Model-Based Map Construction for Robot Localization*

Gregory Dudek and **Paul MacKenzie**
Research Centre for Intelligent Machines

McGill University, 3480 University St.,
Montréal, Québec, Canada H3A 2A7
email: {dudek,mackenzi}@mcrcim.mcgill.edu

Abstract

We consider a robot equipped only with simple range sensing that must move and navigate within an initially unfamiliar environment. The robot is not given a map of the environment and constructs one dynamically. This map is not meant to represent the actual spatial structure of the environment so much as it is meant to represent the major structural components of what the robot "sees". The problem with such an approach is that maintaining an absolute coordinate systems for the map is difficult.

We demonstrate that in a suitable environment it is possible to use sonar data to recalibrate position estimates on an ongoing basis. This is accomplished by incrementally constructing and updating a model-based description of the acquired data. Given coarse position estimates of the robot's location, these can be effectively refined to high accuracy using the stored map. Experimental data are presented indicating when this can and cannot be accomplished.

1 Introduction

A complete and accurate *a priori* map of an environment is a useful tool for navigating within it. In general, however, such accurate metric maps are not available for robotic use. Even more important, when maps (such as architect's floorplans) are available in a form that can be used, they tend to not accurately portray the environment in a manner consistent with typical robotic sensing devices. For example, commonly-used sonar devices fail to detect many existing structures (such as thin pipes or overhangs) and may "detect" many structures that are not physically present (such as illusory walls in either corners or discontinuities in the surface derivative or a floor). For these reasons, it would be extremely valuable if an autonomous robot could construct and maintain a map of a novel environment in terms of its own perceptual mechanisms.

Several authors have attempted to use near-ideal *a priori* maps for localization using either sonar [4, 10] or laser range finders [8], apparently with some success. In contrast with this previous work, we suggest how maps can be constructed and maintained "on the fly" without prior knowledge of the specific structure of the environment, how they can be used with low computational cost, and present quantitative data indicating their effectiveness.

Most simple devices for measuring position and distance are relative measurement tools (e.g. odometers). Imperfect estimates of orientation, distance and velocity must be integrated over time and hence errors in absolute position accumulate disastrously with successive motions and make the general problem of maintaining an accurate absolute coordinate system very difficult [3]. For these reasons, map constructing and long-term localization are dependent on the use of sensory data for recalibrating a robots sense of its own location within the environment.

Given approximate estimates of a robot's position based on odometry and dead-reckoning, we show in this paper how a geometric map can be constructed and used for ongoing re-calibration. Our construction is based on the use of stable detectable structures of sizable spatial extent in the environment and avoids intervention such as the placement of beacons. Unlike the work of Leonard and Durrant-Whyte which also uses sonar-based features [12], this approach uses sonar features of limited spatial extent and combines these with a confidence updating scheme that incorporates a weighting based on the quality of the particular features being used. Although the method is described using sonar sensors, its applicability is not restricted to this sensing modality. Where the environmental structure does not conform sufficiently to the *a priori* models assumed here or when coarse position estimates are unavailable, other more costly methods can be applied from simpler assumptions [5]. In regions where such structures are not present or too sparse, other methods may still be required for global integration by connecting geometric regions in a qualitative or non-metric way [11, 6, 1, 9]. If the robot is able to sense its location accurately within a familiar geometrically modelled region, it can compensate for the cumulative errors that result from uncertainties in its movements as it travels within and between regions.

In this paper we begin by describing the manner in with a map can be constructed using dynamically acquired sonar range data. We go on to show how

* *This paper appears in the Proceedings of* **Vision Interface**, *June 1993, Toronto Canada.*
The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council.

positional re-calibration can be achieved. Finally, we present experimental data indicating the typical accuracy of this process and discuss some extensions to the algorithm.

2 Environmental Models

We will consider here the case of two-dimensional environmental modelling only. We use line segments to model collections of observations of the environment. Each segment can be thought of as representing a section of a wall or other obstacle although, in fact, some linear collections of observations may not correspond directly to existing structures. (More elaborate models are possible, and in fact were planned as part of this project, but were found to be unnecessary given the quality of results obtainable simply with linear models.) Raw sonar data obtained from a robot with a ring of 12 Polaroid sonar transducers is processed to give these line segments.

