this project delivers power at 5 volts in both of its ports. The Raspberry Pi 3 will handle power delivery for the computer vision components, and its port will be directly compatible with the USB port from the battery. Thus, we will just plug the Raspberry Pi in directly to the battery using the 2-amp port.

While the computer vision aspect of the project will be simple, the microcontroller and its printed circuit board will be more involved. The USB port of the battery will need to be converted to the DC Vcc and ground pins in the microcontroller. Also, there will be vibrators, ultrasonic sensors, and the laser rangefinder connected to the printed circuit board. Therefore, there will be some voltage regulation required by the printed circuit board, but not too much as several of these components require 5 volts which is what the battery delivers.

The LIDAR Lite v3, MaxSonar Ez3, and HC-SR04 sensors all utilize 5-volt power supplies. This will simplify the PCB design, as there won’t have to be any voltage conversion between the battery supply and the sensors.

Unfortunately, the vibrators voltage range is lower than 5 volts, and will require a voltage regulator. The Maxim 604 voltage regulator will be sufficient to get it down to 3.3 volts, which is an acceptable voltage to be delivered to the vibrators.

The total power draw that the printed circuit board will take will come out to around 0.5 amps, so the smaller 1 amp port from the battery will be more than sufficient.

5.1.3.1. Voltage Regulation

As stated earlier, the voltage delivered by the battery is too high for the vibrators being used. As a result, it will have to be reduced using circuitry to levels acceptable for the vibrators.

The three most common forms of voltage regulator include the Standard NPN Darlington Regulator, Low Dropout Regulators, and Quasi Low Dropout Regulators. While there are plenty of other kinds of regulators, they will be unnecessary for the purposes of this project. What is needed is a voltage regulator that will convert 5 volts to 3 volts, a relatively small change, and be preferably smaller in size.

Considering the needs of this project, a Low Dropout Regulator will be the ideal choice. These require DC input, which is going to be provided by the battery directly. Low Dropout Regulators also are good for converting voltages when the input voltage is close to the intended voltage output. Considering that it will be fed 5 volts and need to output around 3 volts, this is important. These regulators don't have any switching noise as no switching occurs. Due to the lack of large inductors and transformers they are also fairly small in size, which is ideal for out mobile system. These regulators are also very simple relative to other regulators. The main drawback to these systems is that they do dissipate power to regulate the voltage, however this is acceptable, as we have plenty of amperage to spare from the battery port being used for the PCB.

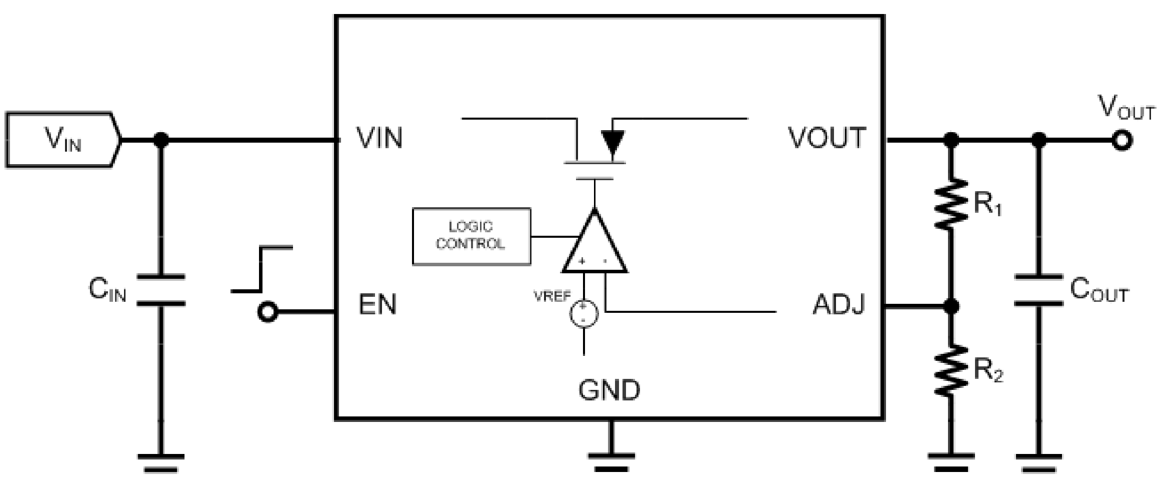


Figure 57. Low Dropout Regulator

Low Dropout Voltage Regulators are also considered to be good to use in conjunction with batteries. They can use the input voltage further into the discharge cycle of the battery for a longer period of time than other voltage regulators. This is because as a battery operates, the terminal voltage will slowly decrease over time as the electro motive force and the internal resistance.

MAX604 Low-Dropout Voltage Regulator	
Input Voltage	3 - 11.5 Volts
Output Voltage	3.30 Volts typical
Maximum Output Current	600mA
Operational Temperature	0 - 158° F
Output Short-Circuit Duration	1 minute
Minimum Load Current	20 µA

Figure 58. MAX604 Specifications

The MAX604 Low Dropout Voltage Regulator is sufficient for the needs of this project. It has a maximum voltage intake of 12 volts, a maximum output current of 600 mA, and a maximum operating temperature of 158 degrees Fahrenheit. It only needs to modulate 5 volts, and output about 250 mA. The company that produces this product also states that it only requires very simple circuitry to be built around it. This circuit is also very safe, and can run on a short circuit for a minute, longer than the short circuit is likely to last. Furthermore, the minimum load current of 20 µA is acceptable as the vibrators that will be on the receiving end of the MAX604 will not be able to run on that low level of current.

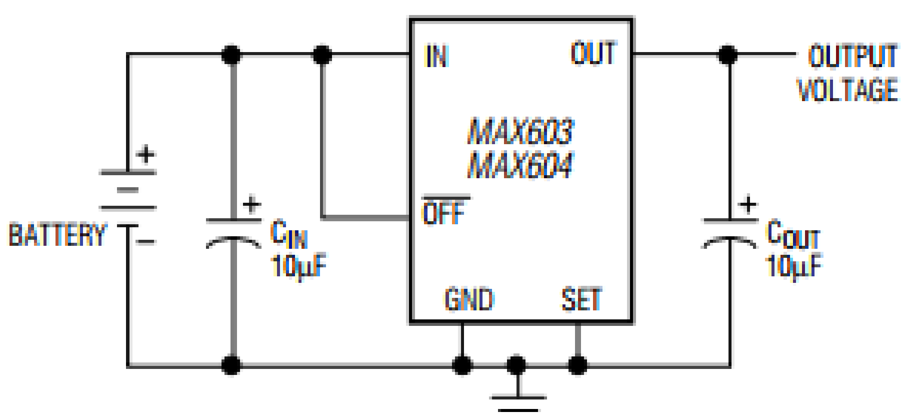


Figure 59. MAX604 Typical Operating Circuit

5.1.3.2. Battery Design and Housing

When selecting the battery that we were going to use it was extremely important to consider the intended operator of the system. Blind people are going to have some special considerations that need to be made with regard for the usability of the product. If disposable batteries were used in the system, that would require the operator to purchase them repeatedly, which is completely unnecessary, especially when rechargeable batteries have come so far.

After deciding upon a rechargeable battery, the difficulty of getting the battery recharged should be considered. A battery with two terminals is going to be difficult for someone without use of their vision to put in its terminal correctly every time it needs to be charged and then put back into the device. As blind people frequently use the sense of touch to handle things, they also have to be safe from shock. Therefore, a battery that takes user friendly means of plugging in and charging was important.

This all why is why for this project a regular, commercially available phone battery is going to be used. These have several advantages over most other batteries. They utilize USB and Micro USB ports to charge and plug into the devices they power. This makes it very easy for a blind person to handle because they can be safely handled by hand without fear of shock. Furthermore, it is simple to feel the proper orientation for a USB or Micro USB port and cable to be inserted. Additionally, USB and Micro USB cables are very easy to replace or get repaired, important considerations for any user. On top of that, most phone battery packs like the one this project will use have very similar outputs. Most of them have at least two ports, one of which is 2 amps and the other being 1 amp. Almost all of them put out 5 volts in USB ports. So even in the unlikely case of the first battery being broken, it will be easily replaceable.

Figure 60. Antun Solar Charger

Ultimately, the Portable Antun Solar Charger was selected as the battery to power this system. It has a 22,400 mAh capacitance. This battery is robust, capable of taking damage without breaking or leaking dangerous chemicals, which can be a serious issue with many batteries. Considering the batteries high capacitance, it is relatively light, a great boon for a mobile system. Like most phone power banks, if is charged via Micro USB and has two USB ports, one maxing out at 2 Amps and the other maxing out at 1 Amp. Both ports output 5 volts, which is very convenient, as the Raspberry Pi that controls the computer vision aspect of this project requires 5 volts input. The other subsystem on the printed circuit board has several components that require 5-volt input as well, namely the ATmega328 microcontroller, the HC-SR04 ultrasonic sensors, the MaxSonar Ez3 ultrasonic sensor, and the LIDAR Lite v3 laser rangefinder. This makes designing the PCB much simpler as it reduces the amount of voltage regulation that needs to occur. Indeed only the vibrators will need a voltage reduction.

Notably, the Antun Solar Charger battery pack also has a solar charging capacity. This is beneficial as this system is meant to be used while walking places. That typically means the user will spend a good amount of time using the system while outdoors. This will at the very least marginally increase the battery's longevity. Furthermore, it gives the operator options with regards to charging the system. Should the user hypothetically be out and about somewhere, and the system runs out of charge, the user could remain where they are and allow the battery to recharge enough to get them home. Now, the ability of the solar cells to produce charge is limited and the manufacturer recommends charging using a wall outlet for regular use as it is much faster and more reliable than using exclusively the solar cells. Additionally, inclement weather can reduce the battery's ability to recharge, but the user will be less likely to be walking outdoors during inclement weather anyway.

This battery will save the design team a lot of effort for multiple reasons. Most batteries that are available don't produce 5 volts. This would require additional circuitry to regulate the voltage, which would have to be large in order to handle all of the current going through to the system. This battery also simplifies the question of what apparatus is needed to recharge the battery. It also means that it will not have to be incorporated the printed circuit board. This will save on production costs as time won't be needed to be spend on soldering or spent on having the soldering professionally done.

With regards to battery housing, the battery will be held on top of the backpack. This will allow it to charge using the solar cells on top of the battery, extending battery life. The way the battery is designed makes it so that it is resistant to rain, not that it's immune to it, but the user will be able to find cover and not have to worry about the battery short circuiting. The USB cables attaching the battery ports to the Raspberry Pi and printed circuit board will simply be plugged in and run into the backpack.

$$\frac{Ampere - Hour\ Rating\ of\ Battery}{System\ Current\ Draw} = Total\ Runtime\ of\ System$$

This equation shows how to calculate how long the system will be able to run off of one full battery charge. The 22,400 mAh rating of the battery gives us one half of what we need to calculate the longevity of the system off of one fully charged battery. For the other half we need to calculate the total current draw of the system. The LIDAR Lite v3, MaxSonar Ez3 ultrasonic sensor, HC-SR04 Ultrasonic sensor, and corresponding vibrators will require an estimated 0.4 Amps. The computer vision portion of this project will use an estimated 0.3 Amps. In total the system will run on 0.7 Amps. Plugging in these values will yield a total runtime of 32 hours. This is a fantastic battery life, as a user will likely not be using the system at all times of the day, indeed it will most likely be turned off the majority of the day. This will allow the system to be useable for several days without needing to be recharged. Furthermore, this system does not even take into account any power the solar cells will generate while the system is sitting in the sun, giving any outdoorsy users additional use out of the system without having to recharge it.

5.1.4. Microcontroller

The microcontroller in the printed circuit board will be the processor that controls the non-computer vision aspect of this project. That is, it will control the LIDAR lite v3 rangefinder along with the vibrator that will report the distance data collected from it to the user, the three HC-SR04 ultrasonic sensors along with their corresponding vibrators, and the singular, forward facing MaxSonar Ez3 ultrasonic sensor as well as its corresponding vibrator. As the microcontroller, we are using is the ATmega328, an Arduino (the Uno R3) utilizing this very same microcontroller is being used for testing.

Specifically, the tasks performed by the ATmega328 are as follows:

- Send the cyclical signals to the ultrasonic sensors telling them when to send out their pulses.

- Receive the signal sent back by the ultrasonic sensors, and convert that to usable distance data.

- From the distances read back from the ultrasonic sensors, modulate the vibrator circuit to the appropriate vibration level.

- Similarly, the microcontroller will signal the laser rangefinder to read distances, receive that information, and relay it to the vibrators in a meaningful way.

The ATmega328 microcontroller is a simple, low-cost, and low powered single chip device. It is commonly used and is popular in Arduino models, specifically the Arduino Uno and Arduino Nano. With an 8-bit RISC architecture, running at 20 MIPS, this microcontroller has more than enough processing power to accomplish the objectives necessary in this project. With 32K bytes of In-System Self-Programmable Flash Program Memory, it will also be capable of storing the small amounts of code necessary to drive the sensors and vibrators. Its operational voltage of 5 volts makes it immediately compatible with the battery we have selected, and compatible with both kinds of ultrasonic sensor we are using as well as the LIDAR Lite v3 rangefinder.

