CS4758: Rovio Augmented Vision Mapping Project

Sam Fladung, James Mwaura

***Abstract*— The goal of this project is to use the Rovio to create a 2D map of its environment using a camera and a fixed laser pointer mounted on the robot. It uses basic visual algorithms to isolate the angular location of the laser dot in the frame, and uses that to determine distance to the object.**

A non-precision mounting setup introduces error into the system. To compensate for this, we have applied supervised learning techniques to derive an estimate for the actual position and angle of the laser pointer on the Rovio.

These parameters and additional training images are processed using statistical error modeling techniques. The specific error models were selected to match the data measured (Gaussian). Using this information and a localization reference from the Rovio base station, the Rovio builds maps.

I. Introduction

The Rovio™ mobile webcam only provides image and localization data. It is a proprietary system that does not allow any additional inputs to be wired in. This precludes the use of traditional range finding sensors, like ultrasonic range finders. In addition when working with a cheap off-the-shelf systems like the Rovio, it is desirable not to have to add expensive sensors on them. This creates a need for a way to add a cheap distance sensor without having to connect to any new hardware lines. This project builds a new sensor by mounting a laser pointer on the Rovio. Using its existing camera, and the dot projected by the laser pointer, it is able to determine the range to an object. This method does not require any additional input channels on the robot. This technique can be applied to any robotic system with access to a camera.

This paper is organized as follows. In section II, we discuss related work in the field then proceed to lay out the basic geometry underlying our system in section III. Next, in section IV, we discuss our application of supervised learning to correctly determine the precise geometry of the given system. We then provide an overview of the software architecture as implemented on our robot in section V. In section VI, we give a short description of the mapping system and then proceed to discuss our experimental setups and testing results in secions VII and VIII respectively. Next, we give a brief system evaluation discussing appropriate application situations for this system in section IX. Lastly, we conclude and present suggestions for future improvements on this system and methodology.

II. Related Work

A paper that describes a similar method is "Obstacle Detection With Active Laser Triangulation" [1]. This paper focuses mainly on the obstacle detection using a fixed laser and camera on a wheel chair. We extend this idea to integrate with data from the Rovio's localization system. This creates obstacle location data that is suitable for use in a map creation system.

Another reference paper on this is by Quigley *et. al* [2] This paper describes the use of an actuated laser to implement a line scan with a camera. This provides more data points but at higher complexity than our system, and places greater requirements on the output capacity of the robotic system. Since robots like the Rovio do not provide outputs capable of controlling extra sensors, this technique cannot be directly used on it. In addition, we use a laser that projects a dot instead of a line. We experimented with using a line laser, but with the power class of the laser we were able to acquire, the line did not provide enough contrast to be visible through the Rovio's camera.

III. Extracting Depth from a Fixed Laser in an Image

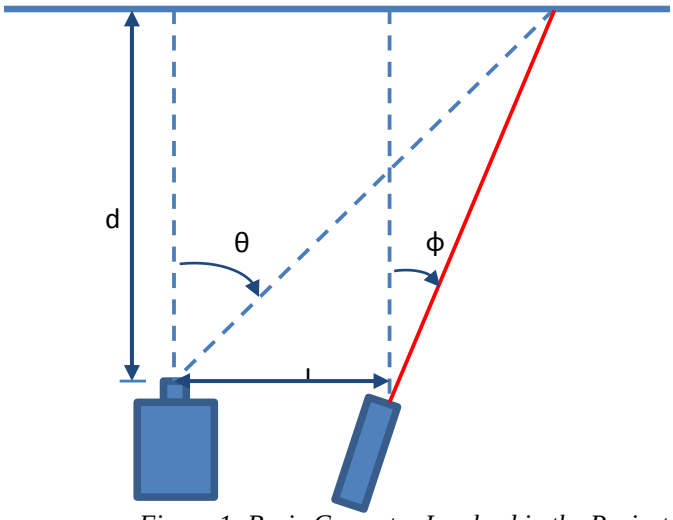


Figure 1: Basic Geometry Involved in the Project

Figure 1 illustrates the basic geometry involved in our project. From this geometry, we are able to extract the distance given the angle at which the laser dot appears in the camera.

$$d = \frac{l}{\tan\theta - \tan\emptyset} \qquad (i)$$

In equation (i), l and ∅ are determined by the mounting position of the laser on the robot. These parameters are learned after mounting, since direct measurement is difficult with any great precision.

