



3 1293 01399 2338

THESIS

2

(1994)

This is to certify that the

dissertation entitled

AN AUTONOMOUS MOBILE ROBOT SYSTEM WITH
ADAPTIVE NAVIGATION STRATEGY AND
VISION-BASED MOTION PLANNING

presented by

Cheng-Chih Lin

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering

Major professor

Date 11/29/94

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
1 OCT 00 2005 121605	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

**AN AUTONOMOUS MOBILE ROBOT SYSTEM WITH
ADAPTIVE NAVIGATION STRATEGY AND
VISION-BASED MOTION PLANNING**

By

Cheng-Chih Lin

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1994

ABSTRACT

AN AUTONOMOUS MOBILE ROBOT SYSTEM WITH ADAPTIVE NAVIGATION STRATEGY AND VISION-BASED MOTION PLANNING

By

Cheng-Chih Lin

Autonomous mobile robots have drawn much attention in both academic research and industrial applications because of their intelligent behavior and versatility. These robots utilize various types of sensors to “perceive” their environments, and use it to perform motion planning. This dissertation addresses the problems of sensor-based navigation in typical manufacturing environments that are structured, partially known, and dynamic. The first half of this work focuses on an adaptive navigation strategy, where the speed and accuracy constraints during a navigation process are treated as functions of the changes in the robot’s environment. The second half introduces a new vision-based mobile robot motion planning system which can be integrated with the dead-reckoning method to achieve accurate goal acquisition. At the end, A hybrid navigation approach is developed to exploit the flexibility of adaptive navigation strategy and the advantages of vision-based motion control. This work can be summarized as follows:

1. Proposed a *Weighted Obstacle Density Function (WODF)* for the quantitative measurement of the clutteriness of the robot’s workspace.

2. Developed a quantitative measurement of the quality of general map construction algorithms, namely the *Match Indices*.
3. Developed the *Adaptive Navigation Strategy (ANS)* for a mobile robot. This approach is based on adjusting sensor configuration and motion speed of the robot according to the clutteriness of the environment.
4. Developed a vision-based self-calibration system using circular disk landmarks. Designed a visual-servo motion control system for mobile robot docking operation.
5. Proposed a *hybrid navigation system* to integrate the ANS and the vision-based motion planning approach.

Copyright by
CHENG-CHIH LIN
1994

To my parents for their love and support

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	x
NOTATIONS	xii
1 Autonomous Mobile Robot Navigation	1
1.1 Defining the Robot Navigation Problem	2
1.2 Sensor-Based Robot Navigation	5
1.3 Motivation	6
1.4 Accuracy and Speed as Control Variables	7
1.5 Vision-Based Motion Planning: Self-Calibration and Docking motion ..	8
1.6 Organization of this Dissertation	9
2 Literature Background	11
2.1 Sensors for Mobile Robot Navigation	11
2.1.1 Sensor-Based Map Construction Using Occupancy Grids	13
2.1.2 Map Construction Using the Sonar Histogram	18
2.2 Vision-Based Mobile Robot Navigation	18
2.2.1 Self-Calibration/ Positioning Using Navigational Landmarks .	19
2.2.2 Tracking Motion Using Visual-Servo Control	20
2.3 Our Research Mobile Robot – SPARTA	22
2.4 Summary	23
3 Multi-Sensor Integration	26
3.1 Sensing Systems of SPARTA	26
3.2 Sensor Modeling and Occupancy Functions	28

