Mobile Robot Relocation

by

Sheri A. Cheng

S.B., Ocean Engineering, Massachusetts Institute of Technology
(1999)

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Ocean Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Author ....
Department of Ocean Engineering
September 1, 2001

Certified by ....
John J. Leonard
Associate Professor of Ocean Engineering
Thesis Supervisor

Accepted by ....
Professor Henrik Schmidt
Chairman, Departmental Committee on Graduate Students

Mobile Robot Relocation

by

Sheri A. Cheng

Submitted to the Department of Ocean Engineering
on September 1, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Ocean Engineering

Abstract

The goal of mobile robot relocation is to determine the position of a robot from sensor measurements using an a priori map. Current methods in relocation are largely based on probabilistic approaches using particle filters. An alternative approach is based on using echolocation measurements of the surrounding environment as data features in a constraint-based search to determine the robot's position. The goal of this thesis research has been to investigate the latter approach, by creating a C++ implementation of search-based relocation, and evaluating its performance with data sets from several different environments. The algorithm considers all feasible assignments of pairs of measurements with pairs of features in a given map to generate candidate locations for the robot. Hypothesized positions are then evaluated by comparing the remaining sensor measurements with predictions from the model. This algorithm has been implemented in C++ and integrated with a visualization software package based on openGL. The method has been tested on several data sets from buildings of the MIT campus, including MIT's Compton Gallery and the corridors of buildings 1 and 5. The results provide suggestions for future research, including the development of more efficient search strategies and the integration of relocation with concurrent mapping and localization.

Thesis Supervisor: John J. Leonard
Title: Associate Professor of Ocean Engineering

Acknowledgments

Dedication

"One time I hired a monkey to take notes for me in class. I would just sit there, my mind a complete blank, while the monkey scribbled on little pieces of paper. At the end of the week, the teacher said, 'Class, I want you to write a paper using your notes.' So I wrote a paper that said, 'Hello, my name is Bingo. I like to climb on things. Can I have a banana? Eek eek!' I got an F. When I told my mom about it, she said, 'I told you, never trust a monkey!' The end."

Brak, from Space Ghost, "Never Trust a Monkey"

To Mom, Dad, Jeremy, and Jillian, without whom I may not have made it through my years of school.

Contents

List of Figures

Chapter 1

Introduction

In order to navigate autonomously in its environment, a mobile robot needs to know where it is. With an a priori map, the robot's position can be estimated either by localizing itself using knowledge of its current and previous movements, or relocating itself solely based on knowledge of its current surroundings. Reliable position estimation is crucial to navigation and mapping; without accurate tracking, an autonomous system will grow in error as time goes on, thus generating unreliable data and maps.

The goal of relocation is to give the best estimate of the robot's position given no information about its previous states, with only a set of measurements describing its immediate environment. This is essential when no prior estimates are available and the robot has become lost within its given map. Current methods in relocation are largely based on probabilistic approaches, using Bayesian filtering and both grid-based and particle-based density representations. Using these approaches, the robot's position estimate improves over time, starting with the notion that it is equally likely to be anywhere in the map. At each time step, the robot then uses the new information to refine its assumptions about its surroundings and position. Our approach uses constraint-based search to determine the robot's position. By matching echolocation measurements of the surrounding environment as data features, this algorithm is more of a "brute force" method to the relocation problem. The algorithm tests all possibilities of measurements and features, eliminating the solutions that don't fit within given sonar properties and geometric constraints.

This search-based relocation work is complementary to work done by Leonard et al. on concurrent mapping and localization (CML) [14, 9]. The goal of CML is to enable a mobile robot to build a map of an unknown environment while using that map to navigate [14]. The algorithm presented in this thesis is written as part of the existing CML code, allowing us to test the relocation method with a large variety of data sets that have recently been acquired.

1.1 Background on Object Identification Problems, Sonar and CML

1.1.1 B21 Robot and Sonars

The B21 mobile robot by Real World Interfaces (RWI) supplies the data sets for our research, and is shown in Figure 1-1. Affectionately named Leo, after Leopold Bloom from Ulysses, the B21 is equipped with 2 rings of 24 sonar sensors, located in the enclosure and in the base of the robot, and a SICK PLS laser scanner. Only the enclosure sonar returns are used in our relocation research; the other data is used in related CML work.

The sonar sensors on the robot are Polaroid Ultrasonic Rangefinders, which are standard in robotics research involving sonar returns. They are plagued with a reputation for being problematic, unpredictable and unreliable [13]. For those expecting sonar to provide data resembling a real-world map (as a simple ray-trace model might), sonar data appears to be "bad", filled with erroneous range readings. In their book, Leonard and Durrant-Whyte counter sonar's reputation by presenting Polaroid sonar data as predictable and accurate, by formalizing the notion of regions of constant depth (RCDs). An RCD is a set of connected set of sonar returns differing in range by less than some predefined threshold. The RCD is defined by its range (the mode of the set of range returns) and width (the difference in angle between the extreme returns in the set). RCDs are assigned an order, defined as the number of surfaces sound has reflected from before returning to the transducer. First-order

