Proc. of the IEEE Int. Conference on Robotics&Automation
Nice, France. 1992

An Iconic Position Estimator for a 2D Laser RangeFinder

Javier Gonzalez*, Anthony Stentz, Anibal Ollero*

Field Robotics Center, The Robotics Institute, Carnegie Mellon University,
Pittsburgh, PA, 15213 USA

Abstract

Position determination for a mobile robot is an important part of autonomous navigation. In many cases, dead reckoning is insufficient because it leads to large inaccuracies over time. Beacon- and landmark-based estimators require the emplacement of beacons and the presence of natural or man-made structure respectively in the environment. In this paper we present a new algorithm for efficiently computing accurate position estimates based on a radially-scanning laser rangefinder that requires minimal structure in the environment. The algorithm employs a connected set of short line segments to approximate the shape of any environment and can easily be constructed by the rangefinder itself. We describe techniques for efficiently managing the environment map, matching the sensor data to the map, and computing the robot's position. We present accuracy and runtime results for our implementation.

1. Introduction

Determining the location of a robot relative to an absolute coordinate frame is one of the most important issues in the autonomous navigation problem. In a two dimensional space, the location of a mobile robot can be represented by a triplet (t_x, t_y, θ) known as the robot pose. A mobile coordinate system (Robot Frame) attached to the robot can be considered such that (t_x, t_y) represents the translation (position) of the Robot Frame with respect to an absolute coordinate system (World Frame) and θ represents its orientation (heading) (Fig. 1).

To estimate the pose (t_x ,t_y, θ) of a mobile robot equipped with a range sensor the matching between the range data and model data is required. This can be accomplished by two different approaches: feature-based and iconic. In the feature-based method, a set of features are extracted from the sensed data (such as line segments, corners, etc.) and then matched against the corresponding features in the model. Shaffer et al. [9] using a laser scanner rangefinder and Crowley [10] and Drumheller [11] using range from a rotating sonar, proposed a feature-based approach for a 2D environment. In contrast, the iconic method works directly on the raw sensed data, minimizing the discrepancy between it and the model. Hebert et al [12] formulated an iconic method to compare two elevation maps acquired from a 3D laser rangefinder. Moravec and Elfes proposed a technique to match two maps represented by *occupancy grids* [4]. Finally, Cox [13] used an infrared laser rangefinder to get a 2D radial representation of the environment which is matched against a line segment map.

In this paper, we present a new iconic approach for estimating the pose of a mobile robot equipped with a radial laser rangefinder. Unlike prior approaches, our method can be used in environments with only a minimal amount of structrure, provided enough is present to disambiguate the robot´s pose. Our map consists of a possibly large number of short line segments, perhaps constructed by the rangefinder itself, to approximately represent any environment shape. This representation introduces problems in map indexing and error minimization which are addressed to insure that accurate estimates can be computed quickly.

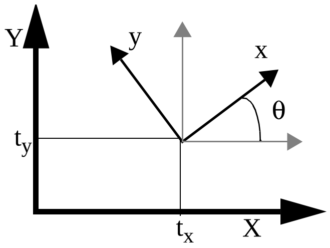

FIGURE 1. World Frame and Robot Frame

2. Iconic Position Estimation

The position estimation problem consists of two parts: sensor to map data correspondence and error minimiza-

*.- On leave from University of Malaga, Spain.

tion. Correspondence is the task of determining which map data point gave rise to each sensor data point. Once the correspondence is computed, error minimization is the task of computing a robot pose that minimizes the error (e.g., distance) between the actual location of each map data point and the sensor's estimate of its location.

In this work we are concerned with scanned data taken from two-dimensional world maps. A convenient way to describe these maps is by means of a set L= {L_1, L_2, ... ,L_m} where L_j represents the line segment between the "left" point (a_j^l, b_j^l) and the "right" point (a_j^r, b_j^r) in the World Frame (see Fig. 2). This line segment lies on the line given in an implicit normalized form by:

$$A_jX + B_jY + C_j = 0 \tag{1}$$

The sensed data consists of range points taken from a radial laser scanner.

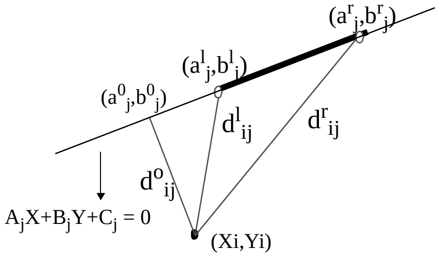

FIGURE 2. Different distances to consider for each line segment.