This microcontroller is also compatible with a wide series of communication protocols. This list includes Serial Peripheral Interface (SPI), Inter-Integrated Circuit/Two Wire Interface (I2C/TWI), Universal Serial Interface (USI), Universal Asynchronous Receiver/Transmitter (UART), and Universal Serial Bus (USB). This is extremely useful for this project, as it is highly modular, and taking off/adding on different functionalities is something that might be done further down the road. For the current purposes of this project, only I2C is being used on the microcontroller, as that is the protocol used by the ultrasonic sensors and laser rangefinder.

ATmega328 Specifications	
Architecture	8-Bit, AVR RISC
Frequency	20 MHz
SRAM	2K Bytes
EEPROM	1K Bytes
Flash Memory	32K Bytes
Data Retention	100 Years at 77°F
Programmable I/O Lines	23
Operating Voltage	1.8 - 5.5 Volts
Temperature Range	-104°F to 221°F
Digital Pins/Analog Pins	23 Max/8 Max

Figure 61. Microcontroller Specifications

As shown in the table, the microcontroller can operate at voltages much lower than the 5 volts that will be provided. This is good because batteries gradually provide lower voltages as they drain. As lower voltage is provided, the microcontroller operates at a lower maximum frequency. This is acceptable as we do not require much processing power.

Also of note in the table is the data retention of the ATmega328. At regular room temperature it will retain data for 100 years. This means that the microcontroller will not likely be a limiting factor in the longevity of the system, something that we would like to be as high as possible.

In order to program the ATmega328, we will utilize an already owned Arduino Uno R3, which uses the ATmega328. The Arduino Uno R3 is a robust development board, and most of the team on this project is already familiar with. This familiarity is why the ATmega328 was selected, in conjunction with its remarkably low cost.

All of this will make prototyping and testing hardware designs and software easy, especially in conjunction with a basic reusable breadboard. Furthermore, the Arduino Uno R3 comes with a wire intended to plug into computers so that you can program it. This wire also happens to be useable with the battery selected for use in the design for this project, further increasing our ability to build a prototype easily for testing and data gathering. The low cost will also allow for many replacements to be purchased, preventing possible disaster.

Conveniently, the ATmega328 is a dual in-line package, which is capable of surface mount. This is especially important for testing prototypes and changing code, as none of the development team is particularly skilled with soldering, and using soldering paste would add on additional, unwelcome difficulty.

5.1.5. Failure Provisions and Environmental Safety

Considering the very low levels of power being delivered to the system, it can be considered very low-risk. Any short-circuits will be very localized due to the modular nature of the design. On the printed circuit board, the only common connection between the vibrators, the ATmega328 microcontroller, the LIDAR lite v3 laser rangefinder, the HC-SR04 ultrasonic sensor, and the MaxSonic Ez3 ultrasonic sensor is at the battery. All of these parts draws small currents and should any of them fail the others will not break as a result. Also, the Raspberry Pi will not draw a large amount of current and is highly unlikely to break.

Due to the low levels of current being drawn by our electronics, the risk of an electrical fire is extremely low. Additionally, direct handling of the electronic equipment is very unlikely to result in electrical shock. This is helped by the actual circuitry being stored inside of the backpack, and everything requiring handling by the user is external. Therefore, the operator has no reason to go inside the backpack and handle components. Unfortunately, the currents within the system are not low enough to be classified as a Limited Current Circuit, which means that by current draw rating alone this system cannot be considered safe for direct access to the electrical components by regular operators.

The voltage of the system also will not exceed 5 volts in any areas. This classifies every part of this system as Safety Extra-Low Voltage (SELV) Circuit. Any circuit lower than 60 VDC can be classified by this as long as it is separated from a non SELV circuit by at least two levels of protection. SELV Circuits are considered to be safe for operator access.

All external components will not have any uncovered electrical components. The buttons to turn off subsystems, the vibrators, and sensors will have sufficient covering over any wires and electrical connections to protect from water damage or accidental handling by the user. Importantly, the area where most current will be flowing through is at the battery. As these connections will be handled by the operator, it is significant to the overall system's safety. As the load on these parts is still well below 1 Amp, and is a USB connection, the operator of the system is still safe from shock. Also, the battery itself is designed to be water-resistant.

While this means it can survive being in the rain for a period of time, it is still advisable to get out of rain as quickly as possible, for both the safety of the electronics, and the comfort of the operator.

While it is not expected that the system will require any heat dissipation, if this is found not to be the case during prototype testing, we can add heatsinks to any components that get hot.

5.1.6. Project Schematic and PCB Design

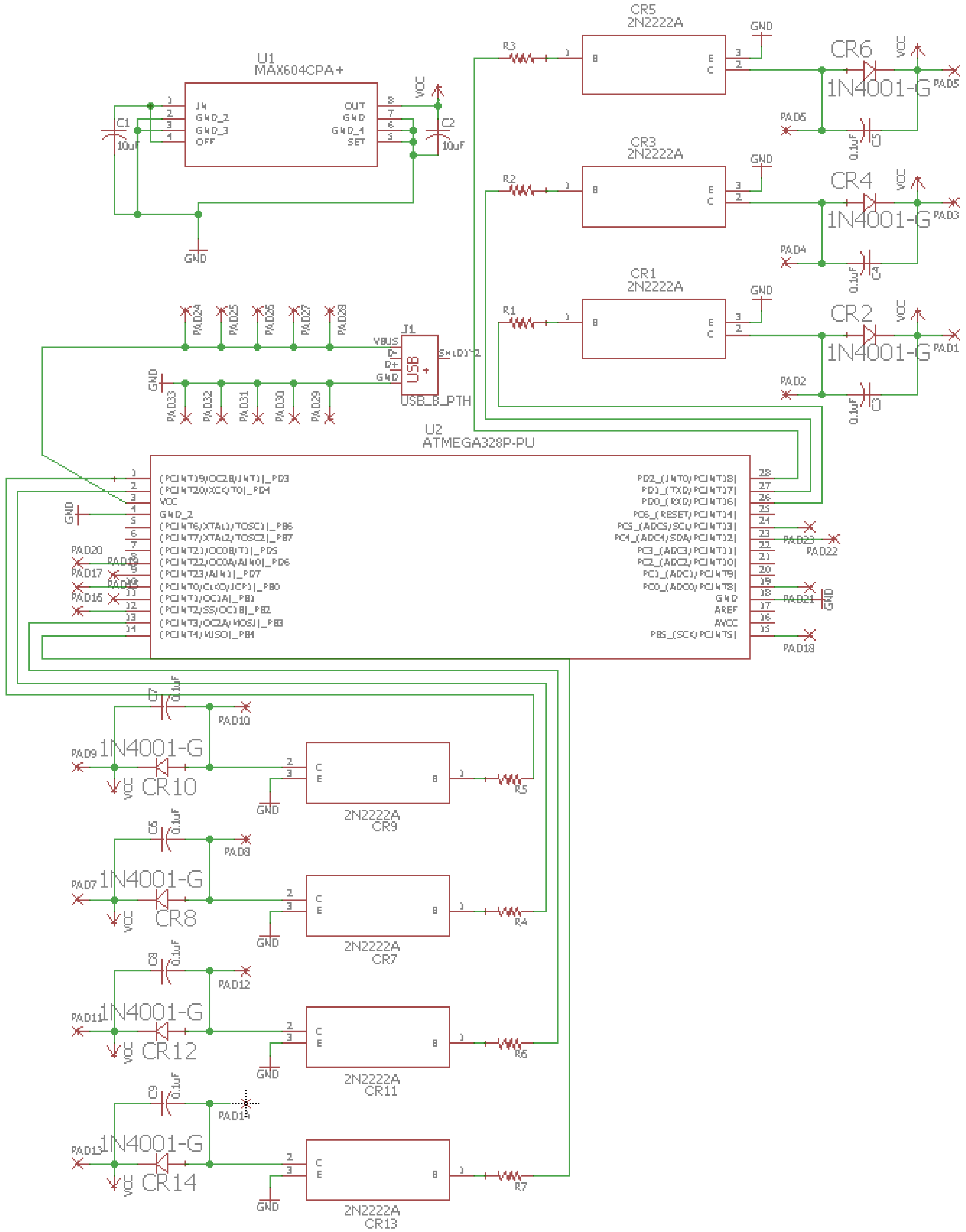


Figure 62. Electrical Schematic

Above is the electrical schematic for the entire project. The printed circuit board will more or less contain all of the circuitry required to drive the ultrasonic sensors, the laser rangefinder, and the vibrators. This was done using EagleCAD, and will provide the base file for the PCB file using the on-board converter.

In the top left of the design is the voltage regulator. This will take in power from the USB 5 volt VBUS, and convert it into 3.3 volts which is usable by the vibrators. Unlike the rest of the components used in this project, the vibrators are not meant to run on 5-volt power, hence the regulator. The MAX604 will be able to handle the current required by the vibrators, especially considering the vibrators will be modulated by the ATmega328.

Below the MAX604 is the power line for the printed circuit board. It gets power from a female USB connection that uses a regular USB cable to connect to the battery. The first thing to connect to power are wire pads that will be used to solder connections to the ultrasonic sensors and laser rangefinder. Then it connects to the voltage regulator used for the vibrators. The last thing it connects to is the ATmega328 itself, at the VCC pin.

The seven 2N2222’s and the related connected circuitry are all used to drive the vibrators. Each set will be connected to the voltage regulator. They each also utilize a 1N4001 transistor to control current flow. The vibrators will be easy to solder on as there only needs to be a positive and negative terminal connection. The pulse-width modulation will be handled on board the printed circuit board. The capacitor is used to prevent current overflow. The reason for there being driver circuitry for every individual vibrator is because otherwise there would be no way to modulate them simultaneously. The vibrator circuits are very good with regards to microcontroller pin economy, as each only needs a single digital pin to modulate them. As such they were attached to PD_0 through PD_4, as well as PB_3, and PB_4. All of these are digital-only pins, saving space for any applications that might need an analog pin, of which the ATmega328 has far less of.

The only other things to note in this schematic are the wire solder pads connected closely to the ATmega328. These are all connections meant for the ultrasonic sensors and laser rangefinder. Of these, only the MaxSonar Ez3 ultrasonic sensor requires an analog pin, at PC_0. This is the only pin used by this sensor. Each of the three HC-SR04 requires two digital pins, one to send a trigger signal, and the other to receive the echo signal. These three sensors occupy pins PB_0 through PB_2, PD_6 and PD_7, PC_5 and PC_4. As these were the final pin connections needed for the ATmega328, and almost every pin in this microcontroller can be used as digital, the pin choice was based upon what was convenient for real world placement.
The printed circuit board that is based off of this design will be a three-layer board. While it was hoped to only be a single layer, this turned out to be nothing

more than a pipe dream. The amount of circuitry required to run the many sensors and vibrators that are necessary for the operator to get as much data about their surroundings as possible meant that at least two layers are absolutely necessary. A single layer would cause overlapping power lines. The second layer allows us the connect everything properly without crossing power lines.

The third layer of the printed circuit board is not strictly speaking necessary, but one of convenience. It is the ground layer of the PCB. This ground layer accomplishes several things at once. It makes the actual job of designing the circuit board much simpler by allowing components that need a ground connection able to connect directly, rather than having to draw an array of power lines. It also makes soldering components on easier. Finally, the large ground plate provides extra heat dissipation. It spreads heat all over the board, rather than concentrating heat where power is dissipated across components. While this is not expected to be a necessary design feature, it never hurts to have more heat dissipation in a circuit, especially considering the PCB will be stored within the backpack itself.

5.2. Software Design

The following subsections cover how each subsystem of the Batpack will perform based on the software designed and the best suited needs to ensure the reliability of the final product.

5.2.1. Design Methodology

While designing the Batpack's software and code to make it function as described throughout this paper our team will be using an agile methodology for our software development. This methodology will allow us to use adaptive planning and development to satisfy the needs and changing requirements of the system. Since the technology and communication protocols being used in the Batpack are new to most of the team, we may discover later that some parts of the system may not be as robust/reliable with the given hardware and software design may need to be adjusted to fit the devices requirements. The agile approach to this will allow the team to respond and fine-tune changes necessary to meet the requirements. Our team will need to give ample time to adjust any of these changes.

Another approach that was considered to design the Batpack on was the Waterfall method. This type of methodology would only allow for a static approach and all the design must be set in stone. Any changes in the design of the Batpack would lead to a total recall of all the previous work. Our team decided since much of this technology is new it would be a poor choice to use this method. An agile approach will allow the team to set requirements, plan, design/develop and test until the Batpack satisfies the requirements in a reliable way.

