UNIVERSITY OF WATERLOO
Faculty of Engineering
Department of Electrical and Computer Engineering

ECE Fourth Year Design Project Final Report

Low Cost Optical Positioning System for Small Scale Robots

Project 50

Project Member	Signature
Ilia Baranov	
Justin Godson	
Naim Hilal	
Saad Kazi	

This report is submitted as the final report deliverable for the ECE.492B course. It has been written solely by us and has not been submitted for academic credit before at this or any other academic institution.

Consultant:
Steven Waslander

Date Submitted:
Feb 15th, 2013

Abstract

Modern robotics requires localization and mapping to be useful in the real world. Current optical sensing solutions target high cost, high performance industrial markets. Advances in modern optical sensing allow for lower cost and higher accuracy results, which have yet to be integrated into the hobbyist, research, and educational markets. Key factors for these neglected markets are the requirements for user friendliness, customizability, and quick integration. The proposed solution aims to address these needs with a low cost, lightweight, integrated system. This solution employs an optical range finding system which uses a camera and laser, combined with image processing, to compute distance vectors. These vectors are then processed by a computing platform, in order to produce a continuous map of the local area. This device would ideally be mounted on an autonomous robotic platform to provide a proof of concept. [1]

Acknowledgements

We would like to thank Steven Waslander for volunteering his time and expertise as a consultant for this project. We also would like to thank the University of Waterloo for funding some of the PCB manufacturing and casing costs. Additionally, we would like to acknowledge Brijesh, from www.negtronics.com, for providing some guidelines for code in the DCMI module in the STM32F4 microprocessor. Lastly, STM Electronics and Omnivision Inc. provided technical documentation and guidelines in the use of their products.

Glossary

3D: Three Dimensional

BIO: Binary Input and Output data

CPLD: Complex Programmable Logic Device

CD: Camera Data pins

DCIM: Digital Camera Images

EOL: End Of Life

GPIO: General Purpose Input and Output pins

HSYNC: Horizontal Synchronization line

I2C: Inter-Integrated Circuit, a unipolar clock and data serial communication bus developed by Philips.

IDE: Integrated Development Environment

IMU: Inertial Measurement Unit

LED: Light Emitting Diodes

LIDAR: Light Detection And Ranging

MCU: MicroController Unit

PC: Personal Computer

PCB: Printed Circuit Board

RGB: Red, Green, Blue colour space.

RTOS: real-time operating system

SAD: SRAM Address pins

SCL: Serial Clock

SDA: Serial Data

SLAM: Simultaneous Localization And Mapping

SRAM: Static Random Access Memory

USB: Universal Serial Bus

VHDL: Very-high-speed integrated circuits Hardware Description Language

Table of Contents

List of Figures

List of Tables

1. Introduction

Group 50 is designing and building a Low Cost Optical Positioning System for Small Scale Robots. [2]

1.1.Motivation

The motivation for this project is that advanced modern robotics systems require positioning to operate. Mobile platforms require constant awareness of their surroundings, and thus need a system for persistent mapping. Even stationary robotic arms, such as those used in the automotive industry require positional awareness. Many systems around the world employ various techniques to meet this need, however among these many solutions, optical sensing has consistently proven to be one of the most effective. LIDAR units are some of the best optical sensing devices for use in simultaneous localization and mapping (SLAM). The problem with LIDAR and many other optical sensing units is their high cost, high performance metric; for example the Hokuyo URG-04LX-UG01 scanning laser rangefinder, Hokuyo's cheapest LIDAR unit, costs over 1200$ [3] . These systems are targeted at industrial markets, while staying well out of the practical reach of hobbyist, educational and research markets. There exists a strong desire and need for a cheap, user friendly, low barrier to entry SLAM system in these markets. Modern advances in optical sensing, such as higher resolution image sensors, have yet to be integrated into robotic sensing systems. A cheap, effective, and user friendly system providing SLAM functionality, in a lightweight form factor, could be designed to provide a solution to the positioning problem that has yet to be addressed in these markets.

1.2.Users

Intended users for the system stem from the various aforementioned target markets; these users would attach our sensor to a robot intended for indoor use.

1.2.1. Educational

Many educational robotics programs are limited in the scope of what can be practically taught on an educational budget. Often many advanced concepts are not covered simply due to the cost of required equipment. In a class of 30, equipping each student with the before

mentioned Hokuyo LIDAR unit would cost around $36000, which is far too expensive for this market. Access to a cheap LIDAR or equivalent system would facilitate the teaching of these concepts.

1.2.2. Hobbyist

The average hobbyist is an individual who usually is limited financially, but unlimited when it comes to passion and enthusiasm. Having access to more advanced sensing and positioning systems would allow hobbyists around the world to design better systems, and realize even more ambitious projects. In this market, it is also important to provide an easy to use device that can be integrated into an existing system quickly.

1.2.3. Research

Much research is being conducted worldwide employing robotic systems. Again, many of these systems require localization and mapping. Access to a cheap SLAM system would open doors to new research opportunities in previously unexamined areas. The low cost and small size would also enable research in the field of multiple, cooperating robots. This field is under intensive study today, and does require such a system.

1.3.Functional Requirements

Functional requirements for this project are summarized in Table 1.

Table 1: Functional Requirements

Requirement	Priority	Description
Distance Measurement	Must	The optic sensor must be able to detect distance to arbitrary objects.
Local Positioning	Must	The system must be able to provide a local position, relative to a reference point.
Persistent Mapping	Must	The system must be able to provide a local 2D map of the area, which is built and updated in near real time.
Data Format	Should	The system should be able to provide processed and raw data to the user.

1.4.Non-Functional Requirements

Non-Functional Requirements for this project are summarized in Table 2.

Table 2: Non-Functional Requirements

Requirement	Priority	Description
Safety	Must	Emitted electromagnetic waves must be non-harmful to the human eye.
Weight	Must	The entire unit must be less than two pound.
Size	Should	The size should be a cylindrical shape with diameter between 5-10cm, and 7cm thickness. The prototype may be larger.
Cost	Should	Should be affordable within the budget of the average hobbyist when produced in mass. <$100
Complexity	Should	The required reading and total setup time for new users should be less than 3 hours.
Versatility	Should	Camera and computational module should be detachable and attachable to multiple mountable vehicle types.
Low Power Consumption	Should	The system should be designed with appropriate components to facilitate minimum power consumption.
Usability	Should	Should be usable in an indoor environment. Hence, no waterproofing or combating direct sunlight.

1.5.Block Diagram

The sensor consists of a motor with a rotating platform. The plat form has an optical emitter and receiver, as shown in Figure 1. [2]

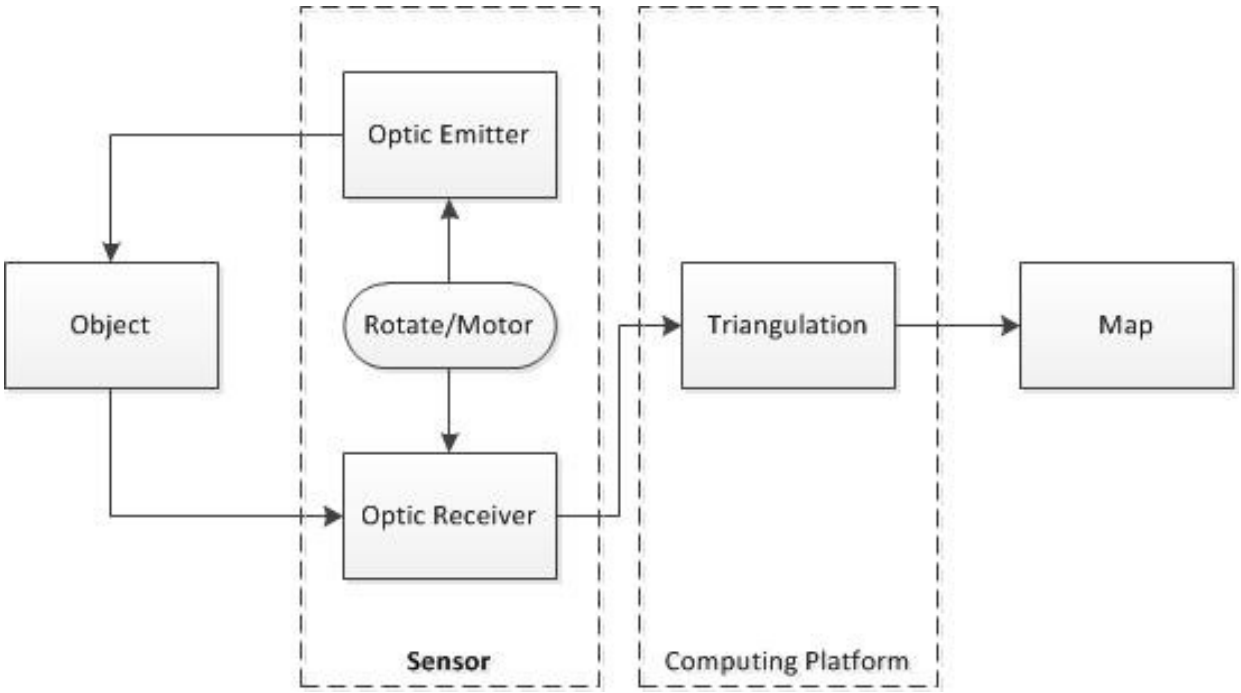

Figure 1: High-Level Block Diagram

The emitter is a laser that is projected onto an object, and the receiver is a camera. The laser dot is then detected in the image, and depending on its distance from the center of the image, triangulation is used to determine the distance of the object. This triangulation is showed in Figure 2.

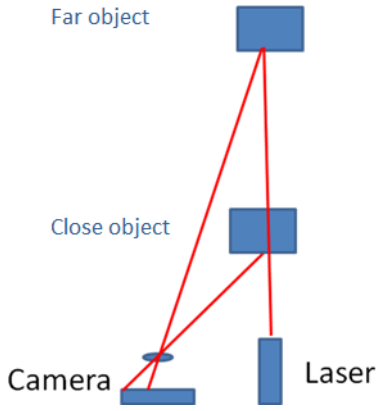

Figure 2: Basic Triangulation of Objects with Different Distance

1.6.Risks

List of potential risks associated with this project are listed in Table 3.

Table 3: Potential Risks

Level	Description	Response
High	Visual image processing too slow for real time motion capture.	Slow down the rotation of the sensor motor or find a more powerful processor.
High	Exposure time for the camera is too long.	Slow down the rotation of the sensor motor or find a more adjustable camera.
High	Camera module falls outside price bracket and does not meet resolution requirements for accurate pixel measurements.	Find another form of light data capture technology or purchase another camera sensor of higher quality.
Medium	Lack of research to be found on mapping algorithms.	Inquire into local professors that work with image technology for their input.
Medium	Direct eye exposure to laser may cause eye damage	Use a class 3 or lower laser with pulsing.
Low	Parts requested from companies do not arrive on time.	Demand expedite delivery or find local vendors for similar parts.

1.7.Plan

A project schedule is shown in Figure 3.