The correspondence problem is formulated as determining which line segment L_j from the model L gave rise to the image point $\mathbf{p}_i$= $(x_i,y_i)^T$. A reasonable heuristic for determining correspondence is the minimun Euclidean distance between model and sensor data. Thus, the distances between the sensed point $\mathbf{P}_i$ = $(X_i,Y_i)^T$ and the line segment L_j are defined as follows:

d_{ij} = d^0 if (a^0_j,b^0_j) is an element of L_j
d_{ij} = min(d^r, d^l) otherwise

where (see Fig. 2):

$$d^o_{ij} = \left|A_jX_i + B_jY_i + C_j\right| \tag{2}$$

$$\text{where } (A_j^2 + B_j^2)^{1/2} = 1$$

$$d^r_{ij} = ((X_i - a^r_j)^2 + (Y_i - b^r_j)^2)^{1/2} \tag{3}$$

$$d^l_{ij} = ((X_i - a^l_j)^2 + (Y_i - b^l_j)^2)^{1/2} \tag{4}$$

and

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{5}$$

Equation (5) defines the transformation between a point $\mathbf{P}_i$ in the World Frame and a point $\mathbf{p}_i$ in the Robot Frame given by (t_x,t_y,θ) (see Fig.1). Once the line segments from the map and scanned points are represented in the same coordinate system it is possible to search for the segment/ point correspondence pairs.

The iconic position estimation problem consists of the computation of (t_x,t_y,θ) that minimizes the sum of square distances between the segment and range point of every correspondence pair. To establish such a correspondences all of the segments from the map could be checked against every range point . In a sensor such as Cyclone Range Finder, which will be described later, one thousand scanned points would have to be matched against a model of hundreds of line segments (which could be built by the robot itself). To avoid this extremely expensive procedure, we propose a two-tier map representation:

1.- *Cell map*: array of grid cells in which every cell is labeled either occupied, if it contains at least one line segment, or empty, if it contains no segments. Elfes and Moravec used a similar approach for sonar navigation in their *occupancy grid* [4].

2.-*Line map*: collection of segments inside each of the occupied cells considered for correspondence.

The correspondence of sensed points to the model segments is accomplished in two steps. First, a set of cells is selected for each of the scanned points. Second, only those segments inside these cells are checked for correspondence. By using this representation, the number of segments to be checked decreases considerably, drastically reducing the matching time [5].

The grid size must be selected according to the characteristics of the particular application. One cell for the whole map is inefficient because it requires all of the line segments to be examined for each sensed point (no improvement at all). A very large number of cells is also inefficient because it requires a large number of empty cells to be checked. We have determined that the appropriate size of the grid is a function of a variety of parameters (number of line segments in the model, type of environment, initial error, etc.) and therefore an empirical method is proposed for choosing it.

3. Cell selection

After the scanned points have been transformed to the World Frame, a set of occupied cells must be selected for each of them (Fig. 3). Due to errors in both dead reckoning and the sensor, in a significant number of cases, the points

$\mathbf{P}_i$ are located in empty cells. We analyze these errors in more detail below.

3.1. Dead Reckoning errors

Dead reckoning is intrinsically vulnerable to bad calibration, imperfect wheel contact, upsetting events, etc. Thus, a confidence region bounding the actual location of the robot is used. This region is assumed to be a circle of radius δ_r proportional to the traversed distance. This uncertainty in the robot position propagates in such a way that an identical uncertainty region centered at the sensed point can be considered (Fig. 3a).

In a similar way, the heading error is assumed to be bounded by $\mp\varepsilon_r$ degrees. This error is also considered to be proportional to the traversed distance. Notice that the effect of this error over the uncertainty region for the sensed point depends on the range (Fig. 3b).

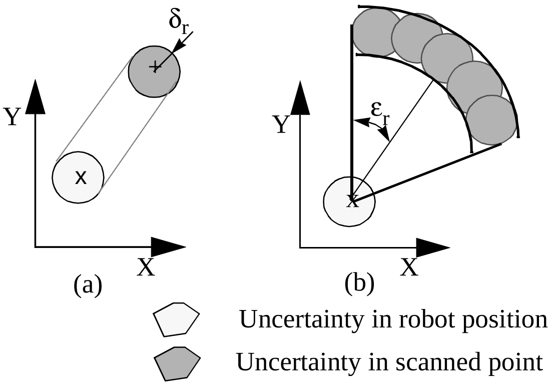

FIGURE 3. Uncertainties in the sensed data due to Dead Reckoning error pose. (a) Uncertainty region caused by position error.(b) Uncertainty region caused by position and orientation error.