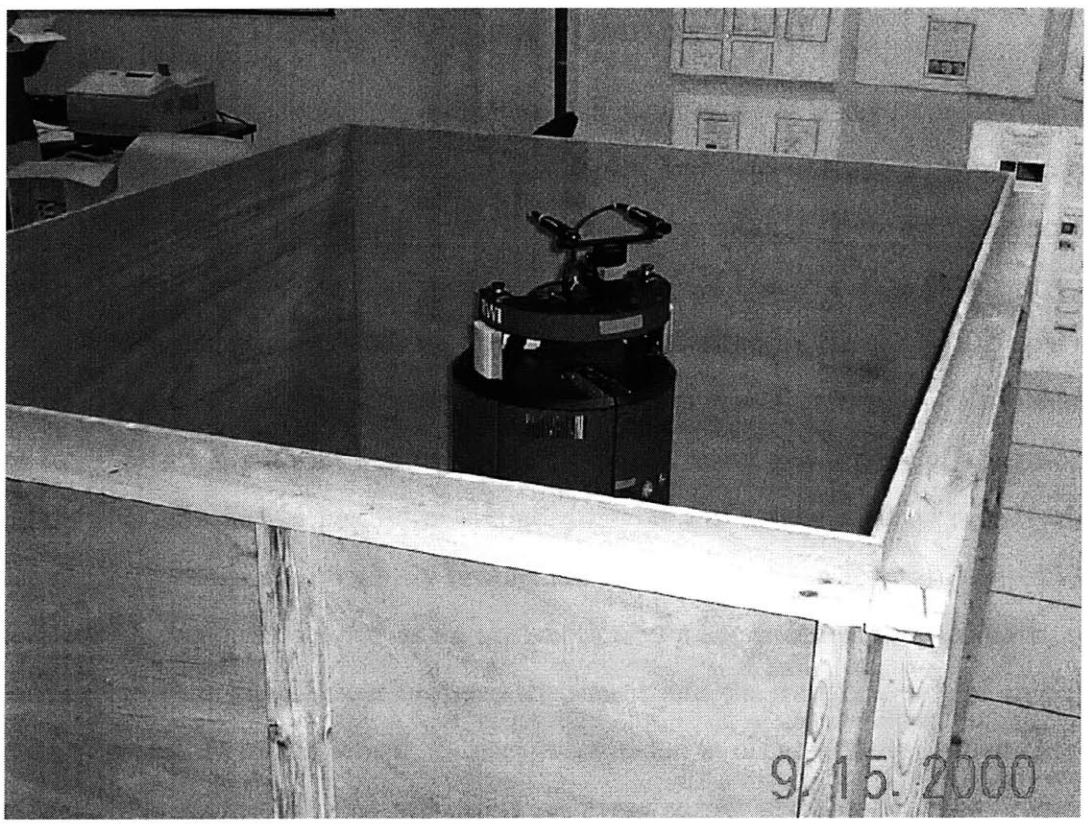

Figure 1-1: The B21 mobile robot (illustrated here in the plywood box used for simple testing of the algorithm.)

returns are produced by planes perpendicular to the transducer, and second-order returns are produced by corners. Higher order returns are given by side-lobe and other reflections when the beam is at a high angle of incidence to a target. These higher order returns are not accurate range readings, and appear to be erroneous data. Incorrect interpretation of these spurious data points is simply one factor in the problems of object identification.

1.1.2 The Problem of Object Identification

Identifying objects in the sensory information a robot receives is key to navigation and tracking techniques. After determining the nature and pose of these objects, localization is made possible by matching the objects to the robot's given map. Relocation is made difficult by the inherent problems of the object identification problem. Identifying objects is made most difficult by three factors cited by Grimson [10]: spurious data, occlusion, and noise. Spurious data stems from the aforementioned problem of data interpretation without regard to order of the returns. Intuitively, if a high-order data point is assumed to be first or second-order, it will return a false range reading. False range readings are obviously problematic in the quest for accurate object identification. Occlusion is defined as an obstruction in a sonar sensor's beam path, such as an object in the line-of-sight between the sensor and the wall, or a side of an object not directly visible to the sensor. Object recognition is necessary even when there is obstruction; in reality, a robot will rarely see every corner and side of its environment at all times. Grimson asserts that good recognition methods must be able to deal with shortcomings of occlusion and spurious data.

Real-world sensors are plagued with noise, which corrupts data and increases uncertainty in readings. Additionally, real environments are filled with extra objects and clutter, and not just simple walls, corners, and edges. Coping with uncertainty is one the key challenges in robotics sensing.

In addition to the problems arising from uncertainty in sensor data, a key difficulty in developing a relocation algorithms is computationally complexity [10].

1.2 Thesis Road-map

In this chapter, we have provided a high-level overview of the issues addressed in the thesis.

Chapter Two reviews previous and current research in mobile robot relocation. It addresses methods using search algorithms, and gives a brief overview of other particle-based and grid-based relocation methods.

Chapter Three summarizes our method for relocation, which is based on constraint-based search, implemented in C within the framework of CML code developed by Leonard.

Chapter Four applies the method to data from a land mobile robot navigating in a series of indoor environments.

Chapter Five concludes the thesis and makes recommendations for future

Chapter 2

Previous research in mobile robot relocation

2.1 Relocation using Sonar using Search Methods

2.1.1 Drumheller

Mobile robot relocation using sonar scans was first approached by Drumheller [7], who referred to it as absolute localization. He developed a search-based algorithm derived from work on the Interpretation Tree method by Grimson and Lozano-Perez [10]. Drumheller used connected line segments as features of sonar data, which are straight segments extracted from a sonar contour. Localization is performed as a two-dimensional matching problem between the sonar segments and the room outline to determine the robot's pose with respect to the room. Model features and data features are paired together, and unfeasible pairings are initially eliminated using a distance constraint (the distances of sonar segments must be comparable to distances between walls they are assigned to), an angle constraint (the range of angles between a pair of sonar segments must include the known angle between pair of walls they are assigned to), and a normal-direction constraint (the same as distance constraint, but measured along the segments' normal vectors).

After the interpretation tree is pared down, Drumheller introduces a new constraint, called sonar barrier test, which basically tests for occlusion. It states that a sonar ray won't intersect a wall, lie outside the cone of reflection for that wall, or has any endpoint outside that wall.

In addition to these basic constraints, Drumheller tested that the vehicle geometries would lie within the map, and also expressed preferences for long segments of sonar data, which are assumed to be more accurate. Drumheller introduced this method of relocation as a "worst-case scenario" for localization due to the "extreme errors due to the nature of sound propagation," but work by Leonard and Durrant-Whyte [13] has asserted that sonar can be a predictable and reliable tool.

2.1.2 Leonard and Durrant-Whyte

Leonard and Durrant-Whyte were the first to introduce the term "relocation" to refer to localization without knowledge of prior states in their book [13]. The term is adapted from orienteering, an outdoor sport where racers' navigation skills are often more important than their speed. In orienteering, relocation means "recovery from getting lost", and is done by matching local features, such as vegetation or rocks, to map features. The authors review basics of extracting RCDs from sonar data, basing their methods on Kuc and Siegel's work [11] presenting sonar data as consisting of circular arcs and not line segments. Features are more easily identified as RCDs in the sonar data, while returns not associated with an RCD can be assumed to be diffuse reflections. Instead of being a "worst-case scenario", good reliable range estimates can be obtained. This work suggested that future work in relocation could be advanced by exploring a substitution of these RCDs for Drumheller's sonar contours. Adding a relocation method with RCDs to their existing Kalman-filter based navigation algorithms would bring them closer to a fully autonomous, navigating, mapping robot.

2.1.3 Lim and Leonard

Lim and Leonard [16] furthered Drumheller's work by using RCDs in an Interpretation Tree method. The substitution of RCDs for line segments is not straightforward, because the corners and edges are indistinguishable from planes if only a single sensor is used. In [16] the testing environment is modeled as geometric primitives such as walls, corners, edges, and cylinders. Their relocation method was applied to a small room environment, with implementation in Matlab for visualization of results. Since this method is the basis for our work, we will cover it in more detail in the next chapter.