IV. Supervised Learning of Mounting Parameters

In order to learn the specific l and ∅ for a given robot and mounted laser, a supervised learning algorithm is used. This learning is accomplished by using a set of training images taken with known distances d to an object. The dataset is created by giving the robot its current distance, and recording that with the angle of the laser dot in the image. This dataset is then processed to find the l and ∅ that minimize the error. Due to the geometry of the system, the resolution is reduced as the distance increases. As such, we use a fractional error metric to weight the errors of the shorter distances higher than the further ones.

$$error = \frac{\sum|(d_{hat} - d)|}{d}$$

$$error = \frac{\sum\left|\left(\frac{l}{\tan\theta - \tan\emptyset} - d\right)\right|}{d} \qquad (ii)$$

Due to the number of local minima in this problem, a straight gradient descent gets stuck in non-ideal values. To work around this, we calculated the error over a large range of l and ∅, and found the global minimum.

To test the values learned, we use a separate set of test images collected in the same way. We calculated d_{hat} using the equation (i) above, and compared this with the measured value of d, and calculated the mean error

$$mean\ error = \sqrt{\sum \frac{(d_{hat} - d)^2}{N}} \qquad (iii)$$

V. Software System Architecture

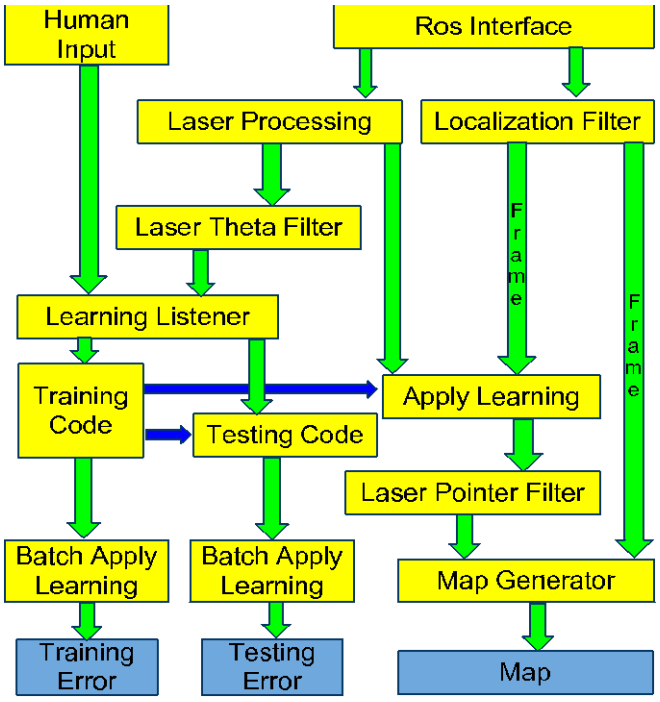


Figure 2: System Architecture

There are three main pathways in the software implementation of this project. These are :

1. Training
2. Testing
3. Application

A. Training

The training code takes input from both the vision processing algorithm and a human trainer. The processing algorithms provide the location of the laser dot in the image which is run through a Gaussian filter to minimize noise. The human input consists of the measured distance to the object that the laser is measuring. These two are written to a file to be used in training the parameters. Once trained, these parameters are passed to various apply learning sections of the code. The training data is run through a batch apply-learning processing which gives the RMS error of each training point and the overall RMS error of the dataset (Training error).

B. *Testing*

The testing code follows a similar path to the training code in taking data from both the visual algorithm and the human input. However, instead of being used to generate new parameters, this dataset is only used to test the previous parameters. This dataset is run through a batch apply-learning process using parameters generated from the Training section of the code. This supplies the RMS error of each point in the testing set as well as the overall RMS error of the testing dataset (Testing error).

C. *Application*

This section of the code uses the learned parameters in order to provide a stream of distance measurements. Each image is processed using the same frontend as the training system and each output position of the dot in the image is converted into a distance measurement. These distances are filtered using a Gaussian filter to reduce noise in the sensor. In parallel to the processing of the laser image, the Rovio localization data is recorded and filtered using a Gaussian filter, in order to provide a stable position reading. The measured distance from the learning algorithm is stamped with this position as a header. This provides the metadata required to figure out where the object was in the global reference frame instead of relative to the Rovio. These annotated distances are then fed into the mapping subsystem.