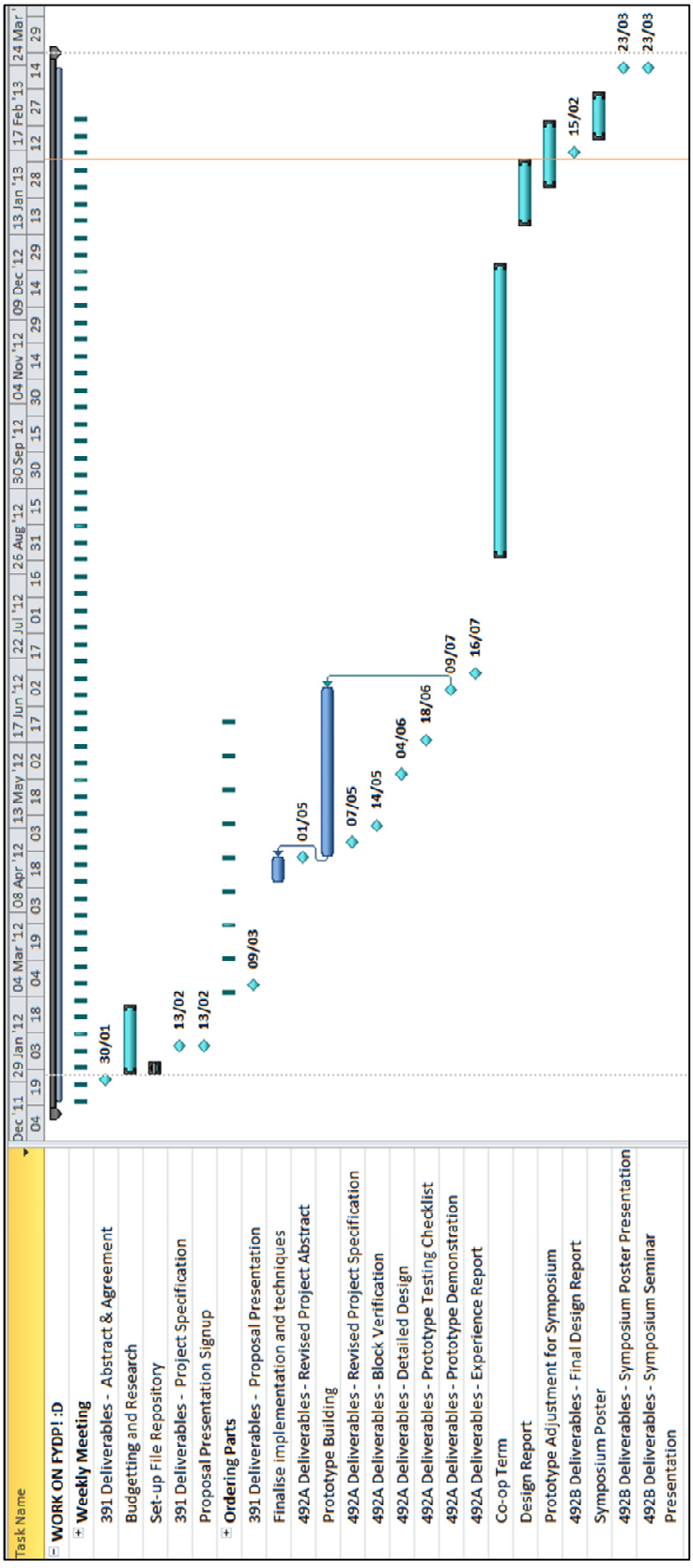

Figure 3: Project Schedule

1.8.Cost

Our first prototype will evidently be more expensive than the final, mass produced version. One of our main goals is to achieve sub $100 for the mass produced cost. For our prototype, the cost will most likely be three times this cost, ~$300. This cost does not include the cost of development boards and repeated prototypes for the various components. After the initial prototype, the cost of the second prototype should be closer to the final cost. A full budget breakdown is available in Appendix A.

2. High-Level Analysis

This section will discuss high level design that went into Group 50's project. It is presented in a chronological order, with the older design conducted prior to design work deliverable shown first. In the detailed design section, the exact components used in the final design are presented. The changes were required either to better satisfy requirements, or to improve the design for Group 50.

2.1.System Level Architecture

The following diagram presents a high level look into the physical connection between major components in the system. The diagram also contains specific component designations for those components which have been selected. [4]

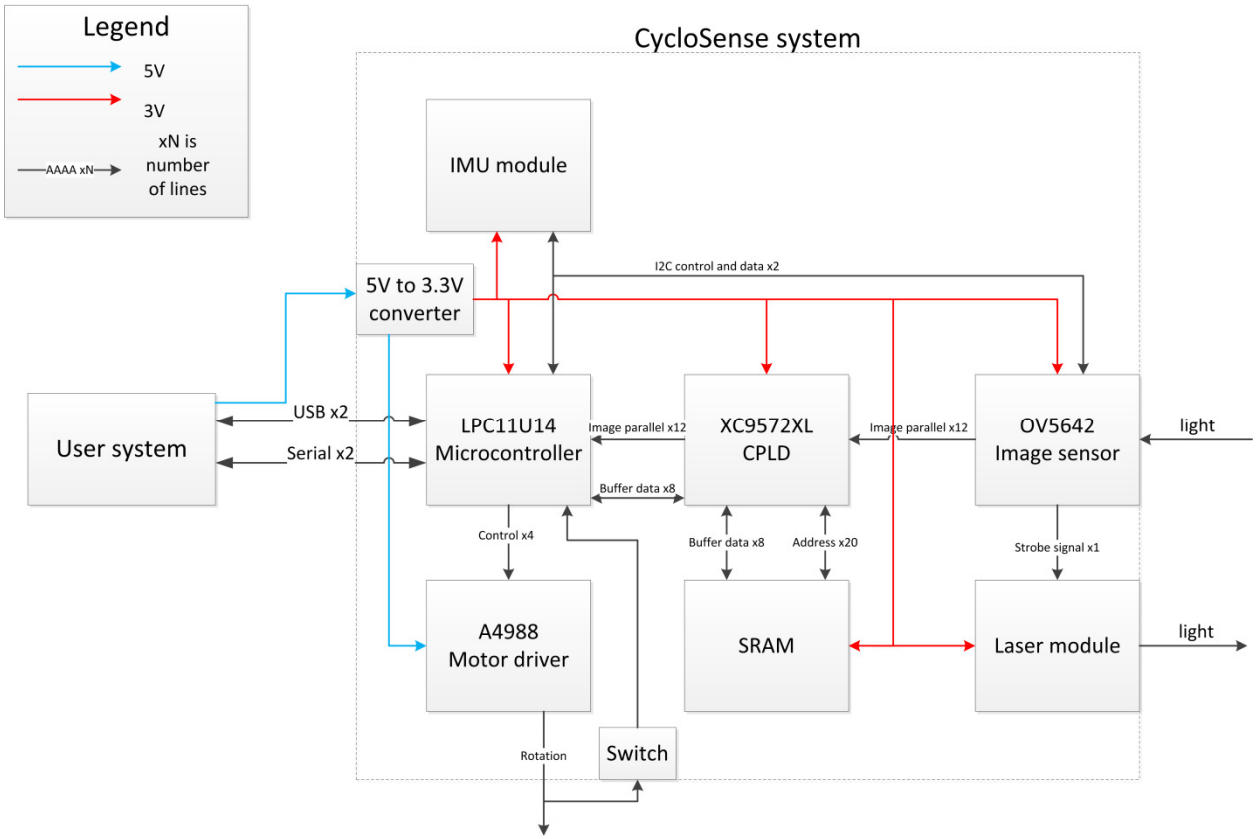

Figure 4: System Level Block Diagram

Each of the major components sections shown above must satisfy the following constraints, outlined in Table 4. These constraints have been put in place using the background knowledge of each team member, relating to their specific fields. While the following table discusses the

component sections in general, the specific parts selections noted in Figure 4 have been found to adequately satisfy the constraints. The specific reasoning for each constraint, and how each component meets these various constraints is discussed in detail in the following sub-sections.

Table 4: Component Constraint Table

Component	Constraint
Global constraints for all components	1. Low cost 2. Available in low quantities 3. Package solder-able by hand
Voltage	1. Should achieve 80% or higher efficiency 2. Must be able to handle 0.5A 3. Should take as little space as possible
Microcontroller	1. Must be sufficiently fast to process data in a timely fashion 2. Must have hardware serial and USB implemented 3. Should be easy to program, with and easy to use debugger and programming environment
CPLD	1. Must be fast enough to communicate with image sensor 2. Should be simple to program 3. Should be small
Stepper Motor Driver	1. Easy to interface with 2. Able to source sufficient current
SRAM	1. Must be fast enough to store image data 2. Should be easy to communicate with
Image Sensor	1. Must be easy to interface with 2. Should have the largest resolution possible 3. Should be well documented and supported
Laser module	1. Should be sufficiently powerful to provide adequate illumination 2. Must be controllable y the Microcontroller
IMU	1. Should communicate using I^2C 2. Must update at least once per second 3. Should provide enough accuracy to be usable indoors

2.2.Assessment of Component Compatibility

The following sections evaluate each component against the specified constraints in Table 4.

2.2.1. Voltage Converter

Previously, the design for the 5V to 3.3V converter was not finalized. However, as detailed later, changes in the design brought about use of a different regulator. Currently a 7-24V to 5V [5]

and a 5V to 2.8V [6] converter is employed. Again, the reasoning for this is detailed later; however it should be noted that both of these converters achieve the requisite efficiency and amperage.

2.2.2. A4988 Stepper Motor Driver & Bipolar Stepper Motor

The platform that the sensor will be resting on is moved by a bipolar, single shaft stepper motor. It has a radial force of 28N and a minimum step size of 1.8 degrees (0.0314rads) [7]. Given that the expected weight of the sensor will be less than 2 pounds (0.907Kg) and the maximum diameter of the sensor is 10cm [2] we can calculate if the force required is below the maximum radial force of the motor.

Let us assume the maximum speed of the motor is required such that it may perform a full turn within 10 seconds. This allows 0.027 seconds per degree, which translates to approximately 0.05s for 1.8 degrees. The distance to be covered is 0.10m*0.0314rads = 0.00314m. Assuming we will halt at every degree to allow exposure for the image sensor, then the sensor must move from rest to half the minimum step distance (0.00157m) in half the time for a step (0.025s), which results in an acceleration of 2.5m/s and a force of 2.27N. This force requirement is well below the available force of the stepper motor.

Bipolar stepper motors are prone to static friction, but since our motor will be starting and stopping within such a small time frame we have essentially implemented a practical form of dithering. The rated electric requirements for the motor are 2.7V and 1A [7]. These rated conditions are met by the motor driver chip, which can supply ±2A and 35V [8]. The motor has 4 leads for input and output, and so does the pin out of the motor driver interface [7] [8].The motor driver is capable of Mixed Decay Mode to constrain sinusoidal reference signal distortion, which is an important risk when controlling stepper motors [8]. The motor driver sits on a Quad-Flat-No-Leads package, but comes pre-soldered on a PCB board with pin outs, which make the physical leads available to both the microcontroller and the stepper motor. To operate the motor driver, a single input pulse to the "STEP" lead converts to a physical step movement on the rising edge [8]. The frequencies we expect from the assumptions made

above about speed should be reasonable, based on personal previous experimentation with both the driver and the motor.

The driver has a 3.3 to 5V logic supply requirement [8] from the microcontroller, and this is within the rated pin out voltages of the selected microcontroller. In summary, the stepper motor is physically fit for the fine movements required by the sensor base and has a compatible interface with the motor driver. The motor driver's electric requirements are met by the microcontroller's physical interface.

2.2.3. Image Sensor

As will be covered in the rest of the document, the image sensor ties the rest of the components together. This image sensor must provide a high enough resolution, in order to provide good accuracy for laser range finding. The image sensor chosen is the OV5642 [9]. This sensor accepts 3VDD. This is slightly lower than system voltage of 3.3V; however we will employ a low drop out regulator to provide 3V from 3.3V. The image sensor produces data in a simple to read, 16bit RGB format. Also, the sensor accepts I2C compatible commands to function. Group 50 has contacted Omnivision, the manufacturer of the sensor, and signed a Non-disclosure agreement (NDA). The documentation provided as a result of the NDA has been sufficiently in depth to provide confidence in the solution proposed. However, further specifications on the Image sensor cannot be publicly released. Thus, this sensor is an optimal choice for our purposes. The cost of the sensor in single quantities is approx. $14 in single quantities [10].

2.2.4. Laser Module

The laser module is a standard, red laser pointer module. There are no specific requirements on the device, other than an operational voltage of 5V or 3.3V. Thus, this component is not precisely specified in this design document, as any laser module should provide sufficient luminance for our system.

2.2.5. Inertial Measurement Unit (IMU)

The IMU will be implemented using an accelerometer, magnetometer and gyroscope. The STMicroelectronics LSM303DLHCTR was selected as the3 axis accelerometer and magnetometer. The

STMicroelectronics L3G4200DTR was chosen as the gyroscope. Both of these chips run on 3.3V (provided by the 5V to 3.3V converter). They are both small in size and light in weight. The LSM303DLHCTR has both accelerometer and magnetometer in the same IC. Both chips are fairly cheap, cost for both is between $12.50 and $21.50 [11] [12] depending on the source. Both ICs have I2C architecture, where the master bus will be driven by the microcontroller. The LSM303DLHC can run at a rate of 1, 10, 25, 50, 100, 200, and 400Hz [13]. L3G4200D can run at a rate of 100, 200, 400 or 800Hz [14]. These rates should be faster than the camera/motor, and thus receiving new IMU data for every distance point. A Kalman filter will run on the microcontroller, in order to stabilize the data from the IMU in order to provide accurate positioning.