2.1.4 Castellanos

The approach investigated in this thesis is similar to the approach of Castellanos [5]. Castellanos describes a search based algorithm for relocation which is called the first location problem. They provide experimental results for a 2D environment using multisensor system: laser range scanner data and vision data, not using sonar like the others. It uses interpretation tree method compared to hand-measured map. Geometric constraints separated are into location dependent and location independent). They discuss and compare two different algorithms: identifying before locating, and identifying while locating. The distinctions between these two approach have an analogy in model-based object recognition [10].

2.2 Alternative methods

2.2.1 Grid-based

An alternative to search-based representation is to use grid-based representation, such as the work done by Moravec [17], Elfes [8], and more recently by Schultz and Adams [19], which involves continuous localization. An evidence grid divides an area (or volume) into cells, and each cell contains a value (-1,1), where a value of -1 represents an unoccupied cell, a value of 1 represents an occupied cell, and a value of

0 represents an equal likelihood of being occupied or unoccupied. The environment is divided into an evidence grid representation, and a long-term map is acquired using many sensor readings over a long period of time. Short-term maps are then built of the robot's immediate environment, from a recent set of sensor readings. The short-term maps are then used to place the robot within the long-term map and correct its overall position estimate. Continuous localization is the process of making these smaller corrections at regular intervals.

The advantages of these evidence grid methods are intuitive. Regular relocation leads to smaller errors at each time step. With the knowledge that the errors will be small, the search space for relocation is greatly reduced, speeding up the correction techniques. In addition, evidence grids lend themselves to using combinations of sensors, so fusion of data is simple and beneficial. The difficulty of an evidence grid representation is apparent in the method in which the long-term map is built. In a real-world environment, there may be objects are present in a short-term map that would not appear in a long-term map, such as people walking through, or movement of furniture. The creation of the long-term map by accumulation of sensor data will lead to problems with dynamic environments, such as hallways, or any environment with people.

There is also the problem of computational overhead involved in discretizing representing environment into a grid, most importantly the requirement that the resolution and precision of the environment be fixed beforehand.

2.2.2 Particle filtering

A popular current alternative to search-based methods is based on the use of particle filtering. Monte Carlo Localization (MCL) work by Dellaert et al. [6] illustrates particle-filtering work on localization and relocation using CMU's B21 robot Rhino. Thrun's work on Bayesian landmark learning uses particle-filtering techniques to allow a robot to determine salient features in the environment on its own, touting flexibility over static approaches to mobile robot localization [20].

Particle filtering is a powerful generic framework based on Bayes' theorem. Particle filtering represents the probability density of the robot's location using a random set of samples (particles) drawn from all the data that is received. Localization of the robot is done recursively computing the density in a prediction and an update phase. In the first phase, samples are drawn from the previous state, and a motion model is applied, resulting in a random sample of the predictive density.

The second phase incorporates incorporates sensor measurements, weights the samples in the predictive density, and then re-samples the weighted set, selecting particles with higher likelihood associated with them. When particle filters are applied to relocation problems, the prior density is represented as a uniform probability density, meaning that the robot is equally likely to be anywhere in the environment, except for where static obstacles are already known to be located. After several iterations, the vehicle will be localized.

Wijk has also investigated relocation using particle filters, with very interesting results [21]. He has reported that even with a very large number of particles (500,000), in some tests the method only achieved a 50% success rate. He developed a new technique, called planned sampling, that achieved better performance with a lower computational burden [21].

Relocation using particle filtering is attractive because of its ability to handle multi-modal distributions. Dellaert et al. [6] have also extended the approach to apply for multiple mobile robots.

2.3 Relation of the current work to previous work

This thesis investigates the use of a search-based algorithm for relocation, based on the Interpretation Tree [10]. We follow this technique because we believe that it has the potential to be integrated with CML more seamlessly than other methods that have been published. The key challenge with all relocation methods is achieving robust performance while maintaining computational efficiency. The goal has been to experimentally investigate the tradeoffs between performance and complexity by creating a new C implementation of the method and testing it with a variety of experimental data sets.

Chapter 3

Localization Algorithm

This chapter summarizes method from [16]. First we the sonar measurement model is reviewed, and then the relocation algorithm is described.

3.1 Measurement Model

Echolocation refers to the process of measuring the location of an object by determining the time for an echo to return from the object to the sonar, in response to an interrogating pulse from the sonar [1]. If the echoes reflected from an object can be simultaneously detected by multiple transducers [2], then one can also measure the direction to a reflection object. The method presented in this chapter assumes that each transducer operates independently, and hence angle cannot be measured directly. Because of the wide beam pattern of the Polaroid sonar, the angle to the reflecting object cannot be reliably estimated from individual returns [13]. The measurement produced by the sensor is a range value $r = \frac{ct}{2}$, where c is the speed of sound in air and t is the round-trip travel time from the sensor to the object and back.

A physics-based sonar sensor model can be used to derive the geometric constraints provided by echolocation measurements for different types of objects [13]. We assume that environmental features can be classified according to four types: planes, cylinders, corners, and edges. We approximate the world as being two dimensional, so that planes are represented by lines, cylinders by circles, and corners or edges by points. Examples of cylindrical features that might be encountered in an indoor environment include building pillars. We use the word "target" to refer to the environmental features.

In addition, we assume that the surfaces of the environment are smooth in relation to the wavelength of the sonar. In a specular wavelength regime, rough surface diffraction [4] can be ignored. If rough surfaces are encountered, extra returns will be produced at high angles of incidence from line targets; these will need to be rejected as outliers as a by-product of the constraint-based search procedure.

For a long duration, single frequency transmitted pulse, the beam pattern $b(\theta)$ of a circular disc transducer such as the Polaroid sonar is given by [18]:

$$b(\theta) = \left(\frac{2J_1(ka \sin\theta)}{ka \sin\theta} \right)^2, \tag{3.1}$$

where J_1 is a first-order Bessel function, θ is the angle from the sensor axis, $k = \frac{2\pi}{\lambda}$ is the wavenumber, and a is the radius of the transducer. For the Polaroid sensor, $a = 39$ mm and $\lambda = 6.9$ mm.

Bozma and Kuc [3] have found that with short, impulsive excitations, the beam pattern of the Polaroid transducer has a Gaussian shape and side-lobe effects are minimized. However, with the standard Polaroid driver circuit, which uses a longer transmitted pulse, side-lobe levels can be significant. While under normal circumstances, a range return is produced by the main central lobe, returns can also be generated from the side lobes of the radiation pattern.