3.2. Sensor errors

Sensor errors arise for the following reasons: the range provided by the laser rangefinder is noisy as well as truncated by the resolution of the sensor, and the angular position given by the decoder has some inaccuracy. Thus, the two errors considered are *range error* and *orientation error.* Although they can be modeled as a gaussian distribution [1], here both of them are modeled as bounded errors, as were dead reckoning errors. Their maximum and minimum values define a new region of uncertainty to be added to the one arising from the dead reckoning errors. Figure 4a shows a region defined by two errors parameters δ_s and $\varepsilon_s$whose values are obtained from the sensor calibration experiments [7]. This region does not increase with the distance traversed by the robot. On the other hand, although it depends on the range value, it is not as significant as the dead reckoning error ($\varepsilon_s << \varepsilon_r$).

Figure 4b shows the final region after considering both dead reckoning and sensor errors. Notice that the sensed point location is not necessarily along the scanning ray but is inside the uncertainty region.

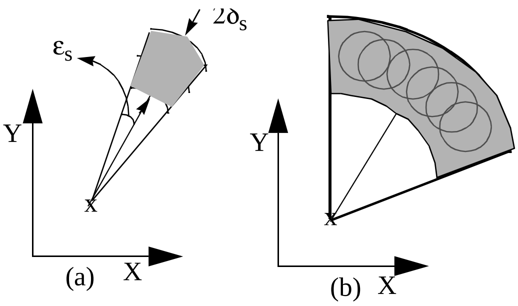

FIGURE 4. (a) Uncertainty in the sensed data due to the sensor errors. (b) Uncertainty region caused by Dead Reckoning and Sensor errors.

3.3. Cells selection algorithm

The algorithm to select the cells takes into account the above mentioned uncertainty regions. Each time the cell which includes the scanned point is labeled empty a search for a nearby occupied cell is performed (Fig. 5). The searching area is selected to be coincident with the uncertainty region given by the sensor and dead reckoning errors (Fig 4b).

If no a priori information is available, the matcher assumes the closest occupied cells are the most likely to contain the corresponding model segment. A distance function based on 8-connectivity is used. The search radiates out from the cell containing the sensed point until the cell containing the nearest line segment is found. For all the cells located at the same distance, only those both occupied and inside the uncertainty region are examined for the closest line segment within them (Fig 5).

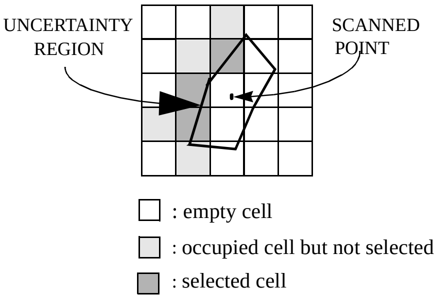

FIGURE 5. Cells to be considered when the original cell is empty.

To make the algorithm robust against outliers, incompleteness of the model, presence of extraneous objects, etc., a progression of increasingly better position estimates is computed (see figure 6). The uncertainty region is reduced along the progression. This approach is based on the fact that the uncertainty due to the error in sensor location decreases as the position estimate improves. However, the uncertainty region due to the sensor errors does not vary. In practice, this is accomplished by weighting the parameters δ_r and ε_r. between 0 and 1.

4. Segment correspondence

To determine which line segment inside the assigned cells matches the scanned point, a minimum distance criterion is used. This assumption is valid as long as the displacement between sensed data and model is small enough. This assumption limits the allowable distance traversed by the robot between consecutive position estimates. However, since after each iteration the point/line-segment pairs are updated, the limitation can be relaxed somewhat (Fig. 6).

Given a scanned point $\mathbf{P}_i = (X_i, Y_i)$, three different distances for each line segment are computed (Fig. 2). They are given by equations 2,3 and 4. The smallest distance to the line segments inside the selected cells determines which line segment l_j is be matched to $\mathbf{P}_i$.