5.2.2. Version Control

A version control system is almost necessary in any software development. It will create a record of all the code that was tested during our agile development phases. The most notable one is Git and is what are team will be using to develop are software with. We will be hosting our code on GitHub's service in a private repository. Hosting on this service will give the team a peace of mind as it is stored on the cloud and not on any one local device. The project's code will also have redundancy in case any code needs to be rolled back. We will be able to pull or commit our code from any device we need; using only the most recent version of the code. This will allow our team to easily collaborate with each other.

Using a version control system, like Git, will give the team a track record of who, what and when code was submitted to the repository. This feature will allow the code to be easily trackable in case any breaking changes were implemented.

5.2.3. Integrated Development Environment

An Integrated Development Environment, or IDE, is a software application that allows the developers to code, build, and debug their application on one spot. Excluding only the computer vision aspect of the Batpack, the rest of the systems will rely on the ATmega328 microcontroller. This is the standard microcontroller used in the Arduino Uno, in which our team will be prototyping their design on. For programming these aspects of the Batpack the Arduino Software 1.8.3 (IDE) will be used.

This IDE will allow the team to code and build the software directly to the Arduino since all the drivers and libraries needed are easily integrated with the Arduino Software. The Arduino language is based and very similar to the C programming language; allowing our C programming experienced team the convenience of a familiar language. This software will also allow the team to use the bootloader on to microcontrollers not on the Arduino, making our final software design easier to deploy.

The computer vision aspect of the Batpack will rely on the OpenCV library. This library is written in C/C++, to focus on performance of multi-core processing. However, a python bindings were created to utilize the OpenCV library. Our team has more experience in the Python language and is new to computer vision. Using Python for our computer vision programs will allow our team to focus more on the reliability of the application rather than learning a new language. The IDE for Python that will be used is the IDLE. This will allow the team to test the extensive computer vision algorithms on desktop and laptop computers before porting it over to the Raspberry Pi. Once it is running on the Raspberry Pi extensive testing and debugging will be done using the same environment, IDLE, to ensure the programs are running smoothly and correctly.

5.3.4. Ultrasonic Sensor Array Algorithm

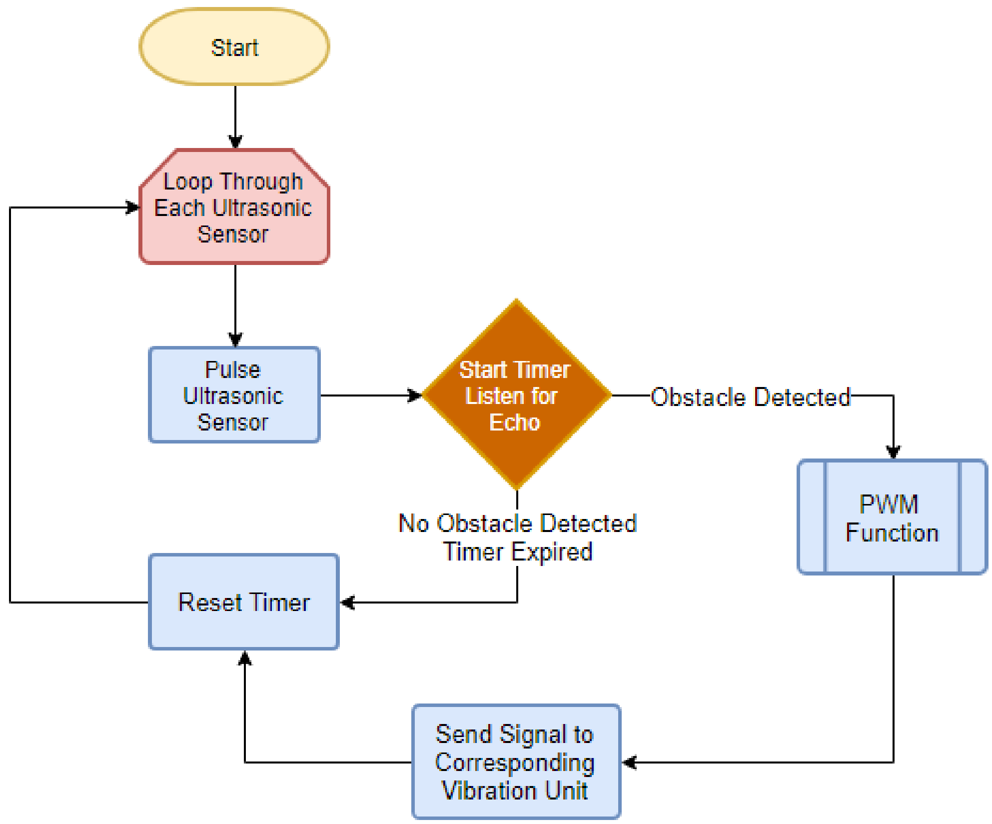


Figure 63. Ultrasonic Sensor Flowchart

Our Ultrasonic sensor array algorithm will function as seen in the Figure above. The program will first send a pulse with the first ultrasonic sensor in the array. Once the pulse is sent the ultrasonic sensor will look for a response, or an echo, from the pulse it just did. If an obstacle was detected within the amount of time, which calculates the distance, the distance will be sent off to our PWM function, step function, in order to calculate the correct duty-cycle the corresponding vibration motor should receive. Once calculated the signal will be sent to the vibration unit using the intensity of the pulse-width-modulation duty cycle. The loop will then restart, resetting the timer, and move on to the next ultrasonic sensor. If no obstacle was detected, meaning the timer ran out and the distance was farther than the device allows it detection rate at, then the loop will restart in order to move on to the next sensor and reset the timer.

The approach we are implementing helps reduce the number of errors that occur due to cross-talk between the sensors. This is due to the timer that waits a given amount of time so the ultrasonic signals diminish and cannot be detected. Using this approach, the number of samples that the Batpack can receive over a given amount of time is limited. Our team decided that this would be the best way to solve this problem as the detection rate will still be in the count of milliseconds for each sensor.

5.2.5. Optical Sensor Algorithm

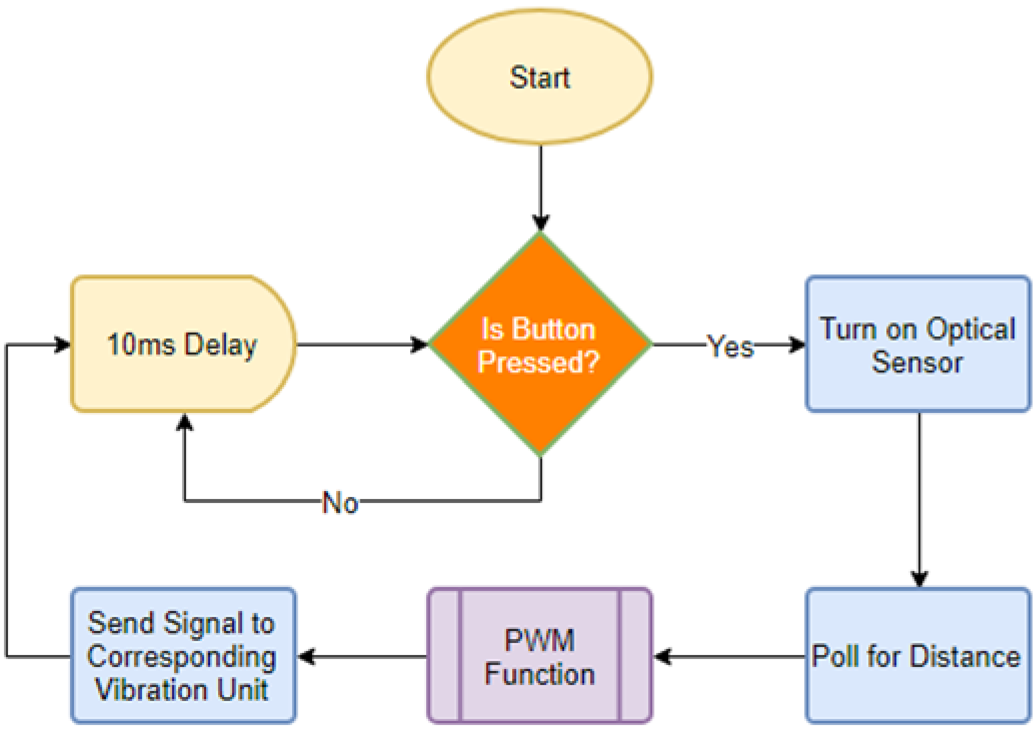


Figure 64. Optical Sensor Flowchart

Unlike the ultrasonic sensors, the optical sensor will only be active for as long as the user presses the button down. This is done for two reasons; first, to save battery life across the device since the device will need to get the individual through a normal day of use. Secondly, the optical sensor provides a much more accurate and precise reading two the obstacles in which it is point at, the user should be able to decide in which situation he or she may need this functionality.

The flowchart in the figure above shows how this will function. Using an I2C communication protocol the device will become active and poll for the distance.

This distance will then be sent to the pulse-width modulation function to determine what intensity the vibration unit should send to the user. This will then delay for a very short amount of time to prevent any misleading results and continue its procedure until the user stops pressing the button.

5.3. Enclosure Design

The design will be implemented in two ways. First, we recognize that we will need a housing unit for the battery and microcontrollers. Our first thought was to go with a hand-held device that could be used to point at objects to read back a distance. Thinking of battery size, and where the MCU's would need to be placed, and how the user is not hands free, we moved on to the next idea. The first way we intend to implement the Batpack is with a fanny pack. It will be perfect size to house the boards and battery, as well as have an adjusting waist size for each user.

The glasses will be connected via wires, which will run down the body to the pack. The belt will be connected with four ultrasonic sensors. At this time, the design will be implemented with three HC - SR04 and one LV-MaxSonar EZ2. The middle facing will be the EZ2, with a SR04 on each side of it, near front hip ends. The final sensor will be placed directly in the middle of the pack of the pack. This pattern will allow for 360 degrees sensing as well as deal with the issue of interference from each other.

After further thoughts, and interviewing with a visually impaired friend, we came up with the idea of using a backpack. This design will include sensors on the front straps, but will need to take in consideration of how to place the sensors to avoid interference. This will be investigated in the building and testing phase. We also plan to investigate if adding a strap across the waist with an ultrasonic sensor connected to it would be better than attaching the sensors to the backpack straps. The housing for the electronics will be the backpack, in which we can weatherproof with certain designs, as discussed below in 5.3.1. Attached to the top of the backpack will be the glasses that will be lined to stretch out and zip back in when not in use. Think of how your blinds work when you open them and close them.

As we use two different products (belt and glasses), we need a way to keep the connected as well as house all the components. As of now, testing the two designs independently will give us a better idea for which design to choose.

5.3.1. Weatherproofing

The design uses multiple electronics and some of which are not friendly to some conditions. We plan to house certain electronics (boards and battery) into a fanny pack or backpack(waterproof) which will be attached with ultrasonic sensors, known as the Batpack. This will help protect the system power from heat and rain. Considerations for weatherproofing the camera and laser distance sensor were made, but rejected due to high increased costs for waterproofed products. Both methods, fanny pack and backpack, will be tested in the design for best comfort and functionality.

5.4. User Interface

The user interface is how the interaction between the product and the user takes place. This falls under two categories, the input from user and the output given from the product back to the user.

5.4.1. User Input

The user input of the Batpack must be very simple to use. Every input must be associated with either touch or speech, and cannot have any visual components in order for the user to operate it. The user input will have an on/off switch for all of its components. This includes an on/off switch for the following: the ultrasonic sensors and midriff vibration motor combo, the optical sensor and chest vibration motor combo, and the computer vision.

5.4.2. Product Output

The first prototype of this idea was to have a hand-held device that would vibrate as objects got closer. Taking this idea, we improved it by turning it into a belt, making it hands free for the user. With more sensors, we could add more vibrators to give a precise idea when objects were either coming from the left, right or right in front of you. Because of the weakness of ultrasonic sensors accuracy, we implemented the glasses which would be attached with a camera and laser distance sensor. From this idea, we move to what we now call the Batpack.

As of now we have three different models we will implement in our testing to see which is most user friendly first hand.

The first model will be a continuous vibration feel. The vibration will be constant, and it approaches an object, the vibration will increase, decrease as we move away from the object. This could be beneficial as you always know what's going on, but could get annoying.

