

# Addressing CML by Using Line Detection Techniques for More Reliable Sonar Information.

*Eric M. Meisner, CSC 242Term Project*

*University of Rochester 2003*

## **Abstract**

The purpose of this paper will be to describe a method for performing concurrent mapping and localization, by extracting more accurate information from Polaroid sonars on a mobile robot. This paper outlines an attempt to cast the problem of noisy sensor data as a computer vision problem, and then apply basic vision techniques, such as the Hough transform and morphological operations to find relevant features in large collections of noisy sonar Data. By utilizing what is known about how sonar reacts to flat surfaces, we can use the resulting image as a map and as a solution to one half of the concurrent mapping and localization problem.

## **Background**

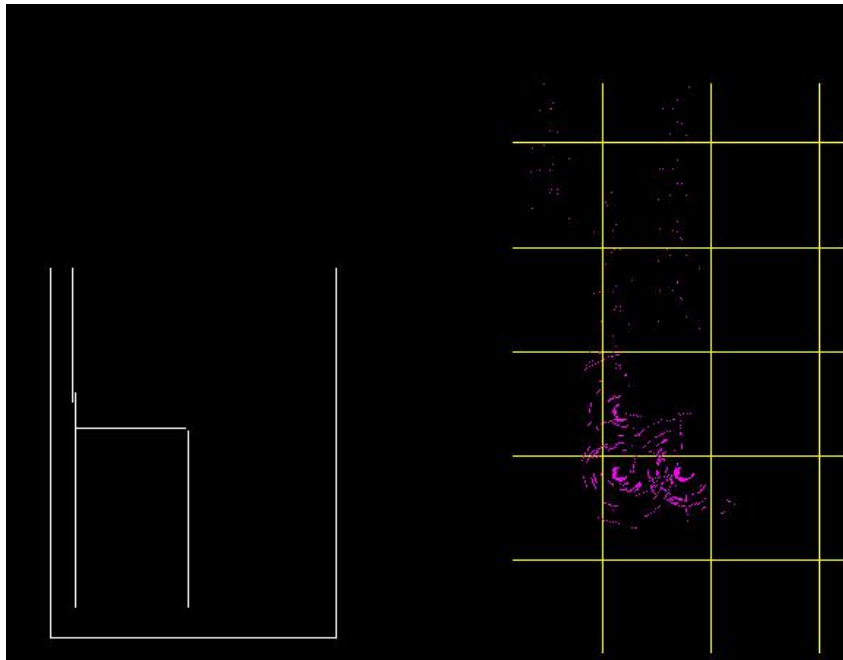
The most generally accepted solution to the problem of localization is Monte Carlo localization described by (Thrun et al.) This probabilistic method of particle filtering requires prior knowledge (i.e. a previously generated map) of the mobile robot environment, but is robust to robot abduction. The problem of concurrent mapping and localization (CML also referred to as SLAM, simultaneous mapping and Localization) is described by (Newman et al.) as simultaneous map generation and localization. This is a particularly difficult problem. The problem of concurrent mapping and localization is described by Newman et al. as a bootstrapping problem in which the robot uses sensor information to estimate geometric features while at the same time estimating the pose (X, Y, and theta offset from the initial) of the robot. This is especially difficult using information from sonar rather than laser, however it is not trivial to pursue given the low cost, portability, and size of sonar, compared to laser devices. Thrun et. al. Describes Monte Carlo localization as a method for representing the uncertainty in position and sensor information. This representation is not totally irrelevant to CML, however in this particular case, there is no prior knowledge of the environment. The core of this problem is to extract reliable data from generally unreliable sensors, so that the Monte Carlo localization can be performed using the generated map.