2.2.6. Microcontroller, SRAM and CPLD

The microcontroller (MCU) controlling the sensor system was an NXP MCU, the LPC11U14. This MCU is an ARM Cortex-M0 running at up to 50 MHz, with a USB 2.0 device controller. The chip contains 32 KB of flash memory, and 6 KB of SRAM (of which 2 KB is reserved for USB). Serial interfaces include both a USART and I2C [15]. The chip also supports up to 54 general purpose input / output (GPIO) pins [15], of which 12 are required for obtaining image data from the CPLD, and additional 8 for buffer data lines to the CPLD, and finally 4 are required to control the motor driver. Of the up to 54 GPIO pins only 24 are required for the design, leaving 30 additional pins. System functional compatibility of the part is met as the chip contains sufficient GPIOs, the required I2C for interfacing with the image sensor, as well as the USB and serial interfaces for communicating with the user system.

Most importantly, this MCU costs just under $4.00 in quantities of 1 [16], easily meeting our project's low cost requirement. At the time of writing this document, ample stock exists at Digikey and Avnet, as well as several other suppliers [11].This particular chip is relatively young

in NXP's lineup, with an end of life (EOL) sometime in 2016. Another aspect to note is that the 11U14 it is pin compatible with NXP's Cortex-M3 MCUs allowing for an easy substitution if ever there were a stock shortage of the 11U14.

The only cause for concern with this chip was the low internal memory. The 11U14 only contains 6 KB of SRAM, of which only 4 KB is usable (as 2 KB is reserved for USB). A full image from the image sensor, filtered on red to half width at 20 pixels wide, is approximately 15.8 KB, which is larger than what can be loaded into the 11U14's internal SRAM. To counter this, the image processing software running on the 11U14 would need to load and analyze smaller parts of a full image, piece by piece. If 3 KB is used to load image data, and 1 KB is used for the rest of the running program, the 15.8 KB image can be processed in 6 chunks. This was attempted, and it was found that the real time operating system (RTOS) and image processing code required more memory than was available on the chip. Due to this a new MCU was selected. This is detailed in the following section, "Design Changes".

The chosen CPLD for our sensor platform was a Xilinx XC9572XL. This CPLD runs on 3.3 V, contains 72 macrocells (with 1,600 usable gates), as well as 5 V tolerant I/O pins [17]. As the gate propagation time for this CPLD is typically 10.0 nanoseconds, the exact same as the SRAM the CPLD will be interfacing with, there should be no timing issues or delays. For the tasks the CPLD will be performing (memory interfacing), 72 macrocells is more than sufficient for logic. The only remaining concern is supply. At the time of writing this document, Digi-key's stock levels were at over 8000 [18]. As mentioned however, problems were encountered with the CPLD and MCU RAM. Essentially, during the design it was found that the CPLD did not possess sufficient macrocells required for the desired functionality of the part. Due to this, and a lack of onboard RAM in the MCU it was decided that these parts would be replaced. This is detailed in the following section.

2.3.Design Changes

Due to the problems encountered with the CPLD and MCU, a re-design was necessary. In order to eliminate these issues and maintain a low cost, a single chip solution was employed in place of the CPLD, SRAM and old MCU. This single chip solution was a more powerful MCU, capable

of meeting the design requirements of the old MCU, the SRAM and CPLD. Chosen was the ARM cortex-M4 based STM32F407 from STMicroelectronics. Besides having a built in DCMI interface, which allows direct communication with our image sensor, the STM32F407 also boasts an operating speed of 168 MHz, 512 KB of flash and 192 KB of SRAM [19]. One of the biggest benefits of the new chip was the increase in onboard SRAM. This meant that instead of loading parts of a complete image from an external SRAM and processing the parts separately, an entire image could be loaded and processed without the need for additional overhead. Note that as the CPLD has been eliminated from the design, the need for a series of GPIO lines has been negated, however, if the design were to call for such a need the new MCU has more GPIO lines than the previous chip. The STM32F407 also has several more I2C lines that the LPC11U14, again amply meeting the needs of the system. Lastly, the STM32F407 also possess USB on-the-go (OTG) with support for both USB host and device modes, which will not only be useful for debugging, but also for interfacing with computer software and potential future expansion of the system.

Electrically, this new chip runs at 2.8V, and as discussed in previous sections, new voltage converters were selected to meet this need, both of which meet the design constraints outlined in Table 4. Software considerations are made in the following sections, but as this chip is more powerful in every way when compared to the previous MCU (which met our requirements, with the exception of a lack of memory) it is safe to say that from a software perspective, this chip will suffice. An updated block diagram which reflects these changes can be found in Figure 5.

2.4.Assessment of Physical Feasibility

System feasibility consists of constraints placed on the design as a whole. These constraints are dictated by Group 50's experience with electrical systems, hobby work, and market segment. The constraints listed in Table 5, are met by our current design.

Table 5: Feasibility Consists

Constraint	Specifications
Size	Maximum 10cm diameter, 8cm height
Weight	Maximum of 2lbs
Operability	The system can operate indoors
Eye Safety	Extended exposure of the laser directly to the cornea should be avoided.

This sensor is not the only of its kind, it was in fact inspired by similar works done by Neato robotics [20]. Their XV-11 robotic vacuum employs a similar design for a cheap LIDAR sensor. Neato regrettably refuses to sell the sensor alone, which costs them (in their large volumes) significantly less than what many other LIDARs on the market frequently sell for, typically over $1000 (for example the $1200 Hokuyo URG-04LX-UG01) [21]. Neato's design employs a rotating platform with a laser and linear sensor. Since a linear sensor is simply a fancy image sensor, our camera approach is quite similar to theirs, and as such the Neato design supplies a convenient proof of concept for our design. In terms of practical use and constraints, much like Neato's LIDAR unit, our system will also be intended for indoor use, primarily due to saturation concerns from intense ambient light present in the outdoors (read: the sun). Due to the fact that Neato refuses to sell their comparatively cheap LIDAR unit, a large demand in the hobbyist community has been brewing ever since news of the cheap sensor first arose; brewing to the point that a hardware bounty was placed with a prize of $400 for the first to successfully hack the device [22]. Given the demand for Neato's sensor, and that Neato refuses to sell their device, a clear desire for our sensor is present.

2.5.Assessment of Software Feasibility

Since the hardware of the system has changed from the initial prototype design, prior analysis based on the CPLD and SRAM chip are no longer valid. All the processing of the system is now accomplished solely by the STM32F407. In terms of the data rate put out by the camera, thanks to the build in DCMI module on the STM32F407, interfacing with the camera is not of concern. The real bottlenecks arise in the RTOS and image processing itself, as well as the memory requirements of said RTOS and image processing.

2.5.1. Data Capacity

In terms of the data capacity of the STM32F407, internal memory comprises 192 KB of SRAM as well as 512 KB of FLASH for program memory. 512 KB of flash is more than sufficient for program memory; the primary concern lies with the 192 KB of SRAM. On this ram must sit any image data being processed, the RTOS and associated data as well as any intermittent data required by the image processing algorithm.

The images put out by the image sensor come at a maximum resolution of 2592x1262 pixels, however we are cutting this image down vertically to only 20 pixels, for a resolution of 1262x20. At 16 bits per pixel this equates to 103,680 bytes, or 102 KB. Of this we are only concerned with half of the image, giving a size of approximately 51 KB per image. With the capacity of 192 KB, even with 3 images stored in memory there would still be 39 KB of memory free, which from previous experimentation with our chosen RTOS, is more than enough memory for smooth operation.

In terms of processing speed, given that the camera at full operation will output images at 120 frames per second (FPS), or approximately 120 Hz, and that our MCU operates at 168 MHz (1.4 million times faster), processing power will not be an issue. The speed of our chip provides more than sufficient time to process several images in the time between images being produced by the camera.

3. Detailed Design

As a review, the device sweeps out an arc with a laser beam, that when read by the device's camera, produce a distance map of the surroundings of the device via triangulation. Thus, this device can be easily mounted on a robot in order to provide mapping or obstacle avoidance. The older block level diagram was discussed in section 2. This block level diagram is now updated, and shown below in Figure 5. [3]

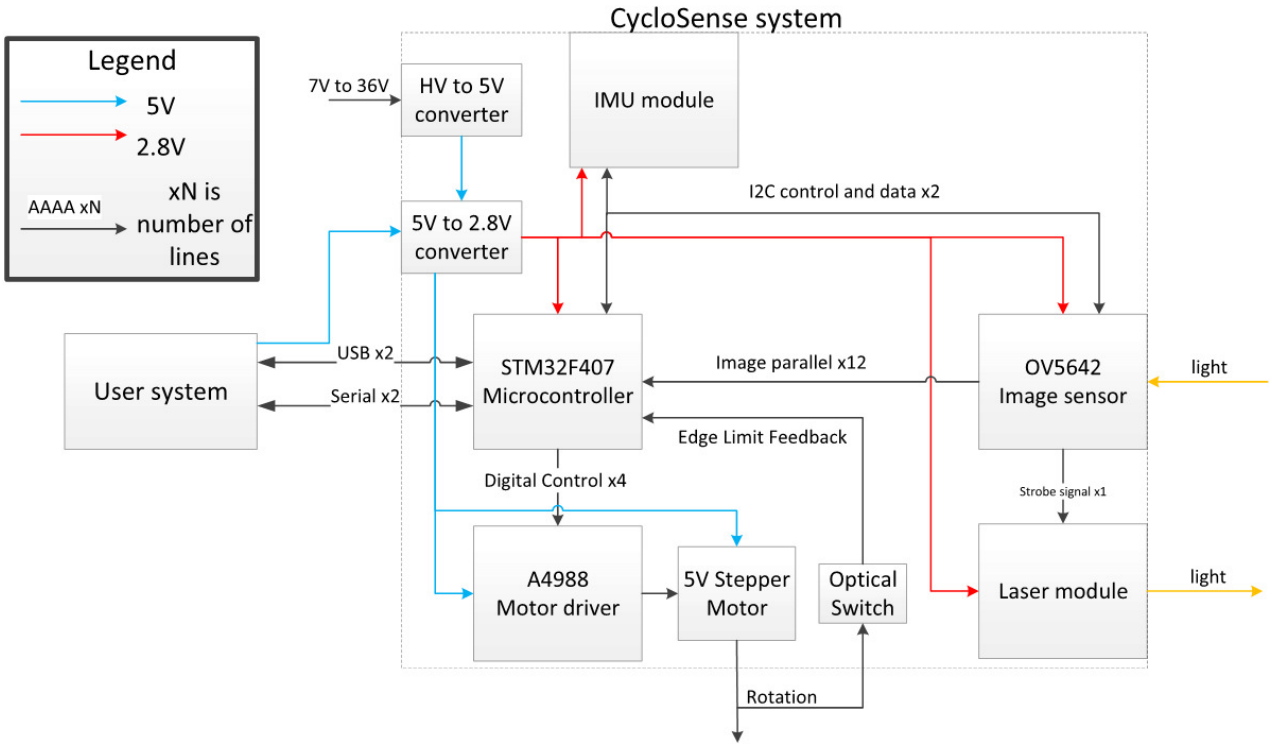

Figure 5: Detailed system architecture

Each of the sections presented above will be examined in detail below. The following sections will primarily focus on the technological and engineering decisions that went into each part selection and system design.

3.1.HV to 5V converter

The system must be useful to hobbyists. Thus, it must be able to accept a wide range of input voltages. For this purpose, the OKI-78SR-5/1.5-W36-C from Murata Power Solutions was chosen. It is an efficient, robust DC-DC converter which can provide our needed current. Also,

the converter is inexpensive. If Group 50 decides to commercialize the product in the future, a custom designed DC-DC converter will be employed to further reduce costs.

3.2. 5V to 2.8V converter

The 5V to 2.8V converter chosen for the project had to conform to the needs of a prototype. Thus, it had to be easy to use, robust and simple to integrate. With this in mind, the following linear regulator, the TLV70228DBVT was chosen. This converter provides sufficient current to allow all the systems in our design to function correctly. It is also easy to solder, and simple to design into the circuit. Lastly, the converter is robust, allowing us to make mistakes with it, such as temporary short circuits, without killing the converter.

