



MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 00881 0917

This is to certify that the

thesis entitled

**Mobile Robot Localization Using  
Pattern Classification Techniques**

presented by

**Jonathan D. Courtney**

has been accepted towards fulfillment  
of the requirements for

M.S. degree in Computer Science

Major professor

Date 6/14/93



**LIBRARY  
Michigan State  
University**

PLACE IN RETURN BOX to remove this checkout from your record.  
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
JAN 20 2001	0 0	

MSU Is An Affirmative Action/Equal Opportunity Institution

c:\circ\datedue.pm3-p.1

MOBILE ROBOT LOCALIZATION USING PATTERN  
CLASSIFICATION TECHNIQUES

By

Jonathan D. Courtney

A THESIS

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

MASTER OF SCIENCE

Computer Science Department

1993



# ABSTRACT

## MOBILE ROBOT LOCALIZATION USING PATTERN CLASSIFICATION TECHNIQUES

By

Jonathan D. Courtney

This thesis describes a methodology for coarse position estimation of a mobile robot within an indoor setting. The approach is divided into two phases: *exploration* and *navigation*. In the exploration phase, the mobile robot is allowed to sense the environment for the purpose of mapping and thus “learning” its unknown surroundings. In the navigation phase, the robot senses the surroundings and compares this information with its learned maps for the purpose of locating itself in the workspace.

We solve the task of coarse-level mobile robot localization via pattern classification of grid-based maps of important or interesting workspace regions. Using datasets representing 10 different rooms and doorways, we estimate a 94% recognition rate of the rooms and a 98% recognition rate of the doorways. We conclude that coarse position estimation is possible through classification of grid-based maps in indoor domains.

*Now to the King  
eternal, immortal, invisible, the only God,  
be honor and glory for ever and ever. Amen.*  
– 1 Timothy 1:17



## ACKNOWLEDGMENTS

First, I want to express my deep love and gratitude to my wife Mickie. Without her prayers, encouragement, and sacrifice, my graduate school studies would not be possible. In many ways, this thesis should be hers.

I would also like to thank my advisor, Professor Anil K. Jain. He provided insight, direction, and encouragement that were crucial for this thesis. He also taught me more about professionalism and self discipline than he realizes.

Thanks as well to my committee, Professors Charles MacCluer and Abdol-Hossein Esfahanian, for their critique of my research and for their personal support. Professor George Stockman and Dr. Mihran Tüceryan were also instrumental in the birth of robotics research in our laboratory.

The work of Dr. Alberto Elfes had a significant impact upon my earliest thinking regarding mobile robotics. Additionally, the advice and insight he has offered in our personal communications has been a source of direction and motivation for me.

No doubt when I look back upon my days at Michigan State some of my fondest memories will be of the PRIP Lab and the PRIPpies that inhabit it. To all of them (faculty included): thanks for the encouragement, stimulating conversations, dinners from around the world, and regular excursions to El Azteco.

Brian Wright and Roxanne Fuller provided invaluable assistance when my electronic abilities failed me (or turned up completely absent). Thanks also goes to John Lees for his friendship and willingness to get his hands dirty in custom-tailoring ROME's operating system and hardware. National Science Foundation infrastructure grant CDA-88-06599 provided the funding for construction of our robot.

My time in graduate school has been seasoned by the love of many fellow Christian students; thanks especially to Steve Walsh, Hans Dulimarta, Tim Newman, Barb Czerny, and Jon Engelsma for their fellowship during my studies.

Finally, thanks to Dave Reynolds, Rick Hallgren, and the Bible Study: Scott and Michelle Sochay, Shawn and Stacy Dry, Todd and Susan Luter, David and Lisanne Reed, and Rick and Clarissa Ross for their support, prayers and patience during the writing of my thesis. For the record, they are easily the best friends I have ever had.

# TABLE OF CONTENTS

<b>LIST OF TABLES</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Mobile Robot Navigation . . . . .	2
1.2 Spatial Representations . . . . .	2
1.3 Localization Using Pattern Classification Techniques . . . . .	4
1.4 Thesis Outline . . . . .	6
<b>2 Mobile Robot Localization</b>	<b>7</b>
2.1 Topological Localization . . . . .	7
2.2 Coarse-level Localization . . . . .	10
2.3 Navigation Using Coarse-level Localization . . . . .	12
2.4 Related Work . . . . .	14
2.5 Research Methodology . . . . .	18
<b>3 Grid-based Maps</b>	<b>21</b>
3.1 Experimental Platform . . . . .	26
3.2 HIMM Grid Map Creation . . . . .	29
3.2.1 Ultrasonic Range Sensors . . . . .	29
3.2.2 Stereo Vision . . . . .	32
3.2.3 Lateral Motion Vision . . . . .	36
3.2.4 Forward Motion Vision . . . . .	42
3.2.5 Infrared Proximity Detectors . . . . .	51
3.2.6 Tactile Sensors . . . . .	53
3.3 Experimental Datasets . . . . .	53
<b>4 Grid Map Processing</b>	<b>56</b>
4.1 Preprocessing . . . . .	56
4.2 Grid Map Descriptions . . . . .	58



4.2.1	Fourier Descriptors . . . . .	58
4.2.2	Standard Moments . . . . .	59
4.2.3	Moment Invariants . . . . .	61
4.3	Experimental Data . . . . .	62
<b>5</b>	<b>Grid Map Classification</b>	<b>66</b>
5.1	Pattern Classification . . . . .	66
5.1.1	Bayes Classifier . . . . .	67
5.1.2	Minimum Mahalanobis Distance Classifier . . . . .	68
5.1.3	K-Nearest-Neighbor Classifier . . . . .	70
5.2	Error Estimation . . . . .	71
5.3	Feature Selection . . . . .	74
5.4	Experimental Results . . . . .	76
<b>6</b>	<b>Conclusions</b>	<b>84</b>
6.1	Contributions . . . . .	85
6.2	Conclusions . . . . .	85
6.3	Future Work . . . . .	86
<b>A</b>	<b>Experimental Datasets</b>	<b>89</b>
A.1	Room Locales . . . . .	89
A.2	Door Locales . . . . .	92
<b>B</b>	<b>ROME Cartographer</b>	<b>96</b>
B.1	Mouse Buttons . . . . .	96
B.2	Main Window . . . . .	98
B.3	Sonar Subwindow . . . . .	99
B.4	Lateral Motion Vision Subwindow . . . . .	99
B.5	IR Subwindow . . . . .	100
B.6	Camera Subwindows . . . . .	100
	<b>BIBLIOGRAPHY</b>	<b>102</b>

## LIST OF TABLES

2.1	Locale types for an example navigation system. . . . .	12
3.1	Comparison of operations on grid-based maps and intensity images. Adapted from [26]. . . . .	25
4.1	Ordering of the features in the experimental dataset. Note that the IR features are not included for the room locales. . . . .	63
5.1	Feature selection and classification error of room locales: (a) 3-NN classifier, (b) MMD classifier. . . . .	77
5.2	Confusion matrix of room locale data resulting from MMD classifier using features $\mu_{00}^V, \mu_{00}^S, \varphi_4^S, \varphi_1^S$ , and $\varphi_2^S$ . The misclassifications are un- derlined. . . . .	79
5.3	The numbering of the locales, used for both room and door confusion matrices. . . . .	79
5.4	Feature selection and classification error of door locales: (a) 3-NN classifier, (b) MMD classifier. . . . .	81
5.5	Confusion matrix of door locale data resulting from 3-NN classifier us- ing features $\varphi_7^I, \varphi_7^S, \mu_{00}^I, \varphi_2^S$ , and $\mu_{00}^V$ . The misclassifications are underlined. . . . .	83
5.6	Bootstrap error estimation for the best classifier of each dataset: (a) room locales, (b) door locales. . . . .	83



# LIST OF FIGURES

2.1	An example distinctive place. The DP is at the intersection of the dashed lines. Adapted from a figure in [40]. . . . .	9
2.2	Locating DP's using hill-climbing search. Adapted from a figure in [40].	9
2.3	An indoor domain where the locale criteria are "any region not more than 25 meters long or wide, and bounded on all sides". The shaded regions qualify as locales. . . . .	11
2.4	The workspace of a delivery robot, with "door" and "room" locales shaded. The "X" marks the initial position of the robot. . . . .	13
2.5	Specifying a local coordinate system. . . . .	13
2.6	Navigation via coarse-level localization. . . . .	14
2.7	Door and room locales on the third floor of the Engineering Building.	20
3.1	Cross section of a conical sonar signal. Any object along arc $A$ will yield a sensed distance of $d$ . Adapted from a figure in [9]. . . . .	22
3.2	Uncertainty profile for a single sonar sensor firing from $(x = 0, y = 0)$ toward $(x = 5, y = 0)$ , with a sensed distance of $d = 5$ . The bivariate Gaussian function $p(r, \theta   d) = \frac{1}{2\pi\sigma_r\sigma_\theta} \exp\left[-\frac{1}{2}\left(\frac{(r-d)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2}\right)\right]$ models sensing uncertainty in range, $r$ , and angle, $\theta$ for cell occupancy and vacancy. Plot (a) shows $p[R_{k+1}   VAC]$ , the probability density function for vacancy, and plot (b) shows $p[R_{k+1}   OCC]$ , the p.d.f. for occupancy.	23
3.3	A sonar occupancy grid map after a single sensor reading. The sensor fired from $(x = 0, y = 0)$ towards $(x = 5, y = 0)$ , with a sensed distance of $d = 5$ . . . . .	24
3.4	ROME. . . . .	26
3.5	Labmate mechanical design. Reproduced from [14]; used by permission.	27
3.6	A functional diagram of ROME. Ellipses indicate robot sensors/actuators. . . . .	28

3.7	Building a sonar histogram grid. The cell at distance $d$ and lying on the acoustic axis is incremented by 3. Other cells on the acoustic axis with distances 0 to $d$ are decremented by 1. A probability distribution is obtained by continuous and rapid sampling of the sensors while the robot is moving. (Adapted from a figure in [9] .)	30
3.8	A sonar map of the PRIP Laboratory.	31
3.9	The PRIP Laboratory, from three views.	31
3.10	The sonar map of the PRIP lab. The arrows indicate the location of the views shown in Figure 3.9. The dashed line indicates the path taken by the robot.	31
3.11	Sources of ultrasonic ranging anomalies.	32
3.12	Four sonar grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot.	33
3.13	The stereo field of view and the range profile form a “wedge” of information. The shaded region is sensed as empty, and the cells along the solid line are sensed as occupied. Adapted from [46].	34
3.14	The stereo depth profile with uncertainty band shaded.	35
3.15	A diagram of the one-dimensional camera. Adapted from a figure in [11].	37
3.16	A side-looking camera tracking a point under constant forward velocity.	38
3.17	Constructing an HIMM grid map using lateral motion vision.	39
3.18	A lateral-motion epipolar plane image. Time advances from top to bottom, and each row is a one-dimensional image captured from the side-looking camera.	40
3.19	First, middle and last images of the lateral motion sequence from which the EPI of Figure 3.18 was obtained. The horizontal lines indicate the scanlines averaged in each image. Averaging these scanlines in image (a) gives the first row of the EPI; likewise, image (c) provides the last row of the EPI.	41
3.20	The set of lateral-motion EPI traces.	41
3.21	An LMV map of the PRIP Laboratory. Cell certainty values have been mapped inversely to intensity.	42
3.22	The PRIP Laboratory, from three views.	43
3.23	The LMV map of the PRIP lab. The arrows indicate the location of the views shown in Figure 3.22. The dashed line indicates the path taken by the robot.	43

3.24	Characteristics of lateral motion vision data. LMV detects the high-contrast surface markings on wall posters and the air conditioner in the lab, but cannot sense the whiteboard. It also lacks the range precision of ultrasound. . . . .	43
3.25	Four LMV grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot. . . . .	44
3.26	A camera undergoing translation in the direction of robot motion. Points to the right of the center of expansion move to the right in the image; points to the left of $C_E$ move to the left in the image. Adapted from a figure in [11] . . . . .	45
3.27	The camera system tracking a point under slight misalignment and constant forward velocity. Adapted from a figure in [11] . . . . .	46
3.28	A forward-motion EPI. . . . .	49
3.29	First, middle and last images of the forward motion sequence from which the EPI of Figure 3.28 was obtained. Image (a) provides the first row of the EPI; likewise, image (c) provides the last row of the EPI. . . . .	50
3.30	Occupancy grid profile for an infrared proximity detector located at $(x = 0, y = 0)$ and facing toward $(x = 0.5, y = 0)$ . A bivariate Gaussian function $p(x, y   d) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left[ \frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{(y-d)^2}{\sigma_y^2} \right) \right]$ models sensing uncertainty. This profile is used for probability of occupancy or vacancy, depending on whether or not the sensor detected an object. . . . .	51
3.31	Building an HIMM grid map with infrared proximity sensors. . . . .	52
3.32	An IR map of the PRIP lab. The light gray regions indicate unoccupied sensed cells, and thus reveal the path taken by the robot through the workspace. . . . .	53
3.33	Four IR grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot. . . . .	54
3.34	Building an HIMM grid map with tactile sensors. . . . .	55
3.35	Sonar, Vision, and Infrared sensor grid maps of a locale. . . . .	55
4.1	Image processing operations on a sample grid map: (a) the original map, (b) occupied regions, (c) disconnected free space and sonar anomalies, (d) the free space of the locale, (e) morphological dilation, (f) morphological erosion. . . . .	57
4.2	Calculating $\psi(s)$ from a boundary. . . . .	59
4.3	A star plot of the class centroids for room locale maps. . . . .	64
4.4	A star plot of the class centroids for door locale maps. . . . .	65

5.1	Pair-wise feature scatter plots of room locale data. Red denotes EB 312; green, EB 323; yellow, EB 325; orange, EB 327; yellow-green, EB 330; pink, EB 335; light blue, EB 350; dark blue, EB 352; orchid, EB 354; peru, EB 360. . . . .	78
5.2	Pair-wise feature scatter plots of door locale data. Colors are assigned identically as in Figure 5.1. . . . .	82
6.1	Sliding windows around a mobile robot navigating in a hallway environment. Both the doors and the turn in the hallway could be considered generic hallway locales. . . . .	87
6.2	Coarse-level localization and its relation to other components of a robot navigation system. The arrows indicate flow of information in the logical system. . . . .	88
A.1	Example sensor grid maps of room locale EB 312. . . . .	89
A.2	Example sensor grid maps of room locales EB 323, EB 325, and EB 327. . . . .	90
A.3	Example sensor grid maps of room locales EB 330, EB 335, and EB 350. . . . .	91
A.4	Example sensor grid maps of room locales EB 352, EB 354, and EB 360. . . . .	92
A.5	Example sensor grid maps of door locale EB 312. . . . .	92
A.6	Example sensor grid maps of door locales EB 323, EB 325, and EB 327. . . . .	93
A.7	Example sensor grid maps of door locales EB 330, EB 335, and EB 350. . . . .	94
A.8	Example sensor grid maps of door locales EB 352, EB 354, and EB 360. . . . .	95
B.1	ROME Cartographer interface. . . . .	97

# CHAPTER 1

## Introduction

Robots and automated mechanisms have long fascinated both scientists and the public. The prospect of endowing machines with the ability to perform commonplace or even mundane tasks has preoccupied researchers and tinkerers for centuries. In the last twenty years, however, there has been particular interest in equipping robots with autonomous navigation capabilities. With advances in computation and sensing technologies, mobile robots are considered candidates for the routine transportation tasks performed by humans. Elfes identifies five areas in which mobile robots may soon find practical application [26]:

- Factory automation projects.
- Operation in hazardous environments, such as mine excavation, nuclear facilities inspection, and deep-sea surveying.
- Planetary and space exploration.
- Handicapper assistance.
- Military battlefield projects.

## 1.1 Mobile Robot Navigation

Unfortunately, the robotic navigation task is not an easy one. According to Leonard, the problem of navigation can be broken down into three questions [43]: “Where am I?”, “Where am I going?”, and “How should I get there?”. The first question has been termed the *localization* problem, that is, how can the navigator determine its location in the environment, given its current view of the world and what it has encountered while traveling thus far? The second and third questions are those of specifying a goal and finding an appropriate path to that goal. Robotic research in path planning and obstacle avoidance is concerned with these last two questions.

However, the localization problem is central to the task of robotic navigation. Without some idea of its current location, the navigator will be unable to follow a proper path to reach its goal. For example, consider the case of a hypothetical robot delivery vehicle. The robot must navigate through busy city streets to reach its destination while avoiding obstacles in its path (the other cars) and obeying traffic laws. But in order to do so, the robot must have accurate information of its location: its position in the traffic lane, its location in the city, whether or not it has reached the goal, etc. Furthermore, the task may be assisted by the ability to recognize landmarks or distinctive areas on the streets of interest.

## 1.2 Spatial Representations

In order for a robot to have a precise assessment of its location, some kind of internal spatial representation of the environment must be employed. These representations can be divided into three different paradigms:

- Geometric
- Sensor-based

- Topological

The geometric paradigm has been the traditional approach to spatial representation; see [42, 28, 1, 18, 48] for examples in mobile robotics. Here, low-level geometric primitives are extracted from sensor data to build up high-level abstractions of the environment. Thus, the world might be modeled using wire-frame, surface boundary, or volumetric representations, for example. This approach is appealing because it provides a concise representation that is easily manipulated and displayed by computers and is readily understood by humans. However, the geometric paradigm has drawbacks in that it leads to world models that are potentially fragile and unrepresentative of the sensor data because it relies heavily upon *a priori* environment assumptions and heuristics [26]. Elfes concludes:

As a consequence of these shortcomings, the Geometric Paradigm implicitly creates a wide gap between two informational layers: the layer that corresponds to the imprecise and limited information actually provided by the sensor data, and the layer of abstract geometric and symbolic world models manipulated by the sensing and world modelling [sic] processes. As a result, geometric approaches may operate successfully in highly structured domains, but are of limited applicability in more complex scenarios, such as those posed by mobile robots. [26]

An alternative to the geometric paradigm is the sensor-based paradigm. Here, the environment is represented as a collection of the sensor readings, obtained through sensor models. No *a priori* assumptions or models of the world are employed. Most typical of this approach are cellular representations such as Elfes' *occupancy grid* [25, 27] and other grid-based mapping schemes [10, 9, 6, 53], wherein cells in a 2- or 3-D map are iteratively updated to reflect sensor readings. These methods make weaker assumptions about the environment than in the geometric paradigm, and therefore have a wider domain of application [43]. Furthermore, they usually have a high "sensing to computation ratio" since no high-level abstractions are generated.



The geometric and sensor-based paradigms are both metric approaches, i.e., the resulting environment representation depends heavily upon quantitative sensor data. Therefore, their strengths can be greatly limited by the accuracy of the sensors employed. For example, if unaccounted for, cumulative errors in movement sensors (used by “dead reckoning” position estimation) can cause fatal inconsistencies in the world representation. To alleviate this problem, some mobile robotics researchers have resorted to a topological representation paradigm [40, 41, 4, 45, 52, 23]. This approach is often considered “qualitative” since exact distances between world landmarks are not used, but adjacency and coarse directional information is stored instead (typically in a graph-like structure). The topological paradigm is supported by evidence from cognitive science which indicates that humans may use such an approach for mental representations of space. Thus, this paradigm is considered “intuitive” to humans. Unfortunately, it does not readily lend itself to implementation in practical sensor systems which typically yield metric data.

### 1.3 Localization Using Pattern Classification Techniques

This thesis describes a paradigm wherein pattern classification techniques are applied to the task of mobile robot localization. In our approach, the robot workspace is represented as a hybrid of two spatial representation schemes: metric grid-based maps interconnected via topological relations. We feel that this maximizes the strengths of the sensor-based and topological paradigms.

Mobile robot localization is then divided into two phases: *exploration* and *navigation*. This two-part approach bears resemblance to that described by Kuipers [40] and is similar to Leonard and Durrant-White’s “weekend experiment” [43]. In this scenario, the robot is assumed to have enough time to adequately learn about its

environment before it must begin work—as if it were “turned loose” on the weekend with no *a priori* knowledge, but with several hours to wander about and explore. After this stage is completed (on Monday morning perhaps), the robot may be employed for tasks in which it uses its new-found knowledge to maneuver about.

In the exploration phase of our system, the robot is engaged in creating grid maps of the various regions (which we call *locales*) that it encounters in the workspace. This phase is essentially performed “offline” and the robot may make as many maps in as much detail as necessary. Thus, this phase corresponds to the *training* phase of pattern classification, and locales correspond to pattern *classes*.

In the navigation phase of our system, the robot localizes itself by matching its map of the current locale with the salient features of the locale maps learned during exploration. This is similar to the *testing* phase of pattern classification. Once the system has recognized the current locale, metric position information is available with respect to the local coordinate system of the locale. We call this method of localization via locale recognition *coarse-level localization*.

We pose mobile robot localization as a pattern classification problem, and then investigate the map recognition process within this context. Major questions addressed in this thesis are listed below.

- What features should be calculated from the maps for the purpose of recognition?
- What pre-processing of the maps will be necessary to extract these features?
- How many features should be calculated from a locale to adequately differentiate it from other locales?
- How many maps of each locale should be collected during the exploration phase, in order to adequately represent the locale?

- What classification method is appropriate for grid map recognition?
- How can multiple sensing sources (sonar, vision, proximity, etc.) facilitate the matching process?
- How successful are grid-based maps as a representation of the sensed environment, and how useful are they for mobile robot localization?

## 1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 discusses issues involved in mobile robot localization, introduces the concept of coarse-level localization, reviews the related literature, and outlines our research methodology.
- Chapter 3 introduces grid-based maps, describes the experimental platform, and discusses the various sensing devices and techniques used in grid map creation.
- Chapter 4 discusses image processing and feature extraction methods applicable to grid-based maps.
- Chapter 5 presents grid-map recognition techniques and results.
- Chapter 6 outlines applications, future work, contributions, and conclusions of this thesis.

# CHAPTER 2

## Mobile Robot Localization

An important subproblem in the robot navigation task is that of localization. That is, given accumulated knowledge about the world and current sensor readings, what is the position of the robot within the workspace? This would be a trivial problem if the navigation system could rely upon its velocity history (i.e., employ “dead reckoning”) to calculate its global coordinates. Unfortunately, distance and orientation measurements made from dead reckoning are highly prone to error due to shaft encoder imprecision, wheel slippage, etc. which accumulate over time. Dead reckoning thus requires supplemental information in order to periodically “flush” the metric error from the system and re-determine the robot’s position in the environment.

### 2.1 Topological Localization

Much work has been done in establishing the instantaneous absolute coordinates of the robot from its sensor data (sometimes called *continuous localization*) in the presence of metric error; see [20, 38, 42, 43] for example. Another approach is to avoid metric representations (such as geometric models or grid-based maps), and to rely upon topological descriptions of the environment instead. Such is the case with Kuipers’ “qualitative spatial learning and navigation” [40] based on his TOUR model [39].

Kuipers' work is motivated by evidence from cognitive science which indicates that humans primarily depend upon topological, rather than metric information for navigation. Here, the topological model is organized as an attributed graph, with the nodes representing landmarks (called *distinctive places* or *DP's*), and the arcs representing connecting routes (called *local travel edges*). In addition, procedural knowledge regarding how to detect distinctive places and follow local travel edges is incorporated into the model. Finally, metric information is associated with each node and arc for local geometric accuracy.

Spatial learning proceeds in two phases: *exploration* and *navigation*. In the exploration phase, the robot searches its workspace, locating DP's and travel edges, and organizing the topological model. In the navigation phase, the robot utilizes the results of the exploration phase in order to move efficiently from one DP to another along the defined local travel edges.

Kuipers' paradigm depends heavily upon the notion of a *distinctive place*. A DP is defined to be a point in the workspace that is locally distinctive according to some geometric criterion. It is identified by its *signature*, a subset of low-level features ("distinctiveness measures") which are maximized at the DP. The following examples of distinctiveness measures are given in [40]:

- Extent of distance differences to near objects.
- Extent and quality of symmetry in a region.
- Temporal discontinuity in one or more sensors, given a small movement.
- Number of directions to nearby open spaces.
- Temporal change in the number of directions to nearby open spaces, given a small movement.
- Position of extreme distance readings.

Figure 2.1 shows a DP defined from a subset of the preceding distinctiveness measures.

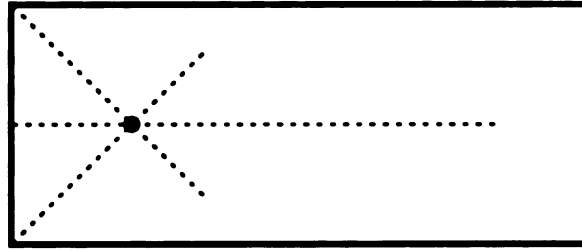


Figure 2.1. An example distinctive place. The DP is at the intersection of the dashed lines. Adapted from a figure in [40].

To locate a DP in the exploration or navigation phases, Kuipers' system must determine which distinctiveness measures are appropriate for the current region, and then maximize them using a hill-climbing search. The point at which the signature is maximized is the DP. See Figure 2.2.

