

A Light, Accurate, Inexpensive, and Easy to Implement Indoor Localization System.

Mark Woodward

INTRODUCTION:

For my project I addressed the problem of creating a light, accurate, inexpensive, and easy to implement localization system for small mobile robots. There are only two assumptions made 1) that the robot will be operating on a uniform flat surface and 2) that the area directly in front of the robot is clear. Both of these will be elaborated on in the Approach section.

Current systems fail these requirements in a number of ways, including being too heavy, too expensive, inaccurate, or difficult to implement. For example, a very popular sensor to use in localization is a SICK laser range scanner. Not only does the SICK scanner cost several thousand dollars, it is also a large sensor, unsuitable for small mobile robots (see Background).

The system presented here uses the well known Canny edge detection algorithm, for which there are numerous implementations one can use, and Monte Carlo Localization (MCL), an algorithm that is straightforward to implement. Images from a video camera are processed to find the edges nearest the robot, which, given the uniform flat ground plane assumption, correspond to obstacles in the environment; the Canny edge detector is used to extract edges. These edges are then projected onto the ground plane to form range and bearing estimates which are then used by the Monte Carlo Localization Algorithm to localize the robot. The system is light (12 grams), inexpensive (\$150), easy to implement (Canny Edge Detection and Monte Carlo Localization), and accurate (see Approach and Results).

BACKGROUND:

Before describing my system I will present a number of current systems and why they fail to meet the requirements. Table 1 lists each approach followed by the reason(s) for failure.

<i>System Specifics</i>	<i>Failure(s)</i>
Laser Range Scanners	Heavy, expensive (~\$2,000)
GPS (non-differential)	Inaccurate, doesn't work indoors
GPS (differential)	Expensive, doesn't work indoors
Stereo Vision	Expensive, Requires texture
Feature based monocular vision (doorways, etc)	Inaccurate (few measurements)
Feature based monocular vision (markers)	Inaccurate (few measurements), need to modify the environment

Table 1. Common localization systems and their failures
for the task of localizing small mobile robots

MY APPROACH:

The system proposed here uses video captured from a single camera. Obstacles appear as the edges nearest to the vehicle, since the ground plane is assumed to be uniform. These edges are detected using the Canny Edge detector. The range and bearing to the obstacles are computed by projecting the image pixels corresponding to the obstacle edges onto the ground plane, which is possible given the

orientation of the camera and because of the flat ground plane assumption. These range and bearing values are then used as measurements for the Monte Carlo Localization algorithm (1).

- 1) Capture a video frame
- 2) Run the Canny edge detector on the video frame
- 3) Find the lowest edge in each column of the image
- 4) Project each edge onto the vehicle ground plane
- 5) Treat the projections as range and bearing estimates
- 6) Run Monte Carlo Localization using the range and bearing estimates

Figure 1. The localization algorithm

Figure 1 summarizes the algorithm presented here. figure 2 shows a typical image captured from a small mobile robot operating in an indoor environment. Figure 3 shows the result of running the canny edge detector on the image from figure 1. The Canny edge detector takes in two parameter which can be adjusted until a small region in front of the robot, which is assumed to be free, is free from any edges; see (2) for details on the Canny edge detector algorithm. Figure 4 then shows the extraction of the nearest edges to the vehicle for each column in the image.



