

# A Mobile Robot Iconic Position Estimator using a Radial Laser Scanner\*

Javier Gonzalez<sup>1</sup>, Anthony Stentz<sup>2</sup> and Anibal Ollero<sup>3</sup>

1.- Dpto. Ingenieria de Sistemas y Automática. Universidad de Malaga. Plaza El Ejido s/n. 29013 Malaga, SPAIN

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

3.- Dpto. Ingenieria de Sistemas y Automática. Universidad de Sevilla. Avda. Reina Mercedes s/n. 41012 Sevilla, SPAIN

## 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 does not require 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.*

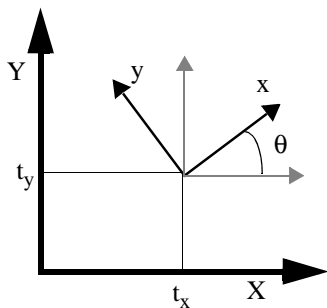
**Key words:** Mobile robots, laser rangefinder, position estimation, iconic methods.

\* A portion of this paper was presented at the IEEE Int. Conference on Robotics and Automation, Nize, France. May 1992.

This work was carried out during the stay of the first and third authors in the Field Robotics Center, The Robotics Institute, Carnegie Mellon University, under grant of the Spanish Government. The research was sponsored in part by the United States Bureau of Mines, contract #358021.

## 1.0 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  $(t_x, t_y, \theta)$  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  $\theta$  represents its orientation (heading) (Figure. 1).



**FIGURE 1.** World Frame and Robot Frame

The simplest approach in estimating the robot pose (position and orientation) is provided by the dead reckoning system, in which both position and orientation are given by counting the revolutions of the wheels. The advantage of dead reckoning is that it is simple and inexpensive. However, assuming even very small errors, as the robot moves from one place to another, the uncertainty about its pose grows.

In order to reduce this uncertainty registration with the environment is required. Different approaches have been proposed. A significant number use landmark recognition, either extracting relevant natural features (corners, objects, etc.) using a camera [8, 12] or identifying prelocated beacons using both a camera [9] and an optical scanner [19]. The major disadvantages of the first are the poor accuracy as well as the need for a suitable environment. Major drawbacks of beacon-based navigation are that the beacons must be placed within range and must be appropriately configured in the robot work space.

An alternate approach consists of the comparison of dense range data to model data. Since both accuracy and robustness in a position estimator depend on the amount of data used in the comparison, this approach presents some important advantages over the two above mentioned.

The matching problem between two sets of data can be formulated in two different ways: 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.* [1] using a laser scanner rangefinder and Crowley [5] and Drumheller [14] using range from a rotating sonar, proposed a feature-based approach for a 2D environment. Krotkov [11] used a single CCD camera to extract angles to vertical edges within the scene and then determine the robot pose by establishing their correspondence in a 2D map.

In contrast, the iconic method works directly on the raw sensed data, minimizing the discrepancy between it and the model. Hebert *et al* [13] formulated an iconic method to compare two elevation maps acquired from a 3D laser rangefinder. Moravec and Elfes [4, 17] proposed a technique to match two maps represented by *occupancy grids*. The occupancy grid models the environment using a spatial grid of cells, where each cell has a probability value assigned to it representing the likelihood that the cell is occupied. Finally, Cox [2] 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 any type of environment, both structured and unstructured. Our map consists of a possibly large number of short line segments, perhaps constructed by the rangefinder itself, to approximately represent any environment shape [7]. This representation introduces problems in map indexing and error minimization which are addressed to insure that accurate estimates can be computed quickly.

## 2.0 Problem statement

The position estimation problem consists of two parts: sensor to map data correspondence and error minimization. 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 section we formalize these two parts.