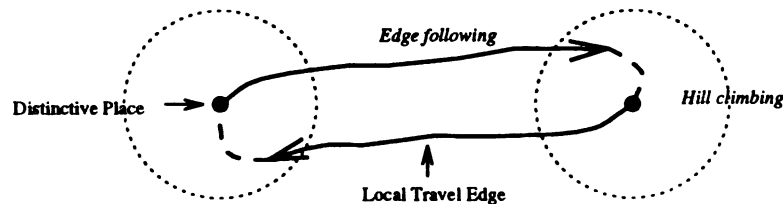


Figure 2.2. Locating DP's using hill-climbing search. Adapted from a figure in [40].

Kuipers' work in qualitative spatial learning is intuitively appealing due to the use of a topological environment representation. His simulation results seem successful, and the claim is that the system can navigate while avoiding cumulative metric error. However, the definition of a distinctive place may be too vague for a real-world implementation. First, it is questionable whether Kuipers' suggested distinctiveness

measures could be obtained from a practical system. Second, systematic sensor errors may make the distinctiveness measures depend upon the orientation of the robot. Thus, during the hill-climbing search, a hysteresis effect in the sensor data would make locating the true local maximum impossible. Finally, one cannot specify an arbitrary location in the workspace as a DP, allowing the robot to navigate to and from the site of a task, for example.

## 2.2 Coarse-level Localization

Instead of a topological model supplemented with metric data, we present an approach to robot localization that uses a metric model supplemented with topological information. In this paradigm, dead reckoning methods are used for robot position estimation within the coordinate system of a local metric map, and this local map is, in turn, topologically related to the other maps in the environment. This spatial representation is employed in what we call *coarse-level localization*.

Coarse-level localization solves the localization problem by matching the current local map (of what we call a *locale*) with the salient features of previously learned locales stored in the system database. A locale is defined to be a connected region of the robot work space that can be reliably delimited via sensing, given predefined criteria. The criteria for a locale depends upon the domain of the robot, but we limit our consideration to notions of proximity and containment. For example, in an indoor setting, appropriate criteria for a locale might be “any region not more than 25 meters long or wide, and bounded on all sides”. In this case, most rooms would qualify as locales; hallways, in general, would not (see Figure 2.3). For outdoor domains, an appropriate locale criterion might be “the region within a 10 meter radius of a local elevation maximum”.

The purpose of defining locales in this manner is to guarantee that the robot

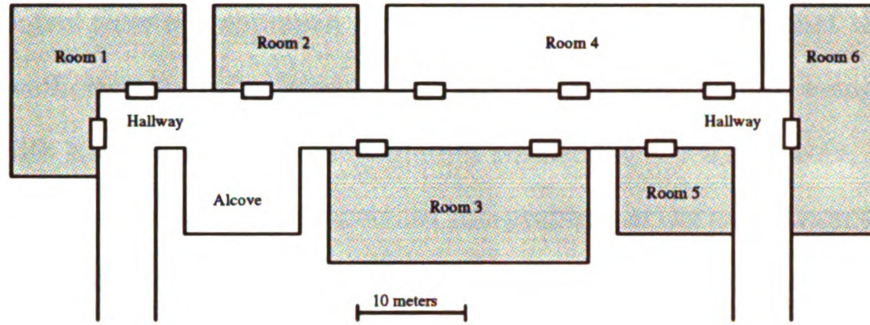


Figure 2.3. An indoor domain where the locale criteria are “any region not more than 25 meters long or wide, and bounded on all sides”. The shaded regions qualify as locales.

segments the environment into maps of identical regions during each pass through the workspace. Such an approach is not uncommon; for example, Leonard and Durrant-Whyte [42] restrict their landmarks of interest to those which can be reliably detected during repeated sensing.

Our proposed system is similar to Kuipers’ model in that the workspace is represented as several topologically related local metric maps, rather than one global metric map. Thus, locales are an analog to DP’s that represent regions, instead of points, in the workspace. As in Kuipers’ model, it is assumed that the interesting regions of the environment are contained in the set of locales and that the remaining areas can be excluded from detailed mapping.

Furthermore, dead reckoning is only employed within the context of a locale; no global coordinate system is used. Instead, the robot’s metric position is specified relative to the local coordinate frame of the locale, if needed. Fennema *et al.* [28] use a similar approach in their spatial representation scheme in order to circumvent global inaccuracies arising from dead reckoning error. Likewise, we assert that if locales are of limited size, the robot will not travel long distances within the local coordinate system, and cumulative metric errors are avoided.

There are robotic applications that require finer coordinate information and global



position control than our approach will allow. However, we believe that the proposed scheme is sufficient for many circumstances; consider that human beings generally navigate with no precise metric information whatsoever. Furthermore, coarse-level localization can be viewed as an important sub-problem in the navigation task—a first step before performing more detailed localization or locale-dependent procedures.

## 2.3 Navigation Using Coarse-level Localization

Consider, for example, a package-delivery robot that relies upon the two locale types described in Table 2.1. We will demonstrate how this paradigm can be used for navigation in an indoor setting.

Table 2.1. Locale types for an example navigation system.

Locale Type	Locale Criteria
Door	Region within 3 meter radius of the center of a doorway.
Room	Region not more than 25 meters long or wide, and bounded on all sides.

Figure 2.4 depicts the workspace of the mobile robot, where the regions that meet either of the two locale criteria are shaded. Assume that for each “door” locale a right-handed coordinate system is specified whose origin is at the midpoint of the doorway and y-axis is perpendicular to the doorway, pointing into the room (see Figure 2.5). Likewise, each “room” locale uses the coordinate system of one of its doors for reference.

Suppose the robot, at the position indicated by the “X”, is given the command to deliver its cargo to Room 3. If the robot moves to the north, it will first en-

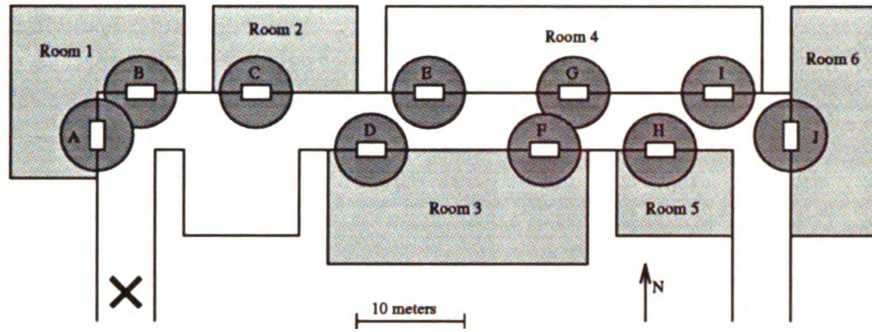


Figure 2.4. The workspace of a delivery robot, with “door” and “room” locales shaded. The “X” marks the initial position of the robot.

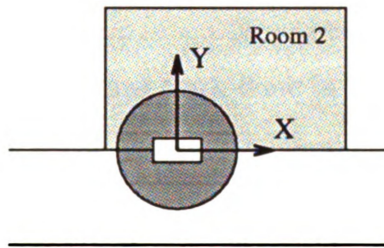


Figure 2.5. Specifying a local coordinate system.

counter and recognize the door locale corresponding to doorway A. Suppose further that the topological relations between locales provide directional information. The robot would then know that doorways D and F (of Room 3) lie to the east of its current location. Moving eastward down the hallway, the robot next encounters the locales for doorways B, C and D. Recognizing the last locale as one of the desired doorways, it enters the door and begins to map Room 3 to verify its location. Once the locale corresponding to Room 3 is identified, the robot has reached its goal, and may deposit its cargo appropriately (see Figure 2.6). To do so, it can employ the map just constructed and the local coordinate frame of the locale.

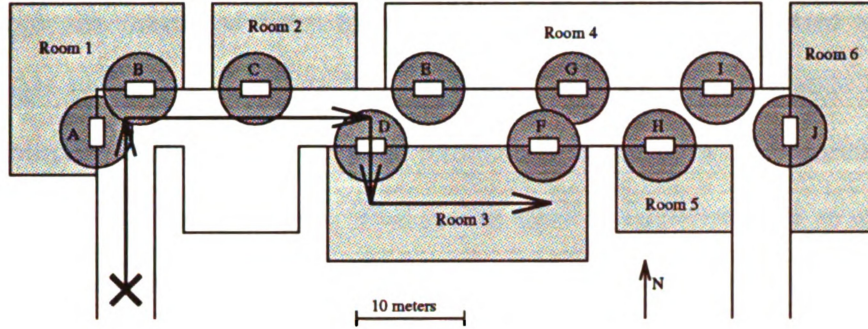


Figure 2.6. Navigation via coarse-level localization.

## 2.4 Related Work

Here we describe some of the previous work done in mobile robot localization.

Hung, *et al.* [33] perform robot localization in a complex indoor environment by pre-locating several unique marks on planar surfaces in the robot workspace such that at least one mark is visible from any location. Once a mark is detected, pattern recognition techniques are used to determine which one is being imaged, and the global location of the robot can be calculated relative to the location of that mark stored in a database. In their experiments, marks labeled with one of the characters “X”, “Y”, or “Z” were employed. Identification of a mark requires extraction of a series of shape descriptors to be used as features in a classification tree. The position of the robot relative to a mark is computed using a geometric method similar to that proposed by Fischler and Bolles [29]. The disadvantage of this approach is that it requires the addition of artificial landmarks to the workspace; these landmarks must be precisely located and cannot be moved without updating the database. Furthermore, it makes no use at all of dead reckoning measures, which can supplement the localization process.

Moravec [50, 25] developed a method to perform robot localization by multi-resolution matching of grid maps. A hierarchy of reduced resolution maps is employed;

coarser maps are produced from finer ones by converting two-by-two subarrays of cells to single cells by averaging. The matching process then proceeds from coarse to fine resolution, searching in  $(x, y, \theta)$  parameter space with two translational and one rotational degrees of freedom. The match itself is implemented as a correlation operation between two maps, and the best match obtained at a low resolution level is used to constrain the subsequent search at the next highest resolution. With this method, alignment of two spatially “close” maps of 3000 cells each takes about 1 second of VAX 11/750 computation time. However, an unconstrained search of an entire global grid map would take much longer. Note that despite the recent interest in grid-based mapping systems, Moravec’s work seems to be the only attempt to apply grid maps to the task of robot localization.

Dudek *et al.* [23] pose the robot navigation and localization task as a theoretical graph-construction problem. The environment is modeled topologically with an attributed graph structure similar to that used by Kuipers [40, 41]. It is assumed that the robot has no sensors for dead reckoning, but that it can reliably recognize locations that correspond to graph vertices and paths that correspond to graph edges. Furthermore, the robot is assumed to be equipped with multiple distinct markers that it can deposit, recognize, and pick up at any of the vertices that it encounters. The authors present and prove the correctness of an algorithm for robotic exploration, and show that the use of portable markers is necessary for this theoretical problem. The authors concede that the abilities of the robot used in their model are particularly primitive, but argue that these capabilities should be the “lowest common denominator” of robotic systems and that more advanced perceptual mechanisms may not be reliable.

Cox [17] performs continuous localization within a single room using an *a priori* map composed of straight-line segments. Data from an optical range finder is registered with the map using a matching algorithm that attempts to minimize the

total squared distance between range data points and line segments in the map. The results are then combined with the robot’s dead reckoning measurements to update the robot’s position.

Fennema *et al.* [28] describe a vision-guided mobile robot system that performs localization using an *a priori* geometric spatial model. Coincidentally, the model is very similar to the spatial representation we propose: it is composed of a hierarchically organized system of neighborhoods (also called “locales”), each with a local coordinate system. In this model, the locales are topologically related by containment; for example, a particular room is contained within a floor which is contained within a building. The location of a point within that room is given in terms of the room’s local coordinate system. Thus, localization is performed by matching visual data to geometric models of the locales to identify the local neighborhood of the robot. Then, precise position estimation can be performed relative to the locale’s local coordinate system. The authors maintain (as we do) that this topological approach avoids problems with global inaccuracies.

Drumheller [20] describes a methodology for localization within a given room using a ring of ultrasound sensors. The room of interest is modeled *a priori* as a series of straight-line segments representing walls and other obstacles. To perform localization, line segments are first extracted from the “sonar contour” generated by the ring of ultrasound sensors, and then registered to the room model using an adaptation of a two-dimensional matching algorithm proposed by Grimson and Lozano-Pérez [31]. This produces a list of “interpretations” representing possible position and orientations of the robot in the room. Each interpretation must then pass a “sonar barrier test” which ensures that individual sonar readings for an interpretation do not violate physical constraints—i.e., given a small enough incident angle, the ultrasonic signal will not appear to pass through solid objects. Finally, a robot position and orientation is chosen from the remaining interpretations that places the greatest amount of

the sonar contour in contact with the line segments of the map.

Leonard and Durrant-Whyte [42] present a method of continuous localization using the extended Kalman Filter (EKF) for a robot equipped with ultrasound sensors. The system employs an *a priori* map of “geometric beacons”—targets that can be reliably observed in successive sensor measurements and that can be represented using a concise geometric description. Examples of geometric beacons are planes, corners, and cylinders; thus, the approach is best suited for an indoor domain. The authors formulate an EKF to produce an estimate of the location of the robot based on the most recent position estimate, the navigation input, and the current beacon observations. This results in a cyclic algorithm with four steps: prediction, observation, matching, and estimation. Instead of extracting line segments from the sonar data and matching them to the environment map (as does Drumheller [20]), this approach predicts the sensor readings that the map will produce, and compares this to the observed readings. The authors contend that their method is more reliable because line segment models are not representative of sonar data.

Krotkov [38] extends the work of Sugihara [54] by developing a robot localization system that relies upon an *a priori* map of the environment and a single video image. The map marks the positions of vertically oriented parts of objects such as doors, desks, wall corners, etc. Upon acquiring an image, the system extracts the dominant vertical edges and computes the angular directions to these landmarks. The position of the robot is then determined by establishing the correspondence between landmark directions in the image and landmark locations in the map. This is done by posing the problem as an interpretation tree search; the author presents an algorithm for efficiently searching the tree while accounting for errors in the landmark directions extracted by image processing techniques. Experiments verify that the system can determine the coarse location of the robot in a given room using a map generated from architectural drawings. The advantage of this approach is that it uses structures



commonly found in indoor scenes as beacons, and that it requires no sophisticated computer vision or image processing algorithms.

Many commercial robots rely upon artificial beacons or landmarks for localization. Leonard [42] reports that the GEC-Caterpillar AGV uses a rotating laser to locate itself with respect to a set of bar codes placed at known locations in the environment. He also reports that “the TRC Corporation has developed a system for localization that uses retro-reflective strips and ceiling lights as beacons that are observed by vision and active infrared sensors.” Artificial landmarks provide a straight-forward solution to the mobile robot localization problem, but their disadvantage is that they require modification of the environment, which may not always be feasible. A system which relies instead upon techniques applicable to a wide range of environments is therefore desirable.

## 2.5 Research Methodology

Our research concerns the application of pattern classification techniques to locale recognition for the purpose of achieving coarse-level localization. We restrict our work to the solution of the “where-am-I” problem rather than navigation *per se*. Thus, the remainder of this thesis will consider the issues involved in metric map recognition within the framework of a topological spatial representation.

Due to its broad applicability, ease of sensor integration, and straight-forward implementation, we chose a grid-based mapping scheme for locale representation. Many common image processing techniques are applicable to grid maps, therefore, we find it reasonable to exploit this duality by using well-founded methods for pattern recognition in gray-scale images.

We assume that the robot localization process is divided into two phases: exploration and navigation. In the exploration phase, the robot is engaged in creating grid

maps of the locales that it encounters in its workspace, and extracting features from these maps. We allow that the robot may make as many maps as is necessary for later recognition. In the navigation phase, the robot performs coarse-level localization by comparing features extracted from its map of the current locale with representative features of locales of interest in the environment.

We simulate the exploration phase by generating sets of grid maps under multiple sensing modalities for various real-world indoor scenes. Navigation is performed manually via teleoperation, so problems with obstacle avoidance and path planning are handled by the human operator. This effectively separates the sensing task from that of navigation. By computing spatial features on the collected maps, a dataset is created wherein each pattern represents a particular map and each pattern class represents a particular locale in the workspace. This dataset is then split into *training* and *testing datasets*, and a classification system is constructed using the training dataset. Coarse-level localization in the navigation phase is simulated by classifying patterns from the testing dataset, resulting in a class designation for each pattern. The accuracy of the coarse-level localization system is given by the percentage of correct class designations returned by the classifier. Furthermore, the classifier may be aided by additional *a priori* class probability information obtained from the topological spatial model.

We consider two locale types for our experiments, as follows:

**Room** An enclosed region not more than 20 meters long or wide.

**Door** The region sensed by the robot moving from the center of a doorway 2 meters into the room.

To generate the experimental datasets, we created grid maps of several room and door locales on the third floor of the Engineering Building. Figure 2.7 shows the spatial relationship of these locales to one another—we use the notation “EB n” to indicate



“room n, Engineering Building”. There are ten locales of each type, and each locale was mapped ten times. This resulted in a dataset of 100 patterns for each locale type that we used in two separate (i.e., door and room) classifiers.

The next chapter describes the methods used to create the grid maps in our experimental data.

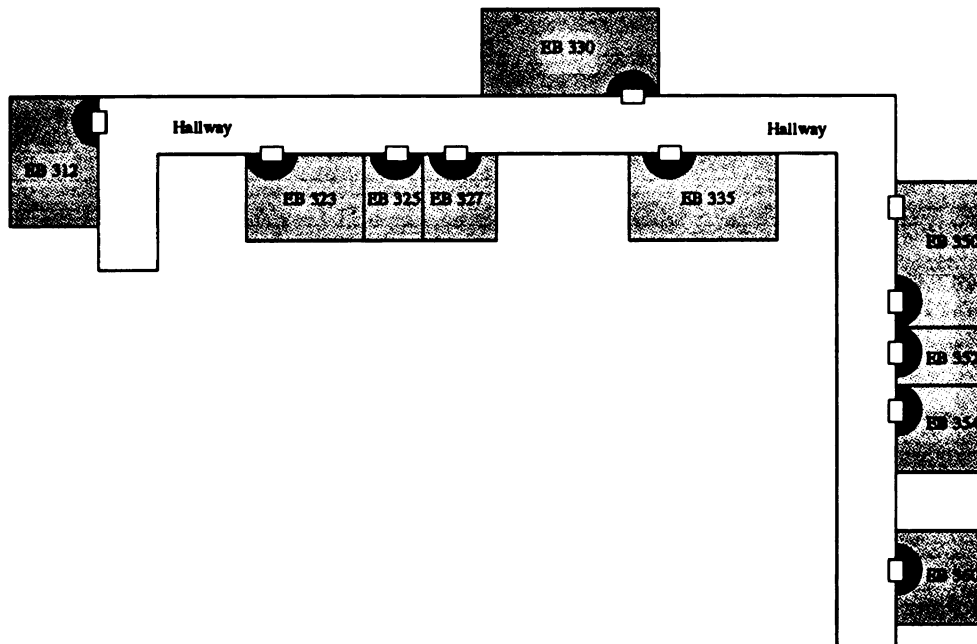


Figure 2.7. Door and room locales on the third floor of the Engineering Building.

# CHAPTER 3

## Grid-based Maps

Moravec and Elfes introduced the concept of grid-based maps, which they called the *certainty grid* [50, 25]. In the certainty grid system, the robot's three-dimensional environment is projected into a two-dimensional "floorplan" and tessellated into an array of square cells. As readings are received from the robot's sensors, these cells are updated to reflect the probability that the corresponding region of three-dimensional space is occupied by an object.

Ultrasonic (a.k.a. "sonar") sensors have often been used in implementations of the certainty grid system. A typical ultrasonic sensor using time-of-flight measurements yields precise range readings for object location, but lacks angular resolution. Figure 3.1 shows the cross section of an ultrasonic signal intercepting an object in the environment. Since the sensor uses merely the first echo it receives for range calculations, it is impossible to determine the exact angular location of the reflecting object with respect to the acoustic axis; only its radial distance can be measured. The arc labeled "A" depicts the angular uncertainty of the ultrasonic sensor—it is bounded by the width of the ultrasonic cone (typically 30 degrees ) [13].

In his *occupancy grid* system, Elfes [27, 26] accounts for this uncertainty with a Gaussian probability function that models the error of the ultrasonic sensor. Here, cells close to the sensed distance from the sensor and near to the acoustic axis have a

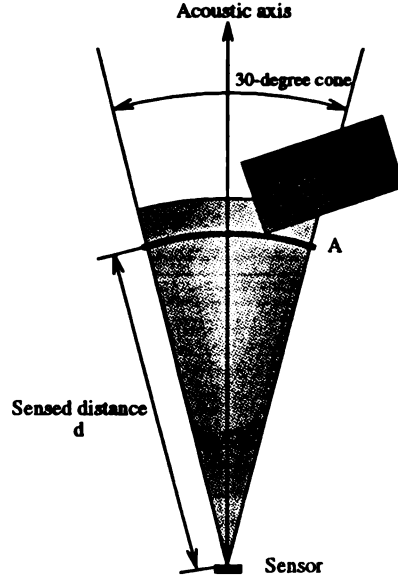


Figure 3.1. Cross section of a conical sonar signal. Any object along arc  $A$  will yield a sensed distance of  $d$ . Adapted from a figure in [9].

high likelihood of occupancy. Likewise, cells near to the ultrasound sensor and close to the acoustic axis have a high likelihood of vacancy. (See Figure 3.2.) After each sensor reading, Bayes' theorem (Equation 3.1) is applied to update the probability of occupancy of each of the cells intersecting the ultrasonic cone. A Gaussian model is used as the evidence of occupancy (OCC) or vacancy (VAC) and the current value of the cell provides the prior probability of occupancy.

$$P[OCC | R_{k+1}] = \frac{p[R_{k+1} | OCC] P[OCC | R_k]}{p[R_{k+1} | OCC] P[OCC | R_k] + p[R_{k+1} | VAC] (1 - P[OCC | R_k])} \quad (3.1)$$

Here,  $P[OCC | R_k]$  is the current value of the cell (after  $k$  sensor readings),  $P[OCC | R_{k+1}]$  will be the new cell value, and  $p[R_{k+1} | OCC]$  and  $p[R_{k+1} | VAC]$  are obtained from the sensor model. Note that  $P[VAC | R_k] = 1 - P[OCC | R_k]$ .

Initially, the occupancy status of a cell is unknown, thus we set  $P[OCC | R_0] = 0.5$  for all cells. After several readings by multiple sensors at different locations in the

environment, a map representing the objects and empty space of the environment is constructed. Figure 3.3 shows an occupancy grid after one sensor reading.

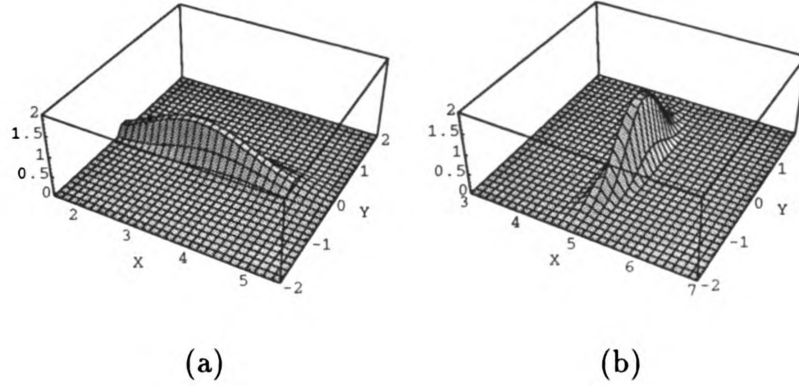


Figure 3.2. Uncertainty profile for a single sonar sensor firing from  $(x = 0, y = 0)$  toward  $(x = 5, y = 0)$ , with a sensed distance of  $d = 5$ . The bivariate Gaussian function  $p(r, \theta | d) = \frac{1}{2\pi\sigma_r\sigma_\theta} \exp \left[ -\frac{1}{2} \left( \frac{(r-d)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2} \right) \right]$  models sensing uncertainty in range,  $r$ , and angle,  $\theta$  for cell occupancy and vacancy. Plot (a) shows  $p[R_{k+1} | VAC]$ , the probability density function for vacancy, and plot (b) shows  $p[R_{k+1} | OCC]$ , the p.d.f. for occupancy.

To reduce the computational cost of updating cell probabilities, other researchers have employed different approaches to handling sensor uncertainty. Beckerman and Obloz [6] developed a system to handle the systematic errors resulting from the accumulation of ultrasonic sensor readings. Instead of a probability value, their discrete-valued grid cells contain one of four labels: Unknown, Empty, Occupied or Conflict. A logic function is used to update the cells when new sensor readings are taken.

Borenstein and Koren [10, 9] handle the sensor uncertainty by exploiting a trade-off between precise sensor models (as in Elfes' work) and frequent sensing. By rapidly and continuously sensing the environment, their "Histogramic In-Motion Mapping"

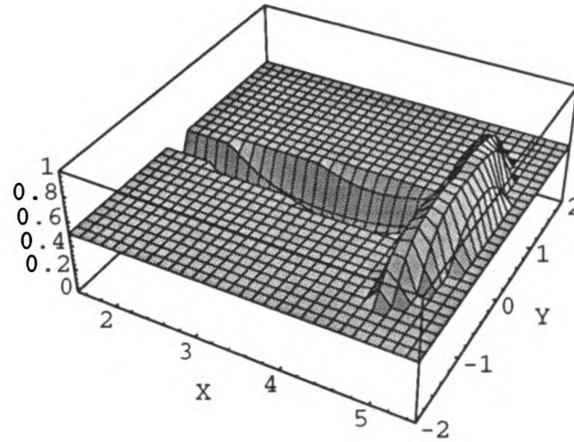


Figure 3.3. A sonar occupancy grid map after a single sensor reading. The sensor fired from  $(x = 0, y = 0)$  towards  $(x = 5, y = 0)$ , with a sensed distance of  $d = 5$ .