3.3	Constructing World Map from Multiple Sensor Data	33
3.4	Errors in Sensing and Data Integration.....	43
3.5	Adaptive Sensing Strategy	51
3.6	Summary.....	53
4	Adaptive Navigation Strategy	55
4.1	The Sensing-Integration-Decision Cycles	55
4.2	Navigation Errors	61
4.2.1	Errors in Robot Position and Heading	61
4.2.2	Errors in Robot Perception	61
4.2.3	Human Responses: Some Cases	62
4.3	Obstacle Density Function	63
4.3.1	Discrete Obstacle Density Function	64
4.3.2	The Weighted Obstacle Density Function	65
4.3.3	SID Cycle Estimation Based on ODF	67
4.4	The Switching Algorithm.....	67
4.5	Summary.....	72
5	Robot Vision System	73
5.1	Characteristics	73
5.2	Imaging Geometry of a Vision System.....	73
5.3	Navigation Using Landmarks.....	75
5.4	Landmark Patterns and Recognition Strategies	76
5.5	Self-Positioning	79
5.5.1	Finding the Center of the Primary Disk	79
5.5.2	The Modified Elliptical Hough Transformation	82
5.5.3	LSSD Line-Fitting	83
5.5.4	MEHT Robustness Test	89
5.6	Error Analysis.....	94
5.6.1	Errors in Aspect Angle and Distance of Landmark	96
5.6.2	Errors in Perceived Disk Center	97

5.7	Tracking the Landmarks Using Visual Servo Control	100
5.8	Selecting Landmark Positions.	102
5.9	Vision-Based Navigation Planning.	107
5.10	Summary.	108
6	Systems Integration	111
6.1	High Level Navigational Constraints	111
6.2	The 3-Stage Hybrid Navigation Plan	112
6.3	System Architecture	114
6.4	Summary.	115
7	Summary and Conclusions	117
7.1	Sensor-Based Robot Navigation	117
7.1.1	The SID Cycle Model	118
7.1.2	Sensor Configuration and Obstacle Density Function	118
7.2	Navigational Errors.	119
7.3	Vision	119
7.3.1	Landmark-Based Self-Calibration	119
7.3.2	Landmark Tracking Using Visual-Servo Control	120
7.4	Systems Integration – The Hybrid Navigation Strategy.	121
7.5	Conclusions.	121
7.6	Recommendations for Future Research	122
	BIBLIOGRAPHY	124

LIST OF TABLES

Table 3.1	The sensor integration algorithm.	35
Table 4.1	The adaptive navigation based on ODF.	70
Table 5.1	The Modified Elliptical Hough Transformation (MEHT).	87
Table 5.2	Errors in the measured aspect angle of the landmark due to occlusion. . .	90
Table 5.3	Estimation of landmark projections using measured elliptic parameters. .	90

LIST OF FIGURES

Figure 1.1	The workspace of a mobile robot.	3
Figure 2.1	The geometry of sonar sensor model.	14
Figure 2.2	Update procedure for sonar histogram.	19
Figure 2.3	The SPARTA mobile robot.	25
Figure 3.1	Geometrical layout of the proximity sensors on SPARTA.	28
Figure 3.2	The interpretation of range data from an ultrasonic sensor.	29
Figure 3.3	3-D surface representation of the probability of occupancy derived from ultrasonic ranging data ($r = 30$ cm in Equation (3.1)).	31
Figure 3.4	The infrared detector output function for two different cases.	32
Figure 3.5	The 3-D surface representation of the probability function $O(\phi) = 1$. . .	32
Figure 3.6	The layout of workspace_1 and observation points: A, B, C, and D. . .	36
Figure 3.7	The SSD error surfaces for 4 observation points in workspace_1	50
Figure 3.8	Match indices as functions of the integration threshold values.	52
Figure 4.1	Two different cases in minimal obstacle-free range.	59
Figure 4.2	An example of WODF surface: the exact occupancy grid map and the corresponding WODF surface with robot heading = 60 degrees.	66
Figure 4.3	Experimental results of the Adaptive Navigation Strategy.	71
Figure 5.1	Point projection in a pinhole camera.	74
Figure 5.2	The geometry of vision system on SPARTA.	75
Figure 5.3	The landmark pattern used by SPARTA.	78
Figure 5.4	Using the secondary pattern to decide the aspect of the landmark.	78
Figure 5.5	Top views of disk projection model, (a) the general cases, and (b) when camera axis passes through the disk center.	80
Figure 5.6	The Modified Elliptical Hough Transformation with gradient information and the corresponding Hough array.	85
Figure 5.7	A sample landmark image.	91