Let A be a set of sensed data points taken from a certain robot position, and B be a set of either sensed data extracted from a different position or a set of data describing a world map. The matching problem for position estimation can be stated as how to determine the best mapping

of A onto B. In others words, how to compute the transformation T which minimizes an appropriate error function:

$$E = \Phi(T(A), B) \tag{EQ 1}$$

Consider a world map M described by a set of data  $M = \{M_1, M_2, \dots, M_m\}$  expressed in an absolute coordinate system (World Frame) and sensed data  $S = \{S_1, S_2, \dots, S_n\}$ , where  $S_i$  is expressed in a local coordinate system attached to the mobile robot (Robot Frame). Then, the robot position estimation consists of solving the minimization problem:

$$\min_T E = \Phi(T(S), M) \tag{EQ 2}$$

In this paper we are concerned with scanned data taken from a two-dimensional world maps. A convenient way to describe these maps is by means of a set  $M = L = \{L_1, L_2, \dots, L_m\}$  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 Figure 2). This line segment lies on the line given in an implicit normalized form by:

$$A_j X + B_j Y + C_j = 0 \tag{EQ 3}$$

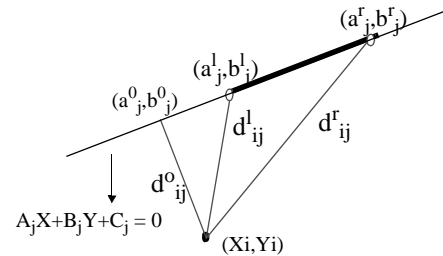


FIGURE 2. Different distances to consider for each line segment.

In the iconic approach presented in this paper, the sensed data will consist of points taken from a radial laser scanner. These points will be referred to as either sensed points or scanned points.

Now, let  $\mathfrak{R}$  be the binary relation between the finite sets S and L

$$\mathfrak{R} \subset S \times L \tag{EQ 4}$$

This relation can be represented by means of the nxm correspondence matrix R which elements are:

$$r_{ij} = 1 \text{ if the point } S_i \text{ is a measurement of segment } L_j$$

$$r_{ij} = 0 \text{ otherwise}$$

Determining this matrix  $R$  is equivalent to solving the correspondence problem formulated as determining which line segment  $L_j$  from the model  $L$  gave rise to the image point  $S_i \equiv \mathbf{p}_i = (x_i, y_i)^T$ . Obviously, because one point can correspond to only one line segment, each of the rows of  $R$  will contain just one value "1".

A reasonable criterion for constructing the matrix  $R$  is the minimum euclidean distance (this will be emphasized in the next section). Thus, let  $D$  be the  $n \times m$  distance matrix whose elements  $d_{ij}$  are the distances between the sensed point  $\mathbf{P}_i = (X_i, Y_i)^T$  and the line segment  $L_j$  defined as follows:

$$d_{ij} = d^0 \text{ if } (a_j^0, b_j^0) \text{ is an element of } L_j$$

$$d_{ij} = \min(d^r, d^l) \text{ otherwise}$$

where (see Figure 2):

$$d_{ij}^0 = |A_j X_i + B_j Y_i + C_j| \quad (\text{EQ 5})$$

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

$$d_{ij}^r = ((X_i - a_j^r)^2 + (Y_i - b_j^r)^2)^{1/2} \quad (\text{EQ 6})$$

$$d_{ij}^l = ((X_i - a_j^l)^2 + (Y_i - b_j^l)^2)^{1/2} \quad (\text{EQ 7})$$

and

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (\text{EQ 8})$$

Equation 8 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  $(t_x, t_y, \theta)$  (see Figure1).

From the matrices  $D$  and  $R$  it is possible to obtain the error vector between the sensed points and the corresponding line segments:

$$\mathbf{e} = (e_1 \dots e_n)^T = \text{diag}(D \times R^T) \quad (\text{EQ 9})$$

It should be noted that  $\mathbf{e}$  is a function of the transformation defined in Equation 8. Then, the parameter to minimize can be expressed as:

$$I = \sum_{i=1}^n e_i(t_x, t_y, \theta) \quad (\text{EQ 10})$$

The iconic position estimation problem consists of the computation of  $(t_x, t_y, \theta)$  to solve:

$$\min_{(t_x, t_y, \theta)} I \quad (\text{EQ 11})$$

In practice, the matrix  $R$  is derived from  $D$  by:

$$r_{ik} = 1 \text{ if } d_{ik} < d_{ij} \text{ for all } j$$

$$r_{ik} = 0 \text{ otherwise}$$

Moreover,  $R$  is a "dynamic matrix" in the sense that it is updated after each iteration of the optimization algorithm to solve Equation 11. A similar approach was used in [2].