(HIMM) system builds a sampling histogram grid of the sonar data. Thus, the ultrasonic sensor errors are accounted as the readings accumulate. The authors contend that this approach is not an oversimplification because “a probability distribution is actually obtained by continuously and rapidly sampling each sensor while the vehicle is moving.”

Grid-based representations have proved useful in several areas of mobile robot research, including range-based mapping [49, 27], multiple-sensor integration [46, 49], path planning and navigation [10], and handling of sensor position uncertainty due to robot motion [26]. In addition, grid-based maps have been employed in many projects, including indoor and outdoor navigation, hazardous environment navigation, underwater navigation, and 3D mapping. The popularity of grid-based maps is due to several reasons:

**Intuition.** Two-dimensional world projections are familiar to humans as “floor-plans”.

**Efficiency.** Grid-based representations allow for swift creation and revision of maps as new sensor readings are taken.

**Sensor independence.** The framework is amenable to sensors other than ultrasonic range devices; scanline stereo and proximity sensors have also been used [49].

**Data abstraction.** Grid maps are a good intermediate step between raw sensor data and geometric models, providing automatic data abstraction by representing the data in a Cartesian projection.

**Correspondence to intensity images.** Elfes notes the similarity of operations on occupancy grids to those for intensity images [27, 26]. One may draw upon knowledge of image processing procedures when working with grid-based representations. A comparison of a few of these operations is shown in Table 3.1.

We chose grid-based maps as the spatial representation for locales due to these attractive features and the recent interest in this approach. In particular, we found the HIMM scheme to be a flexible, efficient method of map creation. Furthermore, we exploit the relationship between grid maps and intensity images when computing features for our classifier.

Table 3.1. Comparison of operations on grid-based maps and intensity images. Adapted from [26].

Grid Maps	Intensity Images
Labeling cells as Occupied, Empty or Unknown	Thresholding
Handling position uncertainty	Blurring / Convolution
Removing spurious spatial readings	Low-pass filtering
Map matching / Motion solving	Multi-resolution correlation
Obstacle growing for path planning	Region growing
Path-planning	Edge tracking
Labeling objects	Segmentation
Determining object boundaries	Edge detection

### 3.1 Experimental Platform

The Pattern Recognition and Image Processing (PRIP) Laboratory has a mobile robot base for research in vision-guided mobile robots. This robot, ROME (RObotic Mobile Experiment), is used for investigations in scene recognition, path planning, pose estimation, and sensor fusion (see Figure 3.4). We employ ROME as a platform to collect grid maps for our research in coarse-level robot localization.

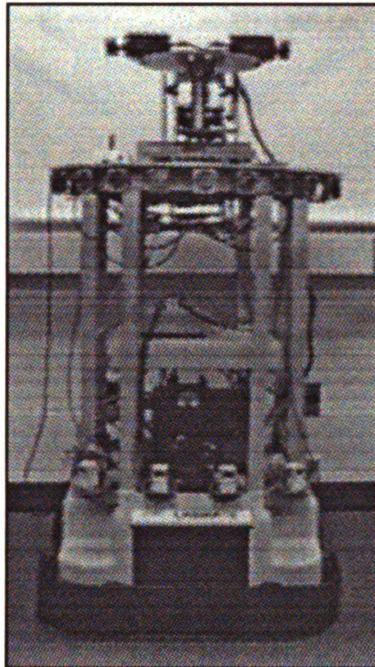


Figure 3.4. ROME.

ROME is built upon a Labmate mobile robot base from Transitions Research Corporation (TRC), and is equipped for experiments utilizing stereo vision, ultrasonic range finding, and infrared object detection. The TRC Labmate robot base is “a battery-powered vehicle . . . which can carry up to 200 pounds of computers, cameras, arms, communications gear, or other application payloads” [14]. Positional feedback

(i.e. dead reckoning) is provided by encoders mounted on the two drive wheels. The mechanical design of the Labmate base is shown in Figure 3.5.

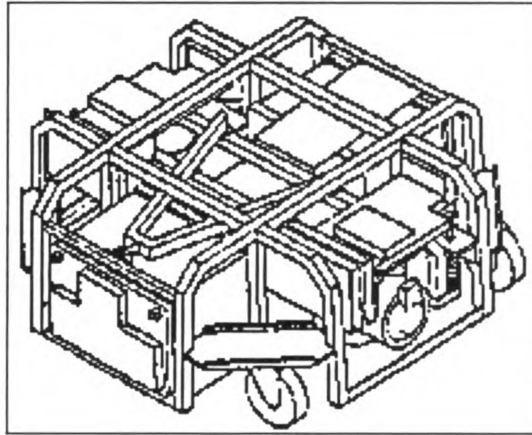


Figure 3.5. Labmate mechanical design. Reproduced from [14]; used by permission.

Our Labmate base carries a Sun SPARCstation 1 running SunOS Unix 4.1.3, a TRC Proximity Subsystem (with sonar and IR sensors), and two Panasonic color CCD cameras. Figure 3.6 is a diagram of the functional components of the mobile robot system.

The Sun workstation controls I/O to and from the Labmate, Proximity Subsystem, and cameras, and is capable of performing all image processing, mapping, and navigation calculations on-board. In addition, the robot can be operated with an ethernet communication line, thus providing remote operating and computing capability.

The TRC Proximity Subsystem consists of a horizontal ring of 24 Polaroid ultrasound transducers which measure distances up to 10 meters, and 8 infrared sensors for object detection up to 1 meter. Images from the stereo cameras are captured by two Sun VideoPix frame grabbers on board the workstation. Finally, the stereo head is computer-controlled using a Rhino Robot pan-tilt carousel.



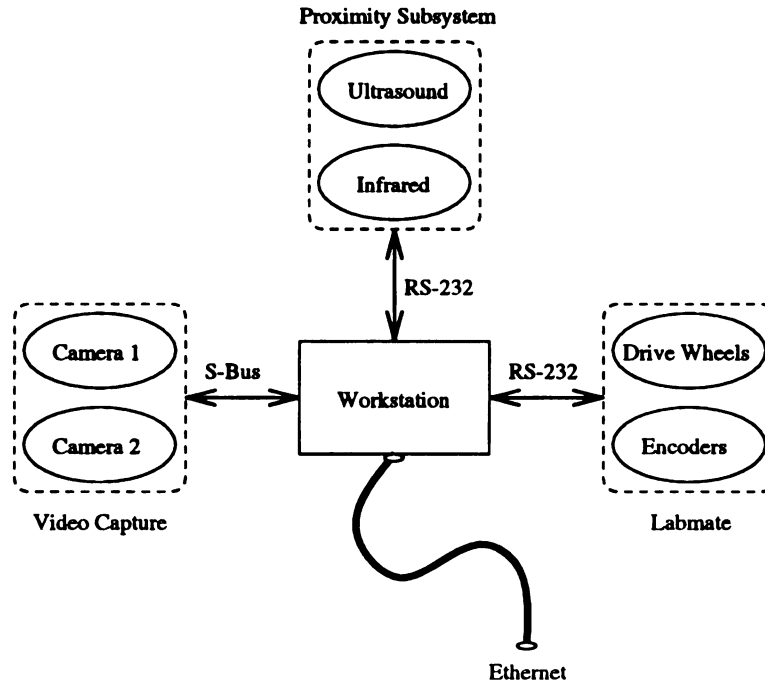


Figure 3.6. A functional diagram of ROME. Ellipses indicate robot sensors/actuators.

We have devised a computer program that performs ultrasound, vision, and infrared grid-map acquisition and remote control of ROME. The grid maps consist of 100 by 100 square cells each representing 20cm by 20cm regions in the environment. Therefore, the program can map any locale not more than 20 meters long or wide.

Navigation is operator-controlled, thereby separating the obstacle-avoidance task from that of map building. Furthermore, a graphical user interface allows the user to view the grid map creation process as the robot acquires sensor readings, as shown in Appendix B. One may extend this interface to allow for additional sensing modalities as the need arises.

## 3.2 HIMM Grid Map Creation

We use Borenstein and Koren’s HIMM scheme to gather grid maps from indoor locales for use as experimental data in our localization system. HIMM collects data on a two-dimensional Cartesian histogram grid, recording each sensor “sample” by incrementing a single cell for each reading. The result is an array of cells each containing a “certainty value” (CV) related to its probability of occupancy by an object.

HIMM was specifically developed for use with ultrasonic range data. However, Moravec claims that “the readings of almost any kind of sensor can be incorporated into a [grid map] if they can be expressed in geometric terms” [49]. We investigated the usefulness of applying sensing sources in addition to sonar to the HIMM scheme and the coarse-level localization paradigm. Here, we present some sensing modalities that are most amenable to a grid-based mapping scheme and how we incorporated them into the HIMM framework.

### 3.2.1 Ultrasonic Range Sensors

Figure 3.7 shows the application of HIMM to ultrasonic sensor data. For each sensor reading, the cell lying on the acoustic axis at sensed distance  $d$  from the robot is updated. In order to “grow” obstacles quickly, this cell is incremented by 3 (rather than 1). Furthermore, the free space of the environment is determined by decrementing (by 1) the cells between distance 0 and  $d$  and lying along the acoustic axis of the sensor—this also removes spurious readings from the grid. CV’s are constrained to be between 1 and 15; cells for which no sensor readings have been made have value 0.

### Experimental Data Collection

We implemented and tested a sonar-based HIMM scheme on ROME. The mapping results show consistency between maps of identical locales and little ill effects from

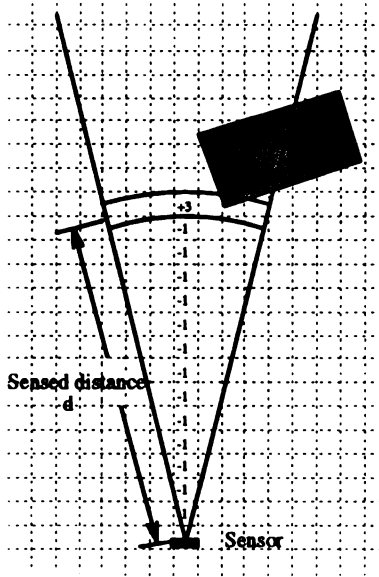


Figure 3.7. Building a sonar histogram grid. The cell at distance  $d$  and lying on the acoustic axis is incremented by 3. Other cells on the acoustic axis with distances 0 to  $d$  are decremented by 1. A probability distribution is obtained by continuous and rapid sampling of the sensors while the robot is moving. (Adapted from a figure in [9].)

dead reckoning error. Figure 3.8 shows a sonar map of the PRIP Laboratory obtained from a continuously-moving robot making a circuit around the room. (For comparison, Figure 3.9 shows three views of our laboratory, as indicated by the arrows in Figure 3.10.) Here, the CV's have been mapped inversely to intensity; thus, the darker the cell, the more likely it is to be occupied.

Ultrasonic ranging creates a few anomalies in the map that are worthy of comment (see Figure 3.11). First, instead of depicting sharp angles at the corners of the room, the corners are rounded due to lack of angular resolution in the sensor. Second, ultrasonic signals are prone to *specular reflections* if they intercept hard, smooth surfaces. A whiteboard on the wall at the bottom of the map provides such a “clean” reflection of the sonar signal that the sensor can not reliably detect an echo. Likewise, a large air conditioning (AC) unit in the corner of the room is undetectable due to

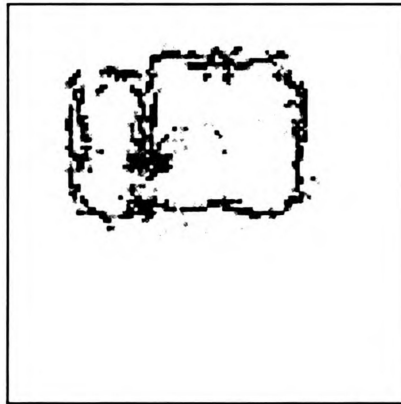


Figure 3.8. A sonar map of the PRIP Laboratory.

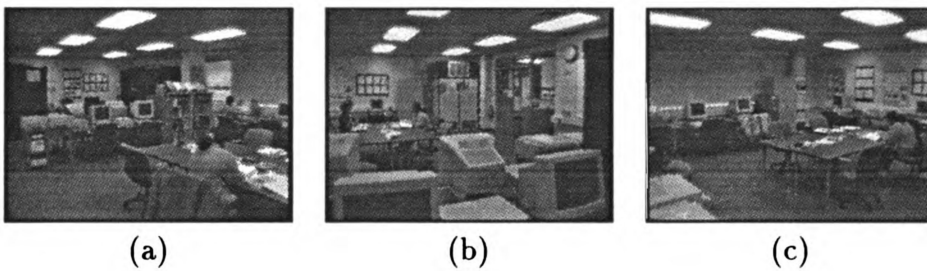


Figure 3.9. The PRIP Laboratory, from three views.

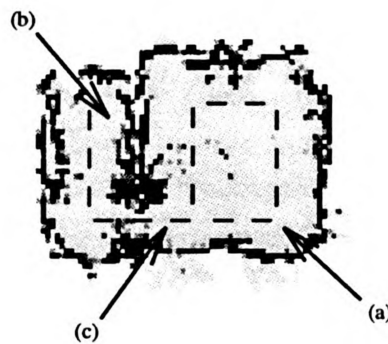


Figure 3.10. The sonar map of the PRIP lab. The arrows indicate the location of the views shown in Figure 3.9. The dashed line indicates the path taken by the robot.

its metallic surface. This accounts for the “streaks” where the ultrasonic ranging “misses” the wall. Third, the sonar sensors, mounted on the robot at a height of approximately 5 feet, tend to “miss” objects that are significantly lower in height than themselves. The noise in the middle of the right “lobe” of the map corresponds to a large conference table in the center of the room! Instead of detecting the table, most of the sonar signals pass over it without creating an echo.

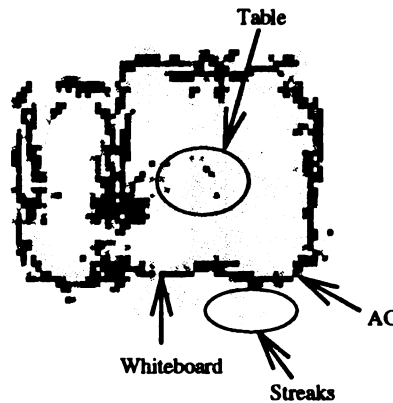


Figure 3.11. Sources of ultrasonic ranging anomalies.

Note that for our purposes, it is important that the grid maps uniquely identify each locale in the workspace, not necessarily that they precisely represent the spatial layout of each locale. Similarity between successive maps is most important—Figure 3.12 demonstrates that this has been attained using the HIMM scheme.

### 3.2.2 Stereo Vision

Matthies and Elfes [46] applied scanline stereo in combination with sonar to generate detailed occupancy grid maps. By joining the set of points found by scanline stereo with line segments, a “depth profile” is formed to approximate the surface structure of the environment. This depth profile creates a wedge (with the profile at the base

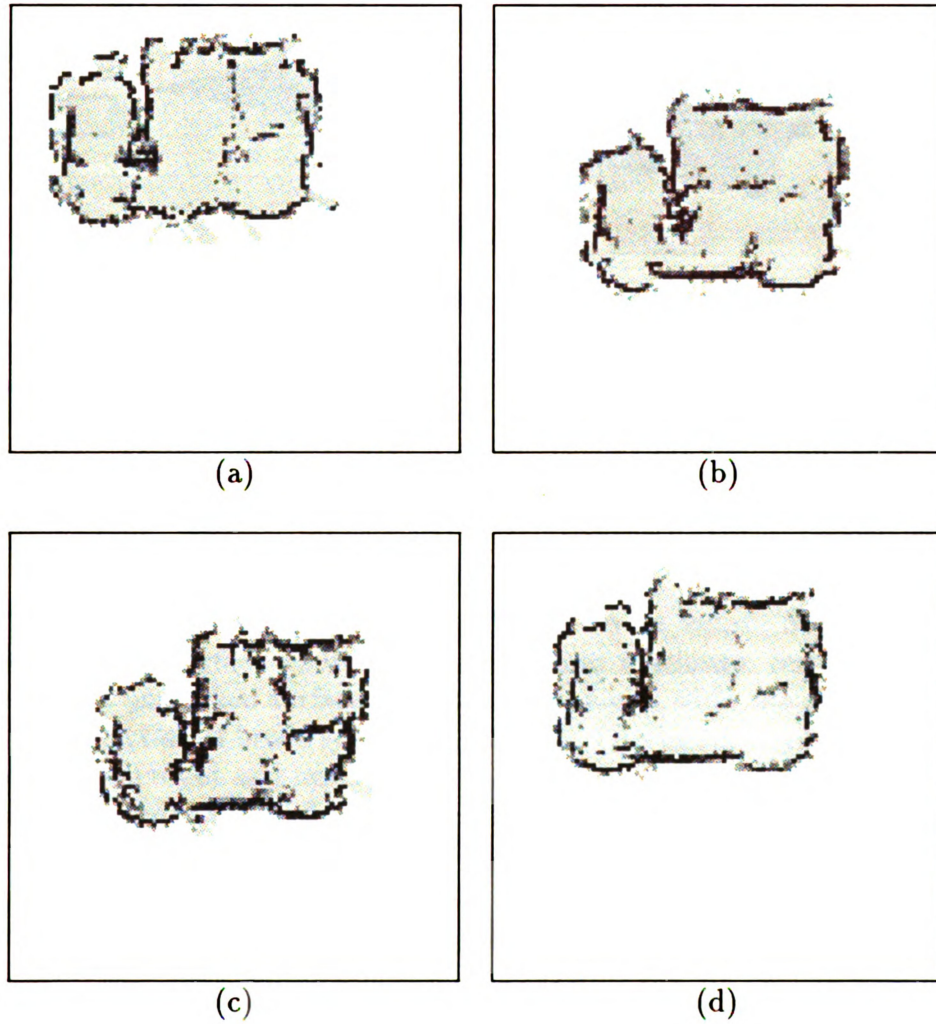


Figure 3.12. Four sonar grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot.

and the cameras at the apex) that can be used to make inferences about the occupied and empty regions of the environment (see Figure 3.13).

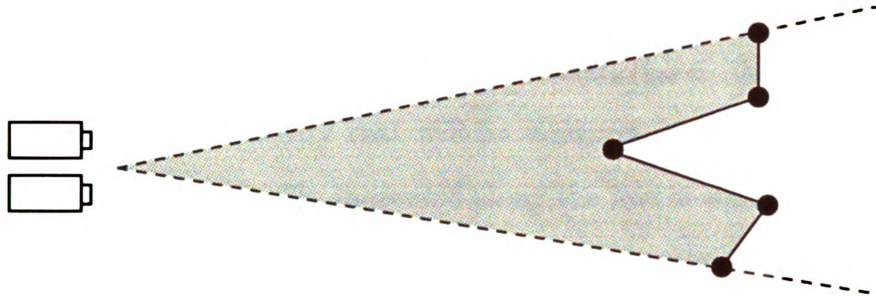


Figure 3.13. The stereo field of view and the range profile form a “wedge” of information. The shaded region is sensed as empty, and the cells along the solid line are sensed as occupied. Adapted from [46].

The authors remark:

Stereo range information is complementary to sonar. Stereo detects and localizes surface boundaries and surface markings, but misses the interior of unmarked surfaces. On the other hand, wide-angle sonar gives better detection of broad surface structure and indicates large *empty* areas between the robot and nearby objects. However, it gives less definition to surface boundaries and fine surface structure. With our current systems, stereo can see through clutter but sonar cannot.

Furthermore, stereo vision provides good angular resolution, whereas ultrasound does not. Conversely, ultrasound yields greater precision in range than does stereo vision.

To model the range uncertainty inherent to scanline stereo vision, Matthies and Elfes define an occupancy grid profile for the vision sensor, as was done for ultrasonic sensors. They note that it has been shown that for a stereo disparity of  $d$  corresponding to a depth of  $Z$ , an error of  $\delta d$  in disparity will yield (approximately) an error of  $\delta Z = Z \delta d / d$  in the depth estimate. Therefore, there is a band of uncertainty around the stereo depth profile that is wide where the profile is far from the cameras and narrow where the profile is near (see Figure 3.14). The width of this band defines the

probability that cells underlying that band are occupied according to the formula:

$$P[OCC | R] = P_{unk} + \frac{P_{occ} - P_{unk}}{N} \times \alpha$$

where  $P_{unk} = 0.5$  is the UNKNOWN probability level,  $P_{occ} \in (0.5, 1.0)$  is a constant indicating the highest probability that will be assigned, and  $N$  is the width of the uncertainty band in units of cells, measured along the line of sight from the cameras. The factor  $\alpha \in (0.0, 1.0)$  gives greater weight to the profile near to the measured points, and less weight to the profile between these points.

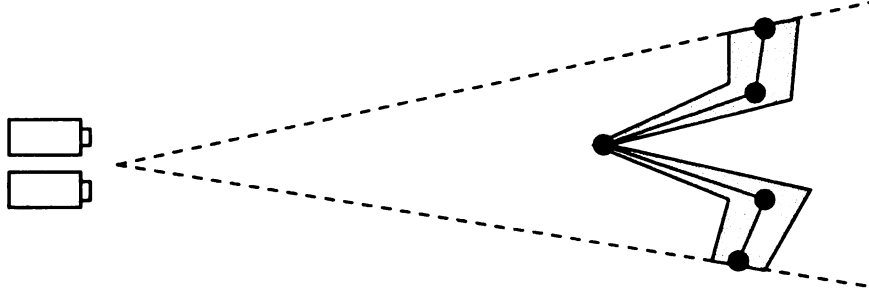


Figure 3.14. The stereo depth profile with uncertainty band shaded.

A more refined stereo error estimate was provided by Matthies and Shafer [47] that models the uncertainty of projected points as multivariate Gaussian distributions. Yet both of these uncertainty models can be accounted for in the HIMM approach: if the stereo algorithm allows for rapid in-motion sampling, the histogram cells corresponding to the depth profile are simply incremented for each reading. Likewise, the cells within the free space of the stereo wedge are decremented. After many readings, the resulting map is an array of CV's indicating the probability of occupancy by visual features.



### 3.2.3 Lateral Motion Vision

Stereo vision has drawbacks in that it requires precise camera calibration and suffers from the *correspondence problem* of matching points from two different views. As an alternative to stereo, we have investigated the application of *lateral motion vision* (LMV) to grid-based maps. Lateral motion vision provides depth information from straight-line, constant motion of a single camera. It does not require precise calibration or knowledge of camera geometries, but provides the same ultrasound-complementing features as does stereo vision. By tracking objects from a side-looking camera while the robot is in motion, LMV delivers a continuous update of range information from the environment. Because the nature of this approach depends upon constant camera motion, it is especially suited for the HIMM scheme. In addition, the correspondence problem found in stereo vision systems is greatly simplified by use of a high sampling rate—features are easier to track since the images are taken close together.

In order to apply this method to two-dimensional grid maps, the camera is assumed to be one-dimensional. One-dimensional images can be obtained from regular CCD cameras by way of extracting single scanlines, optical averaging using a plano-concave cylindrical lens, or through arithmetic averaging of scanlines. The latter two methods highlight strong vertical edges and suppress noise.

Figure 3.15 depicts the LMV camera model. The *image “plane”* is a 1D strip of  $\mathcal{P}$  pixels, numbered 0 through  $\mathcal{P} - 1$ . The *optical center* of the camera is at a distance  $f$  from the image plane;  $f$  is called the *focal ratio*. Vertical lines in the environment are projected as points onto the image plane along lines that run through the optical center. The *optical center plane* is parallel to the image plane and runs through the optical center. The *optical axis* intersects the optical center at right angles to the image plane. It is assumed that the plane formed by the optical axis and the 1D

image plane is parallel to the ground. The point of intersection of the optical axis and the image plane is called the *center of view*; it divides the image plane into segments of equal length.

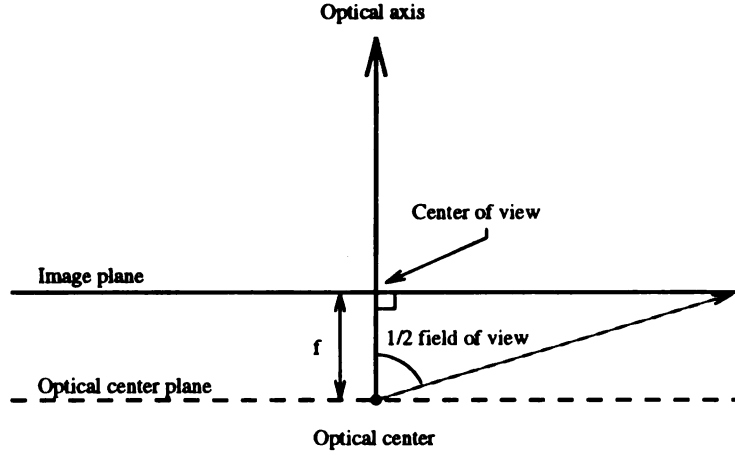


Figure 3.15. A diagram of the one-dimensional camera. Adapted from a figure in [11].