For specular surfaces, only the perpendicular portion of the surface reflects the beam directly back to the transducer [11]. For a line target, if we let θ_t be the angle with respect to the x axis of a perpendicular drawn from the line to the sensor location, as shown in Figure 3-1, the range of values of sensor bearing, θ_s, that can produce a range return is

$$\theta_t - \frac{\beta}{2} \leq \theta_s \leq \theta_t + \frac{\beta}{2}, \tag{3.2}$$

where β is defined as visibility angle of the target and represents the maximum range of angles over which a return is produced by a target. We define Equation 3.2 as the

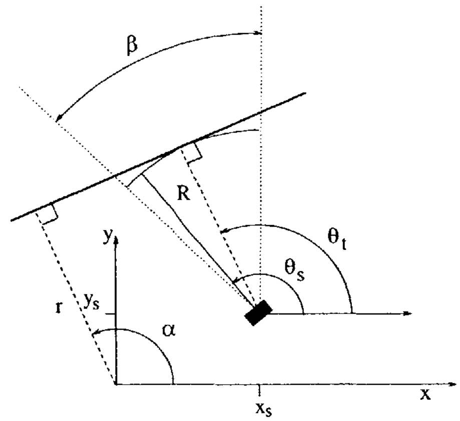

Figure 3-1: Plane target model. A plane is represented by the perpendicular distance r and orientation α. The shaded rectangle indicates a single sonar sensor located at the position (x_s, y_s, θ_s).

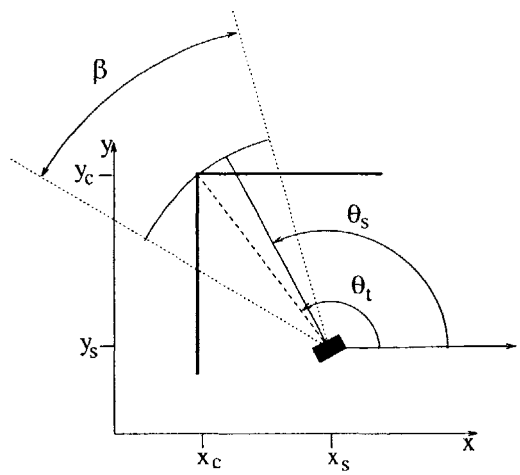

Figure 3-2: Corner target model. A corner is represented by the position (x_c, y_c).

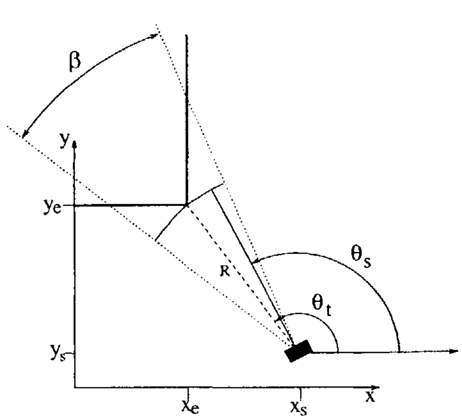

Figure 3-3: Edge target model. An edge is represented by the position (x_e, y_e).

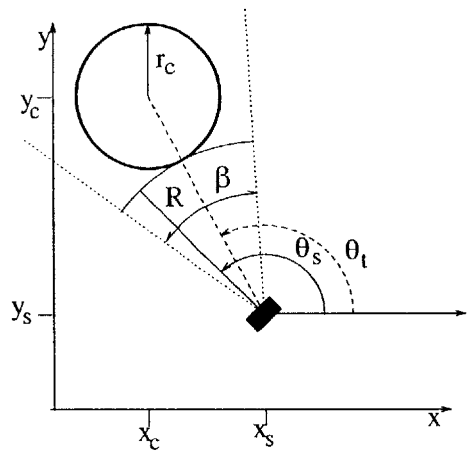

Figure 3-4: Cylinder target model. A cylinder is represented by the position of its center (x_c, y_c) and its radius r_c.

sensor orientation constraint.

For a point target such as a corner or edge, the range of values of θ_s is identical to Equation 3.2, except θ_t in this case is defined as the angle between the x axis of the global reference frame and the line drawn from the sensor location to the point that defines the location of the point target. In practice, edge targets will be visible over a smaller range of angles than other types of targets, because they provide weaker returns. Our method uses the conservative strategy of using the large value of $\beta = 50$ for all types of targets.

A cylinder is represented by a circle and is defined by the x and y coordinates of the center and the radius of the circle. θ_t for a cylinder is defined as the angle between the x axis of global reference frame and the line drawn from the sensor location to the center of the cylinder.

3.2 Relocation Procedure

Relocation is basically a searching problem that finds the best correspondence between sensor data and the model features. Thus the reduction of the search cost as well as the accuracy of the result is very important. If we have m data features (sonar returns) and n model features, the search cost will grow at the rate of $(n+1)^m$ when we use the basic Interpretation Tree algorithm of Grimson and Lozano-Perez [10]. To reduce the number of data values to be considered in such a procedure, one method is to group sonar returns that are hypothesized to originate from the same environmental object into a "data feature". This was the motivation for Drumheller to extract straight line segments from sonar scans, to serve as input to the data.

In Drumheller's work, line segments extracted from sonar scans were effectively used as constraints for relocation. A line segment can reduce dramatically both the search cost and the angle uncertainty of the robot's configuration. As stated earlier, however, it is difficult to extract line segments from even densely sampled data because most of the object surfaces in an indoor environment can be considered to be specular. Furthermore, it is impossible to extract line segments from sparse data such as a circular ring of 16 sonar sensors.

The relocation method presented here uses the constraints derived from a physics-based measurement model in a hypothesize and test search procedure [10]. The algorithm can employ either individual sonar returns or multiple adjacent sonar returns grouped together (circular arcs) as data features. For simplicity, we state the algorithm in terms of the situation when all sonar returns are referenced to the center of the robot, but the method can be generalized to accommodate arbitrary sonar configurations.

The method is summarized in Algorithm 1. The key steps are described below:

Step 1: generation of trial positions. From the combination of any two range returns, f_i and f_j, we consider all possible ways of pairing the returns with targets F_p and F_q, i.e., sets of pairings f_i:F_p and f_j:F_q. The association of a return with a plane target gives a line on which the sensor is constrained to be located. The association of a return with a corner, edge, or cylinder target gives a circle of possible sensor positions. Each pair of feature-to-model associations generates zero, one, or two possible positions of the robot, based on computation of the intersection points of the line and circle position constraints produced from each association. (See Figures 3-5 to 3-7).