3.3. Microcontroller (STM32F407)

The primary block or "brain" of the system is the microcontroller (MCU). The chosen MCU for our system is an STM Electronics STM32F407. This ARM Cortex-M4F MCU will control every other block in the system, either directly or through an interface (such as the motor controller for the motor). The STM32F407 contains many interfaces, including I2C, USB, and PWM to name a few. The MCU also contains a dedicated floating point processor, making it particularly suited to image processing tasks. Through use of these interfaces the STM32F407's primary role will be to collect and process data collected by the image sensor, collect and filter IMU data, and indirectly control the motor. This also carries the implication that the STM32F407 will handle the system's requisite image processing. The STM32F407 will handle the blob detection needed for detecting the laser dot, and thus, triangulation. The algorithm to be employed at this stage will essentially operate like a scanner; as the image data is fed to the STM32F407 from the image sensor, the STM32F407 will perform blob detection on a small slice of the image that will sweep across the image, much like a scanner.

3.4. A4988 Stepper Driver, Bipolar Stepper Motor, Optical Switch

The A4988 is the motor driver to control the stepper motor that turns the sensor.

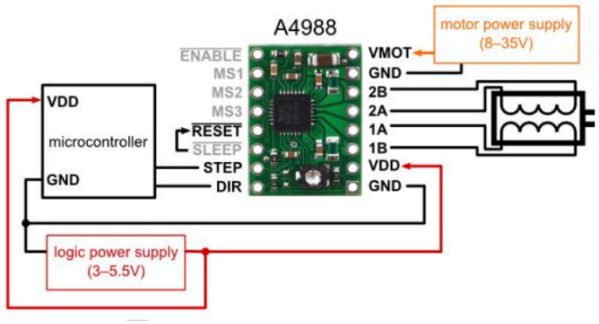

Figure 6: Motor driver schematic

The motor driver seen in Figure 12 takes in input such as voltage supply of 5V to run the stepper motor. It also takes in a 3.3V signal from the MCU to control its direction and motion instance. Each pulse to the "Step" pin of the motor driver causes the motor to increase by minimum of 1.8 degrees. The "DIR" input pin controls whether the motor turns clock wise or anti-clockwise. Our motor is bipolar and can turn in both directions. The driver is a very simple off-the-shelf chip with a few inputs and 4 outputs to the motor. As seen in Figure 12, the output pins to the motor are labeled 2B, 2A, 1A, 1B. There are a minimum of 2 inputs from the microcontroller needed, which is direction and when to cause the next step. The driver can supply ±2A and 35V, which more than enough for the bipolar, single shaft stepper motor in our design, seen in Figure 13.

Another important aspect of motor selection is safety. The stepper motor was specifically chosen to be strong enough to turn the measurement unit, but not strong enough to present a pinch hazard. If a user were to catch a finger in the mechanism, the motor will cause very minor pinching at worst. This improves the safety of our system. [23]

3.5. Image Sensor (OV5642)

In the system block verification document, the image sensor chosen was the OV5642. This sensor is the optimal choice for our design for several reasons. First, the image sensor is inexpensive and easy to obtain. It also provides data in an easy to use format, at high speeds. Compared to any other camera sensor we were able to find, this image sensor provided the highest pixel count at the lowest price. Pixel count is important, as this is a determining factor in the precision our system can achieve.

However, while this component is a good fit for our needs, it is relatively difficult to solder by hand. Also, the pitch on the component is small, making the resulting PCB expensive. The image sensor also does not contain any focusing optics, necessitating an optical mount and lens assembly. As a result of these difficulties, group 50 contacted a manufacturer in China that produces small camera modules. This manufacturer was able to provide us with a small, integrated optical unit.

This unit is ideal for our needs, and no comparable solution was found anywhere else. This module, the JAL-OV5642 V2.0, is shown below.

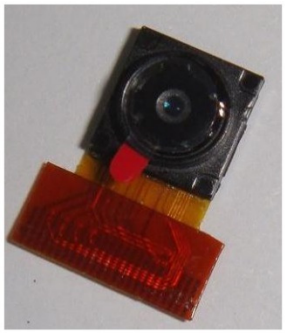

Figure 7: JAL-OV5642 V2.0 camera module for the OV5462

This module obviates the need for group 50 to design a lens package, in addition to providing an easy to use connector.

3.6. Laser Module

The laser module consists of two main parts. First, there is a small, tubular laser module. Specifically, Group 50 uses the VLM-650-03-LPA due to easy of purchase from Digikey. However, any laser module may be used, so long as it can be driven with either 5V or 2.8V. Also, the laser module must have a power output rating of 5mW or lower. This ensures that our laser is classified as Class IIIa laser [24]. These lasers are safe to use when not combined with optical amplification, and when exposure time is suitably limited. Our chosen laser emits 2.5mW, falling well within the eye safe range. Secondly, there is a small transistor used to drive the laser module on and off. This transistor is biased to ground, thus disabling the laser whenever an activation signal is not expressly sent. This improves the safety of the product by reducing the likelihood of accidental laser illumination. The most important safety feature of

the laser module is how the module is controlled. The laser is only illuminated when the strobe signal from the camera module is high. This signal goes high at the start of an image, and is used to control flash on regular cameras. Thus, the laser module will be powered for approximately 0.1 second out of every second. This further ensures that total power emitted by the laser module is kept within safe limits.

3.7. Inertial Measurement Unit (IMU)

The IMU consist of three components: 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer. The combination of data from these three components with the distance measurement from the laser/image sensor can produce a map of the area. After some research the following requirement was determined needed for the IMU components:

- Uses I2C as a form of communication
- Fast data refresh
- 3-axis (for potential 3D mapping)
- Operates on 5V or 3.3V
- Small and light
- Cost effective

The STMicroelectronics LSM303DLHCTR was selected as the 3-axis accelerometer and magnetometer. This chip include 2 of the 3 components for the IMU, and therefore reducing size and weight while it is still large enough to solder by hand. It also uses I2C and runs on 2.16V to 3.6V, so it complies with the 3.3V requirement. The LSM303DLHC can run at a rate of 1, 10, 25, 50, 100, 200, and 400Hz providing a variety of options for data refresh.

The STMicroelectronics L3G4200DTR was chosen as the 3-axis gyroscope, and it is only of the few chips found that has 3-axis and is still solder able by hand. It also uses I2C, and operates on 2.4V to3.6V. The L3G4200D can run at a rate of 100, 200, 400 or 800Hz, providing a variety of options for data refresh. It is also small and light.

These two chips are made by STMicroelectronics, which is a reputable manufacturer, and it will minimize the likely hood of incompatibility. The figure below shows the schematics of the IMU

chips, including filters and coupling capacitors. The IMU will operate based on 3.3V power input, and will output data through the I2C SCL and SDA pins.

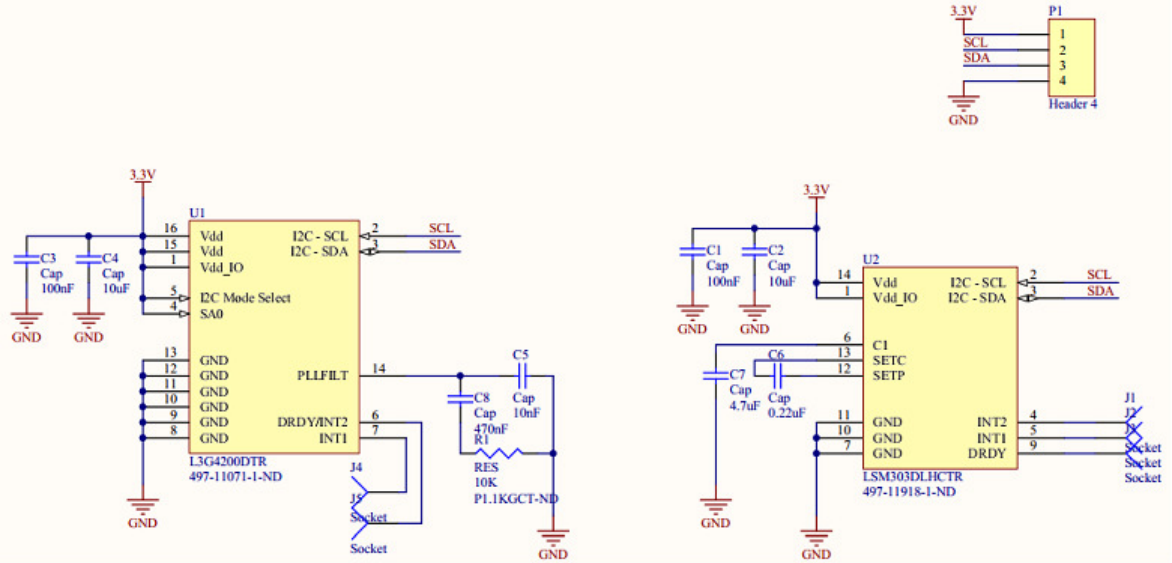

Figure 8: IMU Schematic

3.8.Image Processing

The MCU will run three tasks to collect the data from the camera and the MCU and output a distance measurement. Figure 9 shows the flow of data between these tasks.

The Camera-Motor, Image Processing, and Data Aggregate task are parallel processes in the MCU, that are responsible for converting pixel data from the SRAM and the orientation data from the IMU into a serial data stream to an arbitrary target platform.

The Image Processing block must employ a distance detection algorithm to identity the distance from the sensor to the object reflecting the laser. To provide a better understanding of the physical phenomenon that allows us to infer distance information using reflected laser light, refer to Appendix B. To summarize the conclusions of Appendix B; where the laser dot falls on the sensor correlates to how far the object is from the camera. At infinite object-to-camera distances, the laser will land near the vertical center of the image. As the object approaches

closer, the laser dot will move horizontally towards the edge of the image. The closest distance the camera can detect before the laser dot falls off the image sensor is 15cm.

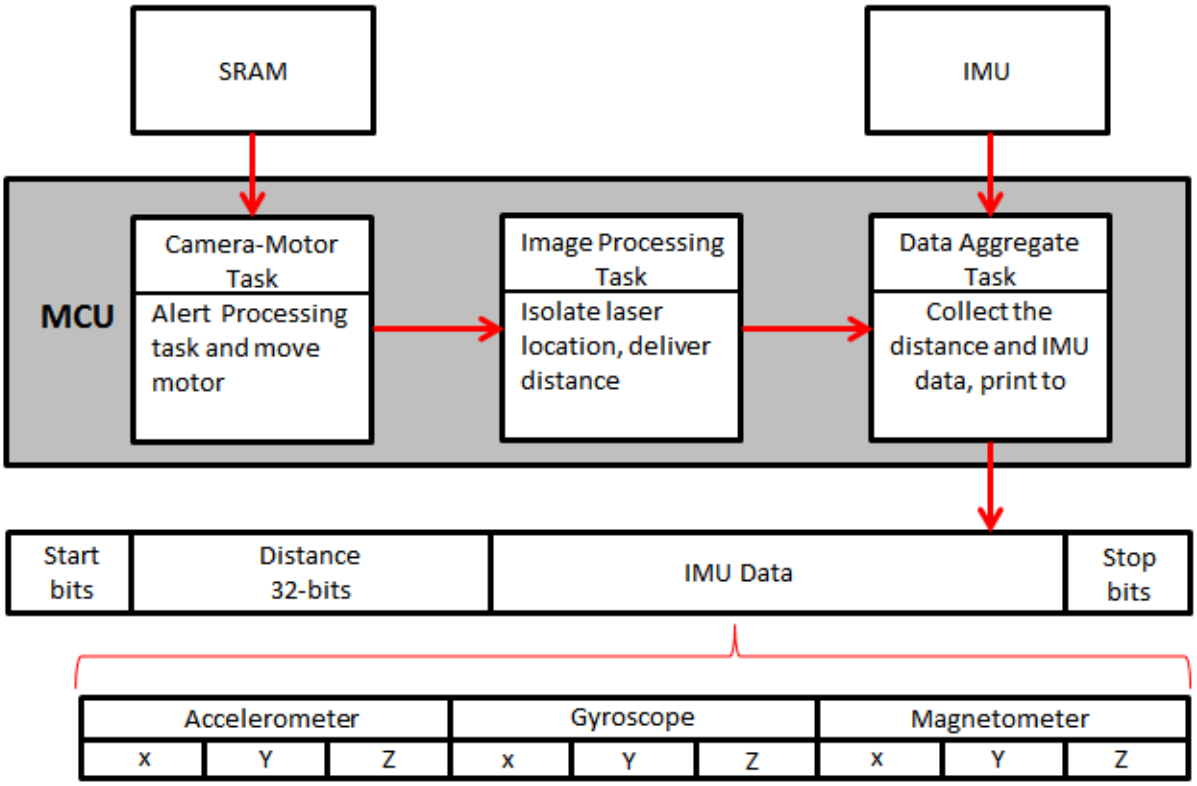

Figure 9: The Image Processing Task

What remains for the distance detection algorithm is to decide how far the laser dot has landed on the image sensor. The first step will filter the image in such a manner that most pixels that do not originate from the laser will be set to zero. The second step is to analyze the remaining pixels and find the corresponding distance measurement. The following sections describe the different approaches to construct each step in the algorithm, along with a design decision to which approach will be accepted.