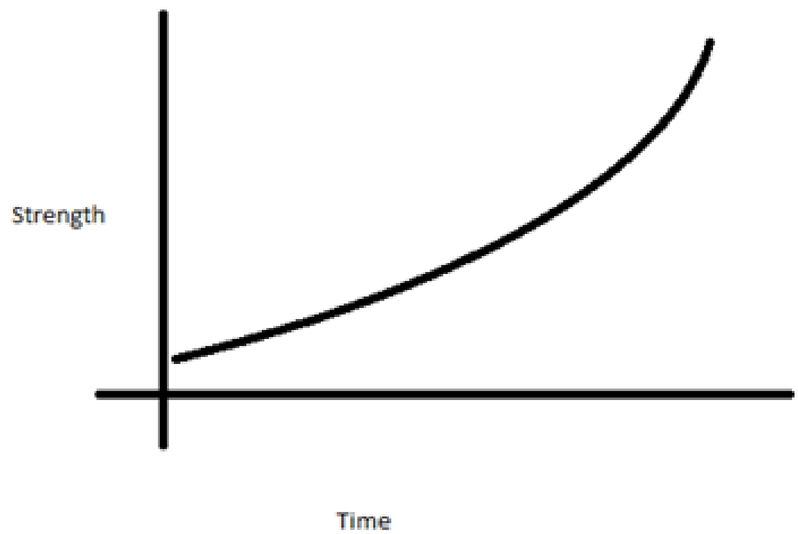


Figure 65. Strength of Vibration Vs. Time for an Approaching Object

The second model will be based around how many times the vibrator will pulse. At a longer range, the vibrator will pulse slowly, and increase the rate of the pulse as we get closer to an object. Below describes the equation that can be used to help determine the correct balance.

Time between pulses=d*k

K is a constant, which can be scaled up to a value of 255. This will be found experimentally based off how many pulses we want to implement based on certain distance. The results of this will be found in the testing section. D is the distance in which the object is from the sensor. We will refer to this idea pulse width modulation.

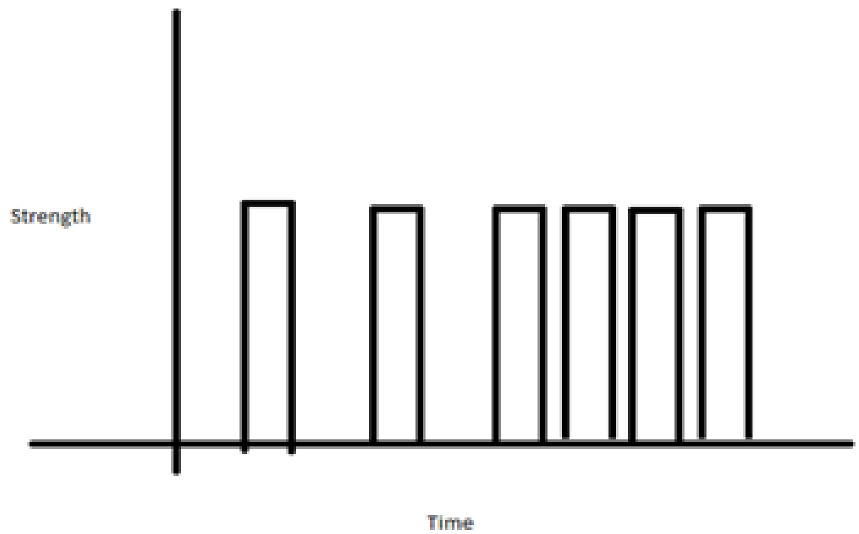


Figure 66. Pulse method for an Approaching Object

The last model will control the vibrations based off certain distances. For example, from ranges 0-2 feet, there would be a strong vibration and moving from 2-4 feet, we lower the vibration, and so on until the end of the range our sensors work in. These values of separate blocks will need to experiment with for comfort level, just as the other two methods will need testing.

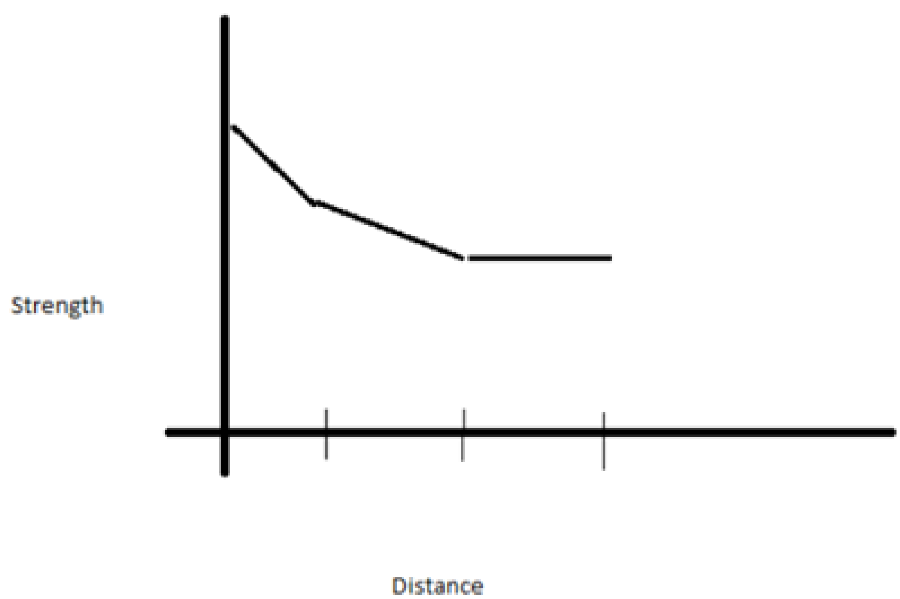


Figure 67. Different modes of operation

Our primary method of attack will be done with pulse width modulation. Using vibration motors with PWM, the user of the Batpack will be alerted of objects within proximity to them. These motors will vibrate at various strengths around the user's midriff, the weaker the vibration the farther away the obstacle is and the stronger the vibration the closer the object is. The vibration units will be placed underneath each ultrasonic sensor, on the side of the person's body, so the user will know which direction the object is.

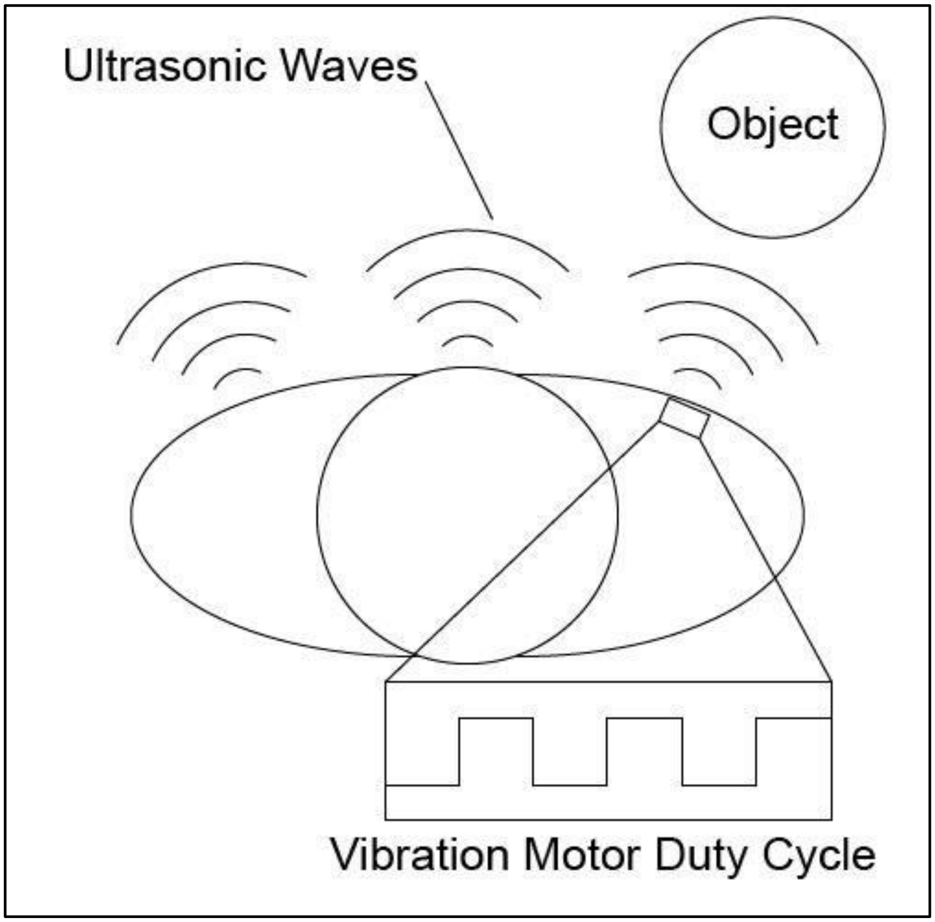


Figure 68. Object Detection via Vibration Motors

For the optical sensor, the user will be notified by vibration units located in the straps of the Batpack. Our team decided that it would not be a good user experience if the vibration units were located by the optical sensor, which is located on the head. As this would interfere with everyday life and leave the customer unsatisfied with the product. Our approach to this solution is to move the vibration motors to the straps of the Batpack to ensure the user still gets the haptic feedback needed to feel the world around them. (See Figure below)

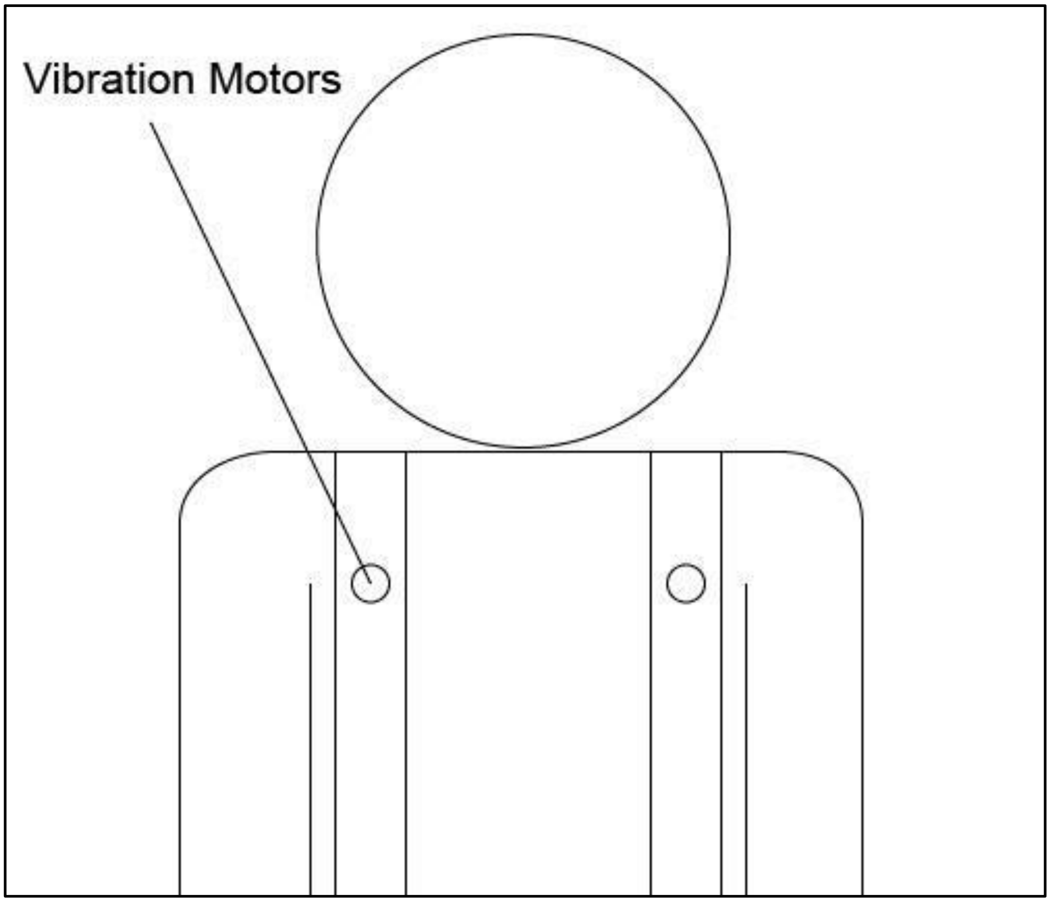


Figure 69. Vibration Straps for Optical Sensor

Another sensor that must be taken into consideration is the camera, which provides the computer vision to the system. This subsystem will allow the user to detect the sidewalk they are walking on and provide feedback in order for the user to stay on the path. This subsystem needs to allow a manual switch in which the user will get feedback on if they are looking at the sidewalk or a grass surface. In order to accomplish this task two more vibration motor will be used on the shoulders of the Batpack straps, as seen in the Figure below.