For associations that match both returns to plane targets, many infeasible pairings

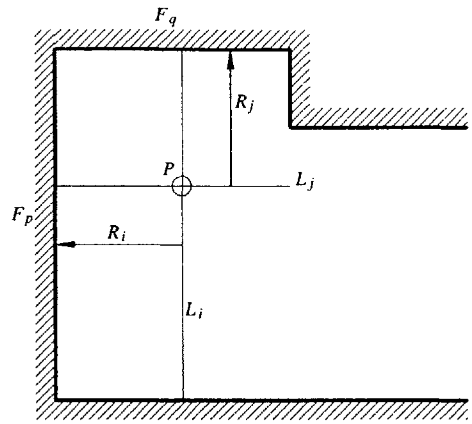

Figure 3-5: A possible trial position P for the robot calculated from the hypothesized match of return R_i with line target F_p and return R_j with line target F_q.

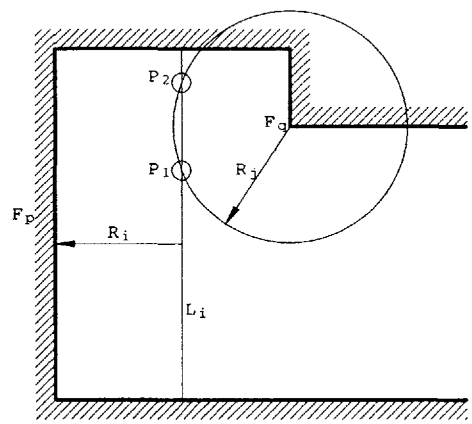

Figure 3-6: Possible trial positions P_1 and P_2 for the robot calculated from the hypothesized match of return R_i with line target F_p and Return R_j with point target F_q.

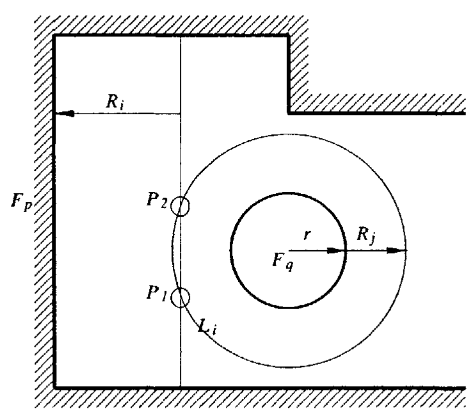

Figure 3-7: Possible trial positions P_1 and P_2 for the robot calculated from the hypothesized match of return R_i with line target F_p and return R_j with cylinder target F_q (r is the radius of the cylinder).

can be quickly removed without calculation of a trial position through application of a binary angle constraint [10]. The binary angle constraint tests the relative orientation of the measurements f_i and f_j against the difference in angle between the normals of F_p and F_q, to see if they agree within the target visibility angle β. Additional binary constraints based on intersection of visibility regions for different targets can also be incorporated in the method.

Step 2: removal of infeasible trial positions. For each trial position $\hat{x}_k$ calculated in step 1, a trial angle is calculated for the sensor using the mean of the angle to each target from the trial position. Then, predicted sonar returns are generated for the features F_p and F_q, and these are matched against the measurements f_i and f_j by the application of the sensor orientation constraint (Equation 3.2). The sonar prediction function incorporates an occlusion test, which determines if target F_p is visible from position $\hat{x}_k$. If the occlusion test fails for either pairing, then the position is considered infeasible.

Step 3: matching of additional range values based on the hypothesized location. Predicted returns (range and angle) are generated for each model feature that is visible from the trial position and matched to the remaining ranges in the data set. Predicted ranges $\hat{R}_t$ and actual ranges R_t are compared with a set threshold

$$|\hat{R}_t - R_t| \leq r_e \tag{3.3}$$

If Equation 3.3 holds true, we get a successful match. The value of r_e is chosen based on the range accuracy of the sensor and the uncertainty of the trial position. If multiple predicted ranges match a measured range, then the closest prediction is used.

Step 4: pose refinement using least squares estimation. For each trial position, suppose that K is the number of matched predictions and observations determined in step 3. If K is greater than a minimum number of matches N, then a least squares algorithm is used to compute an improved location and orientation estimate for the robot based on all of the matched predictions and observations, and this location is added to the solution set. An additional parameter Δ, is used to control the size of the solution set. If $K - \Delta$ is greater than N, then the solution set is pruned by removing any solutions with with less that $K - \Delta$ matches, and N is set to $K - \Delta$. In our experiments, good results have been obtained with $\Delta = 0$. By increasing the value of Δ to a larger value, such as $\Delta = 2$, a greater number of solutions is provided, and a better insight into the the algorithm's performance can be obtained.

Step 5: clustering of computed locations. The locations in the solution set are clustered together to produce a single solution for each set of locations within a specified error range (x_e, y_e, θ_e) of one another.

```
function x = relocation(R, F)
inputs:
  R = {R_1, ..., R_m}                    ▷ a set of m sonar returns (observations)
  F = {F_1, ..., F_n}                    ▷ a set of n model features (targets)
outputs:
  x = {x_1, ..., x_r}                    ▷ a set of r potential robot positions
  M                                      ▷ number of positions in x
control parameters:
  x_e, y_e, θ_e, r_e                     ▷ position, range, and angle tolerances (typical values are 10 cm and 10 degrees)
  β                                      ▷ maximum range of angles over with a target is visible (50 degrees for Polaroid sonar)
  N                                      ▷ initial value for the minimum number of matching
                                           sonar returns for each position in x (typically six)
  Δ                                      ▷ maximum difference in the number of matching sonar returns
                                           for each location in the solution set (typically zero)
local variables:
  a                                      ▷ set of assignments between predicted and observed sonar returns
  R̂                                      ▷ predicted sonar returns
  K                                      ▷ number of matches for a hypothesized location
  x̂                                      ▷ hypothesized vehicle positions

x ← ∅
M ← 0
for i = 1 to m − 1
  for j = i + 1 to m
    for p = 1 to n
      for q = 1 to n
        if binary_constraints(R_i, R_j, F_p, F_q) == TRUE then          ▷ apply binary constraints
          x̂ = {x̂_1, ..., x̂_s} ← generate_locations(R_i, R_j, F_p, F_q)   ▷ generate s hypothesized positions
          for k = 1 to s
            R̂_i ← sonar_prediction(x̂_k, F_p)                          ▷ predict sonar return for feature F_p
            if match_returns(R_i, R̂_i) == TRUE then
              R̂_j ← sonar_prediction(x̂_k, F_q)                        ▷ predict sonar return for feature F_q
              if match_returns(R_j, R̂_j) == TRUE then
                R̂ = {R̂_1, ..., R̂_t} ← sonar_prediction (x̂_k, {F_1, ..., F_n})   ▷ generate predicted sonar
                                                                      returns from hypothesized position k
                (K, a) ← match_returns(R̂, {R_1, ..., R_m}\R_i, R_j)   ▷ match predicted sonar returns
                                                                      with remaining measurements,
                                                                      producing a set of assignments
                if K >= N then
                  x_M ← compute_position(R_i, F_p, R_j, F_q, a)       ▷ calculate improved position
                                                                      estimate using all matched
                                                                      predictions and observations
                  x ← x ∪ {x_M}                                       ▷ add new solution to set of solutions
                  M ← M + 1
                end
                if (K − Δ) > N then
                  (x, M) ← prune(x, K − Δ)                            ▷ remove any solutions with
                                                                      less than K − Δ matches
                  N ← K − Δ                                           ▷ increase the minimum number of matches
                end
              end
            end
          end
        end
      end
    end
  end
end

x ← clustering(x, x_e, y_e, θ_e)        ▷ group positions in x into clusters based on the tolerances x_e, y_e, θ_e
                                         and replace the set x by the set of average positions for each cluster
x ← sort(x)                             ▷ sort solutions based on number of matches and residual from least squares computation
return x, M
```