Part of the reason for the appropriateness of simple linear environmental models derives from the characteristics of simple threshold-based sonar sensing [2, 7]. Each measurement corresponds to the first over-threshold response for a brief ultrasonic "chirp". Each outgoing chirp can be roughly described as a 12 steradian-wide measurement cone and the first object of sufficient size within this cone results in a single response at that object's distance. Consequently, a similar orientation that also includes the same object will return the same measurement (unless it hits a closer object). As a result, even a small object will produce a collection of measurements at similar distances that are nearly linear in structure.

The construction of an environmental map includes the following steps:

1. Sonar data are converted into (inferred) distance measurements using simple thresholding. The raw sensors may be used at multiple orientations to acquire denser data (by rotating the robot slightly).

2. Data that corresponds to pre-existing models is used as described later and removed.

3. The data are spatially clustered into groups of nearby measurements. This associates measurements that may arise from the same object (or interaction). Disparate measurements from the same object may be grouped subsequently.

4. Line segments are generated using an iterative split and merge algorithm.

5. New line segment models are combined with existing models.

2.1 Clustering Algorithm

The clustering algorithm is an adaptation of the sphere-of-influence graph [16]. This is a threshold-free technique for clustering data points based on finding neighbours that fall within a circle whose radius depends on the distance to the nearest neighbour. For each data point p_i, i ϵ $\{1 \ldots \mathrm{n}\}$, let R_i be the distance

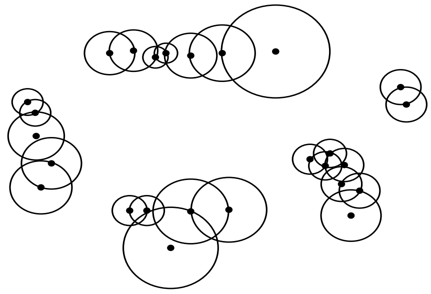

Figure 1: Sphere of Influence Clustering. Points with overlapping nearest-neighbour circles are clustered.

to p_i's nearest neighbour. Two points p_i and p_j $(i \neq j)$ are defined as belonging to the same cluster if the circle of radius R_i centered at p_i intersects the circle of radius R_j centered at p_j. Figure 1 shows a simple example, where the ability to obtain linear clusters is demonstrated.

This clustering algorithm can be computed with complexity $\mathcal{O}(n \log n)$, where n is the number of points [16]. In addition, since range measurements associated with sonar contain sensor-dependent noise, we fix a lower limit on R_i guaranteeing a minimum distance to associate neighbours. This bounds the minimum cluster size and excludes the possibility of very small clusters which would not correspond to physically plausible objects given the limited sensor resolution.

2.2 Fitting Line Segment Models to Data

Assigning line segment models to the data clusters is done with a *fit-split-merge* strategy. It operates as follows: given a cluster of data points, a single line segment is fit to the entire set of points. If the residual of the fit is low (the fit is good and the line segment fits the data well) then this model is retained as a description for this cluster. If the fit is poor, then the data is not well modelled by a single line, and the cluster is divided into two equal parts. Fitting is then attempted on each of the new clusters individually. This is performed recursively until each cluster has a line segment that fits well, or until a stopping condition is reached. An *a priori* stopping criterion is needed to prevent splitting of the clusters into too many tiny groups (in the worst case two points each, since a line fit to two points is always a good fit). After all lines have been chosen, any line segments that are close together and colinear are merged into a larger single compound line.

The line fitting algorithm used based on a least mean squares fit using the singular values of the covariance matrix of the data [15]. This corresponds to fitting an ellipse to the data and finding the major and minor axes, which are the directions of maximum and minimum variance respectively. This removes any dependence on the particular coordinate system, as would be the case if simple least squares fitting were used. The best fit line is taken along the major axis and, if line splitting is required, the minor axis is used

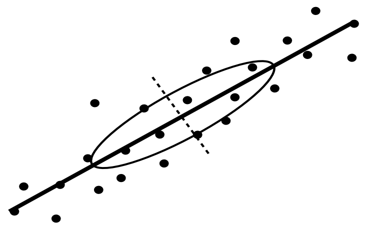

Figure 2: Fitting Lines from Variance Ellipses.

to divide the cluster into two parts (points on one side of the minor axis are in one group, those on the other side are in another) (Figure 2).