The *field of view* of the camera is  $\phi$ , the maximum angle subtended by any two visible points. The focal ratio and field of view are related by:

$$f = \frac{\mathcal{P}}{2 \tan(\phi/2)},$$

which gives  $f$  in units of pixels. For the cameras used in our experiments, the field of view is approximately  $55^\circ$ ; if each row is 256 pixels wide,  $f$  is approximately 246 pixels.

Figure 3.16 is an overhead view of a 1D camera mounted parallel to the ground and perpendicular to the direction of motion on a robot moving with constant known velocity  $v$ . Note that the motion vector, the optical axis and the image line are all coplanar. The system is tracking a point  $P$  in the environment as it moves to the left

in the image plane.

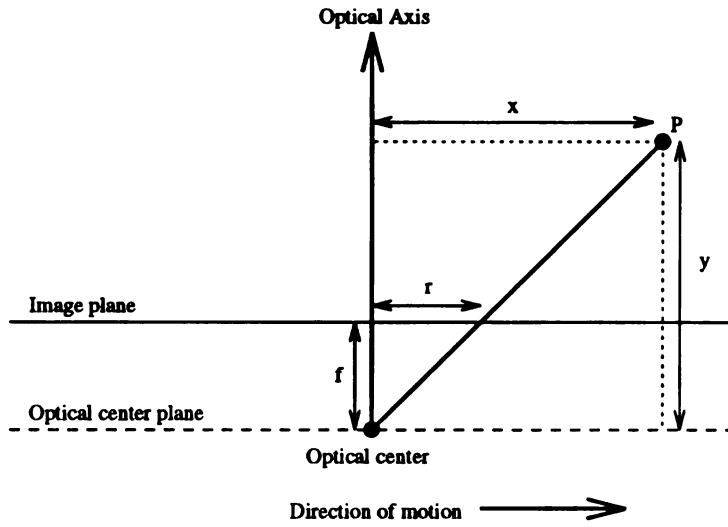


Figure 3.16. A side-looking camera tracking a point under constant forward velocity.

By similar triangles, we have

$$\frac{x}{y} = \frac{r}{f}$$

where  $y$  is the distance of  $P$  from the optical center plane,  $x$  is the distance of  $P$  from the optical axis, and  $r$  is the distance of the image of  $P$  from the optical axis. Differentiating with regard to time and solving for  $y$ , we obtain

$$\frac{1}{y} \frac{dx}{dt} = \frac{1}{f} \frac{dr}{dt}$$

$$y = \frac{f \frac{dx}{dt}}{\frac{dr}{dt}}.$$

We know that  $dx/dt$  is the velocity  $v$  of the robot, thus the distance of the point from the camera is

$$y = \frac{f v}{\frac{dr}{dt}} \quad (3.2)$$

Since  $y$ ,  $f$  and  $v$  are each constant,  $dr/dt$  must be a constant as well. By tracking a point over time and measuring  $dr/dt$ , the coordinates of environment features relative to the robot can be calculated from Equation (3.2) and

$$x = \frac{y r}{f}.$$

Since the images are sampled rapidly as the robot moves forward, scene features will only move by small, predictable amounts from one image to the next, and the correspondence problem is alleviated. Furthermore, it is evident that lateral motion vision is easily incorporated into the HIMM scheme (see Figure 3.17). A histogram is rapidly generated by incrementing the grid cells corresponding to the coordinates of the detected edges that are calculated for each sampled image. Also, since we know that an unobstructed line of sight should exist between the camera and the object, the cells along the line from the optical center to the detected edge are decremented.

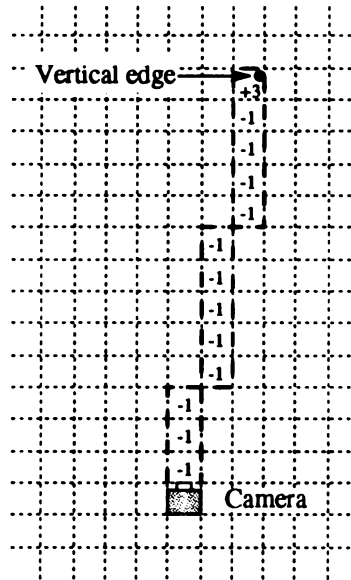


Figure 3.17. Constructing an HIMM grid map using lateral motion vision.

## Experimental Data Collection

We implemented an LMV algorithm on ROME that tracks points through a series of arithmetically-averaged 1D images and plots the coordinates of strong vertical edges in an HIMM grid map. To track edges and estimate  $dr/dt$ , we use a spatiotemporal image defined by Bolles *et al.* called the *epipolar plane image* or EPI [8, 2]. An EPI is created by collecting a series of one-dimensional (spatial) images separated by  $\Delta t$  in time and “stacking” them vertically. Figure 3.18 shows an EPI obtained from a side-looking camera on board a mobile robot traveling with constant velocity through our laboratory. Figure 3.19 shows three images from the lateral motion sequence used to construct the EPI.

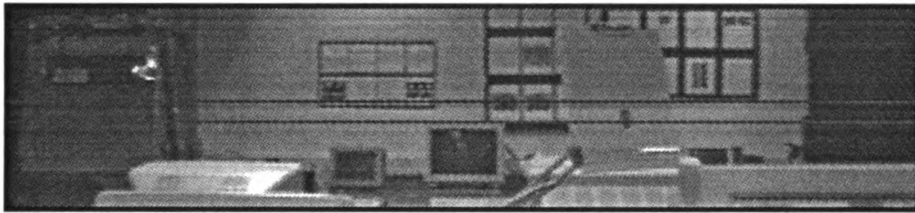


Figure 3.18. A lateral-motion epipolar plane image. Time advances from top to bottom, and each row is a one-dimensional image captured from the side-looking camera.

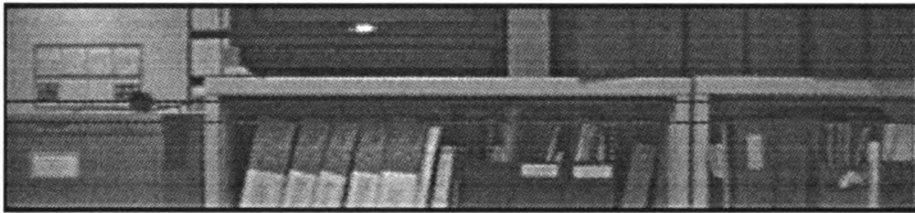
Note that since  $dr/dt$  is constant, features in the environment result in linear paths through the EPI. These paths can be tracked over time by means of a simple 1D edge-detection procedure. Each row of the EPI is convolved with a  $1 \times 3$  Prewitt edge operator [3] in the horizontal direction, thresholded, and then thinned. The resulting image consists of a series of pixel-wide traces of edges through the EPI (see Figure 3.20). We smooth out noise in the EPI by fitting a line to each trace. The quantity  $dr/dt$  of a feature is then the inverse slope of its trace. The farther a detected object is from the camera, the greater the slope of its trace will be.



(a)



(b)



(c)

Figure 3.19. First, middle and last images of the lateral motion sequence from which the EPI of Figure 3.18 was obtained. The horizontal lines indicate the scanlines averaged in each image. Averaging these scanlines in image (a) gives the first row of the EPI; likewise, image (c) provides the last row of the EPI.

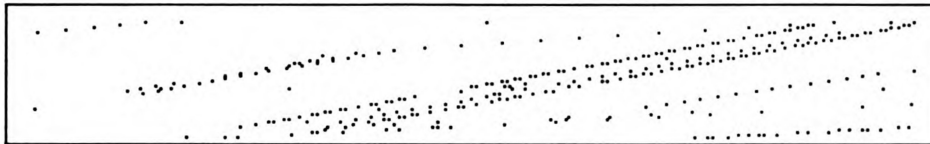


Figure 3.20. The set of lateral-motion EPI traces.

Figure 3.21 shows a lateral motion vision map of our laboratory generated using the HIMM method and dual side-looking cameras. By comparing this map to the sonar map of Figure 3.8, we see that lateral motion vision detects surface detail and object boundaries that ultrasound cannot. However, LMV lacks range precision, as is indicated by the high CV's radiating from the center of the room along the line of sight from the camera. Also, LMV cannot detect objects that are visually uniform, such as the laboratory whiteboard, as shown in Figure 3.24. However, maps generated using LMV are consistent despite varying sensing times and paths, as shown for sonar grid maps (see Figure 3.25).

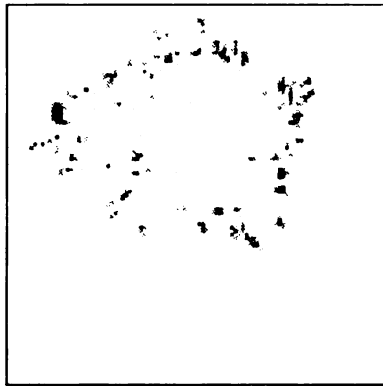


Figure 3.21. An LMV map of the PRIP Laboratory. Cell certainty values have been mapped inversely to intensity.

### 3.2.4 Forward Motion Vision

An algorithm for forward motion vision (FMV) is given by Brooks [11]. It uses the same 1D camera model as for lateral motion vision (Figure 3.15), except that the camera is assumed to be forward-looking. Figure 3.26 is an overhead view of the camera mounted parallel to the ground on a mobile robot which is moving forward

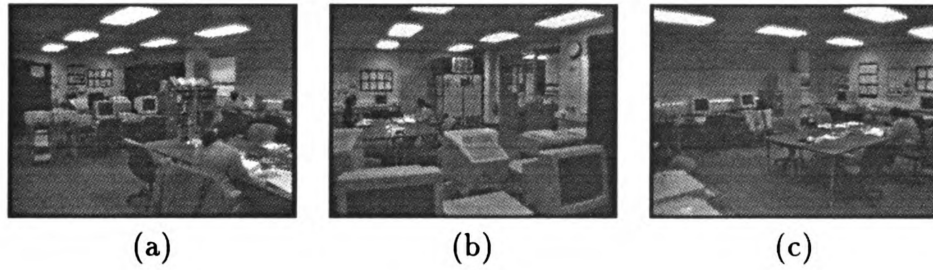


Figure 3.22. The PRIP Laboratory, from three views.

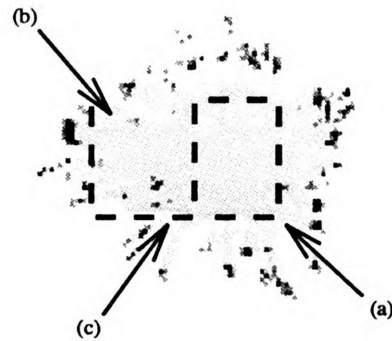


Figure 3.23. The LMV map of the PRIP lab. The arrows indicate the location of the views shown in Figure 3.22. The dashed line indicates the path taken by the robot.

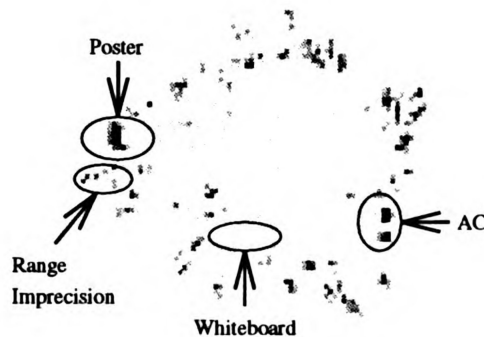


Figure 3.24. Characteristics of lateral motion vision data. LMV detects the high-contrast surface markings on wall posters and the air conditioner in the lab, but cannot sense the whiteboard. It also lacks the range precision of ultrasound.



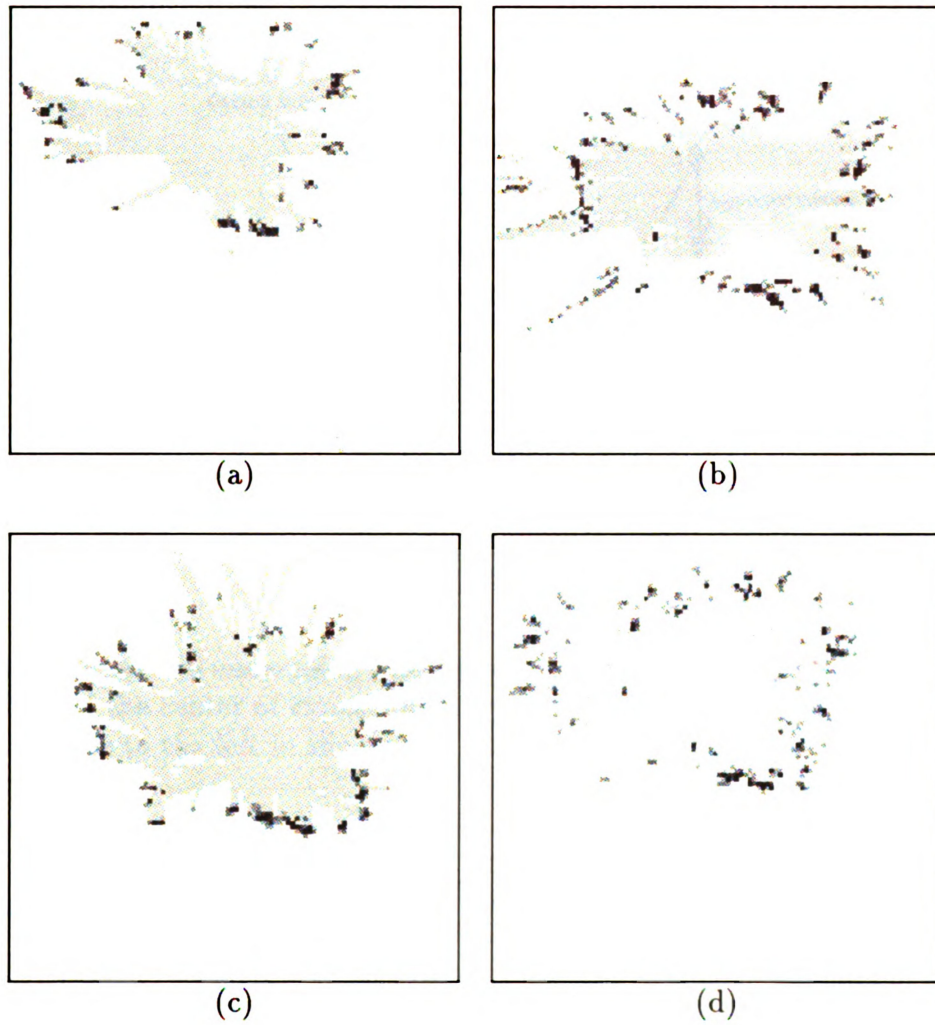


Figure 3.25. Four LMV grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot.

with constant velocity. The motion vector, optical axis, and image line are all coplanar. Only hand-alignment of the camera is required by this algorithm, so it is assumed that the camera will be turned somewhat from the direction of motion. The amount of misalignment is denoted by  $A$  and is assumed to be limited to  $\pm 5^\circ$ .

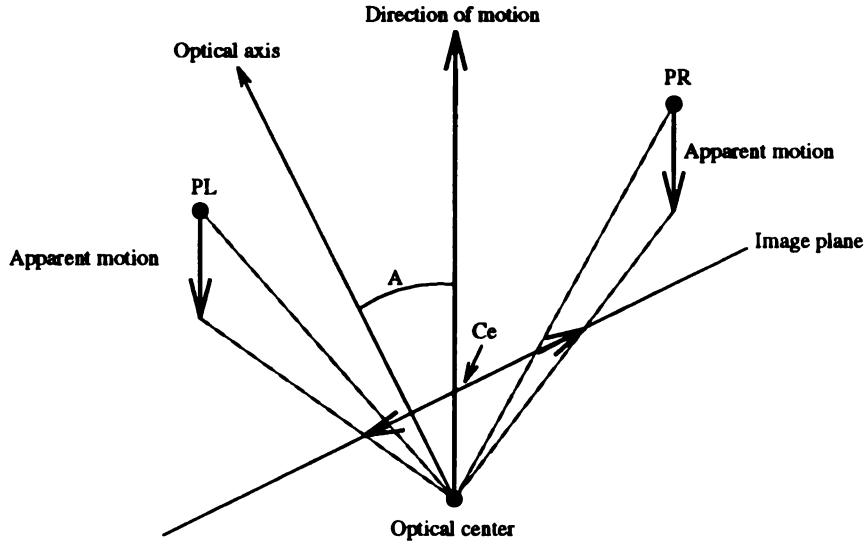


Figure 3.26. A camera undergoing translation in the direction of robot motion. Points to the right of the center of expansion move to the right in the image; points to the left of  $C_E$  move to the left in the image. Adapted from a figure in [11].

Figure 3.26 also shows the motion of projected points on the image plane as the camera is translated along the motion vector. The image of any point  $P_R$  which is to the right of the direction of motion will move rightwards in the image plane. The image of any point  $P_L$  which is to the left of the direction of motion will move leftwards in the image plane. Points directly ahead will not move at all in the image. The projection of such points is called the *center of expansion*;  $C_E$  is its coordinate in pixels. The trajectories of image points under these conditions are described by Bolles *et al.* [8]; Brooks then derived equations describing the distance to objects (in time-to-collision) and center of expansion under this camera system. From these, one

can compute the coordinates of tracked objects.

Figure 3.27 shows the camera system tracking a point in the environment. Here we see that the location of the point can be described by two non-orthogonal coordinates:  $x_2$ , its distance from the camera motion vector parallel to the image plane, and  $y_2$ , its distance from the optical center plane parallel to the motion vector. Note that the derivative of  $y_2$  with respect to time is the constant  $-v$ , where  $v$  is the magnitude of velocity of the robot.

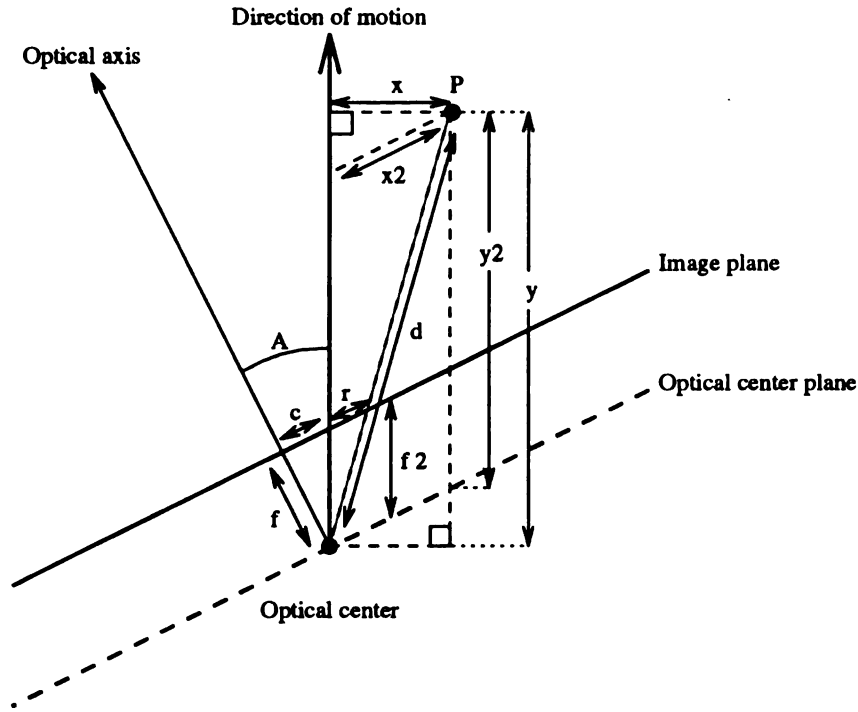


Figure 3.27. The camera system tracking a point under slight misalignment and constant forward velocity. Adapted from a figure in [11].

By similar triangles we see that the distance,  $r$ , of the image of point  $P$  from the center of expansion parallel to the image plane is given by

$$r = \frac{f_2 x_2}{y_2} \quad (3.3)$$

where  $f_2$  is the distance from the optical center to the center of expansion. Since  $x_2$  is constant, the derivative of  $r$  with respect to time is

$$r' = -\frac{f_2 x_2}{y_2^2} y_2'$$

and we observe that

$$\frac{r}{r'} = -\frac{y_2}{y_2'} = \frac{y_2}{v} = \frac{\text{distance}}{\text{velocity}} = \text{time}$$

thus,

$$y_2 = \frac{r}{r'} \cdot v$$

is the distance from the optical center plane to the object feature whose image is being observed.

Since  $C_E$ , the center of expansion, should remain constant, we can calculate

$$r = R - C_E, \quad r' = R'$$

where  $R$  is the distance of the image of point  $P$  from the left end of the image plane.

Therefore,

$$y_2 = \frac{R - C_E}{R'} v. \quad (3.4)$$

Clearly, the accuracy of forward motion analysis is very sensitive to the estimate made of  $C_E$ . Suppose one can measure  $R$  and  $R'$  at two distinct times  $t_0$  and  $t_1$ . Then the estimate of the distance from the image plane to  $P$  should decrease by precisely  $v(t_1 - t_0)$ . Writing the estimates as functions of time, we obtain

$$\frac{R(t_0) - C_E}{R'(t_0)} v - (t_1 - t_0) v = \frac{R(t_1) - C_E}{R'(t_1)} v,$$

and therefore,

$$C_E = \frac{R'(t_0)R(t_1) - R'(t_1)R(t_0) + R'(t_1)R'(t_0)(t_1 - t_0)}{R'(t_0) - R'(t_1)}.$$

One can estimate  $C_E$  and then compute the distance from the image plane to  $P$  using Equation (3.4). However, to determine the location of  $P$  in the workspace, one needs its coordinates  $(x, y)$  in the coordinate system of the robot. From Figure 3.27 we see that

$$y = y_2 + x_2 \sin A.$$

If  $c$  is the distance in pixels along the image plane from the center of view to the center of expansion, then

$$\sin A = \frac{c}{f_2}.$$

Therefore,

$$y = y_2 + \frac{cx_2}{f_2}$$

and by Equation (3.3)

$$y = y_2 + \frac{cry_2}{f_2^2}.$$

Finally, given

$$f_2 = \sqrt{c^2 + f^2},$$

we have

$$y = y_2 + \frac{cry_2}{c^2 + f^2}.$$

Then by the Pythagorean theorem,

$$x = \sqrt{x_2^2 - (y - y_2)^2} = \sqrt{\left(\frac{ry_2}{f_2}\right)^2 - (y - y_2)^2}.$$

As with lateral motion vision, we can track points through a forward-motion

EPI, and use the preceding equations to calculate the relative coordinates of sensed features, as

$$x = \sqrt{\left(\frac{ry_2}{f_2}\right)^2 - (y - y_2)^2},$$

$$y = y_2 + \frac{cry_2}{c^2 + f^2}.$$

Figure 3.28 shows an epipolar plane image obtained from a forward-looking camera on board ROME. Figure 3.29 shows three images from the forward-motion sequence used to construct the EPI.



Figure 3.28. A forward-motion EPI.

Equation (3.3) indicates that traces of features through a forward-motion EPI will be hyperbolas. By fitting a hyperbolic curve to each trace, noise is smoothed out of the motion sequence.

Unfortunately, forward motion vision has an inherent drawback when applied to HIMM mapping. Since the camera is forward-looking, the amount of the workspace that is sensed is limited by the field of view of the camera lens, and thus very narrow. This is exemplified in Figure 3.29, where the area projected onto the image plane becomes progressively smaller as the robot advances. Therefore, FMV mapping is best used in conjunction with dual side-looking cameras using lateral motion vision, in order to sense the maximum amount of the workspace as possible.



(a)



(b)



(c)

Figure 3.29. First, middle and last images of the forward motion sequence from which the EPI of Figure 3.28 was obtained. Image (a) provides the first row of the EPI; likewise, image (c) provides the last row of the EPI.

### 3.2.5 Infrared Proximity Detectors

Moravec [49] reports that infrared (IR) proximity detectors would also be appropriate in a grid-based representation. Unlike ultrasonic or vision sensors, IR proximity detectors return a binary value indicating when an object is detected in its field of view. For this sensor, Moravec suggests that the uncertainty profile take an elliptical shape, with high certainty values along the major axis, tapering to zero at the edges of the envelope. A typical IR proximity detector has a very narrow field of view and a range of only about half a meter. Figure 3.30 depicts a possible occupancy grid profile for this sensor using a bivariate Gaussian distribution for the uncertainty model.

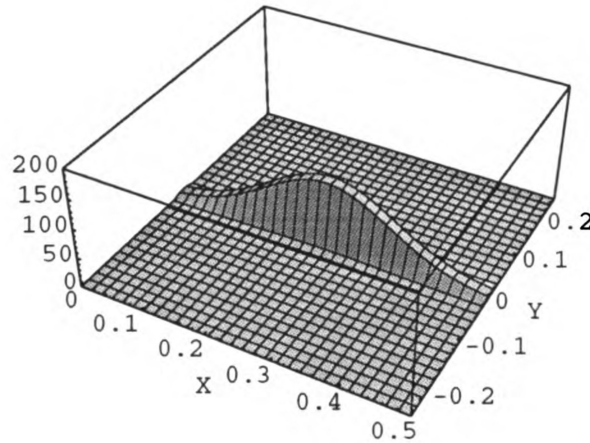


Figure 3.30. Occupancy grid profile for an infrared proximity detector located at  $(x = 0, y = 0)$  and facing toward  $(x = 0.5, y = 0)$ . A bivariate Gaussian function  $p(x, y | d) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{(y-d)^2}{\sigma_y^2}\right)\right]$  models sensing uncertainty. This profile is used for probability of occupancy or vacancy, depending on whether or not the sensor detected an object.