Figure 5.8	The edge enhanced landmark image.	92
Figure 5.9	The peak value in the corresponding EHT array indicates the dimensions of the ellipse approximating the landmark projection.	93
Figure 5.10	The exact projection model of the primary landmark pattern.	95
Figure 5.11	The error surface of landmark aspect angle.	98
Figure 5.12	Errors in the perceived disk center position.	99
Figure 5.13	The block diagram of visual servo motion control system.	101
Figure 5.14	Robot trajectory during visual-servo tracking of a landmark (initial heading error = 8 degrees).	103
Figure 5.15	The robot's angular and linear speed profile during landmark tracking. .	104
Figure 5.16	Setting up navigational landmarks in a typical indoor environment.	106
Figure 5.17	Vision-based navigation: a docking motion example.	109
Figure 6.1	The position uncertainty grows as the robot moves in dead-reckoning. . .	114
Figure 6.2	The block diagram of SPARTA mobile robot system.	116

NOTATIONS

The following conventions and notations are used throughout the dissertation.

- The mobile robot is represented by $\mathbf{m}$, which is the position vector of the robot w.r.t. the world frame.
- The robot's workspace is denoted by W and is modeled as the Euclidean space R^N , where $N = 2$ or 3 .
- A vector is denoted by a boldfaced lower case letter, e.g. $\mathbf{j}$. The length of vector $\mathbf{j}$ is written as $\|\mathbf{j}\|$. The inner product of two vectors $\mathbf{i}$ and $\mathbf{j}$ is denoted by $\mathbf{i} \cdot \mathbf{j}$.
- A point in W is denoted by its coordinate, e.g. $p = (x, y)$ for R^2 . In some cases, it is easier to use vectorized representation for a point, e.g. $\mathbf{p} = \overrightarrow{(x, y)} = x\mathbf{i} + y\mathbf{j}$.
- If the workspace W is discrete, i.e. the *grid cells*, then the basic elements of W are cells. in this case, each cell can be written as $c = (u, v)$ where u and v are *cell indices*.
- A region in W , which represents a set of points (or cells, in the discrete cases), is denoted by upper case letter, e.g. O .

CHAPTER 1

Autonomous Mobile Robot Navigation

Autonomous mobile robots have drawn much attention in both academic research and industrial applications because of their intelligent potential and versatility. In fact, mobile robots have been used for many real world applications such as substitution of human workers in hazardous environments, material transportation in manufacturing environments, and general services in hospitals. These robots utilize various types of sensors to “perceive” their environments, and perform motion planning based on the information collected by these sensors. Compared to the traditional automatic guided vehicles (AGV) where the mobile bases are guided by fixed paths, mobile robots have the advantages of more flexibility and wider application horizon.

The application of mobile robots determines the type of workspace. In general, the robot needs the geometrical information about its workspace, as well as the obstacles present, to perform any motion planning. This information is usually cited as the *prior knowledge* about the workspace which must be accessible to the robot. In fact, all static obstacles in the workspace should be included in the robot’s prior knowledge. Apart from the prior knowledge about the environment, a mobile robot also requires certain control strategy in order to react properly to any changes in its environment. The motion control of a mobile robot, together with its navigation strategy, usually takes the form of control rules, for

example “turn right if anything shows up in the front left direction”. The capability of adapting to changes in the environment is one of the most important features of a mobile robot.

1.1 Defining the Robot Navigation Problem

The most frequently encountered task for a mobile robot is to move itself from the current position to other positions in its work space. The degree of complexity in a navigation problem depends on the following factors:

- (1.) the completeness of knowledge about the environment,
- (2.) the existence of moving obstacles in the environment,
- (3.) the criteria governing the design and execution of navigation plans, i.e. speed, error tolerance, etc.,

The first two factors represent the amount of intrinsic uncertainties associated with the workspace itself, while the third one stands for the efforts required to overcome the uncertainties which come from the interactions between the mobile robot and its environment.

In this dissertation, the frequently used terms *region* and *workspace* are formally defined as follows:

Definition 1.1 A *region* in the 2-D Euclidean space is a connected set of points in R^2 .

