Directed Sonar Sensing for Mobile Robot Navigation

John J. Leonard
Sea Grant College Program
Massachusetts Institute of Technology
Room E38-308
292 Main Street
Cambridge, MA 02139
U.S.A.

Hugh F. Durrant-Whyte
Department of Engineering Science
University of Oxford
Parks Road, Oxford
OX1 3PJ
United Kingdom

To our parents and families

Contents

4   Map Building   97

5   Simultaneous Map Building and Localization   129

6   Directed Sensing Strategies   147

7   Why Use Sonar?   157

A   Hardware and Software   163

List of Figures

List of Tables

List of Abbreviations and Symbols

EKF	extended Kalman filter
MHT	multiple hypothesis tracking
MMSE	minimum mean-square error estimation
NNSF	nearest neighbor standard filter
PDAF	probabilistic data association filter
RCD	region of constant depth
TVG	time variable gain amplifier

$\mathbf{a}_s(k)$	position vector of sensor s in the global frame
$\mathbf{b}_s(k)$	position vector of sensor s in the local (vehicle) frame
$\mathbf{C}_{Rt}(k \mid k)$	cross-covariance for vehicle position and target t
$\mathbf{C}_{ij}(k)$	cross-covariance for targets i and j
$c_t(k)$	credibility of target t
$\nabla \mathbf{f}$	state transition Jacobian
g	validation gate "number of sigmas"
$\nabla \mathbf{h}$	measurement Jacobian
$M(k)$	the current map (set of targets)
$m_s(k)$	number of returns in the scan produced by sensor s
n_O	number of observations
n_P	number of predictions
n_S	number of sonar sensors on the vehicle
n_T	number of targets
$\mathbf{P}(k_2 \mid k_1)$	vehicle position estimate covariance
$\mathbf{p}_t$	true target geometric parameterization (parameter vector)
$\hat{\mathbf{p}}_t(k)$	target state estimate
$\mathbf{Q}(k)$	plant noise covariance matrix
$\mathbf{R}(k)$	observation noise covariance matrix
$\mathbf{r}_i(k)$	return (a single sonar range measurement)
$r_s(k)$	observed range from sensor s
$\hat{r}_{st}(k)$	true range from sensor s to target t

$\mathbf{S}(k)$	innovation covariance
$S_s(k)$	sonar scan (a set of returns)
$\mathbf{u}(k)$	vehicle control input
$\mathbf{v}(k)$	plant noise
$\mathbf{W}(k)$	Kalman gain
$\mathbf{w}(k)$	measurement noise
$\mathbf{x}(k)$	vehicle position state vector
$\hat{\mathbf{x}}(k_2 \mid k_1)$	vehicle position estimate
$\mathbf{x}_S(k)$	system state vector
$\hat{\mathbf{x}}_S(k_2 \mid k_1)$	system state estimate
$Z(k)$	the current set of observations
Z^k	the cumulative set of all observations
$\hat{Z}(k)$	the current set of predictions
$\mathbf{z}_j(k)$	RCD or observation
$\hat{\mathbf{z}}_i(k)$	prediction (predicted observation)
$\alpha_s(k)$	sensor orientation in the global frame
$\alpha'_s(k)$	sensor orientation in the local frame
β	target visibility angle or width of an RCD
δ_R	RCD threshold
$\delta\beta_s$	scan interval angle
$\phi_j(k)$	orientation of RCD j
$\phi_{st}(k)$	true bearing (azimuth) from sensor s to target t
$\mathbf{\Lambda}(k \mid k)$	system covariance matrix
$\mathbf{\Lambda}_t(k)$	target estimate covariance
$\nu(k)$	innovation

The integer k represents the current step of discrete time. The subscript i is used to index sets of returns and sets of predictions, j is used to index sets of observations, s is used to index the set of sensors and t is used to index the set of targets. The term $\hat{\mathbf{x}}(k_2 \mid k_1)$ signifies the estimate of the vector $\mathbf{x}$ at time step k_2 based on all information available up to and including time step time k_1. We use the notation $\mathbf{v} \sim \mathrm{N}(\mathbf{m}, \mathbf{P})$ to indicate that the random variable $\mathbf{v}$ is assumed to be Gaussian with mean $\mathbf{m}$ and covariance matrix $\mathbf{P}$ [54].

Preface

This monograph is a revised version of the D.Phil. thesis of the first author, submitted in October 1990 to the University of Oxford.

This work investigates the problem of mobile robot navigation using sonar. We view model-based navigation as a process of tracking naturally occurring environment features, which we refer to as "targets". Targets that have been predicted from the environment map are tracked to provide vehicle position estimates. Targets that are observed, but not predicted, represent unknown environment features or obstacles, and cause new tracks to be initiated, classified, and ultimately integrated into the map.

Chapter 1 presents a brief definition of the problem and a discussion of the basic research issues involved. No attempt is made to survey exhaustively the mobile robot navigation literature—the reader is strongly encouraged to consult other sources. The recent collection edited by Cox and Wilfong [34] is an excellent starting point, as it contains many of the standard works of the field. Also, we assume familiarity with the Kalman filter. There are many well-known texts on the subject; our notation derives from Bar-Shalom and Fortmann [7].

Chapter 2 provides a detailed sonar sensor model. A good sensor model is a crucial component of our approach to navigation, and is used both for predicting expected observations and classifying unexpected observations. Kuc and Siegel first reported that in a specular environment, sonar data should take the form of circular arcs in Cartesian coordinates. We call these features *regions of constant depth* (RCDs), and show how real sonar data, acquired with the Polaroid ranging system, can be both predicted and explained in terms of these features.

Chapter 3 presents an algorithm for model-based localization that is based on an extended Kalman filter (EKF) that utilizes matches between observed RCDs and RCDs predicted from an *a priori* map to update position. The algorithm has been implemented on several different robots, using real data. Localization runs in two different laboratories are presented.

In Chapter 4, the discussion turns to the map building problem. We view map building as a process of initiating tracks for events unexplainable

in terms of the environment map. Experimental results are obtained for the restricted case of learning an unknown room from precisely known vehicle locations. A simple office scene is mapped to sub-centimeter accuracy.

Chapter 5 presents a unified approach to navigation, in which the multiple requirements of localization, obstacle avoidance, and map building can be simultaneously addressed in a common multitarget tracking framework. Localization while learning and map maintenance in a dynamic environment are discussed.

Chapter 6 describes the concept of directed sensing. The slow data acquisition speed of acoustic sensing makes practical sonar-based navigation difficult to achieve. To overcome this problem, we believe that directed sensing strategies can be used to achieve fast, continuous operation. By tracking environment targets as the vehicle moves, less data needs to be acquired, and the need to re-solve the correspondence problem at each iteration of the perception cycle is obviated.

Chapter 7 assesses sonar's potential in comparison with other range sensing modalities and discusses an agenda for future research.

Appendix A provides hardware and software details for the implementations.

•

Chapter 1

Introduction

Although the robots we see in science fiction movies appear to navigate with effortless precision, in reality mobile robot navigation is a difficult research problem. Indeed, simply answering the question "where am I?" in truly autonomous fashion is a serious challenge for today's mobile robot. We use the term *localization* to refer to the process of determining a robot's position using information from external sensors. Our research has focused on the provision of this capability using airborne ultrasonic range sensing, which shall henceforth be referred to as *sonar*.

1.1   The Navigation Problem

Stated most simply, the problem of navigation can be summarized by the following three questions: "where am I?", "where am I going?", and "how should I get there?" The first question is one of localization: how can I work out where I am in a given environment, based on what I can see and what I have previously been told? The second and third questions are essentially those of specifying a goal and being able to plan a path that results in achieving this goal. Investigations of the latter two questions usually come under the domain of path planning [86] and obstacle avoidance [127], [14]. In this book, we are principally concerned with the first, localization, question, and maintain that finding a robust and reliable solution to this problem is an essential precursor to answering the remaining two questions.

The problem of position determination has been of vital importance throughout the history of humanity [75]. The basic process of distance measurement, correlation, and triangulation was known to the Phoenicians, who successfully managed to build and maintain quite accurate maps of the Mediterranean area. Today, navigation is a well-understood quantitative science, used routinely in maritime, aviation, and space applications. Given

this, the question must be asked why robust and reliable autonomous mobile robot navigation remains such a difficult problem. In our view, the reason for this is clear. It is not the navigation process *per se* that is a problem—it is the reliable acquisition or extraction of information about navigation beacons, from sensor information, and the automatic correlation or correspondence of these with some navigation map that makes the autonomous navigation problem so difficult.

Implementing a navigation system that uses artificial beacons together with sensors that provide accurate and reliable measurements of beacon location is a straightforward procedure used by many commercial robots today. For example, the GEC Caterpillar automatic guided vehicle (AGV) uses a rotating laser to locate itself with respect to a set of bar-codes that are fixed at known locations through the AGV's environment. More recently, TRC Corporation have developed a localization system for the HelpMate$^{\text{TM}}$ service robot that provides 24-hour operation in hospitals and offices, using retro-reflective strips mounted on the ceiling as navigation beacons [128]. The goal of our research is to achieve comparable performance to artificial beacon systems without modifying the environment, by sensing the naturally-occurring geometry of typical indoor scenes, and comparing these observations with a map of the environment. This *competence of localization* would provide a mobile robot with the ability to determine its position without artificial help, such as bar-code markers on the walls.

While a limited form of this competence might use hand-measured maps of the environment provided *a priori* to the robot, completely autonomous operation will require that the robot construct and maintain its own map in a changing environment. For the moment, we neglect the more difficult problem of dynamic map building and maintenance [87], and state our goal as the development of an autonomous system that can

1. build a large-scale, metrically accurate map of a static, people-free environment and

2. use this map for localization at arbitrary locations within the environment.

One test of such a capability would be the following experiment:

The weekend experiment: The scene is a typical office building. At 5:30 pm on a Friday night, we power-up the robot in an arbitrary starting position with a blank memory—i.e. no *a priori* map. We tell the robot "use the next few days to travel at your leisure around your environment to make a map of the accessible space on this floor of the building, without harming yourself or the environment." We turn off the lights and leave. On Monday morning, we return to find the robot back at its starting position, ready to execute commanded trajectories

quickly, using its learned map for accurate position determination at arbitrary locations in the environment.

To our knowledge, this test is beyond the capability of any robot in existence today, using any sensing modality.

1.2   Why Use Sonar?

One motivation for using sonar for mobile robot navigation comes from the impressive ultrasonic sensing capabilities of bats, who rely on echolocation to determine their position and to hunt their prey [132]. Bats use frequency modulation, Doppler, and other ultrasonic ranging techniques that are far more sophisticated than the simple time-of-flight (TOF) ranging approach investigated in the experiments reported here. The use of time-of-flight sonar in robotics has received a great deal of effort, but despite some successes, results have generally been disappointing. For this reason, many feel that more sophisticated sonar devices that incorporate frequency modulation and amplitude-based interpretation are necessary, or that optical alternatives must be used, in order to achieve autonomous navigation. More sophisticated sonar techniques would of course be welcome, and the increasing capabilities and decreasing cost of optical rangefinders cannot be denied. Nevertheless, we believe that our experimental results provide evidence that time-of-flight sonar can in fact fulfill the perception role for autonomous navigation, in a large class of application environments.

1.3   Choosing a Representation

Choosing a representation is perhaps the single most important step in sensor data interpretation, a point stressed by Stewart:

> The type of representation we use determines what information is made explicit in the model; the purposes for which a model can be used and the efficiency with which those purposes can be accomplished follow directly from the choice of representation. [123]

Stewart provides a comprehensive discussion of the considerations involved in choosing a representation [123]. For his application domain of underwater multisensor modeling, he advocates a cellular, volumetric representation, analogous to the *certainty grid* that was first proposed for sonar-based navigation by Moravec [100]. The concept was further developed by Elfes using a Bayesian methodology to become the *occupancy grid* [45].

The occupancy grid divides free space into a discrete, two or three dimensional grid of cells. Each cell is assigned a single value between 0 and 1 to represent the probability that the cell is unknown, occupied, or free space. Using a probabilistic sensor model, this value is increased or decreased as

new sonar returns are obtained from sensors pointed in the direction of the cell. Experimental results have been presented for a broad range of tasks including path planning, obstacle avoidance, motion solving, and sensor fusion [46], [47]. A significantly different grid-type representation, called the *vector field histogram*, has recently been developed by Borenstein and Koren for the purpose of very fast obstacle avoidance [14].

The primary alternative to a grid-based approach is the use of a geometric, *feature-based* representation, in which the environment is modeled by a set of geometric primitives such as points, lines, and planes. The first step in interpretation is the extraction of geometric features from raw sensor data. Several crucial issues are the representation of uncertainty [43], the reliability of feature extraction, and the speed with which the model can be constructed. Crowley developed one of the earliest feature-based approaches to sonar, introducing the concept of the *composite local model* [37]. The composite local model is built by extracting straight line segments from sonar data, and is matched to a previously stored global line segment map to provide localization.

In this work, we shall utilize a geometric representation and will rely upon the following fundamental assumption:

> **The 2-D assumption**: we assume the actual three dimensional geometry of the environment is orthogonal to the horizontal plane in which the robot moves.

An assumption of this magnitude is made with a certain degree of reluctance. For example, there is a good chance that the reader is sitting in a chair that is quite three-dimensional. We cannot hope for any success with amorphous, natural terrain, such as the bottom of the ocean or the surface of Mars. However, we believe that the above assumption can be satisfied to a sufficient degree in practice to permit effective navigation in a wide range of man-made environments.

Based on this assumption, we shall represent the surface geometry of the environment in terms of three basic primitives: points, lines, and arcs, as will be described in more detail in the next chapter.

Grid-based representations make weaker assumptions about the environment than our geometric approach, and hence have a wider domain of application. Also, computational requirements will be much less sensitive to environment complexity. However, our contention is that they are less powerful than geometric models for the purpose of dynamic, globally referenced position estimation. For the fastest and most accurate operation possible, we argue that it is necessary to be able to predict the values of individual sonar returns. We believe that this is not possible using the occupancy grid, because there is no explicit characterization of the geometry of the reflecting surfaces of the environment. It does not capture the physics involved.

We must emphasize, however, that choosing a representation is a subjective endeavor, at least at this early stage of development. A proponent of grid-based representations could dispute the assertions of the previous paragraph, with some justification. An objective comparison between discrete and geometric approaches, using the same robot in a variety of environments, would be a valuable contribution to the robotics community. Unfortunately, such a comparison will probably have to wait until genuine long-term autonomy—self-localization using autonomously learned maps—has been conclusively demonstrated with each approach.

1.4   The Kalman Filter

The basic tool of our approach to navigation is the extended Kalman filter (EKF). A good source of historical and intuitive insight into the subject is the work of Sorenson, which traces the development of least squares estimation techniques from Gauss to Kalman [117]. Kalman filtering techniques have been used extensively in location estimation problems such as missile tracking and ship navigation [119]. There have been many notable applications of the EKF in mobile robot systems. For example, Dickmanns uses an EKF in a real-time vision system that achieves autonomous road-following at speeds over 80 km/hour [39]. Ayache [5], Harris [64], Matthies [93], and Kriegman [78] have used the EKF for visual model building and motion estimation. Smith, Self and Cheeseman have applied the EKF to the problems of map-making and position estimation [115]; we will discuss their work at length in Chapter 5.

The use of the Kalman filter in the robotics community has been somewhat confusing, as pointed out by Hager:

> The true contribution of the Kalman filter, as it was initially stated and derived by Kalman and others, was the efficient computation of a MMSE problem *with a dynamic system* describing variations in the unknown parameters. Much of the [robotics] literature which refers to Kalman filters is in fact using MMSE techniques for estimating *static* parameters—a result dating back to Gauss. [61]

When considering Kalman filtering, understanding notation is more than half the battle. The notation we use in this monograph derives from Bar-Shalom [7], a good introduction to the EKF and multitarget tracking. Other commonly referenced textbooks on the subject include Gelb [54], Jazwinski [73], and Maybeck [94]. Following Bar-Shalom, we use the term $\hat{\mathbf{x}}(k_2 \mid k_1)$ to designate the estimate of the vector $\mathbf{x}$ at time step k_2 given all observations up to time step k_1. An alternative often seen in the literature is the tilde-caret notation [62], in which $\tilde{\mathbf{x}}(k)$ denotes the *prediction* of the state vector $\mathbf{x}(k)$ based on information available strictly before time k, and $\hat{\mathbf{x}}(k)$ denotes the *estimate* of the state vector $\mathbf{x}(k)$ based on information

available up to and including time k. Gelb uses yet another convention, in which $\hat{\underline{\mathbf{x}}}_k(-)$ denotes the predicted state at time k and $\hat{\underline{\mathbf{x}}}_k(+)$ denotes the updated (estimated) state at time k [54].

The Kalman filter relies on two models: a plant model and a measurement model. The plant model describes how the position of the vehicle changes in response to motor control inputs, with an estimate of dead reckoning errors. The measurement model makes explicit the information provided by a sensor measurement in terms of the current vehicle position and the positions of geometric features in the environment, with an estimate of measurement noise. These concepts will be described with more precision in the chapters that follow.

Figures 1.1 through 1.4 illustrate the application of the EKF for a simulated robot vehicle, equipped with a single servo-mounted sonar, moving in a two-dimensional world of specular planar surfaces. Figure 1.1 is a simulation in which no planar surfaces are present, showing how position uncertainty grows in accordance with the system plant model if no sensor observations are available. Figure 1.2 shows the use of range measurements to a wall to update position. As will be discussed in Chapter 2, the only part of a smooth wall that can be "seen" by a sonar sensor is the portion of the wall that is perpendicular to the incident sonar beam. Thus, uncertainty grows as before for the first two cycles of the filter. After the wall comes into view, range measurements to the wall provide a position update perpendicular to the wall, while uncertainty continues to grow in the parallel direction. Figures 1.3 and 1.4 illustrate position estimation from multiple walls. As different walls come in and out of the sonar's view, uncertainty is reduced in different degrees of freedom.

Caution must be taken in using the Kalman filter and its non-linear extensions for the following reasons:

- Are the assumptions of the filter met in practice?

- How are values chosen for the noise sources, validation gate size, and other parameters?

- Divergence of the EKF. The system can become overconfident, and hence the estimate covariance $\mathbf{P}(k \mid k)$ becomes too small. In traditional estimation problems, this has the result that new measurements are given too little weight. In our application, because of the validation gate approach we use to reject outliers, when the filter diverges we will reject correct measurements. This is a crucial issue in the design of an actual working system.

It is important to remember that for the EKF, the state covariance matrix $\mathbf{P}(k \mid k)$ is only an approximate mean square error, not a true covariance [7].
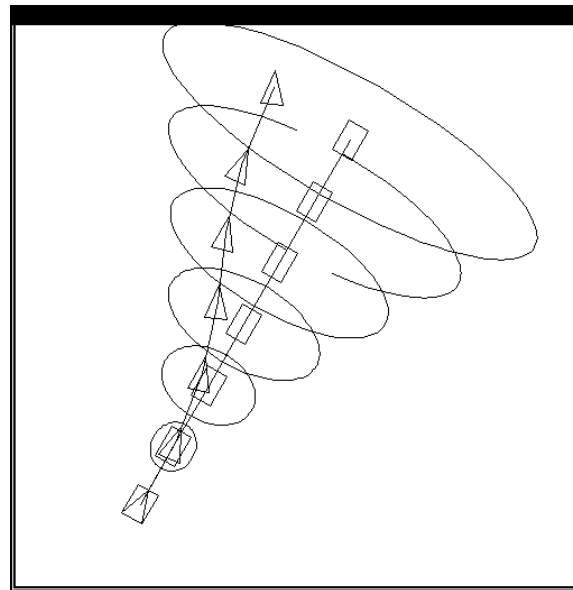
Figure 1.1: A simulated run with no visible beacons. The triangle represents the actual vehicle position and orientation $(x(k), y(k), \theta(k))$, the rectangle represents the estimated vehicle position and orientation, and the ellipse represents the confidence in the estimates of $x(k)$ and $y(k)$.
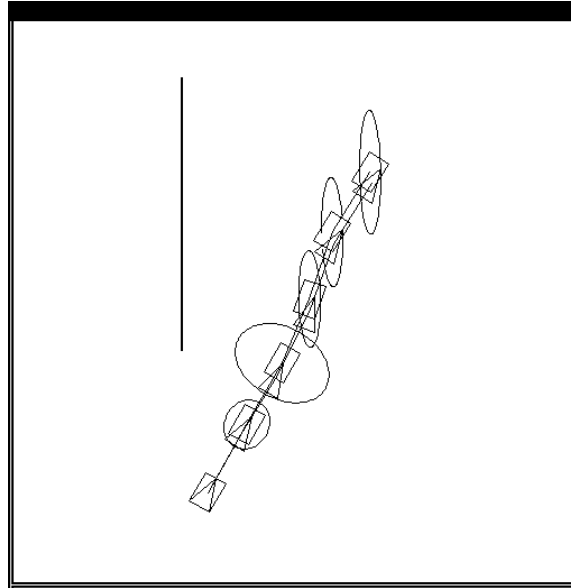
Figure 1.2: A simulated run taking observations of a single wall beacon. After the wall comes into view, the error ellipse shrinks perpendicular to the wall as *a posteriori* confidence in the estimate of $x(k)$ and $y(k)$ increases.
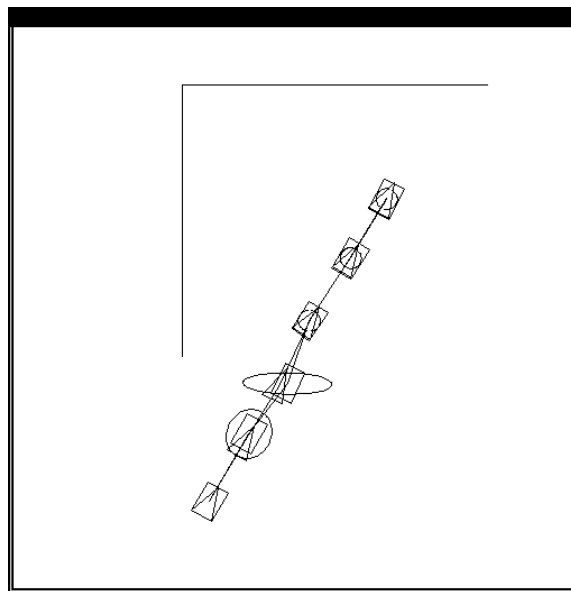
Figure 1.3: Localization from one, then two wall beacons.
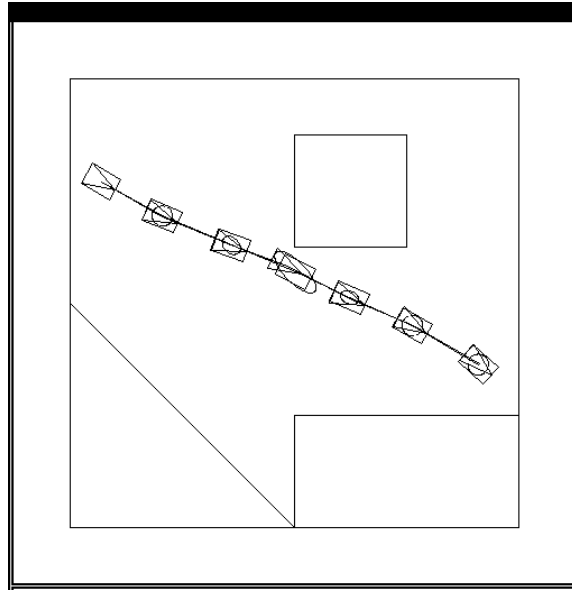
Figure 1.4: Localization from a sequence of wall beacons.

The Kalman filter itself is not the representation. It is simply an engine for recursively computing state estimates of interest. It requires that the value produced by a sensor can be expressed with an *equation*, as a function of vehicle position, target state, and measurement noise. In our approach to navigation, for a wall target the EKF is useful only in estimating the infinite line parameters R and θ. The endpoints of a line segment target cannot be directly estimated from observations of the wall, as the measurement model becomes an *inequality*. One selling point for sonar is that we can directly observe the corners and edges that populate many environments, and estimate their locations directly with the EKF. To produce the map building results of Chapter 4, line segment endpoints are computed by projecting new observations onto each new infinite line estimate. From our limited experience processing infrared rangefinder data, we have found that estimating line segment endpoints can be a difficult issue using an optical modality.

The enthusiasm with which we embrace the Kalman filter has been criticized as approaching religious fervor. We counter that it is a *tool* that we find too useful to do without. The invaluable role of the EKF in so many demanding applications, such as the Voyager spacecraft missions to Jupiter and beyond [26], supports our position. Some of these implementations model the physical system with extreme detail. For example, the Voyager navigation filter used a plant model with 67 states [26]. We cannot

yet approach this level of rigor in modeling because of the pervasive data association problem that we have to tackle first.

1.5   Data Association

A fundamental requirement of perception is the task of *achieving* and *maintaining correspondence*. The correspondence problem (also called the data association or segmentation problem) is what makes perception and sensor-based control different from traditional estimation and control problems. The crucial difference is uncertainty in the origin of measurements. The key to using a measurement from a sensor is knowing "what is being measured?" For example, the sensory correspondence issue is absent from the problem of the end-point control of a robot manipulator, because sensory inputs take the form of shaft encoders, velocity sensors, etc. Extend the problem to include the manipulation of imprecisely located objects with this robot arm, and the character of the problem changes completely. External sensing of some kind, e.g., computer vision, is needed to determine the position of the object. Successful use of sensing requires overcoming the correspondence problem.

Dynamic perception refers to the use of sensing in a fast, continuous cycle, and has been implemented in several notable robotic systems. Andersson has built a robot that can play ping-pong [3]. Dickmanns has achieved highway navigation at speeds over 80 km/hour [39]. These two systems are impressive because of their incorporation of sensing in a real-time control loop. However, note the absence of the sensory correspondence problem from "the loop". Andersson's vision system tracks a single white ping-pong ball. Initial experiments relied on a dark background but this restriction was subsequently lifted. In Dickmanns' system, region-of-interest windows maintain continual correspondence with lane markings and road boundaries.

The aim of *directed sensing* is to separate the correspondence problem from the subsequent estimation and control problem, through the use of tracking sonars. The basic low-level competence is the focus of attention. This is used to maintain correspondence with individual environment features during the vehicle or sensor's motion. By focusing attention on a feature, correspondence can be easily maintained once it has been achieved. By tracking a given environment feature, a high bandwidth stream of correctly associated measurements to this feature becomes available. The success of artificial beacon systems (and aerospace and maritime navigation in use all over the world) demonstrates the straightforward nature of position estimation if the correspondence problem is solved.

We envision a robot that maintains *continuous map contact*, almost effortlessly gliding through its environment, "grabbing hold" of corners,

planes, and cylinders in the environment, using them as handrails. Rather than using sensing every now-and-then to answer "where am I?", by maintaining continual correspondence with some subset of map features, perception can be put "in the loop" to provide high bandwidth position control.

1.6 Overview

We believe that mobile robot navigation can be cast as a multitarget tracking problem. The targets to be tracked are stationary features of the environment. Building a map is the task of estimating the locations of unknown targets in the environment—a process of track initiation. Using the map for localization is a two stage process of 1) determining the correspondence between current sensor measurements and the targets in the map and 2) using this correspondence to update the robot's position. The challenge of truly autonomous navigation lies in the fact that map building and localization must be undertaken simultaneously.

Multitarget tracking provides a framework in which to accomplish these competing objectives. The following chapters present the three basic components of this framework in detail. We begin in Chapter 2 by developing a sonar sensor model that can be used to extract and interpret the information conveyed by sonar data. Chapter 3 describes a model-based localization algorithm that tracks expected events to determine the robot's position. Several implementations of this algorithm are presented. Chapter 4 deals with the bottom-up interpretation of unexpected observations to provide obstacle detection and map building capabilities. Experimental results demonstrate successful map building with real sonar data.

We then proceed in Chapter 5 to bring the results of the previous chapters together to describe a unified navigation framework. Unfortunately, this combined system remains in development—we cannot yet, for example, show results for model-based localization using an autonomously learned model. (Our experiments in Chapter 3 demonstrate a limited form of a competence of localization by using hand-measured environment maps that are provided *a priori* to the robot.) As we state elsewhere, we believe this is the crucial capability that will in the long term judge the success or failure of the approach. Further, success in real applications will require the ability to deal with a dynamic environment, necessitating an inference procedure in which missed detections provide information about state changes of targets in the map. Some results from our recent collaboration with Ingemar Cox in this area will be summarized in this chapter.

To combat the slow data acquisition speeds of acoustic sensors, Chapter 6 describes the application of directed sensing strategies. By tracking environment targets as the vehicle moves, less data needs to be acquired,

and the need to re-solve the correspondence problem at each iteration of the perception cycle is obviated.

Finally, Chapter 7 concludes this work by assessing sonar's potential in comparison with other sensors, such as the phase-based infrared rangefinder [99], and discussing some topics for further research.

Chapter 2

A Sonar Sensor Model

The key to using any sensor is to have a good sensor model [42]. Our goal in developing a sensor model for sonar is to support two capabilities:

- **Prediction**: what data should the sensor produce when observing a known scene from a given position?

- **Explanation**: given observed sensor data, what is the geometry of the scene that produced the data?

2.1   Introduction

Sonar stands for SOund NAvigation and Ranging, and was initially developed for underwater applications [130]. In this book, we use the term sonar to describe airborne ultrasonic range sensing. We are concerned with monostatic ranging systems, in which a single transducer acts as both transmitter and receiver. Range measurements are obtained exclusively via time-of-flight (TOF) information. We assume that echo amplitude information is not made available by the sensor. We do not consider frequency modulation techniques (also known as "chirping") which have been developed for underwater and airborne applications [58]. The sonar device we have used is the Polaroid ultrasonic ranging system [109], the device most commonly used by the robotics research community. A description of the properties of airborne ultrasonic sensing and details of the Polaroid ultrasonic ranging system are given in Appendix A.

The theory behind our approach to sonar interpretation follows directly from the published works of Hallam [62], Brown [22], and Kuc and his associates [83], [81], [8]. The contribution that we feel we have provided is to bridge the gap between prediction and explanation by making clear the differences between theory and real Polaroid sonar data. Our conclusions

have been the result of acquiring, displaying, and processing a large amount of sonar data (over 100,000 range measurements) acquired in a variety of indoor scenes, with several different robots, over three years. This process has been gradual. Some effects repeatedly observed in the initial stages of our research were not sufficiently explained until two years later. As a result of this effort, we have developed the conviction, contrary to popular belief, that sonar is in fact a very good sensor. The aim of this chapter is to argue why we feel sonar's "bad reputation" is undeserved, and to provide a framework in which real sonar data can be both predicted and explained.

2.2   Previous Work

Mobile robot navigation using sonar has been the subject of a large amount of research over the last decade. Table 2.1 lists many of the research projects which have used sonar. We shall refer to many of these projects through the rest of this work.

Initially, sonar was heralded as a cheap solution to the mobile robot sensing problem, because it provides direct range information at low cost. However, despite some successes, most would agree that the performance of sonar has been disappointing. The general conclusion of most previous work is that sonar is plagued by two problems: beam width and specularity. Figure 2.1 shows a real sonar scan taken in an uncluttered office scene. Figure 2.2 shows the same scan superimposed on a hand-measured map of the room. By comparing these two figures, one can see why researchers have made the following comments about sonar:

1. Ultrasonic sensors offer many shortcomings ... 1. poor directionality that limits the accuracy in determination of the spatial position of an edge to 10-50 cm, depending on the distance to the obstacle and the angle between the obstacle surface and the acoustic beam 2. Frequent misreadings ... 3. Specular reflections that occur when the angle between the wave front and the normal to a smooth surface is too large. [13]

2. Ultrasonic range data are seriously corrupted by reflections and specularities. [38]

3. ... the use of a sonar range finder represents, in some sense, a worst case scenario for localization with range data. [40]

Comments like this abound in the literature on this subject. Sonar has been widely characterized as problematic, unpredictable and unreliable.

0.5 meters
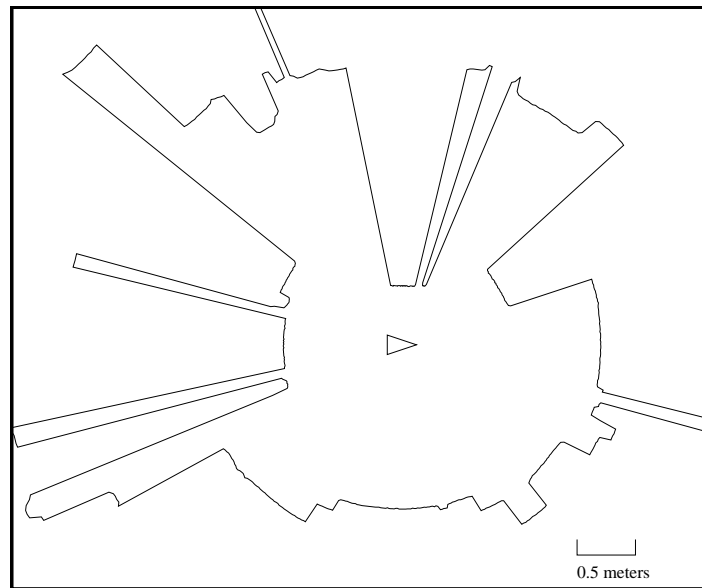
Figure 2.1: A typical sonar scan, which we shall refer to as scan b0.

0.5 meters
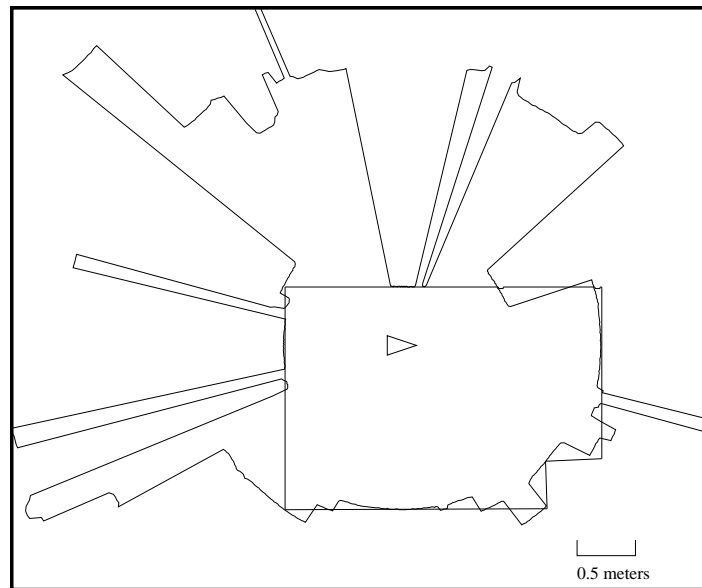
Figure 2.2: The same scan, superimposed on a hand-measured room model.

Ref.	Author(s)	Sonars	Key Features
[8]	Barshan & Kuc	Linear array	Distinguish corners and planes
[9]	Beckerman & Oblow	Phased array; $\delta\beta = 15°$	Elimination of systematic errors in grid-type map building
[12][13] [14]	Borenstein & Koren	24 F	Very fast obstacle avoidance; vector field histogram method
[18]	Brooks	12 F	Behavior-based control
[22][24]	Brown	1 R	Surface tracking and recognition
[29][28]	Chatila	14 F	Obstacle avoidance and wall-following
[37]	Crowley	1 R; $\delta\beta = 3°$	Local composite model
[38]	Crowley	24 F	EKF mapping and localization
[40]	Drumheller	1 R; $\delta\beta = 3.6°$	Search-based localization (relocation); sonar barrier test
[45][47] [101]	Elfes & Moravec	24 F	Occupancy grid mapping and localization
[53]	Flynn	1 R; $\delta\beta = 1.4°$	Sonar-IR sensor integration
[56][49]	Gilbreath *et al.*	35 F, 1 R	Grid-type mapping
[55]	Gex & Cambell	3 F, 4 R	Line segment based mapping; intelligent sensor moves
[62]	Hallam	Narrow beam underwater	Motion resolution; specular event analysis
[69]	Hu *et al.*	12 R	On-the-fly obstacle avoidance
[77]	Kriegman *et al.*	12 F	Wall extraction; good illustration of returns from corners
[83]	Kuc & Siegel	1 R; $\delta\beta = 2°$	Sonar simulation model
[82][81]	Kuc	1 R	Spatial sampling criterion for collision prevention
[85]	Lang *et al.*	24 F	Experimental sensor modeling
[92]	Mataric	12 F	Environment learning using a subsumption architecture
[108]	Noborio *et al.*	1 R; $\delta\beta = 15°$	Quadtree map building
[120]	Steer	9 F	Position estimation from walls
[123]	Stewart	Underwater	Multisensor modeling
[131]	Walter	30 F	Obstacle avoidance
[135]	Zelinsky	24 F	Grid-type map building

Table 2.1: A listing of representative previous work in sonar-based navigation. We use the letter **R** to refer to a rotating (servo-mounted) sonar, and the letter **F** to refer to a fixed (static) sonar.

2.3 Terminology

To make the ensuing discussion more precise, we first define the notation and terminology we use to represent sonar data. Under consideration is a robot vehicle with n_S sonar sensors. The position and orientation with respect to the local vehicle coordinate frame of sonar sensor s at time k is the known vector $\mathbf{b}_s = (x'_s, y'_s, \alpha'_s(k))$. A sonar sensor is considered to be either *static*, in which case $\alpha'_s(k)$ takes a constant value, or *servo-mounted*, in which case $\alpha'_s(k)$ takes a value in the range defined by two known limit angles α_1 and α_2 such that $\alpha_1 \leq \alpha'_s(k) \leq \alpha_2$. All servo-mounted sensors are assumed to rotate in the horizontal plane.

Each sonar range reading generates a *return*, which is an ordered pair consisting of the range value $r(k)$ and the sensor orientation $\alpha'(k)$. At time k, sensor s generates $m_s(k)$ returns. We define the vector $\mathbf{r}_i^s(k) = (r_i(k), \alpha'_i(k))$ as the ith return produced by sensor s at time k. Each static sonar generates at most one return per time step. For instances in which the vehicle is stationary, a servo-mounted sonar can be rotated to produce returns over a range of orientations; we shall consider these returns to occur at the same step k of discrete time. The set of returns produced by a sonar at time k is represented by the vector $\mathbf{s}_s(k) = \{\mathbf{r}_i^s(k) | 1 \leq s \leq m_s(k)\}$, which is termed the *scan* generated by sensor s at time k. The returns in a scan are ordered by orientation in increasing fashion such that any two consecutive returns differ in angle by the *scan interval* $\delta\beta_s(k)$ of the scan:

$$\alpha'_{i+1}(k) = \alpha'_i(k) + \delta\beta_s(k)$$

Two returns which differ in orientation by $\delta\beta_s(k)$ are said to be *adjacent*. A scan which covers a complete 360 degree rotation is termed a *complete scan*. A scan which is not complete is termed a *sector scan*. A set of consecutively obtained returns is termed a *connected* set of returns. Sector and complete scans obtained by rotating a servo-mounted transducer to produce two or more returns represent *densely sampled* data, while we refer to scans consisting of a single return from a static or servo-mounted sonar as *sparse* data. To link these concepts with real data, Figure 2.1 shows a densely sampled, complete scan from a servo-mounted sonar. This scan has $m_s = 612$ returns, yielding a scan interval angle of $\delta\beta_s = 0.588$ degrees. A small value of $\delta\beta$ signifies a high sampling density.

2.4 The Physics of Sonar

We are strong believers in the arguments advanced by Brown and Kuc, namely that the first step in using sonar must be to accept the realities of the physics of acoustic sensing:

> Ultrasonic range sensing has met with limited success in the past for
> several reasons. First, since the acoustic impedance of air is quite
> low (about 415 rayls) and since typical acoustic impedances of solid
> objects are much larger ... all solid surfaces appear as acoustic re-
> flectors. Also, because the acoustic wavelengths are generally quite
> long (on the order of 3-5 mm), most surfaces appear to be acoustic
> mirrors. Consequently, surfaces that are not orthogonal to the direc-
> tion of propagation reflect signal energy away from the source, and
> the surface will not be detectable. Of course, diffraction effects can
> occur, but for large object surfaces the effect is negligible. [22]

These fundamental characteristics of acoustic sensing explain clearly
why the sonar scan looks so completely unlike the room in Figure 2.2.
However, these simple facts have been ignored by many in the robotics
community. Kuc writes that

> problems arise in the straightforward, but naive, interpretation of
> time-of-flight (TOF) readings: objects that are present are not always
> detected and range readings produced by the TOF system do not al-
> ways correspond to objects at that range. Because of these problems,
> many researchers abandon sonar-only navigation systems .... [81]

The ideal range sensor in many people's eyes would be characterized by
a pencil-thin beam and high range accuracy independent of surface re-
flectance properties. We call this imaginary sensor the *ray-trace scanner*.
Scans from a simulated ray-trace scanner "look good", because they bear
a strong resemblance to the actual room map. However, acoustics dictates
that sonar scans will not have this property; sonar scans should "look bad".
We feel that many sonar interpretation algorithms have been designed with
the expectation that sonar should provide ray-trace scanner data [37], [40],
[13]. While we do not dispute the real experimental results of researchers
who have followed this approach, we feel their successes have been limited,
because of the lack of physical justification for treating a real sonar as a
ray-trace scanner plus a large Gaussian noise source. Once we accept that
sonar scans will usually bear little resemblance to the room in which they
were taken, we have to ask: how can we use sonar?

To be useful for navigation, we must be able to predict the data a
sensor will produce from a given position, using some map. When viewed
as a ray-trace scanner, sonar seems very unpredictable. However, Kuc and
Siegel provide strong evidence that sonar scans taken in everyday indoor
environments are indeed predictable [83]. The approach they advocate is
to characterize environment features as belonging to one of two categories:
"*reflecting surfaces* whose dimensions are larger than the wavelength, and
diffracting objects, whose dimensions are smaller than the wavelength" [81].
Huygen's principle is applied to generate predicted sonar data for corner,
wall, and edge targets. Real sonar scans are presented which verify the

predictions of their model. One key conclusion from their work is that corners and walls produce responses that cannot be distinguished from a single sensing location. The responses from these specular targets take the form of circular arcs in Cartesian coordinates—sequences of headings over which the range value measured is the distance to the target. Another conclusion they reach is that edges give rise to diffuse echoes that will be weaker in intensity than reflections from walls or corners.

2.5   Predicting Sonar Data

The following sections develop a simplified framework for predicting sonar data that follows, to a large extent, from the ideas of Kuc and Siegel's initial paper [83].

2.5.1   Target Models

We use the word *target* to refer to any environment feature which is capable of being observed by a sonar sensor. For our purposes, reflecting surfaces are considered *specular* targets, which produce specular returns, while diffracting objects are *diffuse* targets, and produce diffuse returns. We consider four types of target: planes, cylinders, corners, and edges. As stated in the previous chapter, we approximate the world as being two-dimensional, and represent the environment by a map of n_T targets. The geometry of target t is described by the target parameter vector $\mathbf{p}_t$, which takes a different form for each type of target:

1. Plane   (refer to Figure 2.3)
   A plane is a represented by a line in our two-dimensional target map, and is defined by the parameter vector $\mathbf{p}_L = (p_R, p_\theta, p_V)$. We represent the line in hessian normal form [6]: p_R is the minimum (perpendicular) distance from the (infinite) line to the origin of the global coordinate frame, and p_θ is the angle with respect to the x axis of a perpendicular drawn from the line to the origin. A plane is visible only from one side; an additional parameter p_V, which takes the value of 1 or -1, is used to indicate the half-plane of two-dimensional space from which the plane is visible.

2. Cylinder   (refer to Figure 2.4)
   A cylinder is represented by a circular arc in our two-dimensional target map, and is defined by the parameter vector $\mathbf{p}_{CYL} = (p_x, p_y, p_R)$ where p_x and p_y are the x and y coordinates of the center of the circle in global coordinates and p_R is the radius of the circle. The cylinder radius is assumed to be greater than the sonar wavelength. Cylinders are assumed to produce specular returns.
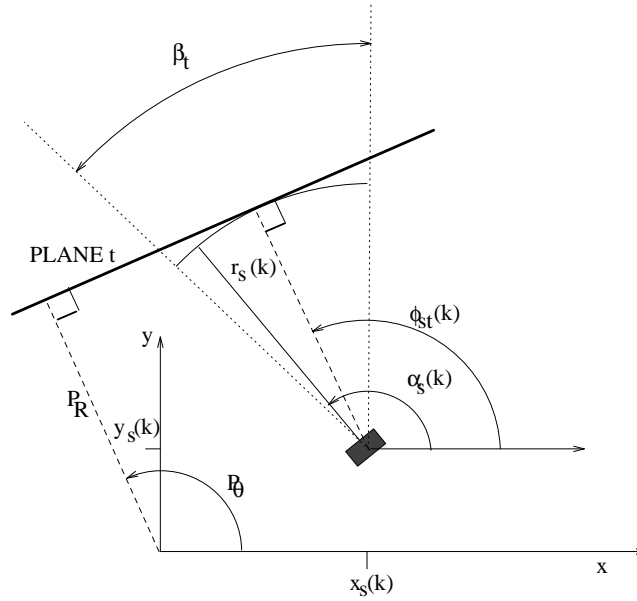
Figure 2.3: Plane target model. A plane is a line target specified by $\mathbf{p}_L = (p_R, p_\theta, p_V)$. The shaded rectangle indicates a single sonar sensor located at the position $\mathbf{a}_s(k) = (x_s(k), y_s(k), \alpha_s(k))$.
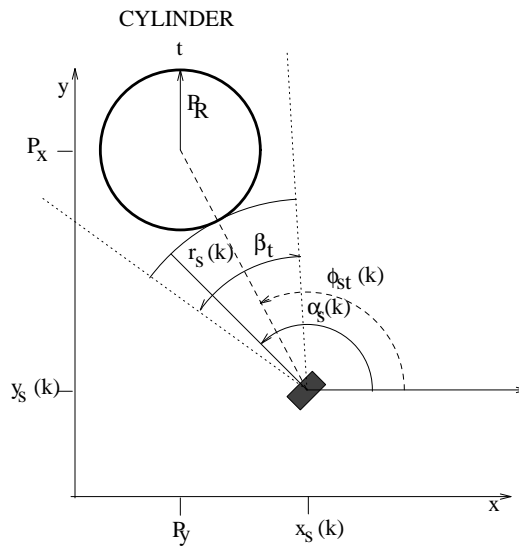
Figure 2.4: Cylinder target model. A cylinder is an arc target specified by $\mathbf{p}_{CYL} = (p_x, p_y, p_R)$.
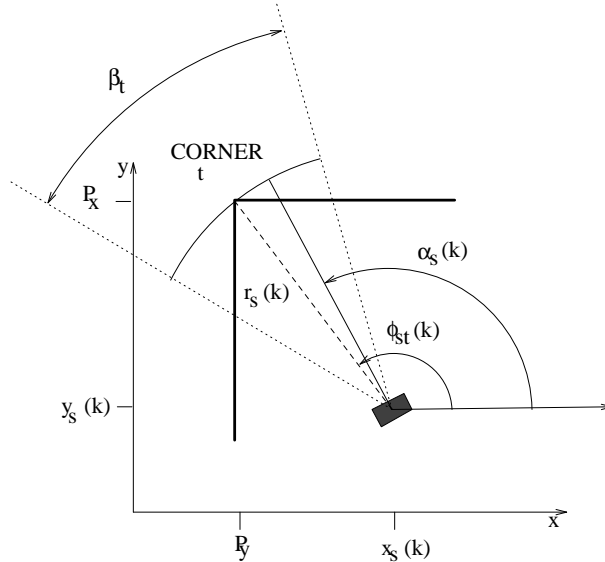
Figure 2.5: Corner target model. A corner is a point target specified by $\mathbf{p}_C = (p_x, p_y)$.

3. Corner   (refer to Figure 2.5)
   A corner is a concave dihedral, and produces specular returns. A corner is represented as a point in our two-dimensional map, and is defined by the parameter vector $\mathbf{p}_C = (p_x, p_y)$ where p_x and p_y are the x and y coordinates of the corner defined in global coordinates.

4. Edge   (refer to Figure 2.6)
   An edge is a convex dihedral. Edges produce diffuse reflections. Like a corner, an edge is represented by a point in our two-dimensional map, and is defined by the target state vector $\mathbf{p}_E = (p_x, p_y)$ where p_x and p_y are the x and y coordinates of the edge defined in global coordinates.