3.8.1. Image Filtering & Blob Detection

As stated in the above section, the image must be filtered to isolate the laser light more effectively. The following sections will discuss several approaches, developed entirely by members of Group 50. The laser will produce bright red pixels. In the RGB565 colour space, the pixels corresponding to the laser light will have their red colour channel saturated, with

medium or low green and blue colour channel values. At the center of the laser blob, all the colour channels will be saturated to white. These pixels will also be in close proximity to each other, within a small area. Using this information, we have the following filter algorithms.

3.8.2. Threshold Filtering

This filter first discards all pixels on the left half of the image, and those not within 5 pixel widths from the vertical center. This is the area of interest where the laser light is expected to be since it moves horizontally along the center on one half of the image. Only red pixel information is used. As the algorithm iterates over the pixels of interest, it stores of running average of all pixels that that exceed a given brightness threshold. The center of the laser dot should be the average of these *bright*, red, and cropped pixels. The output of the algorithm is shown in Figure 10 on the right, where the blue line crosses the center of the laser dot.

Figure 10: Threshold Algorithm Output

Advantages

Speed, as it only requires one iteration.

Disadvantages

Requires preliminary knowledge of the correct threshold. It can also be fooled by bright white spots such as light reflecting off white surfaces. White colours also contain high red values.

3.8.3. Running Average Threshold Filtering

This filter assumes a brightness threshold close to zero. It also crops the image to pixels of interests, retaining only red pixel data. As the pixel data is fed to the algorithm, the brightness threshold is assigned to the third brightest pixel found, or once a certain percentage of total

pixels have been reached. Once all the pixel information has been sent, the algorithm reiterates over the image with this updated threshold and collects the average location of pixels that exceed the threshold. The center of the laser dot should be the average of these *relatively* bright, red, and cropped pixels. The output of the algorithm is shown in Figure 10 on the right, where the blue line crosses the center of the laser dot.

Advantages

The brightness threshold does not need to be provided beforehand, the brightest pixels in the images will not be mistakenly excluded even if they are relatively dim.

Disadvantages

Slower processing speed, as it may take multiple iterations over the image to isolate only the top percentage of bright pixels. At least two iterations are required in the best case. It can also be fooled by bright white spots such as light reflecting off white surfaces.

3.8.4. Black Hole Filtering

This filter does not discard the green and blue RGB values of the image. Instead, it re-assigns each pixel to a single value based on how bright the red colour channel is compared to the green and blue colour channels. By doing this, bright white areas are filtered out. Due to the brightness of the algorithm, the center of the laser is usually observed as white instead of red. Since white pixels are floored, the center of the laser will appear as a black hole, surrounded by a red corona, hence the name of the algorithm. This algorithm was developed independently by members of Group 50. Once all pixels have a re-assigned value, a threshold containing the 30th brightest pixel value (guaranteeing there are at least 30 pixels not floored in the image) will be used to find the center of the laser dot much like the Threshold Filtering Algorithm. The center of the laser dot should be the average of these relatively bright, *purely* red, and cropped pixels. The output of the algorithm is shown in Figure 11 on the right, where the blue line crosses the center of the laser dot.

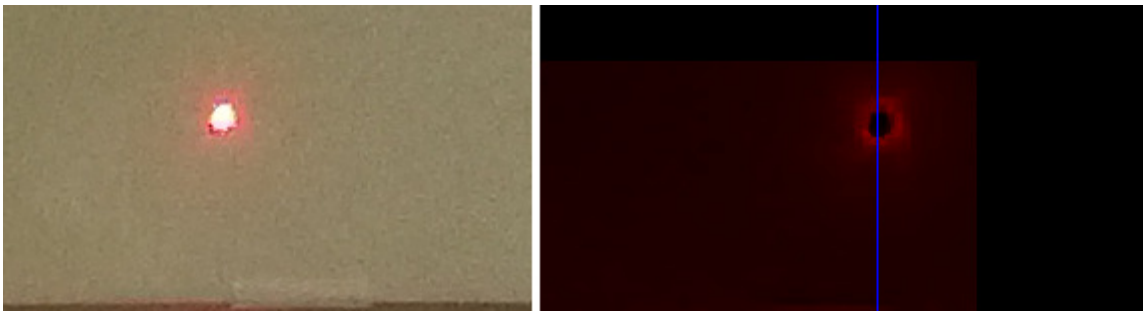

Figure 11: Black Hole Algorithm Output

Advantages

The filter will not be misled by bright white areas of the cropped image.

Disadvantages

Slower processing speed, as it may take multiple iterations over the image to isolate only the top percentage of bright re-assigned pixel values. At least two iterations are required in the best case. Areas that are not brightly red, but more relatively red than they are green and blue will appear to have higher pixel values after processing.

3.8.5. White Hole Filtering

The White Hole filters builds off an image artifact found in the Black Hole Filtering algorithm. The laser dot will always be a white blob surrounded by bright red pixels. Therefore, this algorithm follows the same process as the Black Hole Filter, except that white pixels that are directly beside bright red pixels are not eliminated. The white pixels must also be in close proximity to one another, and exceed a minimum total area. All other pixels are floored. This should preserve the majority if white pixels in the center of the laser dot, and discard dull red pixels that still have low green and blue values. The center widths of the remaining pixels are found similar to the Threshold Filtering Algorithm. The center of the laser dot should be the average of these white pixels, in *close proximity* belonging to the laser dot. The output of the algorithm is shown in Figure 12 on the right, where the blue line crosses the center of the laser dot.

Figure 12: White Hole Algorithm Output

Advantages

The filter will not be misled by bright white areas of the cropped image, or dull red pixels with low green and blue values, scattered noisy pixels, and should home into the true center of the laser dot.

Disadvantages

Slower processing speed, as it may take multiple iterations over the image to isolate only the top percentage of bright re-assigned pixel values. At least three iterations are required in the best case.

3.8.6. Evaluating Filter Algorithms

The criteria the filter algorithms should aspire to are the following:

1. Speed of processing in terms of the average amount of iterations each algorithm needs.
2. Memory requirements in terms of how much of the image must be stored.
3. Accuracy in locating the laser dot in the midst of noise.

To compare the algorithms fairly, we will use a computational decision making matrix, as shown in Table 6. Higher values have higher merit. Therefore we grade criteria using rank. The algorithms that most successfully fulfill their criteria get a rank between 4 and 1, where 4 is best.

Table 6: Computational Decision Making Matrix

	Criteria	Speed	Memory	Accuracy	Total
	Weight	5%	20%	75%	100%
Threshold Filtering	Rank	4	4	1	
	Relative Rank	1	1	0.25	
	Score	5	20	18.75	**43.75**
Running Average	Rank	3	3	2	
	Relative Rank	0.75	0.75	0.5	
	Score	3.75	15	37.5	**56.25**
Black Hole	Rank	2	2	3	
	Relative Rank	0.5	0.5	0.75	
	Score	2.5	10	56.25	**68.75**
White Hole	Rank	1	1	4	
	Relative Rank	0.25	0.25	1	
	Score	1.25	5	75	**81.25**

The speed of the algorithm is not an issue, as the MCU can conduct more than a 100 million instructions per second, and each algorithm is not exceptionally intensive. The memory is also not a concern, since there is enough flash to store 2 cropped images, and each algorithm doesn't use much more memory than only one cropped image. The real importance is the accuracy of the algorithm. Since the location of the laser dot changes slightly at greater distances away from the camera, the accuracy can have a big effect on the correct distance measurement. Because this criteria has much more importance, the White Hole Filter algorithm is chosen to be implemented in the Image Processing Block.

4. Experimental Results

Verification of the components of our prototype requires detailed experimentation and analysis of the results. We examine each component by using our prototype checklist, and then discuss the major successful techniques and flaws we encountered during our design in both the design and process analysis. [25] Some of these components may have been updated/changed since the checklist. However, they still fulfill the original requirements, and are thus considered interchangeable. Hence, they are not discussed in detail here; they are discussed in Detailed Design section 3.

4.1. Prototype Checklist

For us to meaningfully verify our design for future design iterations of our prototype, it was necessary to create a grading scheme for all testable components. This rubric was presented as a checklist to our project supervisor for his feedback during our first prototype presentation. Below is a summary of the criteria for each subsystem and observed performance during testing. [26]

4.2.Over all system

The overall system did not fail on power up, had an overall weight of less than 4 pounds, and was roughly $400 as expected. The distance ranging capability and precision of the LIDAR was not successful due to a failure to form an image due to a test failure in the CPLD which will be covered in the CPLD subsystem test results. The overall system is presented below in Figure 13.

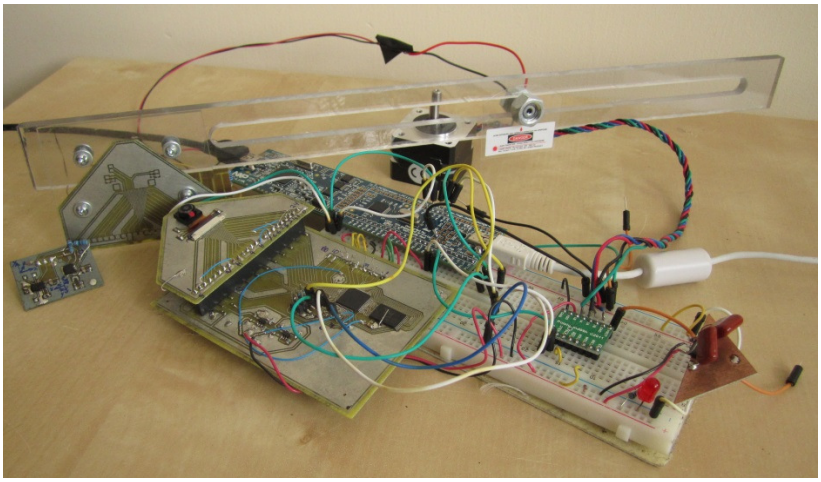

Figure 13: Prototype V1.0 assembled system

4.3.SRAM

The SRAM was able to store and retrieve 8 bits of data at a time. The SRAM thus fulfilled all of its required functions.

4.4.Microcontroller

The microcontroller was unable to run a simple LED blink task, due to a lack of stack space to perform the task operation. The original 4 kilobyte estimate of the memory was half the actual memory available for programs, since 2 kilobytes was reserved. The USB interface was successfully constructed, such that the prototype could be recognized by a PC running windows 7. The MCU was also able to issue commands to the CPLD to store and retrieve camera data, even when an entire image was stored in SRAM.

4.5.Motor & Motor Driver

The motor driver was able to control the stepper driver to precisely move in both directions by a resolution of 5 degrees.

4.6.CPLD

The CPLD recognized commands from the MCU and deliver binary data to the MCU. Reading and writing to and from the SRAM was also established. The interface logic within the CPLD for camera was able to discard green and blue RGB data and subsample areas of interest from the incoming camera data. The logic timing in the CPLD was unable to capture all the horizontal synchronization signals from the camera and hence recognize the edges of the 2D image data. This is the reason why the overall range finding requirement was not satisfied.

4.7.Camera

The camera powered the output logic lines to 1.5 volts. It replied to I2C requests. The camera self-cropped images to the appropriate size. A strobe signal from the camera triggered the laser module at the correct instances.

4.8.IMU

The accelerometers, gyroscope, magnetometer were detected by the microcontroller's I2C response. Each of these sub components was able to receive data that reflected the orientation of the MCU PCB. The design and final version of the IMU is shown in Figure 14.