Although the approach presented here to compute  $R$  is deterministic, this formulation enables one to treat the correspondence problem in a probabilistic way. In this formulation, the elements  $r_{ij}$  of the matrix  $R$  would no longer be either 1 or 0 but would express the probability that the point  $\mathbf{P}_i$  corresponds to the segment line  $L_j$ .

One of the main issues when solving the matching problem is the high cost involved in the computation of the  $n \times m$  matrix  $D$ . In section 3 we present a more efficient approach to determine  $D \times R^T$  avoiding the entire computation of the matrix  $D$ .

### 3.0 Iconic Position Estimation

In this section, the two subproblems (correspondence and minimization) of the iconic position estimation problem are described. Because the world map is represented in an absolute coordinate system (World Frame) and the sensed data is represented in a mobile coordinate system attached to the robot (Robot Frame) a common representation for both must be considered. This can be accomplished by using Equation 8 with a first estimate of the robot pose  $(t_x, t_y, \theta)$  provided by the dead reckoning system (Figure 4).

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.

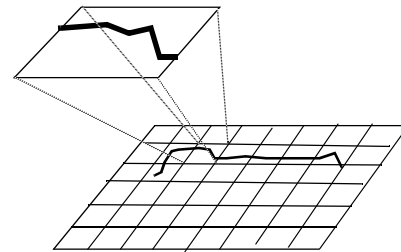


FIGURE 3. Two levels of resolution in the map

To establish such a correspondences all of the segments from the map could be checked against every range point in order to compute the matrix  $D$  ( $O(n \times m)$ ). 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 (Figure 3):

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 (see Figure 3). Elfes and Moravec used a similar approach for sonar navigation in their *occupancy grid* [4, 17].

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

Thus, 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 (Figure 4). By using this representation, the number of segments to be checked decreases considerably, drastically reducing the matching time.

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 segment 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.

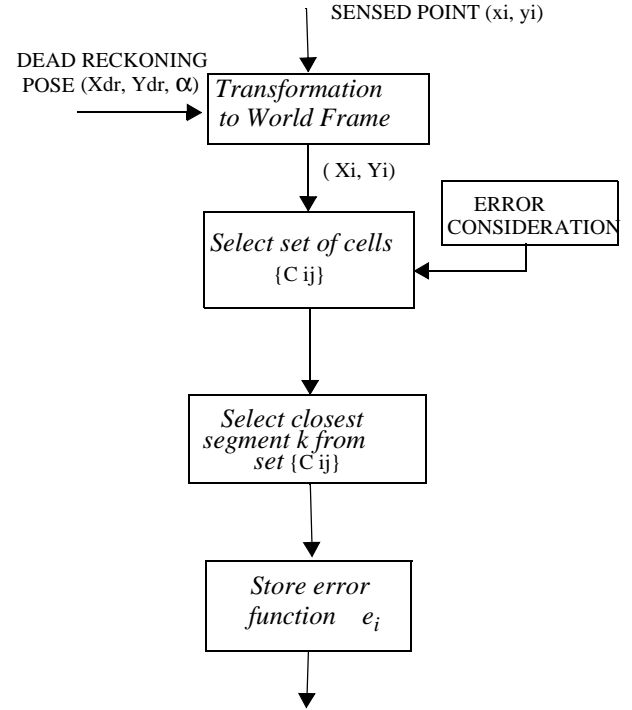


FIGURE 4. Matcher structure

### 3.1 Hierarchical iconic matching

In this section, we formulate iconic matching for the two-tier map. Considering that candidate cells must be selected within which to search for the corresponding line segment, let  $U$  be a binary  $n \times c$  matrix defined as follows:

$$u_{ik} = 1 \text{ if cell } k \text{ is selected for the scanned point } i$$

$$u_{ik} = 0 \text{ otherwise}$$

Let  $V$  be a binary  $c \times m$  matrix defined as follows:

$$v_{kj} = 1 \text{ if the } l_j \text{ segment lies in cell } k (l_j \cap c_k \neq \emptyset)$$

$$v_{kj} = 0 \text{ otherwise}$$

Consider the  $n \times m$  matrix  $W'$  given by the product  $U \times V$ . The elements  $w'_{ij}$  are given by:

$$w'_{ij} = \sum_{k=1}^c u_{ik} v_{kj}$$

Consider the matrix  $W$  defined as follows:

$$w_{ij} = 1 \text{ if } w'_{ij} > 0$$

$$w_{ij} = 0 \text{ if } w'_{ij} = 0$$

Thus, a generic element  $w_{ij}$  from the matrix  $W$  indicates whether the segment  $l_j$  lies in any of the cells selected for the point  $P_i$  or not, and therefore if the distance between  $l_j$  and  $P_i$  must be computed or not.

For a given point  $P_i$ , assuming both  $U$  and  $V$  have been obtained, the number of distances to be computed reduces from  $m$  to  $m'_i$ , where:

$$m'_i = \sum_{i=1}^{\dots} w_{ij} \quad (\text{EQ 12})$$

Although the matrix  $U$  must be calculated for each computed estimate, the matrix  $V$  is computed just once, when the map is built.

Notice that in the extreme case given above in which the number of cells considered for the *cell map* ( $c$ ) is very high ( $c \gg m$ ), the computation of matrix  $U$  ( $n \times c$ ) could become more expensive than the straightforward approach of computing matrix  $D$  ( $O(n \times m)$ ). In the following section the methodology to obtain the matrix  $U$  is presented.

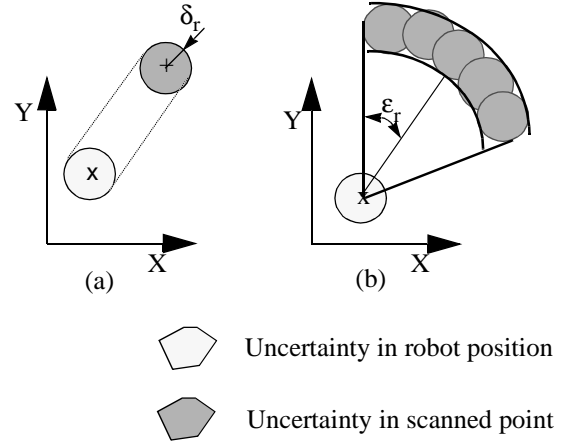
### 3.2 Cell selection

After the scanned points have been transformed to the World Frame, a set of occupied cells must be selected for each of them (Figure 4). Due to errors in both dead reckoning and sensor, in a significant number of cases, the points  $P_i$  are located in empty cells. We analyze these errors in more detail below.

#### 3.2.1 Dead Reckoning errors

Dead reckoning is intrinsically vulnerable to bad calibration, imperfect wheel contact, upsetting events, etc. Thus, a bounded confidence region in which the actual location is likely to be found is used. This region is assumed to be a circle of radius  $\delta_r$  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 (Figure 5a).

In a similar way, the heading error is assumed to be bounded by  $\pm \epsilon_r$  degrees. This error is also considered to be proportional to the traversed distance. Notice that the effect of this error over the uncertainty region depends on the range, the bigger the range the larger the uncertainty region (Figure 5b).



**FIGURE 5.** Uncertainties in the sensed data due to Dead Reckoning error pose. (A) Uncertainty region caused by the position error. (B) Dead Reckoning position an orientation error simultaneously.

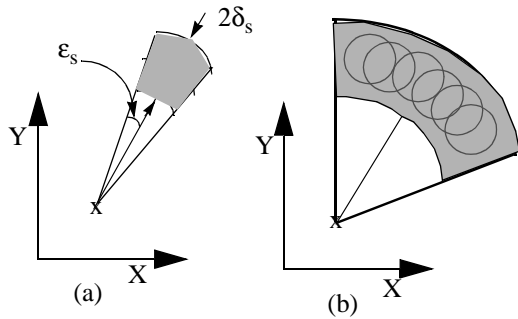
#### 3.2.2 Sensor errors

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.
- The angular position given by the decoder has some inaccuracy.