We shall refer to corners and edges as *point* targets, planes as *line* targets, and cylinders as *arc* targets. Somewhere between the realm of edges and cylinders fall slightly rounded convex edges with radius on the order of the sonar wavelength; the edges of some table legs and cardboard boxes often have this character. Our experience tells us that these produce weak specular echoes, but the determination of their exact nature is a complex process we leave to further research. As these reflections are weaker than echoes from concave corners, such features are treated as weak point targets in this book, the same as edges.
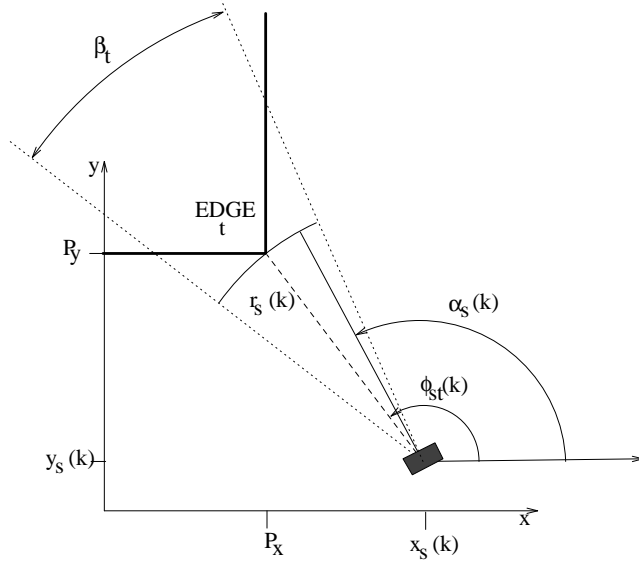
Figure 2.6: Edge target model. An edge is a point target specified by $\mathbf{p}_E = (p_x, p_y)$.

2.5.2 Predicted Target Responses

In a simplified adaptation of the Kuc and Siegel model, we describe a straightforward method here for generating predicted target responses for each of our four target types: planes, cylinders, corners, and edges. First, we define the position and orientation of sensor s in the global frame to be the sensor position vector $\mathbf{a}_s(k) = (x_s(k), y_s(k), \alpha_s(k))$. The vector $\mathbf{a}_s(k)$ is related to the local sensor position $\mathbf{b}_s(k)$ through the coordinate transformation defined by the vehicle position $\mathbf{x}(k)$. We define $\hat{r}_{st}(k)$ as the true range and $\phi_{st}(k)$ as the true bearing to target $\mathbf{p}_t$ from sensor s. We compute $\hat{r}_{st}(k)$ and $\phi_{st}(k)$ differently for each type of target using the target parameter vector $\mathbf{p}_t$. For planes,

$$\hat{r}_{st}(k) = p_V \left(p_R - x_s(k) \ \cos(p_\theta) - y_s(k) \ \sin(p_\theta) \right) \tag{2.1}$$

$$\phi_{st}(k) = p_\theta \tag{2.2}$$

where only positive values of $\hat{r}_{st}(k)$ are valid, corresponding to the visible side of the plane. For corners and edges,

$$\hat{r}_{st}(k) = \sqrt{(p_x - x_s(k))^2 + (p_y - y_s(k))^2} \tag{2.3}$$

$$tan(\phi_{st}(k)) = \frac{p_y - y_s(k)}{p_x - x_s(k)} \qquad p_x \neq x_s(k). \tag{2.4}$$

And for cylinders,

$$\hat{r}_{st}(k) = \sqrt{(p_x - x_s(k))^2 + (p_y - y_s(k))^2} - p_R \qquad (2.5)$$

$$tan(\phi_{st}(k)) = \frac{p_y - y_s(k)}{p_x - x_s(k)} \qquad p_x \neq x_s(k). \qquad (2.6)$$

A line target is *occluded* if a perpendicular drawn from the line to the sensor intersects another target. Likewise, an arc target is occluded if a perpendicular drawn from the arc to the sensor intersects another target, and a point target is occluded if a line drawn from the point to the sensor intersects another target.

In our simplified model, we assume that an unoccluded target t produces a return only if the following condition is satisfied

$$\phi_{st}(k) - \frac{\beta_t}{2} \leq \alpha_s(k) \leq \phi_{st}(k) + \frac{\beta_t}{2} \qquad (2.7)$$

where we define β_t as the *visibility angle* of target t. Our model predicts that when a target is unoccluded and Equation 2.7 has been satisfied, the range value $r_{st}(k)$ will approximately equal $\hat{r}_{st}(k)$, the true range from sensor s to target t. We now define the map more formally to be the set M of target parameter vectors $\mathbf{p}_t$ and visibility angles β_t for all n_T targets:

$$M = \{\mathbf{p}_t, \beta_t | 1 \leq t \leq n_T\}. \qquad (2.8)$$

Qualitatively, the value of β_t is determined by a target's ability to reflect acoustic energy—the stronger the target, the larger the target visibility angle. However, to compute the precise value of β_t in practice is a complex subject that is highly dependent on the sonar hardware. In practice, our algorithms place upper bounds on β_t, rather than try to calculate explicit values. We will discuss this subject further in the presentation of experimental results in subsequent chapters.

2.5.3   Generating Simulated Sonar Scans

Equations 2.1 to 2.7 provide a simple procedure for generating simulated sonar scans. Figure 2.7 presents an algorithm that uses these equations to predict the sonar scan produced by a servo-mounted sonar at the location (x, y) taking observations of an environment described by the map M with a scan interval $\delta\beta$. At a given sensor orientation, the range measurement produced is given by the nearest range of all unoccluded targets for which the angular visibility criterion of Equation 2.7 is satisfied. If all targets are occluded at a given orientation, a value MAX_ECHO which corresponds to the maximum range of the sonar system is produced. The result of applying

this algorithm with a value of $\beta_t = 30$ degrees for all targets to the same scene as Figure 2.1 is shown in Figures 2.8 and 2.9. A value of $\delta\beta = 0.588$ degrees was used to yield 612 returns, as in the real scan of Figure 2.1, and a maximum sonar range of MAX_ECHO $= 10$ meters was used. Note that the simulated scan is qualitatively similar to the real data of Figure 2.1.

2.6   Theory vs Practice

The sonar model proposed by Kuc predicts that in a specular environment comprised of planes, corners, and edges, sonar scans should be comprised of sequences of headings at which the range value measured is constant. In reality, the world presents a complex mixture of specular and diffuse targets. Given this fact, the next step is to try to reconcile the model for specular environments with real sonar data. Figures 2.10 and 2.11 show plots of range vs orientation for the scans of Figures 2.1 and 2.8, respectively. When viewed in this fashion, one can see that the dominant features of both the real and simulated data are horizontal line segments— sets of adjacent returns of nearly the same range. We call these sequences *regions of constant depth* (RCDs), which correspond to arcs in Cartesian coordinates.

While Figure 2.11 consists exclusively of horizontal line segments, and hence RCDs, Figure 2.10 does not. This difference can be seen by comparing Figures 2.12 and 2.13, which show histograms for the real and simulated scans. In the simulated scan, range measurements of only a few values are produced. These are the range values at which RCDs occur. For the real data, even though the dominant features of Figure 2.10 are horizontal lines, a wider spread of range values actually occurs. Because the real world is a mixture of diffuse and specular echo sources, a reasonable assessment of this situation might be that the RCDs are the specular target responses predicted by the theory, and that the other returns are the result of diffuse reflections not accommodated by the specular target model. We held this belief in the initial stages of our research, but let us examine the situation in more detail.

2.6.1   Typical Wall Responses

Figure 2.14 shows an increased resolution view of the response of the left-hand wall of Figure 2.1. This figure displays each return as a *TOF dot*, which is simply a dot placed along the sensor axis at the appropriate range value [83]. A ray is drawn from the sensor location to the TOF dot of every tenth return. Each return is numbered, using 0 to be the index of the on-axis ray. Looking solely at Figure 2.14, one can see the empirical justification for the method of fitting line segments in *Cartesian coordinates*

Algorithm 3.1 (Generation of a simulated sonar scan)

procedure predict_scan($x, y, \delta\beta, M$)
    $\alpha \leftarrow 0.0; \ i \leftarrow 0;$
    while $(\alpha < 2\pi)$
        $\mathbf{r}_i \leftarrow$ predict_return(x, y, α, M);
        $i \leftarrow i + 1;$
        $\alpha \leftarrow \alpha + \delta\beta;$
    endwhile
    $num\_pts \leftarrow i;$
    $scan \leftarrow \{\mathbf{r}_i | 0 \leq i < num\_pts\};$
    return $scan$

procedure predict_return(x, y, α, M)
    $result \leftarrow$ (MAX_ECHO, α);
    for $t = 1, \ \ldots, \ n_T$
        if (target_visible($x, y, \mathbf{p}_t, M$) = TRUE)
        then
            $\hat{\phi}_t \leftarrow$ bearing_to_target($x, y, \mathbf{p}_t$);
            if (($\alpha \geq \hat{\phi}_t - \beta_t/2$) **and** ($\alpha \leq \hat{\phi}_t + \beta_t/2$))
            then
                $temp \leftarrow$ range_to_target($x, y, \mathbf{p}_t$);
                if ($temp < result$) $result \leftarrow temp$
            endif
        endif
    endfor
    return $result$

procedure target_visible($x, y, \mathbf{p}, M$)
    define ray from sensor location (x, y) to target $\mathbf{p}$
    for $t = 1, \ldots, n_T$
        if (ray intersects target $\mathbf{p}$) **return**(FALSE)
    endfor
    return TRUE

Figure 2.7: Algorithm for generating a simulated sonar scan.
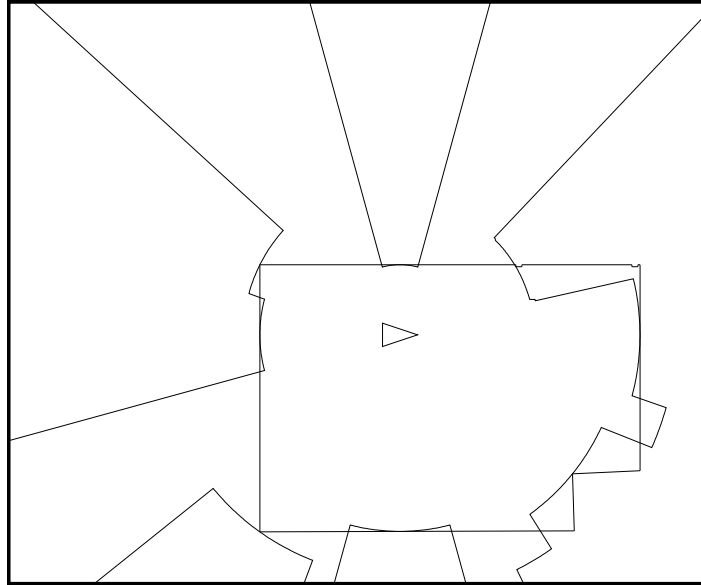
Figure 2.8: Simulated sonar scan (scan sim_b0), displayed by connecting each pair of adjacent returns with a line segment.
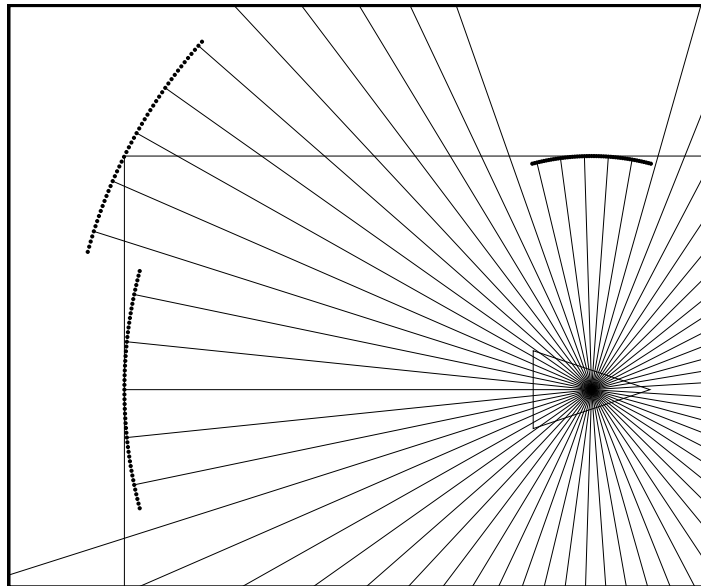
Figure 2.9: A closer view of the same simulated scan, displayed in a slightly different form. TOF dots are shown for every return, and rays are drawn from the sensor location to the TOF dot of every tenth return.
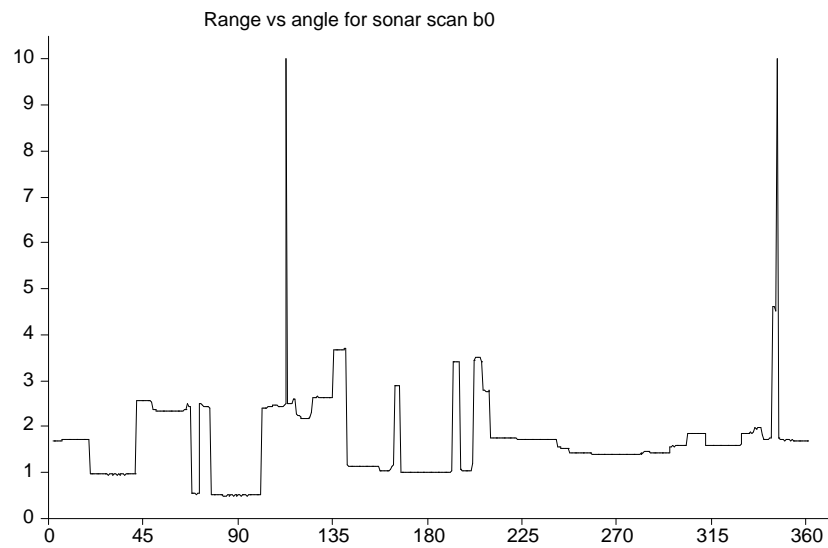
Figure 2.10: A plot of range *vs* transducer orientation for the scan of Figure 2.1. 612 measurements equally spaced in angle were taken. The x axis shows the transducer orientation in degrees. The y axis shows the range in meters. This plot shows that a large proportion of the sonar scan consists of angular regions in which adjacent measurements have nearly the same range, and hence form horizontal line segments (circular arcs in Cartesian coordinates.) We refer to these features as *regions of constant depth* (RCDs).
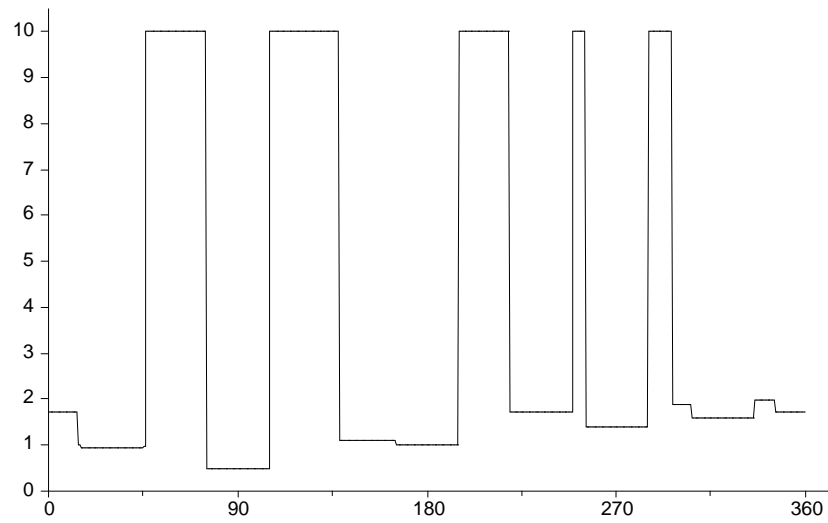
Figure 2.11: A plot of range *vs* transducer orientation for the simulated scan of Figure 2.3. 612 measurements equally spaced in angle were taken. The x axis shows the transducer orientation in degrees. The y axis shows the range in meters.
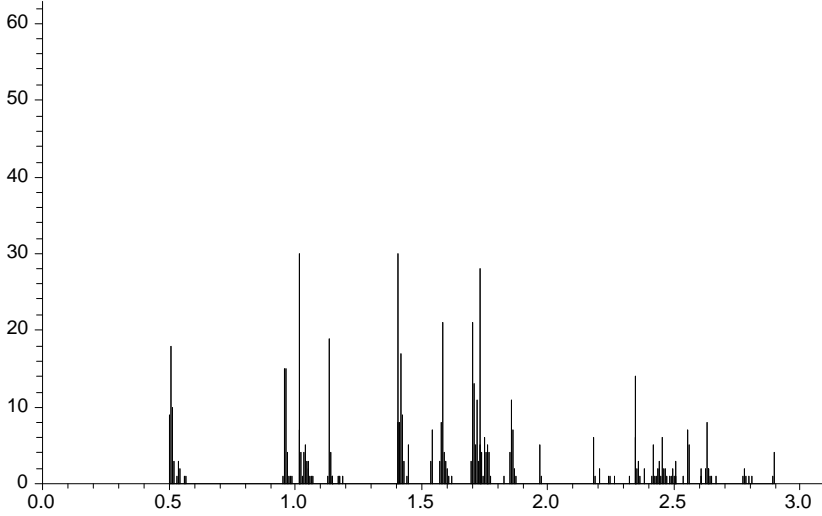
Figure 2.12: Histogram of range values less than 3.0 meters for sonar scan b0, quantized in 5mm intervals.
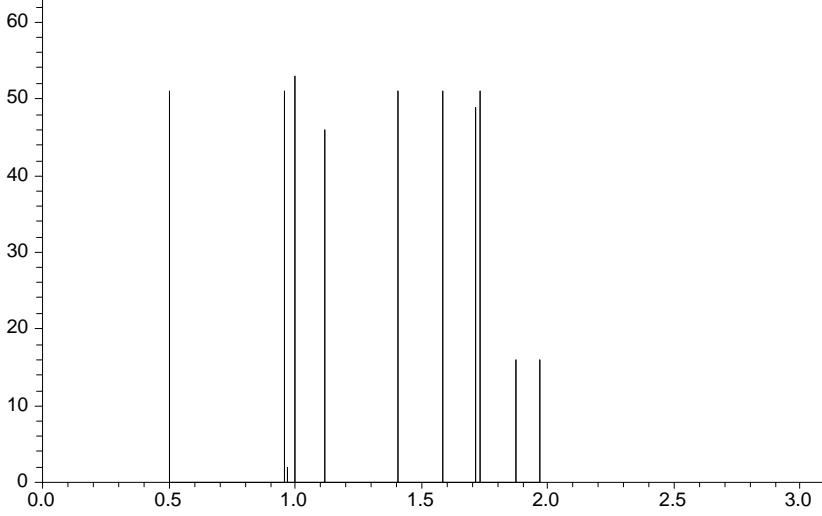
Figure 2.13: Histogram of range values less than 3.0 meters for simulated sonar scan sim_b0, quantized in 5mm intervals.

followed by many previous researchers, for example, Drumheller [40]. However, Kuc's model predicts these sonar returns should form a circular arc, as in Figure 2.9. In Figure 2.15, we remove the line segment corresponding to the room model, and add a circular arc drawn at the minimum range in the response. Figure 2.16 shows a range *vs* orientation plot for this wall response. Roughly, returns -19 to 22 form a horizontal line segment in this figure, as all are within 1 centimeter of 1.015 meters, the most frequently occurring range. This is the RCD predicted by theory. A question we want to answer is "what about returns -26 to -35, and 30 to 39?" Are these the result of diffuse reflections from some part of the wall that is not perpendicular to the beam, or can we account for them in the context of a specular target model?

2.6.2   A Closer Look at the Polaroid Ultrasonic Ranging System

To offer an explanation for this, we need to take a closer look at the standard Polaroid ultrasonic ranging system [109]. Figure 2.18 shows a simplified block diagram of the electronics. The standard Polaroid ranging system transmits 56 cycles of a 49.4 kHz square wave, yielding a total pulse duration of 1.13 mSec. When this is the transmitted waveform, we say that the system is operating in *long-range* mode. Our Robuter mobile robot comes equipped with an additional *short-range* sensing mode, in which only 8 cycles of the 49.4 kHz waveform are transmitted. However, all the real data we have shown up to this point has been obtained by our own scanning device, discussed in Appendix A, which makes use of the standard long-range mode of operation.

After the transmit waveform is applied to the transducer, a blanking period follows during which no echoes can be detected, and then the transducer acts as a receiver. Detected echoes are fed into a circuit that contains a time variable gain (TVG) amplifier to compensate for the effects of spreading loss and the attenuation of sound in air. The receiver amplifier output is then fed into a thresholding circuit, and the time-of-flight measurement is obtained by taking the difference in time from when the beginning of the pulse was transmitted to the first time at which the threshold is exceeded. An appropriate calibration factor is then applied to convert this time value into distance.

We wish to address three particular sources of error in the Polaroid ranging system which can explain the formation of the typical wall response shown above:

1. Transmitted pulse duration
   All timing is based on the assumption that the start of the transmitted pulse is the part of the returned echo that actually exceeds the

Figure 2.14: Typical wall response 1. TOF dots are shown for every return, and rays are drawn from the sensor location to the TOF dot of every tenth return.

Figure 2.15: Typical wall response 1. An arc is drawn at the range of the closest return in the wall response.
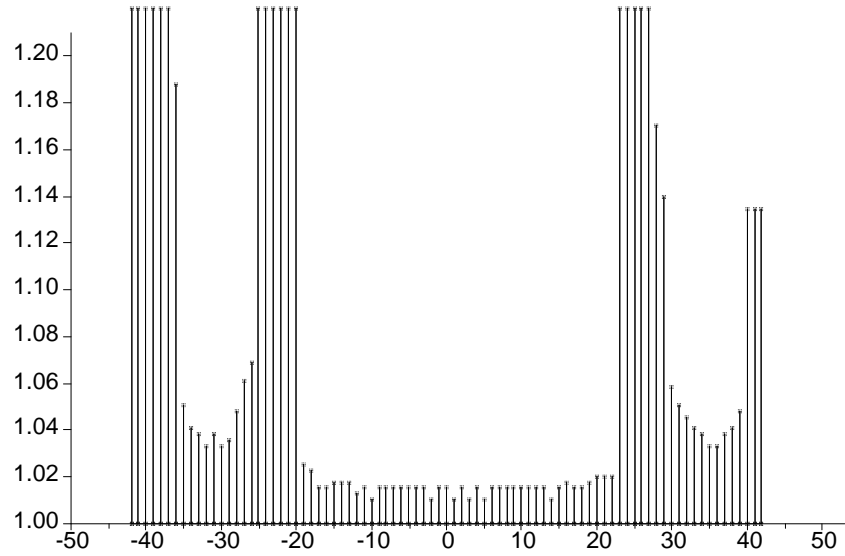
Figure 2.16: Typical wall response 1: range *vs* orientation plot for left wall in scan b0.
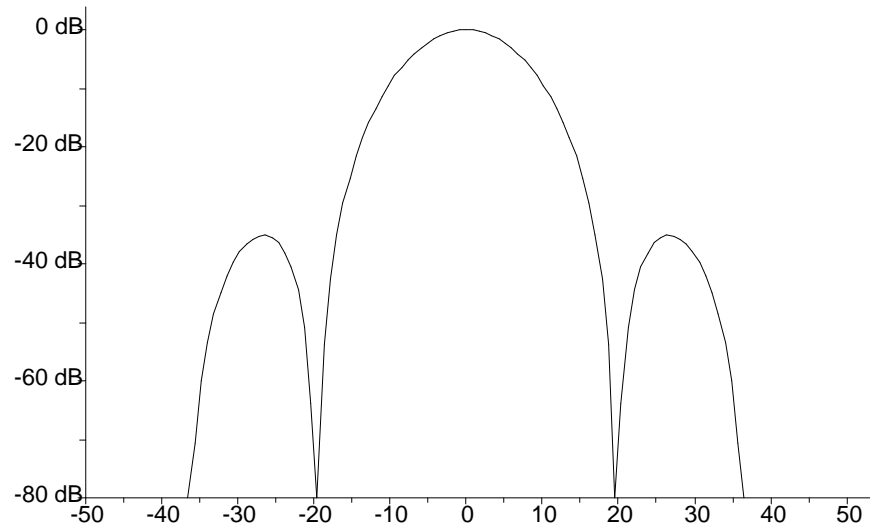
Figure 2.17: Main-lobe and first side-lobes of the plane circular piston radiation pattern.
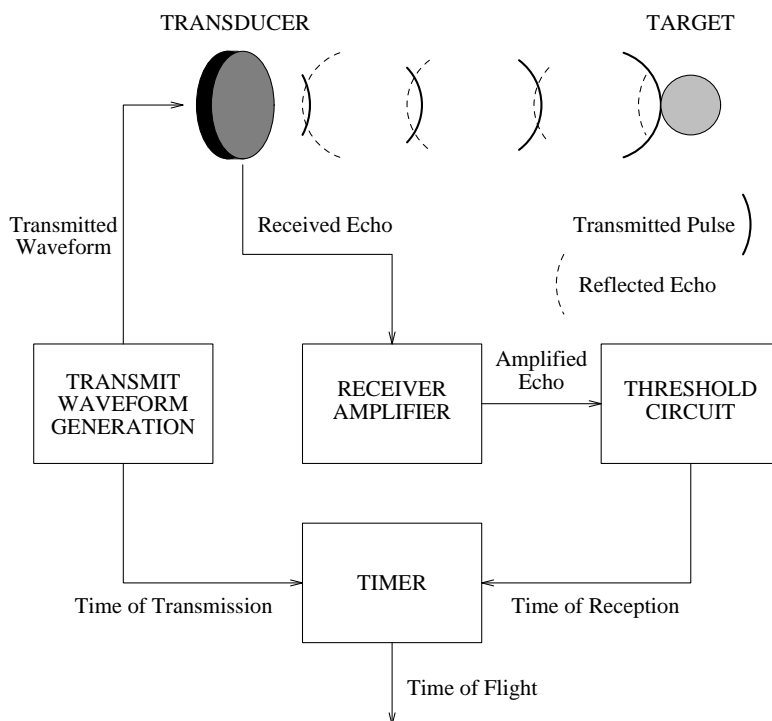
Figure 2.18: The Polaroid ultrasonic ranging system.

detector threshold. If some other part of the transmitted pulse trips the threshold, the error could be as much as 19 centimeters.

2. Time variable gain (TVG) Amplifier
   The Polaroid TVG amplifier only makes a 16-step approximation to the ideal exponential curve that would exactly cancel beam spread and attenuation losses. Further, the TVG response would need to accommodate the complex fashion in which the attenuation of ultrasound in air changes with temperature and humidity. The result of this limitation is to make the visibility angle β_t of target t a complex function of range, humidity, and temperature. The effect is the same as if the threshold were a sawtooth-varying function of range. This effect is clearly demonstrated in the published experimental work of Lang *et al.* [85].

3. Capacitive charge-up in the threshold circuit
   The mechanism by which thresholding is actually carried out involves

> a capacitive charge-up, whose impact can be a significant source of
> error when a long pulse duration is used. For strong reflected signals,
> just 3 cycles are necessary for charging up to threshold. Calibration
> usually accommodates this short charge-up time. For weaker signals,
> charge-up can take place over a considerably longer time, resulting in
> an erroneously elongated range value.

For a specular surface, the perpendicular portion of the wall is the
only part of the wall that reflects sound energy directly back to the trans-
ducer. Hence, whether detection occurs or not depends on the amount
of sound energy perpendicularly incident on the wall, which is a function
of the transducer radiation pattern. A model often used to characterize
transducers of the Polaroid type is that of a plane circular piston in an
infinite baffle [91]. Using values of transducer radius and transmission fre-
quency appropriate to the Polaroid sensor, the complete radiation pattern
is plotted in Appendix A. Here we just show the main lobe and the first
side-lobes, plotted in Figure 2.17, with the x axis labeled not in degrees but
by return index to correspond with Figures 2.14 to 2.16. While the piston
model is a simplification, it does reflect the nature of the actual Polaroid
beam pattern.

The three error sources listed above, coupled with this knowledge of the
sonar radiation pattern, provide a simple explanation for the wall response
of Figures 2.14 to 2.16. At nearly orthogonal sensor orientations (returns
-18 to 18), the perpendicularly incident energy comes from the main lobe,
and hence the range value measured is very accurate. At the edges of the
main lobe, range accuracy degrades slightly, until the wall disappears for
returns -25 to -20 and 23 to 29, which correspond to nulls in the sonar beam
pattern. At these sensor orientations, the incident energy is sufficiently
weak that detection does not occur. As the sensor is rotated further away
from the wall, the perpendicularly incident energy increases as the side-
lobes come into view. However this energy is not strong enough to trip the
detector threshold promptly. Detection comes through a charge-up action
whose duration is proportional to the radiation energy that is currently
perpendicularly incident. In Figure 2.16, the most severely delayed return
that is still detected is return 26, which has a range of 1.068 meters, which
is 5.3 centimeters longer than the mode of the response.

To answer the question we posed earlier, returns -26 to -35 and 30 to
39 of Figures 2.14 to 2.16 are in fact due to specular reflections. Because
of the errors sources listed above, when the perpendicularly incident sound
is low in intensity, range accuracy suffers. These effects reveal themselves
in wall responses over and over again in environments with ordinary walls.
As evidence of this, Figure 2.19 shows three other typical wall responses.
Figure 2.19 (a) and (b) show the same wall as in Figures 2.14 to 2.16,
viewed from the same distance but from a position 0.5 meters closer to

the bottom wall. Figures 2.19 show this same wall viewed from a position 0.71 meters further away. Figure 2.19 (c) and (d) show the top wall in the original sonar scan. Figure 2.20 through 2.22 show range *vs* orientation plots for these wall responses, with each return cross-referenced to Figure 2.19. Understanding these pictures provides an intuitive understanding of the way in which the Polaroid ranging system actually operates.

2.6.3 Strong *vs* Weak Returns

We call those errors which occur at high angles of incidence *time-delay range errors*. Note that this effect is non-symmetric, and hence not Gaussian. The range measurement can be up to 19 centimeters too far, while usually not more than 1 centimeter too close (depending on the manner in which the system is calibrated). Lang *et al.* present remarkable error distribution plots which clearly detail this effect [85]. To put our knowledge of these errors to use for interpreting sonar data, we divide sonar returns into two classes:

1. **Strong returns** possess sufficient energy to exceed the threshold circuit promptly, and hence are very accurate in range (within 1 centimeter).

2. **Weak returns** are characterized by time-delay range errors. These only surpass the detector threshold by the effect of a long charge-up in the threshold circuit and changing gain steps in the TVG amplifier.

It is not possible to classify a single isolated return as strong or weak based solely on time-of-flight information. One solution for this problem is to use a short-range sensing mode only, since the short duration of the transmitted pulse makes the maximum time-delay range error just 1.5 centimeter.[1] Since our Robuter mobile robot sonar system has a short-range mode, this is the approach we adopt for the localization experiments presented in Chapter 3. For high range accuracy we recommend exclusive use of the short-range sensing mode. Despite this recommendation, the densely sampled scans shown in this chapter and used in Chapter 4 for map building use the standard, long-range Polaroid system, and thus possess many weak returns. Most previous researchers have also used the standard, long-range mode. For long-range mode sonar, our approach to interpretation is to identify strong returns using the concept of *local support* to extract RCDs.

---

[1]This is calculated assuming a transmitted waveform of 8 pulses, and a charge-up time of 3 pulses for strong echoes in the threshold circuit, yielding a worst-case where the threshold is exceeded 5 pulses too late, after the 8th pulse.
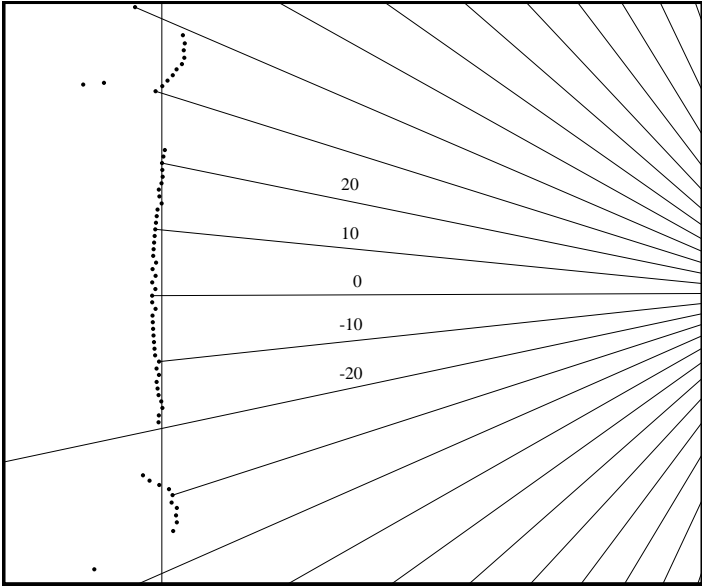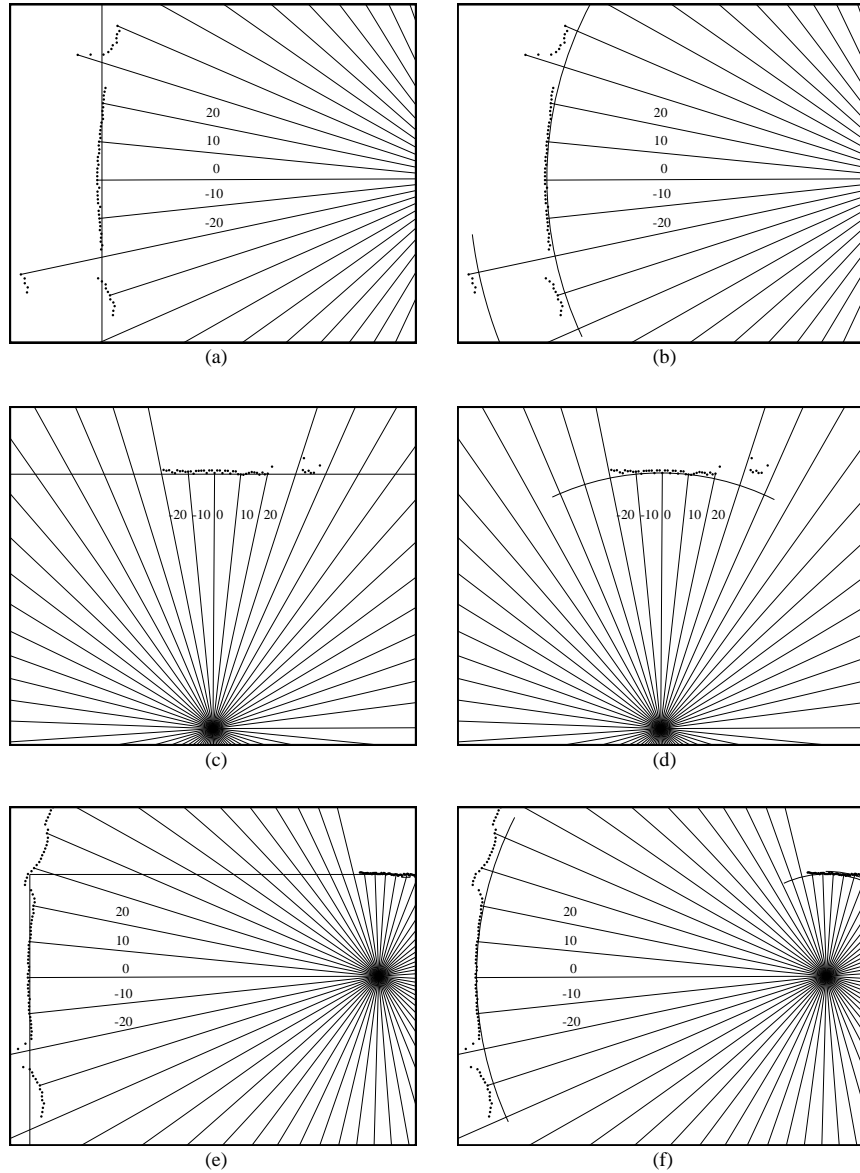
Figure 2.19: Several other typical wall responses. (a) scan and room model for wall response 2. TOF dots are shown for each return, and rays are drawn from the sensor location to the TOF dot of every tenth return. (b) An arc is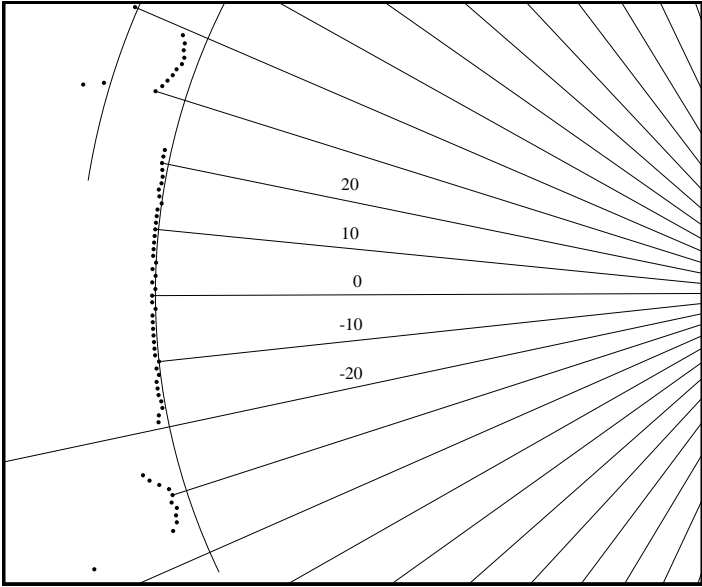 drawn at the range of the closest return of the wall response, illustrating the effect of time-delay range errors at high angles of incidence. Likewise, (c) and (d) show wall response 3, and (e) and (f) show wall response 4.
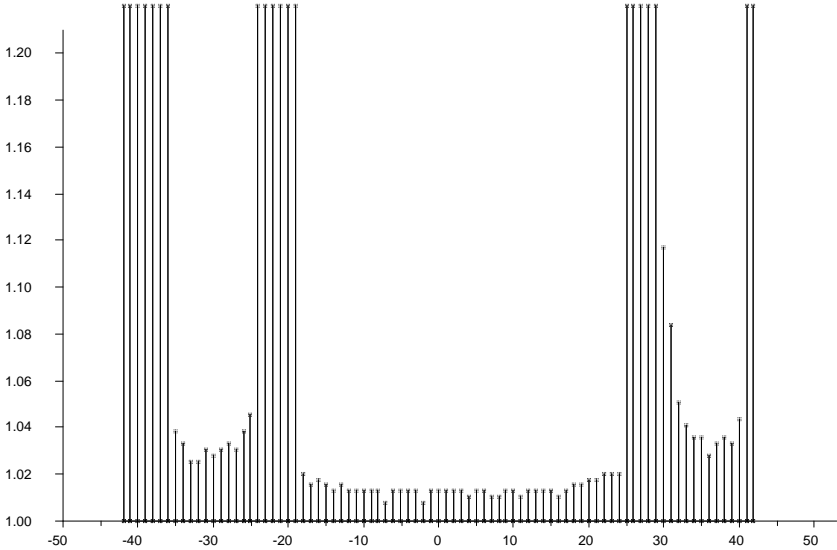
Figure 2.20: Typical wall response 2: range *vs* orientation plot for left wall in scan e0.
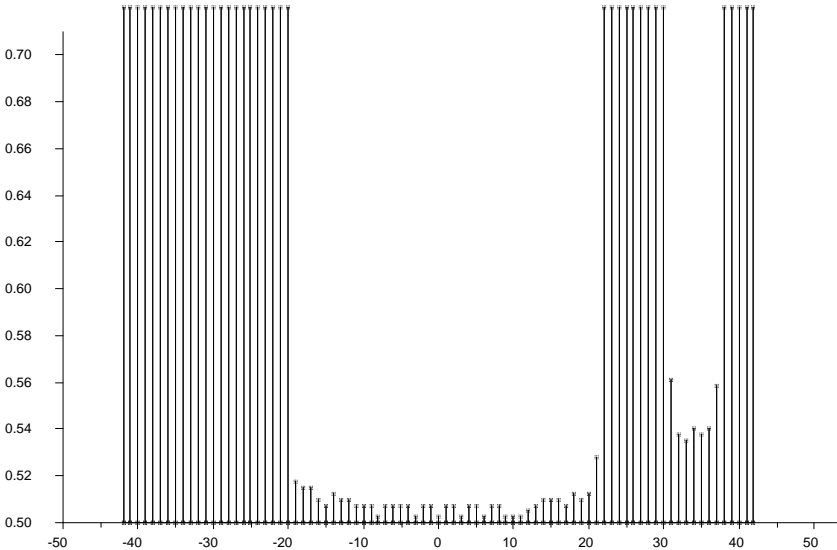
Figure 2.21: Typical wall response 3: range *vs* orientation plot for top wall in scan b0.
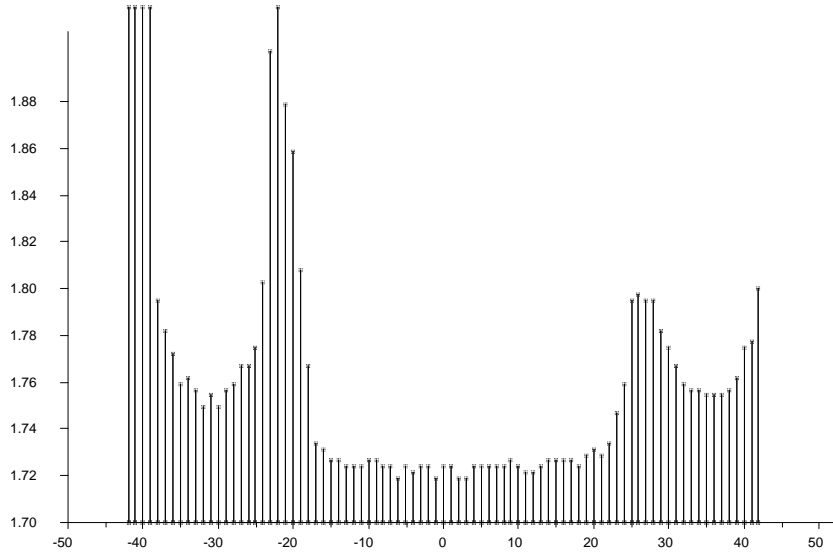
Figure 2.22: Typical wall response 4: range *vs* orientation plot for left wall in scan b3.

2.7   Regions of Constant Depth

The above discussion advanced a qualitative argument to the effect that a specular sonar model predicts that sonar scans will consist of RCDs, which are formed by strong returns. Weak sonar returns which do not form RCDs can be explained by undesirable characteristics of the standard sonar hardware. Based on the assumption that the information content in a sonar scan is conveyed by the RCDs, we now formalize the notion of an RCD, and describe a straightforward process for extracting RCDs from sonar scans.

2.7.1   Extracting RCDs

The *range difference* of a connected set of returns is defined to be the absolute value of the difference between the minimum and the maximum value of the set. The range difference of two returns is simply the absolute value of their difference. We define δ_R to be the range difference threshold. A *Region of Constant Depth* is a connected set of returns with range difference less that δ_R. A return whose range difference with each of its adjacent returns is greater than δ_R defines a single element RCD. The range of an RCD is taken to be the mode of the distribution of range values across the RCD. The *width* β of an RCD is the difference in angle between the
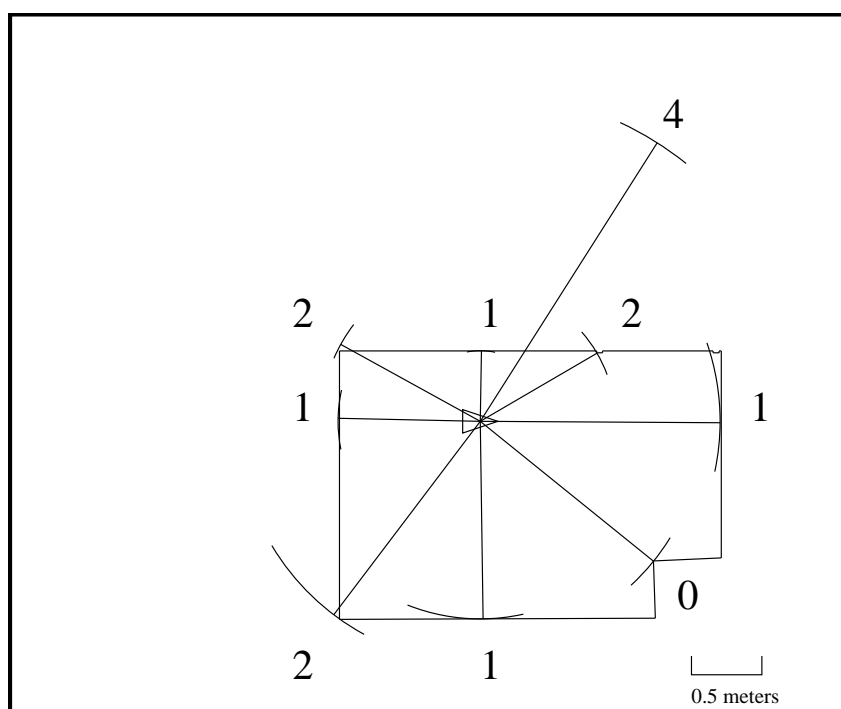
Figure 2.23: Regions of constant depth (RCDs) of width $\beta \geq 10$ degrees extracted from scan b0. The order of each RCD has been labeled. We can see 4 1st-order RCDs from the four walls of the room, 3 2nd-order RCDs from corners, and a single 4th-order RCD resulting from a multiple reflections off the top wall into the lower right-hand corner of the room. There is a single 0th-order RCD resulting from a weaker reflection from the edge in the lower right-hand region of the room.

left-most return and the right-most return of the RCD. Figure 2.23 shows
RCDs of width $\beta \geq 10$ degrees from the sonar scan of Figure 2.1 using a
range difference threshold of 1 centimeter.

2.7.2 Constraint Angles of an RCD

A single isolated return is characterized by a uniformly distributed uncer-
tainty in the true bearing to a target of up to 30 degrees, depending on
target strength. RCD extraction provides a means of reducing this angular
uncertainty, as multiple adjacent returns to the same target constrain the
possible true bearing to the target. For each RCD, we define a series of
angles to constrain the true bearing to a target. We define θ_1 to be the
angle of the right-most return of the RCD, and similarly, θ_2 to be the angle
of the left-most return of the RCD. θ_m is the *orientation* of the RCD, which
is simply the mean of θ_1 and θ_2:

$$\theta_m = \frac{\theta_1 + \theta_2}{2}. \tag{2.9}$$

As discussed above, an unfortunate side-effect of the limitations of the
Polaroid ranging system is the difficulty in specifying the visibility angle
β_t of target t. The approach we take is to choose a value called β_{max},
which will be greater than the maximum value of β_t for all targets in the
environment. We have used $\beta_{max} = 30°$ in our experiments, as discussed in
subsequent chapters. Based on this value, we define two *constraint angles*
for each RCD:

$$\theta_u = \theta_1 + \frac{\beta_{max}}{2} \tag{2.10}$$

$$\theta_l = \theta_2 - \frac{\beta_{max}}{2} \tag{2.11}$$

where θ_u is termed the upper constraint angle of the RCD, and θ_l is the
lower constraint angle of the RCD.

   Figure 2.24 illustrates these different angles for the case of a wide RCD,
showing the way in which multiple returns of the same range reduce the
range of the possible bearing to the target. Figure 2.25 shows the values of
these angles for the RCD defined by an isolated single return. In this case,
uncertainty in the true bearing to the target takes the full value of β_{max}.
These angles are used in Chapter 3 in the process of matching a predicted
RCD to an observed RCD, and in Chapter 4 for matching multiple RCDs
observed from different locations based on different target assumptions.