VI. Vision System

In order to isolate the laser dot, the images are run through a set of filters. Since, by the geometry of the system, the dot can only appear on the right side of the image and in a limited range over the y-axis, we construct a mask to remove the rest of the image data. This mask is constructed by creating a rectangle of white over the area we need to keep and setting the rest of the mask image to black. This mask is ANDed with the input image to remove the unwanted sections of the image. This is applied across all three of the color channels.

The image is then shifted into Hue Saturation Value (HSV) coordinates. A threshold is then applied to the hue channel to isolate the areas that are close to red. The resulting image is ANDed with the value channel to mask of areas that are not red enough. The coordinates of the brightest remaining pixels in the image are obtained and converted into an angle by multiplying the field of view of the camera by the ratio of the pixel position along the x-axis relative to the width of the image.

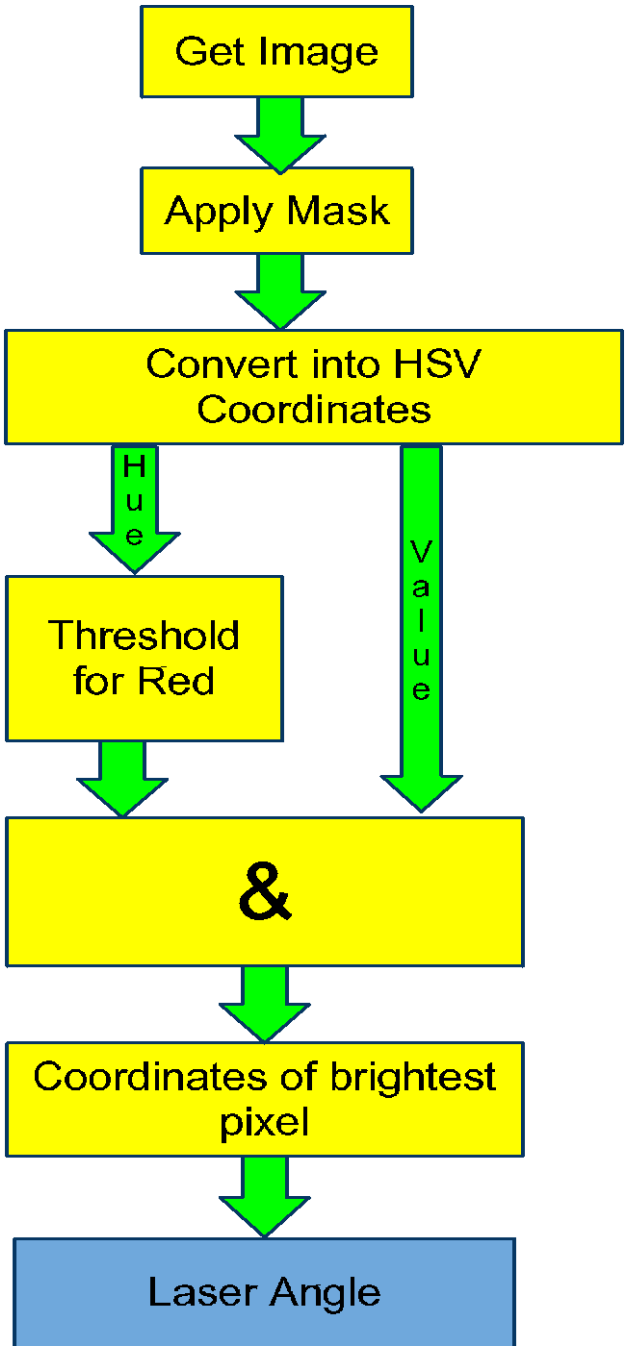


Figure 3: Vision System

VII. Mapping

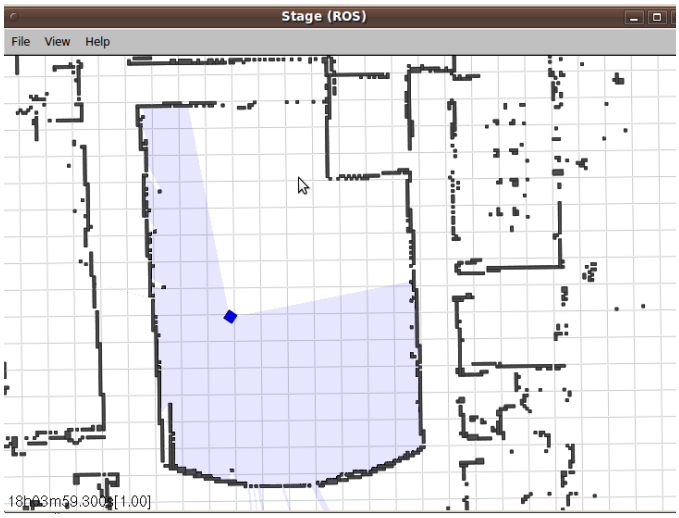


Figure 4: Simulation in Stage

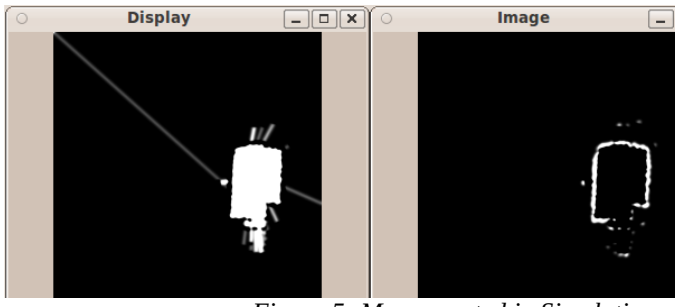


Figure 5: Maps created in Simulation

The mapping code we implemented uses a grid held within an image. Each time a laser reading is taken, a Gaussian blob is added, centered on where the object is believed to be. In addition, a Gaussian blurred line is subtracted along the path between the robot's and object's perceived positions. This results in decreasing the probability of an obstacle in the path that the laser is believed to indicate is empty, and to add to the probability that an object exists at a position that the laser sensor indicates