Algorithm 1: Summary of the relocation procedure.

Chapter 4

Experimental Results

This chapter describes the results of applying the new implementation of the mobile relocation algorithm to several mobile robot data sets.

4.1 Results

The relocation algorithm has been tested in several different buildings of the MIT campus. In particular, tests have been performed with six different model files:

- **Box**: a 1.8-meter by 2.4-meter box (shown in Figure 1-1).
- **Museum**: a 10-meter by 10-meter room (the MIT Compton Gallery, shown in Figure 4-1).
- **Building 1**: an approximately 60-meter long segment of the 2nd floor corridor of building 1.
- **Building 5**: an approximately 60-meter long segment of the 2nd floor corridor of building 5
- **Lobby 7**: Lobby 7 of MIT.
- **Complete Infinite Corridor**: Model file consisting of a rough approximation to the geometry of the 2nd floor of MIT buildings 1 through 11 (Figures 4-2 and 4-3).

Figures 4-1 through 4-3 show representative output from the system for several of these environments. For all but the Complete Infinite Corridor model, the reference files were painstakingly generated for each environment using a tape measure. The Complete Infinite Corridor model tests used a very coarse environmental model that was generated by hand-measuring of prominent features from MIT's database of architectural drawings.

The biggest limitation of the technique is its reliance on very detailed, hand-measured models of the environment. Generation of the model for a small segment of corridor, e.g. 20 meters, could take over 2 hours. Clearly for a real application, it will be necessary to integrate relocation with concurrent mapping and localization (CML) [14].

In addition, we were surprised by the susceptibility of the relocation algorithm to coping with symmetries in the environment. It may be that relocation from a small number of sonar returns, obtained from just a single vantage point is too "aggressive". There are simply too many possibilities for the robot's location. Future research should address methods to perform relocation using data from multiple vantage points.

4.2 Conclusion

While further work is necessary to perform an exhaustive performance analysis for the new implementation, a number of important lessons have been learned in this research. Several key limitations of the relocation algorithm were identified, including its dependency on a very detailed environmental model and its susceptibility to generation of many solutions in environments with a great deal of symmetry. These issues are discussed further in Chapter 5, in the context of future research.