2.7.3 Local Support

The notion of echo strength, and its impact on range accuracy, was quali-
tatively described above in terms of the Polaroid hardware. Strong echoes
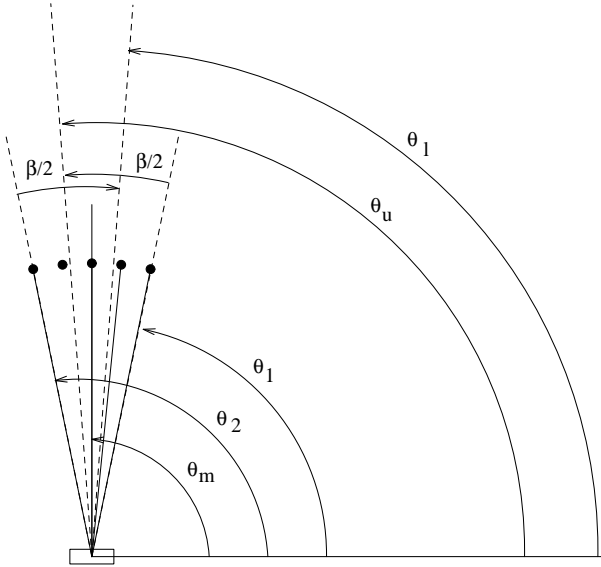
Figure 2.24: Constraint angles for a five element RCD.
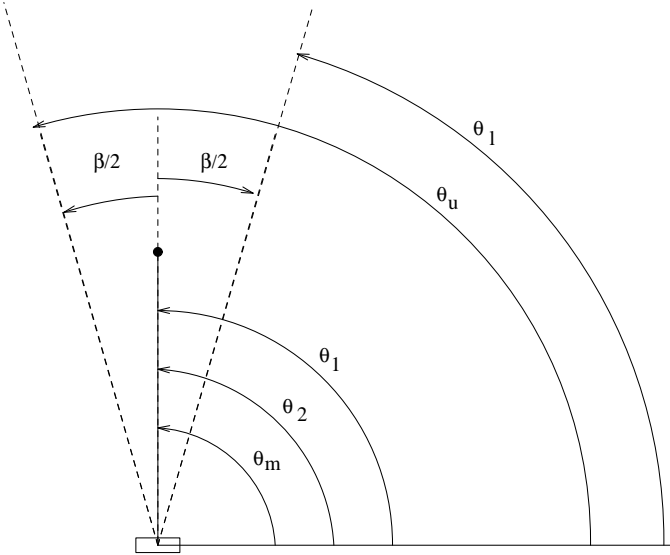
Figure 2.25: Constraint angles for a single element RCD.

promptly trip the threshold of the receiving circuit, giving an accurate distance measurement. Weak echoes do not possess sufficient energy to charge the detection circuit quickly up to the threshold level; the threshold is only exceeded by the random combination of a slow charging-up period and jumps in the non-linear time variable gain amplifier. The significant time delay that has occurred causes an erroneously elongated range reading. As supported by our experimental results and the experimental evidence of others [85], repeated firing at an orientation that results in a weak echo gives a considerably wide spread of actual time-of-flight readings, in contrast to the repeatable, tight distribution of strong echoes.

When the Polaroid hardware is operated in long-range mode, weak returns are a significant source of range error. The process of RCD extraction provides a mechanism for determining the strength of an echo which yielded a return, when densely sampled data is available. The uncertain process by which weak returns are produced means that if the sensor is slightly rotated in either direction, the range can change considerably. In contrast, the well-defined nature in which strong echoes exceed the receiver threshold means that the range will not change significantly for small changes in sensor orientation. To formalize this concept, the *local support index* of a return is defined to be the number of returns in the RCD to which the return belongs. The *local support angle* of a return is the width of the RCD to which it belongs (zero for a return with local support index of one). A *weak return* is defined to be a return with local support angle less than a pre-defined threshold β_{min}. Typical values we have used for β_{min} are between 5 and 10 degrees. A *strong return* is a return with local support angle greater than or equal to β_{min}.

2.7.4   The Order of an RCD

Figure 2.2 shows that in a typical indoor scene, many "false" range readings are produced by the system when the beam is oriented at high angles of incidence to planar targets. At these high angles of incidence, the sound energy emitted from the side-lobes of the beam that strikes the wall perpendicularly is not of sufficient strength to exceed the threshold of the receiving circuit. As a result, the first echo detected by the system is a *multiple reflection* by some other part of the beam reflecting specularly off the wall to some other target and then back. These multiple reflections have been observed by many other researchers, and have commonly been referred to in the literature as "specularities" [38]. We feel this term is misleading, because most accurate, strong returns produced by planes and corners are in fact due to specular reflections.

To alleviate this confusion, we define the *order* of a return as the number of surfaces the sound has reflected from before returning to the transducer. Orienting the transducer perpendicular to a planar surface such as a wall

produces a 1st-order return. Corners produce 2nd-order returns, because the sound has reflected specularly off two surfaces before returning back to the transducer. Multiple reflections will produce 3rd- and higher-order returns. A crucial task in interpretation is to eliminate these higher-order reflections which, if taken to be the distance to the nearest object, yield false range readings. To incorporate diffuse edges in this terminology, weak, diffuse reflections resulting from edges will be defined as 0th-order returns. All returns in an RCD are assumed to have the same order, and so the order of an RCD is the order of each of its returns.

2.8   Sparse *vs* Densely Sampled Data

In the literature, rings of fixed sonars around the vehicle perimeter have been the most popular sonar sensing configuration. Circular rings of 12 or 24 transducers seem to be the most common, but as many as 30 transducers have been used [131]. A circular ring of 24 sonars has an angular spacing between transducers of 15 degrees. Unfortunately 30 degrees is about the upper limit on target visibility angles. (Actually side-lobes are visible outside this angular window at some ranges, but these are usually weak returns.) Thus there is a good chance that with a ring of 24 sonars, many targets will be observed by only one sensor. With long time duration pulses, there is little means of knowing whether an individual return is strong or weak. For this reason, the bottom-up data interpretation capabilities of typical sonar rings seem limited when long-range sensing modes are used.

The data produced by a ring at a given position can change drastically, depending on the vehicle orientation, as shown in Figures 2.26 and 2.27. To imitate a 12 transducer ring with real data we have shown before, Figure 2.26 shows the result of sampling every 51st return from the scan of Figure 2.1, with two different offset values to illustrate a change in vehicle orientation of about 13 degrees. The offset value for Figure 2.26 (b), (d) and (f) has the particularly drastic effect of making the top and left walls invisible, as each transducer happens to line up in the null between side-lobe and main-lobe. Figure 2.27 shows the considerable improvement of a 24 element ring, but some drastic changes are still observed. These pictures do not account for the effect that a non-zero vehicle radius would have on the data produced.

For localization in known environments, the sparse data of multiple static sonars is adequate for on-the-fly position estimation. However, when map building is also a requirement, the lack of local support makes data interpretation more difficult. Short duration transmitted pulses ease this situation somewhat, as weak returns are eliminated, but comparison of typical target visibility angles and ring spacings shows that it is difficult
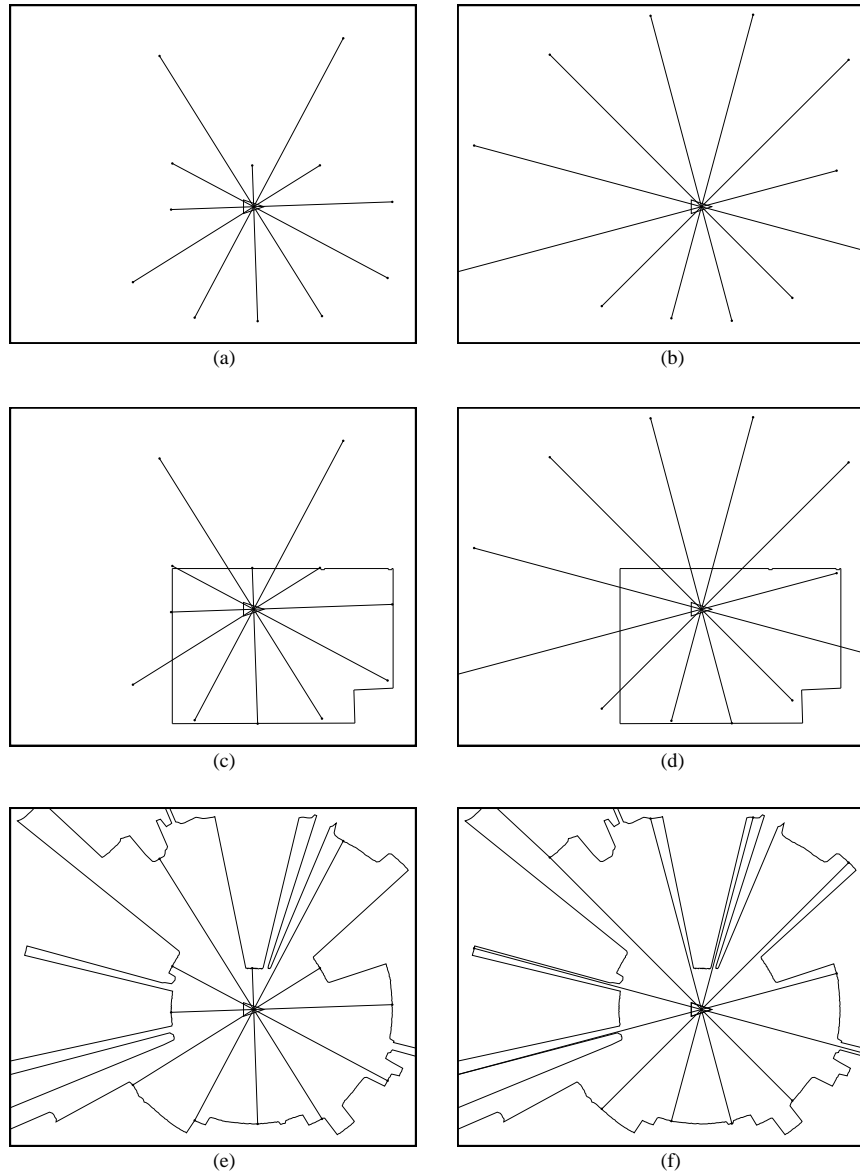
Figure 2.26: The effect of a change in vehicle orientation on the data produced by a 12 transducer ring. Every 51st return of the original 612 point scan is displayed. (a) and (b) show the 12 returns for offset values of 0 and 22 returns, respectively. The latter value was specifically chosen to give disastrous effect. (c) and (d) show the superposition of the room model, while (e) and (f) show the superposition of the entire 612 point scan.
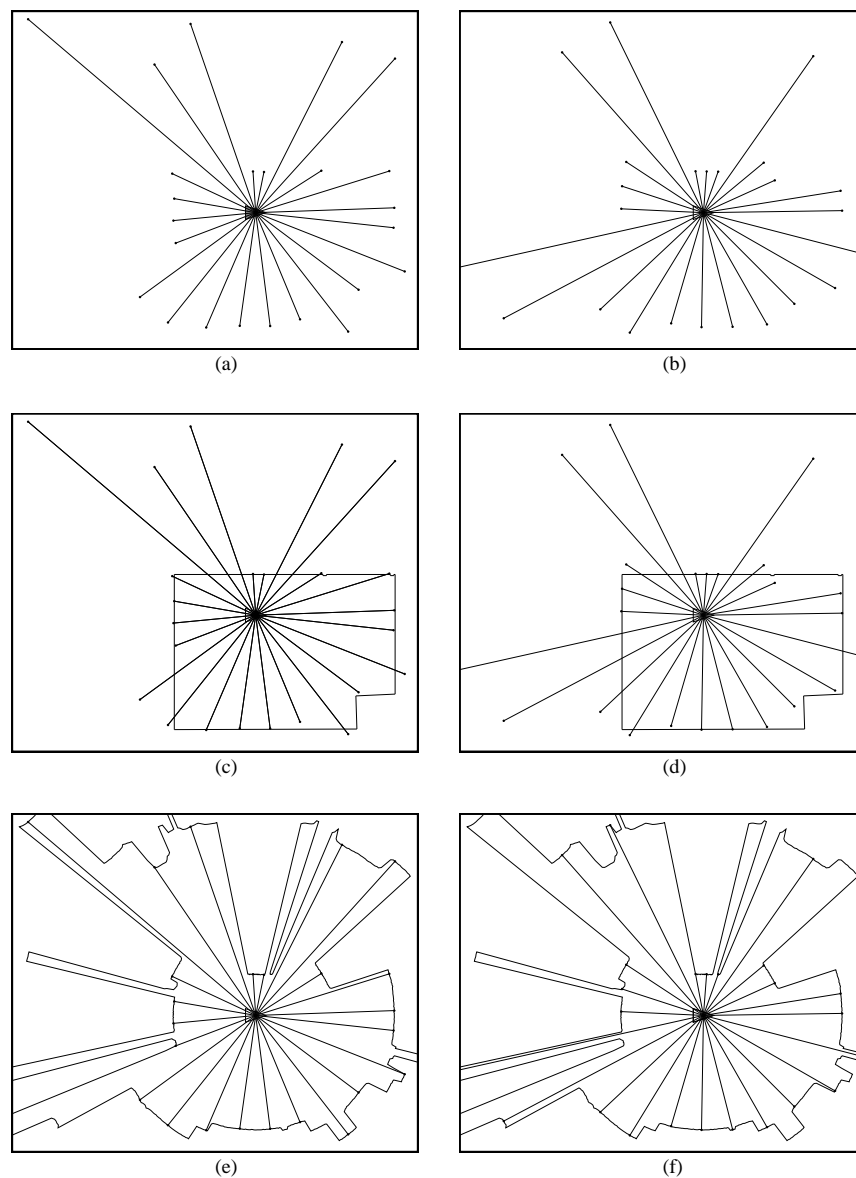
Figure 2.27: The effect of a change in vehicle orientation on the data produced by a 24 transducer ring. Every 25th return of the original 612 point scan is displayed. (a) and (b) show the 24 returns for offset values of 0 and 12 returns, respectively. (c) and (d) show the superposition of the room model, while (e) and (f) show the superposition of the entire 612 point scan.

to get two or more returns that correspond to the same target. The subject of choosing scanning densities to assure target visibility has been given extensive treatment by Kuc and his associates in the pursuit of 100% reliable collision prevention [82], [81]. With regard to our localization and map building requirements, a ring does allow many returns to be acquired quickly, but these returns will not have local support, and targets of interest will not necessarily be observed.

2.9   Discussion

Ironically, weak returns make sonar scans "look better", if the expectation one possesses of what a scan should look like is that of a ray-trace scanner. This effect is clearly visible in Figure 2.19 (c), in which returns 32 through 36 receive just enough time-delay to place them at the true range to the wall along the beam axis. The co-linearity of the returns in this wall response is simply coincidental, for they are produced exclusively by reflections off the perpendicularly incident part of the wall.

It seems that the presence of these weak returns will improve the performance of algorithms that fit line segments to sonar returns [37], [40], [77]. However we feel the physics of sonar clearly dictates that fitting line segments to the returns of a single scan is not advisable. We do not dispute the results of previous researchers; rather, our point is that superior performance should be possible with an RCD-based scheme. Our experience has been that time-delay side-lobe effects are much more common for planes than other target types, and thus the target response of a plane will "look" much more like a line than will the response of a corner, but we see it as unavoidable that line segment-based routines will inadvertently fit line segments to corner responses.

We feel that the occurrence of weak returns has disguised the reality of the acoustic ranging process, and has contributed to some misconceptions held by previous researchers. For example, many report that sonar gives the distance to a planar target along the beam axis until some "critical angle" [40] where a specular reflection occurs. Many failed to realize that the physics of sonar dictates that *all* reflections produced by smooth planar targets are in fact specular reflections. (Thus our insistence that multiple reflections should not be called "specularities" [38], [122].) Although hardware details are not given, it seems clear to us that Kuc and Brown use transmitted pulses of short duration, and thus weak returns are not produced by their hardware. We feel this reconciles the difference in appearance of Kuc's scans with those of most other researchers, and is perhaps why some have been slow to acknowledge the consequences of his work.

2.9.1 Previous Work Revisited

The method of many previous researchers suggests that they regard the Polaroid sensor as a poor approximation to the ray-trace scanner that they really want. Sonar cannot provide this type of information, but it can provide accurate, easy-to-use range measurements. RCDs are what the Polaroid transducer is trying to tell us: precise distance measurements to strong acoustic targets. The rest of the scan results from weak reflections that are inaccurate in range because of limitations in the standard Polaroid electronics.

We stated above that a consequence of acoustics is that a single sonar scan should "look bad", that is, it will look very different to a map of the room in which it was taken. Sonar scans with the unmodified Polaroid ranging system will in fact bear a resemblance to a room map if many corner targets lie along the walls of the room. For example, "Exposed studs that provide excellent beam return properties" lining the walls of the room explain the "exceptional quality" of the scan shown by Everett [49]. In our approach, these studs each define their own corner targets. To remove false multiple reflections, some researchers have artificially added acoustic targets to the environment. For example, Steer has developed acoustic retro-reflectors, that a user could place along the smooth surfaces of a room to provide a large number of corner targets and thus make false multiple reflections unlikely [122]. One drawback of such a technique is that modifying the environment may be undesirable. More importantly, we believe that since the fundamental enemy is the correspondence problem, adding a multitude of acoustic targets makes achieving correspondence between observed data and environment targets more difficult. Indeed, we believe the power of acoustic sensing is that typical environments present many fewer acoustic targets than optical targets, and hence, overcoming the correspondence problem should be considerably easier with an acoustic sensing modality. From the standpoint of localization, the specularity of the environment is a feature, not a bug.

We feel that when using sonar, maps should consist of target locations for all target types: corners, edges, planes, and cylinders. We acknowledge that this increases the map complexity in comparison to other approaches, like that of Drumheller [40], whose maps consist of line segments that crudely approximate the room geometry. For this reason, we believe feasible navigation is only possible when the map is autonomously built and maintained.

Drumheller took his scans at the unusual height of five-and-a-half feet, and his rooms appear to contain a large number of bookcases at this height. The corners defined by the books in a bookcase present a large number of corner targets, and thus false multiple reflections appear to occur less frequently in his data than in our data. Our approach would treat each of these

corners as an individual target, an obviously complex procedure, but in our defense we argue that at the lower heights at which we would mount our sensors, less features like bookcases and curtains would be visible. Lower to the floor, empty wall spaces and chair and table legs would be frequent targets in many application environments; these are spatially isolated.

2.9.2   Limitations

However the issue of environment complexity must be addressed as a potential limitation to our approach. We advocate that the complexity of the environment must be directly confronted. This will be easier in simple environments such as a corridors with smooth walls. This is in almost direct contrast to previous approaches which succeeded in complex environments characterized by a high target density, but which would have failed in an empty corridor. We acknowledge the danger to our approach posed by environment complexity, but counter that the idea behind tracking geometric beacons, discussed in Chapter 3, is to capitalize on the sensor data which is easiest to associate with the correct target. Part of the question of "is this a good beacon?" is "how far away is the closest target?" The more isolated a target is, the easier it will be to associate observations with it correctly. Thus, a measure of local target density seems a good criterion for choosing beacons, but we have not automated this decision at this point.

We are aware that in this chapter we have largely ignored the wave nature of sound. Instead we consider the sonar beam to consist of rays of emitted energy, and trace these rays as they bounce around a "world of mirrors". This is obviously an over-simplification of the complex physical process at work [103]. However, we feel that from the standpoint of predicting and explaining real sonar data in air, the model put forth here is sufficient to achieve effective mobile robot navigation, as we aim to demonstrate in the following chapters.

Rough surfaces characterized by irregularities on the order of the sonar wavelength can indeed reflect sonar energy directly back to the transducer at high angles of incidence for which our model would predict a multiple reflection. This is a limitation of the model, but we note that at nearly perpendicular angles of incidence, the range value produced will still be the orthogonal distance to the wall, and hence can be accurately predicted. While we cannot predict all returns in environments of rough surfaces, we can predict a significant subset of possible returns, and use these to provide localization. Our implementations with the Oxford Robuter, to be presented in Chapter 3, succeed in an environment of painted brick surfaces in just this manner.

In general, the consideration of sonar's viability in comparison with other sensors must address the following issues.

1. Slow data acquisition speeds
   Because of the speed of sound, sonar is drastically slower than optical alternatives such as laser and active infrared rangefingers [67], [99].

2. Moving vehicle effects
   When sensing from a moving vehicle at high speeds, the assumption that the transducer had the same position in space for both transmission and reception is no longer valid.[2]

3. Three dimensional effects
   Three-dimensional effects, such as the observation of the horizontal edge of a table, will be unavoidable in many application environments. These effects increase with range.

These limitations will be discussed is subsequent chapters in the context of our navigation results and our plans for future research.

2.10 Summary

We present the following conclusions to summarize this chapter:

- The ideal sonar sensor would not approximate the ray-trace scanner, but would provide precise range measurements to targets in the environment over a wide range of viewing angles. In the man-made, indoor environments under consideration in this monograph, these targets are planes, cylinders, corners, and edges.

- Local support makes sonar data interpretation easier, by providing a means of distinguishing accurate strong returns from inaccurate weak returns, and of constraining the uncertainty in the orientation to a target.

- The identification of multiple reflections is a crucial task in the interpretation of sonar data.

- Because sonar can be predictable and accurate, we believe sonar does not deserve its bad reputation!

---

[2]This situation is analogous to bi-static radar [113], in which a separate transmitter and receiver are used instead of a single, dual-purpose antenna. We feel techniques from this field will be of use in analyzing this situation.

Chapter 3

Model-based Localization

Localization is the process of determining the position of the robot with respect to a global reference frame. Our objective here is to achieve comparable performance to commercially available artificial beacon systems [128] without modifying the environment, by using the naturally occurring structure of the environment.

3.1   Introduction

First, let us clarify our terminology. Drumheller has used the term *absolute localization* to signify "the enabling of a mobile robot to determine its position and orientation ... in a way that is independent of assumptions about previous movements" [40]. Using line segments as features, Drumheller developed a search procedure for determining the robot's position based on the interpretation tree of Grimson and Lozano-Pérez [60]. We feel the use of search to answer "where am I?" is not opportunistic enough for an ordinary mode of operation. Because our goal is the continual provision of accurate knowledge of position, *a priori* position estimates should be fairly good. Hence, our use of the term *localization* refers to an ordinary, continuous mode of operation that avoids search. We shall use the term *relocation* (a word we adopt from orienteering [96]) to describe search-based position determination. This is only used in our approach for initialization and recovery from "getting lost". These two classifications define a spectrum, from opportunistic use of *a priori* position estimates to freedom from reliance on such estimates. Either can be viewed as too extreme, the former susceptible to getting lost, the latter overly pessimistic because the vehicle's position will be constrained by the previous position estimate and distance traveled since that position. A desirable goal is an algorithm which starts from the *a priori* estimate, but can successively expand its search, moving

toward the consideration of all possible vehicle locations. We must leave this task, however, for our future research agenda.

3.1.1 Previous Work in Localization

The localization system implemented by Cox, using the robot Blanche, fits our definition of continuous localization [33]. Position updates are produced by a matching algorithm that uses an initial estimate of vehicle position from odometry to launch an iterative registration procedure. The system senses the environment using a phase-based infrared range scanner to achieve on-the-fly performance. Hinkel *et al.* present a novel localization technique which capitalizes on the high data acquisition speed of their laser range scanner [65] [67]. Position estimation is performed using histograms. The approach seems to rely on Manhattan geometry environments, such as corridors, but successful results are claimed for more complex domains. Both of these optically motivated approaches would seem to have limited application with sonar, because of sonar's slow data acquisition speed, and the dissimilarity of sonar data to optical rangefinder data.

In computer vision, Sugihara [124] and Krotkov [80] address visual position estimation as a search-based relocation procedure using vertical line segments as features. Implementation of this concept has not been extensively developed. Sarachik presents an alternative to more traditional computer vision approaches that avoids three-dimensional modeling and calibration by the use of a novel camera configuration that can extract ceiling-wall boundaries [112].

Using grid-based representations, localization has been implemented via the correlation of a local occupancy grid with a globally-referenced occupancy grid constructed earlier [45] [100]. Extensive quantitative performance results have not been published. We believe that more reliable performance should be possible with a feature-based approach, as it allows one to predict the sonar data that will be obtained from a given position.

3.1.2 Model-based Vision

The localization problem is related to model-based vision, an approach to visual object recognition that has been advocated by Lowe [90], Mundy [32], and others [59]. The aim of model-based vision is to recognize an object in the environment based on an *a priori* model, and to determine the position and orientation of this object relative to the observer. The recognition process is fundamentally a search through prior information; the goal of visual processing is to provide constraints to guide this search to find the correct orientation and position of the object as quickly as possible.

This top-down approach conflicts with one traditional approach to vision, that first builds an intermediate representation called the 2 1/2-D

sketch bottom-up from the data, using a variety of visual cues such as stereo, motion, and shape-from-X, before attempting correspondence with *a priori* models. Lowe has argued strongly that in many instances, the intermediate representation of the 2 1/2-D sketch does not sufficiently aid the task of visual recognition to warrant the time needed for its construction. Further, uncertainty introduced in the depth and shape computation, for example the result of stereo matching errors, can significantly hinder the recognition process. Correspondence between the 2-D image and the 3-D geometric model is often better achieved without recourse to an intermediate representation.

Lowe's SCERPO vision system has inspired our advocacy of a model-based approach to localization. We view localization as a task of achieving correspondence between observations and an *a priori* model. The construction of intermediate representations, such as an occupancy grid or composite local model, may be useful for obstacle avoidance, but does not directly aid the achievement of correspondence with a global model, and the "blurring effect" they produce can in fact hinder this process. Because of sonar's slow data acquisition rate, the considerable length of time needed to construct a local map and then match this local map with a global map will make continuous localization impossible. With a model-based approach, the directed observation of just a few environment features can swiftly provide position information. Subsequent tracking of these features obviates the need to re-solve the correspondence problem, thus making possible a high bandwidth stream of reliable and useful information.

3.1.3 Motion Estimation *vs* Position Estimation

The fields of study in computer vision known as *motion* and *structure from motion* are closely linked to mobile robot navigation. Two classes of algorithm are generally employed: optic flow-based and feature-based [50], [107]. In this discussion we only address the latter approach. Research in feature-based motion analysis aims to compute scene structure and observer motion by matching features observed in one, two, or three images from one observer location with features observed in images from the next observer location. Many approaches make use of the extended Kalman filter (EKF) for recursive feature estimation [5], [4], [93], [110], [78].

Ayache and Faugeras have pioneered the use of trinocular vision to estimate the camera motion between observed scenes and to build detailed global maps [5], [4]. Processing is based on a "points, lines, and planes" model of the environment. Matthies has implemented an EKF-based algorithm that computes observer motion and maintains and refines a local model of the coordinates of stereo points used for matching [93]. In both of these formulations, strong emphasis is placed on an accurate model of sensor errors. Kriegman achieves hallway navigation by stereo tracking of vertical

edges, again employing an EKF to merge stereo correspondence points projected onto a 2-D model [78]. Semantic hallway models that indicate the relative location of doors and walls have been constructed. Instantiation of the semantic model is used to execute high-level commands such as "Enter the second door on the left". Processing is done by TV transmitter and radio link to an off-board host, and with a typical complete cycle time of 8 to 12 seconds for each 1 meter step of the robot. In a similar approach, Wells has implemented a system in which vertical edge tracking is achieved in corridors using on-board computation [133].

These researchers have attacked the motion problem without an *a priori* model, a considerably more complex task than localization with prior information. Motion estimation, however, does not by itself fulfill our requirements for long-term autonomous localization. The distinction lies in the formulation of the problem. Multiple EKFs are set up to estimate target locations, but there is no Kalman filter for the vehicle. Matches between features observed from two positions are used to compute an accurate estimate of the transformation between the vehicle position at times k and $k + 1$. An estimate of position with respect to a global frame comes indirectly by adding this transformation to the vehicle's prior position. Because each successive motion estimate will contain some uncertainty, the error in the global position estimate must gradually increase, in similar fashion to the errors of odometry, but smaller in magnitude. Because of this gradual accumulation of error, it seems that maintaining globally consistent world models across long time spans is not possible solely with motion estimates. This is clearly evident in the work of Matthies [93], Wells [133], and Kriegman [78], where global geometric consistency is not achieved. The results of Ayache and Faugeras [5] appear globally consistent because of the high accuracy of the calculated motion estimates, but results for long time span runs have not, to our knowledge, been presented.

We feel that after modeling is sufficiently underway, a change in strategy needs to be adopted. At a certain stage, the focus of the estimation problem should be shifted from map construction to position, not motion, estimation. The distinction is one of map *using* instead of map *fusing*. The comments of Pollard *et al.* are insightful in this regard:

> In the context of control of a mobile robotic vehicle it is important to distinguish between the twin goals of obtaining an accurate model of the environment and determining the current position in it.... The visual through-put and temporal response required by each task is very different. [110]

Map using presents the possibility of more efficient operation. A robot needs to say "that's it, my map is good enough, now my priority is to use the map to operate as efficiently as possible." Verification and maintenance of the map are still required, but should entail less computation than map

construction.

3.2 Problem Statement

We use the word *target* to refer to any object feature in the environment that can be observed by one of the robot's sensors. Target models were presented in the previous chapter for planes, cylinders, corners, and edges. We use the term *geometric beacon* to designate a special class of target that is a stable environment feature useful for localization. The target state vector $\mathbf{p}_t$ contains the geometric parameterization of target t with respect to a global reference frame. At this point in our discussion, we assume all targets to be stationary, so that the target state $\mathbf{p}_t$ of target t is not a function of time. We stress that the map is just a set of target state estimates, not an exhaustively detailed world model. For the localization algorithm presented here, the map is provided *a priori* to the algorithm, and the target estimates that comprise the map are assumed precise. Chapter 4 addresses the construction of such a map.

With reference to Figure 3.1, we denote the position and orientation of the vehicle at time step k by the state vector $\mathbf{x}(k) = [x(k), y(k), \theta(k)]^T$ comprising a Cartesian location and a heading defined with respect to a global coordinate frame. At initialization, the robot starts at a known location, and the robot has an *a priori* map of n_T geometric beacons, whose locations are specified by set of the known vectors $\{\mathbf{p}_t \mid 1 \leq t \leq n_T\}$. Localization is a cyclic procedure that is repeated as frequently possible. At each time step, observations $\mathbf{z}_j(k+1)$ of these beacons are taken. The goal of the localization extended Kalman filter is to associate measurements $\mathbf{z}_j(k+1)$ with the correct beacon $\mathbf{p}_t$ to compute $\hat{\mathbf{x}}(k+1 \mid k+1)$, the updated estimate of vehicle position.

The Kalman filter relies on two models: a *plant* model and a *measurement model*. We next discuss each of these in detail.

3.2.1 The Plant Model

The plant model describes how the vehicle's position $\mathbf{x}(k)$ changes with time in response to a control input $\mathbf{u}(k)$ and a noise disturbance $\mathbf{v}(k)$, and in general has the form

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) + \mathbf{v}(k), \quad \mathbf{v}(k) \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}(k)) \qquad (3.1)$$

where $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$ is the (non-linear) state transition function. We use the notation $\mathbf{v}(k) \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}(k))$ to indicate that this noise source is assumed to be zero-mean Gaussian with covariance $\mathbf{Q}(k)$ [54].

The model we have used is based on point kinematics [114]. The control input $\mathbf{u}(k) = [T(k), \Delta\theta(k)]^T$ is a translation forward through the distance
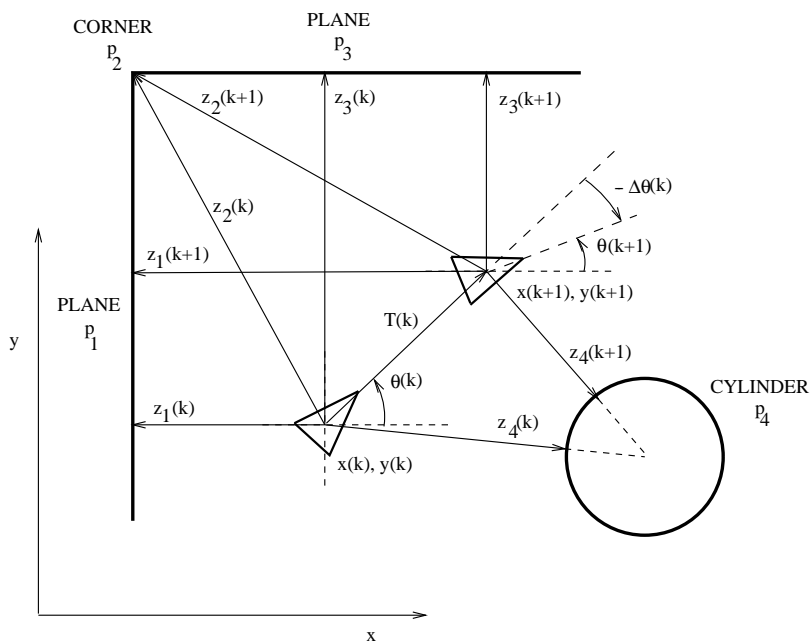
Figure 3.1: Localization by concurrent tracking of several beacons. The vector $\mathbf{x}(k) = (x(k), y(k), \theta(k))$ is the vehicle's position and orientation at time k. Four geometric beacons are in view to a servo-mounted sonar at time k and time $k + 1$: plane $\mathbf{p}_1$, corner $\mathbf{p}_2$, plane $\mathbf{p}_3$, and cylinder $\mathbf{p}_4$. The sonar measurements $\mathbf{z}_1(k)$ and $\mathbf{z}_3(k)$ are the shortest distance from the sensor to planes $\mathbf{p}_1$ and $\mathbf{p}_3$ at time k. The measurement $\mathbf{z}_2(k)$ is the distance from the sensor to corner $\mathbf{p}_2$ at time k. Measurement $\mathbf{z}_4(k)$ is the distance to the central axis of cylinder $\mathbf{p}_4$ less the radius of the cylinder.

$T(k)$ followed by a rotation anti-clockwise through the angle $\Delta\theta(k)$. The state transition function $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$ has the form [114]:

$$\mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) = \begin{bmatrix} x(k) + T(k)\cos\theta(k) \\ y(k) + T(k)\sin\theta(k) \\ \theta(k) + \Delta\theta(k) \end{bmatrix}. \qquad (3.2)$$

More sophisticated vehicle kinematic and dynamic models have been presented in the literature, such as those provided by Muir [106], Steer [121] and Alexander [1]. The point kinematic model has proven adequate in our experiments, and we feel its use is justified because our concern rests with the perception issues of position estimation. However, we do see the next step after position estimation to be position control, and even velocity control, with perception "in the loop". Provision of these capabilities would necessitate a more detailed plant model.

3.2.2 The Measurement Model

The robot is equipped with n_S sonar sensors. The position and orientation of sonar sensor s in the vehicle coordinate frame is given by the sensor position vector $\mathbf{b}_s(k) = (x'_s, y'_s, \alpha'_s(k))$. Perfect knowledge of $\mathbf{b}_s(k)$ is assumed to be available. When static sonars are used α'_s is constant, and hence $\mathbf{b}_s(k)$ is not a function of time; for convenience, time indices will often be dropped when this is the case. The values of $\mathbf{b}_s(k)$ for our vehicles are given in Appendix A. The raw sensor data from sensor s at time k takes the form of a *scan* $S_s(k) = \{\mathbf{r}_i(k)|1 \leq i \leq m_s(k)\}$ as described in Chapter 2. This raw sensor data goes through the process of RCD extraction described in Section 2.7 to yield n_O observed RCDs, which comprise the set of observations

$$Z(k) = \{\mathbf{z}_j(k)|1 \leq j \leq n_O\}. \qquad (3.3)$$

Because of the high angular uncertainty associated with an RCD, only the range of the RCD is used explicitly in this formulation. Orientation information is used in the matching algorithm, but not for computation of the vehicle position. Thus, each observation is the range associated with an RCD, defined in Chapter 2 to be the mode of the elements of the RCD. Note that an RCD can comprise just a single return, making our formulation equivalent for dense and sparse data.

The measurement model relates a sensor observation to the vehicle position and the geometry of the target that produced the observation, and has the form:

$$\mathbf{z}_j(k) = \mathbf{h}_{st}\left(\mathbf{x}(k), \mathbf{p}_t\right) + \mathbf{w}_j(k), \quad \mathbf{w}_j(k) \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_j(k)). \qquad (3.4)$$
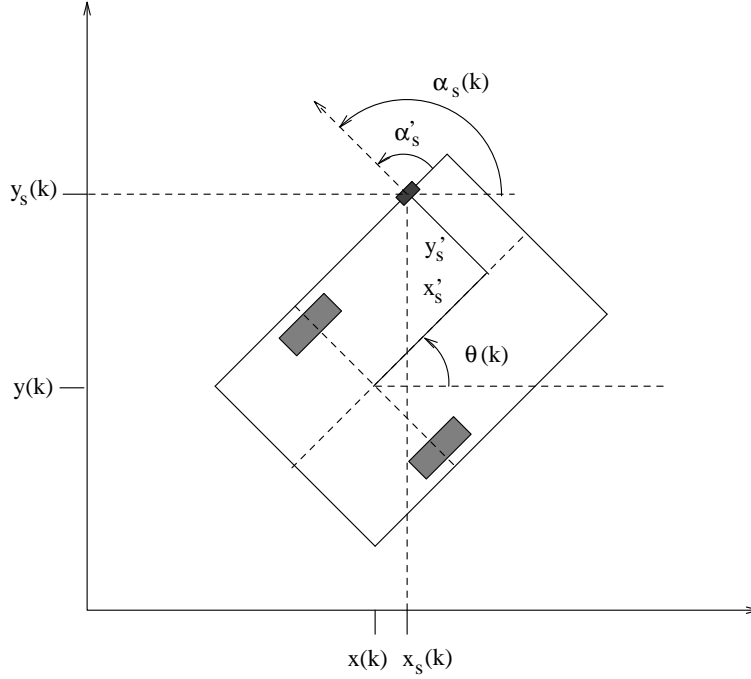
Figure 3.2: Definition of global and local sensor location vectors. $\mathbf{a}_s(k) = (x_s(k), y_s(k), \alpha_s(k))$ specifies the position and orientation of sensor s in the global frame, whereas $\mathbf{b}_s(k) = (x'_s, y'_s, \alpha'_s(k))$ specifies the sensor position and orientation in the local vehicle coordinate frame.

The target state vector takes the form $\mathbf{p}_t = (p_x, p_y)$ for corners and edges, $\mathbf{p}_t = (p_R, p_\theta, p_V)$ for planes, and $\mathbf{p}_t = (p_x, p_y, p_R)$ for cylinders, in accordance with the four target models presented in Chapter 2. The measurement function $\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t)$ expresses an observation $\mathbf{z}(k)$ from sensor s to target t as a function of the vehicle location $\mathbf{x}(k)$ and the target geometry. The target models of Chapter 2 were presented in terms of the global sensor position $\mathbf{a}_s(k) = (x_s(k), y_s(k), \alpha_s(k))$. The vehicle position $\mathbf{x}(k)$ defines the transformation from local to global coordinate frames (see Figure 3.2):

$$x_s(k) = x(k) + x'_s \cos(\theta(k)) - y'_s \sin(\theta(k)) \tag{3.5}$$

$$y_s(k) = y(k) + x'_s \sin(\theta(k)) + y'_s \cos(\theta(k)) \tag{3.6}$$

$$\alpha_s(k) = \theta(k) + \alpha'_s(k). \tag{3.7}$$

Substitution of Equations 3.5 to 3.7 into Equations 2.1, 2.3 and 2.5 yields the following measurement functions for each of our target types:

For planes:

$$\mathbf{h}_P\left(\mathbf{x}(k), \mathbf{p}_t\right) = p_V[\ p_R - (x(k) + x'_s \cos(\theta(k)) - y'_s \sin(\theta(k))) \cos(p_\theta)$$

$$- (y(k) + x'_s \sin(\theta(k)) + y'_s \cos(\theta(k))) \sin(p_\theta)\ ]. \tag{3.8}$$

For corners and edges:

$$\mathbf{h}_C\left(\mathbf{x}(k), \mathbf{p}_t\right) = \mathbf{h}_E\left(\mathbf{x}(k), \mathbf{p}_t\right) = [\ (p_x - x(k) - x'_s \cos(\theta(k)) + y'_s \sin(\theta(k)))^2$$

$$+ (p_y - y(k) - x'_s \sin(\theta(k)) - y'_s \cos(\theta(k)))^2\ ]^{\frac{1}{2}}. \tag{3.9}$$

And for cylinders:

$$\mathbf{h}_{CYL}\left(\mathbf{x}(k), \mathbf{p}_t\right) = -p_R + [\ (p_x - x(k) - x'_s \cos(\theta(k)) + y'_s \sin(\theta(k)))^2$$

$$+ (p_y - y(k) - x'_s \sin(\theta(k)) - y'_s \cos(\theta(k)))^2\ ]^{\frac{1}{2}}. \tag{3.10}$$

Each range value is assumed corrupted by a zero-mean, Gaussian noise disturbance $\mathbf{w}_j(k)$ with covariance $\mathbf{R}_j(k)$. As we discussed in Chapter 2, this Gaussian assumption is only justifiable for strong sonar returns. To facilitate visibility prediction, line segment endpoints and arc limit angles are explicitly contained in the map for plane and cylinder targets; these are not used directly in the EKF.

3.3 The Basic Localization Cycle

The goal of the cyclic localization procedure can be summarized as follows: Given the *a posteriori* vehicle position estimate[1] $\hat{\mathbf{x}}(k \mid k)$ and its covariance $\mathbf{P}(k \mid k)$ for time k, the current control input $\mathbf{u}(k)$, the current set of observations $Z(k+1)$ and the current map $M(k)$, compute the new *a posteriori* position estimate $\hat{\mathbf{x}}(k+1 \mid k+1)$ and its covariance $\mathbf{P}(k+1 \mid k+1)$. The following assumptions underlie the algorithm we shall present.

assumption 1: $\hat{\mathbf{p}}_t(k) = \mathbf{p}_t, \mathbf{\Lambda}_t(k) = \mathbf{0} \quad \forall\ t, k$
In the general navigation algorithm, the map is the set of target parameter vectors, their covariances, and target visibility angles: $M(k) = \{\hat{\mathbf{p}}_t(k), \mathbf{\Lambda}_t(k), \beta_t \mid 1 \leq t \leq n_T\}$. In this chapter we consider the limited case of localization with an accurate map, and thus perfect knowledge is assumed of each target in the map $M(k)$.

assumption 2: A static, two-dimensional environment
The environments under consideration are static, indoor, man-made scenes, such as offices, corridors, hospitals, and factories. These environments have been called "well carpentered" [110], and would be

---

[1]The term $\hat{\mathbf{x}}(k_2 \mid k_1)$ signifies the estimate of the vector $\mathbf{x}$ at time step k_2 based on all information available up to and including time step time k_1.
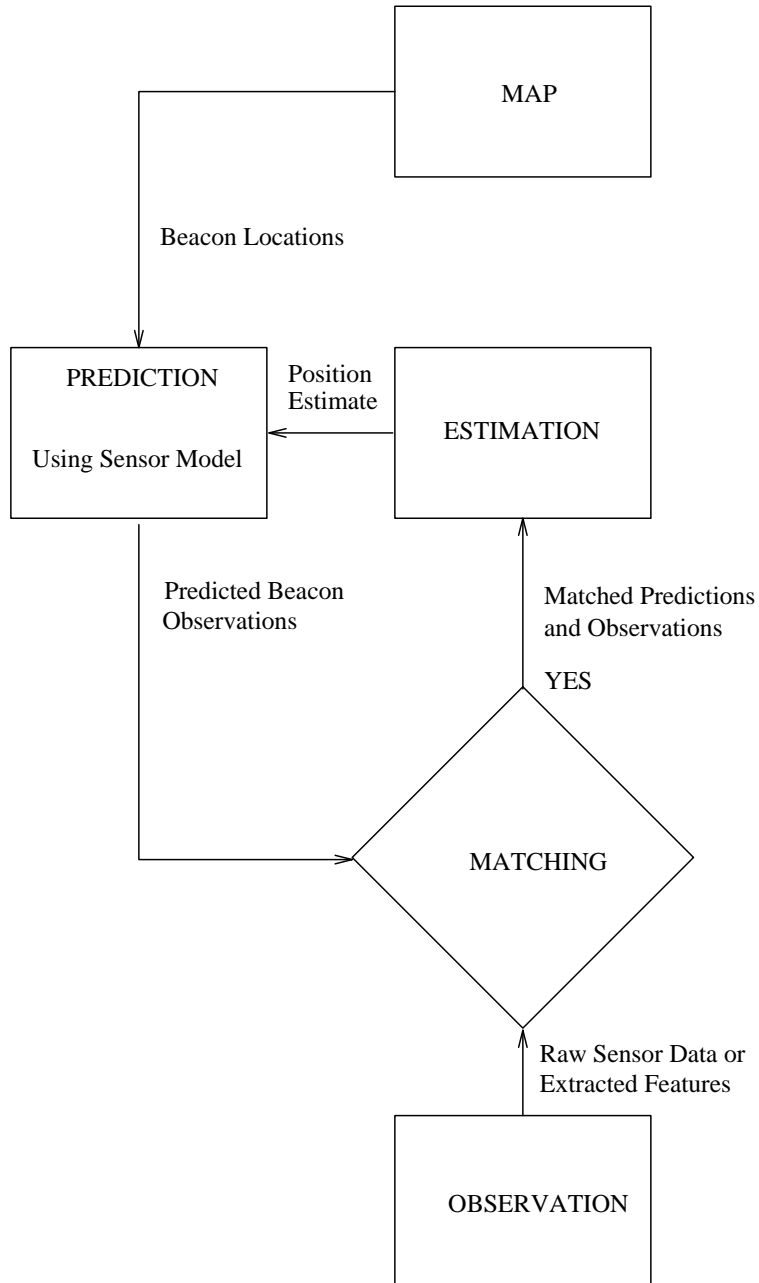
Figure 3.3: The localization algorithm.

characterized primarily by vertically oriented planes, corners, and edges. Because we explicitly model cylindrical targets, our approach accommodates the cylindrical wall junctions, table legs, and pillars that are common in many indoor scenes. However, the inherent three-dimensionality of the world will impose some difficulties.

assumption 3: $p[\mathbf{x}(k)|Z^k] \sim \mathrm{N}\left(\hat{\mathbf{x}}(k \mid k), \mathbf{P}(k \mid k)\right)$

The notation $Z^k = \{Z(i)|1 \leq i \leq k\}$ refers to the cumulative set of all observations up to time k. The starting assumption from which we develop the recursive update algorithm is that the true vehicle position $\mathbf{x}(k)$ conditioned on Z^k is a random variable normally distributed about the current state estimate $\hat{\mathbf{x}}(k \mid k)$ with covariance matrix $\mathbf{P}(k \mid k)$. This implies that the current state estimate and covariance are sufficient statistics to describe the true vehicle position. Strictly speaking, because $\hat{\mathbf{x}}(k \mid k)$ only approximates the exact conditional mean, the matrix $\mathbf{P}(k \mid k)$ is an approximate mean square error rather than a covariance [7].

   The algorithm consists of the following steps: position prediction, observation, measurement prediction, matching, and estimation. Figure 3.3 presents an overview of this cyclic process. We now proceed to discuss each of these steps in detail.

3.3.1   Vehicle Position Prediction

First, using the plant model and a knowledge of the control input $\mathbf{u}(k)$, we predict the robot's new location at time step $k + 1$:

$$\hat{\mathbf{x}}(k + 1 \mid k) = \mathbf{f}\left(\hat{\mathbf{x}}(k \mid k), \mathbf{u}(k)\right). \tag{3.11}$$

We next compute $\mathbf{P}(k + 1 \mid k)$, the variance associated with this prediction:

$$\mathbf{P}(k + 1 \mid k) = \nabla \mathbf{f}\ \mathbf{P}(k \mid k)\ \nabla \mathbf{f}^T + \mathbf{Q}(k) \tag{3.12}$$

where $\nabla \mathbf{f}$ is the Jacobian of the state transition function $\mathbf{f}(\hat{\mathbf{x}}(k \mid k), \mathbf{u}(k))$ obtained by linearizing about the updated state estimate $\hat{\mathbf{x}}(k \mid k)$

$$\nabla \mathbf{f} = \begin{bmatrix} 1 & 0 & -T(k)\sin(\hat{\theta}(k|k)) \\ 0 & 1 & T(k)\cos(\hat{\theta}(k|k)) \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.13}$$

3.3.2   Observation

The next step is to obtain the observation set $Z(k + 1)$ from the vehicle's sensors from the new vehicle location. This set is comprised of n_O observed RCDs.

3.3.3   Measurement Prediction

Next, for each sensor we use the predicted robot location $\hat{\mathbf{x}}(k+1 \mid k)$ and the current map $M(k)$ to generate predicted observations of each target $\mathbf{p}_t$:

$$\hat{\mathbf{z}}_i(k+1) = \mathbf{h}_{st}(\mathbf{p}_t, \hat{\mathbf{x}}(k+1 \mid k), \mathbf{b}_s), \qquad i = 1, \cdots, n_P \qquad (3.14)$$

to yield the set of predictions

$$\hat{Z}(k+1) = \{\hat{\mathbf{z}}_i(k+1) | 1 \leq i \leq n_P\} \qquad (3.15)$$

which contains n_P predicted RCDs. The predicted state estimate $\hat{\mathbf{x}}(k+1 \mid k)$ is used to compute the measurement Jacobian $\nabla \mathbf{h}_i$ for each prediction. For planes,

$$\nabla \mathbf{h}_i = p_V \begin{bmatrix} -\cos(p_\theta) \\ -\sin(p_\theta) \\ x_s' \ \sin(\hat{\theta} - p_\theta) + y_s' \cos(\hat{\theta} - p_\theta) \end{bmatrix}^T \qquad (3.16)$$

while for cylinders, corners, and edges

$$\nabla \mathbf{h}_i = \frac{1}{d} \begin{bmatrix} \hat{x} + x_s' \cos\hat{\theta} - y_s' \sin\hat{\theta} - p_x \\ \hat{y} + x_s' \sin\hat{\theta} + y_s' \cos\hat{\theta} - p_y \\ [(\hat{x} + x_s' \cos\hat{\theta} - y_s' \sin\hat{\theta} - p_x)(-x_s' \sin\hat{\theta} - y_s' \cos\hat{\theta}) \\ +(\hat{y} + x_s' \sin\hat{\theta} + y_s' \cos\hat{\theta} - p_y)(x_s' \cos\hat{\theta} - y_s' \sin\hat{\theta})] \end{bmatrix}^T \qquad (3.17)$$

where d is the distance from the predicted location of sensor s to the point (p_x, p_y).