Thus, the two errors to be considered are *range error* and *orientation error*. Although they can be modeled as a gaussian distribution, 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 6a shows a region defined by two errors parameters  $\delta_s$  and  $\epsilon_s$  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 ( $\epsilon_s \ll \epsilon_r$ ).

Figure 6b 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 inside an uncertainty region.

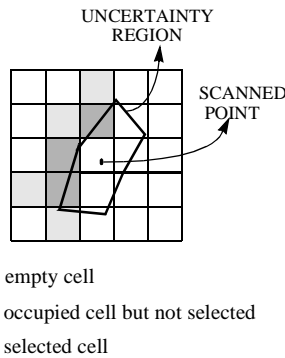


**FIGURE 6.** (a) Uncertainty in the sensed data due to the sensor errors. (b) Uncertainty region caused by Dead Reckoning and Sensor errors.

### 3.2.3 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 (Figure 7). The searching area is selected to be coincident with the uncertainty region given by the sensor and dead reckoning errors (Figure 6b).

If no a priori information is available, the matcher assumes the closest occupied cells are the best ones in terms of confidence. A distance function based on 8-connectivity has been 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 (Figure 7).

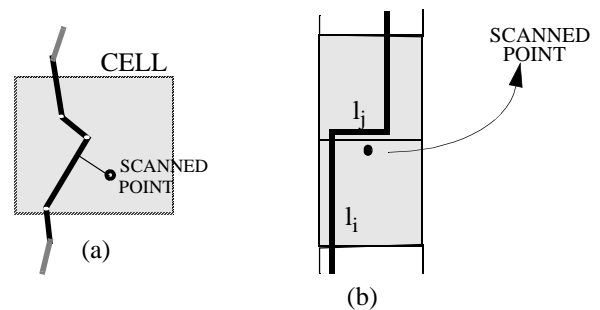


**FIGURE 7.** Cells to be considered when the original cell is empty.

There is no guarantee that every time the scanned point lies in an occupied cell, the closest line segment be inside it. Figure 8b shows a simple example of this case. The scanned point should be assigned to the segment labeled “ $l_j$ ” rather than segment “ $l_i$ ”. The matcher rectifies this problem by including the neighbor occupied cells as candidates to search for the nearest line segment.

The location of previous scanned points with respect to their respective target line segments could be used to decide which cells are more likely to contain the target line segment. This approach has not been implemented so far, but it could have a considerable impact over the efficiency of the algorithm.

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 Section 3.4). 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  $\delta_r$  and  $\epsilon_r$  between 0 and 1.



**FIGURE 8.** (a) Line segments considered for a particular cell when the original cell is occupied. (b) Wrong assignment of the segment to the scanned point.

### 3.3 Segment correspondence

To determine which line segment inside the assigned cells to match to the scanned point, a minimum distance criterion is used (Figure 8a). This assumption is valid as long as the displacement between sensed data and model is small enough. This assumption limits the robot traversed distance between consecutive position estimates. However, since after each iteration the point/line-segment pairs are updated, the limitation can be relaxed somewhat (Figure 9).

Given a scanned point  $\mathbf{P}_i = (X_i, Y_i)$ , three different distances for each line segment are computed (Figure 2). They are given by equations 5, 6 and 7. The smallest distance to the line segments inside the selected cells determines which line segment  $l_j$  is matched to  $\mathbf{P}_i$ .