The first step in this process is to formally define the level of uncertainty in the robot position and sensor information. Robot odometry is used to estimate the position of the robot. The level of uncertainty can be described as a function of drift in motor encoders multiplied by time. No corrections can be made to the robot position until at least a partial map is constructed. This is formalized (Newman et.al.) by saying that the system is solvable when:

$$NK \geq R(K-1) + NF$$

Where  $N$  is the number of sensors,  $R$  is the number of degrees of freedom of the robot,  $K$  is the number of robot locations and  $F$  is the degrees of freedom to determine a feature location. In the case of the Pioneer II AT we have  $N = 16$ ,  $R = 3$ ,  $F = 2$ , and  $K = \text{Average velocity/change in time (Vave/t)}$ . This is equivalent to saying that there is enough sensor information to make a relevant inference about the robot position and it assumes that the geometric features are accurate. The model of information analysis which we will use is described (Newman et. al.) generally: Given a distance returned by some sensor  $S_j$  oriented at  $\theta(S_j)$  on the robot, the distance given by  $S_j$  defines an arc, such that some flat surface is adjacent to that arc.

**Figure 1a – Sonar Responses to real world features.**



**Figure 1b.**

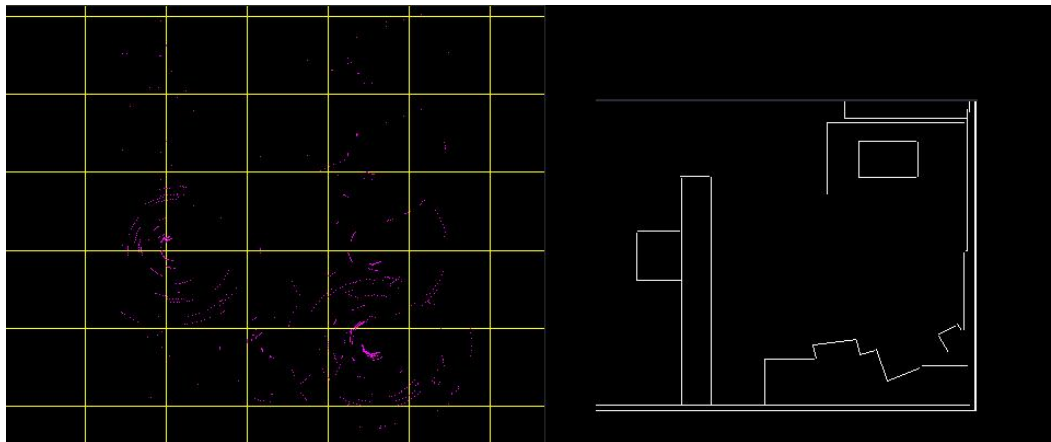
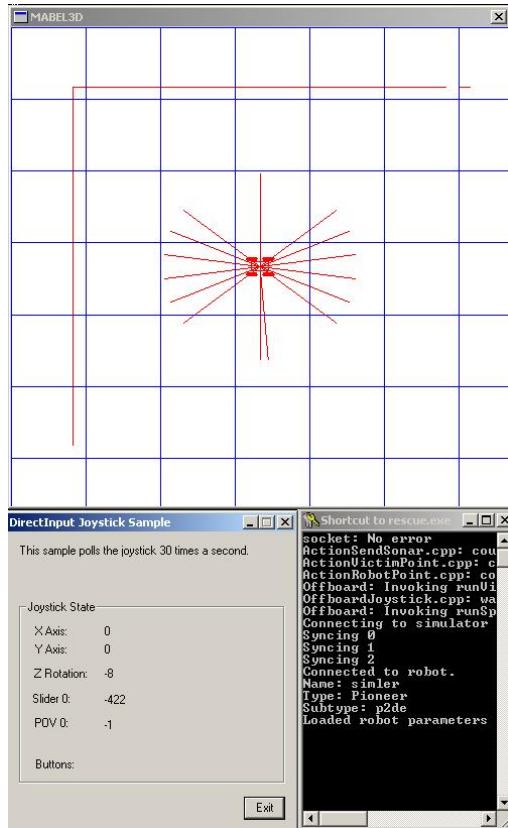


Figure 1a. and 1b. demonstrate this anomaly, showing the real world features and the effect of the sonar response to the feature. The goal is to use a Hough transform to find lines which are tangent to these arcs, since we know that some real world feature must correspond to some tangency to any prominent arc. We know that in general lines representing important features like walls will be tangent to several arcs across the image, and that the more prominent arcs are likely to be more reliable. The implementation of this attempts to optimize this by adjusting parameters for the Hough transform, however the optimal adjustment depends on the data set and is somewhat open-ended.