3.3.4   Matching

The goal of the matching procedure is to produce an assignment from measurements $\mathbf{z}_j(k)$ to targets $\mathbf{p}_t$. For each prediction and observation corresponding to the same sensor s, we first compute the innovation $\nu_{ij}(k)$.

$$
\begin{aligned}
\nu_{ij}(k+1) &= [\mathbf{z}_j(k+1) - \hat{\mathbf{z}}_i(k+1)] \\
&= [\mathbf{z}_j(k+1) - \mathbf{h}_i(\mathbf{p}_t, \hat{\mathbf{x}}(k+1 \mid k))].
\end{aligned}
\qquad (3.18)
$$

If prediction i and observation j correspond to different sensors, the innovation $\nu_{ij}(k+1)$ is set to infinity. The innovation covariance can be found by linearizing Equation 3.4 about the prediction, squaring, and taking expectations to yield

$$
\begin{aligned}
\mathbf{S}_{ij}(k+1) &\equiv \mathrm{E}\left[\nu_{ij}(k+1)\nu_{ij}^T(k+1)\right] \\
&= \nabla \mathbf{h}_i \ \mathbf{P}(k+1 \mid k) \ \nabla \mathbf{h}_i^T + \mathbf{R}_i(k+1).
\end{aligned}
\qquad (3.19)
$$

A *validation gate* is used to determine the correspondence between predictions and observations [7]:

$$\nu_{ij}(k+1) \, \mathbf{S}_{ij}^{-1}(k+1) \, \nu_{ij}^{T}(k+1) \leq g^2. \tag{3.20}$$

This equation is used to test each sensor observation $\mathbf{z}_j(k+1)$ for membership in the validation gate for each predicted measurement. When a single observation falls in a validation gate, we get a successful match. Measurements which do not fall in any validation gate are simply ignored for localization. More complex data association scenarios can arise when a measurement falls in two validation regions, or when two or more measurements fall in a single validation region. At this stage, such measurements are simply ignored by the algorithm, as outlier rejection is vital for successful localization. Sufficient matching percentages are achieved in the experimental results presented below simply by making use of unambiguous matches.

Variations on this validation gate-based matching process are employed extensively in the literature on target tracking as well as in robotics; Bolle and Cooper [11] and Faugeras and Ayache [51], for example, employ the equivalent Mahalanobis distance to match features in visual images. A variety of algorithms for dealing with more complex data association scenarios have been presented in the multitarget tracking literature. When a single measurement falls within two validation gates, a multiple hypothesis tracking (MHT) data association scheme could be undertaken [111]. For the case of multiple observations in a single validation region, a simple algorithm which uses the closest match is the nearest neighbor standard filter (NNSF). A more sophisticated alternative is the probabilistic data association filter (PDAF), which uses a weighted sum of matches in the validation region, combined with the probability that none of the matches is indeed correct.

Our model-based localization system has not yet made use of these data association methods. Because false matches can lead to EKF divergence, if a single match is held in doubt, it is better to discard the measurement than to be led astray by a false application. This rules out the NNSF, as one is quite likely to get false measurements. The PDAF relies on "the underlying assumption ... that the false measurements are randomly distributed in space, time, and intensity; [probabilistic data association techniques] may fail when there is persistent or time-correlated interference" [7]. For localization, our false measurements generally result from unmodeled objects and other repeatable effects such as the multiple reflections discussed in Chapter 2. Thus we feel that the full PDAF is unsuited for this sonar application. Based on recent research in collaboration with Ingemar Cox [35], we are optimistic regarding the potential of Reid's MHT filter to provide improved localization performance in extremely complex environments.

3.3.5   Estimation

The final step is to use successfully matched predictions and observations to compute $\hat{\mathbf{x}}(k + 1 \mid k + 1)$, the updated vehicle location estimate. To do so we use a parallel update procedure [134]. We first stack the validated measurements $\mathbf{z}_j(k + 1)$ into a single vector to form $\mathbf{z}$(k+1), the composite measurement vector for time $k + 1$, and designate the composite innovation $\nu(k+1)$. Next, we stack the measurement Jacobians $\nabla\mathbf{h}_i$ for each validated measurement together to form the composite measurement Jacobian $\nabla\mathbf{h}$. Using a stacked noise vector $\mathbf{R}(k + 1) = diag[\mathbf{R}_j(k + 1)]$, we then compute the composite innovation covariance $\mathbf{S}$(k+1) as in Equation 3.19. We then utilize the well-known result [7] that the Kalman gain can be written as

$$\mathbf{W}(k + 1) = \mathbf{P}(k + 1 \mid k)\ \nabla\mathbf{h}^T\ \mathbf{S}^{-1}(k + 1) \tag{3.21}$$

to compute the updated vehicle position estimate

$$\hat{\mathbf{x}}(k + 1 \mid k + 1) \quad = \quad \hat{\mathbf{x}}(k + 1 \mid k) + \mathbf{W}(k + 1)\nu(k + 1) \tag{3.22}$$

with associated variance

$$\mathbf{P}(k + 1 \mid k + 1) \quad = \quad \mathbf{P}(k + 1 \mid k) - \mathbf{W}(k + 1)\mathbf{S}(k + 1)\mathbf{W}^T(k + 1). \tag{3.23}$$

3.4   Algorithm Summary

To satisfy the general task of autonomous, long-term navigation, this algorithm must be extended to incorporate relocation and map building and maintenance. Map building, the subject of Chapter 4, attempts to explain the rejects from the matching process above to update the map. However, the experiments to follow use an *a priori* map, carefully generated by hand, using a tape measure. Nevertheless, we include a map update step in the algorithm summary of Figure 3.4, to show where it goes.

Relocation will be automatically invoked at start-up and after becoming lost. In general, relocation should use search to evaluate all possible data-to-model associations between the current observation set $Z(k)$ and the map $M(k)$. However, in the implementations that follow, a generic relocation procedure was not available. Instead, a much simpler planar-targets-only docking procedure was used for this purpose. This procedure could successfully initialize the position estimate when the human operator placed the vehicle within 20 centimeters and 10 degrees of a pre-designated home position—usually one of the corners of the room. The docking procedure consisted of one iteration of the basic EKF cycle described above, modified as follows:

1. The *a priori* vehicle position estimate is set to the home position.

2. The initial position covariance is set to a large value (10 centimeter standard deviation in x and y; 5 degrees standard deviation in orientation).

3. The validation gate size g is set to infinity.

4. If three or more unambiguous matches are generated for at least two non-parallel walls, apply the EKF to compute a vehicle location estimate; else, return no answer.

5. To verify the hypothesized correspondence, a new set of predictions is generated from the estimated location. If the set of targets predicted from the home and estimated locations are the same, return the estimate as the vehicle's initial position; else, return no answer.

This procedure was very useful because it had an extremely low rate of false positives; if any correspondence ambiguity was detected, this procedure simply failed to return an answer. Hence, docking worked best in corners of the room, and provided a great saving in effort when conducting experiments, because there was no need for the operator to hand-measure the vehicle's position to begin an experiment. In the future, correspondence verification (step 5) should have wider application for the provision of increased continuous localization robustness, especially when combined with the iterated Kalman filter [54].

To decide when the vehicle is lost we use a very simple collision test that expands the vehicle dimensions by the position covariance $\mathbf{P}(k \mid k)$ plus a safety factor of 10 centimeters and performs an intersection test of this bounding polygon with the map. With these extensions, Figure 3.4 summarizes the full navigation procedure.

3.5 Off-line Processing of Densely Sampled Data

To illustrate the use of the algorithm with real data for point and line targets, the algorithm was run off-line using data obtained from precisely known positions, adding process noise artificially to simulate vehicle motion. These scans were acquired primarily for map building, and are used extensively in Chapter 4. Figures 3.5 to 3.10 illustrate one cycle of the localization algorithm. The system has a 2-D global representation of the environment consisting of a list of line segments, and the corners they define, assumed known with absolute confidence. The vehicle starts from a known location. This starting point is the left-most triangle in Figure 3.5.

repeat

1. **if** (vehicle_moving = TRUE)
   then continue motion
   else send next motion command
   endif

2. $\hat{\mathbf{x}}(k+1 \mid k)$, $\mathbf{P}(k+1 \mid k)$ ← vehicle_position_pre-
   diction($\hat{\mathbf{x}}(k \mid k)$, $\mathbf{P}(k \mid k)$, $\mathbf{u}(k)$)

3. $Z(k+1)$ ← observation($D(k+1)$)

4. $\hat{Z}(k+1)$ ← measurement_prediction($\hat{\mathbf{x}}(k+1 \mid k)$, $\mathbf{P}(k+1 \mid k)$,
   $Z(k+1), M(k)$)

5. $\mathbf{z}(k+1), \hat{\mathbf{z}}(k+1), \nabla\mathbf{h}(k+1), \mathbf{S}(k+1)$ ← matching($Z(k+1), \hat{Z}(k+1)$,
   $\mathbf{R}(k+1), \mathbf{P}(k+1 \mid k)$)

6. $\hat{\mathbf{x}}(k+1 \mid k+1)$, $\mathbf{P}(k+1 \mid k+1)$ ← estimation($\hat{\mathbf{x}}(k+1 \mid k)$,
   $\mathbf{P}(k+1 \mid k)$, $\mathbf{z}(k+1), \hat{\mathbf{z}}(k+1), \nabla\mathbf{h}(k+1), \mathbf{S}(k+1), \mathbf{R}(k+1)$)

7. $M(k+1)$ ← map_update($M(k), Z(k+1), \hat{Z}(k+1), \hat{\mathbf{x}}(k+1 \mid k+1)$)

8. **if** (collision($\hat{\mathbf{x}}(k+1 \mid k+1), \mathbf{P}(k+1 \mid k+1), M(k)$) = TRUE)
   then if (relocation($M(k), Z(k+1)$) = SUCCESSFUL)
         then SOS = FALSE
         else SOS = TRUE
         endif
   endif

until (((vehicle_moving = FALSE)**and**(motion_queue =
   EMPTY))**or**(SOS = TRUE))

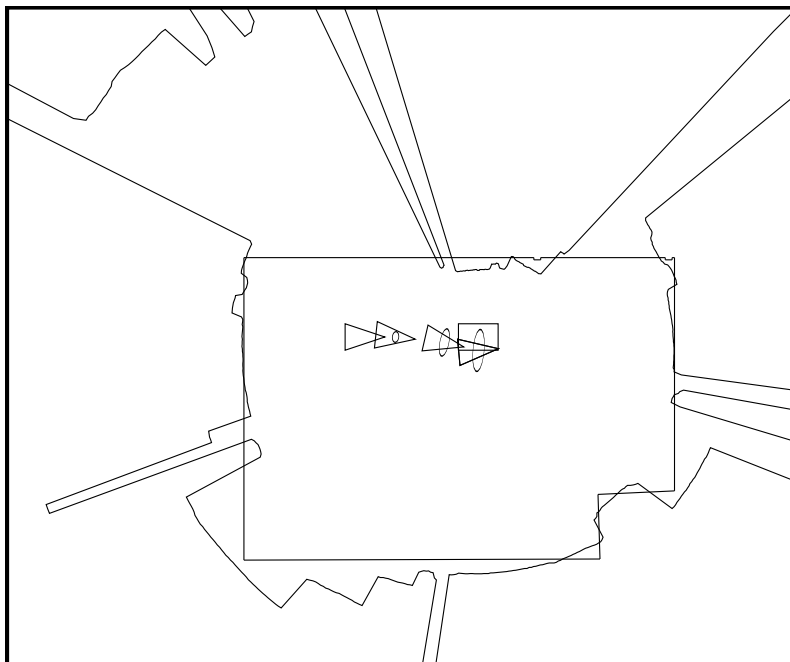Figure 3.4: A summary of the full model-based navigation algorithm.

Figure 3.5: The predicted vehicle motion, and accompanying error ellipses, for time steps 0 to 3. No observations were taken at time steps 1 and 2. The sonar scan taken at time step 3 is displayed with reference to the *a priori* vehicle position estimate. The rectangle shows the vehicle's true position at time step 3.

The initial state covariance matrix $\mathbf{P}(0\,|\,0)$ is set to zero. We show a run in which observation is suppressed for the first two time-steps.

Prediction and Observation

Figure 3.5 shows a sonar scan of 612 measurements taken at time step 3. Based on the predicted vehicle location, Figure 3.6 shows predicted beacon observations generated using equation 3.14, with corresponding validation regions generated using Equation 3.19. A value of $\beta_t = 30°$ is used for corners, planes, and cylinders, while a value of $\beta_t = 15°$ is used for edges. The validation regions take the form of circular arcs blurred in the perpendicular direction by range measurement noise and uncertainty in the *a priori* vehicle position estimate. In conjunction with this, Figure 3.7 shows RCDs of width $\beta \geq 10°$ extracted from the original scan.
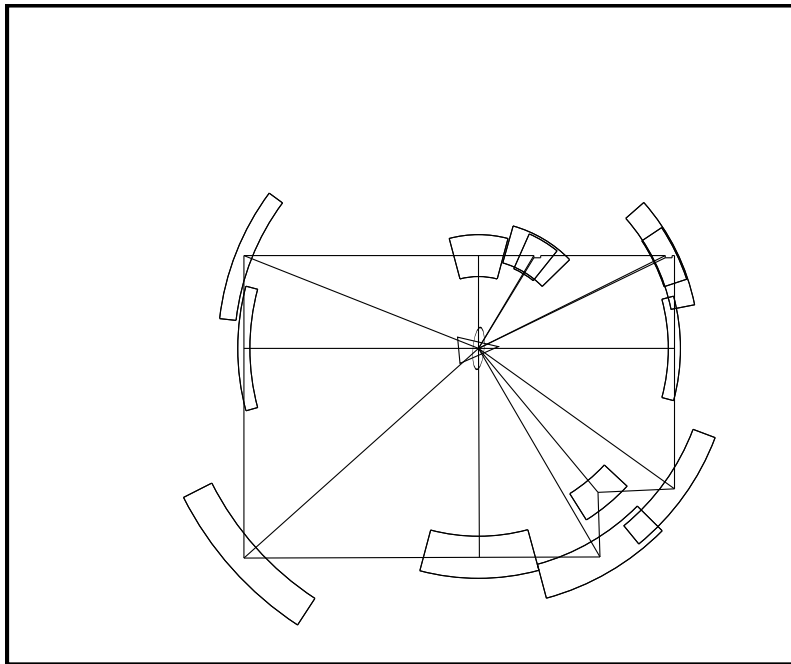
Figure 3.6: Predicted RCDs, with validation gates shown for each prediction. Note the influence of the position estimate covariance on the size of each validation gate: validation gates for horizontal line targets are larger than validation gates for vertical line targets, because the vehicle is less certain of its vertical position. Also, for this run edge visibility angles were set to be half the value for corners and planes.
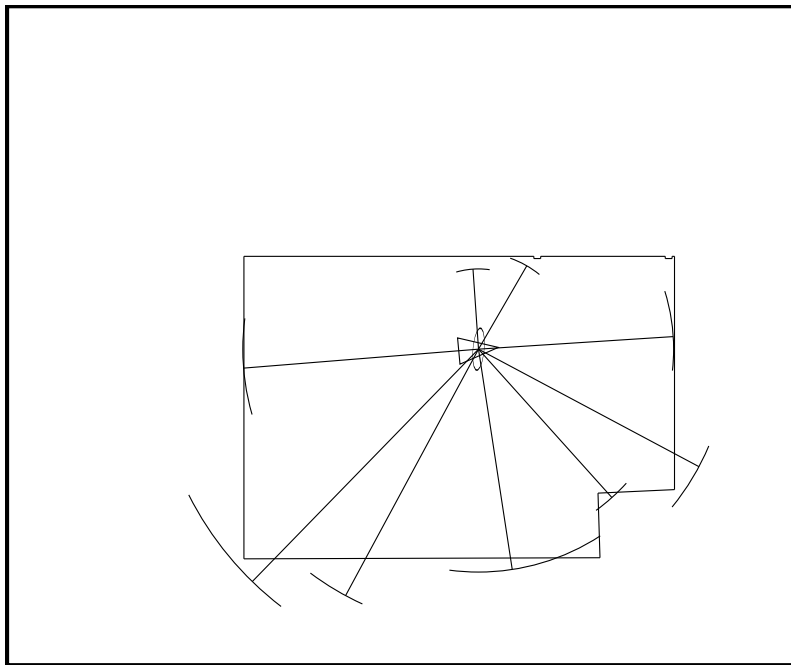
Figure 3.7: Observed RCDs displayed with reference to the *a priori* vehicle position estimate.
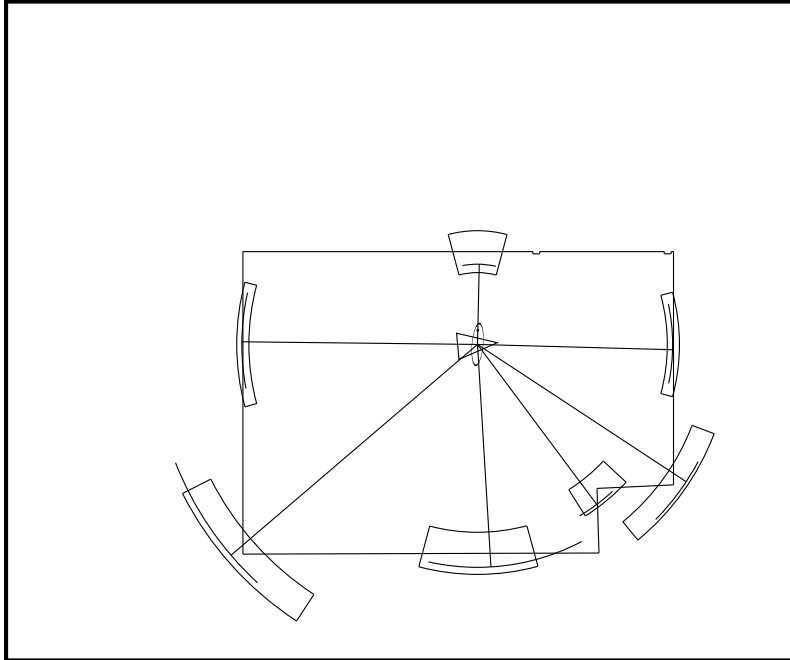
Figure 3.8: Matched predictions and observations.

Matching

Figure 3.8 shows the result of matching the predicted RCDs in Figure 3.6
with the observed RCDs in Figure 3.7 using Equation 3.20. The figure
shows the validation regions and observed RCDs for the seven matches
found.

Estimation

Using these seven matches, the vehicle's *a posteriori* position estimate
$\hat{\mathbf{x}}(k+1 \mid k+1)$ and associated variance $\mathbf{P}(k+1 \mid k+1)$ are computed us-
ing Equations 3.22 and 3.23. Figures 3.9 and 3.10 show the extracted RCDs
and the original scan displayed with respect to the updated position.

   After inspection of Figures 3.5 to 3.10, the careful reader may have
asked: by what means was the orientation of the vehicle updated to bring
the original scan back into alignment with the room in Figure 3.10? From
the arguments of Section 2.7.2, we know that the orientation of an RCD
is not a reliable measurement of target bearing, especially when a target is
partially occluded. For this reason, our localization algorithm does not use
the sensor orientation $\alpha_j(k)$ of a return directly in the filter. In the experi-
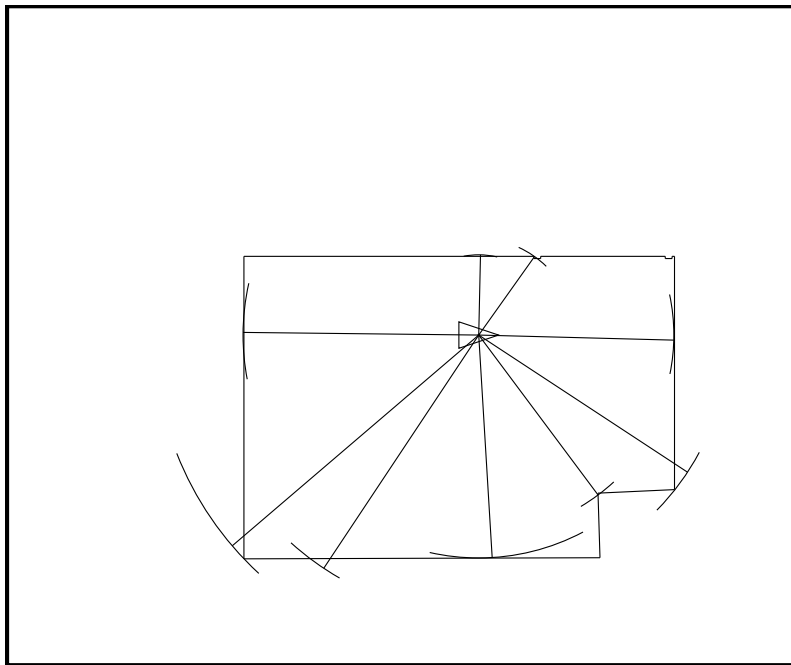ments to be reported below, updates to the orientation of the vehicle come

Figure 3.9: Observed RCDs displayed with reference to the *a posteriori* estimated position.
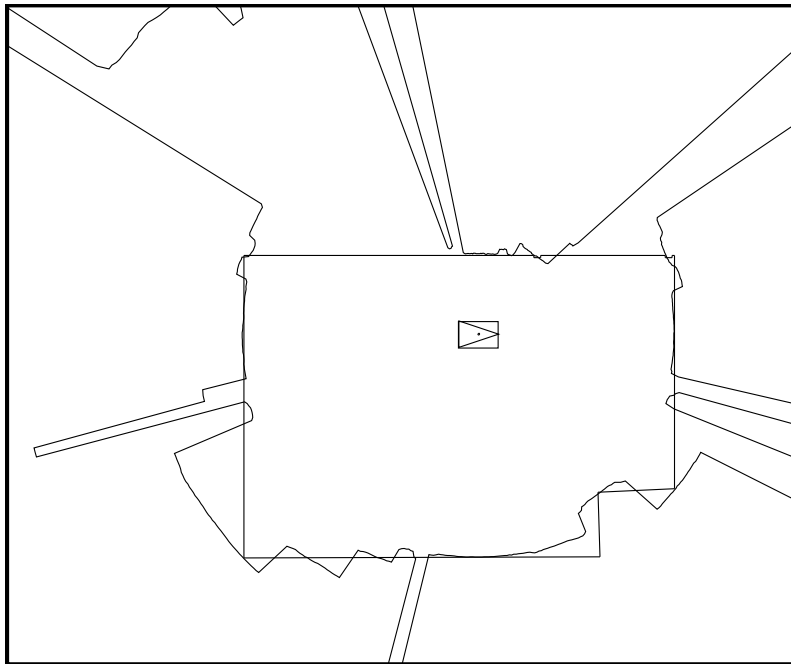
Figure 3.10: Original scan displayed with reference to the *a posteriori* estimated position. The triangle and rectangle show the estimated and true vehicle positions, respectively. The error ellipse, too small to be seen, is the dot at the center of the triangle.

implicitly via range measurements from different sensors around the vehicle perimeter. We use the sensor orientation as a *constraint for correspondence*, to determine if a sonar return might possibly have originated from a particular target using equation 2.7. After this test is passed, matching proceeds by validation gating in range; for each unambiguous match, new entries are added to the composite predicted and observed measurement vectors $\hat{\mathbf{z}}(k+1)$ and $\mathbf{z}(k+1)$.

From these arguments, it is clear that it would be difficult to estimate the vehicle orientation solely on the basis of a single scan from the Polaroid sonar. For this reason, the original, single-scanner implementation for the robot Eric planned to use a digital compass. Hence, to generate Figures 3.5 to 3.10, a simulated compass measurement was used to provide a direct angle update for the EKF. Our subsequent experiments have in fact never used a compass, but we include these figures nevertheless, because they provide a useful step-by-step illustration of the process. Because of the unique layout of the sonar transducers on the Robuter mobile robot, the experiments to follow bypass this issue.

3.6 Sparse Data Error Model

Several versions of this navigation algorithm have been implemented on two Robuter mobile robots through the course of our research. The system has not yet been perfected, but the experimental results we present here demonstrate the feasibility of our approach. Appendix A presents hardware and software details of the implementations.

As shown in detail in Appendix A, each Robuter has a ring of static sonars. The SKIDS[2] vehicle is equipped with six transducers, while the Oxford Robuter, pictured in Figure 3.11, has eight transducers. Localization via complete scans, as illustrated in the previous section, is not feasible because of the long time required to obtain a complete scan. The advantage of a ring of fixed sonars is that a set of returns over a wide spread of orientations can be quickly obtained. However, the disadvantage is that interpreting the data is made more difficult because each range measurement has no local support, as in the densely sampled scan of Figure 2.1 in Chapter 2. From the arguments of Section 2.6, recall that sonar's range is only accurate for strong returns. Weak returns might be delayed by as much as the duration of the transmitted pulse, adding up to 19 centimeters to the true range when a 1.13 millisecond pulse is transmitted. A question that must be considered is, given a return, is it strong or weak? This is easily answered with dense data by considering the range of neighboring returns. The uncertain mechanism by which weak returns trip the

---

[2] *SKIDS* is an acronym for *S*ignal and *K*nowledge *I*ntegration with *D*ecisional control for multi-*S*ensory systems, the title of ESPRIT project P1560.

Figure 3.11: The Oxford Robuter.

detector threshold results in large range fluctuations for small changes in transducer orientation. Since local support is not present, it is impossible to know whether a single return from a ring of sonars is strong or weak from time-of-flight information only. Thus, the range accuracy of isolated returns cannot be predicted.

Fortunately, there is a way out of this dilemma. The preceding paragraph considered the long-range sensing mode, which is standard for the Polaroid system. With a short transmitted pulse, weak returns are essentially eliminated, and range can be trusted to within 1 centimeter. Thus, the experiments which follow make exclusive use of the Robuter's short-range sensing mode. Range values are assumed to be normally distributed about the true range with a standard deviation of 1 centimeter. This value probably underestimates the accuracy of the short-range sensing mode, and the true error distribution will of course not be exactly Gaussian, but we feel the success of our experiments justifies this assumption.

3.7   Tracking Planar Targets

Our first implementation of the algorithm, using the SKIDS vehicle, made use of a map comprised exclusively of line segment targets; our results were

reported in [89]. The vehicle moved continuously, following a sequence of straight-line motions and on-the-spot rotations supplied by a Voronoi diagram trajectory planner. The implementation was based on the principle that range measurements could only be matched if the sonar transducer was nearly perpendicular to a planar target when the measurement was obtained, using a value of $\beta_i = 26$ degrees in equation 2.7. (Thus nearly perpendicular in this case means within 13 degrees of the orthogonal direction to a wall beacon.)

Because of the configuration of the six fixed sensors on the SKIDS Robuter (one sonar facing forwards and backwards, two facing left and right) the system was restricted to follow paths that were nearly perpendicular to the walls of the room. Despite this restriction, the system demonstrated that accurate localization is achievable with sonar, provided one has a good sensor model. The model presented in Chapter 2 tells us that for planes, corners, and cylinders, sonar provides very precise distance measurements, but at other times misleading multiple reflections; the key to using sonar is knowing when it is telling the truth. Our earliest implementation achieved this by only attempting to use updates from 1st-order, planar targets and using tight validation gates to reject outliers.

However, this implementation was unable to accommodate the multitude of returns actually produced by edge and corner targets in the room, even at vehicle orientations nearly perpendicular to the room. It was clear that a system was required which could predict and explain sonar observations for all target types: planes, cylinders, corners, and edges.

3.8   Tracking Planes, Corners, and Cylinders

Using the knowledge derived from the initial implementation results described in [89], an improved system was developed that could track all four types of targets described in Chapter 2. Unfortunately, communication delays between the robot and workstation introduced significant delays that made accurate time-stamping of the sonar returns impossible. Hence, all the experiments we report below are "stop, look, move" runs in which the vehicle came to a complete stop to acquire each sonar data set. While the logical step to overcome this problem would be to download our code to run on the 68020 microprocessor on-board the vehicle, this task would have unfortunately entailed deciphering a great deal of vehicle control software written in French, not to mention many battles with the customs agents of Heathrow and Charles de Gaulle Airports.

3.8.1   Implementation OxRob-1

The results we show here are from a "stop, look, move" run in the Oxford AGV lab, using the Oxford Robuter. The lab is characterized primarily by rough, painted brick surfaces, and has large pillars with rounded corners, each with a 2.25 inch radius of curvature. The room model was measured by hand. The lab is a confined space with workstations and other mobile robots that present complex three-dimensional targets. To bypass three-dimensional effects in testing the program, *clutter areas* were introduced into the model. The clutter areas for this run are shown as shaded regions in Figure 3.12. Prediction is suppressed for targets in clutter areas, and vehicle travel into these areas is prevented.

Appendix A gives vehicle dimensions and sensor positions. The docking procedure described earlier was used to initialize the vehicle position to the known location in the lower right-hand part of the room. Figure 3.13 shows the vehicle location estimated by the algorithm and the dead-reckoning estimate of the vehicle's internal controller for each stage of the complete run. Figure 3.14 shows the validated returns used to update the vehicle position for various stages of the run. For these runs, a path planner was not yet developed. Vehicle motion was guided at the SUN workstation by an operator specifying forward, reverse, left or right motion commands in discrete steps, usually 20 centimeters or 30 degrees.

A value of 2 was used for g, the validation gate "number of sigmas" in Equation 3.20. The matrix $\mathbf{Q}(k)$ was given values to reflect 5 centimeters of position error and 4 degrees of orientation error for each meter of translation or 90 degrees of rotation. At this predicted location, predicted range measurements are generated for each target in the map using the target models presented in Chapter 2. A value of 30 degrees was used for β_{MAX}, the maximum visibility angle for each target. If the predicted sensor orientation is not within 15 degrees of the true bearing to a target, a NULL prediction that cannot be matched is generated. The innovation variance $\mathbf{S}(k+1)$ is computed using $\sigma_r = 1$ centimeter for the standard deviation of range measurement error.

Figure 3.15 shows the number of validated returns for each step of the room. Thirty-seven percent of the 976 returns taken during the run were validated, for an average of 3 matched returns per time step. Figure 3.16 shows the cumulative matching percentage *vs* time. We were surprised by the frequency of matches to the cylindrical targets defined by the pillars of the room. These turned out to be very good navigation beacons. One target that was frequently matched during the run was a vertically oriented 2 centimeter diameter electrical pipe, a scarcely visible dot in the lower right-hand corner of Figures 3.12 to 3.14, that was left out of earlier room models. This was one of many instances in our research in which sonar surprised us by consistently observing environment features that we
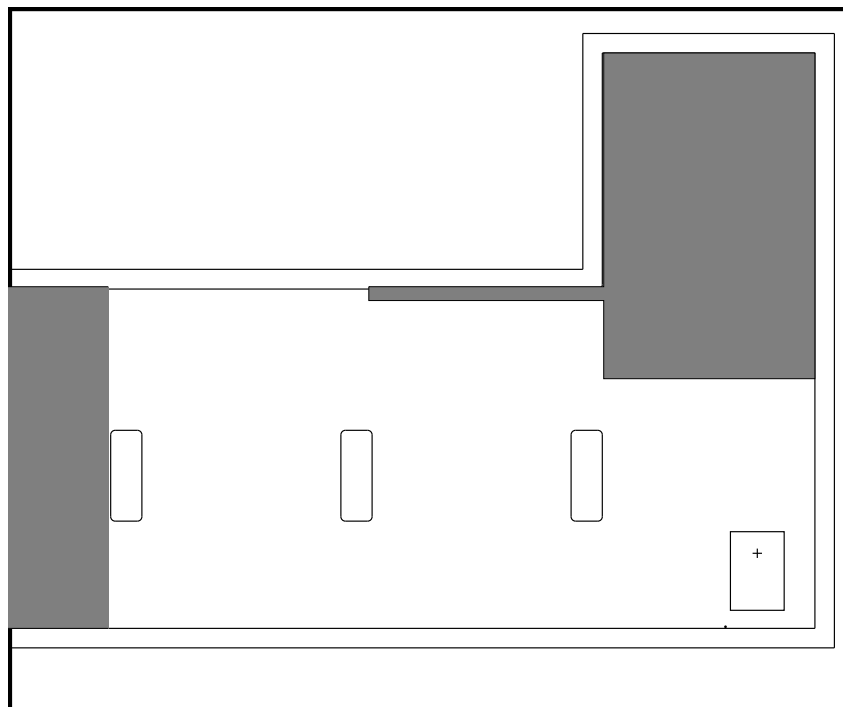
Figure 3.12: Model of the room for localization run OxRob-1. The shaded areas represent regions of clutter, for which predicted observations were not generated. The shaded region in the upper right was occupied by the Oxford AGV. The thin shaded section jutting out to the left from this area enclosed a radiator. SUN workstations and people occupied the shaded region to the left.
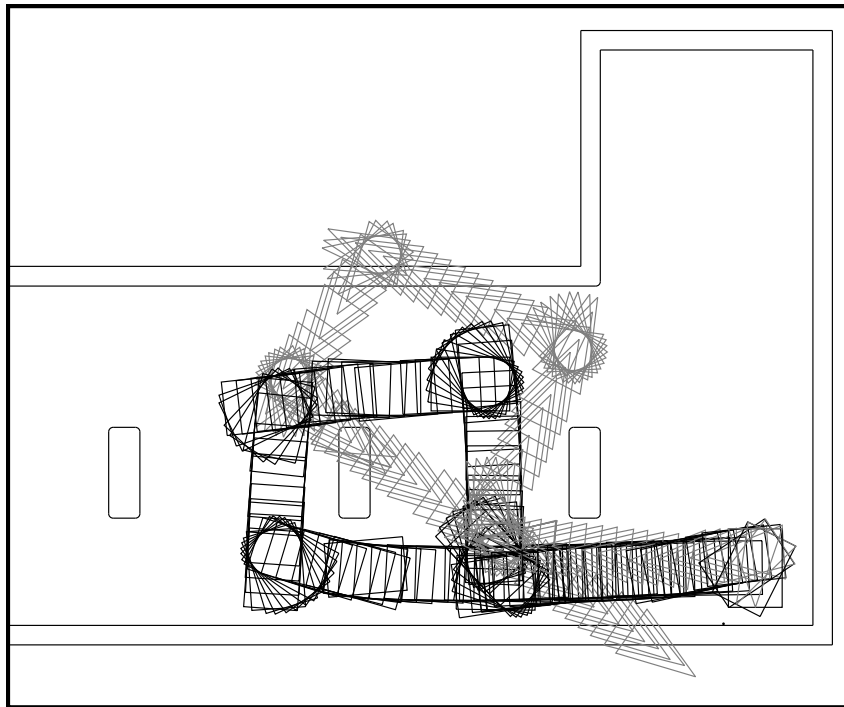
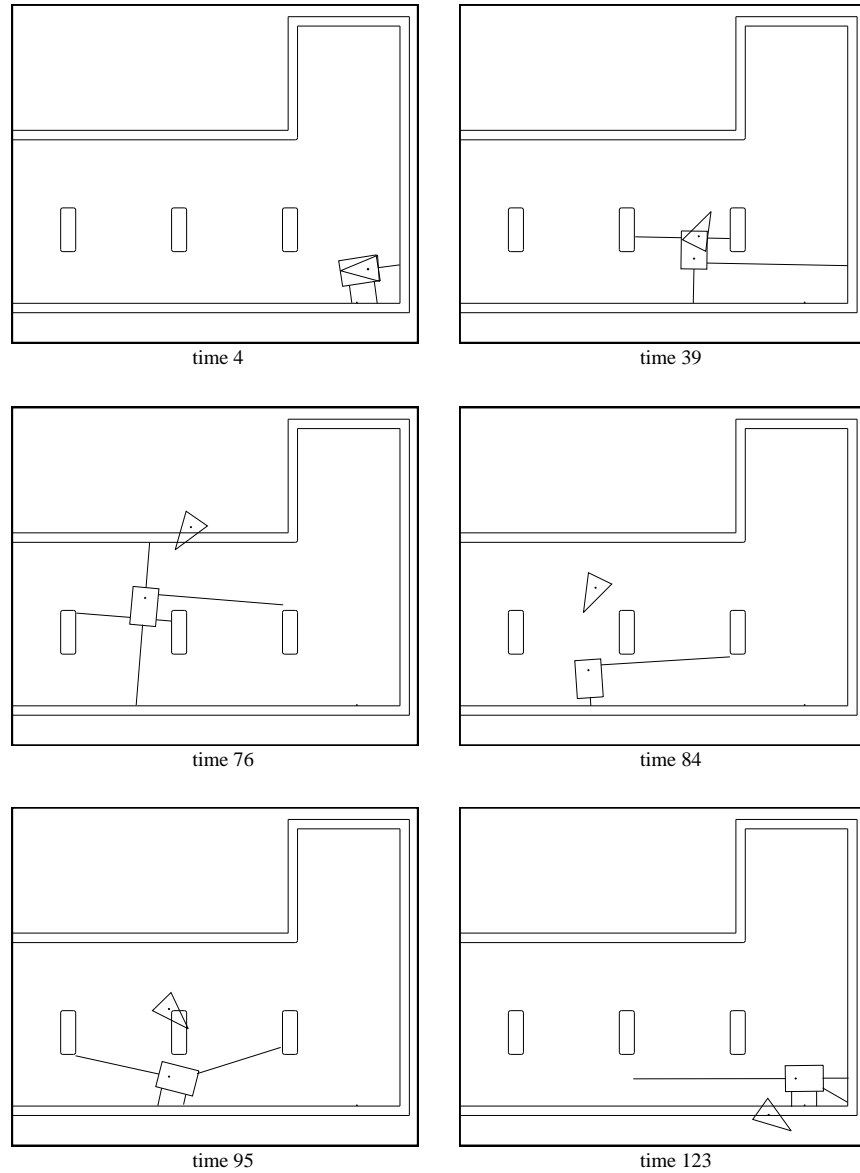Figure 3.13: Estimated (rectangle) and odometric (triangle) positions for each step of the complete run.

Figure 3.14: A localization run in the Oxford AGV laboratory, showing the validated sonar range measurements used to update the vehicle position at various stages in the run. The triangle shows the current position as estimated solely by odometry. The rectangle shows the *a posteriori* position estimate produced by the algorithm.
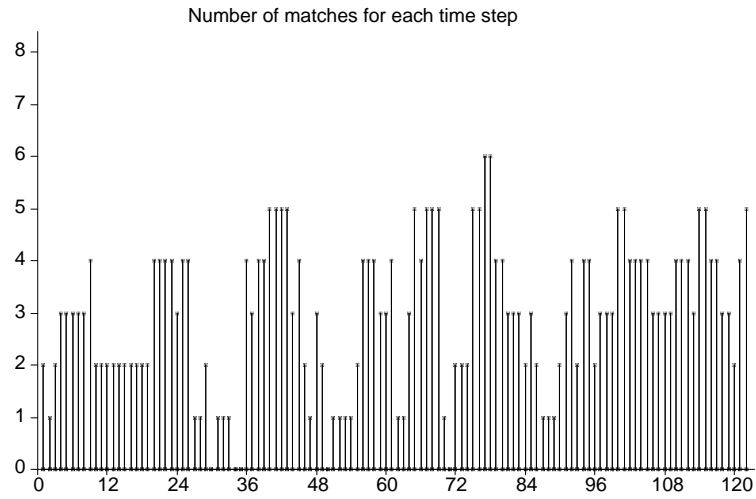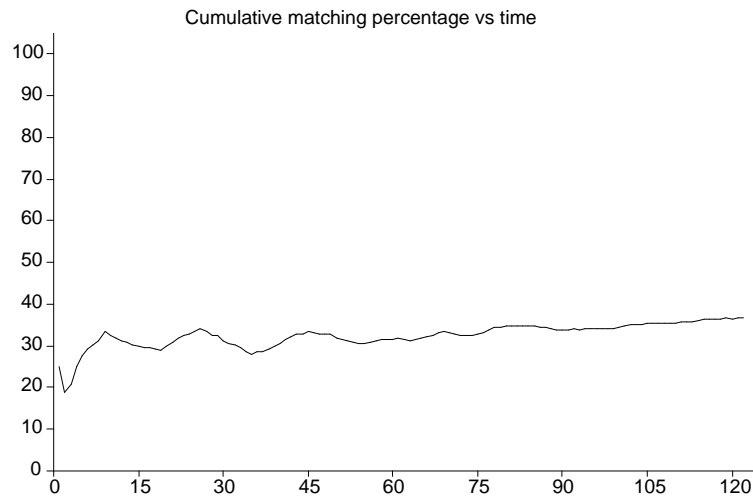
Figure 3.15: Number of validated returns for each time step.

Figure 3.16: Cumulative matching percentage *vs* time.
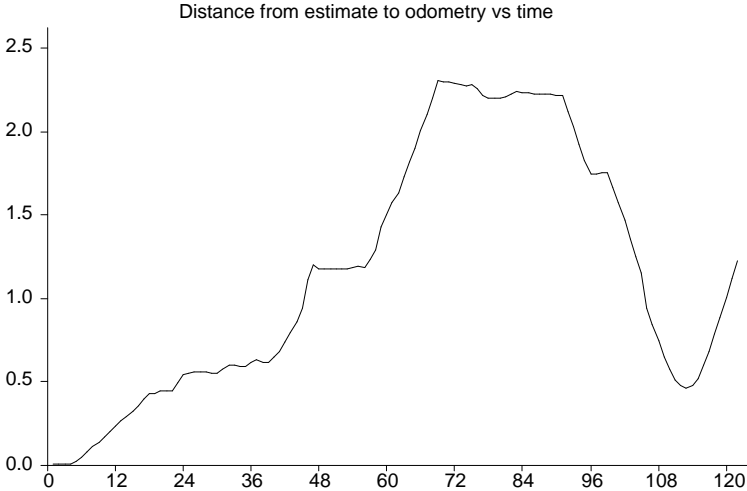
Distance from estimate to odometry vs time

Figure 3.17: A plot of the distance from the estimated to the odometric position *vs* time. Note how the distance decreases as the odometric position value comes back to cross the vehicle's actual path.

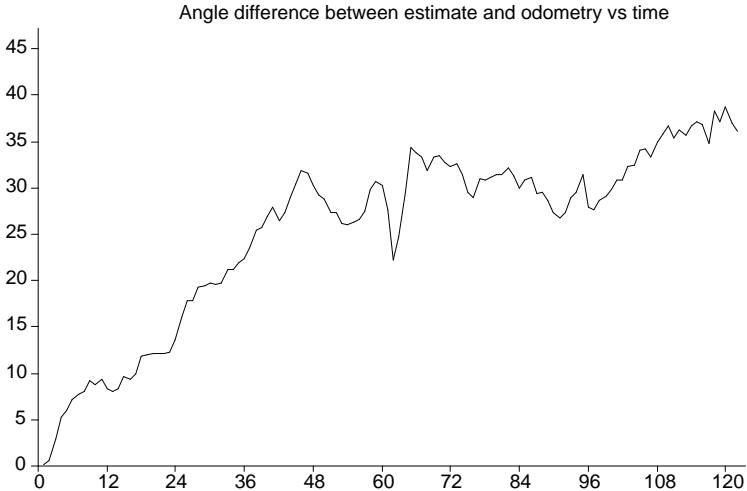Angle difference between estimate and odometry vs time

Figure 3.18: A plot of the difference in angle between the filter's estimated orientation and the orientation estimate of odometry.

previously had not even noticed were in the room.

True vehicle positions were not recorded for this experiment, but the consistently high matching percentage throughout the run indicates the success of the algorithm. Odometry's performance was surprisingly poor, as shown by Figures 3.17 and 3.18, which show the distance and orientation difference between the filter estimate and the vehicle's internal dead-reckoning computation. At the end of the run odometry's position estimate was off by about 35 degrees and 1.3 meters. We believe the dominant reason for this is the design of the vehicle itself. The Robuter vehicle has two driven rear wheels and two passive front casters. The front casters yield unpredictable and sometimes drastic "shopping trolley" effects whenever the vehicle changes direction. These effects introduce large errors into the orientation estimate of dead-reckoning, as clearly shown in Figure 3.13.

3.8.2   Implementation SKIDS-2

The ideal means of evaluating a navigation system is to have accurate measurements of the true vehicle position for comparison with the filter estimate. For "stop, look, move" runs, measuring the vehicle position does not change the character of the experiment, but can be a tedious procedure. For "on-the-fly" runs, obtaining a current and accurate estimate of the moving vehicle's position is a serious challenge to experimental work in navigation. Eventually, we hope to use an artificial beacon system, such as the GEC bar-code system installed for the Oxford AGV [17], as a data-logging device to permit experimentation of this nature with moving vehicles. However, such facilities were not available for our research with the Robuter vehicle.

The SKIDS room is about 12 meters long by 5 meters wide, and is comprised primarily of vertical office partitions and smooth walls. The SKIDS room has a 30 centimeter grid tile floor that facilitates measuring the position of the vehicle when stationary. Positioning markers were fitted to either side of the vehicle, and their (x, y) positions were recorded, allowing computation of the vehicle position and heading after the experiment. This run uses the same version of code as for implementation OxRob-1, with the same values for the EKF parameters g, $\mathbf{Q}(k)$, and $R(k)$. Figures 3.19 and 3.20 show estimated and true (hand-measured) positions for various stages of the run, along with magnified 10σ error ellipses (major and minor ellipse axes each magnified by 10). For this run, each leg of the vehicle's path was again specified by an operator at the SUN workstation.