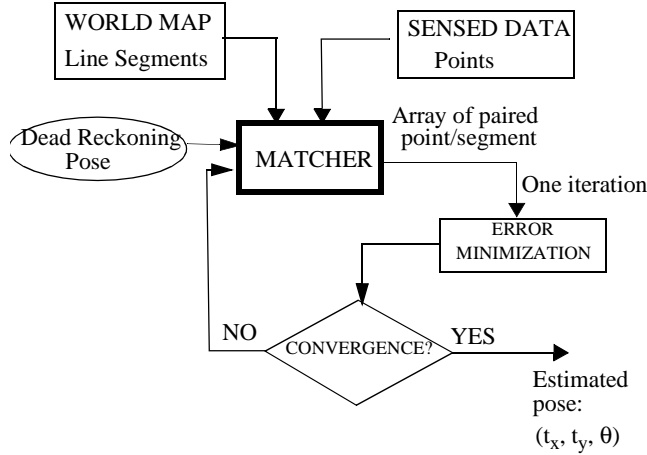


FIGURE 9. Block diagram of the iconic position estimator

### 3.4 Minimization

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

$$\min(\mathbf{e}^T \mathbf{e}) = \min\left(\sum_{i=1}^n e_i^2\right) \quad (\text{EQ 13})$$

where  $e_i = e_i(t_x, t_y, \theta)$  is given in section 2 by Equation 9.

Although the rotation  $\theta$  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 [16]. However, we have opted for an 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:

$$\mathbf{e} + \mathbf{J} \mathbf{d} = \mathbf{0} \quad (\text{EQ 14})$$

where  $\mathbf{e}$  is the error vector given by Equation 9,  $\mathbf{d}$  is the difference vector between the transformation parameters on successive iterations, and  $\mathbf{J}$  is the Jacobian:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial t_x} & \frac{\partial e_1}{\partial t_y} & \frac{\partial e_1}{\partial \theta} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial t_x} & \frac{\partial e_n}{\partial t_y} & \frac{\partial e_n}{\partial \theta} \end{bmatrix} \quad (\text{EQ 15})$$

Notice that Equation 14 is overdetermined for  $n > 3$ . Then we use the pseudoinverse of the Jacobian to find a least square fit of  $\mathbf{d}$ :

$$\mathbf{d} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{e} \quad (\text{EQ 16})$$

Equation 16 is solved iteratively for the displacement vector  $\mathbf{d}$  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 to the final estimate, thus our approach is functionally equivalent to the closed-form solution with updating.

This method has a quadratic convergence and is less sensitive than the conventional Newton method to the starting point and the conditioning of the Jacobian. However, for this algorithm to converge to a correct solution, it must be guaranteed that the initial orientation error is less than 90deg, well within the accuracy of dead reckoning.

## 4.0 Application

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

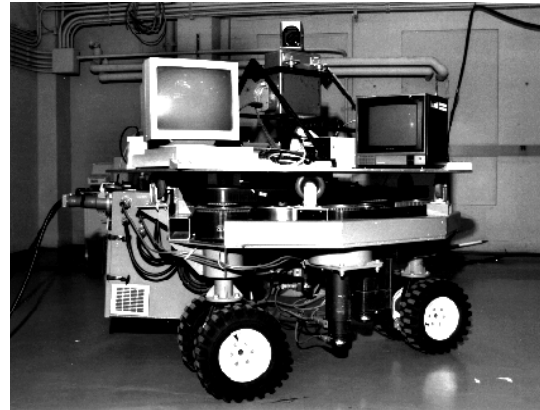


FIGURE 10. The Locomotion Emulator

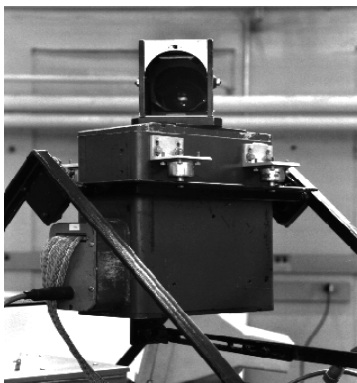
#### 4.1 The Locomotion Emulator

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 (Figure 10). 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].

#### 4.2 Cyclone

The Cyclone laser range scanner (Figure 11) was also developed at the CMU Field Robotics Center to acquire fast, precise scans of range data over long distances (up to 50m). 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  $\pm 20$ cm. 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].