## Implementation

The implementation utilizes the previously existing University of Rochester Undergraduate robot research team's urban search and rescue system. The system consists of two core components: (1) A navigation application which allows the pioneer 2 AT robot to be controlled by a single operator, while collecting sensor information using an array of 16 sonars and (2) a visualization application which generates a map in a 3D world based on the sensor information it receives from the navigation application.

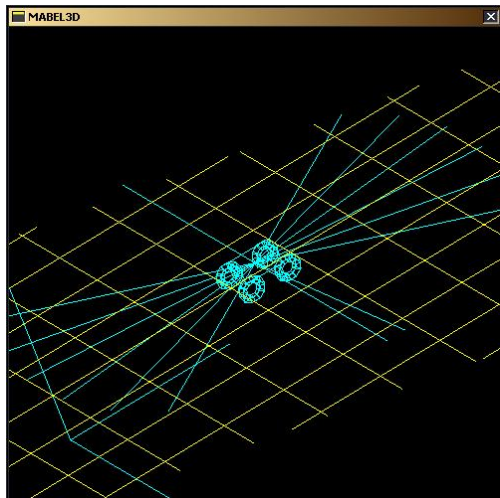


**Figure 2 ,User interface: demonstrates the visualization app. (top) , the remote log in to the robot running the navigation app. (bottom right) and the joystick control client (bottom right).**

All sensor and state information is sent to the visualization application over TCP. The visualization application then generates an image space based on the two degrees of freedom of the sonar information. Line finding techniques are then used to extract features from the image and these features are used to replace point clusters in the robot visualization application. These features should correspond to real world features of the robot environment. The visualization application is implemented using OpenGL and displays the robot state and environment in 3 dimensions, although the sensor data varies only in the x,y dimensions, and has a fixed z direction. The reason for this is, because in the future, additional sensors could be added to the robot with different fixed dimensions. The current sonar array is intended to be useful for object avoidance, but additional sensors could be used to generate more accurate three dimensional maps.

The first step in this process is to gather sonar information in an environment with measurable features. For this process I use the SRIsim robot server simulator. This server, provides feedback and simulates the robot driving in a real environment, including simulated sonar noise and wheel encoder slippage. Previous experience has demonstrated that data collected using the simulator is almost perfectly analogous to real world data collected using the actual pioneer 2 AT. The navigation program connects to this server, and controls the simulated robot, while sending data to the visualization application.

**Figure 3 – 3D visualization application demonstrated from a view other than overhead.**



The visualization app. uses the x, y of the sonar points to generate and an approximate overhead binary image of the robot environment. Do accomplish this ,I utilize the open source Computer Vision (openCV ) and Image processing (IPL) libraries (Intel).