In this run, the vehicle was led astray by a false match to the top wall at time step 10, as shown in Figure 3.19. Figure 3.21 gives a description of what happened. At this time step, the true vehicle orientation is nearly 180 degrees, and a three-dimensional target, caused by a row of electrical outlets along the floor, came into view. This target is not accommodated for in the 2-D map, and hence the filter matched the return to the top wall.

time 1

time 10

time 16
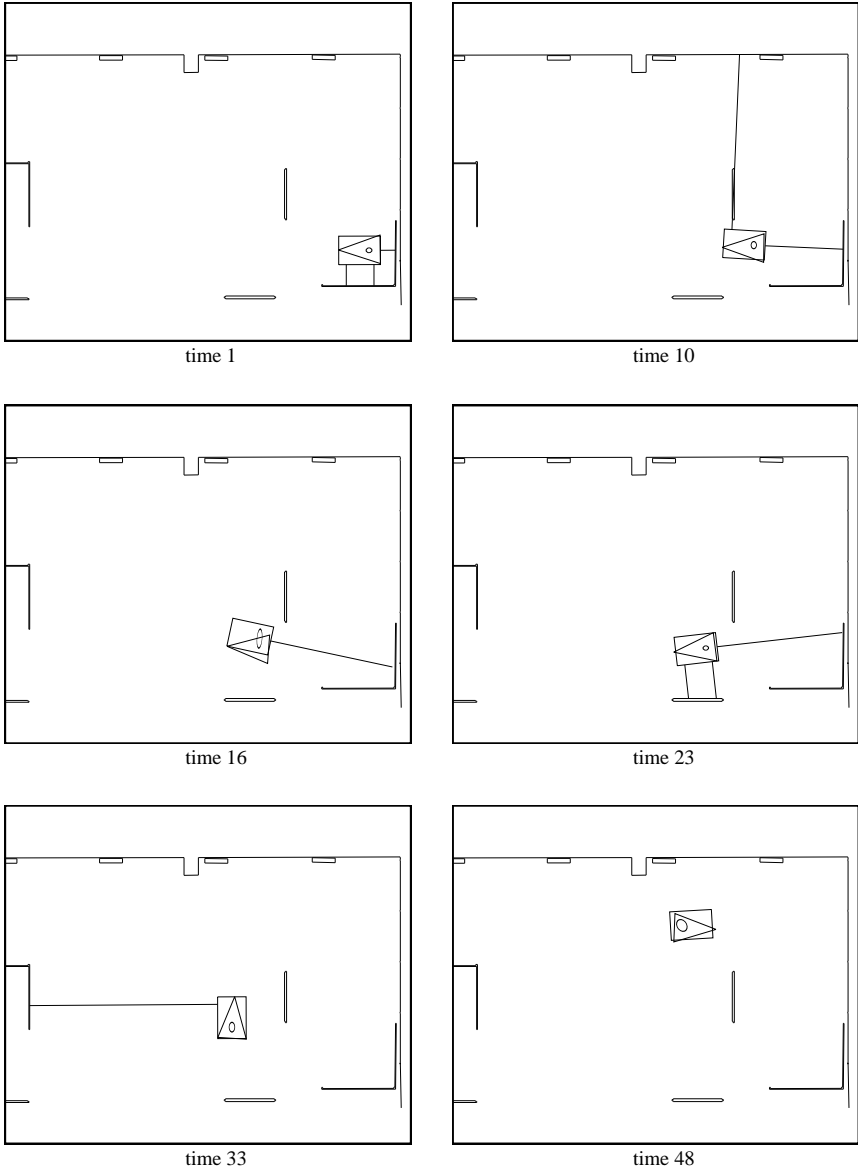
time 23

time 33

time 48

Figure 3.19: Various stages of Localization run SKIDS-2. The validated sonar range measurements used to update the vehicle position at various stages in the run are shown. The triangle shows the current true vehicle position, as measured by hand. The rectangle shows the *a posteriori* position estimate produced by the algorithm.
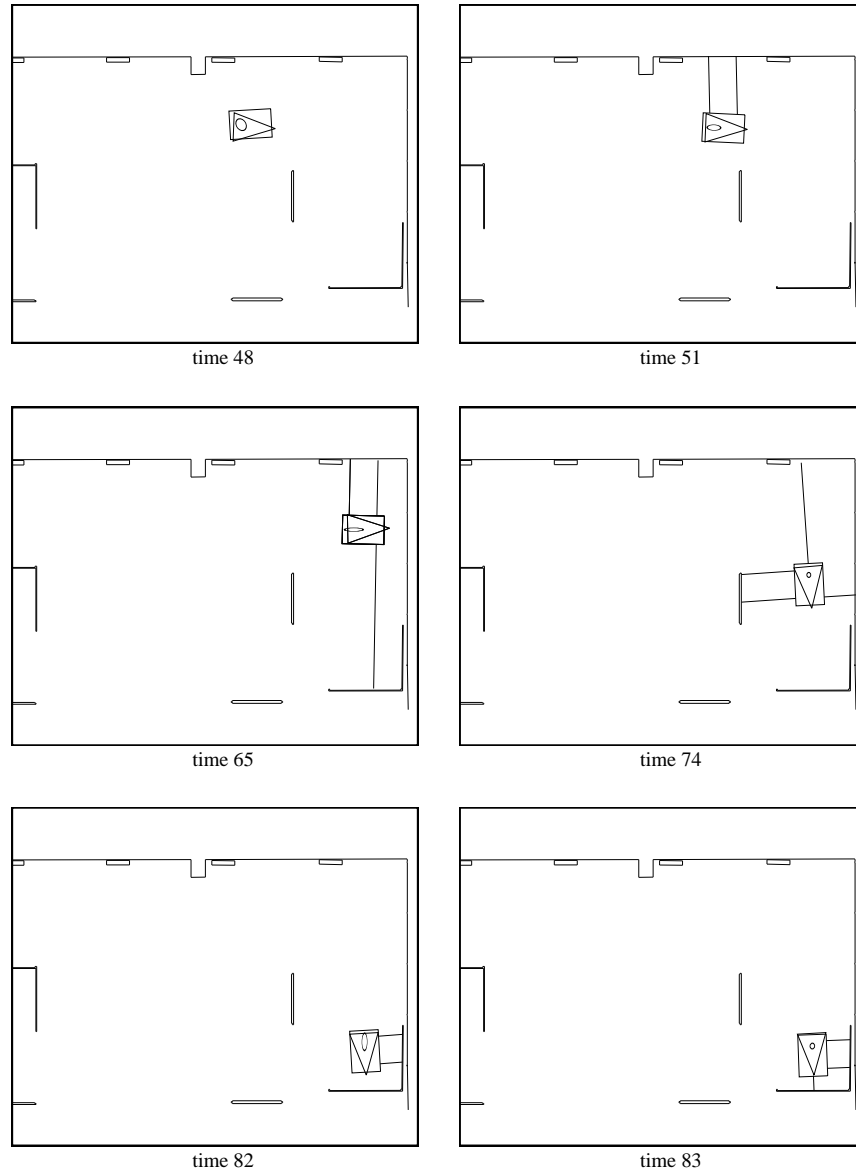
Figure 3.20: Various stages of Localization run SKIDS-2, continued. The validated sonar range measurements used to update the vehicle position at various stages in the run are shown. The triangle shows the current true vehicle position, as measured by hand. The rectangle shows the *a posteriori* position estimate produced by the algorithm.
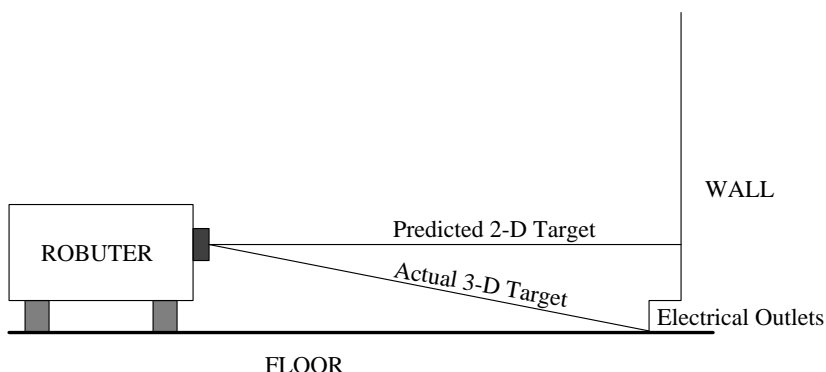
Figure 3.21: What happened at time step 10? Because the true vehicle orientation was nearly 180 degrees, a strong three-dimensional target, caused by electrical outlets along the floor, came into view. These effects are much more likely to happen at longer ranges (4.3 meters in this case). Note: this figure is not drawn to scale.

This false match has the effect of "pulling" the front right of the vehicle towards the top wall, introducing an error in orientation. After this time step, the vehicle steadily veered off course, and relocation was invoked (by the operator) at time step 17, using the office partition below the vehicle (horizontal in the 2-D map) and the wall to the right (vertical in the 2-D map).

Figure 3.22 shows the distance between true and estimated positions, and the distance between true and odometric positions. Figure 3.23 shows the difference in angle between true and estimated orientations, and true and odometric orientations. Examination of these graphs shows that after time step 10 until time step 17, when relocation was invoked, the filter estimate is worse than odometry. Figure 3.24 shows the number of validated returns for each step of the room. Approximately 30 percent of the 498 returns taken during the run were validated, for an average of just under 2 matched returns per time step. Figure 3.25 shows the cumulative matching percentage *vs* time.

3.9 Hands-off Localization Results

More recently, a simple trajectory planner has been added to the system to provide the capability for truly hands-off experiments. A sequence of goal positions (typically about five) are supplied. The vehicle is moved close to the home position and then a single button marked "Go" on the workstation is pressed to send the robot on a path from one goal location to the next
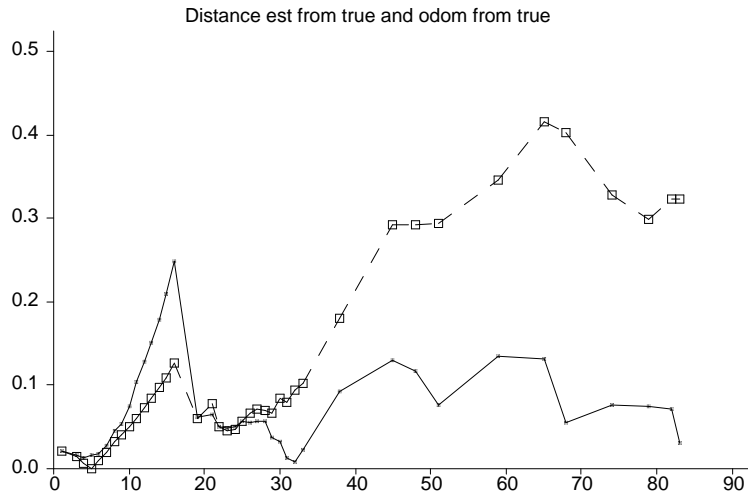
Figure 3.22: A plot of the distance from the estimated position to the true position (solid line) and the distance from the odometric position to the true position (dashed line), specified in meters.
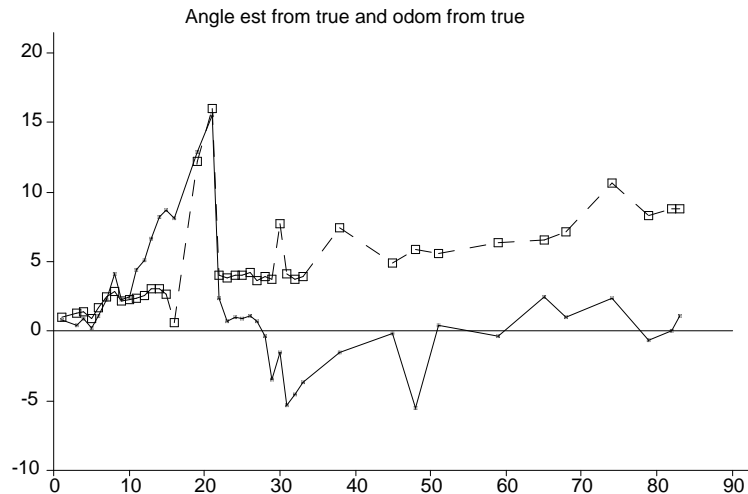

Figure 3.23: A plot of the difference in angle between the estimated and true orientations (solid line), and the odometric and true orientations (dashed line), specified in degrees.
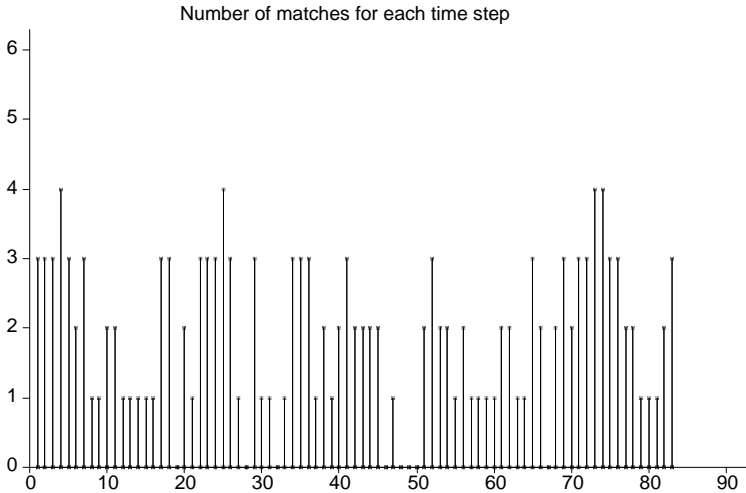
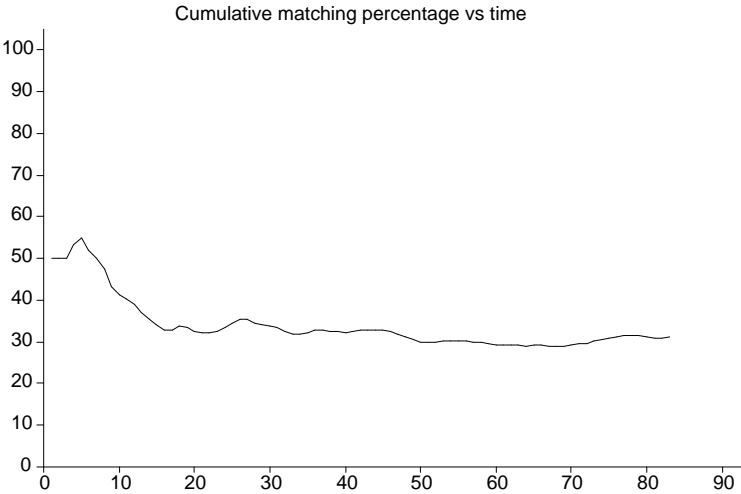Figure 3.24: Number of validated returns for each time step.

Figure 3.25: Cumulative matching percentage *vs* time.

until either 1) the collision test described above decides the vehicle is too close to some target in the map, 2) the vehicle's safety bumpers detect an actual collision, 3) the batteries fail or a robot cpu reset occurs for some other reason, or 4) the operator decides to abort the experiment.

The trajectory planner operates by finding three paths: 1) a direct, straight-line path from the current position to the goal, 2) an indirect, two-line path via the point with the same x coordinate as the goal, and 3) an indirect path via the point with the same y coordinate as the goal. Each of these paths in then tested to see if the vehicle comes within 10 centimeters of the map boundary at any position on the route. After a trajectory is executed, the distance between estimated and goal vehicle positions is compared with a "parking tolerance", chosen to be 10 centimeters and 6 degrees for these experiments. If the parking tolerance is met, a path is generated to the next goal; otherwise, a new path is planned for the current goal.

This control strategy was sufficient to achieve hands-off localization sequences at least thirty minutes in duration on at least twenty different occasions in the Oxford and SKIDS laboratories in May and June of 1991. All of the parameter values used were the same as in the experiments above. The typical time to failure was about one hour, with the longest run in each laboratory exceeding three hours. Some experiments were stopped before a failure because someone else needed the vehicle. Others were aborted when the radio RS 232 link to the vehicle caused mysterious cpu resets. About a third of the experiments terminated because the vehicle either physically collided with a wall or the collision test routine determined the vehicle was too close to a wall.

Figures 3.26 and 3.27 show odometric and estimated positions for a one-hour run (OxRob-2) in the Oxford AGV lab that was aborted because the vehicle physically crashed into the bottom wall. Figures 3.28 and 3.29 show another approximately one-hour run (OxRob-3) taken immediately after OxRob-2 was finished. OxRob-3 was aborted due to the failure of the vehicle's batteries.[3] Figures 3.30 through 3.33 show two runs in the SKIDS laboratory. The first of these (SKIDS-3) ended with the vehicle safely in track, while the latter experiment (SKIDS-4) was aborted by the operator due to impending collision with the bottom wall—the estimated vehicle position was off by about 25 centimeters. Table 3.1 shows the total number of EKF cycles, the total number of returns acquired, and the cumulative matching percentage for each of these four runs.

---

[3]Unfortunately, the battery charger for the Oxford Robuter exploded the week before this experiment was performed. This occurrence, coupled with the pending expiration of the first author's final visa extension from the Home Office, thwarted our plans for a systematic mean-time-to-failure analysis. In defense of the incompleteness of this experimental record, one can argue that extensive experimental verification is best postponed until model-based localization has been demonstrated with autonomously learned maps.

Figure 3.26: Odometric position *vs* time for run OxRob-2.

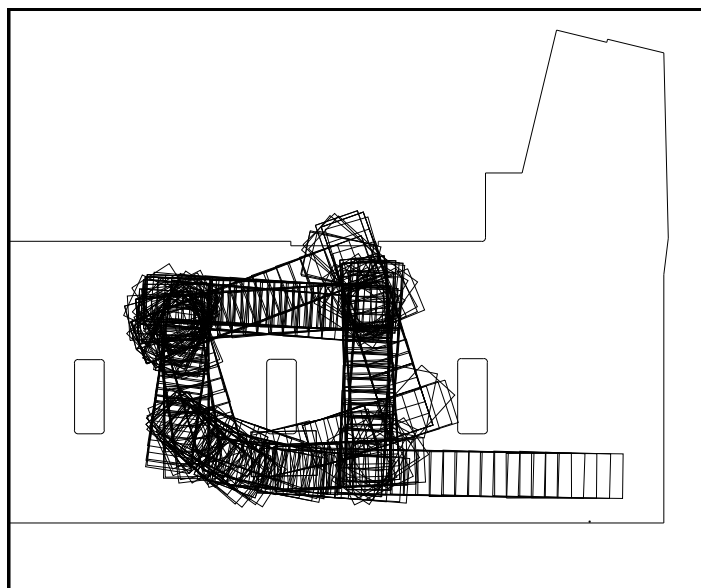Figure 3.27: Estimated position *vs* time for run OxRob-2.

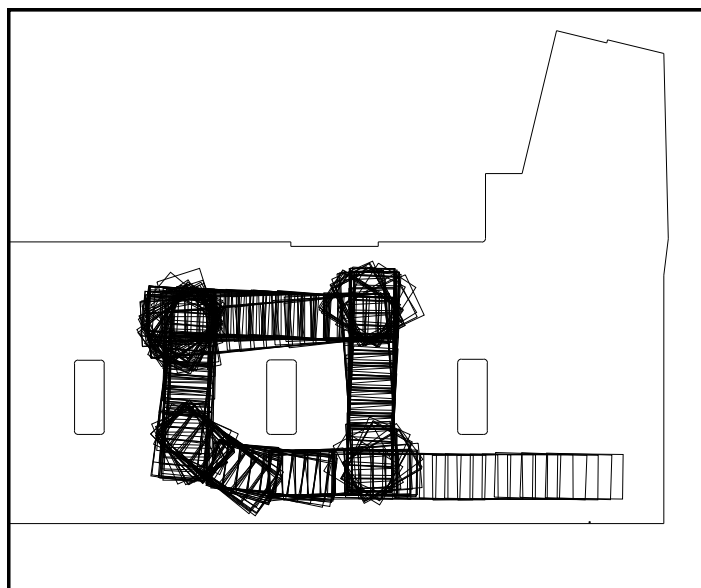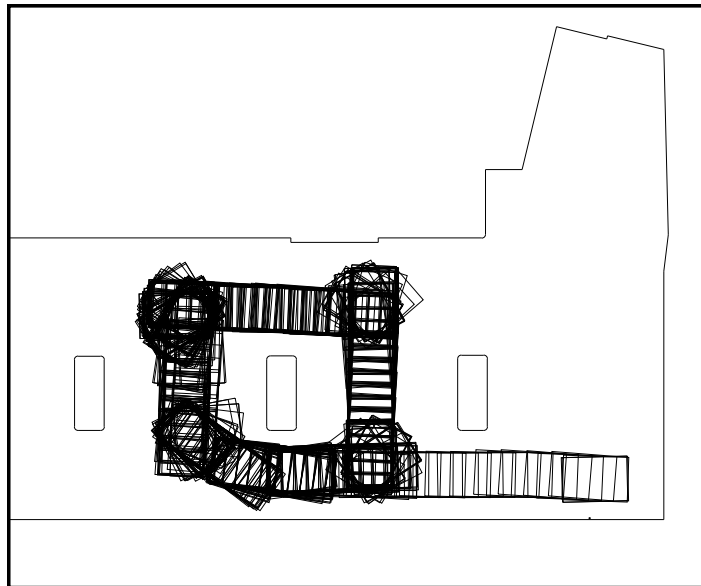Figure 3.28: Odometric position *vs* time for run OxRob-3.

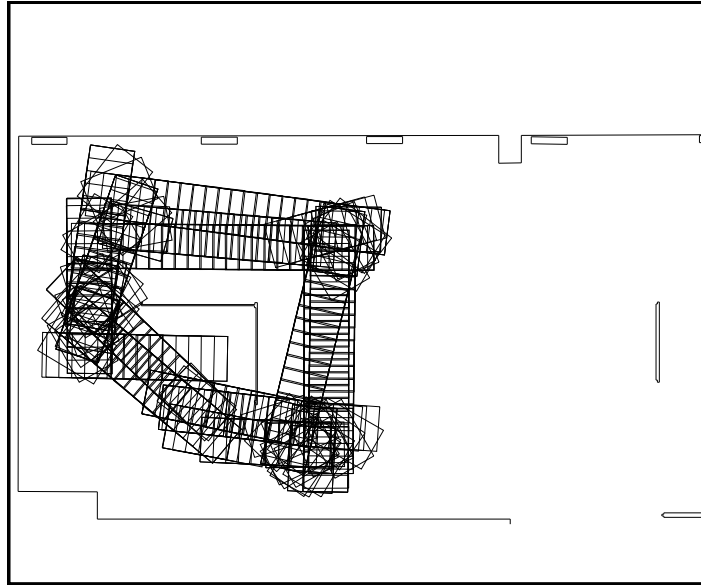Figure 3.29: Estimated position *vs* time for run OxRob-3.

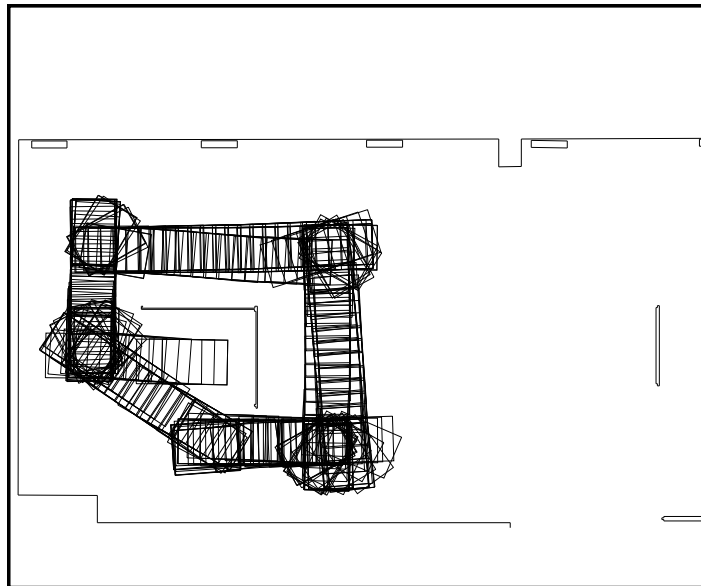Figure 3.30: Odometric position *vs* time for run SKIDS-3.

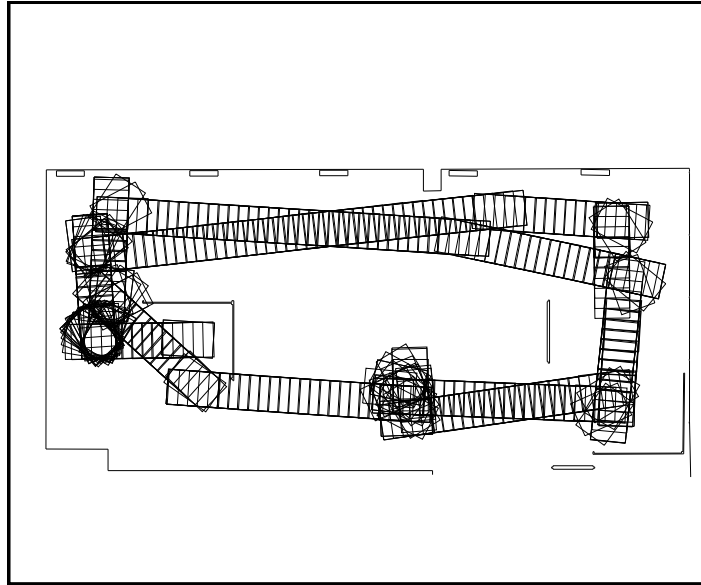Figure 3.31: Estimated position *vs* time for run SKIDS-3.

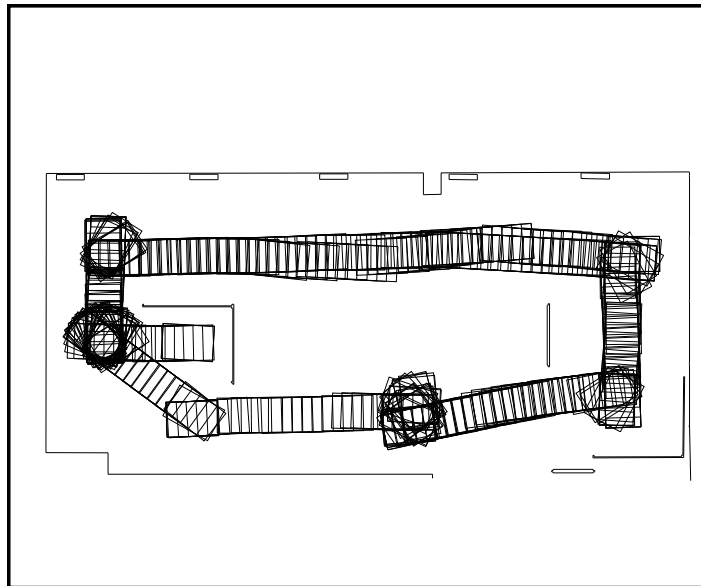Figure 3.32: Odometric position *vs* time for run SKIDS-4.

Figure 3.33: Estimated position *vs* time for run SKIDS-4.

Run	Date	EKF cycles	Returns	Matches	Percentage
OxRob-2	June 11, 1991	457	3656	1428	39.1%
OxRob-3	June 11, 1991	584	4672	1906	40.8%
SKIDS-3	June 5, 1991	303	1818	698	38.4%
SKIDS-4	June 6, 1991	354	2124	771	36.3%

Table 3.1: Cumulative matching percentages for hands-off runs.

3.10 Discussion

An important distinction between these implementations is that the painted brick surfaces of the Oxford AGV lab are considerably less favorable to our specular sonar model than the smooth surfaces of the SKIDS room. The algorithm succeeds nonetheless, because our sensor model can still adequately predict sonar returns at nearly orthogonal angles of incidence. For localization, it is not necessary to predict all sonar returns, just a sufficient subset of returns to achieve navigation. The concept of localization by tracking geometric beacons rests on the assertion that data corresponding to beacons will be easier to predict and explain.

The question "how can this algorithm be implemented most efficiently?" has not yet been extensively addressed. In particular, considerable time is spent in the prediction stage of the cycle. (This time was less than 1 second for the implementations.) The current map is an unordered list of targets. An efficient pre-sorting of the map would increase the speed with which prediction can be undertaken. A large body of previous work in computational geometry has addressed efficient visibility prediction for optical sensors. Acoustic visibility prediction presents an interesting topic of research, but we have not yet undertaken a rigorous pursuit of this subject.

As mentioned in Chapter 1, a primary concern when using the Kalman filter is to justify the choice of values for the parameters such as $\mathbf{Q}(k)$, $\mathbf{R}(k)$, and g. The parameters used here were arrived at through intuition and informal experimentation. Systematic experimentation to "balance" the filter has not yet been undertaken. Taking another look at Figures 3.22 and 3.23, when the vehicle is being successfully tracked the difference between true and estimated positions is still less that satisfactory. However, the repeatability of the experiments in different laboratories with different trajectories does demonstrate a certain level of parameter robustness.

3.11 Alternative Approaches

Our approach to using sonar for localization is different from commonly advocated approaches which compute position by first building up a local representation, either line segment-based[37], or grid-based[45], and then

matching this with a global model or grid. We believe that faster, more reliable performance can be obtained with our approach for two reasons:

- With local-to-global correlation methods, the benefits of achieving correspondence are quickly discarded. Subsequent position estimation requires repetition of the whole process. The time-consuming part of position estimation is achieving correspondence. For the fastest operation possible, the correspondence problem must be kept "out of the loop".

- Because of the presence of a considerable number of outliers in both the local and global representations, the accuracy of the resulting position estimate is degraded. We believe that not all sonar data should be treated equally; some observations will be very reliable, others completely misleading. Because the straightforward use of a sonar sensor model can distinguish many erroneous measurements, position estimation should be undertaken with only "good" data. In contrast, grid-based position estimation by correlation treats all sonar data equally.

Instead of matching a local map to a global map, we associate individual sonar measurements directly with the global map. Just a few range measurements, correctly associated with the environment feature that produced them, can uniquely determine position, and even a single explained measurement provides a single degree of freedom position update. A similar distinction can be found in the literature on traditional navigation applications—we direct the reader to compare the original concept of terrain contour matching (TERCOM) navigation [66] with the terrain-aided navigation (SITAN) algorithm developed more recently at Sandia National Laboratories [68].

3.12   Summary

This chapter has presented a model-based approach to localization. In analogy with model-based vision, correspondence is achieved directly between observations and a geometric model. This should be contrasted to occupancy grid and local composite model-based approaches, which first build up an intermediate representation and then correlate this intermediate representation with a global grid or model to compute the position update. By avoiding the construction of intermediate representations, this algorithm lends itself to a continuous implementation, in which the correspondence problem can be taken "out of the loop" to allow high bandwidth position estimation. In contrast with motion estimation methods, position is estimated with respect to a global reference frame at each time step.

The implementation characterized by Figures 3.26 to 3.33 has several significant limitations:

- The maps were constructed by hand.

- Portions of the room too complex to model by hand were declared "out of bounds".

- Vehicle travel was "stop, look, move", rather than continuous, because of hardware timing uncertainties and workstation-vehicle communication delays.

Despite these limitations, these experiments demonstrate that reliable, long-term localization can be achieved without artificial beacons, using odometry and sonar.

Chapter 4

Map Building

We now turn our attention to the bottom-up interpretation of sonar data to build a map of the environment. This aspect of the overall problem is complementary to our discussion of localization in the preceding chapter. The experimental results presented here deal with the problem of learning with precise position estimates, a limited form of the general problem.

4.1   Introduction

Although the experimental results of the preceding chapter used hand-measured models provided *a priori* to the robot, we contend that sonar-based navigation is feasible only if environment maps can be autonomously learned and maintained. We have several reasons for this belief:

- Very few application environments are completely static, and hence any *a priori* map will soon be inaccurate.

- A sonar sensor "sees" a completely different world than we do, because of the physics of sonar described in Chapter 2. It is difficult to construct a hand-measured map of an environment and know which features will actually be visible to the sensor. The hand-measured maps used in this book were only constructed after acquiring and displaying a variety of data in each of the rooms.

- To predict sonar data in practice, maps need to contain all four targets types described in Chapter 2, and thus require a high level of detail. Previous work has used maps that contain only line segments, and only roughly approximate the room geometry [40].

Our goal for a bottom-up sonar interpretation algorithm is to produce accurate, detailed descriptions of typical indoor scenes, using standard Po-

laroid hardware. Of overriding concern is the utility of the map for position determination.

4.2   Specular Event Interpretation

For underwater sonar, Hallam introduced the concept of specular event analysis:

> Specular echoes, however, have a high intrinsic information content. In common with diffuse echoes, they permit the sonar equipment to determine the relative position of the echo source, but they also constrain the local structure of the object responsible for the echo (because of the alignment of the observer and the object necessary for the specular reflection to occur).... I suggest a new method that uses directly the information present in specular reflections and the history of the vehicle motion to classify the specular echo sources and infer the local structure of the objects bearing them. [62]

Hallam presents three-dimensional target models for six types of specular event sources, which we list in order of decreasing reflected signal strength: concave corners, concave linear sources, planar sources, cylindrical sources, spherical sources and convex linear and corner sources. He presents an algorithm for distinguishing between point and line sources based on the computation of the scatter matrix of target observations. This technique assumes full observability of the target position from a single measurement, a reasonable assumption for an underwater sonar with a beam width of a few degrees. For this reason, however, his technique would be unsuitable for wide beam airborne sonar devices such as the Polaroid ranging system.

Hallam's proposal has strongly influenced this research. However, our application presents very different conditions from the underwater environment. As we discuss elsewhere, the assumption of a two-dimensional world is largely sufficient in our application, and hence we can reduce Hallam's target model set to four target types: planes, concave corners, cylinders, and convex edges. However, the wide beam of airborne sonar makes target bearing unobservable. To implement Hallam's idea in practice, public enemy number one, the correspondence problem, must be overcome. Further, the errors that characterize Polaroid ranging system data, discussed in Chapter 2, must be identified and eliminated.

For sonar in air, Kuc and his associates have treated map building as a process of interpreting specular events. One system [8] uses a linear array of transducers to differentiate wall and corners. In our terminology, this is a process of track initiation. One very important benefit of using a rotatable array of transducers is that the process can be achieved without moving the vehicle on which the array is mounted. Another approach, which has been

described in an as yet unpublished report [15], uses sonar data acquired from a single mobile rotating scanner to differentiate corners, planes and (diffuse) edges. The differences between this formulation and our approach will be subsequently addressed.

Brown has examined the problem of extracting three-dimensional surface information for planar and curved surfaces using sonar [22], [23]. Surface tracking and object recognition have been demonstrated. In Brown's experiments, a robot manipulator-mounted transducer and carefully engineered signal processing electronics are used [98]. Thus errors due to sensor positioning and sonar hardware are minimized, and the primary source of error is variations in the speed of sound in air over the path length of the echo. An analysis of this source of error is presented. The errors of Brown's system are at least an order of magnitude smaller than the errors in our system for two reasons: 1) our standard Polaroid sonar electronics add significant errors and 2) vehicle position errors greatly change the character of the estimation problem from the manipulator-mounted sensor case.

4.3 Rules for RCD-based Sonar Interpretation

We propose a technique for the classification of unknown RCDs based on two simple rules, illustrated by Figures 4.1 and 4.2.

> **Rule 1:** *circle test.* RCDs which correspond to a plane (or cylinder) will all be tangent to the plane (or cylinder), while RCDs which correspond to a corner (or edge) will all intersect in a point, at the corner (or edge).

> **Rule 2:** *constraint angle test.* The bearing to a hypothesized target must lie within the constraint angles θ_u and θ_l of the RCD.

Each RCD defines a circle centered at the sensor location with radius equal to the range of the RCD. Testing data-to-data association hypotheses for multiple RCDs is a process of finding the common tangents and intersection points of the circles the RCDs define. Impossible alternatives are then ruled out using the constraint angles defined in Chapter 2. In this way, single element RCDs obtained by a sonar ring or a moving vehicle are treated identically to the densely sampled data obtained by scanning from a stationary position. The only difference is that the constraint angles θ_u and θ_l will be farther apart for RCDs with less local support, and thus more observations may be required to determine the correct hypothesis.

Before attempting any automatic interpretation of sonar data, we now show a sequence of pictures of real Polaroid data which illustrate the infor-
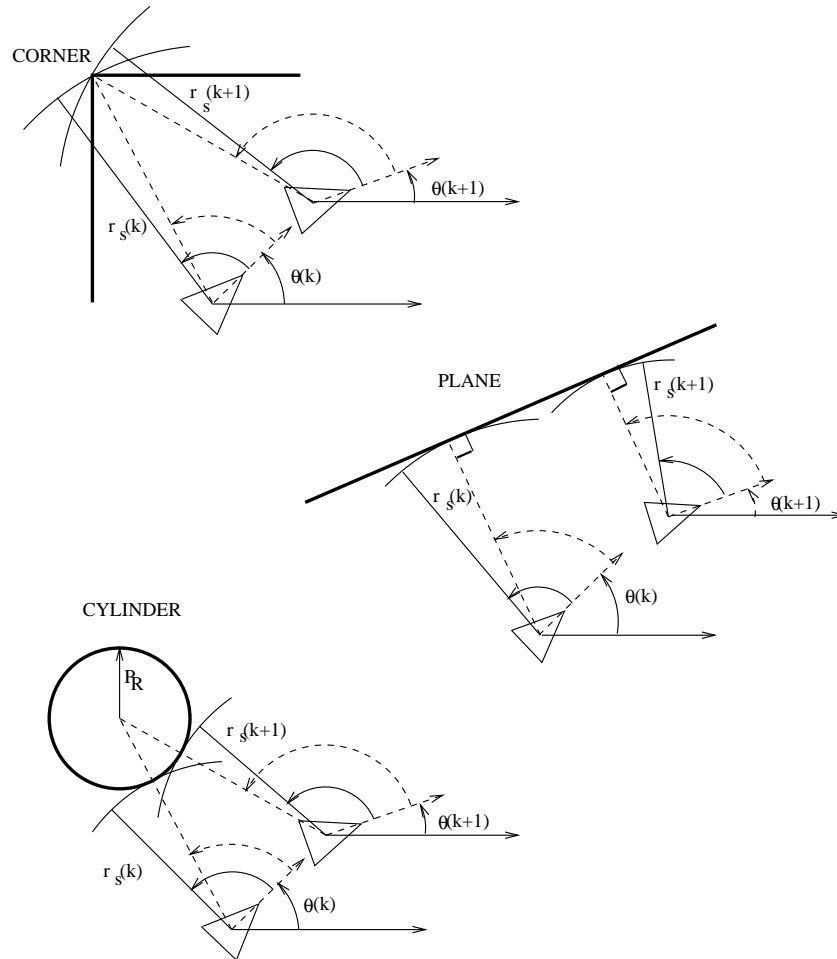
Figure 4.1: RCDs which correspond to a plane (or cylinder) will all be tangent to the plane (or cylinder). RCDs which correspond to a corner (or edge) will all intersect in a point, at the corner (or edge). RCDs for multiple reflections can be distinguished because they follow unpredictable trajectories as the vehicle moves.
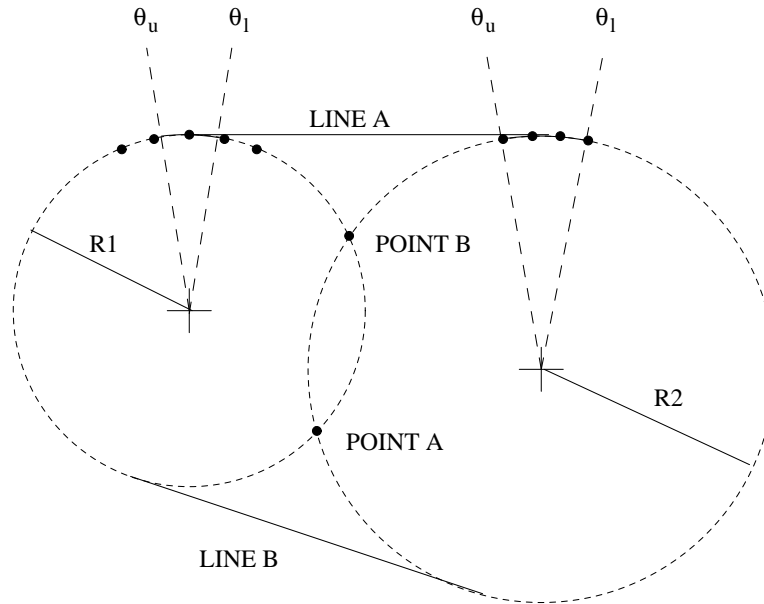
Figure 4.2: The constraint angles θ_u and θ_l are used to rule out incorrect hypotheses produced by the common tangent line and circle intersection routines.

mation content conveyed by RCDs, confirming the rules for RCD interpretation set out above. Figures 4.3 to 4.9 show the results of extracting RCDs from real sonar scans taken in several scenes. These events form the input to bottom-up interpretation. The figure captions provide more detail.

4.4 Map Building via Track Initiation

In the previous chapter, the special case where the map contains precisely known targets was considered. In general, the map is the set of target parameter vectors, their covariances, and target visibility angles:

$$M(k) = \{\hat{\mathbf{p}}_t(k), \mathbf{\Lambda}_t(k), \beta_t \mid 1 \leq t \leq n_T\}. \tag{4.1}$$

Map building involves two stages:

1. Classification of new tentative targets, via data-to-data association. This is a process of clustering together unexpected RCDs that yield a mutually consistent target interpretation.

2. Promotion of tentative targets to confirmed status. After classification, the objective is to match new observations to tentative targets
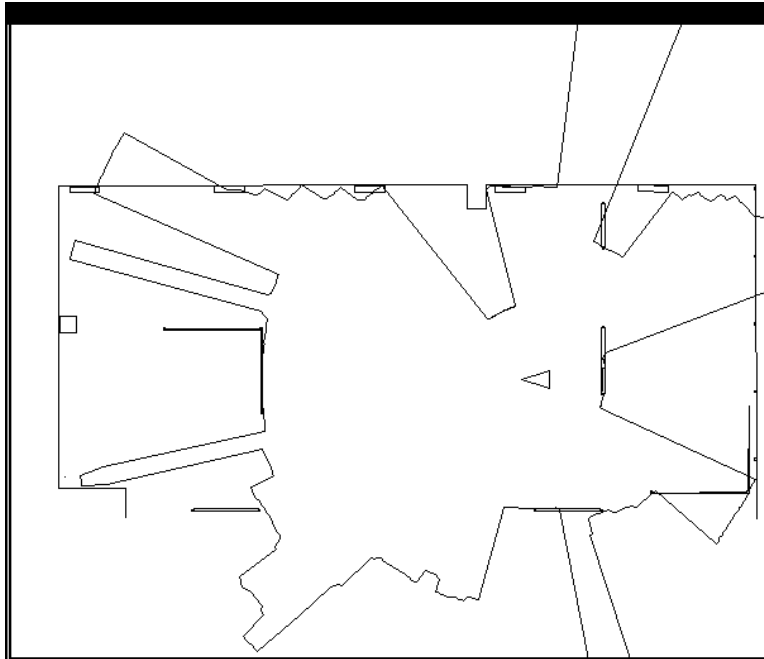
Figure 4.3: A sonar scan with 612 equally spaced returns taken with a single unknown object in the room. The room is 12 by 5 meters and the room model was measured by hand. What is the object? (Note the classic response of the wall directly in front of the vehicle, about 4 meters away.)
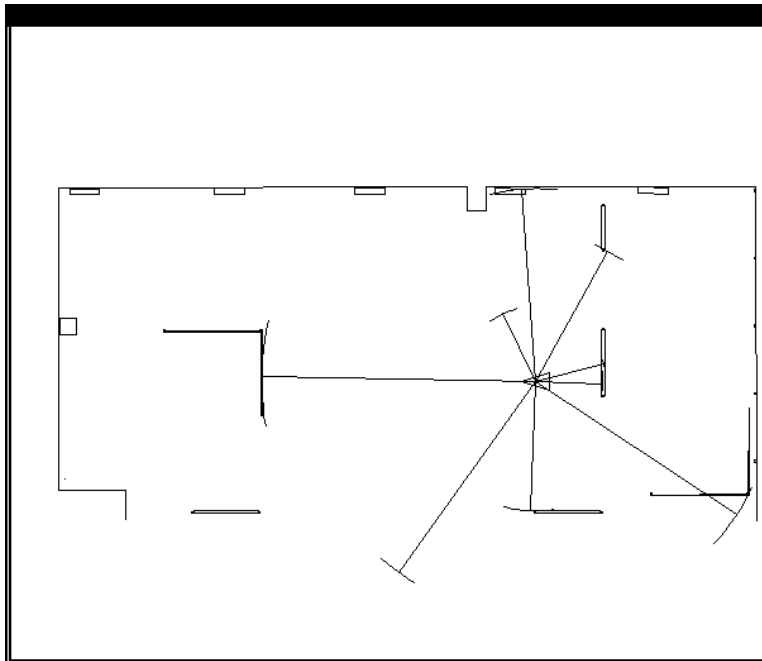
Figure 4.4: RCDs of width $\beta \geq 10$ degrees extracted from this scan. Using time-of-flight information only, corners, planes, cylinders and multiple reflections cannot be distinguished from a single scan. (Kuc and Siegel first reported this result for corners and planes.)
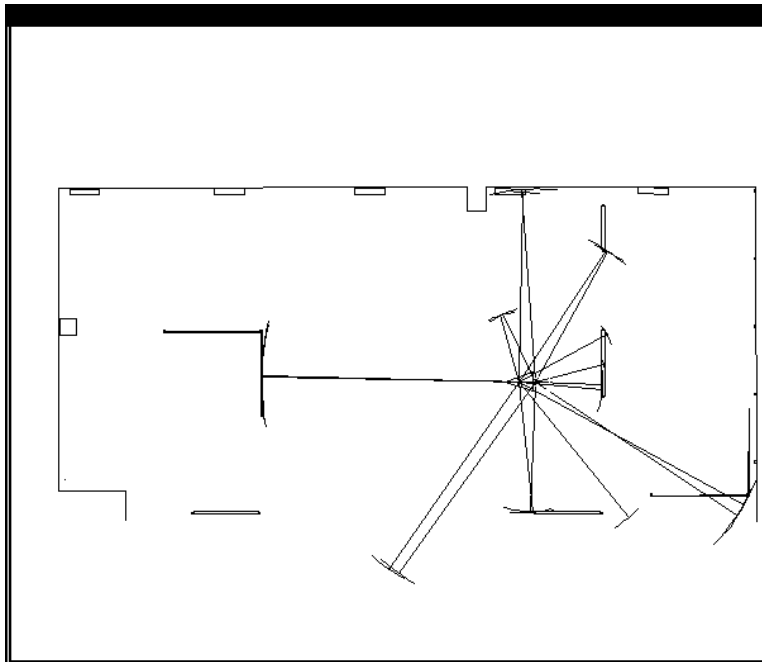
Figure 4.5: To determine the unknown geometry, we need to move the vehicle and take another scan. This picture shows the RCDs extracted from the first two scans of a motion sequence.
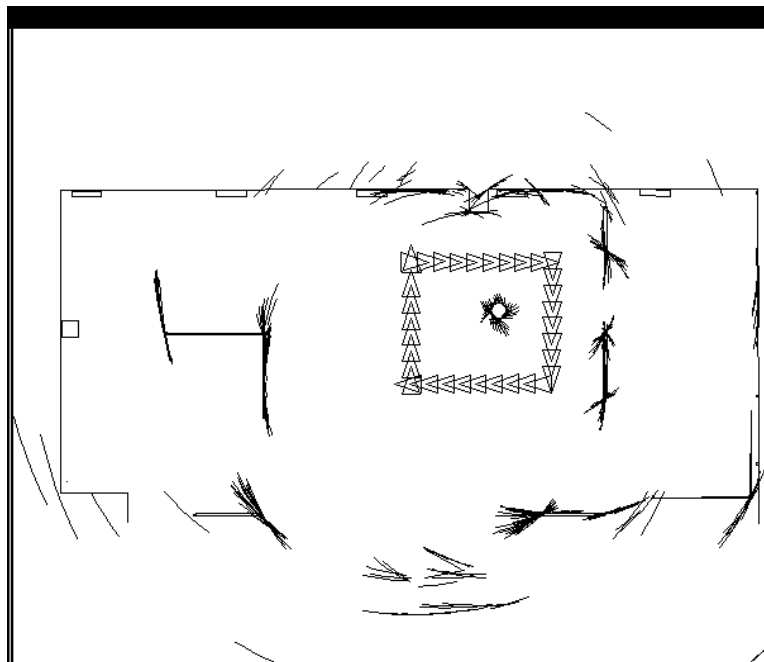
Figure 4.6: RCDs extracted from a complete motion sequence around the unknown object. The vehicle was manually controlled, and its position measured to within a few millimeters at each sensing location.
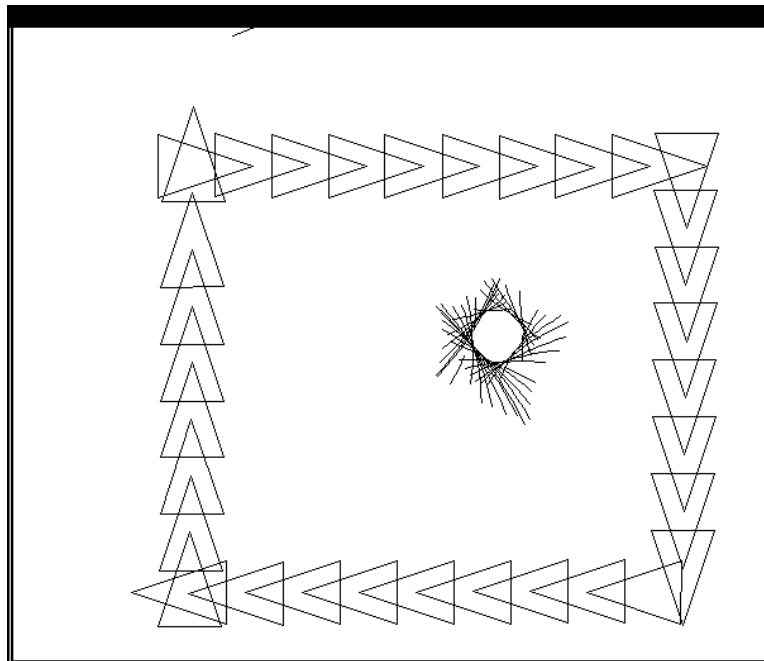
Figure 4.7: A close-up view of the RCDs extracted from a complete motion sequence around the unknown object, which is a cylinder (30 centimeter diameter trash can), yielding an RCD from each sensing location that is tangent to the circle defined by the projection of the cylinder onto 2-D.
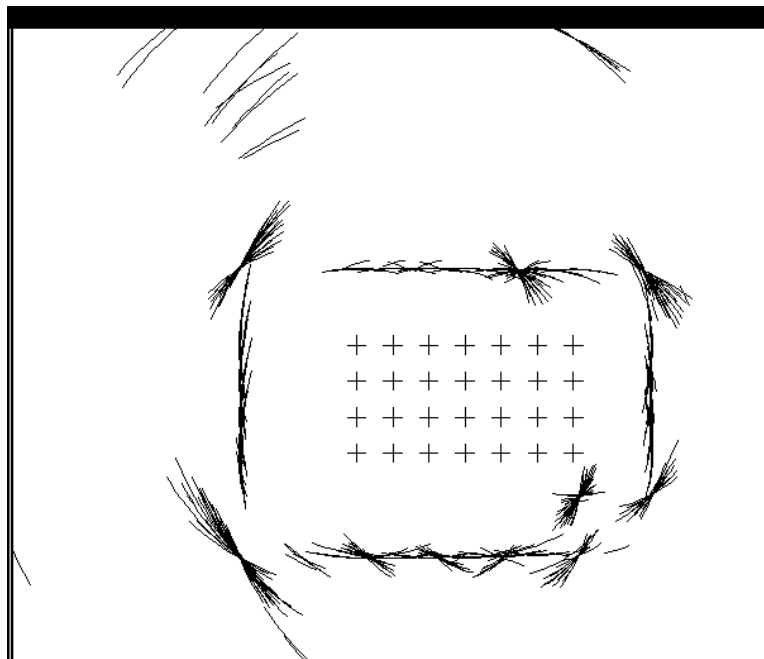
Figure 4.8: RCDs of width $\beta \geq 10$ degrees extracted from a grid of sonar scans. The room is approximately 3 meters long by 2 meters wide, and is described in more detail below. RCDs from different locations give support to each other, giving valuable information for learning the geometry of an unknown environment.
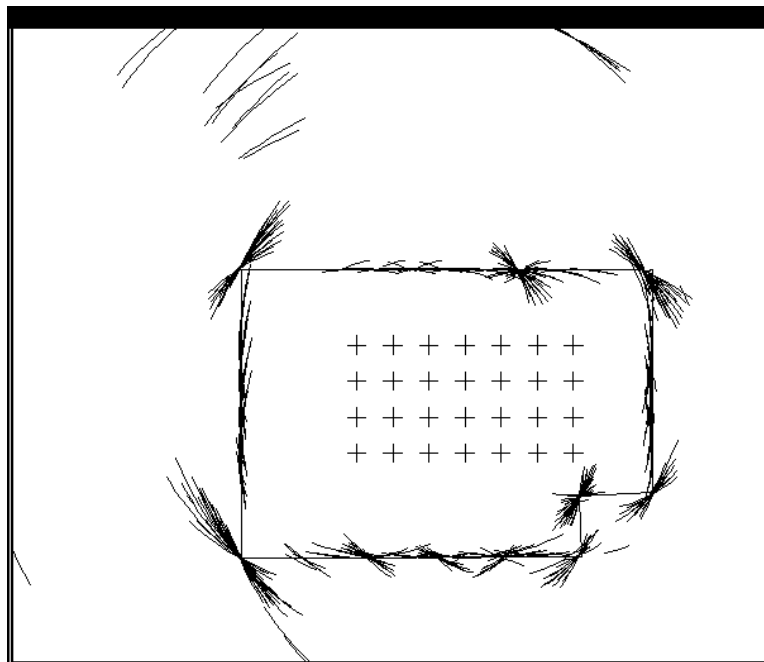
Figure 4.9: RCDs extracted from a grid of sonar scans superimposed on a hand-measured model of the room. 0th and 2nd order RCDs intersect at edges and corners. 1st order RCDs correspond to planes, represented in 2-D by a line that is tangent to all the RCDs that correspond to the plane. 3rd and higher order RCDs, caused by multiple specular reflections, do not give support to each other.

to compute increasingly accurate target estimates. Matches to tentative targets are obtained with the same data-to-target association mechanism used for localization, that is, by matching RCDs predicted from the target estimate $\hat{\mathbf{p}}_t(k)$ to observed RCDs.

