Accurate and Cheap Robot Range Finder [*]

Ivan Papusha

December 12, 2008

Abstract

A novel high-quality distance sensor for robotics applications is proposed. The sensor relies on triangulation with the offset of a laser line as it is reflected off an object into a cheap webcam. An ML approach to finding the error model showed that the sensor was very accurate when the laser line was found. It is suggested that better line-finding methods than simple color filtering would make the sensor more viable.

1 Introduction

Any self-respecting mobile robot requires some sort of distance measurement sensor or range finding mechanism. Among the most accessible are ultrasonic range finders, which time the propagation and reflection of a sonic pulse. While cheap and ubiquitous, ultrasonic rangers, especially when placed in ring configuration, are plagued with fundamental difficulties that make them unattractive for serious localization.

At the other end of the spectrum are high-end laser rangefinders, which use light instead of sound to determine distance from time-of-flight. These are often bundled with precision optics and mechanisms that allow one to infer 3D point clouds over large and small "human" distances. Their precision and reliability makes them ideal for industrial and research applications, but their cost is prohibitive to the hobbyist or researcher on a low budget.

To solve the problem of optimizing user cost vs. sensor precision and reliability, we consider a different sensor design. Here, the only materials needed are a laser line generator and a camera (e.g., a cheap webcam). The principle of operation is simple: one points the line generator and the camera in the same direction, but offsets them some known distance d_{off} ($\sim$ 10cm) The camera sees the laser line, filtered from the rest of the environment based on brightness and color, which is offset whenever it strikes an object. By calculating the offset distance from the expected position of the laser line, the distance to the object can be inferred.

This sensor, of course, is not perfect and is subject to its own errors. In particular, its error model and corresponding error parameters are quite different from the error parameters of time-of-flight based rangers such as the ones discussed above. This paper will formulate the error model and use an automatic Maximum Likelihood (ML) method that will learn the fundamental parameters of the error model from a proposed calibration scheme [3] [1]. Results of the sensor and sensor model in action are also discussed.

[*] I thank Morgan Quigley for his direction, as well as the STAIR Project for the materials.

2 Range Sensor Design

2.1 Geometry and Raycasting

The relations among the fundamental lengths and angles described in Figure 1 are derived using a simple ray casting scheme. The sensor is extremely sensitive to construction parameters, so to eliminate systematic error, a sturdy construction is necessary. Note that the camera must be appropriately calibrated for radial distortion (using e.g., a standard checkerboard algorithm [2]), so that these figures make sense.

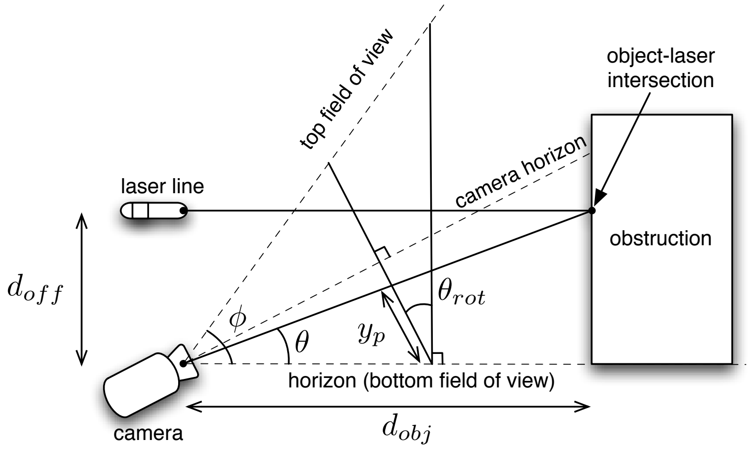


Figure 1: Single-Point simplified ranger geometry, side view. The camera is tilted such that the actual horizon corresponds to the bottom field of view limit. The object is d_{obj} away from the laser line generator, which is separated from the camera by d_{off}, a known distance. The camera sees the point of intersection y_{rc}/y_{res} of the way from the bottom of its viewing plane.

In order to triangulate the distance to given object, we use a standard line-plane intersection method. The camera is assumed to be at the origin of a 3-D spherical coordinate system. The plane of intersection is the one generated by the laser line, offset a distance d_{off} below the camera. After calibration [2] and laser line detection (see below), horizontal and vertical angles, θ_{horiz} and θ_{vert} respectively are inferred from the location of the laser line for each column of the display. Finally, the distance d_{obj} to the object is calculated by finding the intersection of a line and a plane.