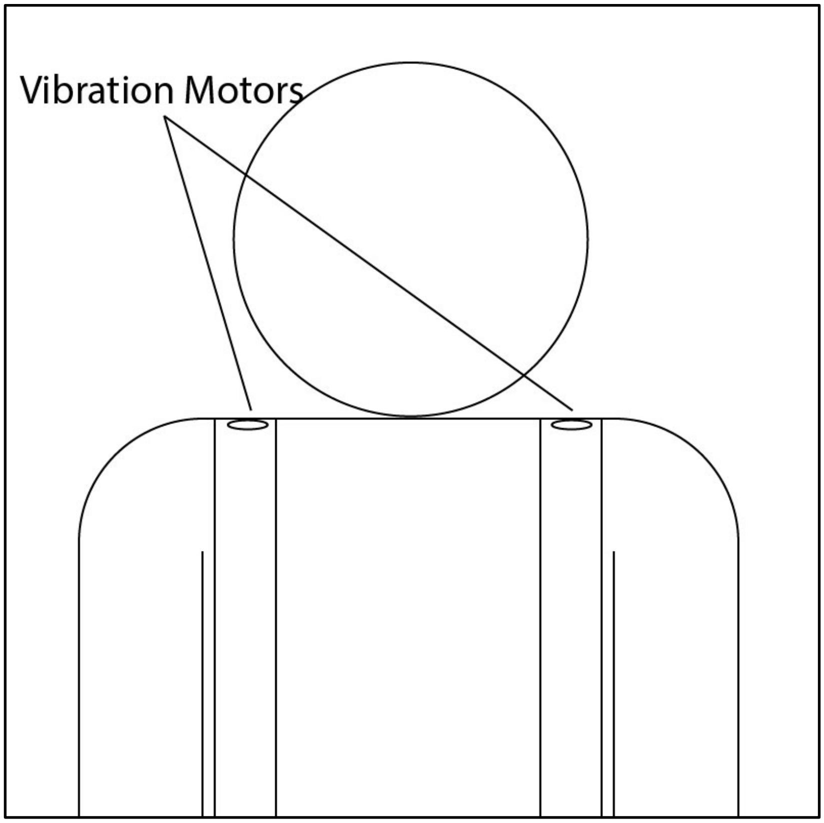


Figure 70. Vibration Motor Placement for Computer Vision

These two vibration motors will work in harmony with one another. Having one sensor on both sides of the user's head will allow the Batpack to notify the user if they are approaching the side of the walkway. These vibration motors will also function with Pulse-width Modulation, so the closer they user is to the edge of the walking path the more intense the vibration pattern will be.

These vibration motors will also convey information such that the user will be able to tell what surface is in front of them. For example, if they are approaching a grassy surface in front of them, both vibration motors will switch on. This is an important feature within the Batpack as the user will be visually impaired and without the feel of a traditional walking stick they will need another way to feel what kind of surface they will be walking upon.

6. Testing

Each unit of the Batpack should be thoroughly tested in order to produce accurate information back to the user. This section breaks off into two separate categories, the hardware focused parts and the software focused parts. Each test should be performed separately from one another and as a group, to ensure that all components of the Batpack work as intended.

6.1. Hardware Testing

For testing purposes, the Batpack can be split into four parts. The microcontroller, the sensor circuits, the power delivery system, and the computer vision subsystem. Most of the testing will be performed in the UCF Senior Design Lab, located in Engineering 1, room 456. This room contains oscilloscopes, multimeters, function generators, and DC power supplies. This will be sufficient for the purposes of checking the circuitry of this project.

Basic components such as resistors, capacitors, and transistors used in the PCB that need to be soldered on will be checked via multimeter before and post-soldering to make sure it works before soldering it on and to ensure the soldering was successful.

6.1.1. Microcontroller

The microcontroller that will be used for this project's PCB will be the ATmega328, the same one used in the Arduino Uno R3. As this is a popular commercially available product, it will be up to industry standards and very unlikely to come broken. In order to verify this, once unpackaged all of the pin outs will be tested with signals appropriate for the pin requirements on a breadboard.

This pin testing can be accomplished on the software side of the team simply. They will ensure communications protocol pins and serial ports are operational. Otherwise it will be sufficient to keep an eye out for odd or unexpected behavior from the microprocessor. Like with basic components, the microcontroller will be tested after soldering onto the PCB to ensure the soldering was a success. A multimeter will accomplish this task.

6.1.2. Sensors

To ensure that all parts of the overall system are working properly, a thorough testing of sensor components is necessary. Each individual sensor will be electrically tested to make sure that it is working and reading data correctly. The SR04 Ultrasonic sensors each draw

For the HC-SR04, and MaxSonar Ez3 Ultrasonic sensors, they will be hooked up to an Arduino, plugged into a laptop, and the run software that displays whatever distance the sensor is reading. Then, a group member will stand at known distances from the sensor, and the display will be read. These distances will be under 2m, as that is the reported distance the ultrasonic sensors can fully accurately detect a human. So long as it is capable that detecting that an object is there at 2+ meters, that will also be acceptable. So long as the values read are correct within a certain range, these sensors will be fine. This test will be performed at different angles from the sensor to establish the actual operable width the ultrasonic sensors can operate in.

Testing the LIDAR-Lite v3 laser will be done similarly to the ultrasonic sensors. It will be hooked up to an Arduino, plugged into a laptop, and distance readings will be read. Unlike the ultrasonic sensors, however, the laser will be expected to read distances more accurately from a greater distance. Also, it will be a pinpoint measurement, rather than having a wide radius cone to detect objects in.

The results:
We run the Lidar Lite using standard I2C protocol and pulse-width modulation, which includes the 5v pin, ground, clock signal and data signal connection. Additionally, we equip a 680uF capacitor to maintain level voltage. Thanks to Garmin, they maintain an Arduino library for the Lidar Lite, making implementation much more smoothly. We begin by setting up the configuration to default and I2C protocol to 400 kHz. Stepping into our loop, we take a measurement after every 100 readings with bias correction. This process will help improve the estimated distance measurement, improving results. The code includes two header files to include, Wire (for I2C library) and LIDAR Lite (for the device).

The device was tested on multiple objects. It was extremely precise on picking up on objects from afar, such as chairs along a pathway, a piece of paper, and a reflector from a bike. This is what was expected from the device, so we moved on to testing more objects to check how accurate the device is. Testing an aluminum container from across the room had expectantly good results. The sensor could pick up the object within a couple of inches of where the laser appeared to be shooting at. This is great insight, as the product will be extremely precise in demonstration and will be used for what we bought it for. Pinpointing how close an object is important to the user, so they can maneuver or even pick up items they are looking for. When an object gets close to the sensor, it appears to read to a minimum of five centimeters, and will fluctuate up to twenty centimeters. Though this isn't a huge issue, as we will use pulse width modulation for the vibrators, we can set the beginning step from five to thirty centimeters as the max vibration for objects getting closer than thirty centimeters.

One interesting object that had trouble being read by the sensor was the test glasses. When swiped in front of the sensor quickly, no object was being detected. We then sat the glasses in front of the sensor and some data was finally being read. Unfortunately, it was not constant and would recognize, then derecognize the glasses over and over. This fluctuation rate was about every second, which could pose to a problem later down the road for certain demonstrations.

Normal Object Reading (paper)	Glasses
22 cm	22 cm
22 cm	23 cm
23 cm	24 cm
23 cm	524 cm
24 cm	460 cm
25 cm	25 cm
25 cm	120 cm

The device was then testing in the dark. Due to the laser like properties, it didn't have to many issues with finding an object. This can be an important feature as testing with other products might show it won't be able to handle this process. Meaning at night time, this might or might the only thing keeping this product viable. Testing the ultrasonic and camera in this setting will also be implemented. The following picture shows the setup for the Lidar Lite.

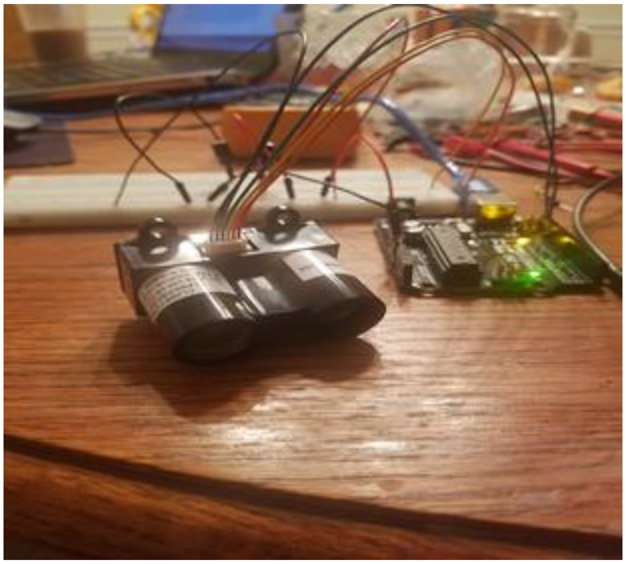


Figure 71. Lidar Lite v3

Ultrasonic Sensors:

The SR04 will be the first sensor to test. There is plenty of code for the SR04 produced by Arduino, as most kits will come equipped with one. With the process described on calculating the distance from before, we begin measuring different types of objects. The first object at hand is a plane book. Moving the object up and down in front of the sensor, it picks up the object when it is within a few inches of the sensor at close distances. As we move further away, the object is detected at a bigger range, thanks to the properties of sound waves increasing as the distance increases.

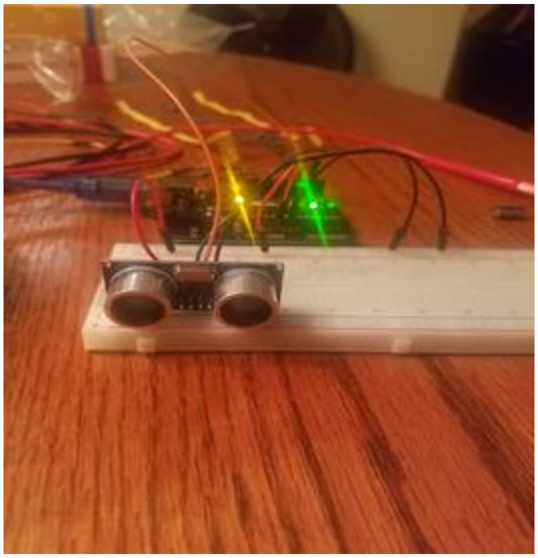


Figure 72. HC-S04 Ultrasonic Sensor

The next object we test is a see-through glass bottle. Because of its round shape and reflective property, the sensor can pick up the object, but after a distance of about a foot, the measurements begin to fluctuate reading as if wasn't there, then reappeared like magic. This can be an issue because the vibrators will be sending false signals to the user that an object is either there or not.

We tested the glasses to check how it compared against the laser sensor. The data was extremely unstable, and a snippet of the data read is shown below:

Normal Object Reading (paper)	Glasses
22 cm	22 cm
22 cm	122 cm
23 cm	3050 cm
24 cm	25 cm
23 cm	119 cm
23 cm	3057 cm
25 cm	29 cm

This is the same issue the laser sensor had. This means some objects will fall in this category of unrecognizable, and could cause the user harm. The next objective is to test the EZ max ultrasonic sensor, and see if it runs into the same issue.

The final test we ran, was to see if the ultrasonic worked in the dark as well. Indeed, it did and the results were like those of when the light was on. This is good news, as we now know two sensors will be working on the dark.

We ran tests on multiple objects, just as the laser sensor, such as boxes, papers, chairs, bottle, see through items, round objects, and more of different texture and styles. These sensors work well with most objects and more examples will be demonstrated in video, after all components are working together, to test the designs full capability or recognizing objects near the user.

The MaxSonar ultrasonic sensor worked just as good as the SR04, with a small change in accuracy, but not much to notice unless observed closely. So the question becomes, should we eliminate the more expensive ultrasonic sensor and use all SR04's for the design. Previously talked about was the interference patterns that can emerge if using the same frequency for the sound waves. With the MaxSonar placed in the middle of the SR04's, we ensure a better chance of no inference from each other and only worry filtering the results. The only real benefit the MaxSonar has is a cleaner signal overall than the SR04. The following image shows the setup of the EZ sonar.

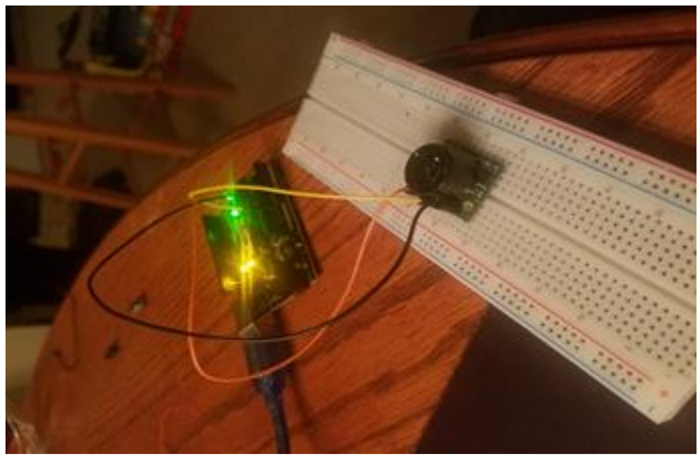


Figure 73. Maxsonar EZ3

Each sensor will be paired with a vibrator, which will communicate readings to the operator. They will be very simple to test, they will be set up with their driving circuit on a breadboard and fed power. If the vibrator vibrates, it's working optimally.