Because of the binary nature of IR data, this sensor is not easily incorporated



into the HIMM framework. One approach is to increment all the cells along a line in front of the sensor when the reading is “on”, and to decrement the same cells if the reading is “off”. See Figure 3.31.

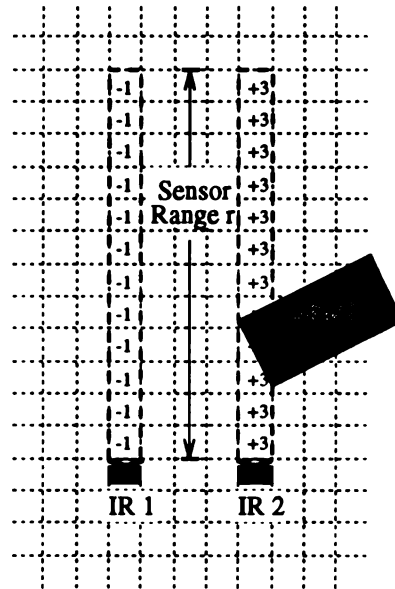


Figure 3.31. Building an HIMM grid map with infrared proximity sensors.

### Experimental Data Collection

Figure 3.32 shows an IR HIMM grid map of our laboratory created using the above method. Since this sensor only makes local measurements of the environment, the resulting maps are highly dependent upon the path taken by the robot during sensing. This sensor is most useful for coarse-level localization within cluttered locales. See Figure 3.33.

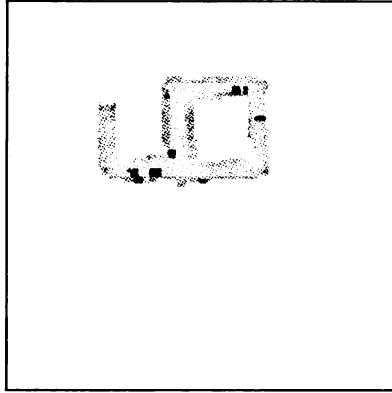


Figure 3.32. An IR map of the PRIP lab. The light gray regions indicate unoccupied sensed cells, and thus reveal the path taken by the robot through the workspace.

### 3.2.6 Tactile Sensors

A final sensor amenable to HIMM grid-maps are tactile sensors. These are typically implemented on mobile robots as bumpers. Like infrared proximity detectors, tactile sensors are local, binary-output devices. Because of this, they are also most appropriate for mapping within highly cluttered domains. Figure 3.34 shows an approach to HIMM grid map construction using tactile sensors.

## 3.3 Experimental Datasets

We employed ultrasonic range sensing, lateral motion vision, and infrared object detection in the HIMM framework to collect grid maps of “door” and “room” locales for the coarse-level localization classifier. When mapping a locale, three independent “sensor” grid maps (one for each sensing source) are simultaneously created (see Figure 3.35). The result is three different but corroborating representations of the same locale. Each sensor map reflects different spatial structures; together they provide a robust model of the workspace. Appendix A shows examples of the three sensor grid maps of the ten locales mapped in the Engineering Building. Ten maps were collected

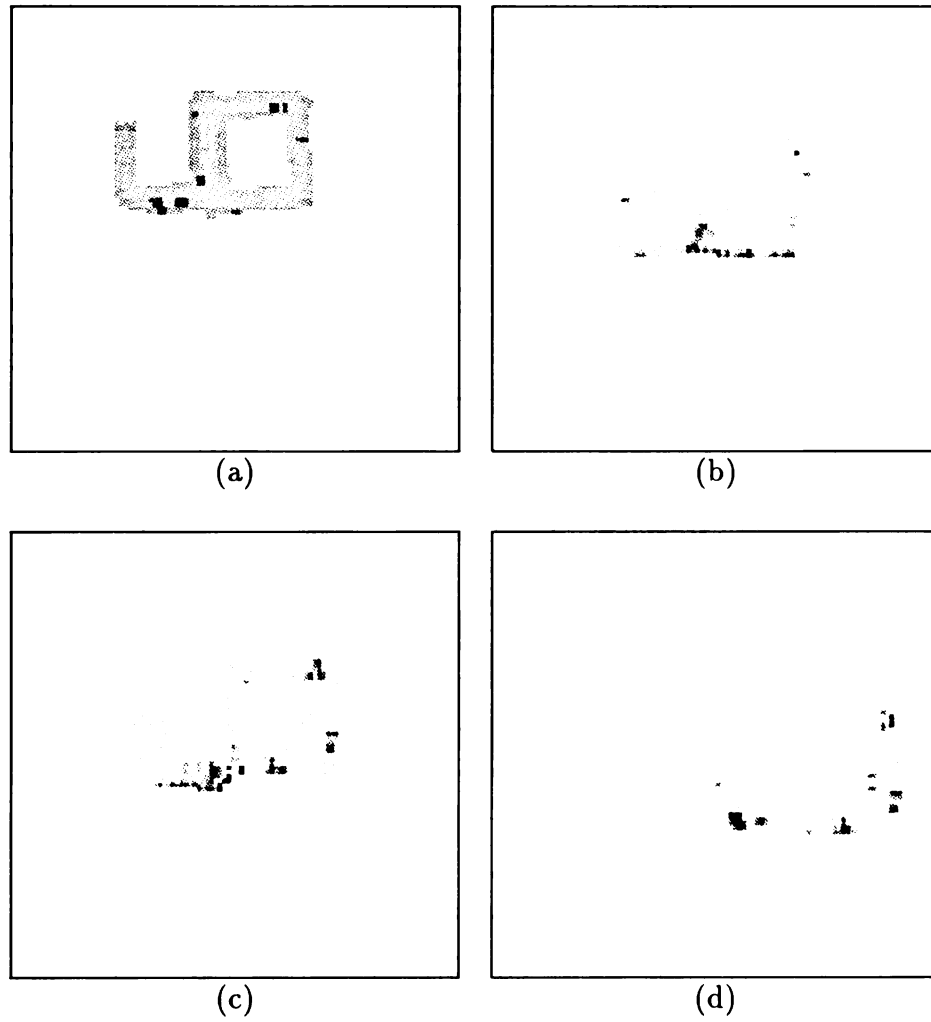


Figure 3.33. Four IR grid maps of the PRIP lab, taken at different times under differing exploration paths of the robot.

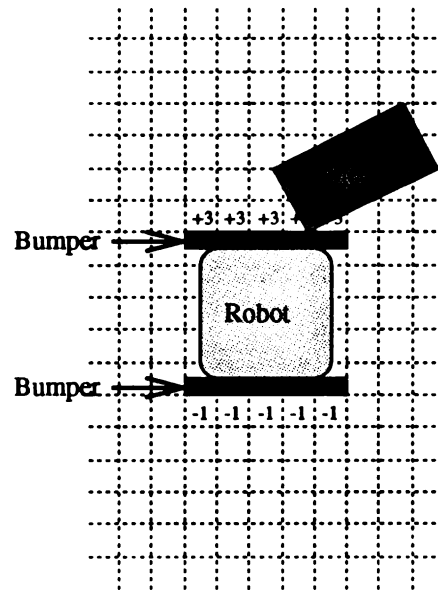


Figure 3.34. Building an HIMM grid map with tactile sensors.

from each locale at different times (sometimes many days apart) and under differing exploration paths.

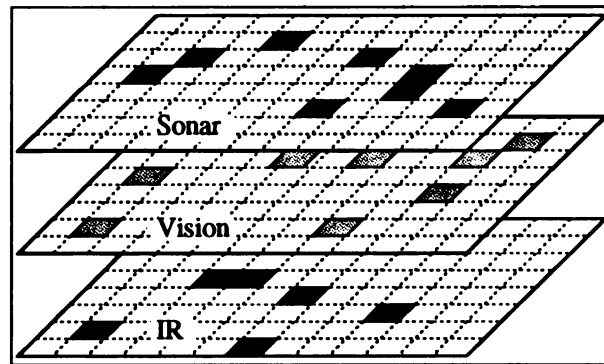


Figure 3.35. Sonar, Vision, and Infrared sensor grid maps of a locale.

# CHAPTER 4

## Grid Map Processing

Since grid maps consist of a two-dimensional array of bytes, they can be subjected to many of the same operations appropriate for intensity images. In the following sections we examine some potentially useful techniques for preprocessing and description of grid-based maps.

### 4.1 Preprocessing

Common image processing operations are useful for identifying spatial structures in grid maps. For example, a threshold operation applied to the map of Figure 3.8 results in a map of the occupied regions of the locale. Combining this map with the original through a logical XOR operation shows the (disconnected) free space regions and sonar anomalies. Selecting the largest of these regions through a connected component analysis yields a (noisy) map of the free space of the room. Applying morphological dilation and erosion operators then results in a smoothed map of the free space, from which shape features may be computed. Figure 4.1 shows the result of each of these operations.

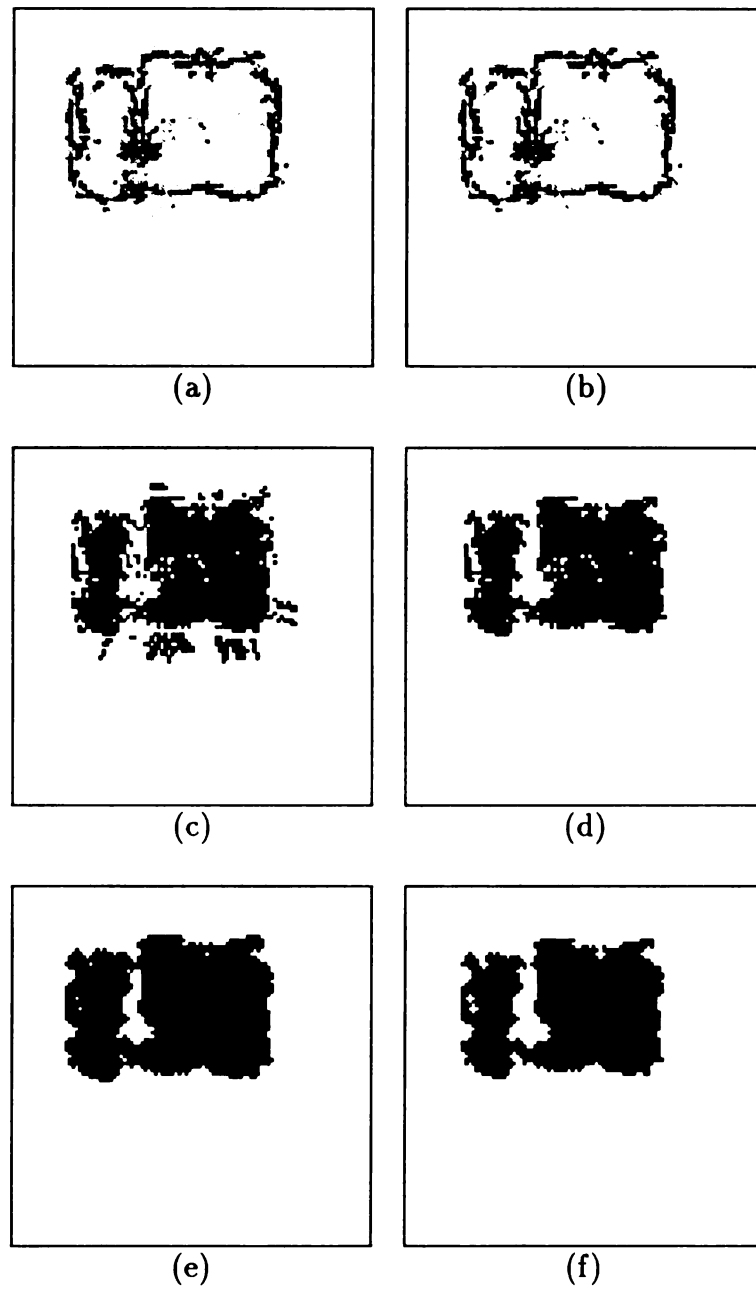


Figure 4.1. Image processing operations on a sample grid map: (a) the original map, (b) occupied regions, (c) disconnected free space and sonar anomalies, (d) the free space of the locale, (e) morphological dilation, (f) morphological erosion.

## 4.2 Grid Map Descriptions

In order to classify a given map as corresponding to a particular known locale, appropriate region descriptions must be chosen to represent the grid maps and provide feature vectors for the classifier. Note that since no global coordinate system is used in coarse-level localization, the grid maps can be expected to be translation and rotation variant, which must be handled by the chosen descriptions. The following sections present a few descriptions that we considered for use with our locale classifiers.

### 4.2.1 Fourier Descriptors

Using this method, the frequency-domain description of the (closed) boundary of the free space of a locale (as in Figure 4.1(f)) is used for shape features [3]. The discrete Fourier series expansion of the “intrinsic function”  $\phi(k)$  of a boundary of length  $L$  is given as

$$\phi(k) = \frac{1}{L} \sum_{n=0}^{L-1} X(n) e^{jnk2\pi/L},$$

where the coefficients  $X(n)$  are given by

$$X(n) = \sum_{k=0}^{L-1} \phi(k) e^{-jnk2\pi/L}$$

and the intrinsic function is

$$\phi(s) = \psi(s) - 2\pi s/L, \tag{4.1}$$

where  $\psi(s)$  is the tangent angle of the boundary with respect to arc length. The term  $2\pi s/L$  in Equation (4.1) subtracts out the rising component of  $\psi(s)$ . Figure 4.2 depicts how the intrinsic function is computed from a boundary.

The magnitudes of the lower-order Fourier descriptors ( $X(n)$  for small  $n$ ) provide

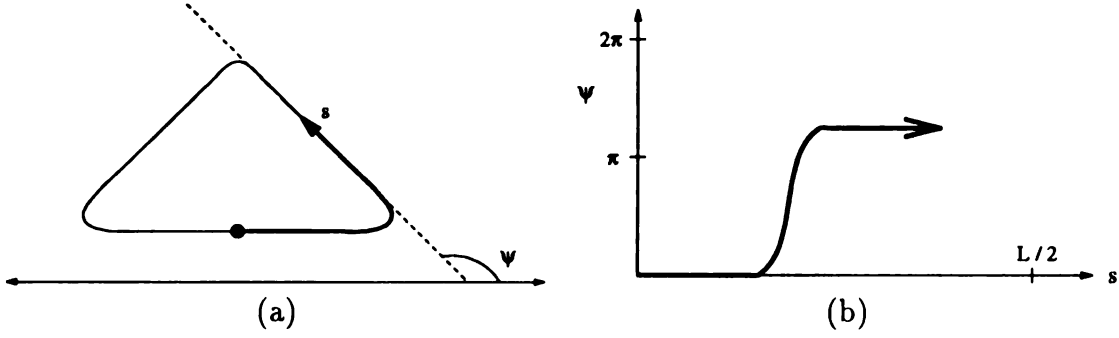


Figure 4.2. Calculating  $\psi(s)$  from a boundary.

useful features related to the “lobedness” of the shape that could be used to identify a locale. Furthermore, the Fast Fourier Transform (FFT) provides a computationally efficient means of calculating these features [12]. However, these features can not account for structures internal to the map; only the boundary is considered and other information is lost.

#### 4.2.2 Standard Moments

The two-dimensional  $(p + q)$ th order moments of gray-level image  $I(i, j)$  are defined as [44]:

$$m_{pq} = \sum_i \sum_j i^p j^q I(i, j) \quad p, q = 0, 1, 2, \dots$$

where  $I(i, j)$  is typically the intensity value of a digitized image at row  $i$ , column  $j$ .

For our purposes, it is the certainty value of the grid map at coordinate  $(i, j)$ .

If instead of  $I(i, j)$ , we use its binary representation  $B(i, j)$ :

$$B(i, j) = \begin{cases} 1 & \text{if cell } (i, j) \text{ has been sensed} \\ 0 & \text{otherwise} \end{cases}$$



the centroid of the grid map is given as

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}},$$

where  $m_{00}$  is the area of the sensed region of the map.

By use of the centroid of the map, translation-invariant central moments are formulated:

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q I(i, j). \quad (4.2)$$

From this, rotation-invariant central moments are derived:

$$\phi_{pq} = \sum_{r=0}^p \sum_{s=0}^q (-1)^{q-s} C_r^p C_s^q (\cos \theta)^{p-r+s} (\sin \theta)^{q-s+r} \mu_{p-r+q-s, r+s} \quad (4.3)$$

where

$$C_r^p = \frac{p!}{r! (p-r)!}$$

and  $\theta$  is the orientation of the major axis of the map, calculated using  $B(i, j)$  in place of  $I(i, j)$  in Equation (4.2):

$$\theta = \frac{1}{2} \arctan \left[ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right].$$

Finally, standard moments can be calculated, which are invariant under translation, rotation, and scale change:

$$N_{pq} = \frac{\phi_{pq}}{\phi_{00}^\gamma} \quad (4.4)$$

where

$$\gamma = \frac{1}{2} (p + q) + 1. \quad (4.5)$$

Note that the zeroth and first order standard moments for any  $I(i, j)$  will be  $N_{00} = 1$ ,  $N_{01} = N_{10} = 0$ , and  $N_{11} = 0$ . Thus, six standard moments of second- and

third-order are available for use as features. (Higher-order moments suffer from a wide dynamic range and noise from digital sampling errors.) The standard moments technique is appealing due to its translation and rotation invariance. However, the scale of a map may be a significant discriminating feature in locale recognition, so the non-scale-invariant moments of Equation (4.3) should be considered as well.

We investigated the application of these six standard moments as feature vectors in locale classification. Unfortunately, we observed that it is often difficult to reliably extract the major axis from the data, which greatly affects the resulting features. This observation is also noted by Reeves [51].

### 4.2.3 Moment Invariants

From the second- and third-order central moments, Hu proposed seven *moment invariants* that are independent of translation, rotation and scale [32]. First, normalized (scale-invariant) central moments are derived from Equation (4.2) as

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma}$$

where  $\gamma$  is defined in Equation (4.5). The seven moment invariants are then given as

$$\varphi_1 = \eta_{20} + \eta_{02},$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2,$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2,$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,$$

$$\begin{aligned}
\varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \\
&\quad \times [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \\
\varphi_6 &= (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \\
\varphi_7 &= (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12}) \\
&\quad \times [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}$$

The first six moment invariants ( $\varphi_1$  to  $\varphi_6$ ) are invariant under rotation and reflection. Moment invariant  $\varphi_7$ , however, changes its sign for reflected images of an object, so it is useful in distinguishing between mirror-image patterns.

A drawback to the moment invariant technique is that there is no intuitive meaning for any of the seven invariants. Thus, it is difficult to judge which of the invariants would be most appropriate to use with a given set of data.

### 4.3 Experimental Data

After some experimentation, we decided to use Hu's seven moment invariants as grid map features, since this approach could account for internal structures and also provided us with the most discriminative features. Seven moment invariants can be calculated for each of the three sensor grid maps (i.e., sonar, vision, and IR); thus, twenty-one features are available for locale description. However, the moment invariants guarantee scale invariance, and this eliminates some potential discriminatory information from the feature space. So we included  $\mu_{00}$ , the zeroth-order central moment, of each sensor map in our feature set. This provides three additional features

related to the size of the locale, resulting in twenty-four candidate features.

We calculated  $\mu_{00}$  and  $\varphi_1, \varphi_2, \dots, \varphi_7$  for each locale map in our database. For simplicity, we excluded the infrared proximity maps of room locales from consideration—by inspection it was evident that these maps provided no useful information in these wide-open regions. However, we did process IR grid maps for the door locales, as they are more useful in constrained environments. Thus, the room locale dataset had 100 patterns in 16 dimensions, and the room locale dataset had 100 patterns in 24 dimensions. Table 4.1 shows how the features are ordered for each pattern. We also use the notation  $v^R$  to denote feature  $v$  calculated from sensor map  $R$ ; i.e.,  $\varphi_2^V$  indicates the second moment invariant calculated on the vision grid map. Of course, not all of these features are useful for recognition, as we will see in the next chapter.

Table 4.1. Ordering of the features in the experimental dataset. Note that the IR features are not included for the room locales.

	$\mu_{00}$	$\varphi_1$	$\varphi_2$	$\varphi_3$	$\varphi_4$	$\varphi_5$	$\varphi_6$	$\varphi_7$
Sonar	1	2	3	4	5	6	7	8
Vision	9	10	11	12	13	14	15	16
IR	17	18	19	20	21	22	23	24

Figure 4.3 shows a star plot [5] of the class centroids in the room locale dataset. Class centroids are calculated by finding the mean feature values of all patterns in the class. The star plot depicts the data by mapping the feature values of the centroids onto the length of the radial lines in each star, beginning with the right-hand side and continuing counter-clockwise. Thus, each star in this plot is composed of 16 radial lines (not all of the lines may be visible due to their length). Conceptually, the more similar that two class centroid stars appear, the more similar the pattern classes are in the data. Note the similarity between the stars for “EB 323” and “EB 360”. One

would expect the classifier to have difficulty discriminating between these two locales.

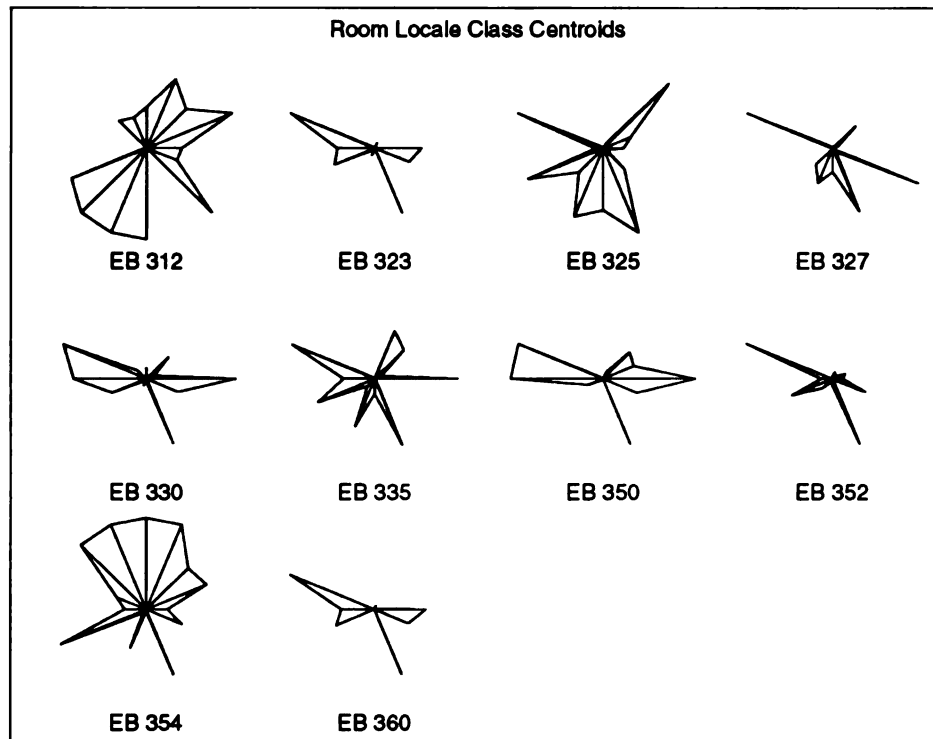


Figure 4.3. A star plot of the class centroids for room locale maps.

Figure 4.4 shows a star plot of the door locale class centroids (in 24 features). The dissimilarity of some of the classes is a favorable indication that map recognition will be feasible using this dataset.

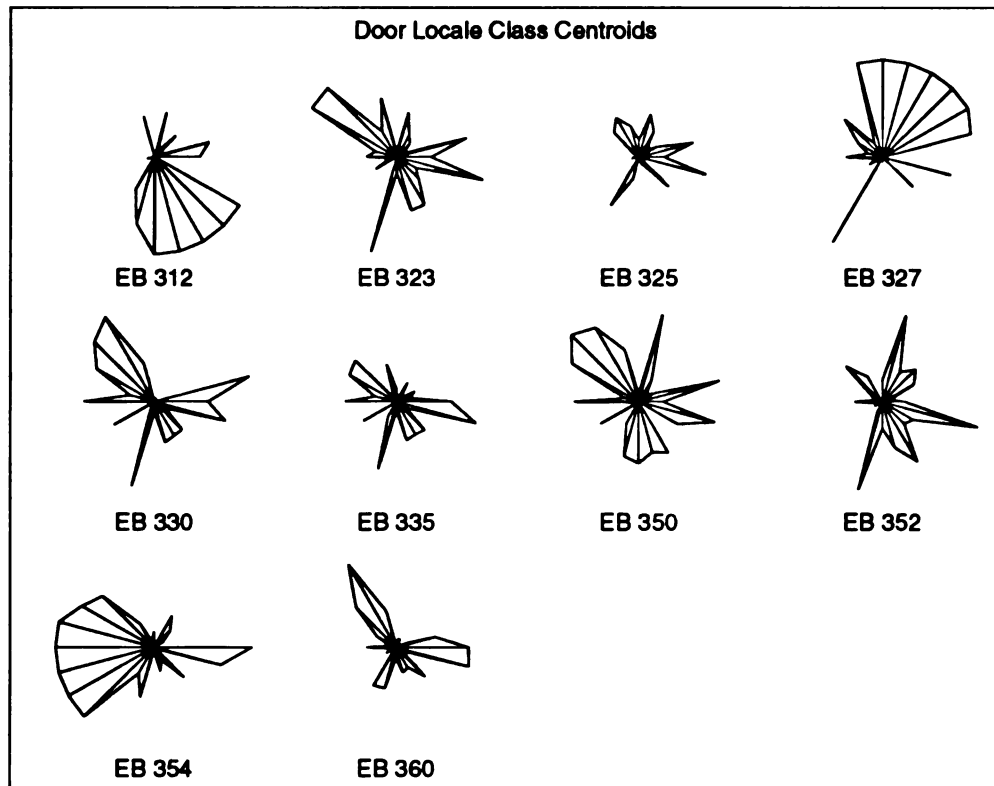


Figure 4.4. A star plot of the class centroids for door locale maps.