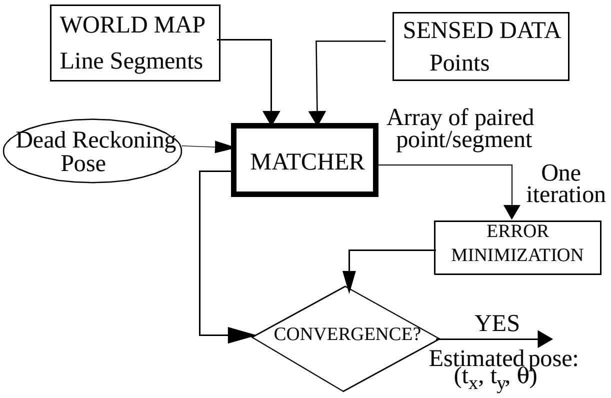

FIGURE 6. Block diagram of the iconic position estimator

5. Minimization

After the matched pairs have been determined, the estimate is computed by minimizing the following:

$$min\,(e^T e) = min\left(\sum_{1=1}^{n} e_i^2\right) \qquad (6)$$

where $e_i = e_i(t_x, t_y, \theta)$ is the distance equation computed for $\mathbf{P}_i$.

Although the rotation θ makes this optimization problem non-linear, a closed-form solution exists. The Schonemann approach treats the rotation elements as unknowns and applies Lagrange multipliers to force the rotation matrix to be orthogonal [2]. However, we have opted for a iterative algorithm (Gauss-Newton) to support the future modelling of gaussian uncertainty in the sensor and robot data. Such modelling requires nonscalar weights on the error. No closed-form solution exists for the minimization. In this method the equation to be solved is:

$$\boldsymbol{e} + \boldsymbol{J}\ \boldsymbol{d} = 0 \qquad (7)$$

where **e** is the error vector, **d** is the difference vector between the transformation parameters on successive iterations, and **J** is the Jacobian:

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial e_1}{\partial t_x} & \dfrac{\partial e_1}{\partial t_y} & \dfrac{\partial e_1}{\partial \theta} \\ \ldots & \ldots & \ldots \\ \dfrac{\partial e_n}{\partial t_x} & \dfrac{\partial e_n}{\partial t_y} & \dfrac{\partial e_n}{\partial \theta} \end{bmatrix} \qquad (8)$$

Notice that Equation (7) is overdetermined for n>3. In this case we use the pseudoinverse of the Jacobian to find a least square fit of **d**:

$$\boldsymbol{d} = -(\boldsymbol{J}^T J)^{-1} J^T e \qquad (9)$$

Equation (9) is solved iteratively for the displacement vector **d** until the absolute value of its elements is less than some tolerance. On each iteration, the correspondence between sensor and model data is recomputed to reduce the effects of outliers and mismatches. We have empirically determined that iterating more than once between correspondence updates yields no additional accuracy in the final estimate, thus our approach is functionally equivalent to the closed-form solution with updating.

6. Application

In this section, we describe the mobile robot and the sensor used in this application as well as the implementation and results.

6.1. The Locomotion Emulator

The Locomotion Emulator (LE) is a mobile robot that was developed at the CMU Field Robotics Center (FRC) as a testbed for development of mobile robotic systems (Fig. 7). It is a powerful all-wheel steer, all-wheel drive base with a rotating payload platform. A more complete description can be found in [3].

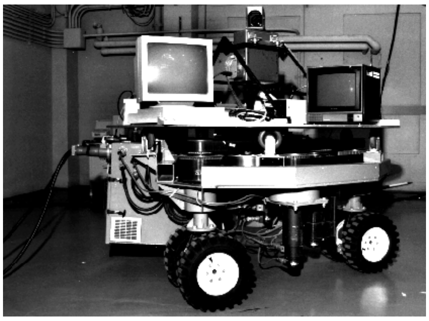

FIGURE 7. The Locomotion Emulator

6.2. Cyclone

The Cyclone laser range scanner (Fig. 8a) was also developed at the FRC to acquire fast, precise scans of range data over long distances (up to 50m)[6]. The sensor consists of a pulsed Gallium Arsenide infrared laser transmitter/receiver pair, aimed vertically upward. A mirror in the upper part of the scanner rotates about the vertical axis and deflects the laser beam so that it emerges parallel to the ground, creating a two dimensional map of 360 degrees field of view. The resolution of the range measurements is set to be 10cm and the accuracy is +20cm [7]. The angular resolution depends upon the resolution of the encoder that is used on the tower motor which is currently programmed to acquire 1000 range readings per revolution.