When the RCDs which comprise an unknown cluster provide a single consistent target interpretation, we promote the unknown cluster to be a tentative target. The identity (point, line, arc of a given radius) of a tentative target is known. The initial values for the target state estimate $\hat{\mathbf{p}}_t(k)$ and covariance $\mathbf{\Lambda}_t(k)$ of a new target are computed using a batch least-squares computation, described below. After a sufficient number of additional sightings, tentative targets are promoted to be confirmed targets. Confidence in confirmed targets is sufficiently high that they are acceptable for use as beacons. The estimation focus is shifted from refining the target estimate to using the target estimate to compute vehicle position.

Unexpected RCDs are the rejects of localization—observed RCDs that do not match any tentative or confirmed target currently in the map. An unknown cluster is a set of one or more unexpected RCDs which yield mutually consistent interpretations for two or more target types. As discussed in Chapter 2, an isolated RCD is consistent with all possible interpretations. The goal of the algorithm is to elevate an unknown cluster to the status of tentative track, providing initial values for the target state estimate $\hat{\mathbf{p}}_t(k)$ and covariance $\mathbf{\Lambda}_t(k)$.

4.4.1 Matching Two Observed RCDs

The basic procedure used in the track initiation algorithm attempts to match two unexpected RCDs based on corner, plane and cylinder assumptions. Following rule 1, we use each RCD to define a circle, centered at the sensor location with radius equal to the range value of the RCD. Without loss of generality, we define a local coordinate system, centered at sensor location 1, with sensor location 2 on the positive x axis at the position $x = d$, where d is the separation between the two sensor locations, as shown in Figure 4.10. Let z_1 and z_2 be the range values of the two RCDs. The general problem is to find a third circle of known radius R which is tangent to the two circles the RCDs define. Then, the special case of line and point targets can be found by taking the limit as $R \to \infty$ and $R \to 0$, respectively.

Let ϕ_1 be the bearing to the hypothesized target from the origin of the local coordinate system. We can compute ϕ_1 using the law of cosines:

$$\cos(\phi_1) = \frac{(z_1 + R)^2 - (z_2 + R)^2 + d^2}{2\ d\ (z_1 + R)}. \tag{4.2}$$

Taking the limit of this equation as $R \to \infty$ yields the result reported by
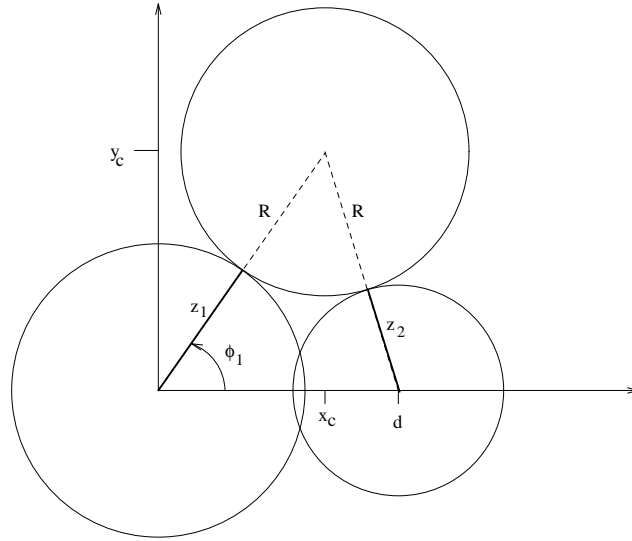
Figure 4.10: Matching two observed RCDs.

Brown for line targets [22]

$$\cos(\phi_1) = \frac{z_1 - z_2}{d} \tag{4.3}$$

while setting $R = 0$ yields

$$\cos(\phi_1) = \frac{z_1^2 - z_2^2 + d^2}{2\ d\ z_1} \tag{4.4}$$

for point targets.

For point and arc targets, the center of the circle is computed by:

$$x_c = (z_1 + R)\ \cos(\phi_1) \tag{4.5}$$

$$y_c = (z_1 + R)\ \sin(\phi_1). \tag{4.6}$$

For each value of R, each pair of circles can define zero, one, or two targets. The next stage after hypothesis generation is to rule out impossible targets using rule 2 above, the constraint angle test. A match is accepted if the bearing to the hypothesized target from each sensor location falls in the range of the constraint angles θ_u and θ_l for each RCD. Each pairwise match defines two *contact points*, which are the points on each RCD which are tangent to the hypothesized target. The contact points for a corner target coincide at the circle intersection point.

Classification of a cluster is decided by the percentage of RCDs in the cluster which match one another according to each hypothesis. In the implementation results which follow, the environments under consideration contain only line (plane) and point (corner and edge) targets. Clusters are considered unknown until they have at least three RCDs and the percentage of pairwise matches in the cluster that support the alternative hypothesis falls below 70 percent.

4.4.2 Initializing the Target Estimate

Once a classification decision has been made, the target estimate for a point target is taken as the average of all the contact points from each pairwise match. The target estimate for a newly classified line target is the best-fit (infinite) line through the contact points for all successful pairwise matches, computed using orthogonal regression. The endpoints of a line target are estimated by projecting new observations onto the infinite line.

4.5 Experimental Results

The above algorithm has been implemented for the case in which vehicle positions were measured by hand, accurate to within a few millimeters. We show results from the off-line processing of data taken in two different environments.

4.5.1 A Small Clutter-Free Room

The first run we show uses data acquired in a small room in Oxford, free of clutter, with plasterboard walls that are typical of most office environments. Figure 4.11 shows triangles at each vehicle location for 18 sonar scans and a hand-measured model of the room in which the scans were taken. The scans were processed off-line in a circular sequence, starting from the upper left of the figure. Figures 4.12 and 4.13 show the state of the learned map after just three time-steps, after which 2 line targets and 3 point targets have been classified. Each RCD in these figures is extended to a full circle to show the mechanism with which the algorithm functions, in accordance with rules 1 and 2 above. For vehicle travel in the direction of the hypothesized target, the circle intersection routine is very sensitive to small errors in sensor location. For this reason, when the direction of vehicle travel is within 15 degrees of the orientation of the two RCDs, matches are approved for both target types but no contact points are calculated. This has the effect of delaying the classification decision until RCDs from other sensing locations are obtained. This case is clearly shown for cluster 4 in Figure 4.13.
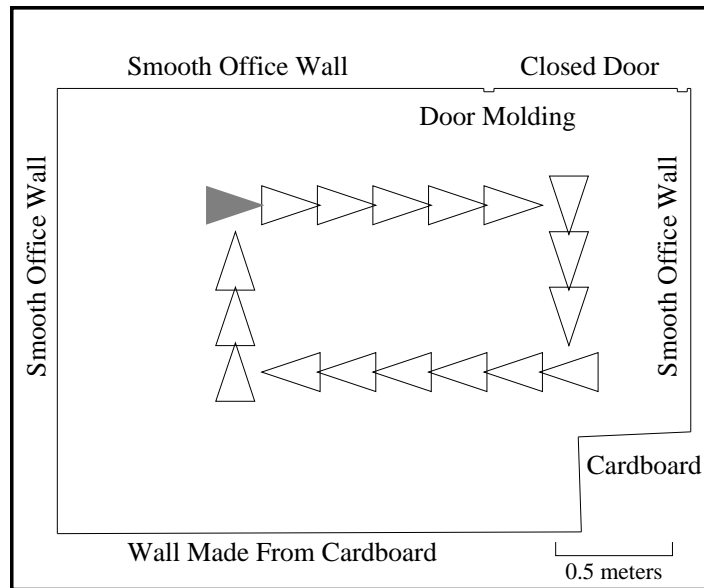
Figure 4.11: Hand-measured map of the room in which the grid of scans were taken. Eighteen scans of the grid were processed by the map building algorithm, starting at the shaded triangle in the upper left-hand corner of the room. The room is about 3 meters wide, with a closed door in the upper right-hand region of the picture. The bottom wall was assembled using sheets of cardboard.

cluster 1

cluster 2
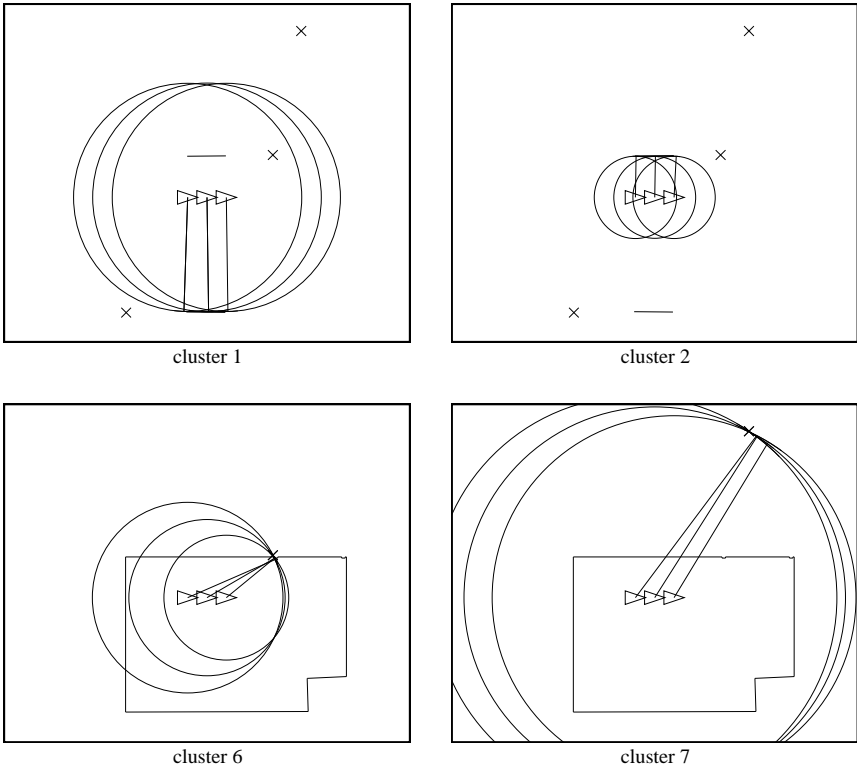
cluster 6

cluster 7

Figure 4.12: Classified clusters at time-step 3, illustrating the manner in which a circle is defined for each RCD. Clusters 1 and 2 have been classified as line targets because the constraint angle test has ruled out a point target interpretation. Similarly, clusters 6 and 7 have been classified as point targets.

Figure 4.13: Two clusters which are as yet unclassified at time-step 3, because both point and line target interpretations are still consistent.
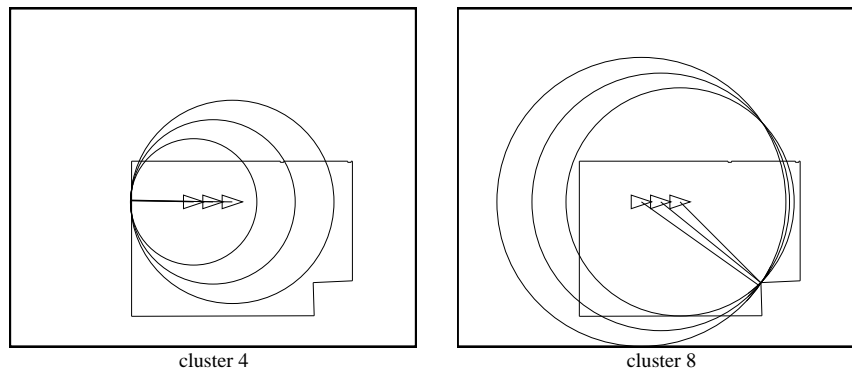
Cluster	Type	Estimated		Actual		Difference	
		R	θ	R	θ	R	θ
1	Line	0.507	89.8	0.50	90.0	0.007	-0.2
2	Line	1.405	-89.2	1.402	-89.8	0.003	0.6
3	Line	1.708	0.5	1.712	0.0	-0.004	0.5
4	Line	-1.009	-0.2	-1.000	0.0	-0.009	-0.2

Table 4.1: Comparison of learned line target parameters with actual values, hand-measured to a few millimeters of accuracy. Range is given in meters, orientation in degrees.

Figure 4.14 shows the map produced by the algorithm after the full 18 scan sequence has been processed. Magnified 8σ error ellipses are shown for point (corner and edge) targets. Target estimates are calculated using orthogonal regression on the set of contact points, as described above. Figure 4.15 shows this map superimposed on the room model, revealing the correspondence between the learned map and the actual room geometry. Figures 4.16 and 4.17 show most of the clusters used to make this map. Tables 4.1 and 4.2 show these results quantitatively. Hand-measured and learned values are usually within 1 centimeter of agreement. These results show the compression of 11,016 sonar returns into just 14 target estimates. This illustrates map building as a process of acquiring more and more range measurements to yield increasingly precise geometric descriptions.

4.5.2   A More Complex Environment

The results we now show are from the SKIDS room, as used for the localization experiments of the previous chapter. This environment is considerably

Figure 4.14: Map of the room produced by the algorithm. 8σ (major and minor ellipse axes multiplied by 8) error ellipses are shown for point (corner and edge) targets.

Figure 4.15: Learned map superimposed over room model.
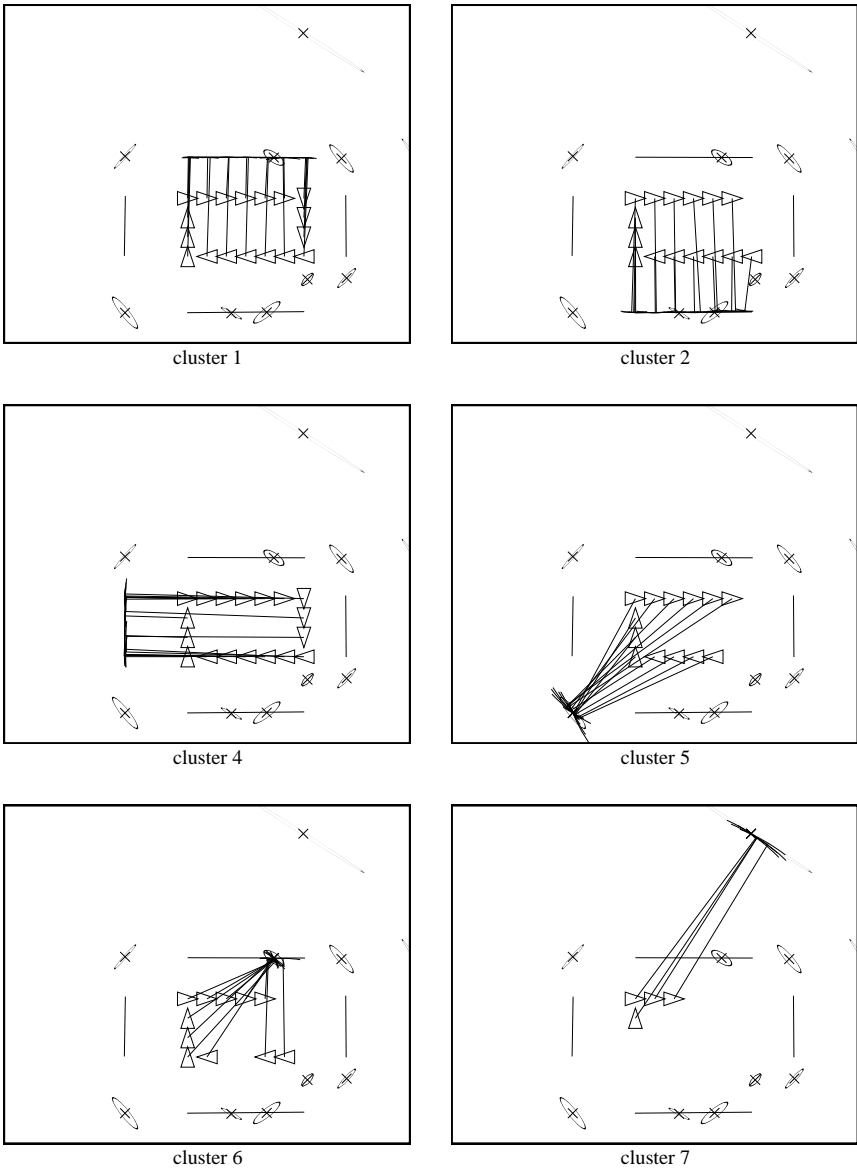
Figure 4.16: Point and line clusters obtained during the run. Cluster 7 is caused by multiple reflections off the top wall to a corner below.
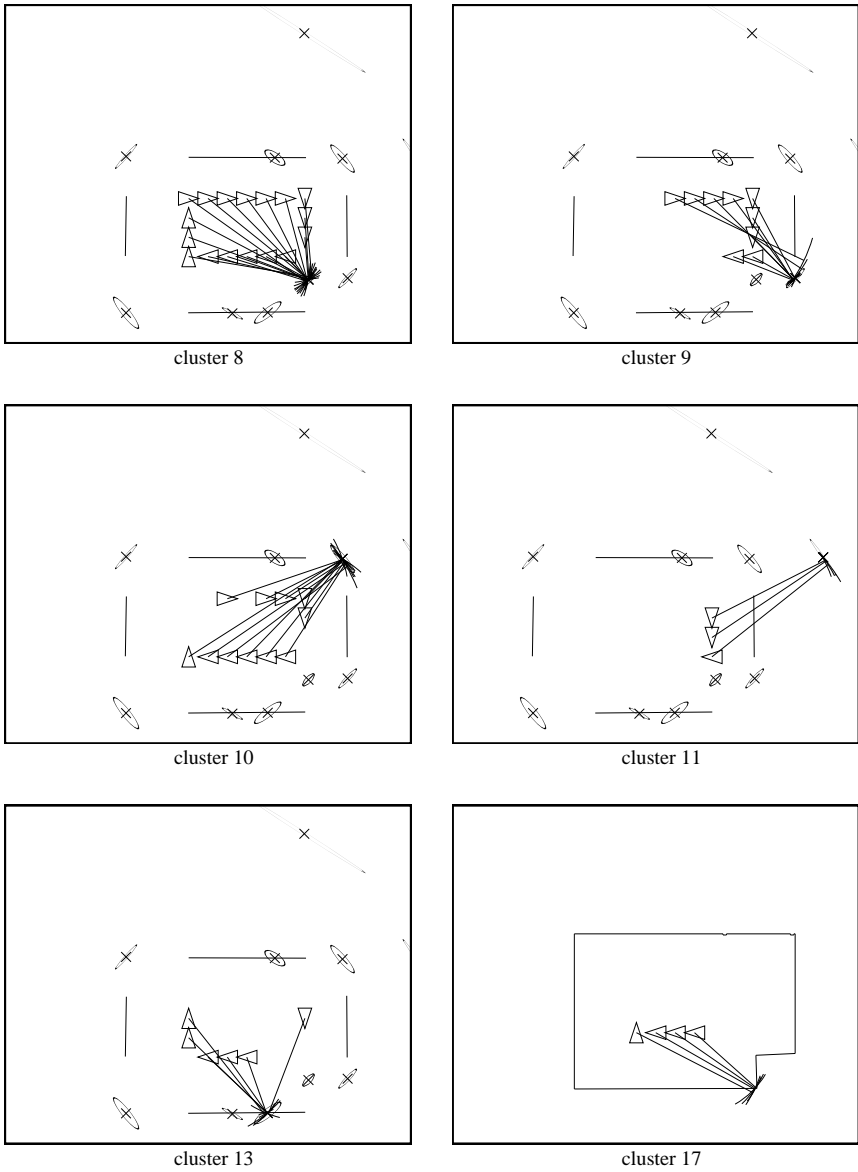
Figure 4.17: More clusters obtained during the run. Cluster 11 is a point target erroneously initiated by multiple reflections off the right wall and the door molding. Cluster 17 is still classified as unknown, despite the fact that it corresponds to a real corner in the lower right-hand part of the room.

Cluster	Type	Estimated		Actual		Difference	
		X	Y	X	Y	X	Y
5	Point	−1.007	−1.405	−1.000	−1.406	−0.007	0.001
6	Point	0.822	0.501	0.826	0.500	−0.004	0.001
7	MR	1.183	2.029	no target (multiple reflection)			
8	Point	1.234	−0.996	1.230	−0.992	0.004	−0.004
9	Point	1.715	−0.979	1.712	−0.970	0.003	−0.009
10	Point	1.652	0.490	1.654	0.500	−0.002	−0.010
11	MR	2.556	0.508	no target (multiple reflection)			
12	Point	0.299	−1.412	unrecorded			
13	Point	0.734	−1.401	unrecorded			
14	Point	−1.006	0.516	−1.000	0.500	−0.006	0.016

Table 4.2: Comparison of learned point target parameters with hand-measured values. Positions are given in meters, with respect to the 2nd vehicle position. Clusters 7 and 11 were tracks initiated for false multiple reflections. Clusters 12 and 13 correspond to edge targets formed by the pieces of cardboard used to construct the lower wall. The exact positions of these edges were not recorded when the experiment was performed.

more complex than the simple room above, but mostly planar, and hence the two-dimensional environment assumption is largely adequate. Figures 4.18 to 4.20 show the result of applying the same algorithm with the same parameters to a sequence of 53 scans taken from accurately measured positions. The scans do not sufficiently cover the room to construct a complete room map, but allow a tremendous amount of detail to be built up for the right-hand part of the room.

The results are not perfect, but show the potential of the approach. Two undesirable effects are visible in Figures 4.18 to 4.20:

- Distinct point targets of close proximity are merged together. This effect is visible in clusters 20 and 21, which correspond to the ends of two offices partitions. Each of these clusters actually corresponds to two distinct corners a few centimeters apart that are only visible from one side of the partition.

- The accidental alignment of RCDs from different targets occasionally hypothesize false line targets. One such line target is visible in the upper right-hand part of the room. Cluster 3 shows two erroneously merged clusters.

These 53 scans represent 32,436 sonar returns. We envisage the accurate mapping of typical indoor scenes, in terms of point, line and arc targets, using thousands and thousands of returns, to produce concise, precise 2-D maps.
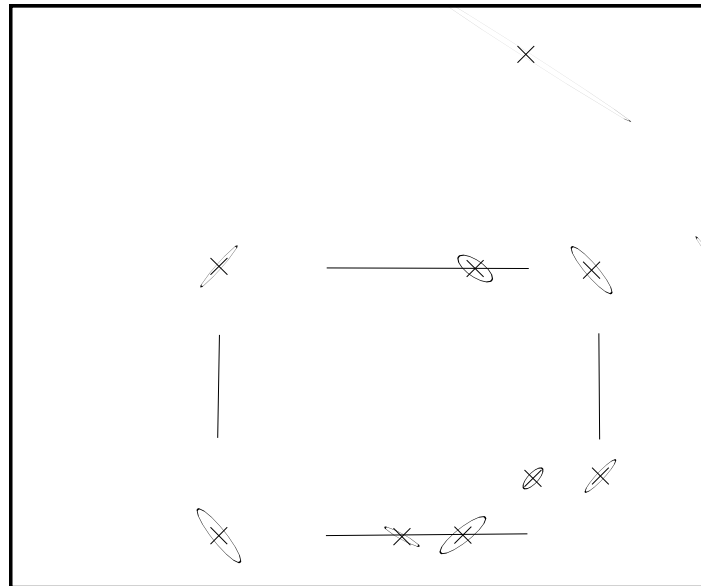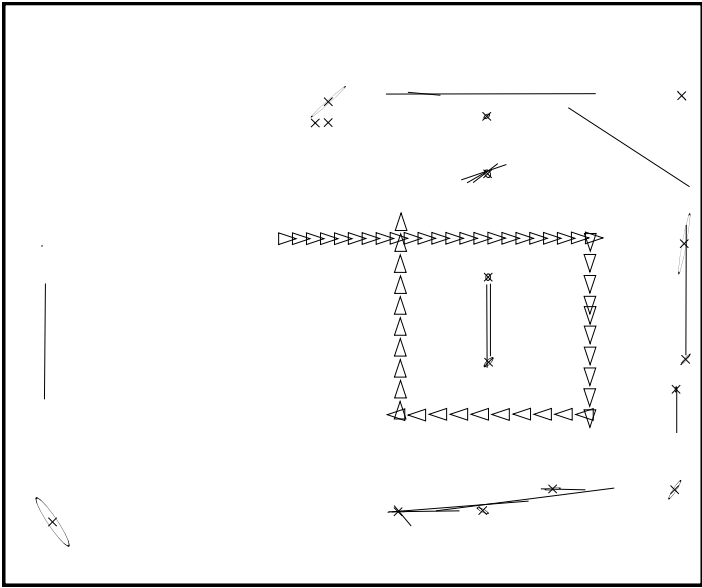
Figure 4.18: Map of the room produced by the algorithm. 2σ (major and minor ellipse axes multiplied by 2) error ellipses are shown for point (corner and edge) targets.
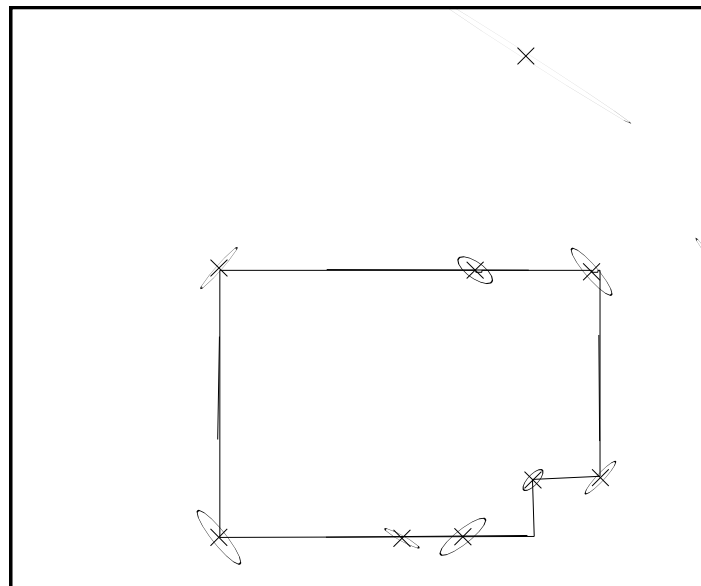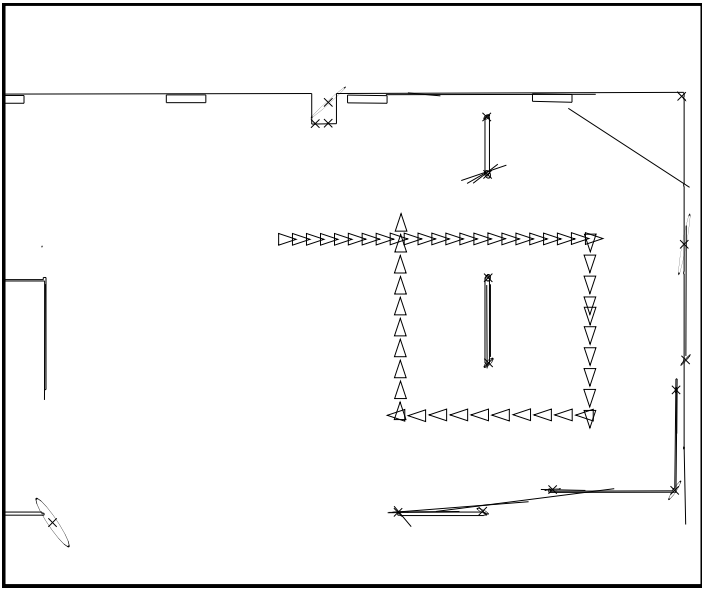

Figure 4.19: Learned map superimposed over room model. Triangles are shown for each location from which a scan was taken.

cluster 1

cluster 3

cluster 17

cluster 20

cluster 21

cluster 31

Figure 4.20: Point and line clusters obtained during the run. RCDs for two distinct line targets were erroneously merged together to form cluster 3.

Cluster	Estimated		Actual		Difference	
	R	θ	R	θ	R	θ
1	1.706	0.6	1.712	0.0	−0.006	0.6
2	1.408	−89.0	1.402	−89.8	0.006	0.8
3	0.506	−89.8	0.50	90.0	0.006	0.2
4	−1.0013	0.1	−1.000	0.0	−0.013	0.1

Table 4.3: Comparison of learned line target parameters with hand-measured values for the four scan map building run. Range is given in meters, orientation in degrees.
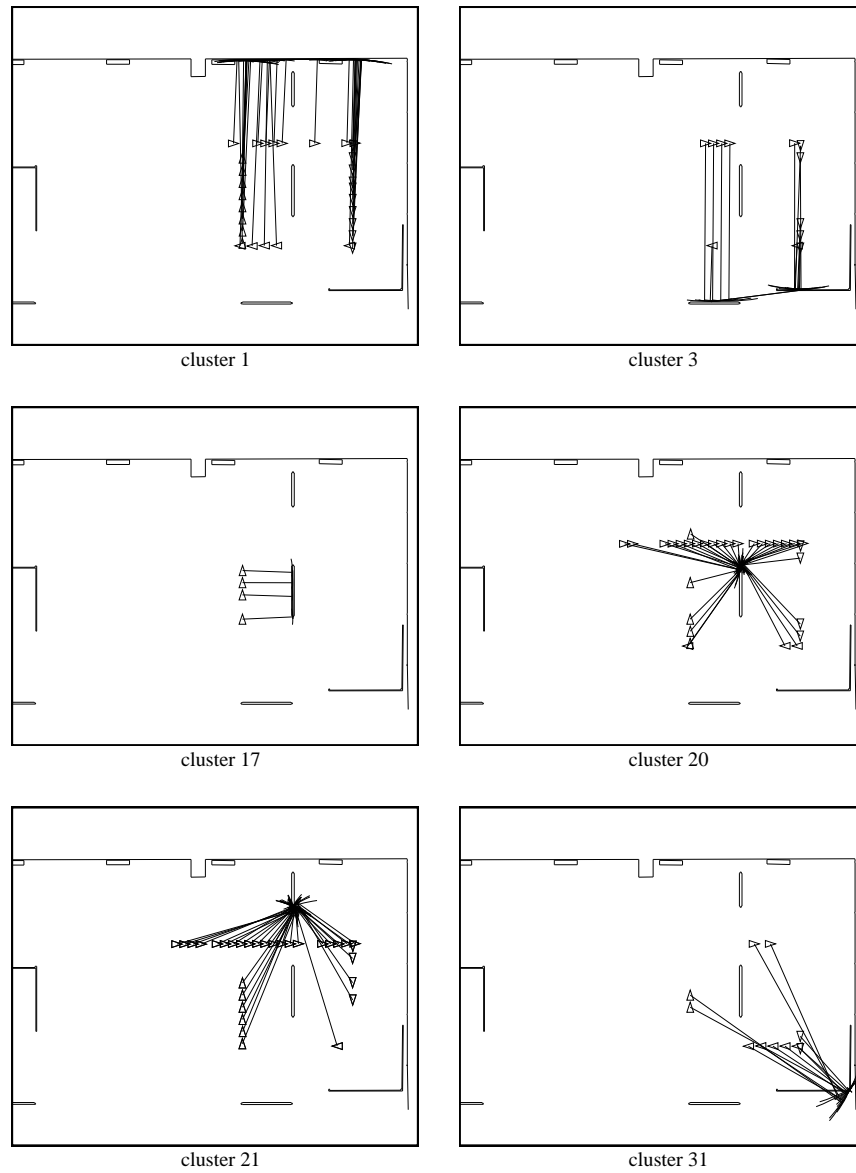
Cluster	Estimated		Actual		Difference	
	X	Y	X	Y	X	Y
5	−1.011	−1.396	−1.000	−1.406	−0.011	0.010
6	0.812	0.499	0.826	0.500	−0.014	−0.001
7	1.235	−0.994	1.230	−0.992	0.005	−0.002
8	1.651	0.490	1.654	0.500	−0.003	− 0.010
9	−1.016	0.500	−1.000	0.500	−0.016	0.000

Table 4.4: Comparison of learned point target parameters with hand-measured values for the four scan map building run. Positions are given in meters, with respect to the 2nd vehicle position.

4.5.3   Learning From Four Scans

Accurate mapping is possible, however, with a small number of returns. Remember, these scans were taken with the long-range sensing mode of the standard Polaroid ranging system, and hence a high scanning density is necessary to eliminate weak, erroneous returns. Also fewer scans can be sufficient for reliable target identification. Figures 4.21 and 4.22 show the re-application of the algorithm to the small clutter-free room above, using just four scans. Tables 4.3 and 4.4 show a comparison of the hand-measured and estimated maps for this run.

4.6   Multiple Hypothesis Track Initiation

In collaboration with Ingemar Cox of NEC Research Institute, we have had the opportunity to formulate the specular event interpretation process more rigorously using the multiple hypothesis tracking (MHT) filter [35]. The seminal work in the field of MHT is the work of Reid [111]. More recent research in this area has been presented by Mori [102], Kurien [84], Chong [30], and Chang [27].[1] The appeal of MHT techniques is that they

---

[1]Reference [102] offers the brave reader a particularly challenging experience, but is highly recommended.
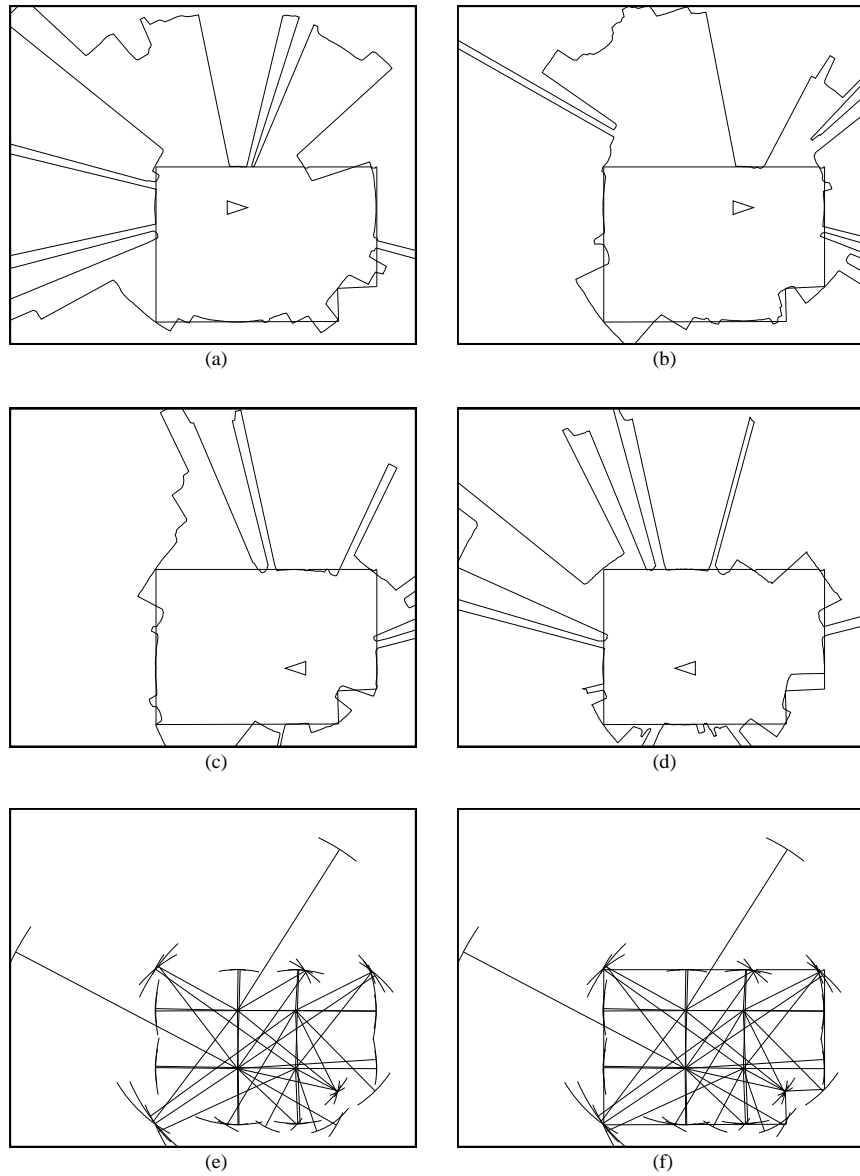
Figure 4.21: Learning with scans taken from four positions, part 1. (a) through (d) show the result of median-filtering the raw scans with a 5 point window. (e) shows the result of extracting RCDs. (f) shows these same RCDs superimposed on a hand-measured room model.
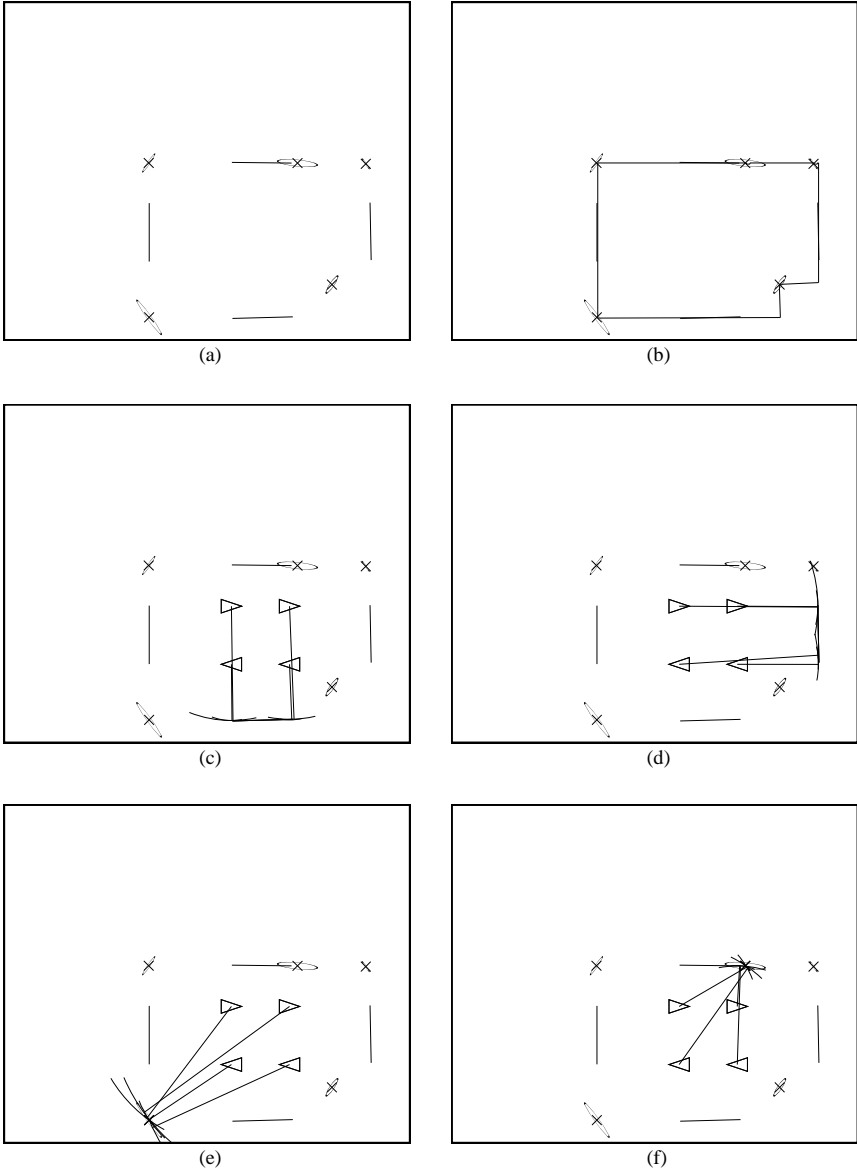
Figure 4.22: Learning with scans taken from four positions, part 2. (a) shows the learned map, with 8σ error ellipses for corner targets. (b) shows the map superimposed on the room model. (c) and (d) show clusters used to initialize planar targets. (e) and (f) show clusters used to initialize corner targets.

can refrain from making irreversible decisions too early in the interpretation process, as described by Chong *et al.*

> The multiple hypothesis approach makes 'soft' data association decisions when the available data are not of high enough quality. Instead of making firm commitments of associating measurements to tracks, it forms multiple data association hypotheses corresponding to alternate explanations of the origins of the measurements. These hypotheses are evaluated with respect to their probabilities of being true using the target and sensor models. [30]

This comment shows the natural suitability of MHT to bottom-up sonar interpretation, for an RCD cannot be classified if observed in isolation; the decision must be delayed until more data can be acquired. MHT techniques use Bayes' theorem to calculate probabilities for multiple data association hypotheses based on known probabilities of detection and assumed densities of clutter and new targets.

Careful analysis of the results of the previous section indicates that some RCDs are assigned to both a corner and a wall target—for example, clusters 1 and 6 in Figure 4.16 share three RCDs. The MHT approach strictly enforces the constraint that a single measurement has a single origin: plane, corner or multiple reflection (false alarm). Each new unexplained observation initializes a tree of possible interpretation hypotheses. The tree is grown as new observations are validated with hypothesized targets, and is subsequently pruned to choose the single best interpretation of all past measurements. Using multiple hypothesis tracking software written at NEC Research Institute, successful map building results are reported in [35] for the same data set used above. Accuracies similar to Tables 4.1 and 4.2 are achieved.

The potential drawbacks of the MHT approach in comparison with the rule-based approach described earlier are:

- It is considerably more complicated.

- It assumes that the target estimate can be initialized from a single observation [111].

This latter point rules out the classification of single, isolated returns from a wide-angle sonar, unless amplitude or some other cue is used to obtain a direct measure of target bearing. We believe, however, that MHT techniques are very powerful, and have broader application beyond the problem of plane-corner-cylinder classification [35].

4.7   Alternative Approaches

The unpublished work of Bozma and Kuc [15] describes an approach for differentiating corner, plane, and edge targets using densely sampled data.

Edges can be distinguished because they have smaller visibility angles than corners and planes. Corners and planes can be differentiated by observing the change in orientation of the arc (RCD in our terminology) each target produces when viewed from two locations. This approach appears well-suited to carefully engineered hardware in which weak returns are not present and target visibility angles do not change with range. Our approach is more complicated, but can accommodate partially occluded targets and single isolated returns in the same framework as for densely sampled data.

With the exception of Kuc and his associates, previous work has tended to adopt one of two alternative approaches: line segment-based or grid-based. Our initial attempts at extracting line segments from sonar data were terribly frustrating, an experience reported by others. (Reconsider Figures 2.26 and 2.27.) Repeated observations of actual data and consideration of the physics of sonar led us to the RCD-based approach developed above, which we feel out-performs the line segment-based approaches of previous researchers. However, it should be mentioned that line segment fitting performance will depend significantly on the sonar hardware configuration. For example, comparing the two experimental systems used by Crowley [37], [38], it is clear that fitting straight-line segments should be much more reliable with the Robuter's sensing configuration than with a single servo-mounted sonar or with a circular ring of sonars.

The brittleness of line segment-based feature extraction was a motivating force behind the occupancy grid concept of Moravec and Elfes [45]. Grid-type representations are primarily useful for obstacle avoidance, as Borenstein and Koren's recent successful results attest [14]. However, we feel a grid-type map is less useful for predicting what sonar data will actually be observed from a given position, because different target types are not explicitly represented. As a result, the explanation of single sonar measurements is made difficult. Hence, position estimation is only possible with a correlation-based technique, which is time-consuming and subject to the cell resolution of the grid. Gilbreath notes that one motivation for using a grid-type representation is that "it is hard to accurately glean polygonal information from inexpensive sensors mounted on a mobile robot" [56]. The results presented above provide strong evidence to the contrary.

For sonar interpretation we feel it is much easier to extract geometry directly from the raw data, rather than that which has accumulated in the grid. The information content of a return is highly dependent on the position from which it was obtained. We think of individual returns as vectors of length equal to the TOF range pointing from the vehicle position at the angle of the sensor orientation. Detaching the TOF dot from "the center of the circle" discards a great deal of information. We feel a highly desirable characteristic of Borenstein and Koren's vector field histogram formulation is that it retains this sense of a return as a vector in the construction of the

polar histogram [14].

Beckerman and Oblow [9], Gilbreath and Everett [56], Noborio *et al.* [108] and Zelinsky [135] have used a grid representation to build sonar maps. Beckerman and Oblow have presented a map building algorithm that uses a consistent labeling scheme to resolve conflicts in a 6 inch cell resolution grid [9]. They use data structures called *strings* to refer to "sequences of returns of similar range from adjacent scan angles bounded at both ends by depth discontinuities" [9]. These seem to be a discrete equivalent to our concept of regions of constant depth. The approach explicitly assumes the absence of environment features smaller than a single grid cell, which would rule out many features that are common in typical environments, such as chair and table legs.

The vehicle Robart II, used by Gilbreath and Everett [56], uses two sonar rings to build grid-type maps, one positioned near the floor and the other about 4 feet off the ground. Mapping is based on the assumption that transient obstacles tend to be lower to the ground, and hence visible to the lower sonar ring, while permanent room features such as walls will be visible to the higher mounted ring. Zelinsky has implemented a successful map building technique which produces accurate grid-type maps with 6 inch cell resolution [135]. Whereas Beckerman and Oblow use scans from accurately measured positions, Zelinsky and Gilbreath show results in which the vehicle moved autonomously to build the map, relying on odometry for position estimates after starting from a known location.

Overcoming multiple reflections is a vital task for any map building scheme. Zelinsky [135] uses the sonar barrier test proposed by Drumheller for this purpose. Gilbreath [56] and Mataric [92] fight multiple reflections by discarding any measurements greater than some pre-defined distance, which is 6 feet in Gilbreath's case. Because of multiple reflections, Noborio *et al.* only use one return out of the 24 returns that comprise each scan [108]. In this approach, a quadtree representation is used. The shortest return obtained in a 24 return sonar scan is used to define a *free circle* which is assumed to be obstacle-free. Position estimation during the process of map construction is not considered. In our approach, the effects of multiple reflections do occasionally manifest themselves, but not to the great extent to which they have plagued previous work. One reason for this is that multiple reflections are often considerably weaker, and thus are discarded in the process of RCD extraction. As shown in Figures 4.14 to 4.17, strong multiple reflections will cause erroneous point targets to be initiated, but these will not be consistently observable from a wide range of vehicle positions.