# CHAPTER 5

## Grid Map Classification

We have investigated locale recognition through grid map classification. Having generated room and door locale datasets, we experimented with several commonly used classifiers to find the one which gives the best recognition performance. We reason that this classifier could be built into a robot navigation system to perform on-the-fly coarse-level localization.

### 5.1 Pattern Classification

The following model is assumed: A given pattern, represented as a feature vector  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ , is presented to the classifier. The classifier then assigns the pattern to one of  $c$  classes  $(\omega_1, \omega_2, \dots, \omega_c) \in \Omega$  based on the decision rule  $\delta(\cdot)$  and the observed feature values. Consider the classifier as a machine that selects the class corresponding to the *discriminant function*,  $g_i(\mathbf{x})$ ,  $i = 1, \dots, c$ , of greatest value. Thus,

$$\delta(\mathbf{x}) = \alpha_i \ni g_i(\mathbf{x}) \geq g_j(\mathbf{x}) \quad \forall j \neq i,$$

where  $\alpha_i$  corresponds to the action “choose class  $\omega_i$ ”.

We denote the set of training patterns as  $\mathcal{X} = \{(\mathbf{x}_1, \theta_1), (\mathbf{x}_2, \theta_2), \dots, (\mathbf{x}_n, \theta_n)\}$ , where each  $\mathbf{x}_i$  is a  $d$ -dimensional feature vector and each  $\theta_i \in \Omega$  is the class

membership of  $\mathbf{x}_i$ . Likewise, the training patterns belonging to class  $\omega_i$  are represented as  $\mathcal{X}_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_{n_i}^i\}$ .

Except where indicated, the material for the following sections is taken primarily from [22] and [21].

### 5.1.1 Bayes Classifier

The *Bayes* classifier for minimizing the probability of error assigns test pattern  $\mathbf{x}$  to class  $\omega_i$  according to the decision rule

$$\delta_B(\mathbf{x}) = \alpha_i \ni g_i(\mathbf{x}) \geq g_j(\mathbf{x}) \quad \forall j \neq i,$$

where

$$g_i(\mathbf{x}) = p(\omega_i | \mathbf{x}).$$

Thus, the discriminant function is the class *a posteriori* density, given by Bayes' formula

$$p(\omega_i | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_i)P(\omega_i)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j)P(\omega_j)}.$$

Here,  $p(\mathbf{x} | \omega_i)$  is the class-conditional density, and  $P(\omega_i)$  is the *a priori* probability of class  $\omega_i$ .

It is a well-known fact that if the class-conditional densities and *a priori* probabilities are exactly known, the Bayes classifier will be optimal in terms of minimum probability of error. However, in the case that the class conditional densities are estimated, the Bayes classifier is still powerful in that it accounts for class *a priori* probabilities.

Consider the case where the class-conditional densities are multivariate Gaussian, i.e.,

$$p(\mathbf{x} | \omega_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^t \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right], \quad (5.1)$$



where  $\boldsymbol{\mu}_i$  and  $\Sigma_i$  are the  $d$ -component mean vector and  $d \times d$  covariance matrix of the class-conditional density of  $\omega_i$ , respectively. Dropping the denominator (which is independent of  $\omega_i$ ) from the class *a posteriori* density, we can rewrite the discriminant function as

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i)P(\omega_i),$$

or equivalently,

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i).$$

Then, since  $p(\mathbf{x} \mid \omega_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$ , we have

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \Sigma_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i).$$

Finally, we can ignore  $\frac{d}{2} \ln 2\pi$  as an additive constant, and obtain

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \Sigma_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i). \quad (5.2)$$

### 5.1.2 Minimum Mahalanobis Distance Classifier

The *minimum Mahalanobis distance* (MMD) classifier assigns patterns to the class whose centroid  $\boldsymbol{\mu}_i$  has the smallest *Mahalanobis distance* from the test pattern  $\mathbf{x}$ . The squared Mahalanobis distance between two vectors  $\mathbf{x}$  and  $\boldsymbol{\mu}_i$  is given as

$$\mathcal{M}^2(\mathbf{x}, \boldsymbol{\mu}_i) = (\mathbf{x} - \boldsymbol{\mu}_i)^t \Sigma_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i),$$

which is twice the first term on the right hand side of Equation (5.2). Thus, this distance measure implicitly models the class-conditional densities as multivariate Gaussian distributions (as in Equation (5.1)), but does not account for *a priori* class probabilities.

The Mahalanobis distance measure effectively scales the distance between two vectors as if each feature had unit variance. Thus, a point of equal Euclidean distance from two class centroids will have greater Mahalanobis distance from the centroid of the class with smaller overall variance.

The decision rule of this classifier is then defined as

$$\delta_{MMD}(\mathbf{x}) = \alpha_i \ni g_i(\mathbf{x}) \geq g_j(\mathbf{x}) \quad \forall j \neq i$$

where

$$g_i(\mathbf{x}) = -\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_i).$$

In practice, the parameters  $\boldsymbol{\mu}_i$  and  $\Sigma_i$  must be estimated from the training samples for each class in order to calculate  $\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_i)$ . A straightforward approach is to apply maximum likelihood parameter estimation; here,  $\boldsymbol{\mu}_i$  and  $\Sigma_i$  are estimated by the class  $\omega_i$  sample mean and covariance given by

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathbf{x}_k^i$$

and

$$\hat{\Sigma}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} (\mathbf{x}_k^i - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_k^i - \hat{\boldsymbol{\mu}}_i)^t.$$

To obtain a reasonable estimate of the covariance matrix  $\Sigma_i$ , the number of training samples must be 5 to 10 times  $d$ , the dimensionality of the data [34]. Furthermore, if there are fewer than  $d$  training samples in a class,  $\Sigma_i$  will be singular and inversion of the matrix is impossible. A common solution to this dilemma is to assume uncorrelated features; thus, if  $\hat{\sigma}_1^2, \hat{\sigma}_2^2, \dots, \hat{\sigma}_d^2$  are the sample feature variances, we estimate  $\Sigma$  with the diagonal matrix

$$\hat{\Sigma} = \begin{bmatrix} \hat{\sigma}_1^2 & & & \\ & \hat{\sigma}_2^2 & & \\ & & \ddots & \\ & & & \hat{\sigma}_d^2 \end{bmatrix},$$

resulting in

$$\hat{\Sigma}^{-1} = \begin{bmatrix} 1/\hat{\sigma}_1^2 & & & \\ & 1/\hat{\sigma}_2^2 & & \\ & & \ddots & \\ & & & 1/\hat{\sigma}_d^2 \end{bmatrix}.$$

The MMD classifier is space efficient in that it requires a simple parametric representation of the training data. However, this approach will not perform well if the class distributions are not approximately Gaussian, particularly if they are not unimodal.

### 5.1.3 K-Nearest-Neighbor Classifier

The *nearest-neighbor* (NN) classifier assigns a test pattern  $\mathbf{x}$  to the same class  $\omega_i$  as training pattern  $\mathbf{x}_r^i \in \mathcal{X}_i$  nearest to  $\mathbf{x}$  in the feature space. In discriminant function formulation,

$$\delta_{NN}(\mathbf{x}) = \alpha_i \ni g_i(\mathbf{x}) \geq g_j(\mathbf{x}) \quad \forall j \neq i,$$

where

$$g_i(\mathbf{x}) = -\|\mathbf{x} - \mathbf{x}_r^i\|$$

and

$$\|\mathbf{x} - \mathbf{x}_r^i\| \leq \|\mathbf{x} - \mathbf{x}_k^i\| \quad \forall \mathbf{x}_k^i \in \mathcal{X}_i, k \neq r.$$

It can be shown that as the number of training samples,  $n$ , goes to infinity, the error rate of the nearest neighbor classifier is bounded by twice the optimal (Bayes) error rate.

The nearest neighbor classifier can be extended to the *k*-nearest neighbor (K-NN) classifier. In this case, the test pattern  $\mathbf{x}$  is assigned to the class most frequently represented among the  $k$  training samples nearest to  $\mathbf{x}$ . Let  $K_i(\mathbf{x}, k)$  be the number of patterns from class  $\omega_i$  among the  $k$  nearest training patterns to  $\mathbf{x}$ . Then,

$$\delta_{KNN}(\mathbf{x}) = \alpha_i \ni g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \forall j \neq i,$$

where

$$g_i(\mathbf{x}) = K_i(\mathbf{x}, k).$$

Ties must be handled or avoided by proper choice of  $k$ , randomization, rejection, or defaulting to the 1-NN case. It can be shown that as  $k$  and  $n$  go to infinity, the error rate of the  $k$ -nearest neighbor classifier approaches the Bayes error rate.

The  $k$ -nearest-neighbor rule is appealing due to its simple, intuitive approach to classification. Unfortunately, this technique requires the storage and search of all the training patterns, which is expensive to implement as  $n$  becomes large.

Although parametric classifiers such as the Bayes and minimum Mahalanobis distance classifiers are powerful tools, in finite-sample-size cases nonparametric classifiers can outperform them. Fix and Hodges [30] concluded that if the parametric model of the class distributions was incorrect or if insufficient training samples were available to properly estimate the parameters for the model, then the  $k$ -nearest-neighbor classifier may outperform a parametric classifier.

## 5.2 Error Estimation

In order to evaluate the performance of a particular classifier, it is desirable to have an estimate of its probability of misclassification or “error rate”. This is typically done by presenting the classifier with a test dataset—the probability of misclassification is

then estimated as the percentage of samples that were misclassified.

Suppose that  $\tau$  is the number of samples out of  $n$  independent, identically distributed test patterns that were misclassified. Then, the presentation of each test pattern is a Bernoulli trial and  $\tau$  is binomially distributed:

$$P(\tau \mid \varepsilon) = \binom{n}{\tau} \varepsilon^\tau (1 - \varepsilon)^{n-\tau}$$

where  $\varepsilon$  is the true but unknown error rate of the classifier. The maximum likelihood estimate,  $\hat{\varepsilon}$ , of  $\varepsilon$  is given by

$$\hat{\varepsilon} = \frac{\tau}{n}.$$

In the case where  $n$  is large ( $n\varepsilon(1 - \varepsilon) > 5$ ), we know by the DeMoivre-Laplace theorem [7],

$$\frac{\hat{\varepsilon} - \varepsilon}{\sqrt{\hat{\varepsilon}(1 - \hat{\varepsilon})/n}} \approx N(0, 1)$$

which can be used to compute a confidence interval for  $\hat{\varepsilon}$ .

A problem in classifier design is that one may have only a finite number of samples to both design and evaluate the classifier. With the *resubstitution* error estimate (also called the *apparent* error rate), the same set of samples is used to train the classifier as well as test it. It is well known that the resulting error estimate is optimistically biased.

In the *holdout* method of error estimation, the available data is divided into training and testing datasets—the training patterns are used to design the classifier, and the percentage of misclassified test patterns is taken as the error rate of the classifier. This estimate is pessimistically biased.

Unfortunately, when the number of available samples is limited, we encounter a dilemma in dividing the data into test and training samples. If we reserve most of the data for the design of the classifier, we will lack confidence in the resulting error rate

estimate. If we reserve most of the data for testing the classifier, we cannot obtain a good classifier design.

The *leave-one-out* error estimate is especially useful in the case of small sample sizes. Suppose a total of  $n$  samples are available to design as well as test the classifier. Here, the classifier is designed with  $(n - 1)$  samples and tested on the remaining sample. This is repeated  $n$  times by leaving out a different pattern each time. The error estimate is then the total number of misclassified patterns divided by  $n$ . Although the leave-one-out estimate makes efficient use of the available samples and is unbiased, it has a large variance and requires the design of  $n$  different classifiers.

*Bootstrap* error estimation techniques [19, 36] provide a more accurate estimate of classifier performance when the number of available samples is limited. By resampling the original data, these methods create “fake” data sets with which to train and test the classifier.

We define bootstrap sample  $\mathcal{X}^*$  as a set  $\{(\mathbf{x}_1^*, \theta_1^*), (\mathbf{x}_2^*, \theta_2^*), \dots, (\mathbf{x}_n^*, \theta_n^*)\}$  randomly selected from the training set  $\mathcal{X}$  with replacement. Bootstrap sample  $\mathcal{X}^{*b}$  is the  $b$ -th of these sets. Furthermore, let  $\Delta(\mathbf{x}, \mathcal{X}^{*b})$  be the classifier decision rule based on training set  $\mathcal{X}^{*b}$ . We then define indicator function  $Q[\cdot, \cdot]$  as

$$Q[\theta, \Delta(\mathbf{x}, \mathcal{X}^{*b})] = \begin{cases} 0 & \text{if } \Delta(\mathbf{x}, \mathcal{X}^{*b}) = \theta, \\ 1 & \text{otherwise.} \end{cases}$$

The *E0* bootstrap estimator counts the number of misclassified training patterns which do not appear in the bootstrap sample. This estimate is obtained by summing the misclassified samples over all bootstrap samples and dividing by the total number of training patterns not appearing in the bootstrap sample. Let  $A_b = \{i \mid (\mathbf{x}_i, \theta_i) \notin \mathcal{X}^{*b}\}$  denote the index set of training patterns which do not

appear in the  $b$ -th bootstrap sample. Then we define the  $E0$  estimator as

$$E0 = \frac{\sum_{b=1}^B \sum_{A_b} Q[\theta_i, \Delta(\mathbf{x}_i, \mathcal{X}^{*b})]}{\sum_{b=1}^B |A_b|},$$

where  $|A_b|$  denotes the cardinality of set  $A_b$ . Efron [24] suggests that  $B$  need not be greater than 200.

To compensate for the pessimistic bias of the  $E0$  estimator, Efron proposes the (unbiased)  $E632$  error estimator, defined as

$$E632 = 0.368 * App + 0.632 * E0,$$

where  $App$  is the apparent error rate of the classifier. This estimator has smaller variance than leave-one-out error estimates [36].

### 5.3 Feature Selection

A fundamental issue in pattern recognition is to determine which features should be employed to obtain the smallest classification error for a given problem. Jain and Dubes [35] identify three reasons why dimensionality reduction is necessary:

1. If a finite number of training samples is available, classifier performance is known to deteriorate if superfluous features are included. This phenomena is called the *curse of dimensionality* [34].
2. Measurement and classification cost is reduced if fewer features are used, resulting in greater efficiency.
3. Successful estimation of class-conditional density parameters require a large ratio of training patterns to dimensionality. For example, Kalayeh and Landgrebe

[37] recommend that there be five to ten times as many training patterns as there are features in the data. If additional training samples are unavailable, one must resort to limiting the number of features considered.

In order to evaluate the effectiveness of a chosen feature subset, some criterion function must be used. Typically this is the estimated error rate of the classifier or some measure of class separation in the feature space. Here we discuss a few of the common approaches to feature selection.

Cover and Van Campenhout [16] have shown that to guarantee the selection of the best subset of  $p$  features out of  $d$  measurements, one must examine all  $\binom{d}{p}$  possible subsets of size  $p$ . Of course, this exhaustive search will explode combinatorically for problems with even moderate dimensionality. In the likely event that exhaustive search is infeasible, other suboptimal feature selection methods have been proposed.

One approach is to select the  $p$  individually best features from the available  $d$  measurements. However, Cover [15] has shown that the best subset of size two may not even include the best individual feature. Thus, this technique will probably never lead to an optimal subset, because it does not account for the statistical dependence among the measurements.

Another method, called *sequential forward selection*, begins with the best individual feature, and then enlarges the subset by adding one measurement at a time which in combination with the chosen features maximizes (or minimizes) the criterion function. Although still suboptimal, this straightforward approach is able to make some account of statistical dependence among the measurements. Whitney proposed the leave-one-out error rate of a 1-NN classifier as a criterion function for sequential forward feature selection; this is known as *Whitney's method* [58].



## 5.4 Experimental Results

We investigated the application of the various classifiers and techniques described above to recognize grid-based maps of locales. Using our two datasets of 100 patterns, we designed and tested both K-NN and MMD classifiers for room and door locale recognition.

We considered the room locales first. Table 5.1(a) shows the result of sequential forward feature selection using a 3-NN classifier on the 16 possible room locale features. (In order to ensure that features with large variance would not dominate the feature space, we normalized the data to have zero mean and unit variance.) The first feature chosen was  $\{\mu_{00}^V\}$ ; the best pair found was  $\{\mu_{00}^V, \mu_{00}^S\}$ , etc. We evaluate the performance of the classifier under each feature subset using the leave-one-out error estimate, denoted  $\hat{\epsilon}_u$ ; the number of patterns misclassified out of 100 is denoted as  $\tau$ .

Likewise, Table 5.1(b) shows the feature subsets selected for use with the room locale dataset and MMD classifier. Since the number of available training samples was limited, we assumed uncorrelated features to obtain diagonal covariance matrices for each class.

Note the effect of the curse of dimensionality shown in the results. For both classifiers, the performance initially improves with added features, but subsequently deteriorates as the feature subset increases in size. Since we collected only 10 maps of each locale, we limit the feature space to the five best features as determined by sequential forward feature selection.

We conclude that the MMD classifier outperforms the 3-NN classifier in room locale recognition. Using the feature set  $\{\mu_{00}^V, \mu_{00}^S, \varphi_4^S, \varphi_1^S, \varphi_2^S\}$  the MMD classifier yields a leave-one-out estimated recognition rate of 96%. Note that the zeroth-order central moments were chosen as the two best discriminating features. We infer that  $\mu_{00}^S$  is related to the size of the room, and that  $\mu_{00}^V$  is related to the amount of “detail”

Table 5.1. Feature selection and classification error of room locales: (a) 3-NN classifier, (b) MMD classifier.

(a)				(b)			
3-NN				MMD			
No.	Feature	$\tau$	$\hat{\epsilon}_u$	No.	Feature	$\tau$	$\hat{\epsilon}_u$
1	$\mu_{00}^V$	29	0.29	1	$\mu_{00}^V$	26	0.26
2	$\mu_{00}^S$	7	0.07	2	$\mu_{00}^S$	8	0.08
3	$\varphi_3^S$	5	0.05	3	$\varphi_4^S$	6	0.06
4	$\varphi_5^V$	5	0.05	4	$\varphi_1^S$	5	0.05
5	$\varphi_4^S$	7	0.07	5	$\varphi_2^S$	4	0.04
6	$\varphi_6^V$	8	0.08	6	$\varphi_3^S$	6	0.06
7	$\varphi_1^S$	8	0.08	7	$\varphi_6^S$	7	0.07
8	$\varphi_2^S$	7	0.07	8	$\varphi_5^S$	8	0.08
9	$\varphi_7^S$	7	0.07	9	$\varphi_1^V$	10	0.10
10	$\varphi_6^S$	6	0.06	10	$\varphi_4^V$	9	0.09
11	$\varphi_5^S$	6	0.06	11	$\varphi_3^V$	11	0.11
12	$\varphi_2^V$	7	0.07	12	$\varphi_7^S$	13	0.13
13	$\varphi_3^V$	6	0.06	13	$\varphi_2^V$	15	0.15
14	$\varphi_1^V$	6	0.06	14	$\varphi_5^V$	17	0.17
15	$\varphi_4^V$	7	0.07	15	$\varphi_7^V$	17	0.17
16	$\varphi_7^V$	8	0.08	16	$\varphi_6^V$	19	0.19

in the room. Additionally, the moment invariants calculated from the sonar grid maps were counted as more significant than those from the vision grid maps. Figure 5.1 shows pair-wise feature projections of the data using this feature set.

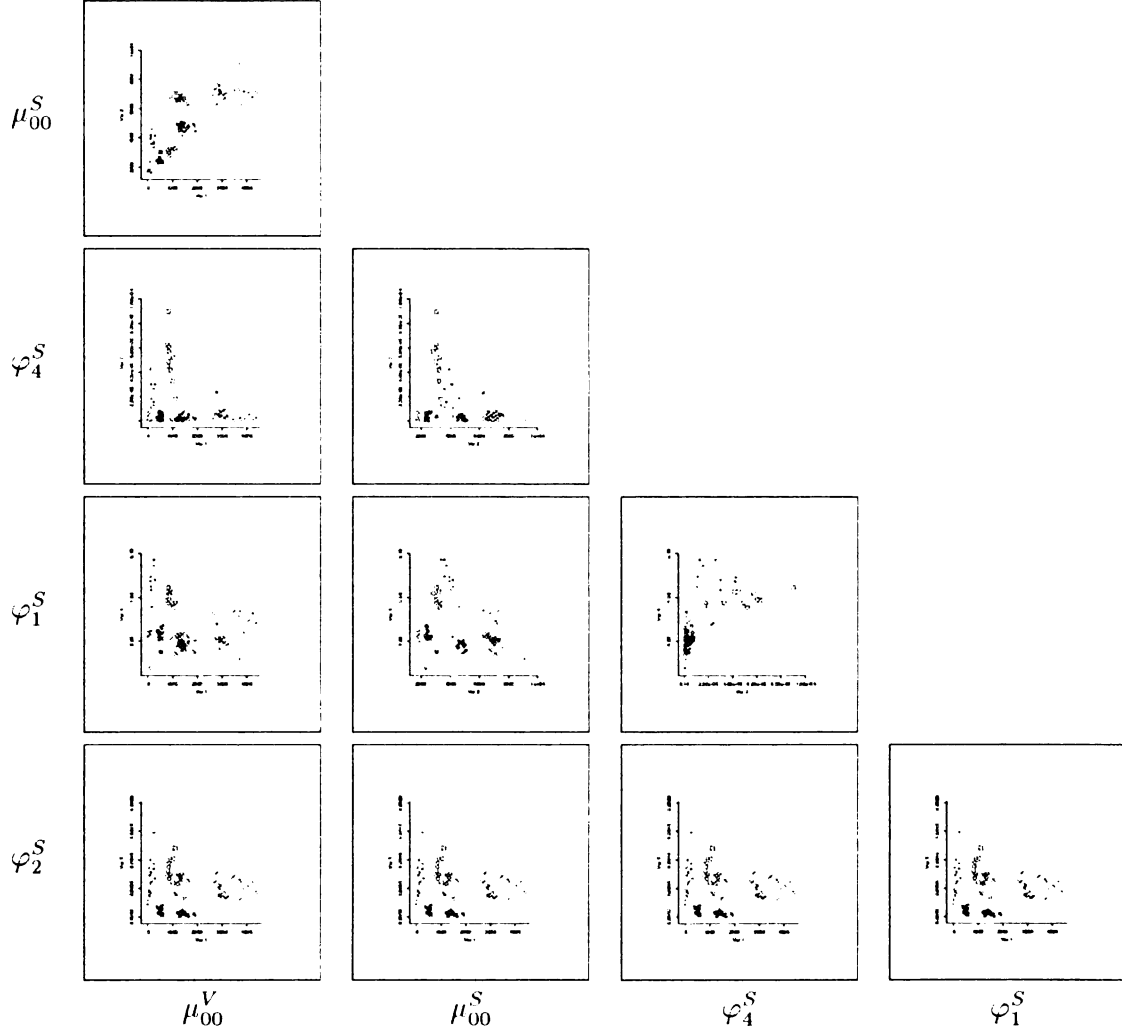


Figure 5.1. Pair-wise feature scatter plots of room locale data. Red denotes EB 312; green, EB 323; yellow, EB 325; orange, EB 327; yellow-green, EB 330; pink, EB 335; light blue, EB 350; dark blue, EB 352; orchid, EB 354; peru, EB 360.

Table 5.2 shows the confusion matrix resulting from the MMD classifier and the

Table 5.2. Confusion matrix of room locale data resulting from MMD classifier using features  $\mu_{00}^V, \mu_{00}^S, \varphi_4^S, \varphi_1^S$ , and  $\varphi_2^S$ . The misclassifications are underlined.

	1	2	3	4	5	6	7	8	9	10
1	10	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0
5	0	0	0	0	9	0	<u>1</u>	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	<u>1</u>	0	9	0	0	0
8	0	0	<u>1</u>	0	0	0	0	9	0	0
9	0	0	0	0	0	0	0	0	10	0
10	0	<u>1</u>	0	0	0	0	0	0	0	9

Table 5.3. The numbering of the locales, used for both room and door confusion matrices.

1	EB 312	3	EB 325	5	EB 330	7	EB 350	9	EB 354
2	EB 323	4	EB 327	6	EB 335	8	EB 352	10	EB 360

5-dimensional room locale feature space—the locales are numbered as indicated in Table 5.3. Here we see that room EB 330 was misclassified once as room EB 350 and vice-versa. Likewise, EB 352 was misclassified once as EB 325, and EB 360 as EB 323. The sensor grid maps shown in Appendix A indicate that the misclassifications are a result of the similar size and structure of these rooms.