that there's an obstacle. Figure 5 shows a simulation run using the mapping software, adding a Gaussian error to the position of the robot, and only using the central laser values from the simulated laser sensors. The left image indicates areas that are believed to be empty (negative values). The image on the right shows where obstacles are believed to be in white (positive values). The black areas in both images indicate areas that the robot has no knowledge of.

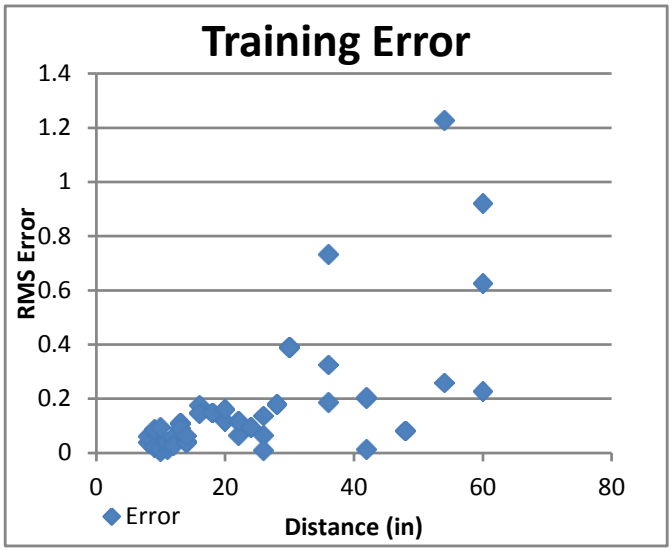
VIII. Experiments

A. Test Methodology

In order to obtain good reference datasets, the Rovio was rigidly mounted on a square platform to keep it aligned with the coordinate system and to provide a reliable way to accurately position the Rovio for measurements. A tape measure was affixed to the floor perpendicular to the object to be used in collecting the data. The Rovio was moved to various points along this axis and several datapoints were collected at each position. Clear outliers from the dataset were eliminated to avoid unduly affecting the training parameters. Testing images were collected in a similar manner.

In order to test the suitability of the system for mapping, the Rovio was positioned in side of an area where most of the walls were within the range of the sensor's reliable operation i.e. not looking down a long corridor. The Rovio base station was setup in this area in order to provide accurate localization. The Rovio was then driven using an open-loop controller to slowly rotate around in circles in discrete steps. This allowed it to see an area around it by 360°. While doing this, the Rovio continuously took range readings from the laser sensor. The laser scans were accumulated in RViz in order to provide a basic drawing of the environment around the Rovio.