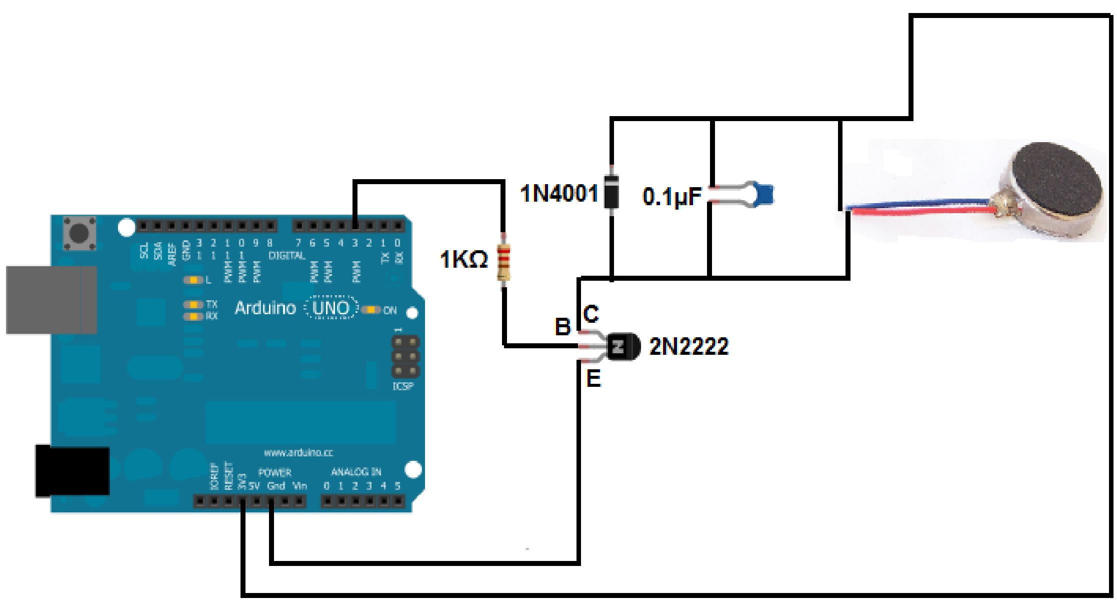


Figure 74. Driver Circuit for the Vibrators

6.1.3. Power Systems

To calculate the maximum power draw required by our system we will test every component for their individual power draws. The vibrators each draw 0.05 Amps regardless of the voltage fed to them, which is lower than was expected. The SR04 ultrasonic sensors each will draw only 0.0033 Amps, which is also very low. The MaxSonar Ez3 Ultrasound sensor draws 0.002 Amps typically. The LIDAR Lite v3 drew much more power, coming in at 0.113 Amps, which is only slightly lower than listed on the specs sheet. Since the microcontroller will be more difficult to test, we will just use the value listed on its specs sheets, which is 0.0002 Amps. The camera module lists its current draw at 0.25 Amps. This can be compiled into an equation:

$$(3.3\ mA \times 3\ SR04) + 2\ mA\ Ez3 + 113\ mA\ LIDAR + 250\ mA\ Camera + (50\ mA \times 7\ Vibrators) + 0.2\ mA\ Microcontroller = 0.725\ Amp\ Peak\ Current$$

This equation describes the maximum current draw the circuit will experience. Given that the system will have buttons giving the user the option to turn off certain sections of the system should they want. This will reduce the load the overall electrical system will experience, to a varying effect depending on what systems are turned off. Additionally, this assumes that all components will be running at peak current draw. The vibrators especially will be requiring less than maximum power during operation because they will operate at lower levels of operation depending on how far their respective sensor is sensing an object.

To ensure that the system can operate at peak power draw without failure, all of the sensors, the vibrators, the Arduino, and the microcontroller will be set up on breadboards. Then it will be fed the power required to operate from a DC power generator, and the temperature will be monitored. If the system begins to heat up after prolonged use, power will be cut and the circuit will be allowed to cool. Any overheating components will require heat sinks, and any overheating wires will need to be replaced with higher quality materials.

If it is determined that too much amperage is being routed through the wire from the battery to the PCB, it can be split between multiple wires feeding to different sections of the PCB. The PCB itself might need a heat sink system, but everything else will most likely be fine.

6.1.4. Computer Vision Subsystem

Considering most physical components to the computer vision aspect will be premade by companies and are more expensive, they will be very reliable components. Despite that, they will be tested to make sure they are working properly. Parts that fall under this subsystem include the Raspberry Pi 3, and the corresponding camera component. These will be tested while the coding process is underway. It will be apparent while writing and testing code whether or not the hardware is functioning correctly.

6.2. Software Testing

The testing process for software testing has multiple steps to it to ensure your system is being optimized. It is important to remember than testing a process, not a single activity.

The first step in the process is planning and control. We must first determine if there are any risks and to identify the goals of testing. One risk in our system is shorting out the system by applying to much voltage, or having the system malfunction in a way that could start a fire if the wiring is done poorly. We then discuss how the test approach will be conducted. This could be unit testing each individual piece, then integration testing, such as adding vibrators to the sensors, and creating software to handle how much vibration we want to have depending on the distance and the rate at which it does it. From here we discuss the required test resources, such as someone to use the device and the type of environment they are in (room with lots of obstacles). We will have control over our testing by a few means. This will be done analyzing results and reviews from the tests and begin to adjust when necessary.

The next step is analysis and design. The testing basis comes from requirements, risk analysis, design requirements and other interfaces. From this we can create the tests we need for analysis and help form the design.
Following this is the implementation and execution step. In this procedure, we create test data for the tests designed previously and group test cases to check for a specific set of behaviors. After this is done, we simply test the environment. Often when testing, tests will fail and should be retested to confirm any fixes. All testing should be documented into a test log, which indicate with a pass/fail of the current test.

The fourth step is evaluating the exit criteria and report. Basically, what this means is to compare our risk assessment to each test level and determine whether enough testing has been done. If all is good, a summary report can be written to stakeholders of the system.
The final step is test closure activities. This allows the creator to check if all deliverables were met, finalize any scripts or testing environments for later, deliver the test ware to who will be doing maintenance, and lately reflect on the testing, to improve for the future. Following these key five steps can drastically improve the testing process of software.

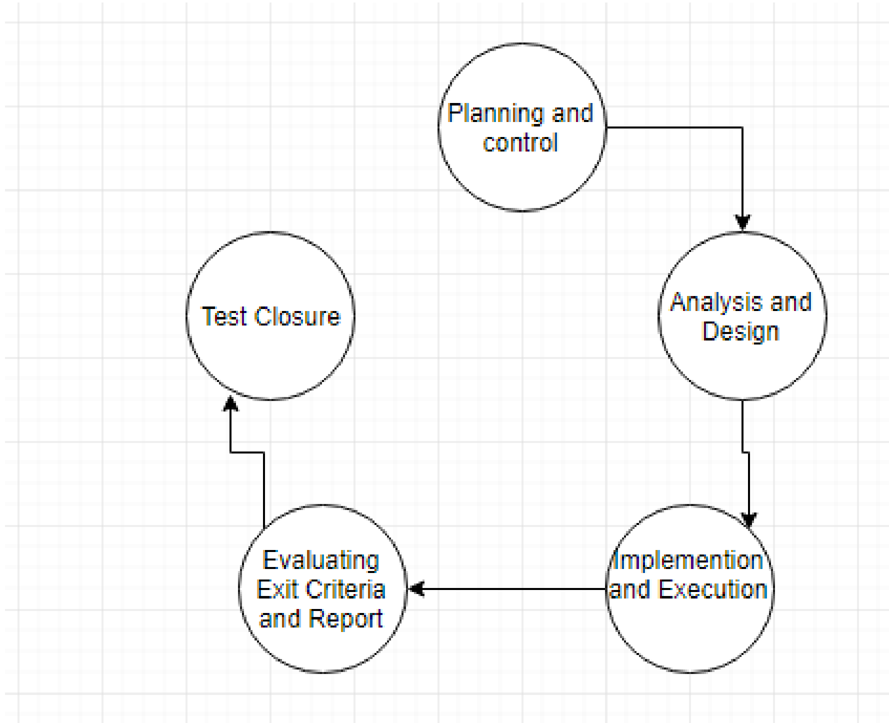


Figure 75. Software Cycle

6.2.1. Software Test Environment

The testing environment for software testing and debugging will be held on the team's individual computers. Each of these computers will be developing on the Windows 10 operating system to ensure similarity across devices and making debugging easier.

6.2.2. Ultrasonic with Air Temperature Testing

Ultrasonic sensor testing is one of our more extensive test as they are the main component in our design. Since ultrasonic sensors can change with the air temperature, we will need to make test each individual sensor to confirm that the temperature sensor is working in sync with each one. Without proper testing and accurate results the user could be lead astray from where obstacles are actually located.

6.2.3. Ultrasonic Ghost Echoes Testing

With an array of ultrasonic sensors, and not being able to use techniques such as sensor fusion, the Batpack must not produce crosstalk or ghost echoes between each sensor. In order to alleviate this issue a single pulse to one of the ultrasonic sensors will be fired and the next one will not be fired until the ultrasonic pulse is known to dissipate, this will be in the matter of milliseconds. This technique will produce less samples overall it should not affect the user experience.

Supplies:
- Microcontroller w/ multiple pins
- Multiple Ultrasonic Sensors
- Debug Screen
- Meter Stick

Preparation:
- Multiple ultrasonic sensors to be hooked up to a microcontroller
- Output to be displayed on a debug screen (the feet measured)

Procedure:
1. Using the meter stick, set up an obstacle for each of the Ultrasonic sensors to detect.
2. Using the debug screen make sure that the distances are correlating correctly with that of the meter stick.
3. Move the obstacle and repeat to confirm the ultrasonic sensors are measuring correctly.

Expected Results:

The distance of the obstacle should be the same, within a couple centimeters, between the ultrasonic sensors and the meter stick. No crosstalk should occur.

6.2.4. Optical Sensor Testing

Much like the ultrasonic sensors, the optical sensor must be tested properly to ensure that the correct distances are being retrieved. The optical sensor must work within the given requirements set above.

Supplies:
- Microcontroller
- LIDAR Lite V3
- Debug Screen
- Meter Stick

Preparation:

- Using I2C, connect the LIDAR to the microcontroller
- Output set to display on the debug screen

Procedure:

1. Using the meter stick, set up an obstacle for the LIDAR sensor to detect.
2. When the program is running check the debug screen to ensure the obstacle is read as the correct distance the meter stick provided.
3. Move the obstacle to different distances and confirm that they are correct and are picked up within the requirements of the Batpack.

Expected Results:

The distance of the obstacle should be the same of that of the meter stick. It should be precise and within a couple of millimeters of the exact result.

6.2.5. PWM Testing

The vibration motors must be tested to make sure that the PWM, described more in section 5.1.3. Given set values the team will have to make sure that the vibrator units are following the appropriate step function programmed for a given distance; this will be the main way the project communicates back to the user. To properly test this, function our boards output for the vibration motors will be attached to an oscilloscope. The function will value will be declared statically to observe the duty cycle. Once set the oscilloscope should output the corresponding cycle with the correct HI and LO amplitude values. The waveform frequency will be measured to confirm that the duty cycle is at its correct form.

Supplies:

- Microcontroller
- Vibration Motor
- Vibration Breadboard Circuit

Preparation:

- Duty cycle assigned for range values
- Test if the vibration motor is properly functioning

Procedure:

1. Pick the first threshold range value (closest distance) that the ultrasonic sensor will give for a given duty cycle.
2. Record the vibration intensity
3. Pick the last threshold range value (farthest distance) that the ultrasonic sensor will give for the duty cycle in step 1.
4. Record and compare the vibration intensity to make sure they are the same.

5. Repeat for each duty cycle in the step function

Expected Results:

The vibration intensity should be the same if the distance is within the same step function range.

6.2.6. Computer Vision Testing

In order to facilitate the computer vision, and allowing the user to know if they are walking across a sidewalk or a grass surface, it must undergo extensive testing. Using the OpenCV's built in Canny Edge Detector function and Hough Line Transform the Batpack will be able to alert the user of the surfaces they are walking near. In order to test the functionality of this feature, pre-processed images will be run through the functions to make sure we have the desired output. These images will be of standard walking surfaces that include outlier variables or objects in the way. Each test will give us a better understanding on how this functionality can be improved to achieve more accurate data.

7. Administrative

All appropriate measures will be covered in this section to ensure that the project is on time and within our budget. This timeline includes everything from the start of our project to the final presentation date. The budget will show all components for this project with their relative information including, quantities, price and manufacturer.

7.1. Budget

Table 3. Budget Analysis

Part Description	**Quantity**	**Price ($)**	**Manufacturer**
Optical Sensor	**1**	**$150**	**Sparkfun Electronics**
Ultrasonic Sensor (HC-SR04)	**3**	**$5.50 ea.**	**Parallax**
Ultrasonic Sensor (LV-MaxSonar-EZ3)	**1**	**$24.95**	**Maxbotix**
Camera (Raspberry Pi Camera Module V2)	**1**	**$29.50**	**Raspberry Pi**
Raspberry Pi v3 Model B	**1**	**$36.00**	**Raspberry Pi**
Backpack	**1**	**$20 - $30**	**Unknown**
Glasses	**1**	**$5 - $10**	**Unknown**
Battery	**1**	**$30**	
Misc (resistors, capacitors, etc…)	**N/A**	**~$15**	**N/A**
Total:	**10**	**$325 - $510**	N/A

The overall budget of the project will come close to $800-$1000. Specifications of the pieces will be added later. This project will be entirely funded by the team without any extra funding from outside sources. In order to stay within the team's budget a careful understanding and research must be taken before ordering any component for the final product.