We also applied the 3-NN and MMD classifiers to the door locale dataset. Table 5.4(a) reports the results of sequential forward feature selection using the 3-NN classifier and 24 available door locale features. Again, the data was normalized beforehand. Table 5.4(b) gives the feature subsets selected under the MMD classifier; we assumed uncorrelated features as before. Note that features calculated from the IR grid maps play a significant role in both classifiers, while those calculated from the vision maps

are not ranked as important. We reason that this is due to the “confined” nature of these locales.

Considering the best subset of five features reported in Table 5.4, we see that the 3-NN classifier performs the best for door locale recognition. With the feature set  $\{\varphi_7^I, \varphi_7^S, \mu_{00}^I, \varphi_2^S, \mu_{00}^V\}$ , the leave-one-out method gives an estimated recognition rate of 99%. Thus, the classifier can identify the robot’s location based on the doorway locale alone! Although the area of the mapped regions is small, the scale-invariance property of the moment invariants has the effect of “magnifying” the structures in each sensor map. Furthermore, based on the inferior results of the MMD classifier for this dataset, we conclude that either an insufficient number of patterns was available to estimate the class-conditional density parameters, or that the door locale data is not well represented by a Gaussian distribution. Projections of the data using this feature set are shown in Figure 5.2.

As before, the confusion matrix resulting from the 3-NN classifier under the 5-dimensional door locale feature space is given in Table 5.5. The door locale for EB 354 was misclassified once as the door to EB 325. Comparing this confusion matrix with the room locale confusion matrix of Table 5.2, we see that the location of the robot, as decided by the two classifiers, is in agreement 95 times out of 100. There were no cases where the door and room locale classifiers both gave the location of the robot incorrectly (i.e., for doorway and interior maps of the same room).

Finally, we used bootstrap error estimation techniques to obtain a more reliable evaluation of classifier performance for the two datasets. Table 5.6(a) reports the  $E632$  error estimate for the MMD classifier with room locale data as 6.26%. Likewise, Table 5.6(b) gives  $E632$  for the 3-NN classifier with door locale data as 2.50%. These correspond to recognition rates of 93.74% for room locales and 97.50% for door locales.

Table 5.4. Feature selection and classification error of door locales: (a) 3-NN classifier, (b) MMD classifier.

(a)				(b)			
3-NN				MMD			
No.	Feature	$\tau$	$\hat{\epsilon}_u$	No.	Feature	$\tau$	$\hat{\epsilon}_u$
1	$\varphi_7^I$	39	0.39	1	$\varphi_3^S$	44	0.44
2	$\varphi_7^S$	19	0.19	2	$\varphi_7^S$	21	0.21
3	$\mu_{00}^I$	9	0.09	3	$\varphi_1^I$	12	0.12
4	$\varphi_2^S$	2	0.02	4	$\varphi_2^S$	9	0.09
5	$\mu_{00}^V$	1	0.01	5	$\varphi_5^I$	5	0.05
6	$\varphi_1^S$	1	0.01	6	$\varphi_1^S$	5	0.05
7	$\varphi_2^I$	0	0.00	7	$\mu_{00}^S$	7	0.07
8	$\varphi_3^S$	0	0.00	8	$\varphi_4^S$	6	0.06
9	$\varphi_4^S$	0	0.00	9	$\mu_{00}^I$	5	0.05
10	$\varphi_5^S$	0	0.00	10	$\varphi_5^S$	6	0.06
11	$\mu_{00}^S$	0	0.00	11	$\varphi_4^I$	6	0.06
12	$\varphi_6^S$	0	0.00	12	$\varphi_6^S$	7	0.07
13	$\varphi_1^I$	0	0.00	13	$\varphi_2^I$	7	0.07
14	$\varphi_3^I$	0	0.00	14	$\varphi_3^I$	8	0.08
15	$\varphi_4^I$	0	0.00	15	$\varphi_7^I$	8	0.08
16	$\varphi_5^I$	0	0.00	16	$\varphi_6^I$	9	0.09
17	$\varphi_6^I$	0	0.00	17	$\mu_{00}^V$	10	0.10
18	$\varphi_1^V$	1	0.01	18	$\varphi_1^V$	10	0.10
19	$\varphi_2^V$	1	0.01	19	$\varphi_2^V$	10	0.10
20	$\varphi_5^V$	1	0.01	20	$\varphi_4^V$	12	0.12
21	$\varphi_7^V$	1	0.01	21	$\varphi_6^V$	13	0.13
22	$\varphi_3^V$	2	0.02	22	$\varphi_3^V$	14	0.14
23	$\varphi_4^V$	3	0.03	23	$\varphi_5^V$	15	0.15
24	$\varphi_6^V$	3	0.03	24	$\varphi_7^V$	16	0.16

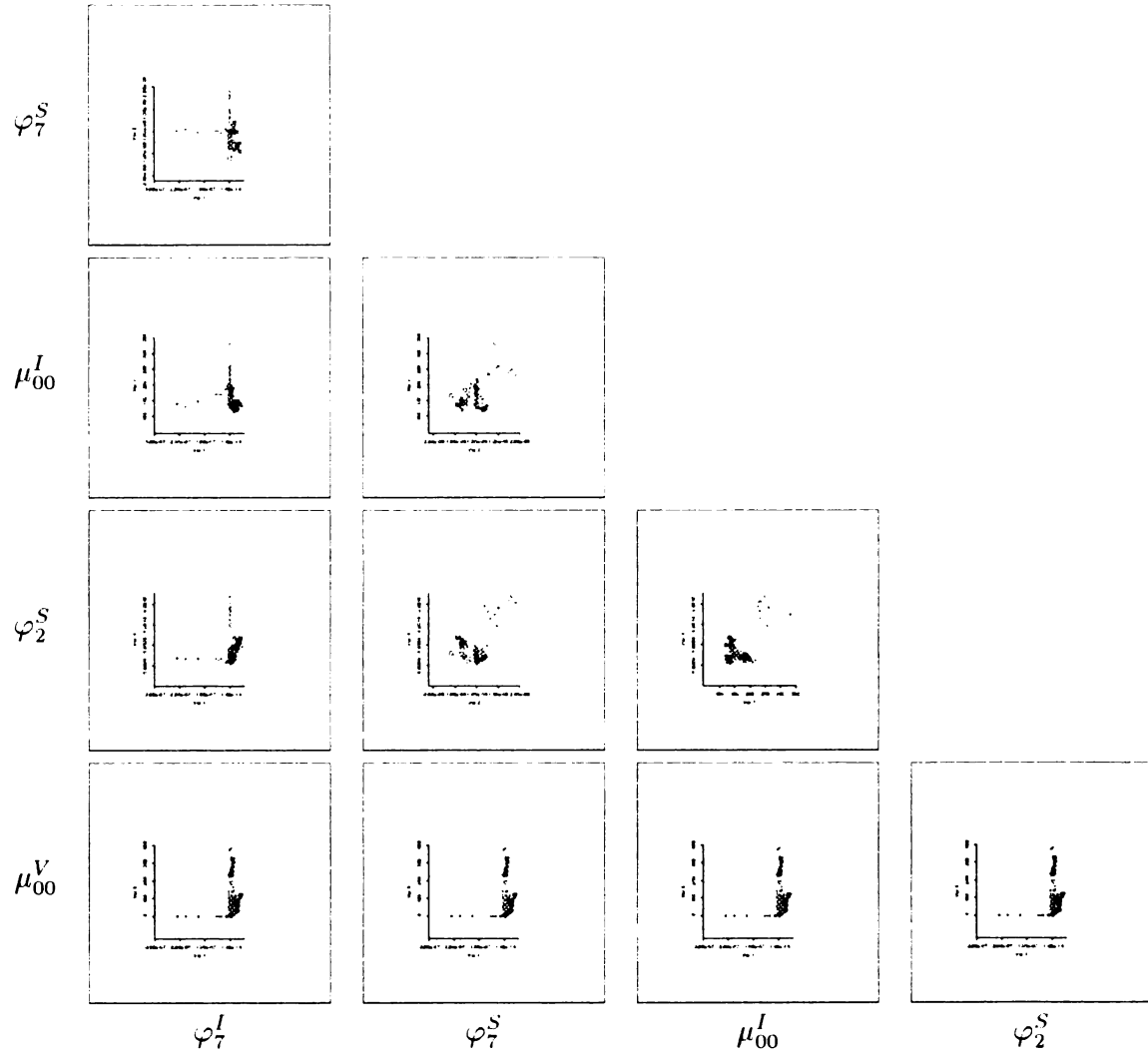


Figure 5.2. Pair-wise feature scatter plots of door locale data. Colors are assigned identically as in Figure 5.1.

Table 5.5. Confusion matrix of door locale data resulting from 3-NN classifier using features  $\varphi_7^I, \varphi_7^S, \mu_{00}^I, \varphi_2^S$ , and  $\mu_{00}^V$ . The misclassifications are underlined.

	1	2	3	4	5	6	7	8	9	10
1	10	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	10	0	0
9	0	0	<u>1</u>	0	0	0	0	0	9	0
10	0	0	0	0	0	0	0	0	0	10

Table 5.6. Bootstrap error estimation for the best classifier of each dataset: (a) room locales, (b) door locales.

Dataset	Classifier	Features	$\hat{\epsilon}_{E632}$
Room	MMD	$\mu_{00}^V, \mu_{00}^S, \varphi_4^S, \varphi_1^S, \varphi_2^S$	0.0626

(a)

Dataset	Classifier	Features	$\hat{\epsilon}_{E632}$
Door	3-NN	$\varphi_7^I, \varphi_7^S, \mu_{00}^I, \varphi_2^S, \mu_{00}^V$	0.0250

(b)



# CHAPTER 6

## Conclusions

In this thesis, we have presented a methodology for coarse position estimation of a mobile robot within an indoor setting. Important regions of the robot workspace (locales) are represented using grid-based maps collected during the exploration phase. In the navigation phase, robot localization is performed by identifying maps of known regions. Thus, position estimation is posed as a pattern classification problem.

We generated experimental datasets representing ten different rooms and doorways in the third floor of the Engineering Building by mapping each locale ten times. Each map in the dataset consisted of independent sensor grid maps made from three sensing sources: ultrasonic range sensors, lateral motion vision, and infrared proximity detectors. We calculated the zeroth-order central moment and seven moment invariants from the sensor grid maps, resulting in 24 representative features for each locale.

Finally, we experimented with various pattern classifiers to perform locale recognition. Using the 24 features extracted from the sensor grid maps, we trained and tested 3-nearest-neighbor (3-NN) and minimum Mahalanobis distance (MMD) classifiers. We estimate that with a 5-dimensional feature space, the MMD classifier can recognize the room locales with a 94% success rate, and that the 3-NN classifier can recognize the door locales with a 98% success rate. Furthermore, we found that the

location of the robot, as decided by the door and room locale classifiers, was rarely in disagreement. We conclude that indoor coarse-level localization is possible using these pattern classifiers.

## 6.1 Contributions

Major contributions of this thesis have been:

1. A spatial representation for mobile robot navigation that combines grid-based maps with topological relations.
2. A framework for coarse-level mobile robot localization using pattern classification techniques.
3. Application of lateral motion vision and epipolar plane image analysis to grid-based map creation.
4. Integration of three common sensing sources (i.e., sonar, vision, IR proximity detection) for sensor-based spatial representations.
5. A feature-level approach to sensor fusion from registered grid maps using spatial moments and moment invariants.
6. A method for recognition of grid-based maps that is invariant under translation and rotation.

## 6.2 Conclusions

We conclude:

- Sonar, vision, and infrared proximity detection provide different but corroborating spatial information that is useful in mobile robot navigation.

- Grid-based maps are an adequate spatial representation to solve the “where-am-I” problem of indoor navigation.
- Coarse position estimation is possible for mobile robots via classification of grid-based maps.

## 6.3 Future Work

We have identified three areas in which the work presented in this thesis could be extended or applied:

**Utilization of topological relations.** We did not use the topological relations between the locales in our experimental database to aid locale recognition. We reason that such information could be used to determine *a priori* probabilities for each locale, for use in a Bayes classifier (e.g., Equation (5.2)) or a class-weighted K-NN classifier. For example, if a locale is behind the robot, the *a priori* probability that the robot will soon encounter that locale should be lower than for those locales ahead of the robot. Likewise, two spatially adjacent locales should have similar *a priori* probabilities. A method should be developed to rate these *a priori* probabilities based on the topological location of the robot within the workspace.

**Recognition of “generic” hallway locales.** In addition to identifying door and room locales, our approach could be applied to recognition of recurring (or “generic”) hallway locales. Examples of such locales are right-angle turns, T-junctions, intersections, dead ends, and open or closed doorways. We propose that these locales be identified through classification of grid-based maps of regions in the vicinity of the robot. Figure 6.1 shows a robot navigating through a hallway. Five “sliding windows” are defined at the front and sides of the

robot, from which grid-based maps are continuously collected. Classification of the maps from the individual windows could be used to identify the characteristics of each region near the robot, and a rule-based recognition system could be used to integrate the results and form a hypothesis regarding the type of generic locale present. Here, topological relations between instances of generic locales would be essential to assist the robot in localizing itself in the hallway environment.

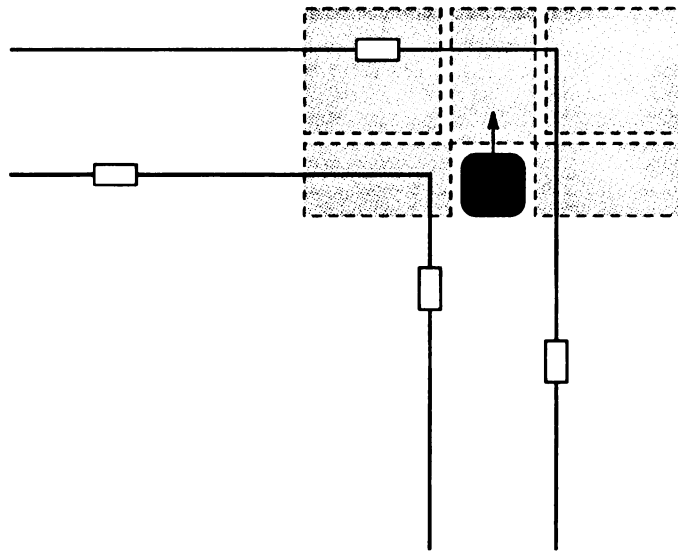


Figure 6.1. Sliding windows around a mobile robot navigating in a hallway environment. Both the doors and the turn in the hallway could be considered generic hallway locales.

### **Incorporation of coarse-level localization in a robot navigation system.**

Finally, coarse-level localization should be incorporated into a mobile robot navigation system wherein components for goal planning, path planning, and obstacle avoidance are also included. This would provide the navigation system with position information for use in goal and path planning. Figure 6.2 depicts the role of coarse-level localization in a navigation system.

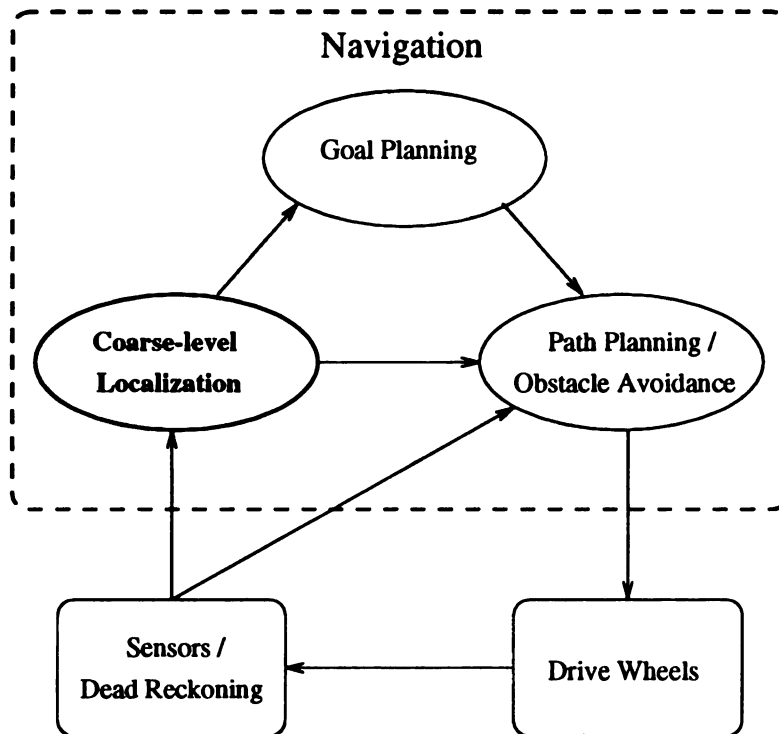


Figure 6.2. Coarse-level localization and its relation to other components of a robot navigation system. The arrows indicate flow of information in the logical system.

# **APPENDICES**

# APPENDIX A

## Experimental Datasets

Following are example grid maps of the locales used in our experiments. Ten maps were collected from each locale at different times and under differing exploration paths.

### A.1 Room Locales

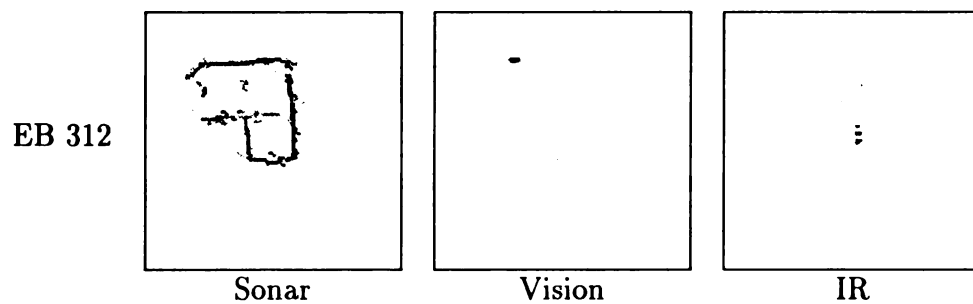


Figure A.1. Example sensor grid maps of room locale EB 312.

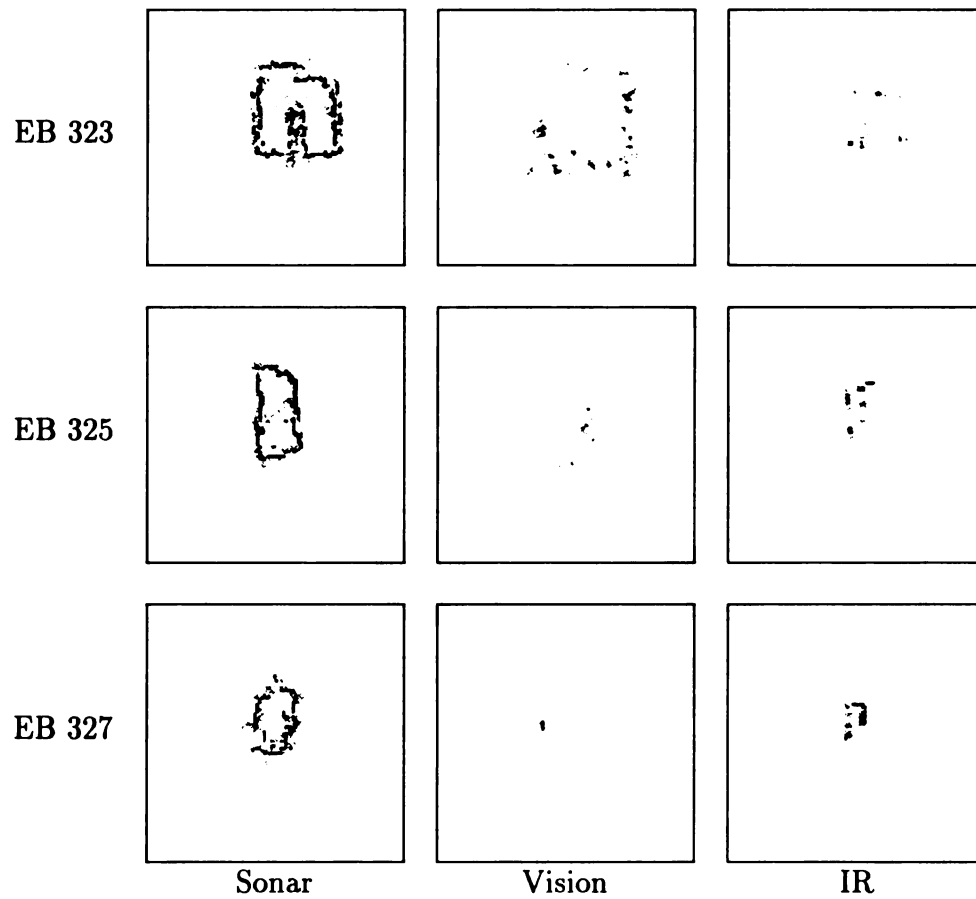


Figure A.2. Example sensor grid maps of room locales EB 323, EB 325, and EB 327.



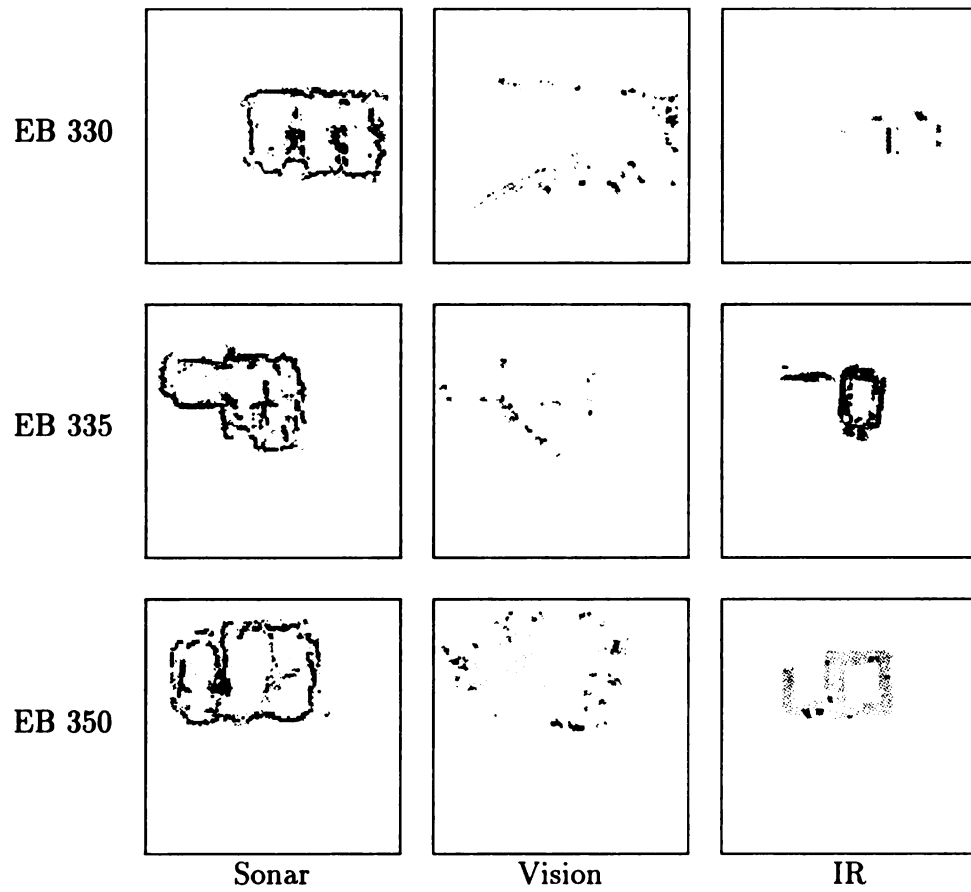


Figure A.3. Example sensor grid maps of room locales EB 330, EB 335, and EB 350.

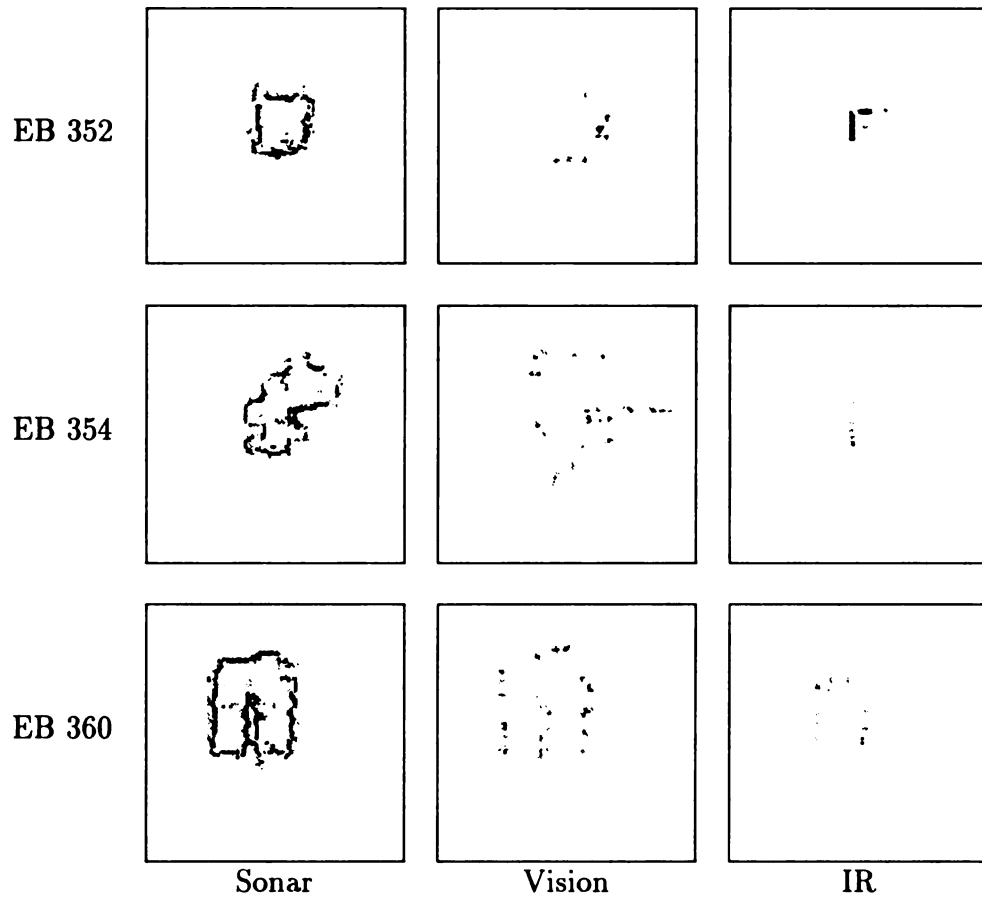


Figure A.4. Example sensor grid maps of room locales EB 352, EB 354, and EB 360.