Definition 1.2 A mobile robot’s *workspace*, denoted by W , is a set of points in R^2 such that for each point in W , there exists a legal configuration of the robot.

In general, the workspace of a mobile robot can be partitioned into two sets of regions,

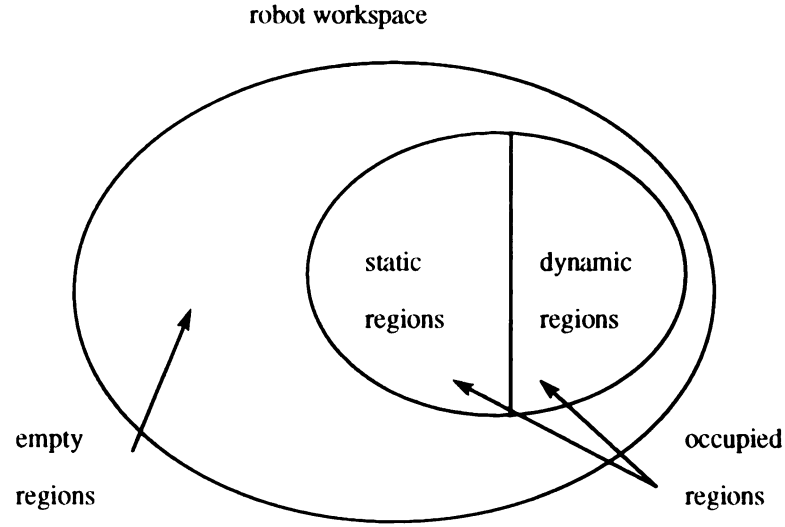


Figure 1.1. The workspace of a mobile robot.

namely the *occupied regions* $\mathcal{R}$, and the *empty regions* $\mathcal{R}$. The former represent the region occupied by the obstacles and the later are the regions within which the robot can freely travel. The occupied regions can be further categorized into *static* and *dynamic* regions. Each dynamic region corresponds to at least one moving obstacle. If there is no moving obstacle in a region, then that region is considered as *static*.

Definition 1.3 A dynamic workspace is a workspace containing at least one dynamic region. If a workspace is not dynamic, then it is a static workspace.

Figure 1.1 illustrates the partitioning of robot workspace into different types of regions.

In order to focus on the general applications of mobile robots in a manufacturing environment, this work will consider the class of partially known, semi-structured environments with dynamic obstacles. Such environments can be described as follows:

Given the geometry of

- the workspace W ,
- static obstacles $O_i, i = 1, \dots, m$,
- dynamic obstacles $D_j(t), j = 1, \dots, n$, for $t \geq 0$,

the set of occupied regions is

$$\mathfrak{R}(t) = \bigcup_{i=1}^m O_i + \bigcup_{j=1}^n D_j(t) \quad (1.1)$$

and the set of empty regions is

$$\mathfrak{K}(t) = W - \mathfrak{R}(t) \quad (1.2)$$

Therefore, a mobile robot's environment can be formally defined as follows:

Definition 1.4 An *environment* is a quadruple $\mathfrak{W} = [W, \mathfrak{R}(t), \mathfrak{K}(t), t]$ for $t \geq 0$ where W is the workspace, $\mathfrak{R}(t)$ is the set of occupied regions, and $\mathfrak{K}(t)$ is the set of empty regions at time t .

Given the definition above, the problem of mobile robot navigation can be formally stated as follows:

Definition 1.5 Given an environment $\mathfrak{W}$, the robot's current position $r(t)$, an initial position $s \in W$, and a goal position $g \in W$, find a control $\Gamma(\mathfrak{W}, s, g, r)$ such that

(Rule 1) $r(t_0) = s$ where t_0 is the initial time,

(Rule 2) $r(t_f) = g$ for some finite time t_f , and

(Rule 3) $r(t) \in \mathfrak{K}(t), \forall t \in [t_0, t_f]$.