The design will have multiple sensors in it to track everything going on around the user. The final product will cost roughly up to $500 to make. This does not include the sensors we plan to purchase for testing and comparing to the other pieces.

7.2. Milestones

Shown in Table 2 is the Team's milestones that need to be meet in order for the project to be completed in time for the presentation, in early December. Staying focused and on top of each task before it is due will ensure the success of the overall project, and allow time for additional testing and debugging.

Table 4. Project Milestones

Description	Semester	Start	End	Duration	Days until Due	Percent Complete
Brainstorm Senior Design Ideas	Summer	5/22/17	5/26/17	Due	Past	100%
Confirm Project Idea	Summer	5/27/17	5/27/17	Due	Past	100%
Goals and Objectives for Project	Summer	5/29/17	6/1/17	Due	Past	100%
Initial Project Documentati on	Summer	6/2/17	6/2/17	Due	Past	100%
Research sensor technology	Summer	6/3/17	6/20/17	Due	Past	100%
Research sensor communicati on protocols	Summer	6/3/17	6/20/17	Due	Past	100%
60 Page Documentati	Summer	7/7/17	7/7/17	Due	Past	100%

on Draft						
Power Supply Calculations	Summer	7/06/17	7/17/17	Due	Past	100%
100 Page Documentation Submission	Summer	7/21/17	7/21/17	Due	Past	100%
Final Documentation Due SD1	Summer	8/1/17	8/1/17	Due	Past	100%
Finalize Components	Summer	7/21/17	8/1/17	11 Days	Past	100%
PCB Layout	Summer	7/20/17	8/5/17	16 Days	5	50%
Assign Programming tasks	Summer	7/21/17	7/30/17	9 Days	Past	100%
Order Components	Break	7/1/17	8/7/17	38 Days	7	100%
Initial Testing	Fall	8/21/17	8/26/17	5 Days	26	0%
Build Prototype	Fall	8/27/17	9/16/17	20 Days	47	0%
Test & Debug Prototype	Fall	9/17/17	10/28/17	41 Days	89	0%
Final Documentation Due SD2	Fall	11/11/17	11/25/17	14 Days	117	0%
Finalize Project	Fall	10/29/17	11/25/17	27 Days	117	0%
Final Presentation	Fall	TBD	TBD	1 Day	TBD	0%

Appendix A References

1. Massa, D. (1999). *Choosing an Ultrasonic Sensor for Proximity or Distance Measurement*. [pdf] Peterborough, NH: Massa Products Corp., pp.1-2. Available at: https://www.massa.com/wp-content/uploads/Sensors_February-March_1999.pdf [Accessed 6 Jul. 2017].
2. S. Hill, "5 amazing gadgets that are helping the blind see", *Digital Trends*, 2017. [Online]. Available: https://www.digitaltrends.com/mobile/blind-technologies/. [Accessed: 21- Jul- 2017].
3. D. Yapa, "Devices that help Blind or Visually Impaired person to get around", *Dushyantha Yapa*, 2017. [Online]. Available: https://dushyapa.wordpress.com/2013/06/04/devices-that-help-blind-or-visually-impaired-person-to-get-around/. [Accessed: 21- Jul- 2017].
4. "Helping blind people navigate", *Economist.com*, 2017. [Online]. Available: https://www.economist.com/news/science-and-technology/21723097-new-way-assist-those-poor-eyesight-helping-blind-people-navigate. [Accessed: 21- Jul- 2017].
5. "How do you make walking easier for people who are blind? Put submarine tech in a cane", *Public Radio International*, 2017. [Online]. Available: https://www.pri.org/stories/2014-08-07/how-do-you-make-walking-easier-people-who-are-blind-put-submarine-tech-cane. [Accessed: 21- Jul- 2017].
6. Starlite-intl. (2017). *Lidar Lite v3 Operation Manual*. [online] Available at: http://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf [Accessed 6 Jul. 2017].
7. Maxbotix. (2017). *LV-MaxSonar-EZ Series*. [online] Available at: http://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf [Accessed 6 Jul. 2017].
8. Anon, (2017). Introduction To Color Theory. [online] Available at: http://infohost.nmt.edu/tcc/help/pubs/colortheory/colortheory.pdf [Accessed 1 Aug. 2017].
9. Docs.opencv.org. (2017). Canny Edge Detector — OpenCV 2.4.13.2 documentation. [online] Available at: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html [Accessed 1 Aug. 2017].
10. Docs.opencv.org. (2017). Hough Line Transform — OpenCV 2.4.13.2 documentation. [online] Available at: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html [Accessed 1 Aug. 2017].

11. Wikipedia-Optics. (2017). *Distortion*. [online] Available at: https://en.wikipedia.org/wiki/Distortion_(optics [Accessed 31 Jul. 2017].
12. Brown, Duane C. (May 1966). "Decentering distortion of lenses" (PDF). Photogrammetric Engineering. 32 (3): 444–462.
13. Docs.opencv.org. (2017). Camera calibration With OpenCV — OpenCV 2.4.13.2 documentation. [online] Available at: http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html [Accessed 1 Aug. 2017].
14. GitHub. (2017). davidawad/Lane-Detection. [online] Available at: https://github.com/davidawad/Lane-Detection [Accessed 1 Aug. 2017].
15. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.
16. Sebastian Raschka's Website. (2017). Machine Learning FAQ. [online] Available at: https://sebastianraschka.com/faq/docs/classifier-history.html [Accessed 15 Jul. 2017].
17. Medium. (2017). Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. [online] Available at: https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721 [Accessed 7 Jul. 2017].
18. En.wikipedia.org. (2017). Convolutional neural network. [online] Available at: https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-26 [Accessed 12 Jul. 2017].
19. Fehlhaber, K. (2017). Hubel and Wiesel & the Neural Basis of Visual Perception - Knowing Neurons. [online] Knowing Neurons. Available at: http://knowingneurons.com/2014/10/29/hubel-and-wiesel-the-neural-basis-of-visual-perception/ [Accessed 1 Aug. 2017].
20. Medium. (2017). Object Detection using Deep Learning for advanced users (Part-1). [online] Available at: https://medium.com/ilenze-com/object-detection-using-deep-learning-for-advanced-users-part-1-183bbbb08b19 [Accessed 22 Jul. 2017].
21. Slideshare.net. (2017). Computer Vision. [online] Available at: https://www.slideshare.net/csclub/20141117-laptev-computervision [Accessed 24 Jul. 2017].
22. Quora. (2017). How does Keras compare to other Deep Learning frameworks like Tensor Flow, Theano, or Torch?. [online] Available at: https://www.quora.com/How-does-Keras-compare-to-other-Deep-Learning-frameworks-like-Tensor-Flow-Theano-or-Torch [Accessed 25 Jun. 2017].
23. Pyageng.mpastell.com. (2017). Measurements and Data Analysis for Agricultural Engineers using Python. [online] Available at:

http://pyageng.mpastell.com/book/dsp.html#cid71 [Accessed 1 Aug. 2017].

24. reddit. (2017). What Filter to use for Ultrasonic Sensor? [X post from /r/Arduino] • r/robotics. [online] Available at: https://www.reddit.com/r/robotics/comments/1fjt8b/what_filter_to_use_for_ultrasonic_sensor_x_post/ [Accessed 1 Aug. 2017].
25. Czerniak, G. (2017). Greg Czerniak's Website - Kalman Filters for Undergrads 1. [online] Greg.czerniak.info. Available at: http://greg.czerniak.info/guides/kalman1/ [Accessed 1 Aug. 2017].
26. Transcat. (2017). *Why use a Laser Distance Meter?*. [online] Available at: http://www.transcat.com/media/pdf/cordex-laser-distance-meters.pdf [Accessed 6 Jul. 2017].
27. Learn.sparkfun.com. (2017). I2C - learn.sparkfun.com. [online] Available at: https://learn.sparkfun.com/tutorials/i2c [Accessed 21 Jul. 2017].
28. Cs231n.github.io. (2017). *CS231n Convolutional Neural Networks for Visual Recognition*. [online] Available at: http://cs231n.github.io/neural-networks-1/ [Accessed 1 Jul. 2017].
29. En.wikipedia.org. (2017). HSL and HSV. [online] Available at: https://en.wikipedia.org/wiki/HSL_and_HSV [Accessed 1 Aug. 2017].
30. "Automatic Thresholding", 2017. [Online]. Available: http://www.math.tau.ac.il/~turkel/notes/otsu.pdf. [Accessed: 01- Aug-2017].
31. C18.2M. (2007). [ebook] Rosslyn: National Electrical Manufacturers Association, pp.1-8. Available at: https://www.ansi.org/ [Accessed 1 Jul. 2017].
32. Power Supply Safety Standards, Agencies and Marks. (2014). [ebook] Tualatin: CUI Inc., pp.1-18. Available at: http://www.cui.com/catalog/resource/power-supply-safety-standards-agencies-and-marks.pdf [Accessed 1 Jul. 2017].
33. IPC-2221A Generic Standard on Printed Board Design. (2003). [ebook] Northbrook: Institute for Printed Circuitboards, pp.1-73. Available at: http://www.ipc.org/ [Accessed 1 Jul. 2017].
34. Soldered Electrical Connections. (1997). 2nd ed. [ebook] Washington D.C.: NASA, pp.1-80. Available at: https://nepp.nasa.gov/docuploads/06AA01BA-FC7E-4094-AE829CE371A7B05D/NASA-STD-8739.3.pdf [Accessed 1 Jul. 2017].

Appendix B Author Communications

From: Gregory Czerniak <gregczrk@gmail.com>
Sent: Thursday, July 6, 2017 9:14 PM
To: Lucas Pasqualin
Subject: Re: Permission to use graphic on Kalman Filter from your website.

Hi Lucas,

You have my permission to use the Kalman images in your senior design paper.

Thanks,
Greg Czerniak

On Thu, Jul 6, 2017 at 9:06 PM, Lucas Pasqualin <lucas.pasqualin@knights.ucf.edu> wrote:
Hello!

My name is Lucas Pasqualin and I am a computer engineering student at the University of Central Florida. I'm currently working on my senior design paper, and I found some very relavent information here:
http://greg.czerniak.info/guides/kalman1/

Would it be ok with you if I included some of your images in my paper? My paper will not be published in any academic journals or anything of that nature.

Thank you!
Lucas Pasqualin

Greg Czerniak's Website - Kalman Filters for Undergrads 1

greg.czerniak.info

Preparation. The next step is to prepare the Kalman filter inputs and constants. Since this is a single-variable example, all matrices are 1x1. Matrix "A" is what you ...

From: Pastell Matti (Luke) <matti.pastell@luke.fi>

Sent: Thursday, July 6, 2017 8:58 PM
To: Lucas Pasqualin
Subject: Automaattinen vastaus: Permission to Use Graphs in my Senior Design Paper

Hei,
Kiitos viestistäsi.

Olen lomalla 25.7. asti.

Thanks for contacting me. I´m on vacation until 25th of July.

*Ystävällisin terveisin/ Kind regards
Matti Pastell

From: Lucas Pasqualin
Sent: Thursday, July 6, 2017 1:39 PM
To: enquiries@precisionmicrodrives.com
Subject: Permission To Us graphs in class Paper

Hello,

I'm a Computer Engineering student at the University of Central Florida and I'm writing a technical paper on my Senior design project. I found some really useful information here:
https://www.precisionmicrodrives.com/application-notes/ab-012-driving-vibration-motors-pulse-width-modulation

AB-012 : Driving Vibration Motors with Pulse Width ...

www.precisionmicrodrives.com

How to use Pulse width Modulation (PWM) with vibration motors. See how PWM signals can vary the speed and amplitude of vibration motors, or to add haptics

And I was hoping to get permission to use some of the graphs and images found in this presentation.

My senior design paper will not be published in any journals or anything of that nature.

Thank you so much and please contact me if you need more information regarding my project!
Lucas Pasqualin
Computer Engineering B.S.P.E
University of Central Florida

From: Slinger Fu <info@nfpmotor.com>
Sent: Thursday, July 6, 2017 10:15 PM
To: Lucas Pasqualin
Subject: Re: Permission To Use Images found in Power Point in class Paper

Dear Sir,

Thanks for your request about the permission to use some of the graphs and images.

Of course you can use it, please note that our website www.nfpmotor.com

Need For Power Motor - Micro DC Motors

www.nfpmotor.com

Need-For-Power Motor Co., Ltd, is an experienced ISO9001 supplier of micro DC vibration motors with factory in Shenzhen, China and was found in 2005.