We feel the ultimate test of a map is not "does it look good?" but "how accurately, adequately, or quickly can X be performed with it?" (X in our case stands for localization, while for Borenstein and Koren it stands

for obstacle avoidance.) We have not yet made the connection of feeding an autonomously built map back to be used for localization. This is high on our agenda for future research, as we see this as the only real proof of performance.

4.8 Why Build Accurate Maps?

A basic premise of our research is that precise, dynamic localization is necessary for many applications. Our line of reasoning progresses to assert that to provide this capability, maps of some kind are necessary. Further, to use *sonar* for this purpose, detailed and precise maps are necessary, so that actual sonar observations can be predicted. Many people, especially those in the behavior-based control community, would dispute this line of reasoning:

> Since globally consistent world models are hard to build and maintain, and perhaps not useful, consider the possibility of not building them at all. Instead let us consider the possibility of using the world as its own model. [19]

Brooks would take issue with our assertion that precise (x, y, θ) knowledge of position with respect to a world coordinate frame is necessary. Human beings do not answer the question "where am I?" in millimeters and degrees, so why must a robot? Mataric has provided experimental proof of successful sonar-based map building and navigation without an explicit geometric approach [92]. For example, her robot Toto could "handle getting lost, and relocate itself in the map later" [20].

The answer to this question is application-dependent; while qualitative knowledge of position may suffice for many applications, precise numeric knowledge of position is essential for others. The maps we want to build are not an attempt at complete, detailed "world models". We make no claims regarding their sufficiency for planning. They are not three dimensional, as visual maps need to be. Our purpose is to capture a sufficient amount of real-world geometry to make subsequent registration with respect to a world coordinate frame possible and efficient. Our maps are grounded in the physical reality of acoustics. We dispute the assertion that "sonar data, while easy to collect, does not by itself lead to rich descriptions of the world useful for truly intelligent interactions" [18]. We feel that although maps have been difficult to build with vision (though considerable progress has been made, for example by Ayache and Faugeras [5]), sonar map building is feasible with today's sensors, as long as we think acoustically. These models will of course be insufficient for many intelligent tasks, such as manipulation, but they can be sufficient in many application environments for navigation.

We believe the experimental results of the proceeding chapter show that globally consistent maps are useful for position estimation. The aim of this chapter has been to show that with a good sensor model, accurate maps can be built with sonar. Maintenance of the map is still a potential difficulty; we will discuss this subject in the next chapter.

Chapter 5

Simultaneous Map Building and Localization

This chapter attempts to combine our earlier results to develop a unified framework for navigation. Unfortunately, combining the capabilities of localization and map building described earlier is not as straightforward as one might hope; a number of complicating research issues enter the picture, which will be described below. While these issues are important regardless of the sensing modality employed, for our purpose here we shall maintain our exclusive focus on sonar.

5.1   A Unified Approach to Navigation

Previous work in navigation has tended to treat the problems of localization, obstacle avoidance, and map building in isolation. Approaches to obstacle avoidance, such as the method of potential fields [127], are typically of no use for localization. Algorithms for globally referenced position estimation, such as the work of Cox [33], Sugihara [124], or Drumheller [40], rely on an *a priori* map, but do not incorporate the construction of such a map. Many algorithms for map building do not address the issue of localization while the map is being constructed, relying instead on odometry or hand-measuring of sensing locations [9], [56], [135]. The challenge posed by a complete implementation of autonomous navigation is the simultaneous realization of these different capabilities.

Figure 5.1 presents an overview of our approach to navigation, which unifies the various aspects of the navigation problem in a common multitarget tracking framework. As stated in the previous chapters, the use of the term *target* in our application refers to a naturally occurring feature of the environment that can be observed by the sensors of the robot.

The estimation of both vehicle and target locations by the use of sensor information is carried out via multiple EKF's. The crucial task in this procedure is *data association*, the process of achieving correspondence. Data association comes in two forms: *data-to-target* association and *data-to-data* association.

The set of state estimates for all currently known targets is called the *map*. Observations that can be associated with targets in the map will be called *expected* data. In contrast, observations that cannot be associated with the map represent *unexpected* data. All sensor data is either expected or unexpected. Localization is a process of using expected data to estimate vehicle state. Detecting obstacles and changes to the environment requires the explanation of unexpected data. In this sense, localization and obstacle avoidance can be regarded as complementary. Map building and map maintenance require the use of both of these capabilities. Expected observations verify the presence of targets in the map and provide a means of localization while further building the map. Unexpected observations signify new environment features that need to be added to the map.

Targets in the map are classified at two levels: *confirmed targets* and *tentative targets*. Confirmed targets are known with a high degree of geometric precision, and are suitable for use in updating the vehicle position. Tentative targets are still in the stage of being learned, and hence have a large geometric uncertainty associated with them.

Data-to-target association takes place in measurement space, by comparing actual sensor observations with predicted sensor observations generated for targets in the map. A sensor model provides for prediction: given the map and a hypothesized vehicle location, what observations should the sensors produce? Matched observations and predictions for confirmed targets are used to update the vehicle position estimate, while matches to tentative targets update the target state estimate.

The rejects from data-to-target association are unexpected observations. Multiple unexpected observations from different vehicle locations are matched to one another in a process of data-to-data association. Matches that yield a unique target interpretation are used to initialize new tentative targets into the map. Subsequent matches to tentative targets are used to update target state, not vehicle state. After a number of subsequent observations of a tentative target are obtained to attain a sufficient level of geometric precision, the target is promoted to be a *confirmed* target. Subsequently, matched observations to this target are used to update the vehicle position.

The sensor model described in Chapter 2 tells us that in the case of sonar, expectations and observations take the form of regions of constant depth (RCDs). Observed and predicted RCDs are matched using a validation gate defined by the Mahalanobis distance [7]. Matched RCDs provide

Figure 5.1: A unified approach to navigation.

innovations to update the vehicle location and confirm the presence of prior known targets. The rejects from the validation gate represent unexpected data, and need to be explained. As discussed in Chapter 4, a given unexpected RCD can be caused by a variety of geometric targets. By evaluation of data-to-data association hypotheses for unexpected RCDs from different vehicle locations, the correct target identity can be classified. This provides a means of precisely learning the geometry of a scene by observing unknown targets from multiple positions as the vehicle moves.

5.2   Research Issues

Unfortunately, implementing this navigation framework is not as straightforward as one might hope. A major reason for this is the *correlation problem*: if a mobile robot uses an observation of an imprecisely known target to update its position, the resulting vehicle position estimate becomes correlated with the feature location estimate. Likewise, correlations are introduced if an observation taken from an imprecisely known position is used to update the location estimate of a geometric feature in the map. A rigorous solution to simultaneous map building and localization must explicitly represent all the correlations between the estimated vehicle and target locations.

One solution was provided by Smith, Self and Cheeseman [115], who developed a Kalman filter based approach for building a *stochastic map* of spatial relationships. Moutarlier and Chatila have presented a framework similar to the stochastic map, but with the added feature that colored and correlated noise is accommodated, and have implemented their approach in two dimensions using laser range data [104]. Our engineering instincts tell us that the stochastic map would be tremendously difficult to implement in practice. One issue is computational complexity: in a two-dimensional environment containing n geometric features, the stochastic map requires a system state vector of size $2n + 3$. Because the EKF is $\mathcal{O}(n^3)$ [97], the computational burden of the approach would be substantial in typical application environments with hundreds of map features.

Two additional considerations are *data association uncertainty* and *environment dynamics*. As discussed earlier, data association is the process of determining the origins of sensor observations, that is, associating measurements with the geometric features that produced them while at the same time rejecting spurious measurements [35]. Because the works of both Smith, Self and Cheeseman and Moutarlier and Chatila implicitly assume perfect data association, spurious measurements and incorrect matches or groupings would likely have a severe impact on the stability of the stochastic map representation. To further complicate the problem, dynamic environments require an inference procedure in which missed detections provide

information pertaining to state changes in the map, such as the "disappearance" of previously learned geometric features [87].

5.3 Restatement of the problem

To facilitate a more detailed consideration of these three issues, we now present a brief restatement of the problem. The state of the environment is specified by the *true map*, the set of target locations

$$M = \{\mathbf{p}_t | 1 \le t \le n\} \tag{5.1}$$

where n is the number of targets (geometric features) in the environment. As described earlier, we model the environment in two dimensions with three types of target: points, lines and arcs. The point target classification encompasses both corners (concave dihedrals) and edges (convex dihedrals [83]). Lines represent 3-D planes and arcs represent 3-D cylinders. The form of the target parameterization depends on the type of the feature, as described in Chapter 2.

In the general setting of a dynamic environment, the true map is time-varying as feature locations can change with time according to unknown feature plant models. For our purposes here, we shall assume a static environment in which feature locations do not change with time. Together, the location of the vehicle and the state of the environment comprise the *system state vector* $\mathbf{x}_S(k)$:

$$\mathbf{x}_S(k) = [\mathbf{x}(k), \mathbf{p}_1, \ldots, \mathbf{p}_n]^T. \tag{5.2}$$

As described in Chapter 3, the set of sonar observations obtained at time k comprise the current data set $Z(k)$:

$$Z(k) = \{\mathbf{z}_j(k) | 1 \le j \le m(k)\}. \tag{5.3}$$

The cumulative data set Z^k is the set of data sets up through time k:

$$Z^k = \{Z(j) | 0 \le j \le k\}. \tag{5.4}$$

We introduce the notation

$$\mathbf{z}_j(k) \leftarrow \mathbf{p}_t(k)$$

to indicate that target t **generates** measurement $\mathbf{z}_j(k)$ at time k. Each measurement in a data set is assumed to be generated by (i.e. originate from) a single target in the environment or to be a false alarm (in which case we write $\mathbf{z}_j(k) \leftarrow \emptyset$). The value of a measurement $\mathbf{z}_j(k)$ is a function of the vehicle location at time k and the location of the target from which

it originated, subject to a noise disturbance, as given by the measurement models described in Chapter 3.

The problem of simultaneous map building and localization requires the computation of the number of features present, the type of each feature (wall, corner, or cylinder), and $p(\mathbf{x}_S(k) \mid Z^k)$, the *a posteriori* probability distribution of vehicle and target states conditioned on the cumulative measurement set Z^k [102]. The objective of the navigation framework shown in Figure 5.1 is to use the EKF to recursively compute a minimum mean square error (MMSE) estimate for $\mathbf{x}_S(k)$:

$$\hat{\mathbf{x}}_S(k \mid k) = [\hat{\mathbf{x}}(k \mid k), \hat{\mathbf{p}}_1(k), \ldots, \hat{\mathbf{p}}_n(k)]^T \tag{5.5}$$

which is designated the *system state estimate*, and its covariance $\mathbf{\Lambda}(k \mid k)$ (the system covariance matrix):

$$\mathbf{\Lambda}(k \mid k) = \begin{bmatrix} \mathbf{P}(k \mid k) & \mathbf{C}_{R1}(k \mid k) & \cdots & \mathbf{C}_{Rn}(k \mid k) \\ \mathbf{C}_{R1}(k \mid k) & \mathbf{\Lambda}_1(k) & \cdots & \mathbf{C}_{1n}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{Rn}(k \mid k) & \mathbf{C}_{1n}(k) & \cdots & \mathbf{\Lambda}_n(k) \end{bmatrix} \tag{5.6}$$

where $\mathbf{C}_{Rt}(k \mid k)$ is a vehicle to target cross-covariance matrix and $\mathbf{C}_{ij}(k)$ is a target to target cross-covariance matrix.

The application of the EKF to undertake this computation was first presented by Smith, Self, and Cheeseman [115], under the assumption of perfect data association. Briefly, they advocate a two stage process of

1. vehicle position prediction
   compute: $\hat{\mathbf{x}}(k + 1 \mid k)$, $\mathbf{P}(k + 1 \mid k)$, and $\{\mathbf{C}_{Rt}(k + 1 \mid k) | 1 \leq t \leq n\}$
   given: $\mathbf{u}(k)$, $\hat{\mathbf{x}}(k \mid k)$, $\mathbf{P}(k \mid k)$, and $\{\mathbf{C}_{Rt}(k \mid k) | 1 \leq t \leq n\}$.

2. system state update
   compute: $\hat{\mathbf{x}}_S(k + 1 \mid k + 1)$ and $\mathbf{\Lambda}(k + 1 \mid k + 1)$
   given: $Z(k + 1)$, $\hat{\mathbf{x}}(k + 1 \mid k)$, $\mathbf{P}(k + 1 \mid k)$, $\{\mathbf{C}_{ij}(k) | 1 \leq i < j \leq n\}$,
   and $\{\hat{\mathbf{p}}_t(k), \mathbf{\Lambda}_t(k), \mathbf{C}_{Rt}(k + 1 \mid k) | 1 \leq t \leq n\}$.

In the absence of spurious measurements or incorrect data-to-target assignments, the equations reported in [115] provide the optimal linear estimate for the system state at time k. To our knowledge, implementation results with this algorithm have not been published.

Moutarlier and Chatila have observed that in practice, the stochastic map approach is subject to divergence, principally because of biases introduced by linearization. In our own sonar-based experiments with the direct algorithm, we have also observed this divergent behavior. To alleviate this effect, Moutarlier and Chatila have proposed a suboptimal, three stage procedure, called the *relocation-fusion* approach, in which the system update step above is replaced by:

2 (a). vehicle position update (relocation)

compute: $\hat{\mathbf{x}}(k+1 \mid k+1)$, $\mathbf{P}(k+1 \mid k+1)$, and
$\{\mathbf{C}_{Rt}(k+1 \mid k+1)|1 \le t \le n\}$
given: $Z(k+1)$, $\hat{\mathbf{x}}(k+1 \mid k)$, $\mathbf{P}(k+1 \mid k)$, and
$\{\hat{\mathbf{p}}_t(k), \boldsymbol{\Lambda}_t(k), \mathbf{C}_{Rt}(k+1 \mid k)|1 \le t \le n\}$.

2 (b). map update (fusion)

compute: $\{\hat{\mathbf{p}}_t(k+1), \boldsymbol{\Lambda}_t(k+1)|1 \le t \le n\}$, and
$\{\mathbf{C}_{ij}(k+1)|1 \le i < j \le n\}$
given: $Z(k+1)$, $\hat{\mathbf{x}}(k+1 \mid k+1)$, $\mathbf{P}(k+1 \mid k+1)$,
$\{\hat{\mathbf{p}}_t(k), \boldsymbol{\Lambda}_t(k), \mathbf{C}_{Rt}(k+1 \mid k+1)|1 \le t \le n\}$, and
$\{\mathbf{C}_{ij}(k)|1 \le i < j \le n\}$.

By updating the vehicle state and then re-linearizing before attempting any feature state updates, stability is enhanced [104].

We note that with either approach, $\boldsymbol{\Lambda}(k \mid k)$ is a $2n + 3$ by $2n + 3$ matrix whose cross-covariance sub-matrices $\mathbf{C}_{Rt}(k \mid k)$ and $\mathbf{C}_{ij}(k)$ are non-zero. Because the vehicle and target estimates are correlated, then each time the vehicle position is updated, each vehicle-target covariance matrix $\mathbf{C}_{Rt}(k \mid k)$ must be recomputed. Similarly, each time a target state estimate is updated, one must update all cross-covariance matrices involving that feature.

The results of the previous chapters addressed two limited forms of this general problem. Localization with an *a priori* map can be restated as: compute $\hat{\mathbf{x}}(k \mid k)$, a globally referenced estimate of the robot location at time k, given the set of control inputs $U^k = \{\mathbf{u}(i)|0 \le i \le k\}$, a map of the environment $M(k) = \{\mathbf{p}_t(k)|1 \le t \le n\}$ and the cumulative data set Z^k. Map building from precisely known sensing locations can be stated as: compute an estimate $\hat{M}(k) = \{\hat{\mathbf{p}}_t(k)|1 \le t \le n\}$ of the state of the environment given $\{\mathbf{x}(i)|0 \le i \le k\}$ and the cumulative data set Z^k, i.e. estimate the globally referenced locations of all geometric features in the environment, given sensor measurements from known locations.

The motion and structure from motion problem discussed in Chapter 3 can be stated as: estimate the displacement $\mathbf{d}_k$ (comprised of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$) between two sensing locations $\mathbf{x}(k-1)$ and $\mathbf{x}(k)$, and position estimates for features visible at time k in the *relative* coordinate frame defined by the vehicle's current position. By compounding [114] successive motion estimates starting at time $k = 0$, a globally referenced estimate of vehicle location can be computed:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(0 \mid 0) \oplus \hat{\mathbf{d}}_1 \oplus \hat{\mathbf{d}}_2 \oplus \ldots \oplus \hat{\mathbf{d}}_k \qquad (5.7)$$

Our objection to this formulation rests with the covariance $\mathbf{P}(k \mid k)$ that accompanies this global position estimate—it increases without bound:

$$\lim_{k \to \infty} \text{trace}(\mathbf{P}(k \mid k)) = \infty \qquad (5.8)$$

The robot is gradually getting lost, albeit perhaps quite slowly. No globally referenced geometry is computed to enable the robot to determine its position when it travels through the same part of the world an hour or a week later. The difficult question is "could the robot return to its starting position, within a desired tolerance, such as 1 centimeter and 1 degree, after traveling a considerable distance?" To do this, one needs to compute globally referenced geometry, which requires one to confront the correlation problem.

5.4   A Strategy for Decoupling $\boldsymbol{\Lambda}(k \mid k)$

Qualitatively, the utility of an observation from a map building perspective depends on the accuracy with which the vehicle position from which the observation was taken is known. Conversely, the utility of an observation for position estimation depends on the accuracy with which one knows the location of the feature being observed. We propose to take this view to its extreme: measurements from inaccurately known positions should not be used to update the location estimates of any geometric features; rather, they should be simply thrown away.

We propose the following strategy: the robot can choose where it *goes* and where it *looks* to build the map incrementally in a fashion such that 1) correlations are eliminated and 2) position in the global frame can be estimated to within a desired tolerance, such as 1 centimeter and 1 degree. The robot must have the ability to determine its position precisely, and to do so it must precisely know the locations of some targets.[1]

To evaluate the accuracy of a state estimate, we shall compare the trace of its covariance matrix $\mathbf{P}(k \mid k)$ or $\boldsymbol{\Lambda}_t(k)$ with a tolerance parameter ϵ_R or ϵ_F, as appropriate. The values of ϵ_R and ϵ_F reflect when a covariance matrix can be safely approximated as zero, and can be chosen experimentally. Feature estimates that meet the criterion $\mathrm{trace}(\boldsymbol{\Lambda}_t(k)) < \epsilon_F$ will be designated *confirmed* targets. Likewise vehicle position estimates that satisfy $\mathrm{trace}(\mathbf{P}(k \mid k)) < \epsilon_R$ will be designated confirmed vehicle locations.

The proposed sensing strategy can be stated as follows:

> **if** $\mathbf{z}_j(k) \leftarrow \mathbf{p}_t(k)$:
>
> > **if** $\mathrm{trace}(\boldsymbol{\Lambda}_t(k))$ update $\hat{\mathbf{x}}(k \mid k)$ with $\mathbf{z}_j(k)$
> >
> > **else if** $\mathrm{trace}(\mathbf{P}(k \mid k)) < \epsilon_R$ update $\hat{\mathbf{p}}_t(k)$ with $\mathbf{z}_j(k)$
> >
> > **else** ignore observation $\mathbf{z}_j(k)$.

---

[1]This does not mean that the robot must precisely knows its position *all the time*, only that it must do so before updating any target state estimates in its map.

First, before moving, the robot needs to learn accurately the locations of an initial confirmed feature set. Subsequently, after each move, the robot takes observations of the initial feature set to determine accurately its position, then takes observations of new features to estimate their locations. Additional observations of these new features from subsequent accurately known sensing locations will result in sufficiently accurate estimates that these features can in turn be promoted to confirmed target status, and used for vehicle localization.

We claim that this approach can sidestep the correlation problem. The use of individual vehicle and target covariances will suffice, without explicit representation of cross-correlations. Observations of confirmed targets update vehicle position but not target state; observations of tentative targets update target state but not vehicle position.

5.5 Initialization With Sonar

When using sonar, however, the initialization of this incremental procedure seems impossible. The vehicle must move precisely to obtain sufficient multiple views of targets to classify them and precisely estimate their locations. However to move precisely, the vehicle must have some map. We are confronted with a question of *which came first, the chicken or the egg?* (the map or the motion?).

To deal with this issue, we advocate the use of multiple servo-mounted sensors mounted on the vehicle. One approach would be a *four corners* sensing configuration, in which servo-mounted sensors are placed at each corner of the vehicle. This configuration provides a way out of the getting started paradox. The portion of the world visible from the initial vehicle location can be precisely learned *without* moving the vehicle, because of the known baseline between sensors. To provide evidence for this assertion, reconsider Figures 4.21 and 4.22 of the previous chapter, that show a repeat of the original map building run using the data of just four scans. In this way, an initial confirmed target set can be learned without vehicle motion. These confirmed targets can then be tracked as the vehicle moves to provide accurate localization, while other sensors can be used to map out new targets.

5.6 Dynamic Map Building and Maintenance

This section summarizes joint research with Ingemar Cox [87]. An important issue that we have not yet addressed is the maintenance of a map in a dynamic environment. The framework of Figure 5.1, is lacking a facility by which targets no longer present in the environment can be removed. This

issue is vital, despite the fact that discussions of this topic are hard to find in the literature. To incorporate changing environments, we propose the following extension to the navigation framework. In addition to the covariance matrix $\mathbf{\Lambda}_t(k)$ used to represent uncertainty in the geometric estimate of target t, we add a measure of belief in the *existence* of target t. We define $c_t(k)$, a scalar taking values between 0 and 1, to be the *credibility* of target t. The expanded map is now defined to be:

$$M(k) = \{\hat{\mathbf{p}}_t(k), \mathbf{\Lambda}_t(k), c_t(k), \beta_t | 1 \leq t \leq n_T\}. \tag{5.9}$$

The result of matching is to yield three sets: matched predictions and observations, unobserved predictions, and unexpected observations. For each target, let $n_s(k)$ be the number of matched predictions, and $n_u(k)$ be the number of unobserved predictions, up through time k. The following update rule for $c_t(k)$ was proposed in [87]:

$$c_t(k) = 1 - e^{-(n_s/A - n_u/B)}. \tag{5.10}$$

The coefficients A and B dictate the rate of learning and forgetting, respectively. Targets for which $c_t(k)$ falls below zero are removed from the map. Figure 5.2 shows the extension of the navigation framework of Figure 5.1 to incorporate map building and maintenance.

To provide results that incorporate this extension, Figure 5.3 shows a longer map building run from the same data set as used in the previous chapter. Scans 2 and 26 were taken with a changed environment, in which a chair was inserted into the left part of the room, and a door leading into the upper right part of the room was opened, to present two transient objects. Figures 5.4 and 5.5 show these two modified scans. The scans were processed off-line in a spiral sequence, starting from the upper left of the figure. For this run, A and B, the learning and forgetting factors for the credibility update function (Equation 5.10), were both set to a value of 5.

5.7   Summary

While we believe that Smith, Self, and Cheeseman [115] and Moutarlier and Chatila [104] have made pioneering contributions in this area, we feel the general problem of simultaneous map building and localization remains open for three reasons: 1) the impact of correlations on computational complexity; 2) data association uncertainty, and 3) dynamic environments. As described in [35], we believe that multiple hypothesis tracking (MHT) provides a rigorous framework in which to address the latter two problems.

To deal with the correlation problem, we have sketched a sensing strategy that may permit one to sidestep the issue. To achieve genuine long-term autonomy, it is not enough just to *represent* uncertainty; we need to *reduce*
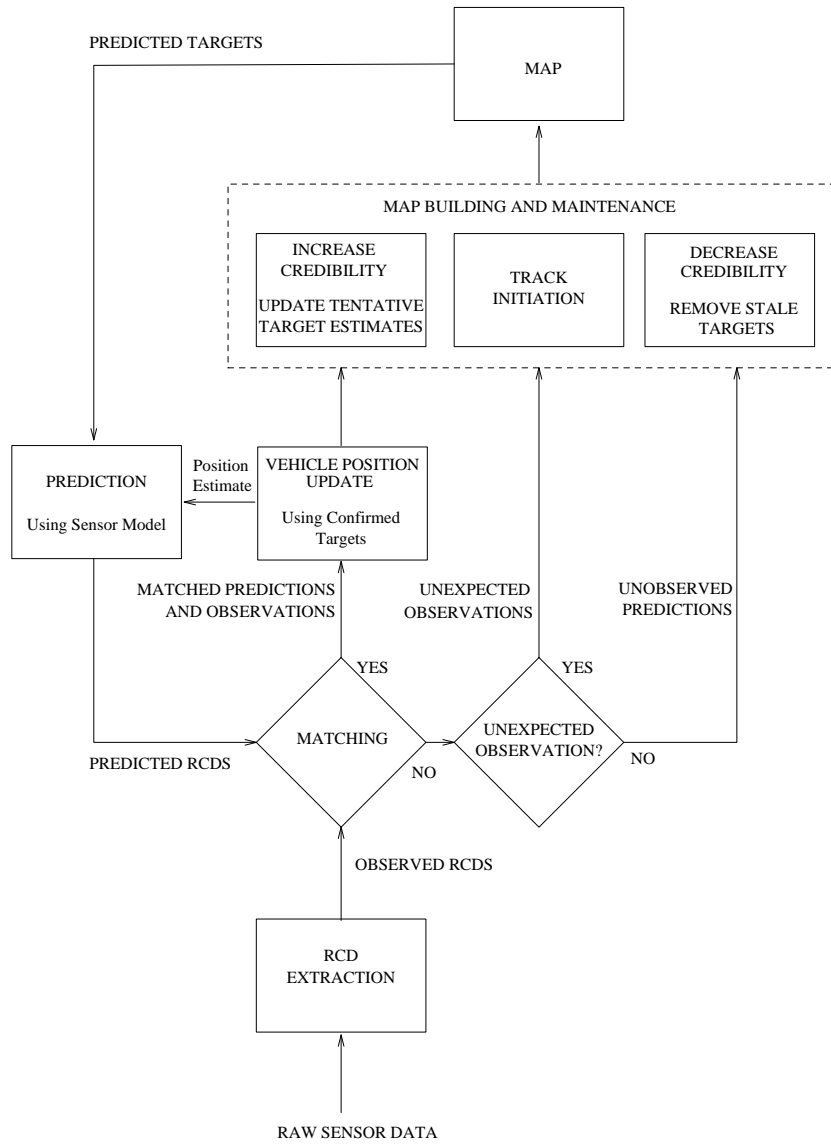
Figure 5.2: Expanded navigation framework incorporating map building and maintenance. (This expansion of the framework was the product of collaboration with Ingemar Cox of NEC Research Institute.)
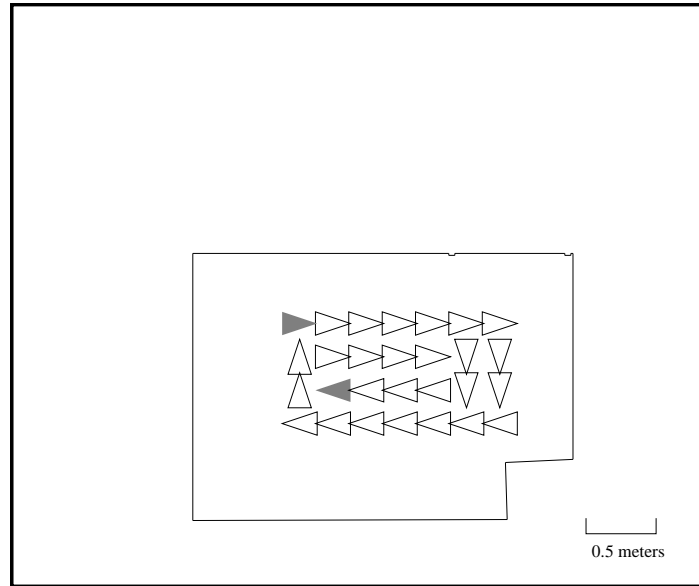
Figure 5.3: Scan locations for an extended run in the same room as before, using the dynamic map building extension. The shaded triangles indicate the start and finish of the run.

uncertainty. The sensing strategy presented above aims to eliminate the vehicle position uncertainty associated with a new set of measurements before using those measurements to update the map, thereby eliminating the cross-correlations between map features. This decouples the system covariance matrix, easing the state estimation computational burden and simplifying the data association procedure.

The global frame is anchored by an initial confirmed target set that is learned before any vehicle motion takes place. Because walls, corners, cylinders, and multiple reflections are indistinguishable in a single sonar scan, we require that multiple servo-mounted sonars be mounted on the vehicle, e.g., in a four corners sensing configuration. However, we cannot at this stage verify the approach with experimental results. Experimentation is required to establish what level of correlations can be safely ignored, and at what stage significant performance degradation (i.e. EKF divergence) occurs.

To address the issue of map maintenance in a changing environment, we have summarized an expansion of the basic navigation framework that uses the information provided by unobserved predictions (missed detections) to decrease the credibility of targets in the map [87].

One might argue that we overestimate these difficulties. Despite their
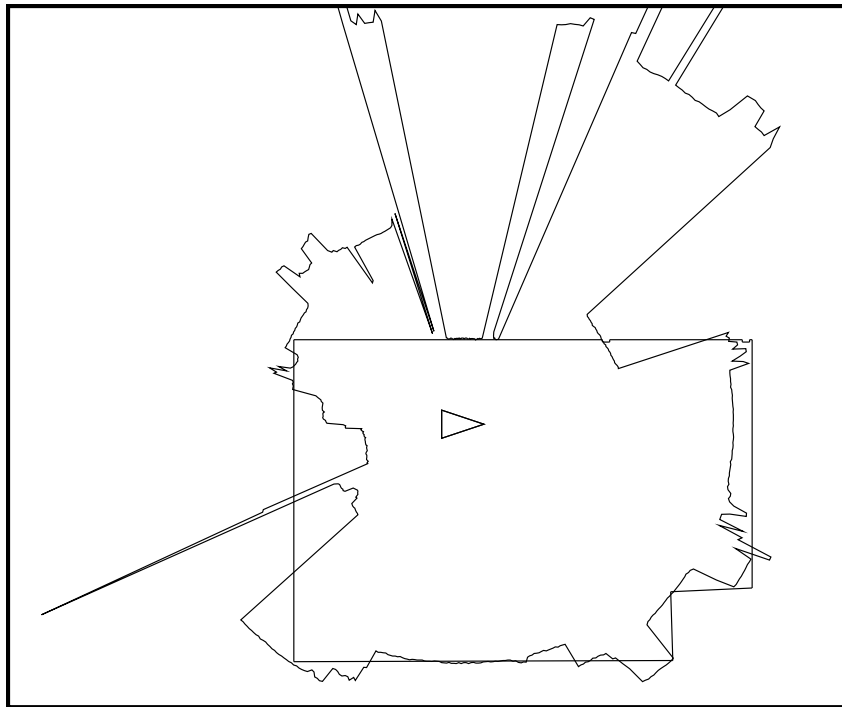
Figure 5.4: Scan taken with a modified environment. A chair was placed in the left part of the room, obstructing the left wall, and the door in the upper right-hand part of the room was opened. This is the 2nd scan processed in the sequence, and was taken from the same position in the room as the scan b0, shown at the start of Chapter 3.
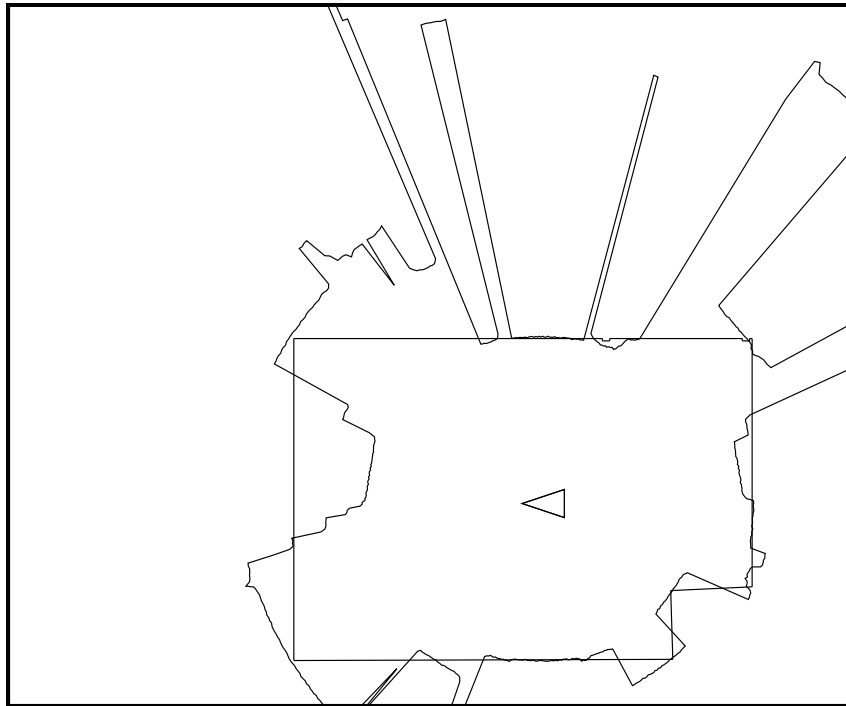
Figure 5.5: Scan taken with a modified environment. This is the 26th scan processed in the sequence.
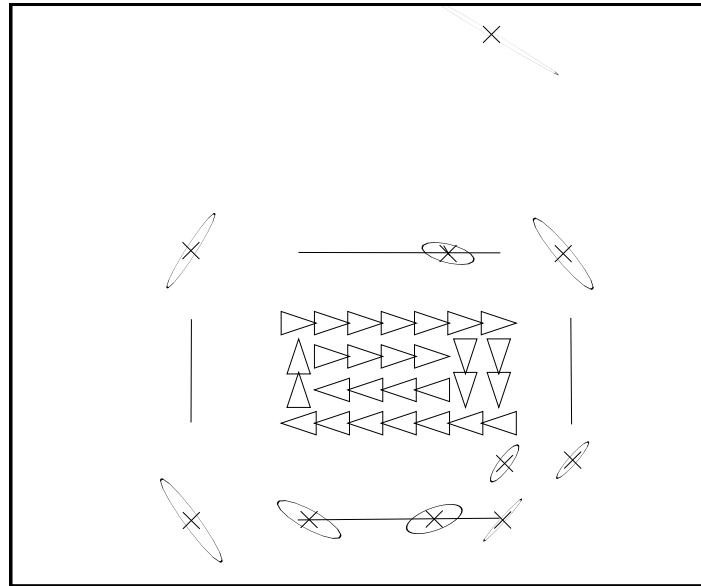
Figure 5.6: Map of the room produced with the dynamic map building algorithm. 8σ (major and minor ellipse axes magnified by 8) error ellipses are shown for point (corner and edge) targets.
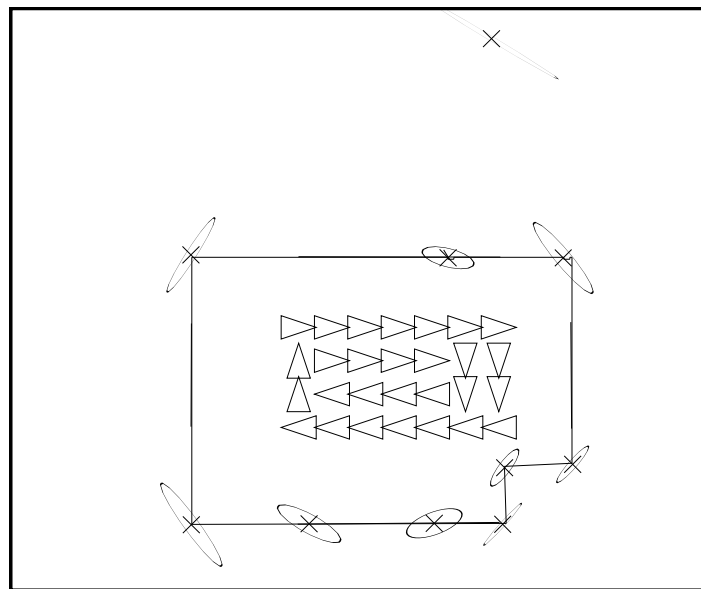
Figure 5.7: Learned map superimposed over room model.
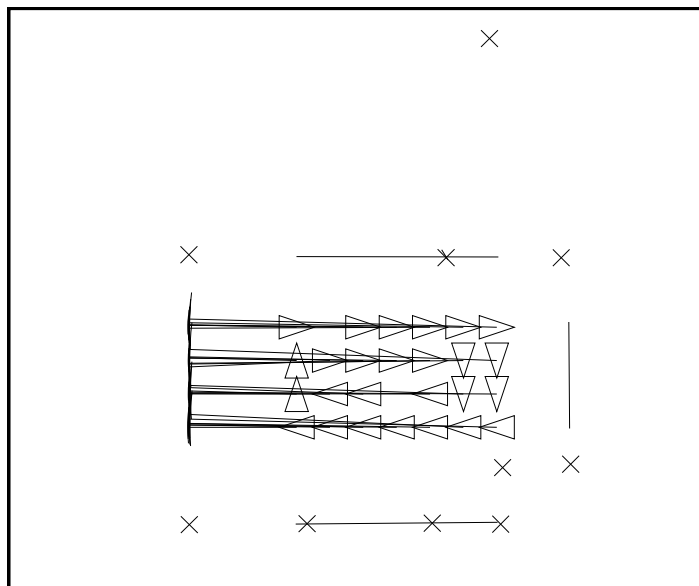
Figure 5.8: RCDs matched to initialize a line target for the left wall. A triangle is shown at each location from which the target was observed. Note the absence of a triangle for scans 2 and 26, the modified scans. A line is drawn from each vehicle location to the midpoint of each RCD.
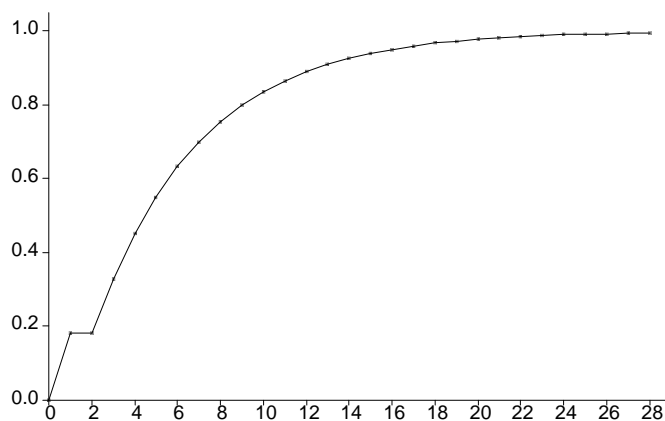
Figure 5.9: Credibility *vs* time for the line target shown in Figure 5.27. The credibility $c_t(k)$ rises exponentially as this target is repeatedly sighted, except for steps 2 and 26 when the environment was modified to occlude this target.
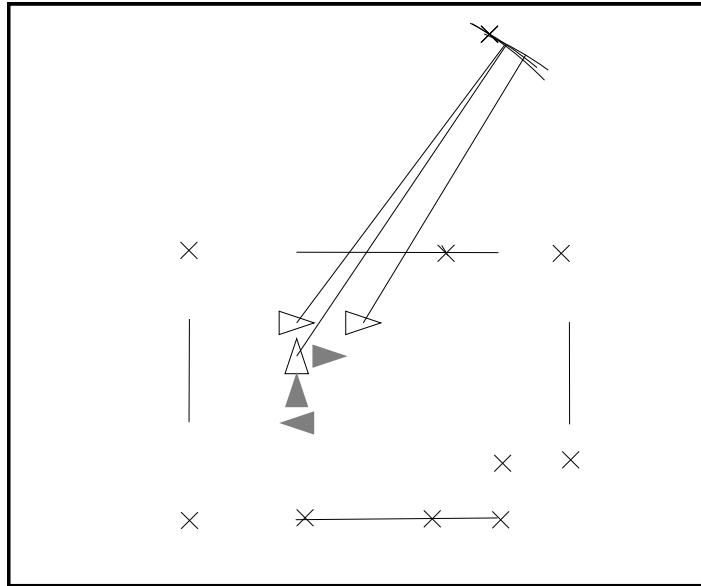
Figure 5.10: A point track initiated by multiple reflection RCDs. The open triangles show locations from which the hypothesized target was observed. The shaded triangles show locations "between" the first two sightings from which the hypothesized target was not observed.
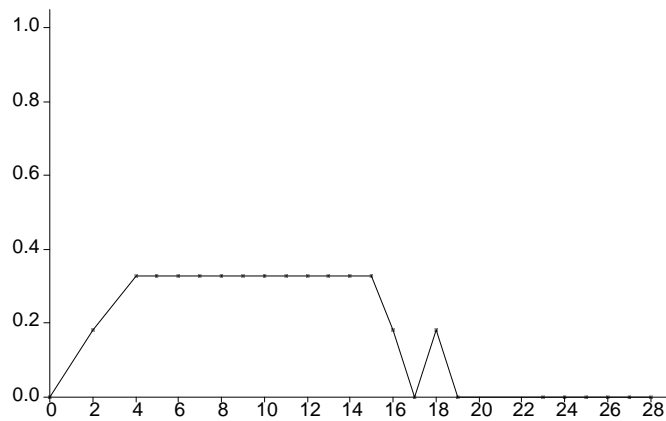
Figure 5.11: Credibility *vs* time for the target shown in Figure 5.29. The credibility $c_t(k)$ is increased for each observation (open triangles in Figure 5.29) and decreased for each unobserved prediction (shaded triangles in Figure 5.29).

large size, the covariance matrices of the stochastic map may be manageable in practice. Moutarlier and Chatila report that "the complexity of the computations are such that computing time is satisfactory (a few seconds for each step in the experiments shown ....)" [104]. In some situations, data association ambiguity may be infrequent, as in Moutarlier and Chatila's experiments, where the use of the Mahalanobis distance suffices. However, if this is not the case, an optimal algorithm that considers all possible data association hypotheses and represents all cross-correlations would require *exponentially many* $2n + 3$ by $2n + 3$ system covariance matrices.

An anonymous reviewer for one of Crowley's papers has compared simultaneous map building and localization to "pulling yourself up by your bootstraps" [38]. Rock-climbing offers a useful analogy for the problem. We see map building as a methodical, incremental process, in which each successive foothold (vehicle position) is made secure before the next advance. The uncertainty introduced into the system by each move of the vehicle must be eliminated as early and as often as possible. As a climber relies on a safety rope, secured to the rock face at regular intervals, our robot can hopefully retreat to safe, familiar terrain if it suddenly finds itself in a tough spot with no previously mapped targets in sight, and hence cannot drive its globally referenced position covariance to zero.

Chapter 6

Directed Sensing
Strategies

 The use of a variety of sensing modalities is a good idea, of course. We believe, however, that the key to progress in perception is extending the capabilities of the sensors we already have, not waiting for better sensors. At the least, such attempts should define what we want from better sensors. And so, while we expect commercial mobile robots to rely on multiple sensor systems for their redundancy and adaptability to a wide variety of environments, we suspend reason for this chapter and ask "could navigation (in all its meanings) be implemented exclusively with sonar?" and if so, "how?"

6.1   The Time Problem

We would now like to readdress the results of the previous chapters in terms of the question posed above, "is sonar-based navigation practical?" Previous work characterized sonar's two main problems as beam width and specularity. We have shown how navigation is possible despite these characteristics, and that in fact these "bugs" can be thought of as features. However, as a result of our approach, a new enemy has emerged: time.

   The scans processed in Chapter 4 contained 612 returns per scan, for a separation between returns of 0.588 degrees, far lower than any we have seen in the literature. The scanning device used, described in Appendix A, has a data acquisition rate of 5 returns per second. This implies a complete data acquisition time of 2 minutes per scan—a very impractical length of time. However, this does not reduce the validity of these results. Our reasons for the high scanning density were to try to develop a good sensor model, i.e. to explain the errors of real Polaroid data. These turn

out to be weak returns, arising from the 1.13 mSec duration of the standard transmitted waveform. The high scanning density provides two functions:

- The use of local support to form RCDs.

- The elimination of weak returns.

An important point to remember is that weak returns will be absent from data obtained in short-range sensing mode. Also, a point we have repeatedly stressed is that we rely heavily on range accuracy, while the information implied by the sensor orientation is used as a constraint for correspondence but not directly for estimation purposes. Hence, with a short-range sensing mode, significantly fewer returns suffice. The implementations of localization in Chapter 3 rely on this fact. In general, the problem sonar interpretation poses is that the more returns you can acquire, the easier interpretation becomes, but the more returns you acquire, the longer data acquisition takes.

6.2   Tracking Sonars

In analogy with tracking radars [113], we propose the use of tracking sonars. Distributed around the vehicle periphery, each possesses its own servomotor and a limited amount of computational resources. This local processing can be used to focus attention on environment targets as the vehicle moves. In this way, drastic changes in observations due to vehicle rotations can be prevented by fast, local counter-rotations. An example layout of tracking sonars is the "four corners" sensing configuration discussed in the previous chapter. Tracking sonars seem ideally suited for realization via the LISA (locally intelligent sensor agent) distributed estimation and control concept being developed at the Oxford Robotics Research Group [44].

Tracking sonars can give a robot the ability to "grab hold" of environment features as it moves to localize position. The concept of wall-following is well-known in robotics. For example, an early realization of this was achieved by Hilare [28]. Many wall-following approaches fit line segments to multiple sonar returns obtained by a scanner or ring [77]. In our approach, this is unnecessary. The model-based localization algorithm described in Chapter 3 provides a method for using single range measurements to provide an accurate position update in just one degree of freedom, the direction perpendicular to the wall. By "range-locking" on a wall, for example when traversing a corridor, a high rate stream of position updates perpendicular to the wall can be achieved with a single sensor. A smart local processor can substantially reduce the correspondence problem, by counter rotating to stay fixated on the target as the vehicle moves, and determining transitions from one target to another, for example from the

wall to a corner as the end of the corridor or a doorway is reached. This is analogous to the use of region-of-interest windowing in computer vision.

6.2.1 Corridor Navigation

For example, consider the task of a robot going down a corridor. Directed sensing can be used to focus attention on each wall of the corridor. Position updates to the wall can be used to reduce uncertainty in the vehicle position estimate orthogonal to the wall. We have said nothing new so far, for wall-following has been implemented by many researchers [28], [116]. But what about position information in the direction parallel to the wall? Tracking sonars, looking ahead and behind the vehicle, can be used to "grab" and track corner features along the corridor. Vertically oriented corner features caused by door moldings, as commonly used for visual hallway navigation [78], [133], present ideal ultrasonic navigation beacons.[1]

We see wall-following as just one instance of a more generic target tracking capability for which sonar is ideally suited. Doorway moldings, table legs, and chair legs are all good candidates for sonar focus-of-attention. An added benefit of the of the use of multiple servo-mounted sensors on a single vehicle would be the capability of "looking" backwards as well as forwards. We envision a typical composite sensing strategy to traverse a corridor as follows:

- side-mounted sonars tracking wall targets to each side, to provide position information perpendicular to the corridor;

- corner-mounted sonars on the rear of the vehicle tracking point targets behind the vehicle to provide position information parallel to the corridor;

- front-mounted sonars "looking" ahead for obstacles, corner targets marking doorways, and the wall at the far end of the corridor.