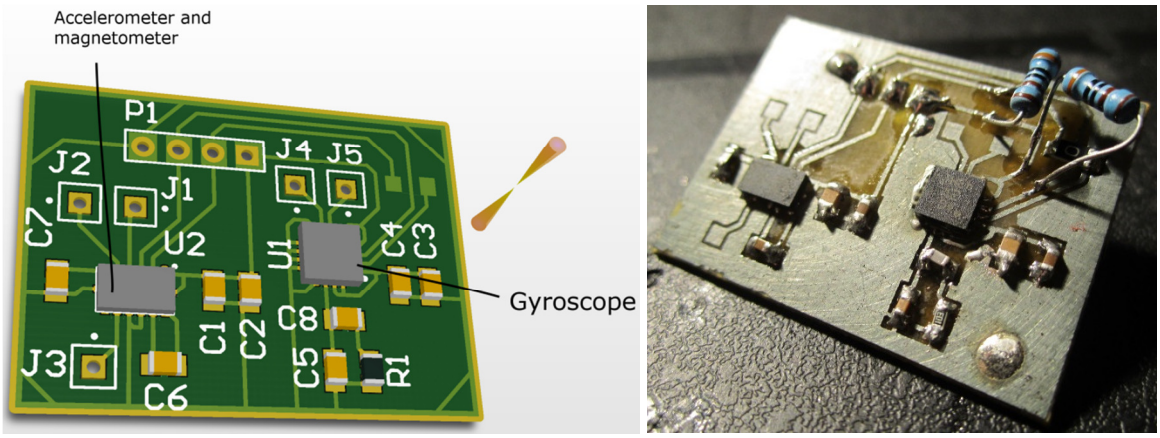

Figure 14: IMU design and complete prototype

4.9.Design Analysis

Throughout the design and assembly of the prototype, several key strengths and weaknesses of the design became quite apparent. First and foremost, our hardware design is quite excellent. We did not encounter a single problem with the hardware at any stage. Another key strength which was invaluable throughout the construction of the prototype was the ease of troubleshooting. The inherent modularity in the design, and the inclusion of dedicated test points made debugging many of the issues that arose quite trivial. Our selection of parts not only provided sufficient throughput for processing the image data necessary for triangulation, but did so while maintaining our budget requirements.

Along with these strengths several weaknesses in our design also presented themselves. The biggest and most concerning weakness was in our CPLD. We ran into issues with horizontal synchronization (hsync) of image data. Figure 15 below demonstrates a sample output of our camera; note that we are filtering out all but red image data.

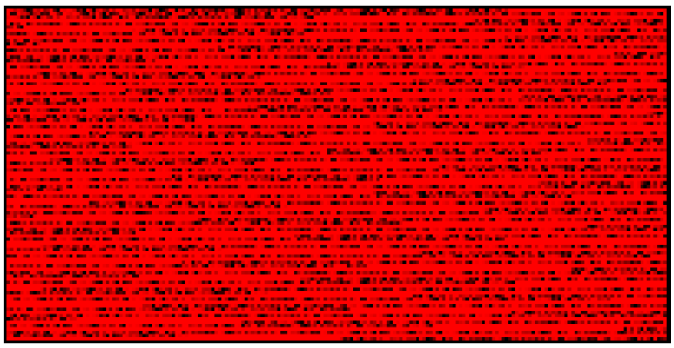

Figure 15: Corrupted CPLD data

Due to a problem with the hsync line junk data was being inserted into the image, as evident by Figure 15. A solution to the problem was found yet this solution could not be implemented as our chosen CPLD lacked sufficient macro-cells. Therein lied the weakness of our design; we chose a relatively small part with very few macro-cells, in order to save board space and reduce cost. For a prototype though, it would have been better to use a larger part (with additional macro-cells) and if necessary, transition the design to a smaller and cheaper part later.

Another weakness in the design was in the microcontroller (MCU) memory. Specifically, our decision to use FreeRTOS [3] should have been researched more. We ran into issues with insufficient memory when trying to run FreeRTOS with the USB device stack necessary for our system. Finally, the last weakness in our design was the USB device stack itself. Sufficient research into the stack itself was not conducted, and issues arose with some of our general purpose input output (GPIO) pins on the MCU while we were trying to simultaneously read data from our SRAM and transmit it out through USB.

4.10. Process Analysis

Our design process fit in accordance to the accepted engineering design flow of specification, design, modeling, assembly, and testing. Team members were assigned to work with the modules for which they were technically suited for. We also took the opportunity to learn how other components expected our modules to behave, so that we could create the appropriate interfaces between modules.

One of the design goals that took longer than expected was the programming of the CPLD in VHDL. To do this we used the XILINX ISE Webpack, which worked well for the first few

iterations of the code, but then produced unexpected behavior in the simulation and synthesis that were not repeatable if the program restarted. This caused uncertainty for the programmers of the CPLD on whether the faults of the simulation were due to the code or the Integrated Development Environment (IDE) itself. For example, previously working code when recompiled and programmed onto the CPLD caused high current draws and heat dissipation on the chip. This problem caused a delay of 2 days in our design, which was only resolved after a re-install of the IDE. During the re-design, certain group members had to subsist on rice and water with very little sleep in order to meet project deadlines.

There were two other weaknesses we did not expect. The most major was how we distributed time between the documentation and the actual building of our prototype. We had scheduled periodic meetings every week to work on the project, but instead they were used to meet documentation deadlines. In the end we only had one week available to integrate the entire system. The other weakness was the linear method of our design. Most of the components could not be tested if other components had not been previously built. For example, The SRAM could not be tested in its speed of storage and recovery of data if the CPLD interfacing with it didn't have working code. The triangulation could not be verified if the CPLD, SRAM, Camera and MCU communication ability wasn't ready.

5. Discussion and Conclusions

Group 50 has endeavored to create a novel, camera based laser range finding system for use in robotic systems. In this task, we have been reasonably successful. The team working on this project has coordinated and executed required tasks on time, with good efficiency. Group 50 would like to take this time to acknowledge the hard work and dedication that each team member has shown in the completion of this project.

Current limitations of the design include the inability of the system to take video for the end user, and also group 50 has not yet dedicated time to developing a robust mapping algorithm. In the first case, due to the fact that the system uses a cell phone type camera, it is conceivable that high quality still images or video could be produced for the end user. This would mean that the end user would have access to vision data for their own processing. It is evident that allowing the end user to have such a capability would increase the worth of the device. However, the main limitation preventing video are two-fold. First, the system requires large amounts of external SRAM to be added to the system to buffer the video stream. This would require the sourcing, design, and control of an SRAM chip. Secondly, video would require a high throughput, high reliability connection to the host computer, such as USB. While our design already implements a USB connection, it currently emulates a serial port. For video streaming, the USB protocol would have to be re-written, a non-trivial task. The second limitation of the system is a robust mapping system. Currently, the sensor feeds back a stream of positional data. This is good for making maps of the current surroundings of the sensor. However, in an ideal case, Group 50 would develop mapping software as a demonstration for the host computer. Thus, end users could quickly download our demonstration software, and create maps of their immediate environment.

Competitors in a similar market space for this product are Hokuyo LIDAR products, and the Parallax Laser Range Finder. The first competitor basically summarizes the entire reason Group 50 took this project. Hokuyo LIDAR are widely regarded as the cheapest, smallest, and most user friendly LIDAR systems on the market today. However, these metrics are only true when compared with real LIDAR units. The Hokuyo costs from $2000 - $5000, and is larger and

heavier than our design. Our design was intended to be cheaper, and more easily integrated into small robotics systems, where weight and size are important to the overall design. The second system, the Parallax Laser Range Finder, is similar in concept to our system; using a camera for data gathering and a laser for range finding. However, the system has much lower resolution, is slower, and costs relatively the same as our design. In addition, both competitors do not incorporate an IMU, and are not open source. Both these factors make Group 50 confident that our design will be well received in the hobbyist and education marketplaces.

Novel features of this product compared to similar products are as mentioned above; we incorporate an IMU, we are lower cost than other competitors, and plan to open source all of our designs. There has historically been many IMU's developed for robotics uses, and a few laser rangefinders as well. However, there has been no product that combines these two devices for ease of use for the end user. In addition to this combination, the low cost of the device should prove particularly attractive to end users.

Additional features that may be developed in the short term are the mapping algorithm, and a better, fuzzy-based blob detection algorithm. The mapping algorithm, as discussed above, would constitute of a SLAM based approach to localization. If demonstration software could be written on a computer, end users could then easily incorporate this work into their own designs. This would increase general interest in the project, and provide further incentive for users to try out the system. The fuzzy-based blob detection would allow for higher precision in the detection of the centroid of the laser blob on the image. This would translate to higher accuracy in laser range finding measurement. This is a possibility that Group 50 is studying in order to tie in our studies with ECE 457B. In addition, fuzzy-based blob detection has a possibility of being far more robust than most other blob detection algorithms. This would reduce the amount of erroneous data gathered by the system.

Long term goals for this project are improvement of the accuracy of the data, auto calibration of the range finding, better physical packaging, commercialization and support. Improvements to data accuracy will be made with better algorithms, and higher quality lens modules on the camera itself. This would allow for more accurate laser blob localization, which leads to better

distance measurement. Thus, the system would be both more reliable and able to detect further objects. Secondly, calibration of the system has been accomplished by hand so far. This is a painstaking process involving one of Group 50's members fiddling with tape, textbooks and sticky notes, as seen in Figure 16.

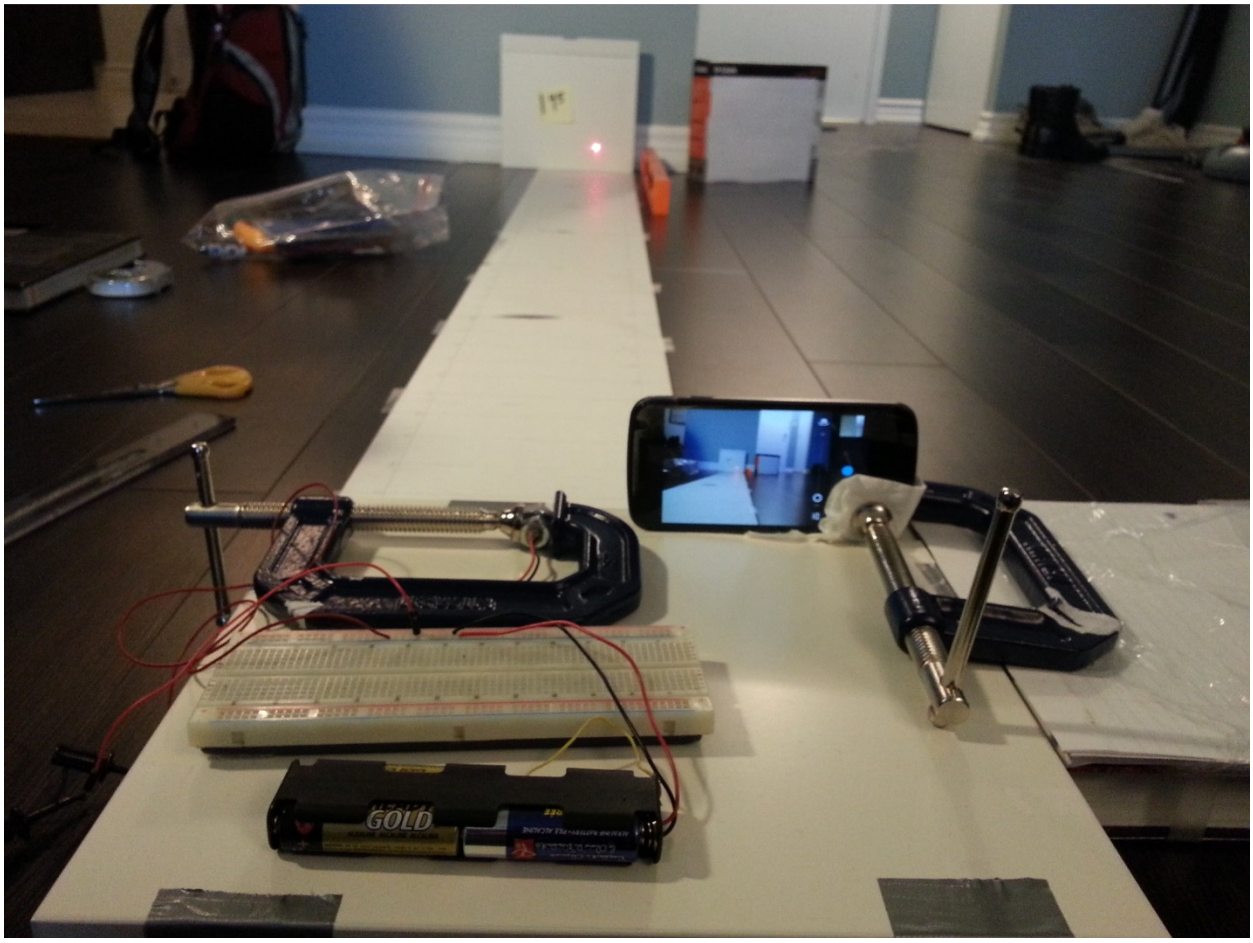

Figure 16: Tape, Textbooks and dreams

Thus, Group 50 plans to dedicate time to developing an automated way to calibrate each sensor. This would include some computer controlled movement and decision making. In addition, the physical packaging of the project must be improved and made more durable. Currently, the 3D printed casing is too expensive for mass production, and slightly too fragile. Thus, improving the case would allow Group 50 to go ahead with commercialization of the product as well. Lastly, providing technical support in the form of a community of users, with tutorials and quick start guides, would enable more people to use the system.