B. Test Results



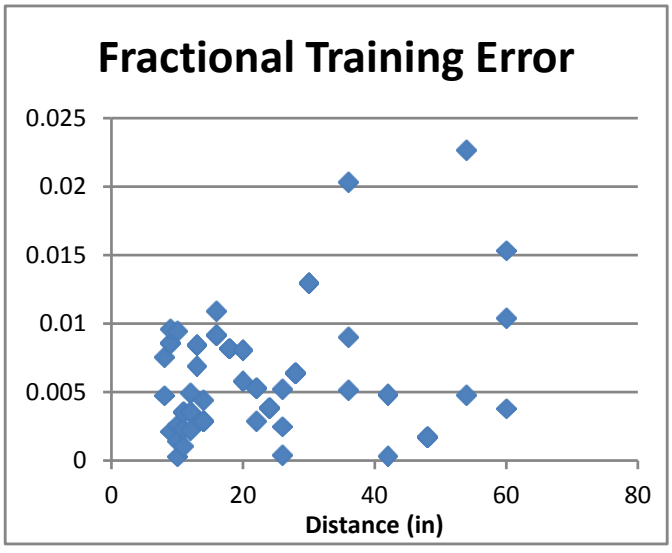
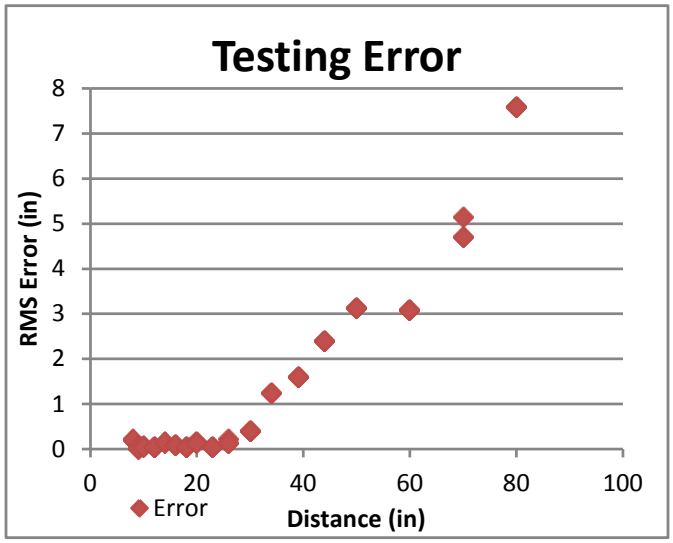


Figure 6: Training Error

The average training error was found to be 0.2657in RMS (0.625%). As can be seen from the graph, the bulk of this error is located in the larger distance measurements as expected when using the fractional error metric.



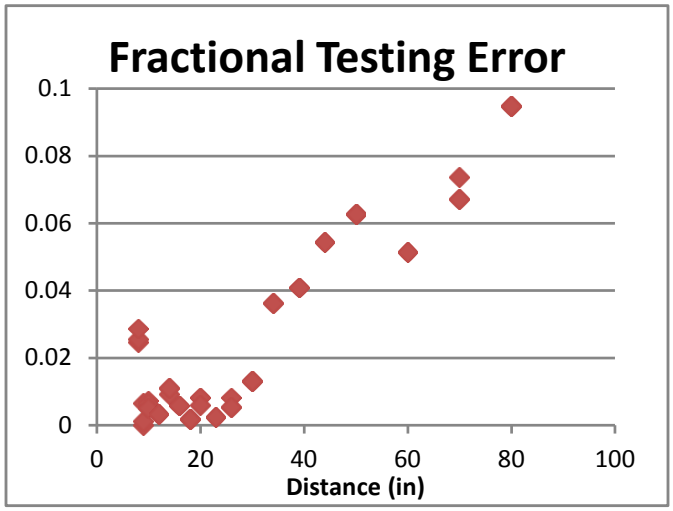


Figure 7: Testing Error

The testing error was found to be 2.4956in RMS (2.72%). Most of this error is again in the longer distances. The RMS error for distances less than 30in decreases to a miniscule 0.1150in (0.715%). This implies that this algorithm and setup work well for measuring moderate distances, but loses accuracy as the distance increases.