6.2.2 Going Through a Door

From the manner in which sonar has been described in the literature, one might think that sonar is particularly ill-suited for the task of guiding a robot through a doorway. We believe the opposite is the case, but cannot yet provide experimental support for our position. Instead we show

---

[1]Kriegman provides a good illustration of sonar data taken in a hallway, pointing out a number of returns caused by doorway dihedrals [77]. His data looks somewhat different to the scans we have shown earlier, for it is hard to see circular arcs (RCDs) in his data. One explanation for this is that his scan was not taken by a single, centrally mounted scanner, but by a sonar ring of non-zero diameter rotated on-the-spot through a number of orientations. In this case, range values are not quite constant, and corner target visibility angles can be much smaller, depending on how close the corner is.
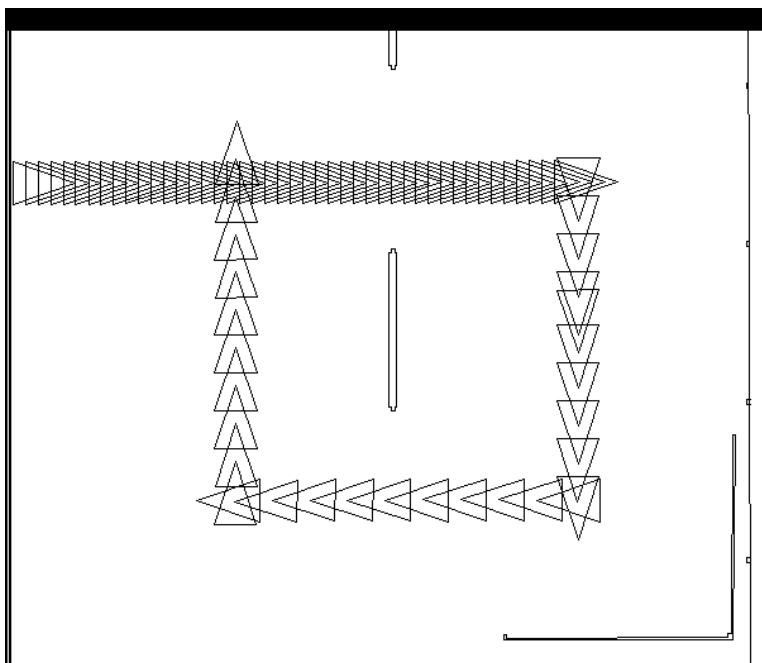
Figure 6.1: A motion sequence around an office partition. Each office partition has corners at each end that are strong acoustic targets.

real sonar data obtained during a carefully staged run through two office partitions that form a much wider opening.[2] (See Figures 6.1 to 6.3.) The office partitions have concave vertical dihedrals at each end that make good acoustic targets. Almost all doorways we have encountered possess similar corners. Going through a doorway with sonar can be achieved by looking for, finding, and then tracking these corner targets. Figure 6.3 shows the result of off-line processing of this data to achieve just this behavior. Two RCDs from the initial scan of the sequence were selected by hand, and were automatically tracked through subsequent scans to yield Figure 6.3. This sequence shows the potential information that can be obtained by tracking corner targets.

6.3   Orienteering

The sport of orienteering, established in thirty countries but perhaps not very widely known, offers a useful analogy to the mobile robot navigation problem. Like cross-country running, orienteering is a race against other

---

[2] Jean-Michel Valade of Matra MS2I originally proposed this experiment.
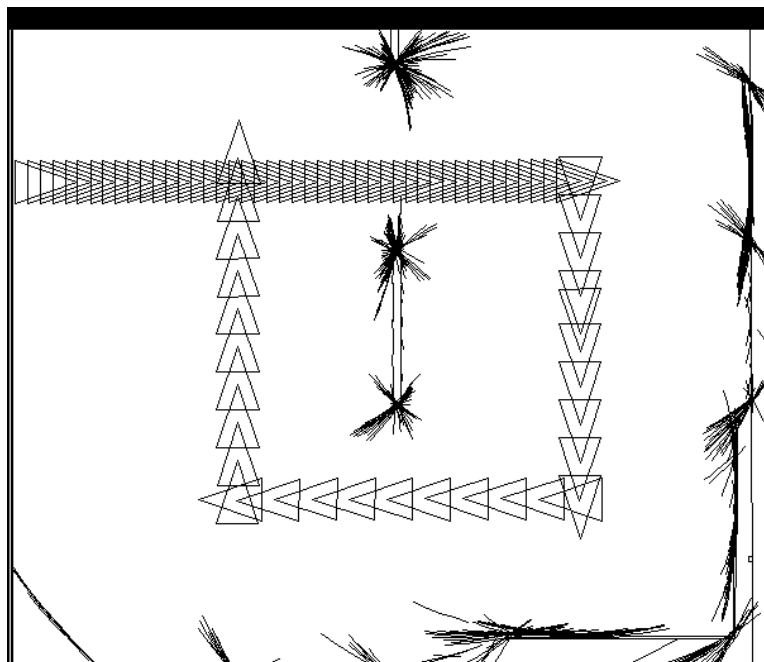
Figure 6.2: RCDs extracted from the motion sequence, with the line segment model of the lower office partition removed from this picture. Each corner in the model produces 2nd order RCDs which intersect at the corner. From vehicle locations alongside the office partition, we get 1st order RCDs which are tangent to it.
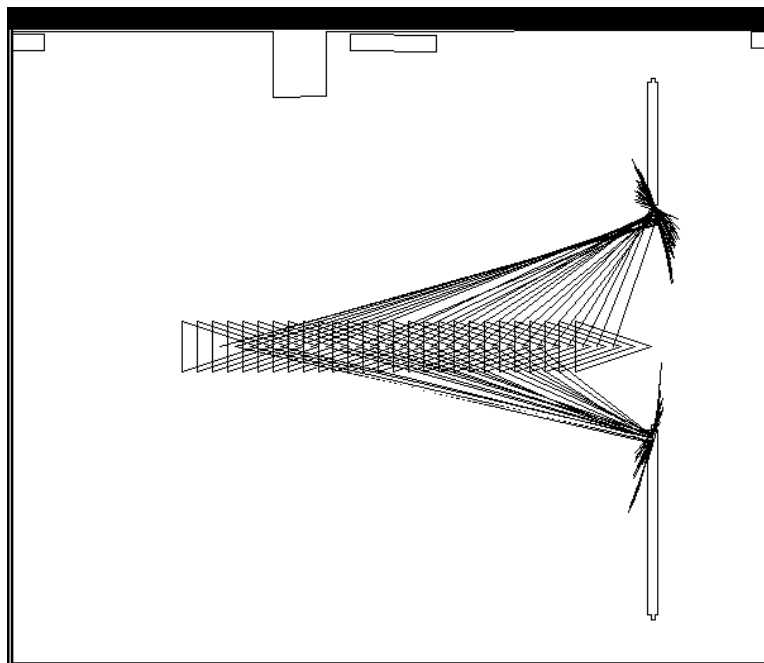
Figure 6.3: Tracked RCDs corresponding to the edges of the two office partitions. A line is drawn from the vehicle's position to the center of each RCD. Because of the asymmetry of Polaroid beam, the center of the RCD does not correspond to the corner location, and hence the lines from the vehicle to the target do not intersect in a point. We compute the location of the corner by finding the point of intersection of the circles the RCDs define.

competitors and the clock. In addition it poses the challenge of navigation. Getting lost dominates the orienteering experience for novices, who usually finish the course in more than twice the time of the winner. Fast running without careful navigation usually leads to tiredness and frustration. In orienteering, "the main doctrine to follow at every level is to think orienteering fast, not running fast" [96]. As an orienteer becomes more experienced, map using skills increase. One key skill is map-to-world prediction: given the map, what should I see? Another is world-to-map explanation: given an environment feature, such as a vegetation boundary or a re-entrant, what would it look like on the map? and where is it on the map?

Good orienteers never stop running. They are continually thinking, checking off observed features against the map and planning ahead, making route choices for future legs of the run. The first author's recent attempts at orienteering bore little resemblance to such prolific behavior, and instead a "stop, look, move" procedure was employed. The sport is a race, but the tremendous difference in skill levels (more navigation than fitness) results in a broad spread of finishing times. Fitness has little utility if you cannot navigate.

A major concern in route choice is the ease of position determination along the route. This concern has to be weighed with the distance of the route, but the former usually takes precedence for a novice orienteer. It is far better to follow a 30 percent longer route along a well-marked path than to use a compass bearing to charge straight for the goal through traversable but visually complex vegetation. The reason for this is that "where am I?" is not an issue along the path. Thinking resources can be used to plan ahead or simply rested to allow concentration on pure running speed. In contrast, straight-line navigation direct to the goal usually entails a much higher risk of getting lost.

Path planning in robotics has almost exclusively used distance as a criterion in evaluating route choice optimality. The risk of collision has been addressed, by providing increased clearance at the expense of longer paths. Criteria which have not been considered are the *cost* of determining "where am I?" along a route and the *risk* of getting lost along a route. From the standpoint of localization, some paths are easier to follow than others. For example, because the six sensors on the SKIDS vehicle are oriented parallel and perpendicular to the vehicle axis (see Appendix A), sonar feedback from planar targets is only possible when the vehicle is nearly perpendicular to the Manhattan-geometry environment.

From orienteering, we adopt the notion of *map contact*. In our framework for navigation, we define: a mobile robot has attained map contact if it has achieved correspondence between one or more current observations (observed features) and targets (map features). An observation is current for only the time step during which it was acquired, and thus $Z(k)$, the set

of observations taken at time k, can only be used to ascertain map contact at time k. Map contact for a given perception cycle means: "were one or more matches obtained during this cycle?" For fast localization, we want to build a system that maintains *continuous* map contact, by obtaining at least one match during each perception cycle. The power behind this approach is that when correspondence can be assured, by "locking on" to a target, a high bandwidth stream of easily usable information becomes available.

Another requirement of mobile robot navigation which has an analogy in orienteering is *relocation*, which is recovery from getting lost. In analogy with orienteering, we believe that once a mobile robot does not "know" where it is (loses map contact for a significant time interval), it should stop. There is no sense for a robot to wander around avoiding obstacles if it just makes the robot more and more lost. Relocation entails achieving correspondence between observations and the map without accurate *a priori* position knowledge, but search can begin from the last known position and take a rough estimate of distance traveled into account. Orienteering map features take the form of points, such as isolated trees and re-entrants, and linear features, such as streams and paths, in similar fashion to the point and line acoustic targets abundant in indoor scenes. Instantiation of a single feature will not uniquely determine position, because of the multiple occurrences of feature types in the map, often very close to one another. The many paths of an orienteering map are analogous to the many walls an acoustic indoor map would contain. Relocation only comes from the simultaneous consistent observation of several primitive features.

6.4   Related Research: Sensor Control

In sonar research, despite the prevalence of the fixed ring of sonars, some researchers have used multiple servo-mounted sonars. Gex and Campbell used *intelligent sensor moves* to direct their sonars to acquire desired information [55]. The Oxford AGV has 12 servo-mounted sonars which are repeatedly scanned back-and-forth to detect obstacles [69]. Some vehicles have multiple fixed sonars mounted on a movable head. These can be finely rotated to acquire local support, but individual sonars cannot independently track targets of interest.

Our notion of directed sensing is closely related to previous work which has fallen under the title *sensor control*. Our earlier work addressed sensor control within the context of a multi-sensor system [41]. Hager presented discrete Bayesian computational methods for active information gathering [61]. The discrete Bayesian methodology attained very high resolution, such as using a CCD camera to estimate the width and height of a book to an accuracy of tenths of a millimeter. The standard assumptions of MMSE

estimation break down completely at these levels of accuracy. Cameron developed a Bayesian approach to optimal sensor placement [25]. A decision-theoretic framework is applied to the questions "where should I look next?" and "where should I look from next?" to make object recognition as efficient as possible. Elfes has recently extended the occupancy grid concept to incorporate sensor control; the expanded representation is referred to as an inference grid [48].

The difference between directed sensing and sensor control is one of emphasis. Directed sensing focuses its efforts on maintaining correspondence. Sensor control has focused on the optimization of the subsequent estimation problem when multiple corresponding observations are available. In this sense the two can be seen as complementary. Directed sensing can provide a high bandwidth stream of correctly associated observations to decision-theoretic estimation procedures. Because we address issues of correspondence rather than refinement, we feel our use of less sophisticated MMSE estimation techniques is justified.

6.5 Summary

This chapter has been speculative; our conclusions are based largely on intuition, not hard evidence. The validation of these concepts through experimental results is part of our future research agenda. We propose an alternative methodology for sonar data acquisition based on the use of locally intelligent, controllable sensors that can track point, line and arc targets in the environment as the vehicle moves. We call these *tracking sonars*. Our aim is to use them to make localization a continual question of "am I lost yet?" instead of a now-and-then question of "where (out of all possible locations) am I?" Mobile robots will get lost, and relocation is essential, but localization via relocation is too slow for the basic perception cycle.

An inspiration for this approach comes from the sport of orienteering, both from the first author's not-so-successful attempts at the sport and interviews with experienced orienteers. The orienteering concepts of continuous map contact and tracking features such as streams, vegetation boundaries, and fences are analogous to the directed sensing competences we wish to provide our mobile robot in typical indoor environments, for example "grabbing hold" of corners to "pull" itself through doorways or using walls as "handrails". Our approach remains unimplemented at this stage, but hardware to implement such a system is currently in development at Oxford.

The challenge of perception can be thought of as a war against the correspondence problem, and a race against time. The aim of directed sonar sensing is to acquire just a few returns during each iteration of the

perception cycle, so you can win the race, but to acquire the best returns, so you can win the war.

Chapter 7

Why Use Sonar?

> *I am getting on nicely in the dark.*
> James Joyce, *Ulysses*, Ch. 3

    This monograph has investigated the problem of sonar-based navigation. By way of conclusion, this chapter assesses the potential of sonar in comparison to other sensing alternatives and discusses an agenda of future research.

7.1   Sonar *vs* the Infrared Rangefinder

In Chapter 2, we argued that many previous researchers followed a methodology suitable for an ideal ray-trace scanner, but less useful for sonar. Recent advances in sensing technology have made available optical rangefinders that do in fact approximate a ray-trace scanner. In our opinion, devices that operate on a phase measurement principle are the best. The first phase-based rangefinder was developed by AT&T Bell Laboratories in the mid-1980s [99], [33]. Bell labs do not produce a commercial version, but have licensed the technology to other companies. At the Oxford robotics group, Brownlow, Adams, and Tarassenko have recently developed a novel phase-based ranging device that may out-perform the AT&T sensor [126].

    Optical rangefinders vary in scale from laser-based systems, of the type used by Hinkel *et al.* [65], to inexpensive amplitude-based infrared devices, as used by Flynn [52] and Connell [31]. The former offer precision at high cost, while the latter cannot provide accurate range information because of their sensitivity to surface reflectance properties. The AT&T design achieves comparable performance to laser-based systems at much lower cost. The transmitted signal is modulated at 5 MHz, enabling direct range information to be obtained by measuring the phase shift between transmitted

and received waveforms. An automatic gain control amplifier overcomes variations in signal strength due to attenuation losses and changing surface reflectance properties. A useful operating range of 20 feet is obtained, with a typical range resolution of 1 inch. Because an infrared LED is used instead of a laser, potential eye safety problems are avoided. As discussed in Chapter 3, Cox has presented the successful use of the device to achieve on-the-fly localization with an *a priori* map [33].

The AT&T rangefinder can take a complete, dense scan in less than 1 second. Given this fact, sonar seems hopelessly outclassed. However, we believe that sonar's great potential lies in the fact that typical indoor environments contain surprisingly few acoustic targets. The task of achieving and maintaining correspondence between observations and map targets will be considerably easier in many environments with acoustic sensing. Interpretation paradigms are possible in which single, isolated returns can be used to update directly the vehicle position.

The limitations imposed by physics must be distinguished from the limitations of the Polaroid ranging system design criteria. Sonar's rate of data acquisition is limited by the speed of sound, which is 343.2 meters per second at 20 degrees Celsius. Through a policy of directed sensing, it should be possible to obtain and process 100 returns per second in tracking a target 1 meter away. Very high vehicle position update rates should be possible if directed sensing strategies are combined with a faster firing capability. To prevent interference among sensors, the firing of multiple sensors must be coordinated, but careful engineering should be able to meet this requirement. Frequency modulation and amplitude-based interpretation present the opportunity for higher resolution [58]. However, we feel the simple time-of-flight system has not been pushed to its resolution limit. Carefully engineered hardware could completely eliminate the effects of weak returns and assure constant target visibility over all ranges.

We believe sonar is a much better sensor for position estimation than its reputation would suggest. The experimental evidence given in the previous chapters supports this conclusion. Conversely, the straightforward use of a ring of transducers to form a "sonar bumper" is fraught with difficulty, though Borenstein and Koren's success shows it can be done. Their use of very rapid sensor firing rates agrees in spirit with the concept of continuous map contact described in the previous chapter. Dynamic operation will require interpretation strategies that can accommodate fast sensor udpate rates. Kuc is the only researcher who, to our knowledge, has made the claim of 100 percent reliable *collision prevention*, using a good model of edge visibility and conservative path planning. We propose the following challenging and conclusive test of sonar obstacle avoidance: navigation through a field of randomly placed vertical planes of glass. (Perhaps plexiglas would be a wise substitute for real glass?)

7.2 Sonar *vs* Vision

Of course, for many tasks computer vision will be necessary. With what other sensing modality could a robot hope to confront the typical graduate student's bedroom after the laundry has been neglected for a few weeks? Vision is by far our most powerful sense. Forget for a moment, however, the miraculous data acquisition system through which you are reading this sentence, and reconsider the typical CCD camera as a *sensor* that produces two hundred and sixty-two thousand, one hundred and forty-four 8-bit numbers every 40 milliseconds or so. From the more than **seven million** numbers that flow from this fire-hydrant each second, to compute the robot's position all we really want is a drop of water—the three numbers x, y, and θ. In a world that is "awash with photons" [107], it is impossible to predict, or usefully process, any of these 8-bit numbers in isolation; the information is embedded in a *huge* array. The existence proof of human vision certainly tells us that a wealth of information is there to be grasped, but how can the useful information be extracted from the array in timely fashion? As a whole research field is dedicated to answering this fundamental and far-reaching question, we must send the reader elsewhere to ponder the issue further.

With the luxury of an active sensor, the sonar-based localization system described in Chapter 3 uses, on average, **three** 12-bit numbers to update the robot's position each time it moves.

To use sonar to its potential, one has to learn to look at the world differently—to think like a sonar sensor. Under the influence of Figure 2.1 and many other figures like it, the first author gradually adopted this alternative outlook through the course of 1989. For example, the realization that we needed to add cylinders to the representation came one summer evening during a (human) run around the Grand Canal of Versailles. Over the next few days, another eighty or so scans were acquired in the SKIDS room, thirty-four of which were used to form Figures 4.6 and 4.7. While there can be no substitute for carefully acquiring dozens of sonar scans in a variety of environments, the avid reader may wish to consult the first five paragraphs of Chapter 3 of James Joyce's *Ulysses* for additional inspiration in this regard.

7.3 Sensor Fusion

The reader may have questioned the competitive tone of the previous two sections—since different sensors have different strengths and weaknesses, why not combine them? This appealing concept has received considerable attention in the literature [41], and could fill another book on its own. For our purpose here, we shall consider just two issues: the difference in physics

between modalities and the virtue of self-reliance.

Consider two mobile robots, one equipped exclusively with infrared range sensing, the other exclusively with sonar. Because these modalities rely on different physics, the maps that each robot would need for navigation would not be the same; the characteristics of the environment that need to be made explicit in each representation would be different. For example, suppose the environment had two walls, of similar dimensions, one glass, the other plasterboard. These features would appear vastly different to each modality, but relying exclusively on geometry, would have an identical representation in the map. Individual sensor observations could not be predicted for each modality from a single, exclusively geometric map. These considerations can lead one to question the "adequacy of geometry as a common currency for sensing" [16]. Can it bridge the difference in physics between modalities?

One sensor fusion paradigm that has received considerable interest is the combination of infrared and sonar range sensing on a single vehicle [52]. Based on the argument above, we believe that the tempting proposition of fusing sonar and infrared range data as the first step in interpretation should be resisted. For example, we have stated in the past that sonar is "bad at finding corners", while infrared is "bad at measuring range", but now realize this is misleading. It may sometimes be hard to see the corners in a sonar scan, but a sonar transducer produces measurements of the *distance to corners* quite reliably. We feel that a system that operates these different modalities in parallel, each with its own representation, would provide the most capable robot in the field. In pursuit of this goal, we should push the capabilities of each sensing modality, operating in isolation, to its limit. Competition between sensing modalities in the research lab should someday yield a more robust multisensor product on the factory floor or in the office.

7.4   Future Research

The plant and measurement models for our Kalman filters would benefit greatly from increased sophistication. The vehicle motion error model, illustrated by Figure 1.1, attempts nothing more than "each time the vehicle moves, it gets slightly more lost." The method of Muir's thesis cries out for application here, to develop a true dynamic model [105]. Improved sonar error modeling would require another painstaking round of data acquisition, but in our opinion should be postponed until the sonar electronics have been completely redesigned.

As discussed previously, we use the term relocation to refer to the process of determining position without the assistance of an *a priori* position estimate to help in achieving correspondence. This ability would be necessary as part of a general navigation capability for error recovery

and initialization. The standard work of this nature using sonar was performed by Drumheller, who used line segment features in sonar data for matching with a hand-measured line segment model [40]. Our arguments against the use of line segments as features for sonar interpretation were put forth strongly in Chapter 2. It would be fruitful in our opinion to repeat Drumheller's work using RCDs instead of line segments as features for matching within the Grimson and Lozano-Pérez interpretation tree framework [60]. We would like to link this with the map building capability we have attained to perform matching with autonomously learned room models.

On the theoretical front, a rigorous treatment of the problem of data association, so prevalent in robotics [36], may require a whole new set of tools:

> The problem of multitarget tracking is in general, given a *cumulative collection of data sets* ... (i) to compute the *a posteriori* probability of [the number of] detected targets... and (ii) to obtain the *a posteriori* probabilistic distributions of target states. This problem, particularly the second one, differs radically from traditional estimation problems because the basic objects in this problem, i.e. *targets* and *data sets*, are not random vectors with fixed dimensions, but are actually *random sets*. ... the only known *theoretically consistent* way of expressing the above two *a posteriori* probabilistic assessments is through *tracks* and *data-to-data association hypotheses* as defined below.
>
> ...
>
> It is clear from [these equations] that this theory of multitarget tracking is actually an extension of general *state estimation* or *filtering* theory. [27]

Can the theory of multitarget tracking offered by Chang, Mori, and Chong be extended from the domain of swift point targets far off in the sky to the everyday realm of three-dimensional shape [76] so close at hand?

Experimentally, our ultimate objective is a complete, hands-off implementation of navigation that achieves the multiple objectives of localization, obstacle avoidance, and map building and maintenance. We would like to place a robot in an unknown room, tell it "go", and then come back a few hours later to find the robot had completely mapped the room and was ready to execute commanded trajectories quickly. We believe that such a system can be implemented exclusively with today's sonar hardware. The current version of our software contains most of the code that would be needed to do this, but the difficult task of simultaneous map building and localization remains unimplemented.

As a first step, we want to build a very boring robot—one that could efficiently determine its position using autonomously learned maps, and could use this capability to navigate for hours and hours in unknown, static,

unpeopled environments. We believe that this competence can be implemented with the standard Polaroid transducer. The solution can then be extended to more challenging dynamic environments.

7.5   Contributions

In summary, this research has made the following contributions:

- A sensor model that provides the ability to predict and interpret real Polaroid sonar data in typical indoor environments has been presented.

- A novel model-based localization system has been implemented on several different robots, and its performance has been analyzed.

- A map building algorithm based on the bottom-up interpretation of RCDs has been developed, and its successful implementation with real sonar data has been presented.

- A unified approach to the sonar-based navigation problem has been presented that combines the multiple problems of localization, obstacle detection, and map building in a common multitarget tracking framework.

- Directed sensing strategies to permit practical sonar-based navigation have been described. Our localization and map building results provide evidence that navigation with the exclusive use of sonar is possible, but perhaps infeasibly slow. Our belief is that the application of directed sensing strategies can make sonar-based navigation practical.

Appendix A

Hardware and Software

A.1 Mobile Robots

Our research has made use of two different versions of the Robuter mobile robot, manufactured by the French company Robosoft. One version of the robot, unimaginatively referred to in this monograph as the SKIDS vehicle, resides in the experimentation room of Matra MS2I in France. The other vehicle, the latest addition to the Oxford AGV laboratory, will be called the Oxford Robuter. Both Robuters have two on-board microprocessors: a 68020 vehicle controller and a 68000 dedicated to control of the sonar data bus. The primary difference between the two vehicles is that the SKIDS vehicle has six static sonar sensors, while the Oxford Robuter has eight. A radio link provides wireless RS232 communication for both vehicles. Dimensions and sensor placings are shown in Figure A.2. The vehicle has two rear wheels that can be independently driven and two passive casters at the front. The front casters yield unpredictable and sometimes drastic "shopping trolley" effects when the vehicle changes direction. Similar versions of the Robuter are used by many other researchers, for example Crowley [38] and Ayache and Faugeras [5].

A robot built by Martin Adams, called Eric, was used for part of our research. The vehicle control interface eventually used for the Robuter was initially developed for this vehicle, but for the results presented in Section 3.5, Eric was accurately placed by hand at each sensing location. Eric was equipped with the sonar scanning device described below.

A.2 The Polaroid Ultrasonic Ranging System

The transmission properties of air are much less favorable than for water, for which sonar was originally developed. For example, the absorption of

Figure A.1: A look inside the Oxford Robuter. This version of the Robuter has a 68020 central vehicle controller and a 68000 dedicated to the sonar system. A special purpose 4-axis drive board and power supplies are also contained in the computing rack. The eight sonars are connected via a local network, with modified Polaroid driver boards located near each transducer.

sound in air at 80 khz is 2.9 dB/m compared with 0.02 dB/m for water [58]. The attenuation of ultrasound in air increases with frequency, and is a function of temperature and humidity. Maslin characterizes this loss by an attenuation constant A_0 in dB/foot [91]. At 50 kHz he reports values for A_0 in the range of 0.2 to 0.6 dB/foot for variations in temperature from 17 to 28 degrees Celsius and variations in relative humidity from 15% to 70%. The speed of sound in air is temperature-dependent, and characterized by the relationship

$$c = 331.4 \ \sqrt{T/273} \quad m/sec \qquad (A.1)$$

where T is the ambient temperature in degrees Kelvin.

A.2.1   The Circular Piston Model

Electrostatic transducers of the Polaroid type are usually modeled as a plane circular piston in an infinite baffle [103], [91], yielding a radiation
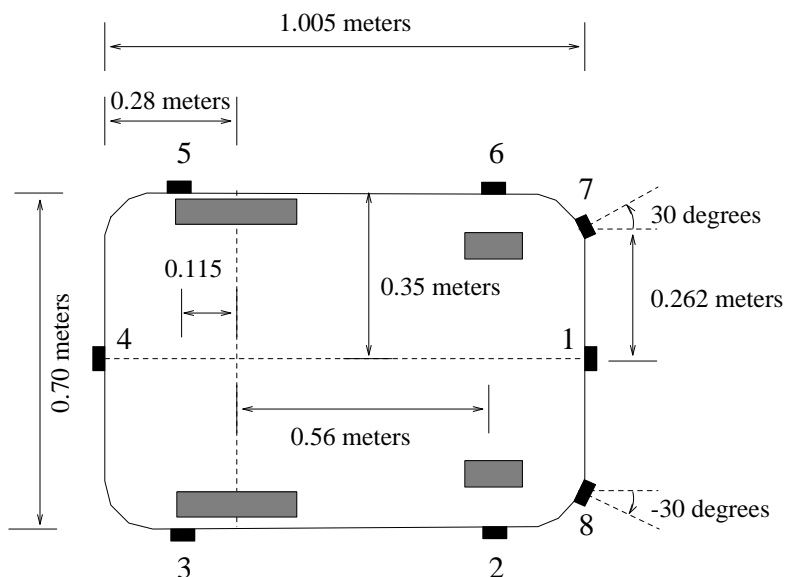
Sensor	x'_s	y'_s	α'_s
1	0.725	0.000	$0°$
2	0.560	-0.350	$-90°$
3	-0.115	-0.350	$-90°$
4	-0.280	0.000	$180°$
5	-0.115	0.350	$90°$
6	0.560	0.350	$90°$
7	0.713	0.262	$30°$
8	0.713	-0.262	$-30°$

Figure A.2: Dimensions and sensor locations for the Robuter. The vehicle has two driven rear wheels which are 24 centimeters in diameter, and two passive front casters which are 12 centimeters in diameter. The six sonars on the SKIDS vehicle are each fired individually. Sensors 3 and 7 and sensors 6 and 8 are fired simultaneously on OxRob.

characteristic function of:

$$P(\theta) = \frac{2J_1(k\ a\ \sin(\theta))}{(k\ a\ \sin(\theta))} \qquad (A.2)$$

where $k = 2\pi/\lambda$ is the wave number, a is the transducer radius, and θ is the azimuth angle measured with respect to the transducer axis. For the Polaroid transducer, we use the values $a = 19$ mm and $\lambda = 6.95$mm.[1] The circular piston beam pattern is plotted for these values in Figure A.3. Equation A.2 can be solved to yield the -3dB point of the radiation pattern by solving

$$\theta = \sin^{-1}(1.62/ka) = 5.41°. \qquad (A.3)$$

This yields a 3dB full beam width of about 11 degrees, but we believe that this value is not very meaningful for interpreting Polaroid sonar data. As discussed in Chapter 2, the crucial question is "over what range of angles is a target visible?" The answer depends on the characteristics of the transmitted pulse, target surface, receiver amplifier, and thresholding circuit. For example, the figures of Chapter 2 illustrate that for planar targets, side-lobe energy more than 20 dB lower than the on-axis radiation level can generate TOF readings for the standard Polaroid system.

From our experiences with real sonar data, we have found that the radiation patterns of Polaroid transducers are not symmetric, and vary from transducer to transducer. These effects are more significant for the side-lobes. We have not yet had the opportunity to quantify these conclusions. As discussed earlier, Lang *et al.* have presented detailed experimental evidence to characterize the Polaroid sensor [85]. We feel more work of this nature needs to be done; for example, an analysis of variations between transducers seems essential to properly characterize a multiple sonar system.

A.2.2   Sonar Device Electronics

In the early stages of our research, we constructed a simple sonar scanning device using the single-frequency driver board supplied by Polaroid. The waveform transmitted by this driver circuit is 56 cycles of a 49.4 kHz square wave, yielding a total pulse duration of 1.13 milliseconds. All of the densely sampled sonar scans used in this book were obtained with this device. Our sonar scanner uses three different transducers to take a complete scan, an approach that was taken because of the limited angular rotation of the servo-motor that was available. Variations in target responses from one transducer to another were observed, but due to our RCD-based methodology, this had little effect on our sonar interpretation results.

---

[1]This wavelength is calculated assuming a frequency of 49.4 kHz and a temperature of 20 degrees Celsius (speed of sound of 343.2 m/s)
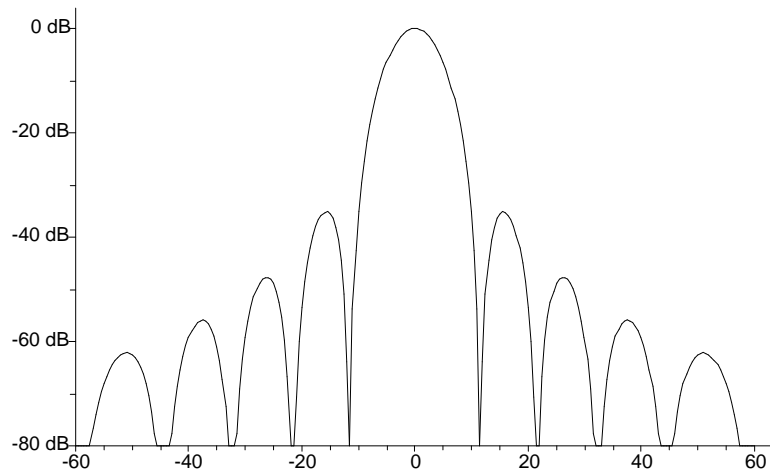
Figure A.3: Radiated sound intensity pattern for the Polaroid transducer, as predicted by the plane circular piston model of equation A.2. The x axis shows azimuth with respect to the transducer axis, in degrees. The y axis shows acoustic intensity in dB relative to the on-axis level.

The analysis of the preceding chapters has implicitly assumed zero radial offset from the motor axis of rotation to the transducer. In fact, the scanning device had a radial offset of about 2.5 centimeters. This adds a small distance to the range to a target at high angles of incidence, which we feel is safe to ignore for our hardware. For example, at an angle of incidence of 15 degrees to a planar target, just under 1 millimeter is added to the sensor range. The effect is more significant for larger radial offsets. For example, the 30 centimeter radial offset of typical sonar rings would add about 1 centimeter. Another impact of a large radial offset is decreased point target visibility angles; this can explain some differences between scanner and ring data which are apparent in the literature.

The sonar system supplied with the Robuter is equipped with two different pulse transmission modes: a long-range mode, essentially identical to the scanning device we constructed, and a short-range mode, which transmits a short pulse comprised of 8 cycles of the 49.4 kHz signal, for a signal duration of 160 μSec. As discussed in Chapter 2, the short-range mode provides greater range accuracy because weak returns are effectively eliminated. For this reason, the short-range sensing mode has been used exclusively in our work with the Robuter.

Note: as of this writing, the latest release of the Polaroid ranging system uses a transmitted pulse that is 16 cycles (320μSec) in duration.[2]

A.3   Software

The experiments described in the preceding chapters have been implemented via a single C program running on SUN-3 and SUN-4 workstations, using SUNVIEW graphics facilities. The final version of the program contained about 30,000 lines of C code. A suite of Kalman filtering routines written by Chris Brown of the University of Rochester was used extensively [21].

---

[2]Thanks to Ian Traherne for this information.

Bibliography

[1] J. Alexander and J. Maddocks. On the kinematics of wheeled mobile robots. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[2] J. Anaya-Velayos, J. Martin-Abreu, and C. Yusta. A comparative study of several approaches for high accuracy range measurements using ultrasonic sensors. In *Int. Workshop on Sensorial Integration for Industrial Robots*, pages 343–345, Zaragoza, Spain, 1989.

[3] R. L. Andersson. *A Robot Ping-Pong Player*. MIT Press, 1988.

[4] N. Ayache and O. Faugeras. Building, registrating, and fusing noisy visual maps. *Int. J. Robotics Research*, 7(6):45–65, 1988.

[5] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robotics and Automation*, 5(6):804–819, 1989.

[6] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, 1982.

[7] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[8] 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.

[9] M. Beckerman and E. Oblow. Treatment of systematic errors in the processing of wide-angle sonar sensor data for robotic navigation. *IEEE Trans. Robotics and Automation*, 6(2), 1990.

[10] C. Biber, S. Ellin, E. Sheck, and J. Stempeck. The Polaroid ultrasonic ranging system. In *67th Audio Engineering Society Convention*, New York, October 1980. Reprinted in Polaroid Ultrasonic Ranging System Handbook.

[11] R. M. Bolle and D. B. Cooper. On optimaly combining pieces of information, with application to estimating 3-D complex-object position from range data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(5):619–638, 1986.

[12] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journal of Robotics and Automation*, RA-4:213–218, April 1988.

[13] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. Systems, Man, and Cybernetics*, 19:1179–1189, September 1989.

[14] J. Borenstein and Y. Koren. The vector field histogram – fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288, June 1991.

[15] 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.

[16] M. Brady. Personal communication during the first author's viva.

[17] M. Brady, S. Cameron, H. Durrant-Whyte, M. Fleck, D. Forsyth, A. Noble, and I. Page. Progress towards a system that can acquire pallets and clean warehouses. In *4th International Symposium on Robotics Research*. MIT Press, 1987.

[18] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[19] R. A. Brooks. The whole iguana. In *Robotics Science*. MIT Press, 1989.

[20] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, pages 3–15, June 1990.

[21] C. Brown, H. Durrant-Whyte, J. Leonard, B. Rao, and B. Steer. Kalman filter algorithms, applications, and utilities. Technical Report OUEL-1765/89, Oxford U. Robotics Research Group, 1989.

[22] M. K. Brown. Feature extraction techniques for recognizing solid objects with an ultrasonic range sensor. *IEEE Journal of Robotics and Automation*, RA-1(4):191–205, December 1985.

[23] M. K. Brown. The extraction of curved surface features with generic range sensors. *Int. J. Robotics Research*, 5(1):3–18, 1986.

[24] M. K. Brown. On ultrasonic detection of surface features. In *IEEE International Conference on Robotics and Automation*, pages 1785–1790. IEEE, 1986.

[25] A. J. Cameron. *A Bayesian Approach to Optimal Sensor Placement.* PhD thesis, University of Oxford, 1989.

[26] J. Campbell, S. Synnott, and G. Bierman. Voyager orbit determination at Jupiter. *IEEE Trans. Automatic Control*, AC-28(3):256–268, March 1983.

[27] K. Chang, S. Mori, and C. Chong. Performance evaluation of a multiple hypothesis multitarget tracking algorithm. In *IEEE Int. Conference on Decision and Control (CDC)*, pages 2258–2263, 1990.

[28] R. Chatila. Mobile robot navigation space modeling and decisional processes. In *Third Int. Symposium Robotics Research*. MIT Press, 1985.

[29] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*. IEEE, 1985.

[30] C. Chong, S. Mori, and K. Chang. Distributed multitarget multisensor tracking. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pages 247–295. Boston: Artech House, 1990.

[31] J. Connell. *A Colony Architecture for an Artificial Creature.* PhD thesis, Massachusetts Institute of Technology, 1989.

[32] C. Connolly, J. Mundy, J. Stensrom, and D. Thompson. Matching from 3-D range models into 2-D intensity scenes. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 65–72. IEEE, 1987.

[33] I. J. Cox. Blanche – an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. Robotics and Automation*, 7(3):193–204, 1991.

[34] I. J. Cox, R. A. Boie, and D. A. Wallach. Line recognition. In *10th Int. Conf. on Pattern Recognition*, pages 639–645. IEEE, 1990.

[35] I. J. Cox and J. J. Leonard. Probabilistic data association for dynamic world modeling: A multiple hypothesis approach. In *International Conference on Advanced Robotics*, Pisa, Italy, 1991.

[36] I. J. Cox and J. J. Leonard. Unsupervised learning for mobile robot navigation using probabilistic data association. In *2nd Annual Workshop on Computational Learning Theory and Natural Learning Systems (CLNL)*, September 1991. Proceedings to be published by MIT Press.

[37] J. L. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, RA-1(1):31–41, March 1985.

[38] J. L. Crowley. World modeling and position estimation for a mobile robot using ultra-sonic ranging. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 674–681, 1989.

[39] D. D. Dickmanns. 4D-dynamic scene analysis with integral spatio-temporal models. In *4th International Symposium on Robotics Research*, pages 311–318. MIT Press, 1987.

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

[41] H. F. Durrant-Whyte. *Integration, coordination and control of multi-sensor robot systems*. Boston: Kluwer Academic Publishers, 1987.

[42] H. F. Durrant-Whyte. Sensor models and multi-sensor integration. *Int. J. Robotics Research*, 7(6):97–113, 1988.

[43] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, RA-4(1):23–31, February 1988.

[44] H. F. Durrant-Whyte, B. Y. S. Rao, and H. Hu. Toward a fully decentralized architecture for multi-sensor data-fusion. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1990.

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

[46] A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1988.

[47] A. Elfes. A tesselated probabilistic representation for spatial robot perception and navigation. In *Proc. NASA Conference on Space Telerobotics*, 1989.

[48] A. Elfes. Strategies for active robot perception using a probabilistic spatial model. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, 1990.

[49] H. Everett. A multi-element ultrasonic ranging array. *Robotics Age*, pages 13–20, July 1985.

[50] O. Faugeras. A few steps toward artificial 3-D vision. In *Robotics Science*. MIT Press, 1989.

[51] O. Faugeras, N. Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 1433, 1986.

[52] A. M. Flynn. Combining sonar and infrared sensors for mobile robot navigation. *International Journal of Robotics Research*, 7(6), 1988.

[53] A. M. Flynn and R. A. Brooks. MIT mobile robots–what's next? In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 611–617. IEEE, 1988.

[54] A. C. Gelb. *Applied Optimal Estimation*. The MIT Press, 1973.

[55] W. Gex and N. Campbell. Local free space mapping and path guidance. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 424. IEEE, 1987.

[56] G. Gilbreath and H. Everett. Path planning and collision avoidance for an indoor security robot. In *Proc. SPIE Conference on Mobile Robots III*, volume 1007, pages 19–27, Boston, MA, 1988.

[57] G. Giralt, R. Chatila, and M. Vaisset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In *First International Symposium on Robotics Research*. MIT Press, 1984.

[58] P. Gough, A. de Roos, and M. Cusdin. Continuous transmission FM sonar with one octave bandwidth and no blind time. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[59] 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).

[60] W. E. L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[61] G. Hager. *Task-directed Sensor Fusion and Planning: A Computational Approach*. Boston: Kluwer Academic Publishers, 1990.

[62] J. Hallam. *Intelligent Automatic Interpretation of Active Marine Sonar.* PhD thesis, University of Edinburgh, 1984.

[63] S. Y. Harmon. The ground surveillance robot (GSR): An autonomous vehicle designed to transit unknown terrain. *IEEE Journal of Robotics and Automation*, RA-3(3):266–279, June 1987.

[64] C. G. Harris and J. M. Pike. 3D positional integration from image sequences. In *3rd Alvey Vision Conference*, pages 233–236, 1987.

[65] R. Hinkel, T. Knieriemen, and E. von Puttkamer. Mobot-III an autonomous mobile robot for indoor applications. In *International Symposium and Exhibition on Robots*, page 489, Sydney, Australia, 1988.

[66] P. R. Hinrichs. Advanced terrain correlation techniques. In *Record of the 1976 Position Location and Navigation Symposium*, pages 89–96, 1976.

[67] P. Hoppen, T. Knieriemen, and E. von Puttkamer. Laser-radar based mapping and navigation for an autonomous mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1990.

[68] L. Hostetler and R. Andreas. Nonlinear Kalman filtering techniques for terrain-aided navigation. *IEEE Trans. Automatic Control*, 28:315–323, March 1983.

[69] H. Hu, P. J. Probert, and B. Y. S. Rao. A transputer architecture for sensor-based autonomous mobile robots. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, 1989.

[70] S. Hung. A description of the environment from the data of stereo vision assisted by a single sonar sensor. In *Int. Workshop on Sensorial Integration for Industrial Robots*, pages 294–299, Zaragoza, Spain, 1989.

[71] R. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):122–139, March 1983.

[72] R. Jarvis and J. Byrne. An automated guided vehicle with map building and path finding capabilities. In *4th International Symposium on Robotics Research*, pages 497–504. MIT Press, 1987.

[73] A. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[74] M. Kadonoff. Prolog-based world models for mobile robot navigation. In *Proc. SPIE Conference on Mobile Robots II*, volume 852, pages 305–310, 1987.

[75] M. Kayton. *Navigation: Land, Sea, Air, Space.* IEEE Press, 1989.

[76] J. J. Koenderink. *Solid Shape.* MIT Press, 1990.

[77] D. Kriegman, E. Triendl, and T. Binford. A mobile robot: Sensing, planning, and locomotion. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 402. IEEE, 1987.

[78] D. Kriegman, E. Triendl, and T. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. Robotics and Automation*, 5(6), December 1989.

[79] R. Krishnamurthy. Helpmate: A mobile robot for transport applications. In *Proc. SPIE Conference on Mobile Robots III*, volume 1007, Boston, MA, 1988.

[80] E. Krotkov. Mobile robot localization using a single image. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 978, 1987.

[81] R. Kuc. A spatial sampling criterion for sonar obstacle detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(7):686–690, July 1990.

[82] R. Kuc and B. Barshan. Navigating vehicles through an unstructured environment with sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1422–1426, May 1989.

[83] 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.

[84] T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pages 43–83. Boston: Artech House, 1990.

[85] S. Lang, L. Korba, and A. Wong. Characterizing and modelling a sonar ring. In *Proc. SPIE Conference on Mobile Robots IV*, pages 291–304, Philadelphia, PA, 1989.

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

[87] J. J. Leonard, I. J. Cox, and H. F. Durrant-Whyte. Dynamic map building for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 89–96, 1990. Also published in *Autonomous Mobile Robots*, edited by S. Iyengar and A. Elfes, Los Alamitos, CA: IEEE Computer Society Press 1991.

[88] J. J. Leonard and H. F. Durrant-Whyte. Application of multi-target tracking to sonar-based mobile robot navigation. In *29th IEEE Int. Conference on Decision and Control*, 1990.

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

[90] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[91] G. D. Maslin. A simple ultrasonic ranging system. In *102nd Convention of the Audio Engineering Society*, Cincinatti, Ohio, May 1983. Reprinted in Polaroid Ultrasonic Ranging System Handbook.

[92] M. Mataric. Environment learning using a distributed representation. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 402, 1990.

[93] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE J. Robotics and Automation*, RA-3(3):239–248, June 1987.

[94] P. Maybeck. *Stochastic Models, Estimation, and Control, vol. 1*. Academic Press, 1979.

[95] P. J. McKerrow. Simulation of sonar echolocation. In *IARP 1st Workshop on Domestic Robots and 2nd Workshop on Medical and Healthcare Robotics*, September 1989.

[96] C. McNeill. *Orienteering: the Skills of the Game*. Crowood Press, 1989.

[97] J. M. Mendel. Computational requirements for a discrete Kalman filter. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*, pages 219–229. IEEE Press, 1985.

[98] G. Miller, R. Boie, and M. Sibilia. Active damping of ultrasonic transducers for robotic applications. In *IEEE Int. Conference on Robotics*, pages 379–383. IEEE, 1984.

[99] G. Miller and E. Wagner. An optical rangefinder for autonomous robot cart navigation. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 122–134. Springer-Verlag, 1990.

[100] H. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor Devices and Systems for Robotics*, pages 253–276. Springer-Verlag, 1989. Nato ASI Series.

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

[102] S. Mori, C. Chong, E. Tse, and R. Wishner. Tracking and classifying multiple targets without a priori identification. *IEEE Trans. on Automatic Control*, AC-31(5), May 1986.

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

[104] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, Tokyo, 1989.

[105] P. Muir. *Modeling and Control of Wheeled Mobile Robots*. PhD thesis, Carnegie Mellon University, 1988.

[106] P. Muir and C. Neuman. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. *IEEE J. Robotics and Automation*, pages 1772–1778, 1987.

[107] D. Murray and B. Buxton. *Experiments in the Machine Interpretation of Visual Motion*. MIT Press, 1990.

[108] H. Noborio, K. Kondo, and A. Noda. A good extension method of free space in an uncertain 2D workspace by using an ultrasonic sensor. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, page 665, 1990.

[109] Polaroid Corporation, Commercial Battery Division, Cambridge, MA. Ultrasonic ranging system, 1984.

[110] S. Pollard, J. Porrill, and J. Mayhew. Predictive feed-forward stereo processing. In *Alvey Vision Conference*, pages 97–102, September 1989.

[111] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, AC-24(6), Dec. 1979.

[112] K. Sarachik. Characterizing an indoor environment with a mobile robot and uncalibrated stereo. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 984, 1989.

[113] M. Skolnick. *Introduction to Radar Systems*. New York: McGraw-Hill, 1981.

[114] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56, 1987.

[115] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[116] M. Soldo. Reactive and pre-planned control in a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 1128, May 1990.

[117] H. W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7(7):63–68, July 1970.

[118] H. W. Sorenson, editor. *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[119] H. W. Sorenson, editor. Special issue on applications of Kalman filtering. *IEEE Trans. Automatic Control*, AC-28(3), 1983.

[120] B. Steer. Experiments and theory with a 0.5 ton mobile robot. In *1st Int. Symposium on Experimental Robotics*, Montreal, Canada, 1989.

[121] B. Steer. Trajectory planning for a mobile robot. *Int. J. Robotics Research*, 8(5):3–14, 1989.

[122] B. Steer. Design for navigation. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 942, May 1990.

[123] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information*. PhD thesis, Massachusetts Institute of Technology, 1988.

[124] K. Sugihara. Location of a robot using sparse visual information. In *4th International Symposium on Robotics Research*, pages 319–326. MIT Press, 1987.

[125] S. Tachi and K. Komoriya. Guide dog robot. In *2nd International Symposium on Robotics Research*, pages 333–340. MIT Press, 1985.

[126] L. Tarassenko, M. Brownlow, G. Marshall, J. Tombs, and A. Murray. Real-time autonomous robot navigation using VLSI neural networks. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3, pages 422–428. Morgan Kaufmann, 1991.

[127] R. B. Tilove. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 566, 1990.

[128] Transitions Research Corporation, 15 Great Pasture Road, Danbury, CT 06810. Commercial offer: HelpMate, 1991.

[129] F. Tsuzuki, L. Moscato, and J. Adamowski. A sonar based environmental sensory system for mobile robots. In *Int. Workshop on Sensorial Integration for Industrial Robots*, pages 228–233, Zaragoza, Spain, 1989.

[130] R. Urick. *Principles of Underwater Sound*. New York: McGraw-Hill, 1983.

[131] S. A. Walter. The sonar ring: Obstacle avoidance for a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 1574, 1987.

[132] T. H. Waterman. *Animal Navigation*. Scientific American Library, 1989.

[133] W. Wells. Visual estimation of 3-D line segments from motion–a mobile robot vision system. *IEEE Trans. Robotics and Automation*, 5(6), December 1989.

[134] D. Willner, C. B. Chang, and K. P. Dunn. Kalman filter algorithms for a multi-sensor system. In *IEEE Int. Conference on Decision and Control (CDC)*, pages 570–574, 1976.

[135] A. Zelinsky. Environment mapping with a mobile robot using sonar. In *Proceedings of the Australian Joint Artificial Intelligence Conference - AI'88*, pages 373–388, November 1988.

Index

181