6.3. Experimental results

The iconic position estimation algorithm presented in this paper was tested at the highbay area of the FRC. The corridor is about 6m wide and 20m long (Fig.8). The solid line segments denote walls which were constructed from wood partitions. We picked this configuration because its simplicity and reliability in being surveyed. The dotted line represents the path that the LE was instructed to follow. It consists of a symmetrical trajectory 19m long. The LE, initially positioned at the beginning of the path, was moved by steps of 1m. At each of these positions, the position estimator was executed and the robot pose was surveyed using a theodolite. Figure 8b shows the sensed data taken by the Cyclone at the 7th step. Notice that a considerable number of points from the scanner corresponds to objects that are not included in the model of figure 8c.

The estimator was programed to use two different representations of the model. In the first one, the model was represented by the 8 long line segments shown in figure 8c. In the second, each of these line segments was split into a number of small segments 10cm long, providing a model with almost 400 line segments. The parameter values used were: δ_r = 5cm and ε_r = 5deg for the LE (5% of the step size) and δ_s = 10cm and ε_s = 0.7deg for the Cyclone. The grid size was 0.6x0.6m^2.

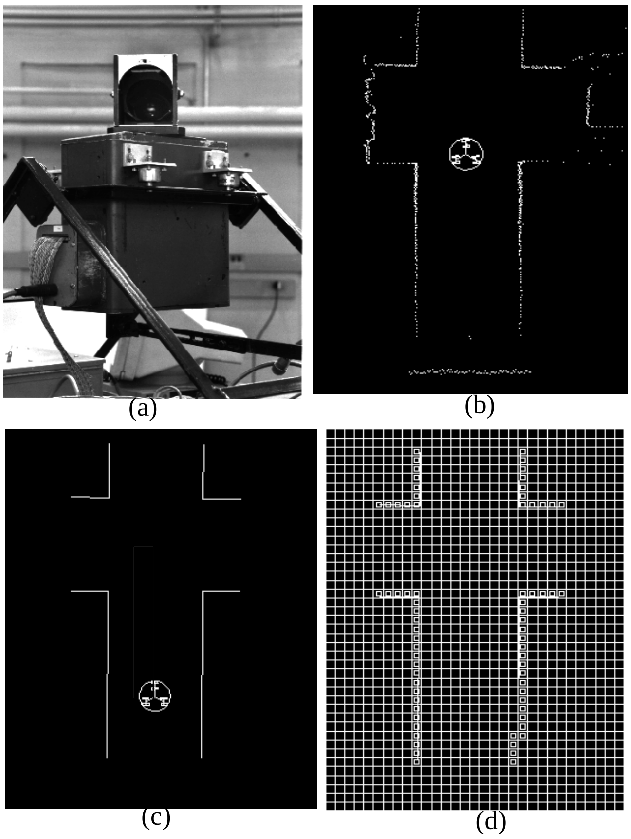

FIGURE 8. (a) The Cyclone laser rangefinder. (b) Range scan provided by the Cyclone. The circular icon represents the LE at the position where the scan was taken from. (c) World model representation. (d) Map representation.

As expected the computed error (surveyed minus estimate) for the two representations was exactly the same at the 20 positions along the path (Fig. 9). The maximum position error was 3.6cm, and the average position error was 1.99cm. The maximum heading error was 1.8deg and the average was 0.73deg. These results are significant given the resolution (10cm) and accuracy (20cm) of the scanner.

Another important result is the run times. The estimator was run on a Sun Sparc Station 1 with a math coprocessor. For the 8 line segment representation the approximate run times were 0.37sec for the preprocessing (computation of the *cell map*), 0.27sec for the minimization and 1.76 for the segment correspondence, giving a total cycle time of 2sec. For the 400 line segment representation, run times were 12.9sec for the preprocessing, 0.29sec for the minimization and 3.22 for the segment correspondence, giving 3.5sec of total cycle time. Note that by multiplying the number of line segments by a factor of 50, the preprocessing time increases considerably, however the matching time is increased only by a factor of 1.75.

In the event that the uncertainty regions for the sensed points can be approximated by circles centered on the points, the segment correspondence can be computed rapidly using a numerical Voronoi diagram. This approximation worked well for our highbay experiments [8].