2.2 Detecting the Laser Line

2.2.1 Color Filtering

Assume that the laser line has color represented by the RGB vector $v = (r, g, b)^T$, where r, g, and b are intensity values for the red, green, and blue components of the color respectively. Experimentally, typical values for the components of v are $(68, 127, 12)^T$ on a scale from 0 to 255. For every pixel with color w, we can project w onto v to see how "strongly" w corresponds to v using the dot product.

$$y = \frac{w \cdot v}{\|v\|_2} \hat{v} \tag{1}$$

where $y \in \mathbb{R}$ is the strength of the projection onto v and $\hat{v}$ is the unit vector in the direction of v. Colors strongly correlated with v will give large values of y and colors weakly correlated with v will give small values.

2.2.2 Fitting a Gaussian

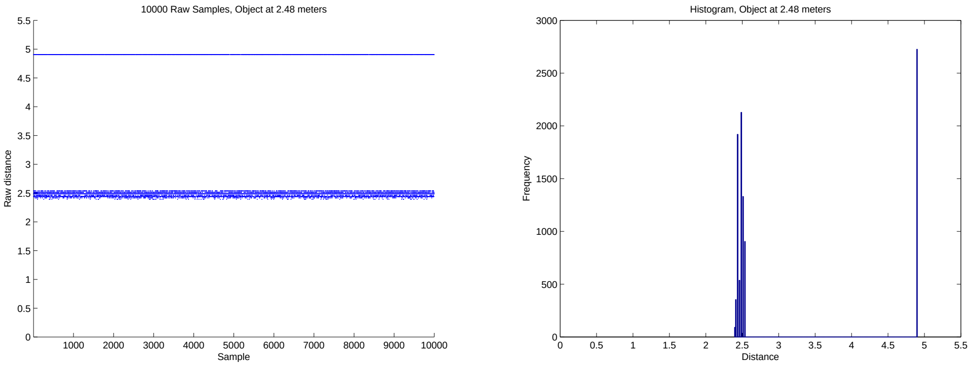
Since each column of pixels corresponds to one offset, and hence one measurement, we can find where the line is by a simple fit to a gaussian. To make this more concrete, suppose that in a given column, the projections onto v are the scalars $\{y^{(1)}, y^{(2)}, \ldots, y^{(r)}\}$, where r is the camera's vertical resolution in pixels. We can therefore treat the values $y^{(1)}, \ldots, y^{(r)}$ as entries in a histogram with r bins of size 1. The expected position of the laser line is simply the weighted mean

$$E[y] = \left(\sum_{i=1}^{r} y_i \right)^{-1} \sum_{i=1}^{r} i \cdot y_i. \tag{2}$$

Notice that $E[y]$ can be a fraction, which effectively allows a little extra resolution beyond the camera's pixel resolution. In practice, detection works better if the camera is taken slightly out of focus, which blurs out some noise at the expense of a wider distribution over the y values.

3 Sensor Calibration

3.1 Gathering Data



(a) Raw data

(b) Histogram

Figure 2: Raw data and histogram for 10,000 measurements of a box 2.48 meters from the sensor. About 63% of the measurements are centered around the true distance. A small number of values are scattered randomly through the entire measurement space, and the rest return z_{max}, when the laser line was not found.

A box is placed a known distance $d_{obj} \sim 1$m in the center third of the camera's field of view. Only those pixels which correspond to the object are considered. After the laser line is found and raycasting is used to extract the distance to the object, one should expect all the measurements in the point cloud to center around the actual distance $d_{obj} = \mu$ to the box. Label all the measurements $z^{(1)}, z^{(2)}, \ldots, z^{(m)}$, where in this case $m = 10,000$.

3.2 Error Model

We use a modified decomposition of the measurement density discussed in [3]. In particular, the probability for each measurement $z^{(i)}$, the probability $p(z^{(i)}|\mu)$ of that measurement given that the actual distance is μ, is split into three distributions

- **Correct measurement**: A single-variable gaussian distribution $p_{hit}(z^{(i)}|\mu;\sigma_{hit}^2)$ centered around the correct measurement. Its variance, σ_{hit}^2 is a parameter of the model that measures noise. We include in the model only those values corresponding to the measurement space that were not failures, i.e.,

$$p_{hit}(z^{(i)}|\mu;\sigma_{hit}^2) = \begin{cases} \eta\mathcal{N}(\mu,\sigma_{hit}^2) & \text{if } 0 \leq z^{(i)} < z_{max}, \\ 0 & \text{otherwise.} \end{cases}$$

where η is a normalization parameter equal to the cumulative density $\eta = \int_0^{z_{max}} \mathcal{N}(\mu,\sigma_{hit}^2)\,dz$

- **Random Measurement**: A uniform distribution $p_{rand}(z^{(i)}|\mu)$ corresponding to measurement noise through the entire measurement space, i.e.,

$$p_{rand}(z^{(i)}|\mu) = \begin{cases} 1/z_{max} & \text{if } 0 \leq z < z_{max}, \\ 0 & \text{otherwise.} \end{cases}$$

- **Failure**: A point-mass distribution $p_{max}(z^{(i)}|\mu) = 1\{z = z_{max}\}$ equal to 1 only if the measurement is a failure - e.g., the camera failed to detect a line.