Figure 2. A typical image captured from a small mobile robot

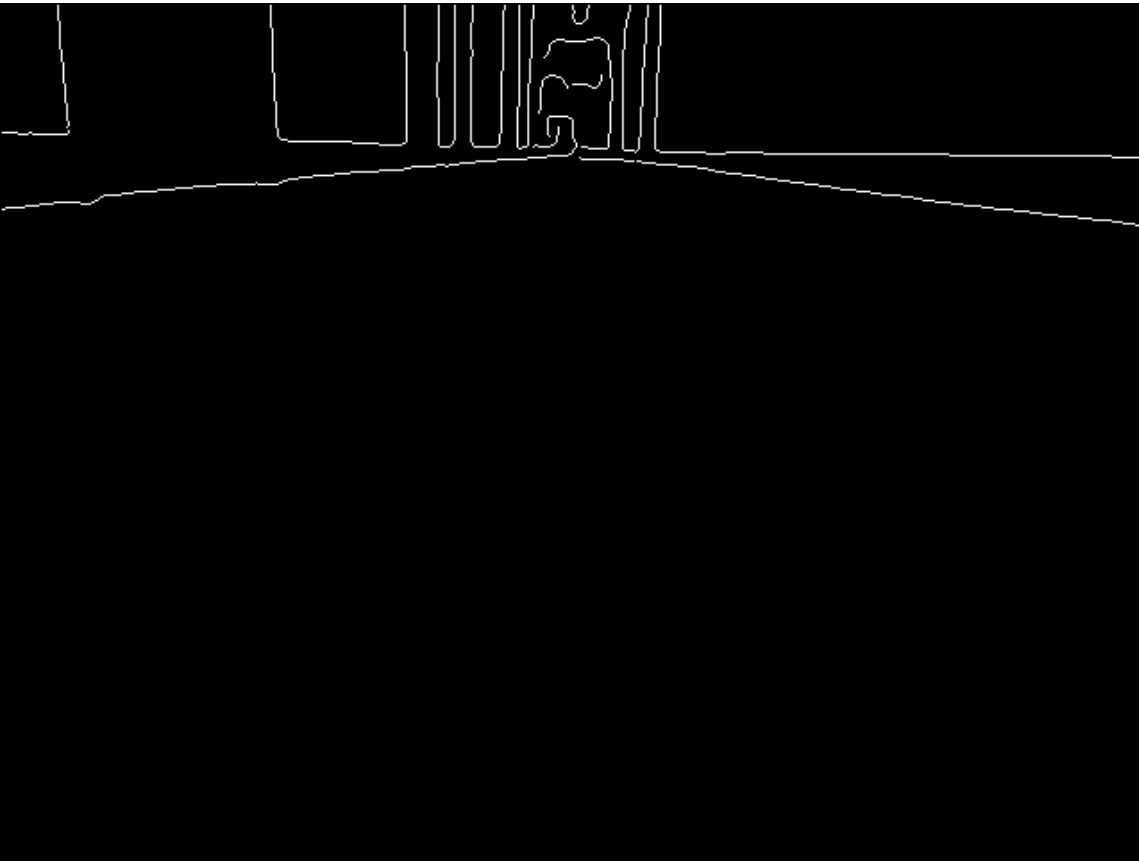


Figure 3. The output of the Canny edge detector run on figure 2

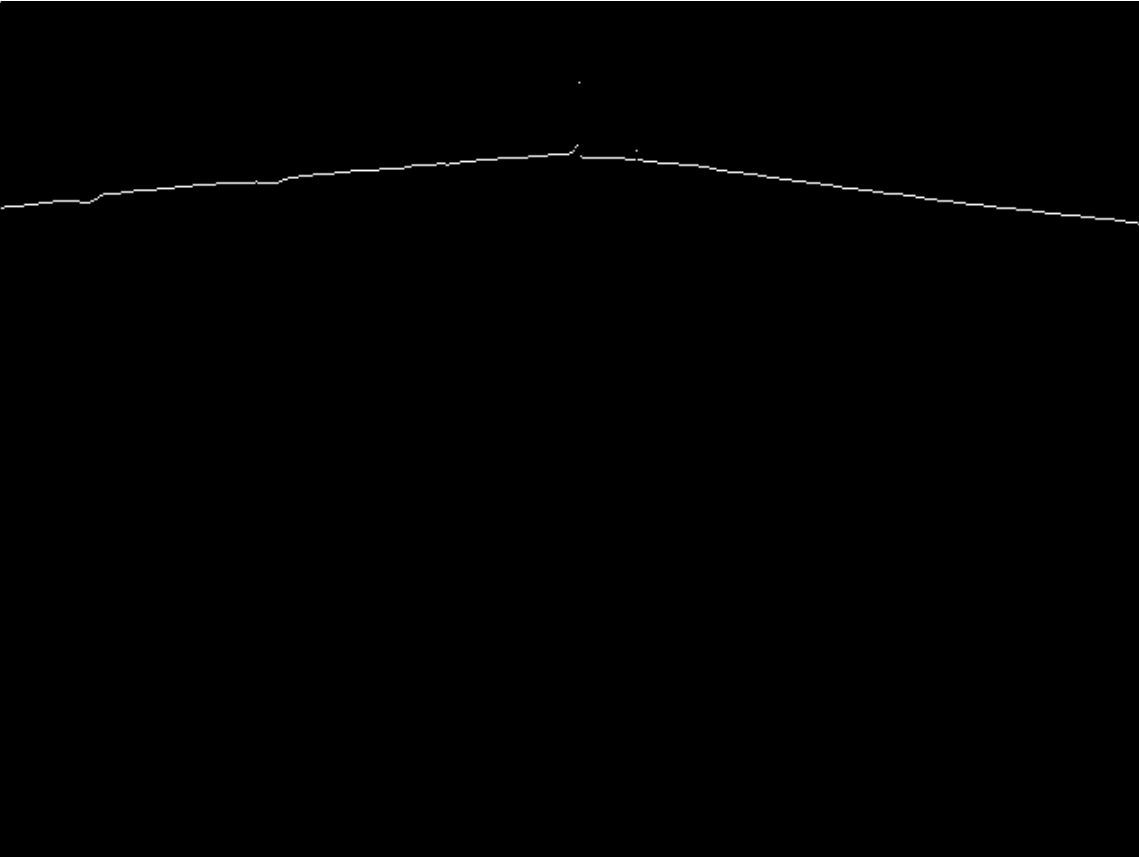


Figure 4. The lowest edges in figure 3

In the next step of the algorithm the pixel representing the nearest edge is projected onto the ground plane, producing range and bearing values in robot coordinates. Figure 5 shows the result of projecting the pixels in figure 4 onto the ground plane for a specific robot.