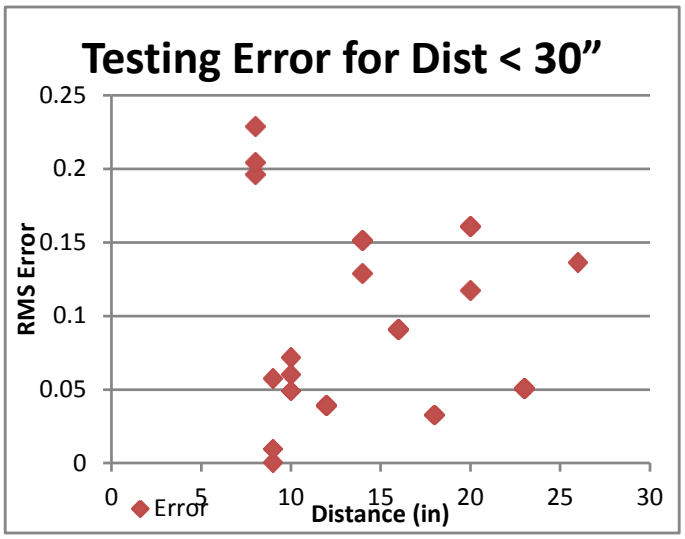


Figure 8: Testing Error for distances less than 30in

The test results can be seen in the screen shot of RViz in Figure 9.