Of great importance in this operation are the criteria for line splitting. If a cluster is a corner (i.e. junction of two walls), we would like to split it such that we can fit a line segment model to each of the perpendicular sections. If the cluster takes the shape of a single line, we would not like the line to split.

The two main factors in this decision are the elongatedness of the fit ellipse (see above) and the length of the line. We define elongatedness simply as the ratio of the maximum to minimum singular values of the covariance matrix of the data [15]. An elongated ellipse tells us that the maximum variance of the data (which is along the fitted line through the data) is much greater than the minimum variance (along a perpendicular line to the fitted line), and thus the data tends to fit the shape of a single line. However, if the minimum variance is still significant (i.e. more than just a few centimetres) there may some structure perpendicular to the fitted line that we do not want to miss. Only if the fit line is very long will we have an elongated ellipse AND a significant minimum variance. Therefore we would like to split long lines just in case this perpendicular structure exists. If it does not and the data is indeed shaped like a long ling segment, then the merging of line segments after the fitting process will give this result.

In order to compensate for the interplay of line length and condition number, the following relationship is used to estimate line quality and determine line splitting:

$$\text{Quality of Line} = \frac{\text{length of fit line segment}}{\text{elongation of variance ellipse}} \quad (1)$$

Additional factors in the line generation process are the actual length of the fit line segment (we do not want to split a tiny line into even tinier ones) and the number of data points used to fit the line segment model (after splitting a number of times, there may only be a few points left in a given subsection of the cluster).

Figure 3 shows the results of the modelling process in an area of our lab. The area was scanned by moving about the partially enclosed area (roughly 4 square meters) where the robot is shown. The linear models

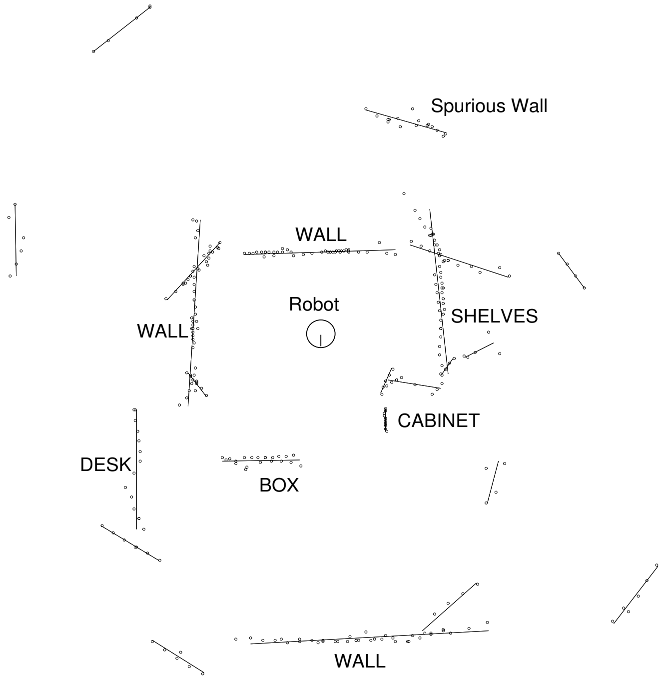

Figure 3: Modelling of Range Data with Line Segments. Dots are sonar measurements; lines are inferred models

corresponding to major structures in the environment (such as walls) are evident. In addition, line segments can be observed modelling non-physical collections of data points that correspond to artifacts of the sonar sensor. As will be mentioned later, it is valuable to retain these apparently spurious data. This process of selecting the minimum number of sufficiently good lines is related to minimal length encoding theory.