Environmentally, this product as a whole is quite acceptable. The product does not create air pollution. The device in itself does not contain any power source, so no recycling concerns to mention. The production quality version of the device will use lead free components, hence there is no hazard in disposal of the product. Materials used in the construction are plastic, copper, and misc. electronic components.

From a societal perspective, the device strives to increase the availability of sophisticated sensors to markets that are currently ill served by existing solutions. By decreasing the cost of device, and making it easy to use, we encourage the next generation of students to become involved in engineering and design. This device will allow many people to begin experimenting with advanced robotics concepts, such as localization and mapping, that they are unable to currently do due to lack of funding. Group 50 also hopes that the low cost and accessibility of the device will pressure other manufacturers into providing more accessible solutions to the general public.

Our device does not have evident cultural implications at the moment. However, upon mass production, Group 50 would ensure that the device was available across the world, in as many languages as possible.

References

[1] I. Baranov, S. Kazi, N. Hilal, J. Godson, "Abstract," 2012.

[2] I. Baranov, S. Kazi, N. Hilal, J. Godson, FYDP Project Specification, University of Waterloo, 2012.

[3] I. Baranov, S. Kazi, N. Hilal, J. Godson, "Detailed Design," 2012.

[4] I. Baranov, S. Kazi, N. Hilal, J. Godson, "Block Verification," 2012.

[5] DigiKey, "OKI-78SR-5/1.5-W36-C," [Online]. Available: http://www.digikey.ca/product-search/en?x=27&y=16&lang=en&site=ca&KeyWords=OKI-78SR-5%2F1.5-W36-C. [Accessed Jan 2012].

[6] DigiKey, "TLV70228DBVT," [Online]. Available: http://www.digikey.ca/product-detail/en/TLV70228DBVT/296-27891-1-ND/2425105. [Accessed Jan 2013].

[7] RobotShop, "Bipolar Stepper Motor Specifications," [Online]. Available: www.roboshop.com. [Accessed May 2012].

[8] Allegro Mircosystems, DMOS Microstepping Driver with Translator and Overdrive Protection, Allegro Microsystems, 2012.

[9] Omnivision, OV5642 Image Sensor, 2012.

[10] Digikey, "13 04 2011," [Online]. Available: http://www.digikey.ca/product-detail/en/IS61LV256AL-10TLI/706-1037-ND/1555410. [Accessed May 2012].

[11] ECIA, "LPC11U14FHN33 Priccing and Availability from Authorizeed Distributers," [Online]. Available: http://www.eciaauthorized.com/search?pn=LPC11U14FHN33. [Accessed 2012 May].

[12] DigiKey, "L3G4200DTR STMicroelectronics | 497-11071-1-ND | DigiKey," [Online]. Available: http://www.digikey.ca/product-detail/en/L3G4200DTR/497-11071-1-ND/2587903. [Accessed May 2012].

[13] STM Microelectronics, "L3G4200D," [Online]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00265057.pdf. [Accessed May 2012].

[14] STMicroelectronics, "LSM303DLHC," [Online]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/DM00027543.pdf. [Accessed May 2012].

[15] NXP Semiconductors, "LPC11U1x," Jan 2012. [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC11U1X.pdf. [Accessed May 2012].

[16] DigiKey, "LPC11U14FHN33/201," [Online]. Available: http://www.digikey.com/product-detail/en/LPC11U14FHN33%2F201,/568-8581-ND/2791886. [Accessed May 2012].

[17] Xilinx, "XC9572XL High Performance CPLD," [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds057.pdf. [Accessed May 2012].

[18] DigiKey, "XC9572XL-10VQG44C Xilinx Inc | 122-1448-ND | DigiKey," [Online]. Available: http://www.digikey.com/product-detail/en/XC9572XL-10VQG44C/122-1448-ND/966629. [Accessed May 2012].

[19] STMicroelectronics, "STM32F405xx, STMF407xx," [Online]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/DM00037051.pdf. [Accessed Jan 2012].

[20] K. e. al., A Low-Cost Laser Distance Sensor, Pasdena, CA USA: IEEE, 2008.

[21] Robotshop.com, ""Hokuyo urG-04LX-UG01 Scannig Laser Rangefinder"," [Online]. Available: http://www.robotshop.com/ca/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder-3.html. [Accessed May 2012].

[22] J. Munns, "Newest Hardware Bounty," [Online]. Available: http://hackaday.com/2010/11/17/newest-hardware-bounty-the-open-lidar-project/. [Accessed May 2012].

[23] OSHA.gov, "CalculateSafetyDistances.doc," stability tech, Atlanta, 2007.

[24] OSHA.gov, "OSHA Technical Manual SECTION III: CHAPTER 6 Laser Hazards," [Online]. Available: http://www.osha.gov/dts/osta/otm/otm_iii/otm_iii_6.html. [Accessed 12 02 2013].

[25] I. Baranov, S. Kazi, N. Hilal, J. Godson, "Experience Report," 2012.

[26] I. Baranov, S. Kazi, N. Hilal, J. Godson, "Check List," 2012.

Appendix A: Budget

Note: Distributor refers to an online component distributor, ex: Digikey, Arrow, Mouser, etc

Note2: We assume that SPF will cover 3D printed materials, as they are similar in concept to Printed Circuit Boards, i.e: custom designed parts.

Note3: Costs below include shipping, duties, taxes, etc. They are estimates, and more precise values will be available when the item is bought.

Table A-1: Projected budget

Item	Description	Cost	Funding	Vendor
Laser Modules	Multiple Test Laser Modules	$40	Team	Var. Distributor
Camera Modules	Multiple Test Camera Modules	$100	Team	Var. Distributor
PCB	Vars. Printed Circuit Boards	$200	Student Project Fund	Golden Phoenix PCB
Electronic Components	Vars. ICs and chips	$100	Team	Digikey Mouser
3D printed casing	3D printed plastic Casing, Multiple Versions	$200	Student Project Fund	Ponoko
Microprocessor Development Kit	Programming and debugging interface for a microcontroller	$100	Team	Var. Distributor
Misc Mechanical	Fastners, screws, motors, gears	$50	Team	Var. Distributor
	Student Project Fund Total	$400		
	Total	$790		

The actual costs incurred during the project are as follows, for two prototypes:

Table A-2: Actual budget

Item	Cost	Funding	Vendor
Laser Module	$20	Team	Digikey
Camera Module	$25.34	Team	Kai-Liang tech, China
PCB	$136.89	Student Project Fund	Alberta Printed Circuits

Elec. Components	$147.82	Team	Digikey
3D printed casing	$54.64	Student Project Fund	E5 Student Design Center
Microprocessor Development Kit	$32	Team	Digikey
Misc Mechanical	$0	Team	N/A
Total	$416.69		

As can be seen, we remained relatively true to the budget estimate provided prior to project commencement.

Appendix B: Distance Measuring Image Algorithm

This Appendix discusses some of the algorithms developed and used by Group50.

B1. Introduction

The goal of the image algorithm described in this document the process of extracting distance measurements from reflected laser light. Distances are calculated by triangulation using a laser, image sensor and lens. The basic idea is reviewed in section B1.1 Basic Triangulation. In Section 2 this idea is then applied to the current camera and laser arrangement our group has available. From this arrangement, a detailed triangulation scheme will be explained in Section B2.1, along with the distances that are known and those that need to calculate. Section 2 will lead to the conclusion that it is necessary to first calibrate the unknown distance from the camera sensor to the camera lens in order to calculate the distance from the camera lens to the object in the image. Section 3 will then cover the entire discussion on finding the sensor to lens distance, along with an explanation of the procedure of calibration and results. In Section 4, the results from Section 3 are used to interpolate the image data to calculate the lens to object distance.

B1.1 Basic Triangulation

To review, the distance from an object to another can be described by using one side of a right angle triangle. If the a side length and an angle is known inside a right angled triangle, then the other side lengths can be calculated. For our purposes, the corners of the right angle triangle we will use are formed by a laser, a camera and an object.

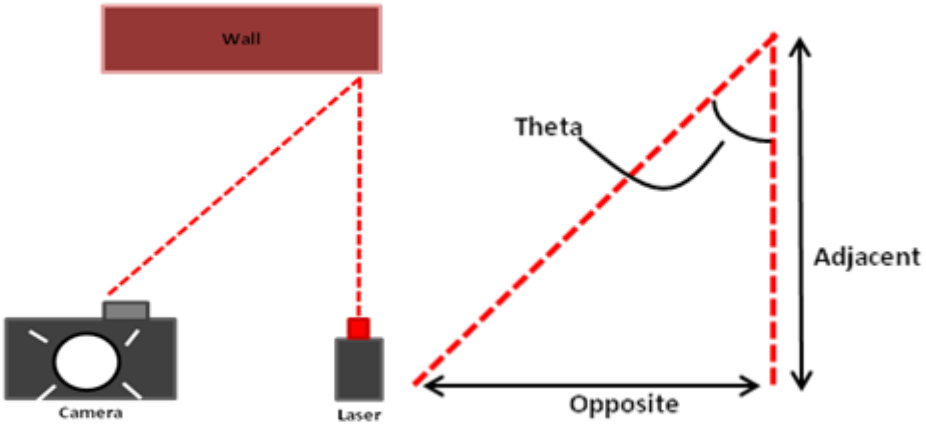

Figure B1: The basic distance triangle

The right angled triangle formed between these three objects is illustrated in Figure B11. Here the distance that would be calculated is the adjacent side (with respect to angle Theta) between the laser and the wall, which is also the direct distance from the camera to the wall. The most easiest distance to predetermine is the opposite side, which is the distance between the camera and the laser, or to be more precise, the laser beam and the centre of the camera lens. Using the arctan function, we could calculate the distance of the adjacent side if we knew both the opposite side length and the angle theta made between the laser beam and it's reflected component. Making this calculation along with gathering all the necessary information is the goal of all the experiments to follow.

B2. More Specific Top-down Level View

From the concepts of basic triangulation in the previous section, we are required to know two values. The distance from the camera to the laser can be set, but the angle made between the laser and it's reflected component are not known. To find this angle, a closer look to the distances involved inside the camera is required.

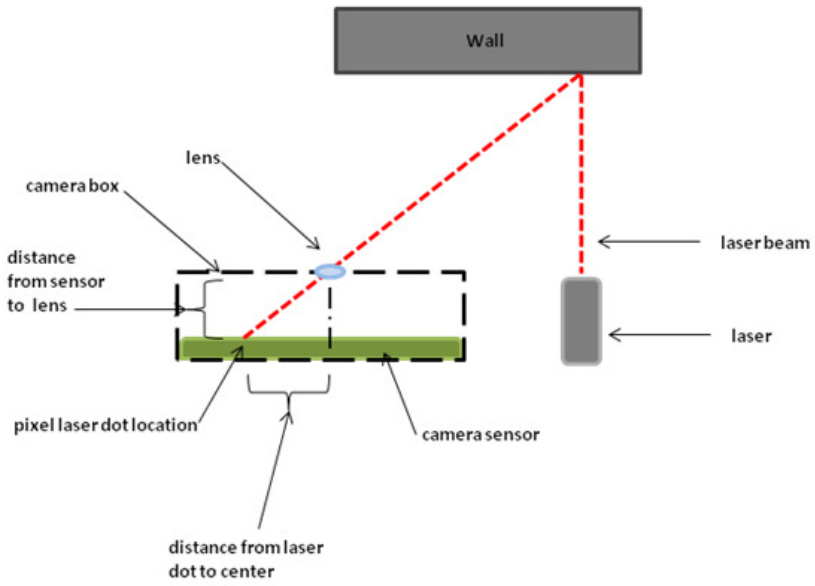

Figure B2: Distances within a camera

In Figure A2, the reflected laser beam off the wall enters the camera box through the lens. The distance from the sensor to the lens, controls how far the distance from the laser dot to the

center of the image sensor is. Below is a more concise descriptions of all the distances involved in triangulation.