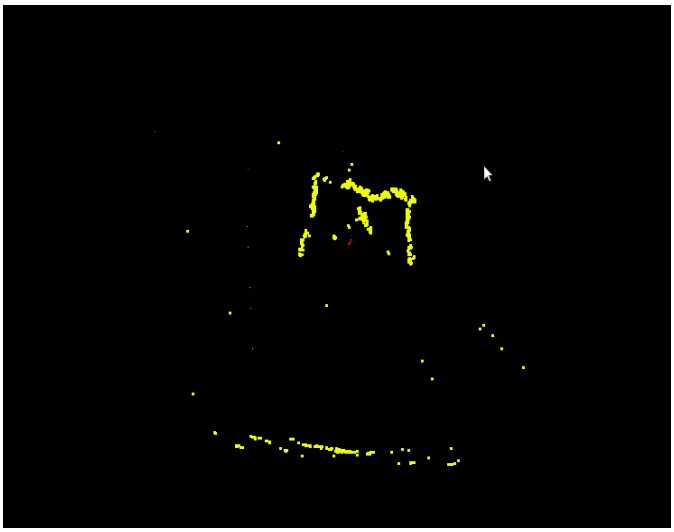


Figure 9: Map created in tests

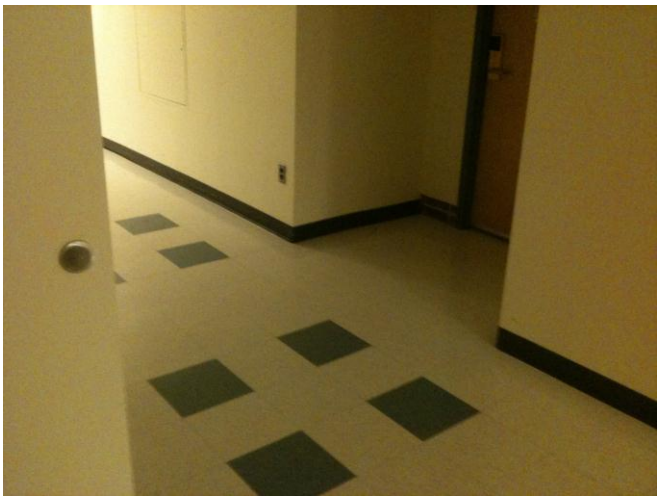


Figure 10: Location where map was taken

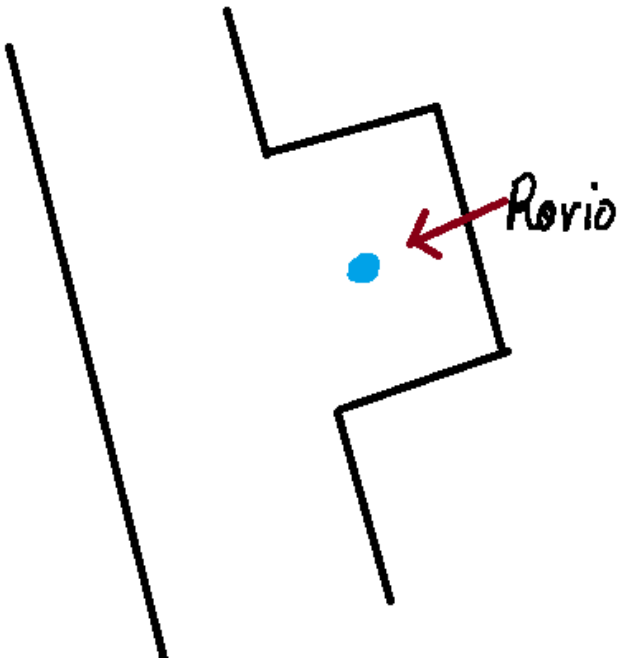


Figure 11: A sketch of the floor plan of the location

The image outline shows the three walls of an alcove and the opposite side of a hallway. Some erroneous points can be seen where the laser was not correctly identified on the longer distances due to adverse lighting conditions. This resulted in the Rovio measuring bad values, usually near zero.

Another interesting feature that can be seen is the bow that appears in the back wall of the alcove. This occurs because the distance measured is to the laser dot and is not in line with the camera. This transform was not taken into account in the code. This artifact can be corrected by applying an additional shift in the angle by the parameter $\emptyset$, as this is the direction the laser pointer was pointed at relative to the Rovio's camera.

IX. System Evaluation

This system has a number of features that make it desirable for a certain subset of robotic work. Laser pointers are cheap, and the mounting does not require any great precision. This allows it to be quickly installed on a system without any specialized skills. This method does not require any direct access to the system hardware and can be added to any robot system that has a camera. This is particularly useful when working with proprietary platforms not originally intended for robotic work.

While this system boasts a number of advantages, it is not without its limitations. Since the laser dot and camera are not on the same axis, it is possible for objects to occlude the laser dot from the camera. In addition, due to the non-linearity of the geometry of the setup, there is a tradeoff between the minimum distance that can be measured, and accuracy at distance. This tradeoff is accomplished by varying the angle $\emptyset$ of the laser. The larger $\emptyset$ is, the higher the accuracy at range, and the greater the minimum measurable distance. Also, as the distance increases, the area represented by a pixel increases faster than the laser disperses. This makes it hard to pick up the laser dot in the image.

This technique also has difficulty with objects that are extremely reflective (shiny metals) or light absorbent (matte black). This is because these materials do not effectively radiate the laser light back in the direction of the camera.

X. Conclusion

In conclusion, we have demonstrated a method for finding distances using a single camera and a fixed laser. We've demonstrated that a supervised learning technique eliminated the need for a precise mounting and alignment of the laser. The system can be applied to any other system with a camera on it and can have a laser glued, stapled, taped or otherwise affixed on it. We've also demonstrated that this type of distance measuring technique can be used to create basic maps of an environment.

There are a few a few areas that future research into this type of system could explore. One obvious method to improve performance on this setup would be to use a more powerful laser or a different color laser. This would improve contrast and increase range. A more powerful laser would also enable the projection of different shapes, like a line, and this would allow a mapping in three dimensions instead of two.

The vision algorithms could also be improved to better isolate the laser from the surrounding environment. This would allow this method to work with a greater variety of textures and at greater distances.

Another possible improvement would be to use Kalman filters. A Kalman filter would allow greater movement of the Rovio while mapping, as opposed to the current Gaussian filters. With a Kalman filter in place, it makes sense to create a SLAM system, getting rid of the need for a base station.

In this project, we demonstrated the ability to get distances and create maps using only the data form the laser. An extension on this would be to use the rest of the image data from the camera for augmenting the map.

XI. Acknowledgements

We would like to thank Professor Saxena for providing feedback on our project and suggesting alternative error metrics to use for our supervised learning implementation. We would like to thank the fine folks at WowWee™ for developing the robotic platform we used and the developers at ROS.org for developing a framework for inter-process communication and linking together the various robotic packages.

XII. Additional Media

A video showing the project in operation was also submitted as part of the poster session.

XIII. References

1. *Obstacle Detection With Active Laser Triangulation.* **Klancnik, S, Balic, J and Planinsic, P.** 2007, Advances in Production Engineering & Management, pp. 79-90.

2. *High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening.* **Quigley, M., et al., et al.** 2009, International Conference on Robotics and Automation, pp. 3604-3610.