3 Calibration

Position estimation or localization is performed using a relatively straight forward strategy that does not involve any formal pattern matching. It assumes that at some prior time a map of the type previously discussed was made of the working environment, i.e. the environment was scanned, clustered and line segment models were fitted, either from a single location or multiple locations. There are two phases in calibration: 1) Classification of Data Points and 2) Weighted Voting of Correction Vectors. The classification stage entails examining sensed data points and *classifying* each as to which line segment model it should belong, if any. The voting stage consists of computing a vector difference between each point and its associated model and a weighted sum of these *correction vectors* to give an estimate of the robot's true position. This process is then repeated until the position can no longer be further refined.

3.1 Classification of Data Points

The purpose of classification is to decide whether the sensor data is the result of responses from previously modelled obstacles in the environment, i.e. to test if the robot can see any parts of the world that it saw before. A neighbourhood is defined around each line segment model, and if a point is within this neighbourhood, it is a candidate for belonging to the model. If a point is a candidate for more than one model, it is assigned to the model to which it is closest. The size of this neighbourhood is governed by an estimate of the error in a sonar range measurement, as well as an estimate of the maximum distance for which the robot's assumed position is different from its actual position.

3.2 Weighted Voting of Correction Vectors

Once some subset of the sonar points has been matched with the line segment models, we compare the position of each point to that of its matched model and find a simple vector displacement $(\Delta x_i, \Delta y_i)$ from the point to the closest point on the line segment model. We cannot simply sum these vector displacements together to find an average displacement - we must bias their effects with respect to the orientation of the models being used. The following weighted averages are used to calculate the overall displacement $(\Delta X, \Delta Y)$:

$$\Delta X = \frac{\sum_{i=1}^{n} \omega_{xi} \Delta x_i}{\sum_{i=1}^{n} \omega_{xi}} \quad (2)$$

$$\Delta Y = \frac{\sum_{i=1}^{n} \omega_{yi} \Delta y_i}{\sum_{i=1}^{n} \omega_{yi}} \quad (3)$$

where

$$\omega_{xi} = \cos\theta_i \times f(d_i) \quad (4)$$

$$\omega_{yi} = \sin\theta_i \times f(d_i) \quad (5)$$

ω_{xi} and ω_{yi} are the *weights* of the i^{th} displacement, and θ_i is the angle of the normal of the line segment model to which point i is classified. Therefore, these weights are the projections of the normal of the line segment onto the coordinate axes, in addition to a function $f(d)$ that weights sonar point i as a function of its distance d_i from its paired line segment model. $f(d)$ is a sigmoid function, having values close to unity for small distances d that are intrinsically more reliable and zero for large d. This "soft-nonlinearity" serves to reject outliers and ensures that points close to their matched line segments have a greater voting strength in the overall displacement (based on the assumption that points close to lines are more likely to be correctly classsified).

The necessity for the first factor of this weighting comes from the geometric constraint (in fact, the lack of constraint) that derives from matching to a one dimensional (line) model. This problem which manifests itself in this context as the *long hallway effect*, is precisely analogous to the *aperture problem* in motion estimation [14]. Observation of position (or motion) of a section of a straight line provide information only in the direction of the normal to the line. In practice, a robot in the middle of a long hallway (such that it cannot see the ends) can only correct its position in the direction perpendicular to the hallway. Movement parallel to the axis of the hallway gives no displacement information since for our chosen model of the line segment all parts of the walls look identical, and therefore cannot be distinguished in order to calculate a displacement. Kalman filtering can be used in such situations to combine dead-reckoning information with sensory input [12].

3.3 Convergence to the True Position

If the robot's initially estimated position is very close to its actual position then with ideal data the above method should allow the robot to calculate its true position with a single iteration. However, to allow position correction using a noisy sensor (such as sonar) and given large errors in the initial position estimate, it is necessary to perform the calculations iteratively, incrementally updating the robot's position estimate. This is due to the fact that some points that belong to a given model may not be correctly classified and be outside of an appropriate model's classification neighbourhood. As the position estimate for the robot changes, the point-to-model correspondences can change substantially. Not all the points may be properly classified initially, but only a few are needed to start moving in the right direction; incorrect correspondences tend not to be consistently organized and hence are readily outweighed by correct ones. As the assumed position converges to the true position, the points belonging to a model will come into the model's neighbourhood providing greater accuracy.

The key to this process is clearly a reliable initial classification of sonar data points with the appropriate models of objects that produced them. In practice, this depends to a large extent on the distribution of objects in the world.