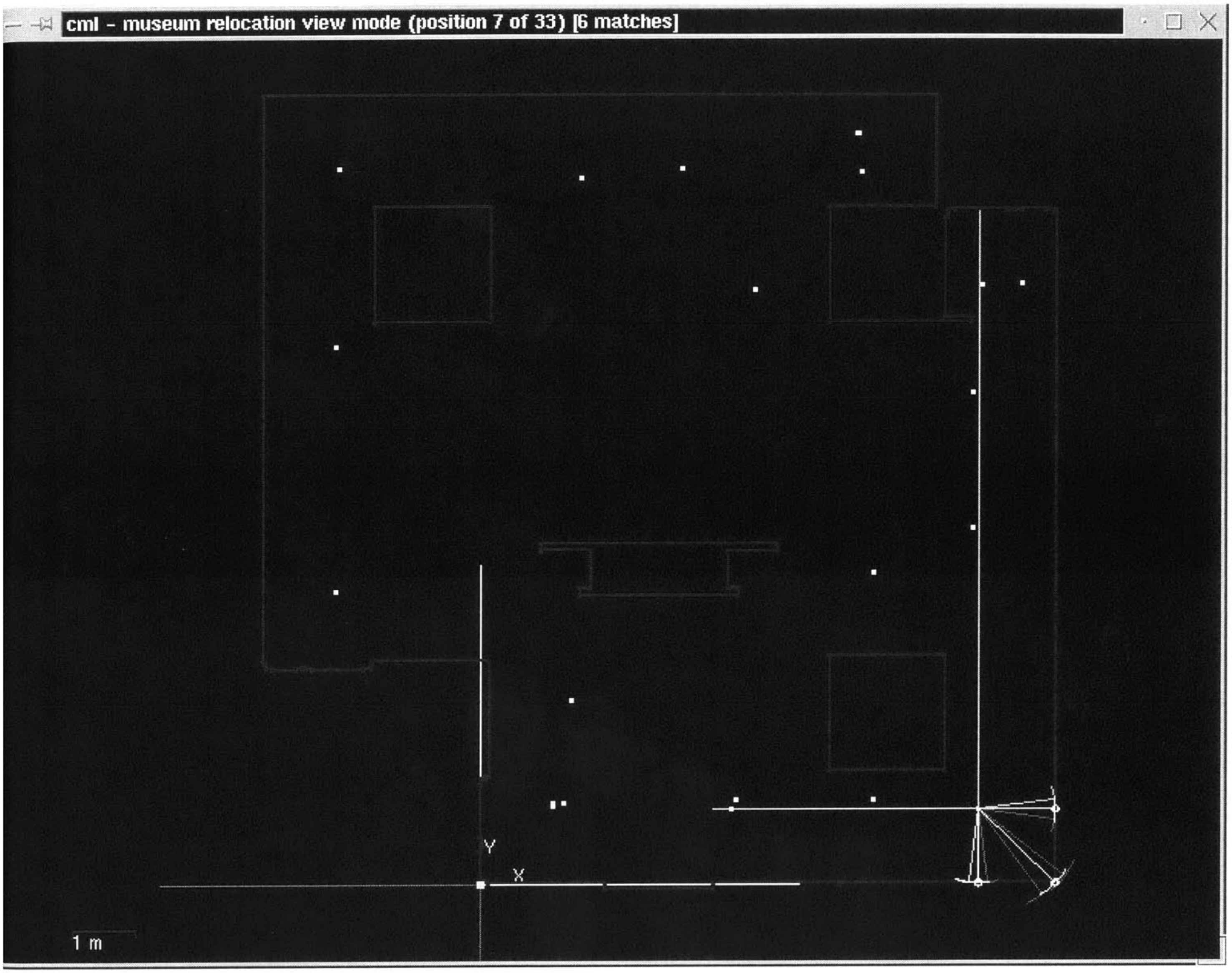

Figure 4-1: Typical result for testing of the algorithm in the Museum environment.

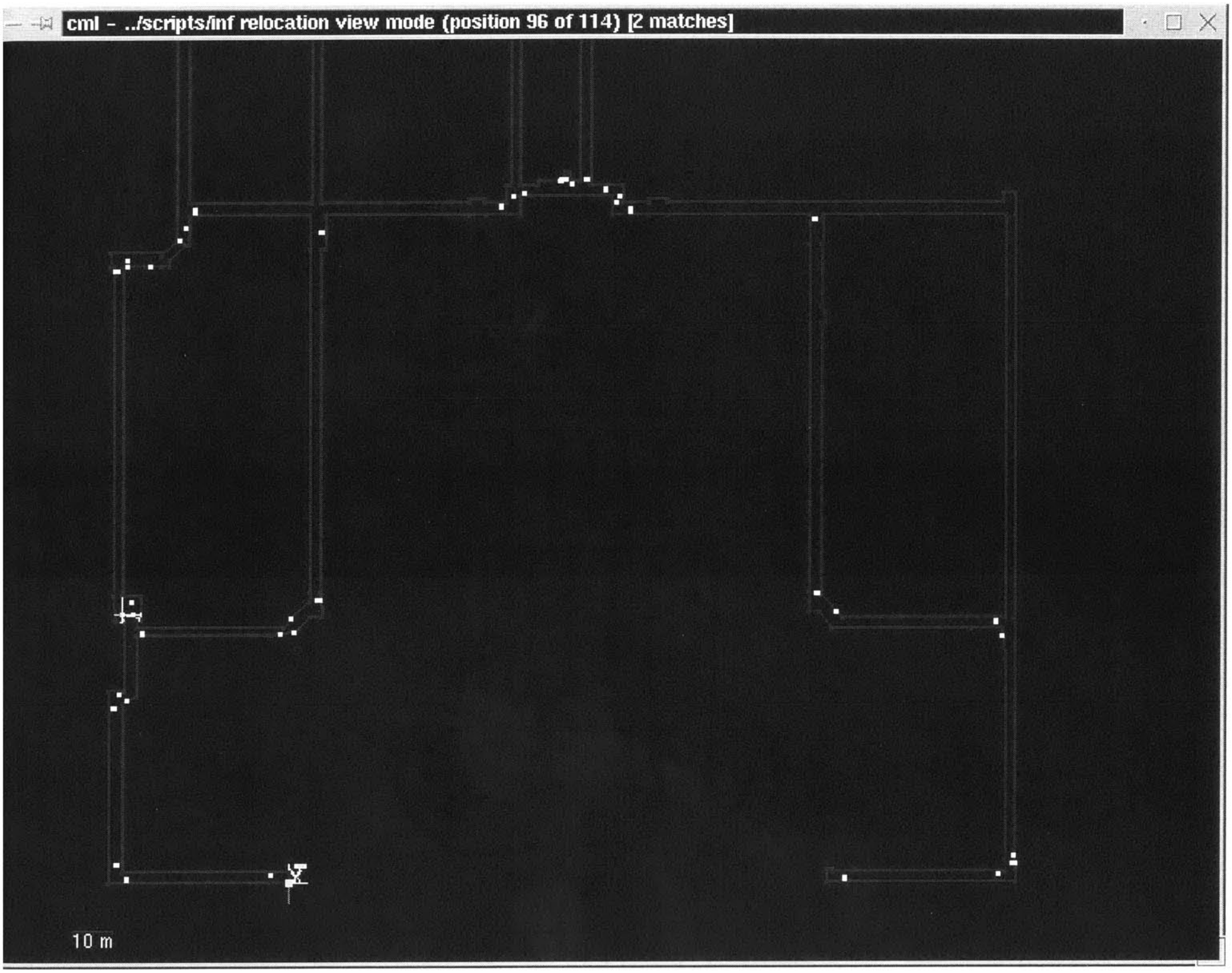

Figure 4-2: Testing of the algorithm in the Complete Infinite Corridor environment.