If possible it's appreciate to get the senior design paper to learn from your project.

Thanks a lot!

Slinger

2017-07-07 1:36 GMT+08:00 Lucas Pasqualin <lucas.pasqualin@knights.ucf.edu>:
Hello,

I'm a Computer Engineering student at the University of Central Florida and I'm writing a technical paper on my Senior design project. I found some really useful information here:

http://www.nfpmotor.com/The%20Differences%20Between%20ERM%20and%20LRA%20Actuators.pdf

Haptics - Need For Power Motor

www.nfpmotor.com

Haptics Solutions for ERM and LR A Actuators Overview The eccentric rotating mass (ERM) motor and linear resonant actuator (LRA) are two of the most common types of

And I was hoping to get permission to use some of the graphs and images found in this presentation.

My senior design paper will not be published in any journals or anything of that nature.

Thank you so much and please contact me if you need more information regarding my project!

Lucas Pasqualin
Computer Engineering B.S.P.E
University of Central Florida

--

Need-For-Power Motors Co., Ltd.
Address: Building 4, Room 205,
Zhoushi Rd 108th, Shiyan,
Bao'an, Shenzhen, 518109
Guangdong, China.
Web: www.NFPmotor.com
Tel: 86.18660037716
Skype: fu.slinger
Contact by E-mail:
info@nfpmotor.com,

From: Caleb Kirksey <ckirksey3@gmail.com>
Sent: Thursday, July 20, 2017 1:18 PM
To: Lucas Pasqualin
Subject: Re: lane-detection-with-opencv

Yeah, go for it!

On Thu, Jul 20, 2017 at 10:05 AM Lucas Pasqualin <lucas.pasqualin@knights.ucf.edu> wrote:

Hello!

My name is Lucas Pasqualin and I'm a Computer Engineering student at the University of Central Florida. I'm wrapping up my senior design project, and I've found some really good resources on your github repo (lane-detection-with-opencv). I wanted to get permission to use some of your (edited) code and perhaps some of the images in my paper. The paper will not be published in any academic journals, if that makes any difference.

Thank you!

Lucas Pasqualin

From: support.49769.123fc26489d2e898@helpscout.net <support.49769.123fc26489d2e898@helpscout.net> on behalf of Adrian at PyImageSearch <ask.me@pyimagesearch.com>
Sent: Friday, July 7, 2017 4:42 PM
To: Lucas Pasqualin
Subject: Re: Permission to use recourses in Senior Design Paper

Hi Lucas,

Congratulations on getting to your senior year, I know that represents a lot of hard work!

As for whether you can use anything in the PyImageSearch.com blogs in your own research, that would be perfectly fine. Thanks for asking!

Unless your institution prefers a different style, you can use this citation format in research papers:

Adrian Rosebrock, *Face Alignment with OpenCV and Python,* (retrieved June 25, 2017),http://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/

Your ideas for making the Canny detector easier or more effective sound pretty clever themselves. Why not try them out with some of your own images, and see what you discover? There might even be a research paper in it, if you engineer something useful.

Please let us know if you have any other questions.

--
David McDuffee
Correspondence Coordinator
ask.me@pyimagesearch.com
On Thu, Jul 6, 2017 at 11:48 PM EDT, Lucas Pasqualin
<lucas.pasqualin@knights.ucf.edu> wrote:

Hello,

My name is Lucas Pasqualin and I am a Computer Engineering student at the University of Central Florida. I'm currently working on my senior design paper and I noticed some really good recourses (and code) on your blog post here:http://www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/

I was hoping you wouldn't mind if I included these resources in my paper. My paper will not be published in any academic journals or anything of that nature.

Thank you,
Lucas Pasqualin

P.S. Feel free to ignore this bit, but I thought that your auto_canny function was incredibly clever. Do you think there are any opportunities for using a different statistical estimator for your threshold basis? I was considering trying out some results with just the average,mode, etc. but perhaps there is something more clever.

Zero-parameter, automatic Canny edge detection with Python ...

www.pyimagesearch.com

PyImageSearch - Be awesome at learning OpenCV, Python, and ...

www.pyimagesearch.com

This OpenCV and Python blog is written by Adrian Rosebrock. Learn OpenCV, Python, and computer vision through my OpenCV articles, tutorials, and guides.

Today I've got a little trick for you, straight out of the PyImageSearch vault. This trick is really awesome - and in many cases, it completely alleviates the ...

From: Alexander Castel

Hello Lucas,

You are free to use the information from our website, as long as you make reference to the source of information (ie: link to our website)

Please note that the information on our website is for indicative information purposes only, and we cannot be held responsible for its accuracy.

Good luck with your design project.

Kind Regards,

Alexander Castel

Sales Director - Precision Microdrives Ltd

T: +44 (0) 1932 252482

F: +44 (0) 1932 325353

E: alexander.castel@precisionmicrodrives.com

W: www.precisionmicrodrives.com

TEL: +44 (0) 1932 252 482

WEB: www.precisionmicrodrives.com

Precision Microdrives - Leading suppliers of miniature DC motors, gear motors & vibration motors.

Registered in England and Wales No. 5114621. Unit 1.05, Canterbury Court, 1 Brixton Road, London, SW9 6DE, United Kingdom

----

Sign-up to our super useful monthly technical email here, or read our technical blog which contains details of latest technical developments, and guidance for applications.

Please note our critical component and life support policy. This is on the final page of every datasheet and in our terms and conditions of sale. A critical component is one which could cause damage to property, or harm to living beings, if it were to fail. You must disclose to us if you are using, or intending to use one of our parts as a critical component in an application. This is so that we can run a risk assessment, and provide you with parts that are most suitable for the application.

Name *

Michael Ferraro

Email *

kickflipper2448@aol.com

Company

School

Phone

(555) 555-5555

Inquiry *

My name is Michael Ferraro and I am a Computer Engineering student at the University of Central Florida. I'm currently working on my senior design paper and I noticed some really good resources (and code) on your blog post here:
https://www.massa.com/wp-content/uploads/Sensors_February-March_1999.pdf

Tony Ferraro <ferraro2ndbase@gmail.com> 8:51 PM (1 hour ago)
to sales

Hello,

My name is Michael Ferraro and I am a Computer Engineering student at the University of Central Florida. I'm currently working on my senior design paper and I noticed some really good resources (and code) on your blog post here:

http://www.transcat.com/media/pdf/cordex-laser-distance-meters.pdf

I was hoping you wouldn't mind if I included these resources in my paper. My paper will not be published in any academic journals or anything of that nature.

Thank you,
Michael Ferraro

Tony Ferraro <ferraro2ndbase@gmail.com> 8:53 PM (1 hour ago)
to info

from: **Tony Ferraro** <ferraro2ndbase@gmail.com>
to: info@maxbotix.com
date: Mon, Jul 31, 2017 at 8:53 PM
subject: Project
mailed-by: gmail.com

Hello

My n student at the University of Central Florida. I'm currently working on my se es (and code) on your blog post here:

http:/ heet.pdf

I was my paper. My paper will not be published in any academic journals or anything of that nature.

Thank you,
Michael Ferraro

Tony Ferraro <ferraro2ndbase@gmail.com> 8:58 PM (1 hour ago)
to customerservice

from: **Tony Ferraro** <ferraro2ndbase@gmail.com>
to: customerservice@sparkfun.com
date: Mon, Jul 31, 2017 at 8:58 PM
subject: Project
mailed-by: gmail.com

Hello,

My name is Mich he University of Central Florida. I'm currently working on my senior desigr e) on your blog post here:

https://learn.spa

I was hoping you My paper will not be published in any academic journals or anything of that nature.

Thank you,
Michael Ferraro

Tony Ferraro <ferraro2ndbase@gmail.com> 9:15 PM (1 hour ago)
to captyapa

Hello,

My name is Michael Ferraro and I am a Computer Engineering student at the University of Central Florida. I'm currently working on my senior design paper and I noticed some really good resources (and code) on your blog post here:

https://dushyapa.wordpress.com/2013/06/04/devices-that-help-blind-or-visually-impaired-person-to-get-around/

I was hoping you wouldn't mind if I included these resources in my paper. My paper will not be published in any academic journals or anything of that nature.

Thank you,
Michael Ferraro

Tony Ferraro <ferraro2ndbase@gmail.com> 10:04 PM (14 minutes ago)
to agetget

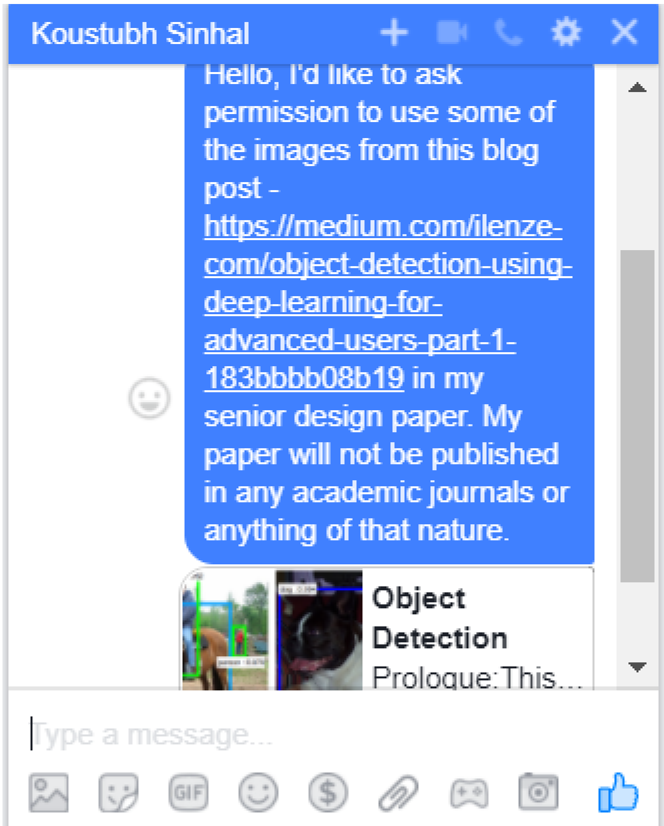
Hello,

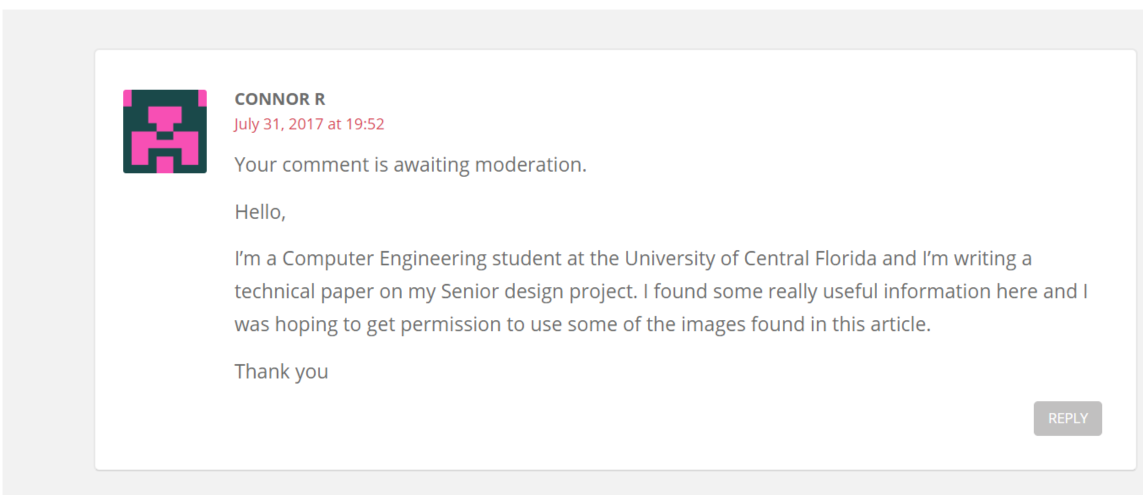
My name is Michael Ferraro and I am a Computer Engineering student at the University of Central Florida. I'm currently working on my senior design paper and I noticed some really good resources (and code) on your blog post here:

https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

I was hoping you wouldn't mind if I included these resources in my paper. My paper will not be published in any academic journals or anything of that nature.

Thank you,
Michael Ferraro


Koustubh Sinhal
Hello, I'd like to ask permission to use some of the images from this blog post - https://medium.com/ilenze-com/object-detection-using-deep-learning-for-advanced-users-part-1-183bbbb08b19 in my senior design paper. My paper will not be published in any academic journals or anything of that nature.
Object Detection
Prologue:This…
Type a message...
GIF


CONNOR R
July 31, 2017 at 19:52
Your comment is awaiting moderation.
Hello,
I'm a Computer Engineering student at the University of Central Florida and I'm writing a technical paper on my Senior design project. I found some really useful information here and I was hoping to get permission to use some of the images found in this article.
Thank you
REPLY