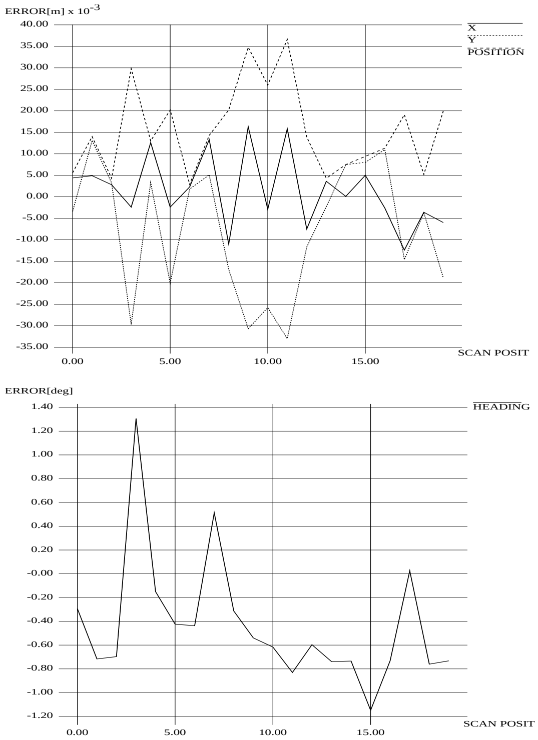

FIGURE 9. Computed errors for the 20 positions along the path.

7. Conclusions

In this paper a two dimensional iconic based approach for position estimation was presented. By considering two resolution levels in the map, a two-stage method is proposed to solve the point/line-segment correspondence. Furthermore, the uncertainty due to errors in both dead reckoning pose and sensed data are considered in order to bound the searching area. This approach drastically reduces the computation time when the map is given by a high number of line segments (e.g. map built by the robot itself). This algorithm was implemented and tested using a 2D radial laser scanner mounted on a omnidirectional robot, showing for the first time an explicit quantification of the accuracy of an iconic position estimator. The estimator has shown to be robust to incompleteness of the model and spurious data, and provides a highly accurate estimate of the robot position and orientation for many-line environments.

Acknowledgements

We wish to thank to Gary Shaffer for his collaboration in the experiment and contribution to the development of the programs, Kerien Fitzpatrick for facilitating the testing, In So Kweon for suggesting the use of small line segments, and Sergio Sedas and Martial Hebert for their valuable comments and discussions.

References

[1] C. M. Wang, “Location estimation and uncertainty analysis for mobile robots”, in Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1230-1235, 1988.

[2] P. H. Schonemann and R. M. Carroll. “Fitting one matrix to another under choice of a central dilation and a rigid motion”, Psychometrika, vol. 35,pp. 245-255, June 1970.

[3] K. W. Fitzpatrick and J.L. Ladd. “Locomotion Emulator: A testbed for navigation research”. In 1989 World Conference on Robotics Research: The Next Five Years and Beyond, May 1989.

[4] H. P. Moravec and A. Elfes. “High resolution maps from wide-angle sonar”. In Proc. IEEE International Conference on Robotics and Automation”, March 1985.

[5] J. Gonzalez, A. Stenz and A. Ollero. “An Iconic Position Estimator for a 2D Laser RangeFinder”. CMU Robotics Institute Technical Report CMU-RI-TR-92-04, 1992.

[6] S. Singh, J. West, “Cyclone: A Laser Rangefinder for Mobile Robot Navigation”, CMU Robotics Institute Technical Report, CMU-RI-TR-91-18, August 1991

[7] S. Sedas and J. Gonzalez. “Analytical and Experimental Characterization of a Radial Laser RangeFinder”. CMU Robotics Institute Technical Report CMU-RI-TR-92 1992.

[8] G. Shaffer and A. Stenz. “Automated Surveying of Mines Using a Scanning Laser Rangefinder”. to be submitted to the SME Annual Meeting Symposium on Robotics and Automation, February, 1993.

[9] G. Shaffer, A. Stentz, W. Whittaker, K. Fitzpatrick. “Position Estimator for Underground Mine Equipment”. In Proc. 10th WVU International Mining Electrotechnology Conference, Morgantown, W. VA, July 1990.

[10] J. L. Crowley. “Navigation for an intelligent mobile robot”. IEEE Journal of Robotics and Automation, vol. RA-1, no. 1, March 1985.

[11] M. Drumheller. “Mobile robot localization using sonar”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 2, March 1987.

[12] M. Hebert, T. Kanade, I. Kweon. “3-D Vision Techniques for Autonomous Vehicles”. Technical Report CMU-RI-TR-88-12, 1988.

[13] I. C. Cox. “Blanche-An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle”. IEEE Transactions on Robotics and Automation, Vol. 7, no. 2, April 1991.