## A.2 Door Locales

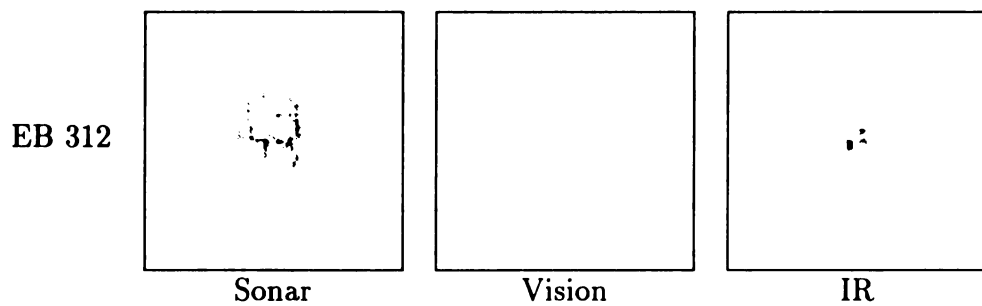


Figure A.5. Example sensor grid maps of door locale EB 312.

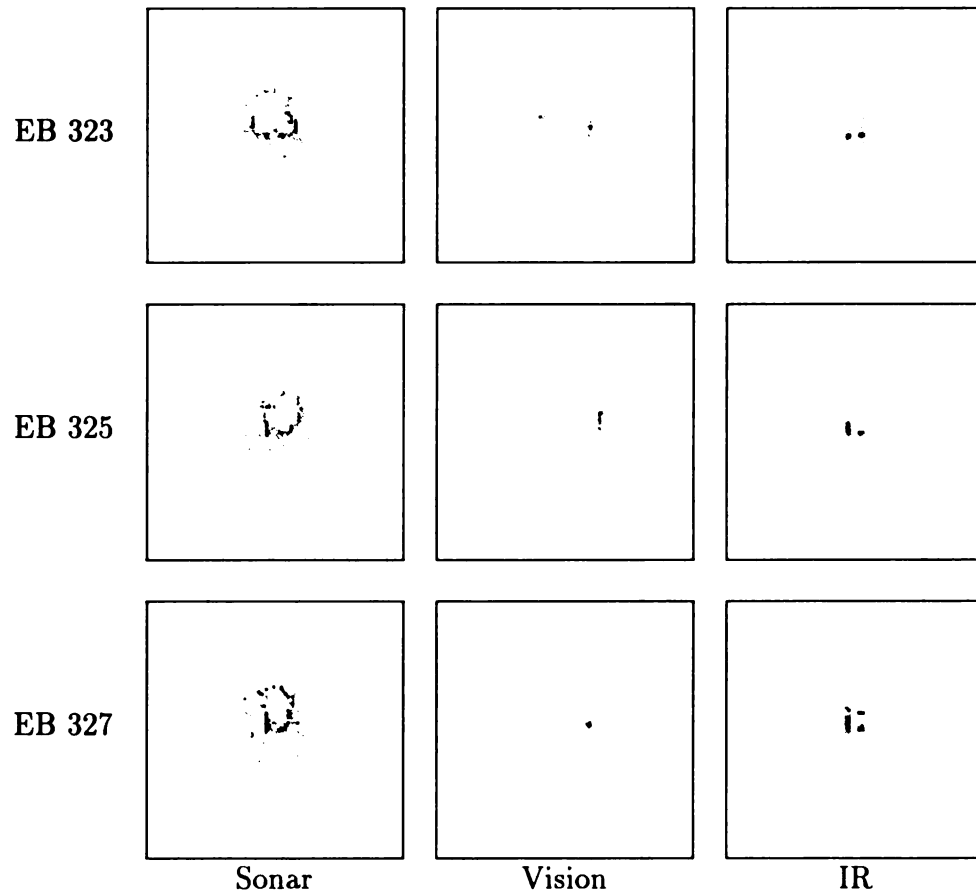


Figure A.6. Example sensor grid maps of door locales EB 323, EB 325, and EB 327.

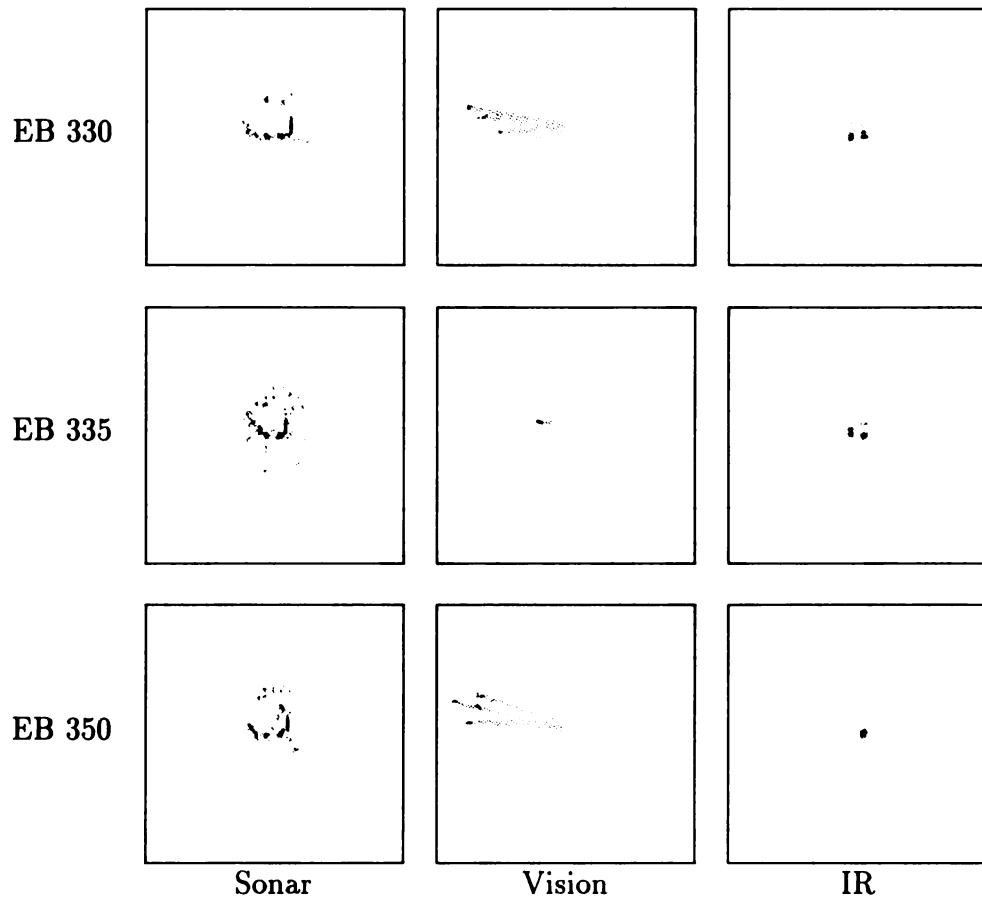


Figure A.7. Example sensor grid maps of door locales EB 330, EB 335, and EB 350.

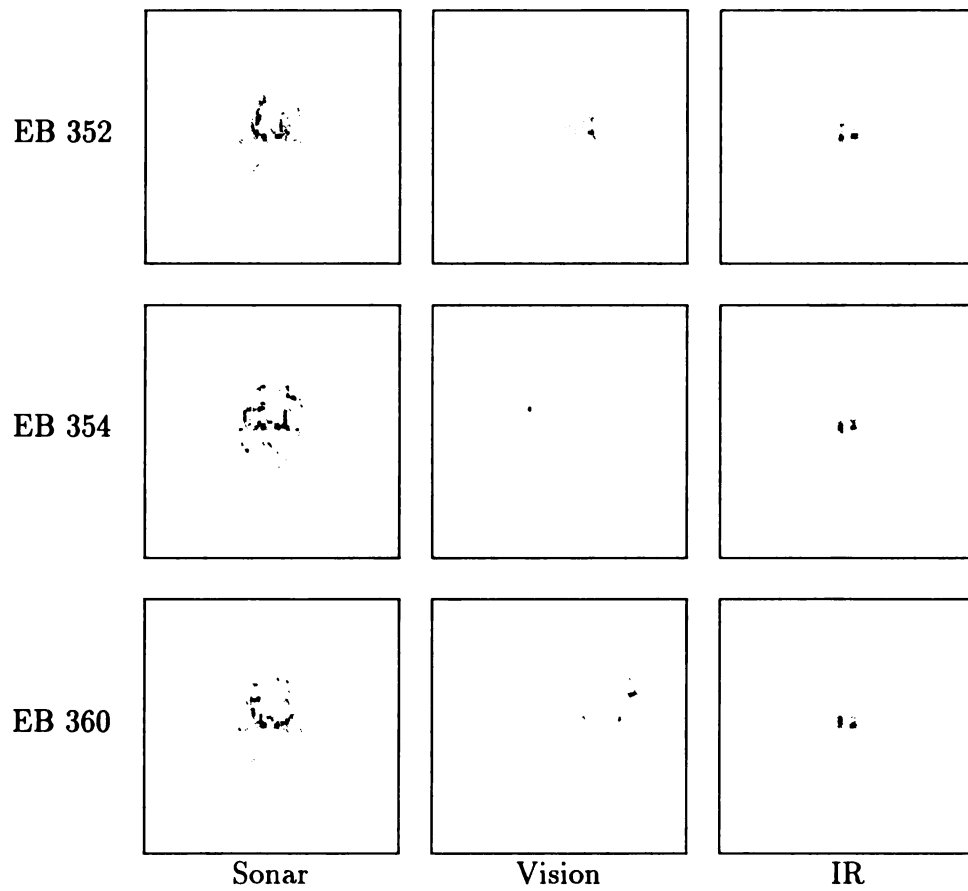


Figure A.8. Example sensor grid maps of door locales EB 352, EB 354, and EB 360.

# APPENDIX B

## ROME Cartographer

*ROME Cartographer* is the robot control and grid map acquisition program used in our experiments. It creates sonar, vision, and IR sensor grid maps of 100 by 100 cells from workspace regions 20m by 20m in area.

Figure B.1 shows the interface to *ROME Cartographer*. The following sections describe the interface on a component-by-component basis.

### B.1 Mouse Buttons

Robot motion may be controlled via the three mouse buttons within the three sub-window canvases.

**Left.** Pressing the left mouse button rotates the robot to the left. Releasing the button stops the motion. Pressing (and releasing) the button while the Control key is down causes the robot to rotate 90 degrees to the left. Pressing the button while the Caps Lock key is engaged causes the robot to rotate continuously to the left. Releasing the Caps Lock key stops the motion.

**Middle.** Pressing the middle mouse button moves the robot forward. Releasing the button stops the motion. Pressing the button while the Control key is down causes the robot to move backwards. Pressing the button while the Caps Lock

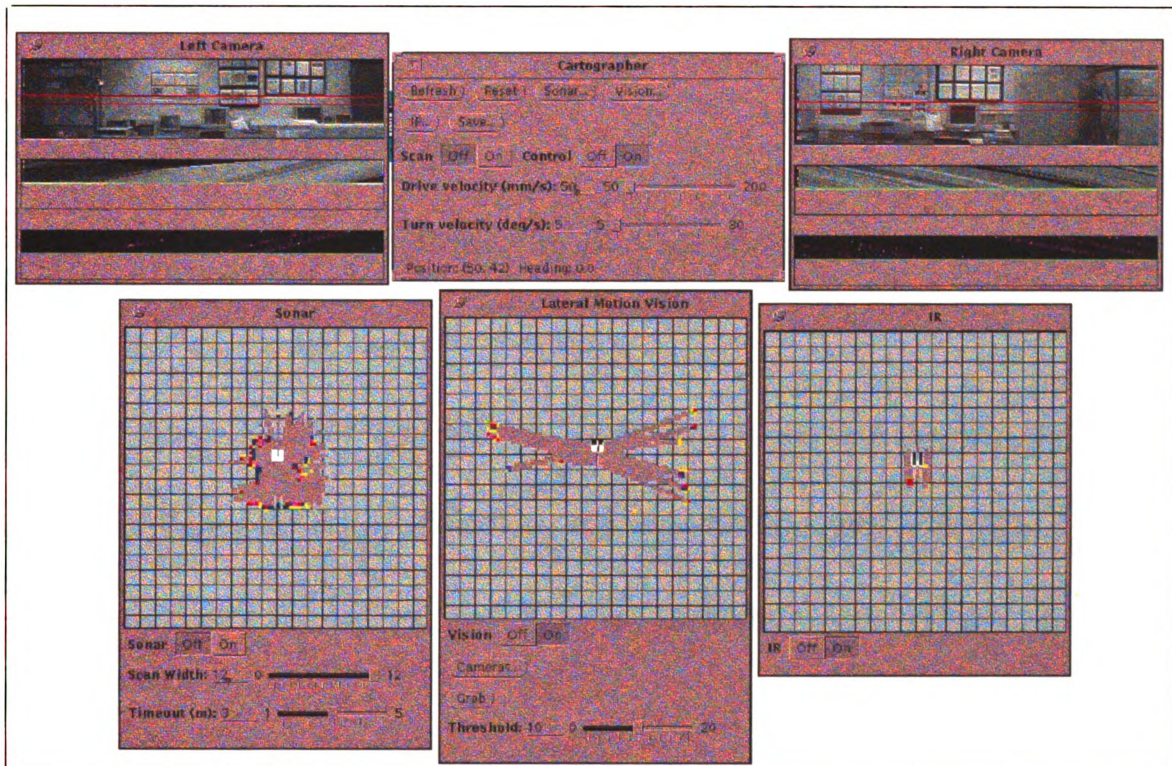


Figure B.1. ROME Cartographer interface.

key is engaged causes the robot to move forward continuously. Releasing the Caps Lock key stops the motion.

**Right.** Pressing the right mouse button rotates the robot to the right. Releasing the button stops the motion. Pressing (and releasing) the button while the Control key is down causes the robot to rotate 90 degrees to the right. Pressing the button while the Caps Lock key is engaged causes the robot to rotate continuously to the right. Releasing the Caps Lock key stops the motion.

## B.2 Main Window

The main window controls the program subwindows and mobile robot base.

**Refresh.** Redraws the subwindow canvases.

**Reset.** Resets *ROME Cartographer* to initial state, discarding any mapping data.

**Sonar...** Raises the Sonar subwindow, and initiates sonar data collection.

**Vision...** Raises the Lateral Motion Vision subwindow, and initiates vision data collection.

**IR...** Raises the IR subwindow, and initiates infrared data collection.

**Save...** Raises the Save subwindow, from which grid maps may be saved to disk. Maps are stored as 3-frame, byte format HIPS images.

**Scan.** Toggles data collection on and off.

**Control.** Toggles robot control on and off. If robot control is off, robot motion commands from the mouse only affect the logical position of the robot (in the map), rather than moving the robot as well.



**Drive velocity.** Sets the forward/backward velocity in mm/sec.

**Turn velocity.** Sets the turn velocity in degrees/sec.

**Position.** Reports the position and heading of the robot within the grid map.

## B.3 Sonar Subwindow

The sonar subwindow monitors and controls ultrasonic range sensor grid map creation.

**Canvas.** Monitors grid map creation. Cell certainty values are mapped to the following colors: yellow, red, magenta, and blue, where yellow is least likely occupied and blue is most likely occupied. Each cell corresponds to a 20cm by 20cm region in the workspace. Within the canvas, the three mouse buttons may be used for robot motion control.

**Sonar.** Toggles sonar sensing on and off.

**Scan width.** Sets the number of active sensors, starting from the front of the robot.

A scan width of zero corresponds to no active sensors; a scan width of 1 means the frontmost sensor is active; a scan width of 2 means the front three sensors are active, etc. A scan width of 12 corresponds to all 24 sensors active.

**Timeout.** Sets the sonar timeout in meters.

## B.4 Lateral Motion Vision Subwindow

The lateral motion vision subwindow monitors and controls vision grid map creation.

**Canvas.** Monitors grid map creation. Cell certainty values are mapped to the following colors: yellow, red, magenta, and blue, where yellow is least likely occupied and blue is most likely occupied. Each cell corresponds to a 20cm by 20cm

region in the workspace. Within the canvas, the three mouse buttons may be used for robot motion control.

**Vision.** Toggles vision sensing on and off.

**Cameras...** Raises the left and right camera subwindows.

**Grab.** Grabs an image from each camera, and displays them in the camera subwindows.

**Threshold.** Sets the number of features to track in the left and right images.

## B.5 IR Subwindow

The IR subwindow monitors and controls infrared proximity detector grid map creation.

**Canvas.** Monitors grid map creation. Cell certainty values are mapped to the following colors: yellow, red, magenta, and blue, where yellow is least likely occupied and blue is most likely occupied. Each cell corresponds to a 20cm by 20cm region in the workspace. Within the canvas, the three mouse buttons may be used for robot motion control.

**IR.** Toggles IR sensing on and off.

## B.6 Camera Subwindows

The camera subwindows monitor the CCD cameras used for vision grid map creation.

**Top canvas.** Shows the middle third of the video image from the corresponding camera. New images are displayed upon initialization, pressing the Grab button,

or stopping the robot. The horizontal red lines indicate the scanlines that are averaged to create 1-D images for the EPI.

**Middle canvas.** Shows the creation of the EPI as 1-D images are concatenated. The horizontal green line shows where the next 1-D image will be added; the image wraps from bottom to top.

**Bottom canvas.** Shows the features tracked through the EPI. Traces colored magenta have been tracked successfully; those that are white have been abandoned. The horizontal green line shows the progress of the tracking algorithm; the image wraps from bottom to top.

# **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] Nicholas Ayache and Oliver D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, December 1989.
- [2] Harlyn H. Baker and Robert C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–9. Computer Society Press, June 1988. Ann Arbor, Michigan.
- [3] Dana H. Ballard and Christopher M. Brown. *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
- [4] Kenneth Basye, Thomas Dean, and Jeffrey Scott Vitter. Coping with uncertainty in map learning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 663–668, August 1989.
- [5] Richard A. Becker and John M. Chambers. *S: An Interactive Environment for Data Analysis and Graphics*. Wadsworth, Inc., Belmont, California, 1984.
- [6] Martin Beckerman and E. M. Oblow. Treatment of systematic errors in the processing of wide-angle sonar sensor data for robotic navigation. *IEEE Transactions on Robotics and Automation*, 6(2):137–145, April 1990.
- [7] Donald A. Berry and Bernard W. Lindgren. *Statistics: Theory and Methods*. Brooks/Cole Publishing Company, Pacific Grove, California, 1990.
- [8] Robert C. Bolles, Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.
- [9] Johann Borenstein and Yoram Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4):535–539, August 1991.

- [10] Johann Borenstein and Yoram Koren. The vector field histogram—fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, June 1991.
- [11] Rodney A. Brooks, Anita M. Flynn, and Thomas Marill. Self calibration of motion and stereo vision for mobile robot navigation. A.I. Memo 984, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, August 1987.
- [12] Bruce A. Carlson. *Communications Systems*. McGraw-Hill, third edition, 1986.
- [13] Polaroid Corporation. Ultrasonic range finders, 1982.
- [14] Transitions Research Corporation. Labmate autonomous mobile robot base. 15 Great Pasture Road, Danbury, Connecticut 06810, 1989.
- [15] T. M. Cover. The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man, and Cybernetics*, 4:116–117, 1974.
- [16] T. M. Cover and J. M. Van Campenhout. On the possible orderings in the measurement selection problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:657–661, 1977.
- [17] Ingemar J. Cox. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, April 1991.
- [18] James L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 674–680, Scottsdale, Arizona, May 1989.
- [19] Persi Diaconis and Bradley Efron. Computer-intensive methods in statistics. *Scientific American*, pages 116–130, 1981.
- [20] Michael Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):325–332, March 1987.
- [21] Richard C. Dubes and Anil K. Jain. Notes for CPS 806, Pattern Recognition. Computer Science Department, Michigan State University, East Lansing, Michigan 48823, Spring 1991.
- [22] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

- [23] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, December 1991.
- [24] Bradley Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78:316–331, 1983.
- [25] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Transactions on Robotics and Automation*, RA-3(3):249–265, June 1987.
- [26] Alberto Elfes. *Occupancy Grids: A Probabilistic Framework for Mobile Robot Perception and Navigation*. PhD thesis, Carnegie Mellon University, 1989.
- [27] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, pages 46–57, June 1989.
- [28] Claude Fennema, Allen Hanson, Edward Riseman, J. Ross Beveridge, and Rakesh Kumar. Model-directed mobile robot navigation. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1352–1369, December 1990.
- [29] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.
- [30] E. Fix and J. L. Hodges, Jr. Discriminatory analysis, nonparametric discrimination: Small sample performance. Report, USAF School of Aviation Medicine, Randolph Field, Texas, August 1952.
- [31] W. E. L. Grimson and T. Lozano-Pérez. Recognition and localization of overlapping parts from sparse data in two and three dimensions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 61–66, St. Louis, Missouri, March 1985.
- [32] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8(2):179–187, February 1962.
- [33] King-Chu Hung, Chia-Nian Shyi, Jau-Yien Lee, and Tsong-Chiang Lee. Robot location determination in a complex environment by multiple marks. *Pattern Recognition*, 21(6):567–580, 1988.

- [34] A. K. Jain and B. Chandrasekaran. *Dimensionality and Sample Size Considerations in Pattern Recognition Practice*, volume 2, chapter 39, pages 835–855. North-Holland Publishing Company, 1982.
- [35] Anil K. Jain and Richard Dubes. Feature definition in pattern recognition with small sample size. *Pattern Recognition*, 10:85–97, 1978.
- [36] Anil K. Jain, Richard C. Dubes, and Chaur-Chin Chen. Bootstrap techniques for error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):628–633, September 1987.
- [37] H. M. Kalayeh and D. A. Landgrebe. Predicting the required number of training samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(6):664–667, November 1983.
- [38] Eric Krotkov. Mobile robot localization using a single image. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 978–983, Scottsdale, Arizona, 1989.
- [39] Benjamin J. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, February 1978.
- [40] Benjamin J. Kuipers. A robust, qualitative method for robot spatial learning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 774–779, Saint Paul, Minnesota, August 1988.
- [41] Benjamin J. Kuipers and Tod S. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, 9(2):25–43, Summer 1988.
- [42] John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.
- [43] John J. Leonard and Hugh F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, Norwell, Massachusetts, 1992.
- [44] Martin D. Levine. *Vision in Man and Machine*. McGraw-Hill, 1985.
- [45] Tod S. Levitt, Daryl T. Lawton, David M. Chelberg, and Philip C. Nelson. Qualitative landmark-based path planning and following. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 689–694, Seattle, Washington, July 1987.



- [46] Larry Matthies and Alberto Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 727–733, Philadelphia, Pennsylvania, 1988.
- [47] Larry Matthies and Steven A. Shafer. Error modeling in stereo navigation. *IEEE Transactions on Robotics and Automation*, RA-3(3):239–248, June 1987.
- [48] David Miller. A spatial representation system for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 122–127, St. Louis, Missouri, March 1985.
- [49] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, Summer 1988.
- [50] Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 116–121, St. Louis, Missouri, March 1985.
- [51] Anthony P. Reeves and Abdolrahim Rostampour. Shape analysis of segmented objects using moments. In *PRIP-81, Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pages 171–174, Dallas, TX, August 1981.
- [52] Kurt D. Rueb and Andrew K. C. Wong. Structuring free space as a hypergraph for roving robot path planning and navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):263–273, March 1987.
- [53] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information*. PhD thesis, Massachusetts Institute of Technology, 1988.
- [54] K. Sugihara. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing*, 42(1):112–129, April 1988.
- [55] Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, CA 94042-1100. *Using VideoPix*, 1991.
- [56] Transitions Research Corporation, 15 Great Pasture Road, Danbury, CT 06810. *Labmate User Manual*, version 5.211-e edition.
- [57] Transitions Research Corporation, 15 Great Pasture Road, Danbury, CT 06810. *Proximity Subsystem User Manual*, release 4.6g edition.

- [58] A. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 20:100–103, September 1971.

MICHIGAN STATE UNIV. LIBRARIES



31293008810917