The probabilities are mixed using the mixing parameters $\alpha_{hit}, \alpha_{rand}, \alpha_{max}$, where we constrain $\alpha_{hit} + \alpha_{rand} + \alpha_{max} = 1$

$$p(z^{(i)}|\mu;\sigma_{hit}^2,\alpha_{hit},\alpha_{rand},\alpha_{max}) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{rand} \\ \alpha_{max} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z^{(i)}|\mu;\sigma_{hit}^2) \\ p_{rand}(z^{(i)}|\mu) \\ p_{max}(z^{(i)}|\mu)\} \end{pmatrix} \tag{3}$$

3.3 Learning from Data to Infer Error Parameters

We formulate the learning problem of finding the probability density function by maximizing the total likelihood of the parameter $\theta = (\sigma_{hit}^2, \alpha_{hit}, \alpha_{rand}, \alpha_{max})^T$. That is,

$$\theta = \arg\max_\theta \prod_{i=1}^m p(z^{(i)}|\mu;\sigma_{hit}^2,\alpha_{hit},\alpha_{rand},\alpha_{max}) \tag{4}$$

subject to $\alpha_{hit} + \alpha_{rand} + \alpha_{max} = 1$, where m is the number of training examples, in this case 10,000.

3.4 Results

The Expectation Maximization (EM) algorithm [1] was used to solve the constrained optimization problem of Equation 4. The results are summarized in Figure 3. Having run the algorithm for several distances, the mixing parameters remain about the same for the same lighting conditions, while the gaussian width σ_{hit} increases with distance. This makes sense, because as the distance to the object increases, the offsets approach the resolution of the camera.

4 Conclusion

4.1 Advantages Over Ultrasonic Rangers

The horizontal resolution of the sensor is limited only by the horizontal resolution of the camera. That is, the distances provided by the sensor are much more granular than a single or even several sonar elements. In addition, the webcam sensor does not suffer from sonar cross-talk, and can make measurements as quickly as the camera can take pictures and hardware can process them. In ultrasonic

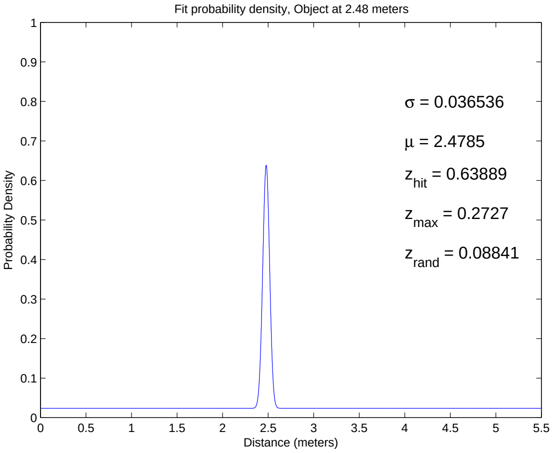


Figure 3: Plot of the probability density function for measurements of distances to a box 2.48 m away. Notice the uniform density spread throughout the entire space and the gaussian centered around the actual distance. The mixing parameters show that about 64% of the results come from the gaussian, and 27% belong to the point-mass failure function. The rest are random readings.

sensors, one needs to wait a long time in clock cycles after sending a ping before the ping returns (or fails to return). During this time, no more measurements can be made. The slow measurement-to-measurement time can be detrimental in a dynamic environment, as the robot can fail to see objects that move quickly in and out of view. Because the webcam sensor relies on light, the camera sees the laser and can calculate offsets almost instantaneously.

4.2 Disadvantages

The main problem with the sensor is the sensitivity to the laser line detection. The measurement z_{max} is returned whenever the laser line was not found, which happens quite often. Because the scheme used was based on simple color filtering, any environment containing colors similar to the laser's make it hard to find the line. In particular, the sensor is currently unsuitable in use in bright environments, e.g., outside in daylight. It is possible to make the laser line detection more robust by using knowledge about what it should look like in the camera. We know, for instance, that the line is horizontal or almost horizontal. One can train a classification algorithm to look for horizontal lines in an image, and then decide whether the horizontal line is part of the environment or the laser line. Indeed, this can be the subject of further research.

References

[1] Andrew Ng. Em algorithm. CS229 Class Notes, 2008.

[2] Hai Nguyen. Ros wiki: Camera calibration, 2008.

[3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*, chapter 6.3, pages 124–139. MIT Press, 2005.