4 Experimental Results

In order to evaluate the performance of this algorithm, a variety of scenarios of correcting initial position estimates were examined. In order to evaluate the region of convergence from incorrect position estimates to an accurate estimate, the position estimation algorithm was started with initial estimates whose error ranged up to 100 cm in both the x and y directions from the robot's actual position. Actual positions were measured manually to within one-half centimeter using a tiled grid on the floor of the test area. Figure 4 shows the resulting converging and nonconverging paths to the correct position (the centre of the figure) from the various starting estimates (with a classification neighbourhood of 50 cm). Figure 5 shows more clearly the region around the true location for which the estimated robot position converges correctly. The other initial estimates do not converge correctly due to sonar points being incorrectly classified (a point very far from its true model may end up close to another model). Correct convergence for this figure corresponds to a final error of less than 3 cm.

Position estimation errors after convergence tend to be on the order of 2 cm or less in moderately well structured environments such as the office space depicted earlier. Figure 6 shows the results as a function of initial position error. When the position estimate

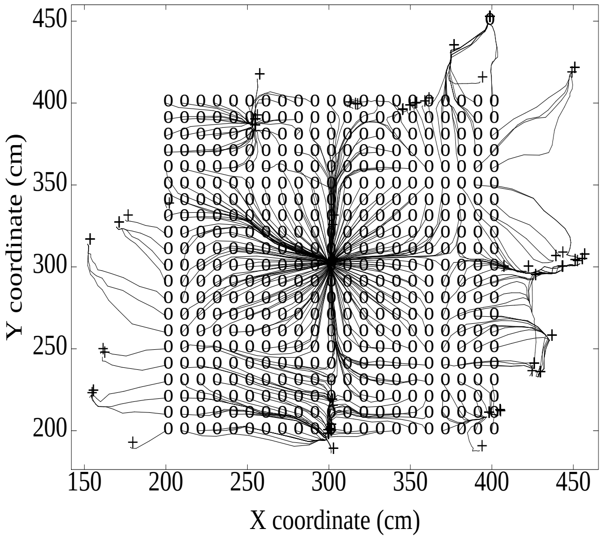

Figure 4: Paths of Convergence to the Correct Position of the Robot (see text for details)

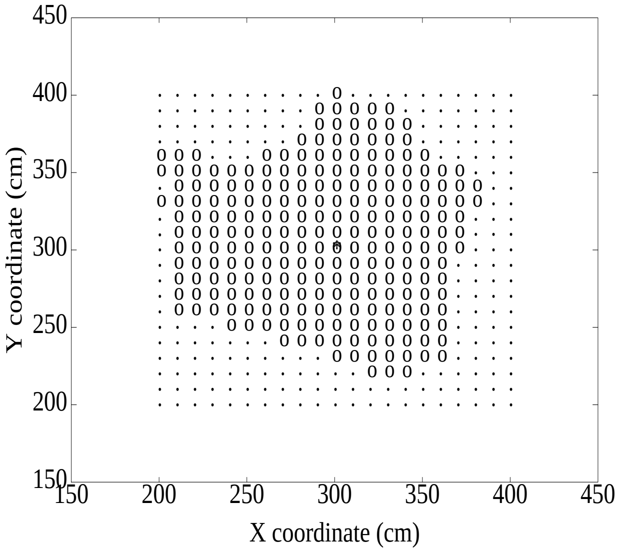

Figure 5: Initial Position Estimates that Converge to the Correct Position of the Robot (circles indicate successful convergence within 3cm of true position). The filled dot in the centre is the correct position.