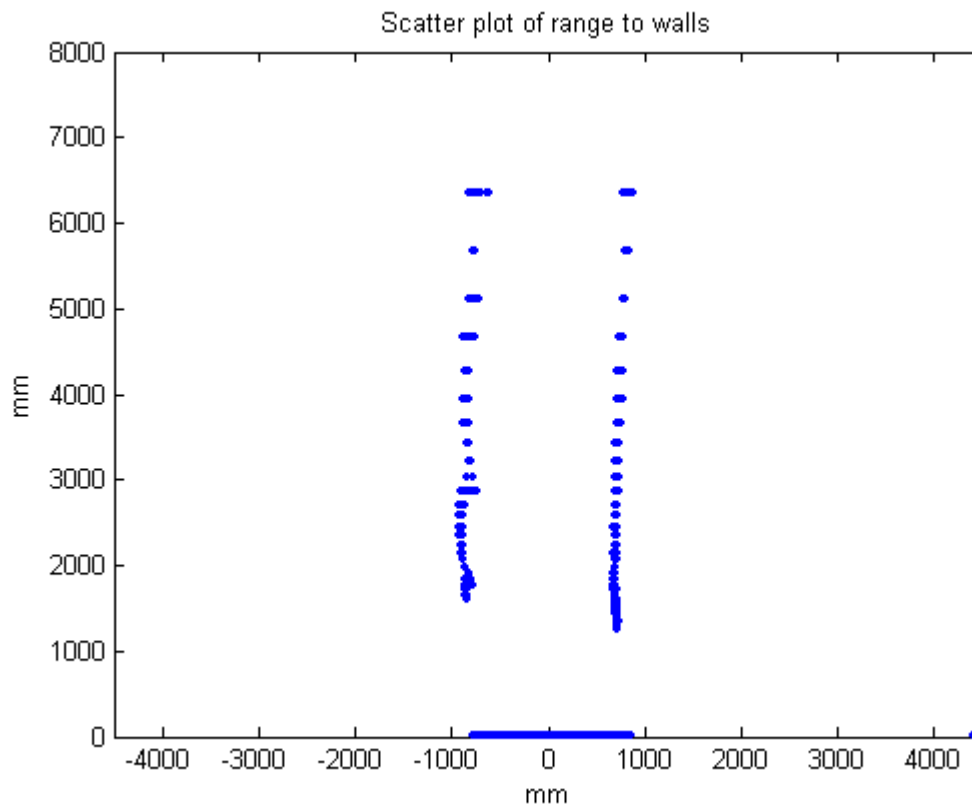


Figure 5. The projection of figure 4 onto the ground plane for a specific robot.

These range and bearing values are then used as sensor measurements in the Monte Carlo Localization algorithm. For details on the Monte Carlo Localization algorithm see (1).

PLATFORM:

The platform used to develop this technique is shown in figures 6 and 7. The mobile robot is a 1/18th scale remote control truck equipped with a wireless camera. Control commands were issued and logged using a standard hobby radio transmitter modified with a PC interface circuit which I developed.



Figure 6. 1/18th scale R/C truck equipped with a wireless video camera

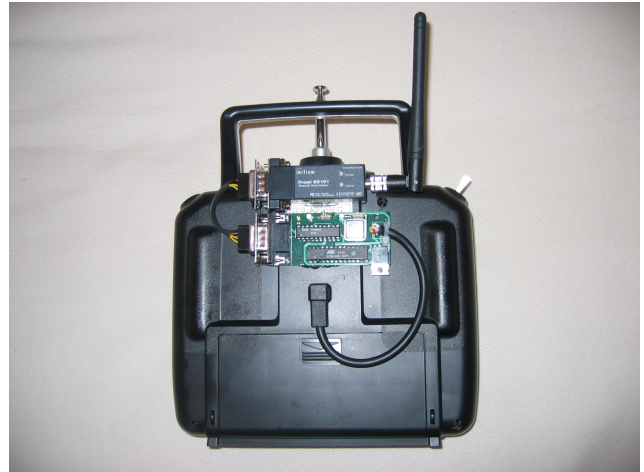


Figure 7. Standard hobby R/C radio transmitter equipped with my PC interface board

It was also necessary to develop a motion model for this vehicle. Figure 8 shows the performance of the motion model and figure 9 shows the cost curve as one of the parameters is being fit.