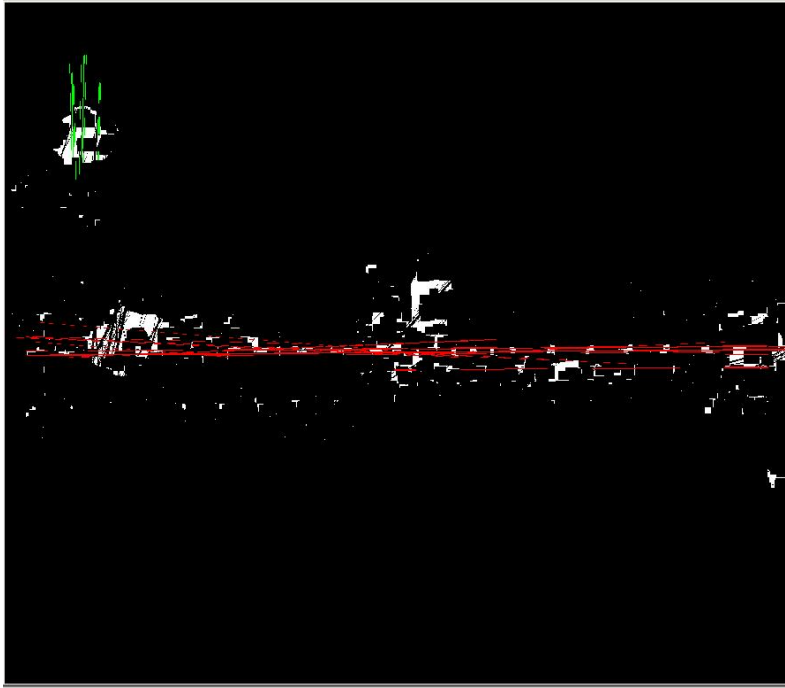
### **Morphological Operations**

Once the image is generated, the application performs several morphological operations on the image to get rid of some of the noise. This is done using Intel's IPL erode and dilate operators . The Erode operator shrinks features in the binary image, by removing the outer pixels of any clusters of light pixels in the binary image. The dilate operator adds pixels back to the edges of the light features. The result of this process is that edges of large features in the image are smoothed and stray pixels are removed, adding uniformity to the image [Brumby et. al.]. The process can be done any number of times. The number of dilation's and erosions which I found produced a uniform, but sufficiently detailed image was five.

### **Hough transform**

After the image is dilated and eroded, the next step is to perform a Hough transform to locate the lines which correspond to world features. The form of Hough which is fastest and most effective on the noisy data sets such as this one is the Progressive probabilistic Hough (Matas et. al.) . The probabilistic Hough operates by, selecting a point randomly from the image space, and casting it's vote in the hough space. The point is then removed from the image space so that it cannot be selected. The probabilistic Hough, has several thresholding parameters, one of which is the number of lines. The voting threshold determines the number of votes which a line must receive to be confirmed. Some of the other parameters are, minimum line length, maximum line gap. Each time a vote is cast, the probabilistic Hough checks the thresholds and selects the relevant lines. Once the maximum number of lines are detected the probabilistic Hough stops. This implementation attempts to optimize these parameters for the given task, but again this depends on the data set and is an open-ended problem. I also added another threshold to the output of the Hough, which is to restrict the slope of any detected lines. This allows us to seek lines which form perpendiculars in the image. In this domain, these lines are more likely to represent world features.

**Figure 4 – Sensor Data cast as an Image, eroded, dilated and finally Hough transformed. DifferentColor lines represent the different classifications of Lines based on length, and orientation.**



The result of this implementation as can be seen in figure (4) . This image demonstrates the result of performing the previously described operations on the image generated from the sonar information. The ultimate goal of this is to be able to use these lines as a map as opposed to the noisy clusters of sonar. The same protocol which is used to send information from the robot to generate the map can be used to communicate this presumably more accurate world feature representation back to the mobile robot. As previously described by Newman et. al. ,a sufficient amount of sensor data must be collected before localization can be attempted but this process may allow for robust techniques such as Monte Carlo localization without prior knowledge.

## References

- 1.) [Thrun et. al.] Sebastian Thrun, Dieter Fox, Wolfram Burgard, Frank Dellaert. *Robust Monte Carlo Localization for Mobile Robots*. Elsevier Science B.V. 2001
- 2.) [Newman et.all.] Paul M. Newman, John J. Leonard, Juan D. Tardos, Jose Neira. *Robust Mapping and Localization in Indoor Environments Using Sonar Data*. MIT marine Robotics Laboratory, 2001.
- 3.) [Intel] *Intel Image Processing Library Reference Manual*, Intel Corporation. Document Number 663791-005 1997 – 2000.
- 4.) [Matas et. al.] J. Matas, C. Galambos , J. Kittler. *Progressive Probabalistic Hough Transform*. Center For Machine Perception Czech Techincal University, 1998.
- 5.) [Brumby et. al.] Stephen P. Brumby, James Theiler, Simon J. Perkins, Neal Harvey, John J. Szymanski, Jeffrey J. Bloch, Melanie Mitchell. *Investigation of Image Feature Extraction by a Genetic Algorithm*. Los Alamos National Laboratory, Space and Remote Sensing Sciences, 1998.

## *Acknowledgments*

This project was completed using software and hardware licensed to the University of Rochester, and the University of Rochester Undergraduate robotics Research Team. The software used includes but is not limited to:

Intel Open Computer Vision Library

Intel Image Processing Library

OpenGL

Aria, by ActivMedia robotics.

Microsoft Visual Basic