It has been our experience that the effects of the truncation due to the resolution of the scanner and the non-linear behavior of the range measurement along different operating target distances must be considered to obtain accurate results in any position estimation algorithm [15]. Rather than using the standard 10cm range resolution and  $\pm 20$ cm accuracy numbers mentioned above, we considered the Cyclone scanner's range offset<sup>1</sup>, truncation and angular accuracy characterization obtained in a separate study [7].



**FIGURE 11.** The Cyclone laser rangefinder

1.- The range offset is a function which describes the difference between the measured range value and the true distance to the target.

#### 4.3 Experimental results

The iconic position estimation algorithm presented in this paper was tested at the highbay area of the FRC. Figure 12 shows the line segment model that was used as world map in this application. The corridor is about 6m wide and 20m long. 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 is 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 13 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 12.

The estimator was programmed to use two different representations of the model. In the first one, the model was represented by 8 long line segments shown in figure 12. 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:  $\delta_r = 5$ cm and  $\epsilon_r = 5$ deg for the LE (5% of the step size) and  $\delta_s = 10$ cm and  $\epsilon_s = 0.7$ deg for the Cyclone. The grid size was  $0.6 \times 0.6$ m<sup>2</sup>.

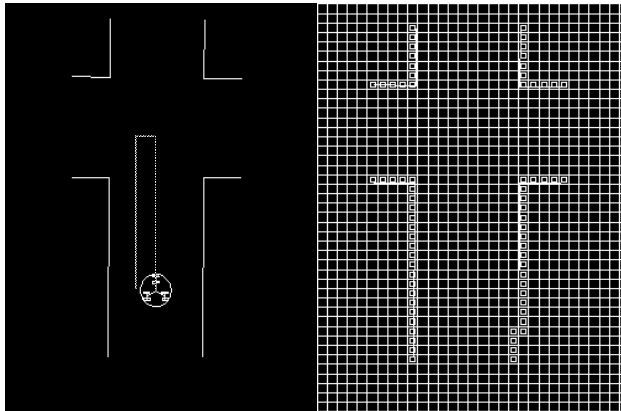
As expected the computed error (surveyed minus estimate) for the two representations was exactly the same at the 20 positions along the path (Figure 14). The maximum position error was 3.6cm, and the average position error was 1.99cm. The maximum heading error was 1.8deg. and the average was 0.73deg. 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.

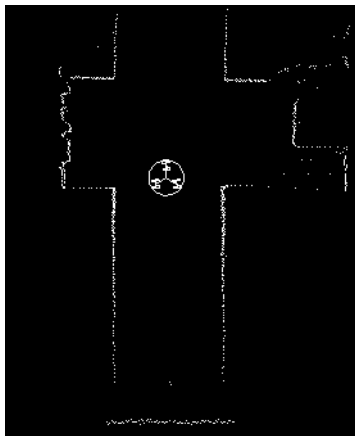
In the event that the uncertainty regions for the sensed points can be approximated by circles centered on the points, the segment correspondence can be computed rap-



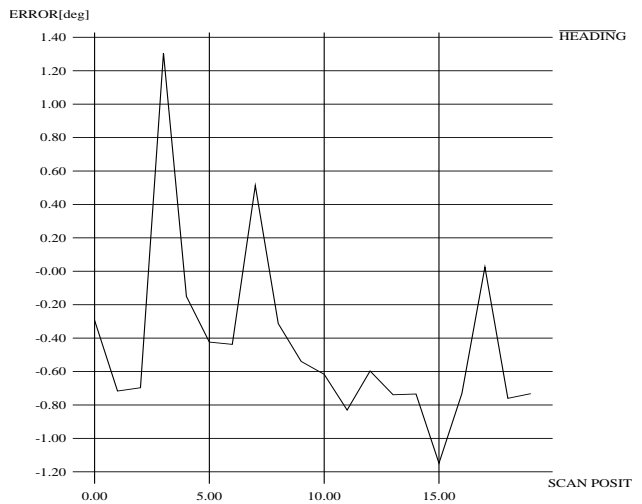
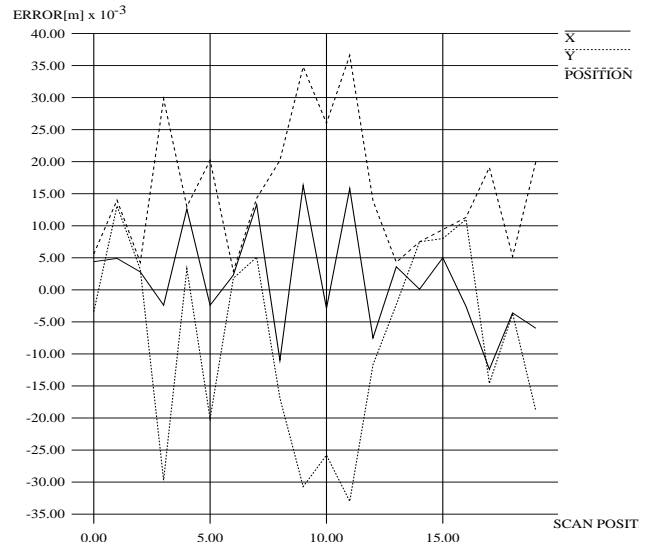
idly using a numerical Voronoi diagram. This approximation worked well for our highbay experiments [18].