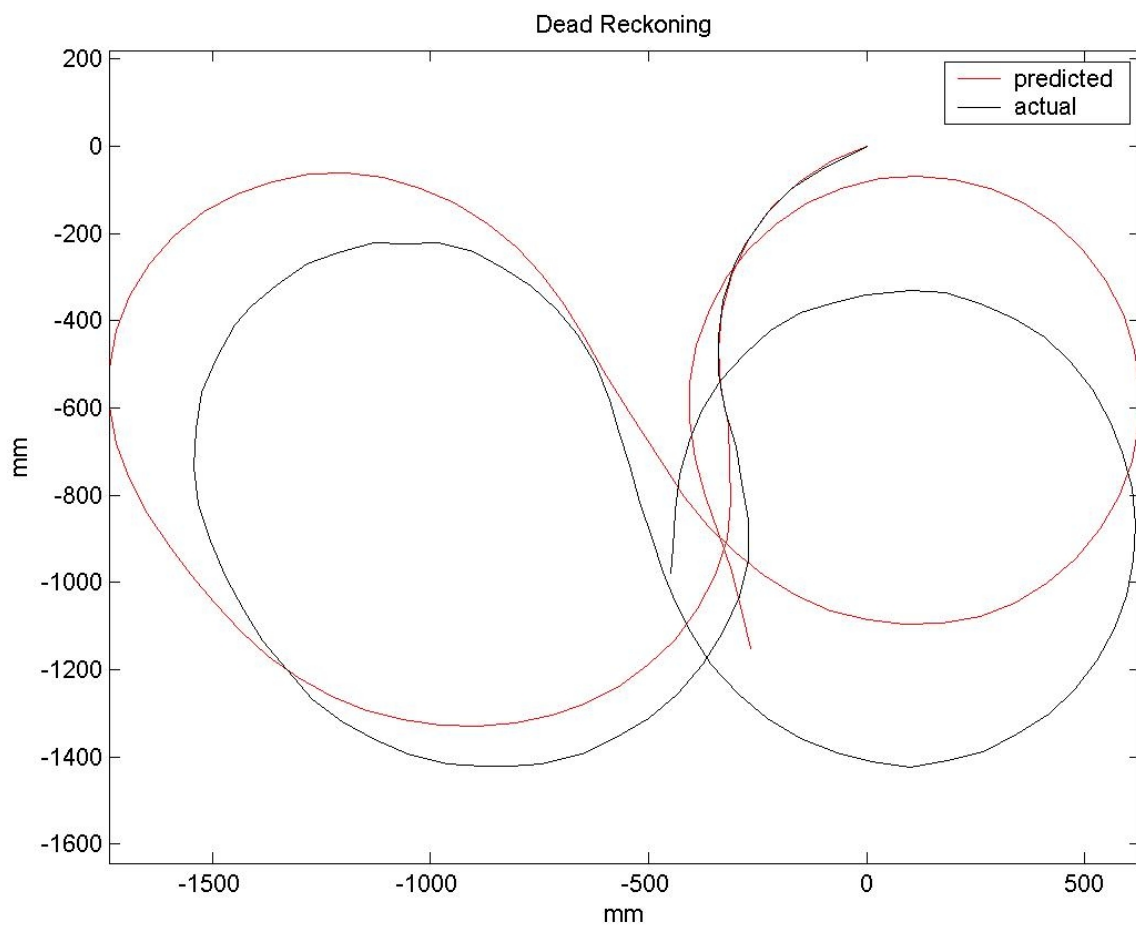


Figure 8. Dead reckoning with the trained motion model vs. the actual trajectory.