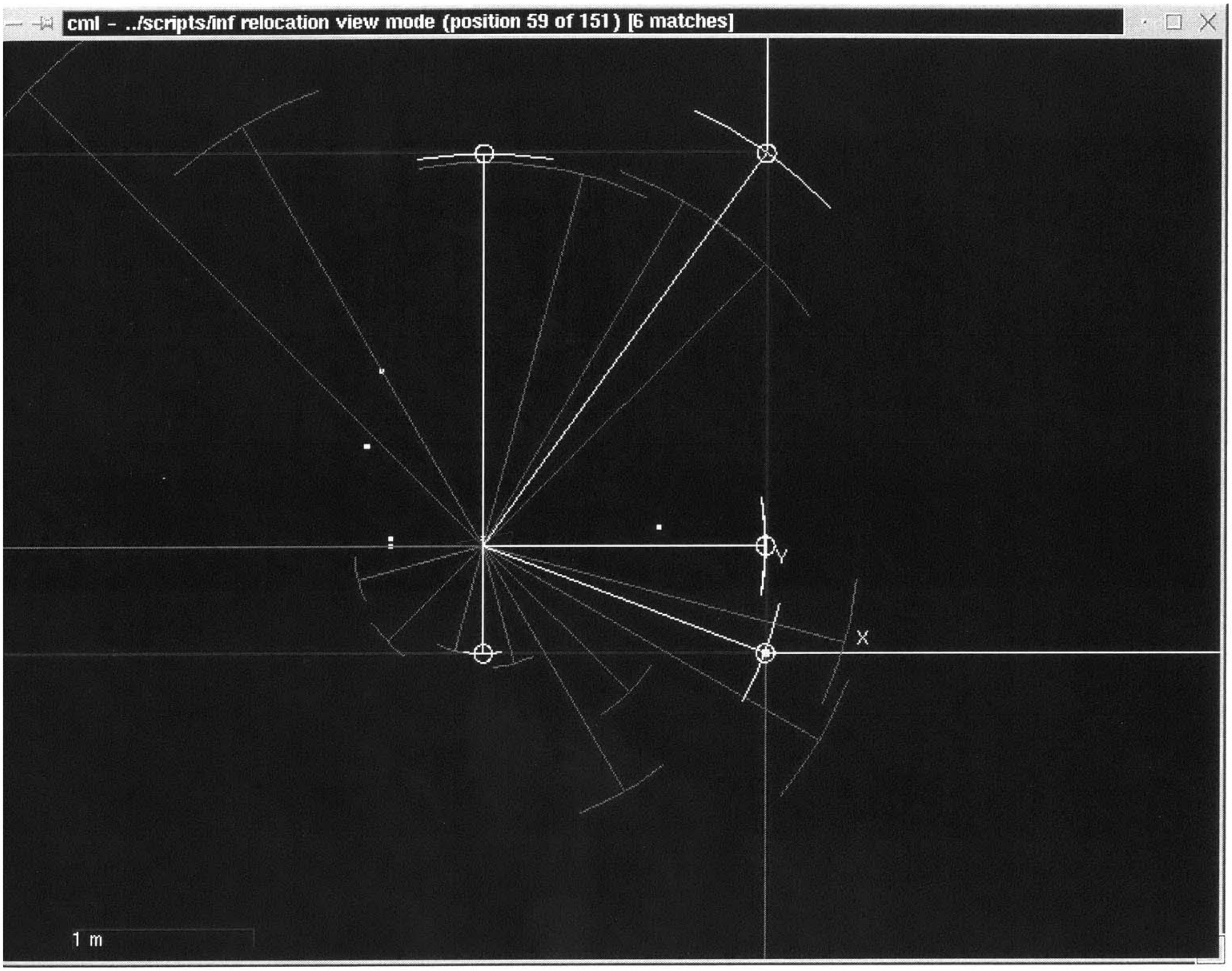

Figure 4-3: Close-up of the "correct" solution for testing in the Complete Infinite Corridor environment.

Chapter 5

Conclusion

5.1 Summary

This thesis has investigated the relocation problem for mobile robots. A search-based relocation algorithm has been implemented in C and tested with a variety of real data sets. The C++ implementation demonstrated a dramatic improvement in computational efficiency, with relocation taking on the order of a few seconds for small environments and a few tens of seconds for somewhat larger environments. Time constraints prevented a careful and exhaustive analysis of the algorithm's performance in a wide range of environments. Work in progress by the MIT Marine Robotics is anticipated to bring this work to a more complete conclusion, through integration of relocation with CML.

5.2 Future Research

Our work on this problem has generated a wealth of ideas for future research. We recommend three key ideas that should be pursued for future research: (a) integration with CML, (b) multi-vantage point relocation, and (c) generation of "saliency maps".

First, it is vital to integrate relocation with a CML algorithm to free the technique from the tedium of generating hand-measured maps. As discussed in Chapter One, the goal of CML is to enable a mobile robot to build a map of an unknown environment while using that map to navigate. Clearly, relocation has a very important role to play in CML for error recovery when the robot gets lost. As CML algorithms are deployed in more and more situations, we expect relocation to be a very valuable capability.

Second, it is very challenging to attempt relocation using sonar data from just a single vantage point. In the future, methods should be explored for relocation using data from multiple vantage points. For example, a CML algorithm could be applied to a small segment of data, to build a small local map. Subsequently this local map could be matched against the global map to perform relocaiton. This should make it easier to estimate the robot position in situations with high symmetry. An alternative to building a local map might be to perform relocation directly with sonar returns from multiple vantage points using delayed decision making techniques developed for CML [15].

Finally, it should be very interesting to combine relocation with adaptive motion control [9] to direct the robot, when it gets lost, to try to find unique, distinctive locations where relocation has a much higher chance of success. An off-line process could be executed in which a simulation is performed for many randomly generated positions throughout the environment. One could make a "saliency" map which highlights the best areas of an environment from a relocation perspective. This could be integrated with a path planning algorithm [12] to steer the robot along routes where it is least likely to get lost.

Bibliography

[1] W. Au. *The Sonar of Dolphins.* New York: Springer-Verlag, 1993.

[2] B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):560–569, June 1990.

[3] O. Bozma and R. Kuc. Building a sonar map in a specular environment using a single mobile transducer. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(12), December 1991.

[4] O. Bozma and R. Kuc. Characterizing pulses reflected from rough surfaces using ultrasound. *J. Acoustical Society of America*, 89(6):2519–2531, June 1991.

[5] J. A. Castellanos. *Mobil Robot Localization and Map Building: A Multisensor Fusion Approach.* PhD thesis, University of Zaragoza, Spain, 1989.

[6] F. Dellaert, D. Fox, and and S. Thrun W. Burgard. Monte carlo localization for mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1999.

[7] M. Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Patten Analysis and Machine Intelligence*, PAMI-9(2):325–332, March 1987.

[8] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[9] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Int. J. Robotics Research*, 18(7):650–668, July 1999.

[10] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, 1990. (With contributions from T. Lozano-Perez and D. P. Huttenlocher).

[11] R. Kuc and M. W. Siegel. Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(6):766–778, November 1987.

[12] J-C. Latombe. *Robot Motion Planning.* Boston: Kluwer Academic Publishers, 1991.

[13] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Boston: Kluwer Academic Publishers, 1992.

[14] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D Koditschek and J. Hollerbach, editors, *Robotics Research: The Ninth International Symposium*, pages 169–176, Snowbird, Utah, 2000. Springer Verlag.

[15] J. J. Leonard and R. Rikoski. Incorporation of delayed decision making into stochastic mapping. In D. Rus and S. Singh, editors, *Experimental Robotics VII*, Lecture Notes in Control and Information Sciences. Springer-Verlag, 2001.

[16] J. H. Lim and J. J. Leonard. Mobile robot relocation from echolocation constraints. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(9):1035–1041, September 2000.

[17] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1985.

[18] P. M. Morse and K. U. Ingard. *Theoretical Acoustics.* New York: McGraw-Hill, 1968.

[19] A. C. Shultz and W. Adams. Continuous localization using evidence grids. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2833–2839, 1998.

[20] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[21] O. Wijk. *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.