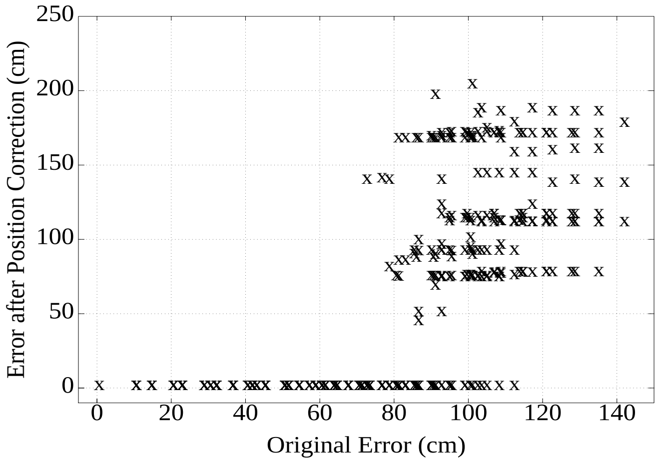

Figure 6: Error in Position Estimate as a function of Starting Error

is too far off many classifications are incorrect and the final estimate may converge to an incorrect position (although this is generally detectable). The most common source of error aside from poorly fit models (due to sparse and/or noisy data) is the "hallway effect" previously mentioned, where there may be insufficient structure to correctly estimate the position along some orientation.

5 Discussion and Conclusions

While this work has addressed position localization, orientation correction is also an important issue. One way of generalizing the approach presented here is to perform the calibration at a number of different angular displacements from the assumed angle and select the one that produced the most consistent point-to-model matches.

This rotational estimation problem is closely related to the problem of estimating the confidence in a position estimate. Using the expected values for sonar correspondences from a random position it is possible to estimate the degree of non-accidentalness in a set of data and thus derive a threshold to apply to the confidence measure or quality estimate for a position estimate [13]. This suggests that global position estimation without any prior estimate is also feasible.

This paper presented a geometric method to generate maps from sonar data and to perform localization based on a line segment map of obstacles that need not be individually identified. Individual sonar data were classified using a weighted soft non-linearity that combined robustness with graceful degradation. Performance was very good for the range of position errors likely to be encountered in actual use and for actual environments. In areas where insufficient environmental structure is observable practical systems would be wise to make use of dead-reckoning information as well.

References

[1] Kenneth Basye and Thomas Dean. Map learning with indistinguishable locations. In M. Henrion L. N. Kanal J. F. Lemmer, editor, *Uncertainty in Artificial Intelligence 5*, pages 331–340. Elsevier Science Publishers, 1990.

[2] C. Biber, S. Ellin, E. Shenk, and J. Stempeck. The polaroid ultrasonic ranging system. *Proc the 67th Convention of the Audio Engineering Society*, 1980.

[3] Ernest Davis. *Representing and Acquiring Geographic Knowledge.* Pitman and Morgan Kaufmann Publishers, Inc., London and Los Altos, California, 1986.

[4] Michael Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.

[5] G. Dudek and G. Hinton. Navigating without a map by directly transforming sensory input to location. *(sumbitted for publication)*, 1992.

[6] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Robotic exploration as graph construction. *Transactions on Robotics and Automation*, 7(6):859–865, December 1991.

[7] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Sonar sensing and obstacle detection. In *Proc. of the Conference on Military Robotics*, Medecine Hat, Alberta, September 1991.

[8] Javier Gonzalez, Anthony Stentz, and Anibal Ollero. An iconic position estimator for a 2d laser rangefinder. In *IEEE International Conference on Robotics and Automation*, pages 2646–2651, May 1992.

[9] S. Harmon. Autonomous vehicles. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 39–45. John Wiley, 1987.

[10] Alois A. Holenstein, Markus A. Müller, and Essam Badreddin. Mobile robot localization in a structured environment cluttered with obstacles. In *IEEE International Conference on Robotics and Automation*, pages 2576–2581, Nice, France, May 1992.

[11] Benjamin J. Kuipers and Y. T. Byun. A qualitative approach to robot exploration and map-learning. In *Proceedings of the IEEE workshop on spatial reasoning and multi-sensor fusion*, pages 390–404, Los Altos, CA, 1987. IEEE.

[12] John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[13] Paul MacKenzie and Greg Dudek. Robotic pose estimation using model-based maps. Technical Report CIM-93-500, McGill Research Centre for Intelligent Machines, McGill University, Montréal, 1993.

[14] David Marr. *Vision.* W. H. Freeman and Co., New York, 1982.

[15] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing.* Academic Press, New York, 1976.

[16] Godfried Toussaint. A graph-theoretical primal sketch. *Computational Morphology*, 1988.