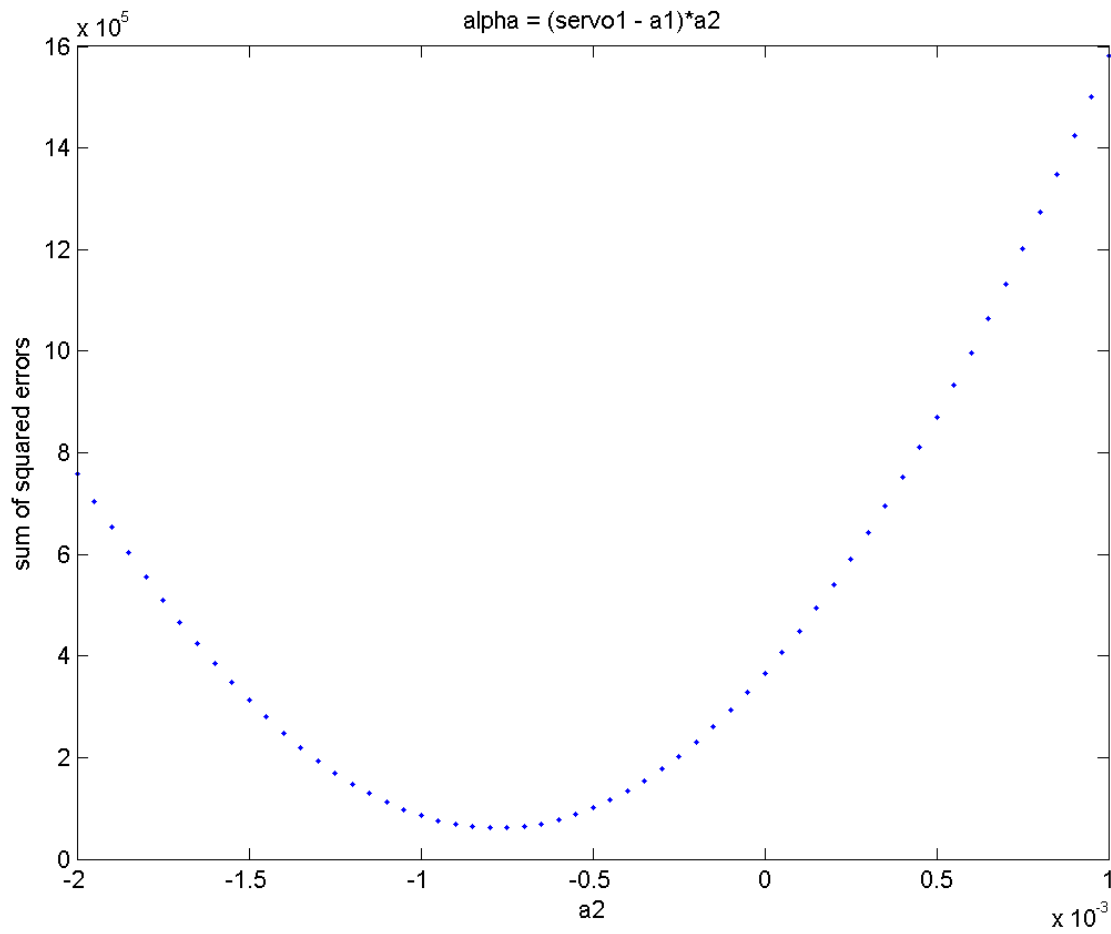


Figure 9. The error curve during one of the minimization steps for fitting the parameters of the motion model.

RESULTS:

The results from running this system in the on the basement of the Stanford Gates building are shown in figure10. Monte Carlo Localization is able to successfully localize the robot using the range and bearing measurements computed using the system described in this paper.



Figure 10. The path of the mean particle (blue) and the path predicted by the motion model alone (white).

PROBLEMS:

There were two major sources of error in the range and bearing estimates during my experiment, these were inaccuracies in the pitch of the robot and corrupted images.

Because the robot is so small, the distance between the camera and the ground is much smaller than the distance between the robot and an obstacle. Given this setup, a tiny error in the pitch estimate can result in a large (± 1 meter) error in range estimation, as a result my experiment ignored measurements beyond 7 meters. In future work, I hope to solve the pitch problem by estimating pitch from cues in the image, i.e. parallel obstacles can be assumed to continue to be parallel, and thus the distance between them should remain constant, which can be used to correct for pitch.

The problem of corrupted images is due to corruption during the wireless transmission of images from the robot to the laptop for processing. Since the frame rate of the video is sufficiently high, corrupted images, if detected, can be thrown away. So, all that is needed is a classifier for corrupted images.

CONCLUSION:

In this paper I presented a light, accurate, inexpensive, and easy to implement localization system for small mobile robots. An Image from a video camera was processed to find the edges nearest the robot, which, given the uniform flat ground plane assumption, correspond to obstacles in the environment; the Canny edge detector was used to extract the edges. These edges were then projected onto the ground plane to form range and bearing estimates which were then used by the Monte Carlo Localization Algorithm to localize the robot. Experimental results showed that the system is accurate and can be used to localize a small mobile robot.

REFERENCES:

1. Monte Carlo Localization for Mobile Robots, Fox, Burgard, and Thrun, ICRA May 1999
2. "Introductory Techniques for 3-D Computer Vision," Trucco and Verri