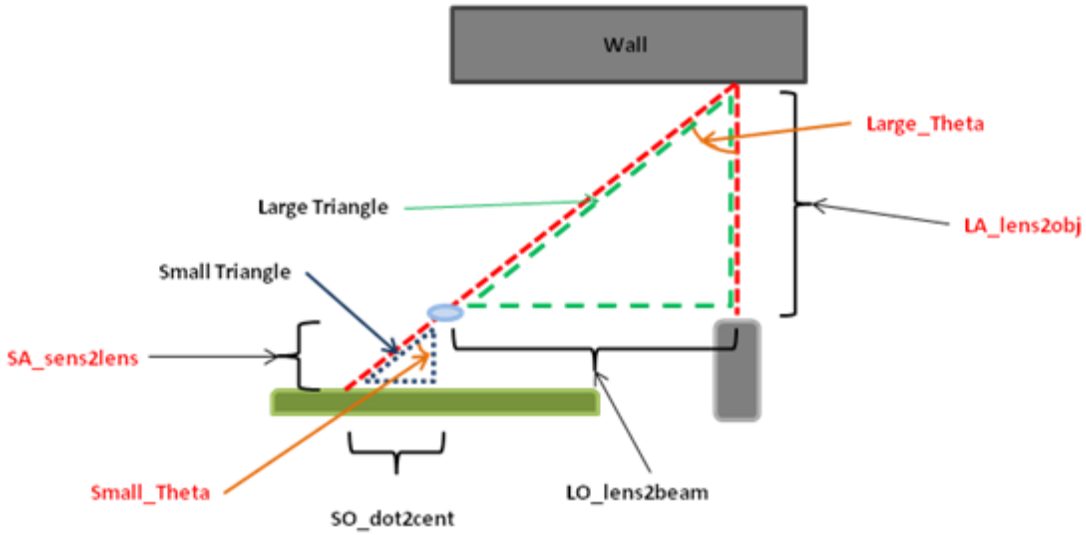

Figure B3:Labels of distances and angles. Unknown values are labeled in red

Below is a description of all the terminology and labels presented in Figure B3:in order of top to bottom.

- **Wall**: The wall or object the laser beam reflects off.
- **Large_Theta**: An angle made between the original laser beam and the reflected laser beam. This angle belongs to the *Large Triangle.*
- **Large Triangle**: The right angle triangle formed between the camera lens, laser beam, and wall.
- **LA_lens2obj**: A distance standing for "Large Adjacent side from Lens to Object". This is the distance from the camera lens to the wall, which is the adjacent side with respect to *Large_Theta* of the *Large_Triangle*.
- **Small Triangle**: The right angle triangle formed between the position where the reflected laser beam hits the image sensor, the centre of the image sensor, and the back of the lens.
- **SA_sens2lens**: A distance standing for "Small Adjacent side from Sensor to Lens". This is the distance from the surface of the image sensor to the back of the lens, which is the adjacent side with respect to *Small_Theta* of the *Small Triangle*.
- **Small_Theta**: An angle made between reflected laser beam after it enters the lens and the line from the centre of the image sensor to the centre of the lens. This angle belongs to Small Triangle.

- **SO_dot2cent**: A distance standing for "Small Opposite side from Dot to Centre of sensor". This is the distance from where the reflected laser beam hits the image sensor causing a red dot to the centre line of the image sensor, which forms the opposite side with respect to *Small_Theta* of the *Small Triangle*.
- **LO_lens2beam**: A distance standing for "Large Opposite side from Lens to laser Beam". This is the distance from the centre of the camera's lens to the centre of the laser beam, which forms the opposite side with respect to *Large_Theta* of the *Large Triangle*.

B2.1 Known and Unknown Distances

From Figure B3, the distances that are already known are LO_lens2beam and SO_dot2cent. LO_lens2beam, or the distance between the camera's lens to the laser beam can be easily measured with a ruler before any image capture. SO_dot2cent is represented as a pixel count from where the blob detection algorithm detect the reflected laser to the centre of the image. This assumed to be perfected in software.

The distances and angles that must be calculated through triangulation are LA_lens2obj, SA_sens2lens, Large_Theta, and Small_Theta. The distance that is most significant to us is LA_lens2obj. this is the distance from the camera to the object in the image. The distance that is most inaccessible is SA_sens2lens, since this distance is within the camera and may need calibration for every camera model. Due to the fact both the Small and Large Triangle have right angles and share the same hypotenuse, their angles Large_Theta and Small_Theta are the same magnitudes.

To calculate the most important distance, LA_lens2obj, we would use the following tan equation, hereby called the "Interpolation Equation":

Equation 1: Interpolation Equation

$$\text{LA\_lens2obj} = \frac{\text{LO\_lens2beam}}{\tan(\text{Large\_Theta})}$$

Since LO_lens2beam is one of the distance already known, we can interpolate the distance from the camera to the object if Large_Theta is known. Large_Theta can be indirectly calculated by solving for Theta2 using the following formula:

Equation 2

$$\text{Small\_Theta} = \tan^{-1}\frac{\text{SO\_dot2cent}}{\text{SA\_sens2lens}}$$

Equation 3

$$\text{Large\_Theta} = \text{Small\_Theta}$$

The only unknown in Equation 2 is SA_sens2lens, the distance from the sensor to the lens. If this is known, it can be use in the interpolation formula. Therefore, a method to calibrate this distance must be developed.

B3. Calibrating Sensor to Lens Distance

As justified in the previous section, the image sensor to lens distance, SA_sens2lens, must calibrated for the camera we have available to us before we can interpolate the distance from the camera to an object.

The following experiment proposes a theoretical procedure to go about finding this distance.

B3.1 Experiment 1:

THEORY

In this experiment to calibrate SA_sens2lens distance, multiple images will be taken at different distances from an object. For each image captured, the distance from the object to the camera will be recorded. This will provide a reference LA_lens2obj value. Also the distance from the centre of the camera lens to the laser beam will be recorded. This will be the reference for the LO_lens2beam value. In this manner the angle, Small_Theta can be readily calculated using:

Equation 4

$$\text{Large\_Theta} = \tan^{-1}\frac{\text{LO\_lens2beam}}{\text{LA\_lens2obj}}$$

Equation 5

$$\text{Small\_Theta} = \text{Large\_Theta}$$

And hence, the SA_sens2lens distance can be calibrated by using:

Equation 6

$$\text{SA\_sens2lens} = \frac{\text{SO\_dot2cent}}{\tan(\text{Small\_Theta})}$$

A theoretical experiment would have a set up much like Figure A4, where a camera and a laser would be placed horizontally together at a fixed distance, while moving at fixed intervals away from a wall. At each interval distance, a test image would be captured.

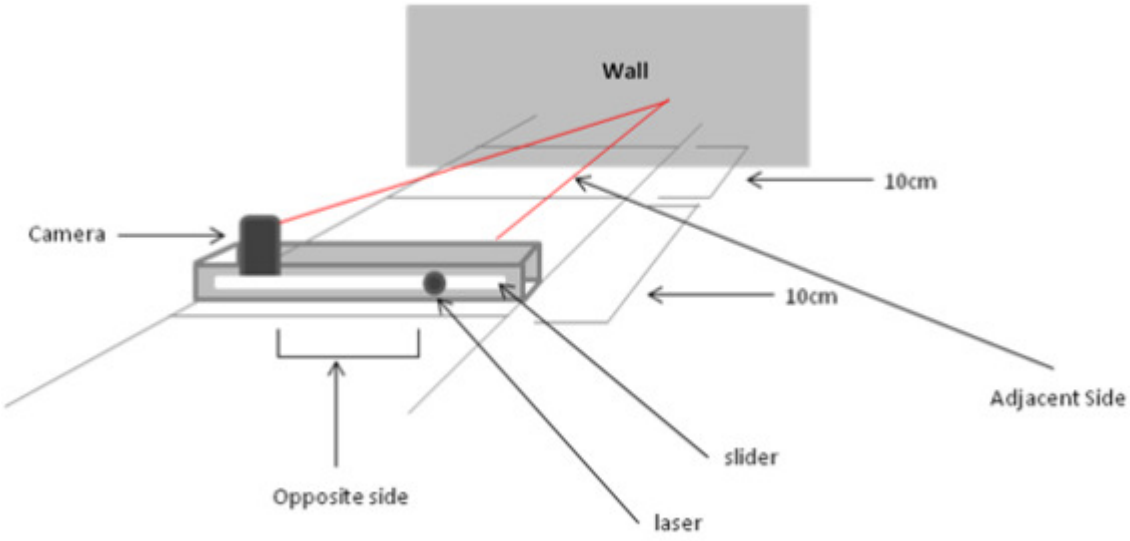

Figure A4: Procedure of Experiment 1

From this arrangement, we can now find values for SA_sens2lens using equations 4 to 6.

NOTE 1:

Since the size of a pixel in meters is not readily available to us, we will convert the SO_dot2cent distance to a normalized percentage, in respect to half the width of the image sensor (measured in pixels).

That is, we will divide the pixel count, between the laser dot and the sensor's middle, by the total number of pixels within half the image. For example, if the laser dot falls three quarters

away from the center of the image to the end, the “normalized percentage pixel distance” between the laser dot and the image center would be 0.75.

If we use the normalized value for SO_dot2cent to calculate SA_sens2lens, then the SA_sens2lens distance will also be a normalized value in respect to half the width of the image sensor (measured in pixels). Equation 6 would then be written as:

Equation 7

$$\text{SA\_sens2lens}_{\text{NORMALIZED}} = \frac{\text{SO\_dot2cent}_{\text{NORMALIZED}}}{\tan(\text{Small\_Theta})}$$

Once the normalized value is for SA_sens2lens is confirmed, we have an appropriate comparable ratio to use during interpolation:

RESULTS

The relative location of the laser on half the image width for each distance interval is shown in Figure B5.

Figure B5: Blob detection for Experiment 1

The following values for SA_sens2lens were calculated during experimentation, given in Table B1.

Table B1: Results of Experiment 1

LA_lens2obj [m]	LO_lens2beam [m]	Theta [degrees]	SO_dot2cent [pixels]	SO_dot2cent [normalized] (1296 pixels wide)	SA_sens2lens [normalized]
0.1	0.07	N/A	Dot off sensor	infinity	N/A
0.2	0.07	19.29	804	0.6203704	1.7724874
0.3	0.07	13.13	540	0.41666666	1.7857126
0.4	0.07	9.26	372	0.28703704	1.6402127
0.5	0.07	7.96	300.0	0.23148148	1.653437
0.6	0.07	6.654	252	0.19444445	1.6666644
0.7	0.07	5.71	228	0.17592593	1.7592565
0.8	0.07	5.00	156	0.12037037	1.3756623

The average value of the SA_sens2lens normalized distance (if the outlier measurement at 80cm is rejected) is 1.75 times the half of the image sensor. This constant can now be used to find the distances in other experiments by using the following formula:

Equation 8

$$\textbf{LA\_lens2obj}(\text{SO\_dot2cent}, \text{LO\_lens2beam}) = \frac{\text{LO\_lens2beam}}{\tan(\tan^{-1}\frac{\text{SO\_dot2cent}_{\text{NORMALIZED}}}{1.75})}$$

B4. Conclusions

The distance from the sensor to the lens, and the distance from the lens to the laser is required to triangulate the distance from the lens to an object reflecting a laser. This document has shown the equations used to calculate distances using the location of laser light on the image sensor, and the reasoning behind those equations. Equation 8 should be used to provide distance measurements, once internal camera measurements are found using the experiments laid out in this document.