**FIGURE 12.** World model representation and map representation



**FIGURE 13.** Range scan provided by the Cyclone. The circular icon represents the LE at the position where the scan was taken from.



**FIGURE 14.** Computed errors for the 20 positions along the path.

## 5.0 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.

Future work includes a more complete navigation system including map building capability and a Gaussian uncertainty model for both the sensor and the robot motion.

## 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] 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.
- [2] 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.
- [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 Automation", March 1985.
- [5] J. L. Crowley. "Navigation for an intelligent mobile robot". IEEE Journal of Robotics and Automation, vol. RA-1, no. 1, March 1985.
- [6] S. Singh, J. West, "Cyclone: A Laser Rangefinder for Mobile Robot Navigation", Technical Report, CMU-RI-TR-91 1991
- [7] J. Gonzalez. "Estimacion de la Posicion y Construccion de Mapas para un Robot Movil equipado con un Escaner Laser Radial". PhD dissertation. Dpt. Ing. Sistemas y Automatica, Universidad de Malaga. Junio 1993.
- [8] B. C. Bloom. "Use of landmarks for mobile robot navigation". In Proc. SPIE Conference on Intelligent Robots and Computer Vision, vol. 579, 1985.
- [9] Z. L. Cao, J.J. Roning and E. L. Hall. "Omnidirectional vision navigation integrating beacon recognition with positioning". In Proc. SPIE Conference on Intelligent Robots and Computer Vision, vol. 727, 1986.
- [10] M. P. Case. "Single landmark navigation by mobile robots". In Proc. SPIE Conference on Intelligent Robots and Computer Vision, vol. 727, 1986.
- [11] E. Krotkov. "Mobile robot localization using a single image". In Proc. IEEE Robotics and Automation Conference, May 1989.
- [12] H. P. Moravec. "Robot Rover Visual Navigation". Ann Arbor, MI:UMI Research Press, 1981.
- [13] M. Hebert, T. Kanade, I. Kweon. "3-D Vision Techniques for Autonomous Vehicles". Technical Report CMU-RI-TR-88-12, 1988.
- [14] M. Drumheller. "Mobile robot localization using sonar". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 2, March 1987.
- [15] S. Sedas and J. Gonzalez. "Improvement in Robot Position Estimation through Sensor Calibration". IFAC Int. Symposium on Intelligent Components and Instruments for Control Applications. Malaga, Spain. May, 1992.
- [16] 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.
- [17] A. Elfes. "Sonar-based real-world mapping and navigation". IEEE Journal of Robotics and Automation, vol. RA-3, no. 3, June 1987.
- [18] G. Shaffer and A. Stentz. "Automated Surveying of Mines using a Scanning Laser Rangefinder". SME Annual Meeting Symposium on Robotics and Automation, February, 1993.
- [19] C. D. MacGillem and T. S. Rappaport. "Infra-red Location System for Navigation of Autonomous Vehicles", in Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1236-1238, 1988.