Clearly, if $\mathcal{W}$ is completely known to the robot then the above problem reduces to the combination of path planning problem and path tracking problem. However, the problem becomes much more complex when there exist unknown obstacles in W , because part of the information required for off-line path planning is inaccessible to the robot. One feasible solution is to utilize various types of sensors to explore the environment as the robot moves toward its goal position. The sensor data provides realtime assessment of the environment which helps the robot in making decisions on-line. This approach is generally referred to as sensor-based robot navigation.

1.2 Sensor-Based Robot Navigation

Moving a mobile robot in a dynamic environment requires reliable sensory feedback and intelligent motion control. In general, there are three phases in sensor-based robot navigation:

- **sensing:** activate the sensors, convert the output signal of each sensor into numerical data.
- **integration:** combining the data from different sources into a unified representation of the environment.
- **decision:** using a path planning algorithm to generate / modify robot paths, decide the motion speed and the heading of the robot.

These phases are repeated until the navigation task is completed. We will use the term *Sensing-Integration-Decision* (SID) *cycle* to represent this cyclic operation throughout this dissertation.

1.3 Motivation

The integration phase, which is also called multi-sensor fusion, is a relatively new area in robotics research. One of the main reasons for using multi-sensor fusion is its potential for improved accuracy in robot perception. This is critical to achieve autonomous navigation in dynamic environments. On the other hand, the robot's speed performance is also critical for most navigation tasks. It is clear that these two criteria, namely the accuracy and the speed, are competing factors in a robot system.

To further clarify the relation between multi-sensor fusion and robot speed control in a partially known / dynamic environment, the following observations are made:

- The effective range of each type of sensor is fixed due to the physical limits. For example, the popularly used Polaroid ultrasonic range sensors have an effective range of approximately 3.4 m, which means any ranging information beyond that limit is not considered reliable.
- Multi-sensor fusion only improves the robot perception within the minimum effective range among all sensors involved. Any information beyond that range is either unknown or unreliable.
- The length of each sensing-integration-decision cycle depends on the number of sensors, and therefore the amount of data, involved in that cycle.
- To ensure safety during navigation, the robot must not move more than the effective perceptual range of the sensors before the next SID cycle is completed.
- From the sensing strategy point of view, the combination of sensors to use at any given time and place should be as flexible as the hardware permits. The amount of information collected and processed, which depends on the accuracy requirements, is therefore controllable.

These observations suggested an adaptive control system where the sensory feedback as well as the robot's speed can be adjusted on-line according to the constraints of accuracy and speed. The author is interested in finding a feasible solution to investigate the level of sensor configuration such that both the accuracy and speed requirements can be satisfied.

1.4 Accuracy and Speed as Control Variables

The constraints of accuracy and speed can be assigned during the initialization of each navigation process, or they can be formulated as system variables to be controlled by certain algorithm. In general, it is desirable from the flexibility point of view to allow these constraints to reflect the changes in the robot's environment. For example, the accuracy constraint should be dominant when the robot is sensing nearby obstacles or getting close to its docking position. On the other hand, if there is no "visible obstacle" around then the speed constraint should be the dominant factor.

Inspired by human response to the changes in environment, the constraints of navigation can be formulated as functions of the clutter of the environment. To describe the clutter of the environment, we need a quantitative measurement of the "density" of obstacles with respect to the position and heading of the mobile robot. This obstacle density measurement can be used, not only as an input to the speed control, but also as an indication of the type of sensor feedback required. For this reason, a *Weighted Obstacle Density Function* (WODF) and its application to robot navigation will be presented in this dissertation.

Based on the obstacle density measurement, the flexibility in sensor configuration, and the controllability of motion speed, an *Adaptive Navigation Strategy* (ANS) will be developed for intelligent robot navigation with human-like response to the clutter in the environment.

1.5 Vision-Based Motion Planning: Self-Calibration and Docking motion

Vision-based robot navigation is one of the major topics in this work due to the following reasons:

- A robot system is not complete without the function of correcting navigational errors, and vision provides a feasible solution to problems such as self-calibration.
- A vision system provides rich environmental information which usually can not be collected by sonar and infrared sensors, e.g. patterns on structured surfaces, landmark, etc. Vision systems also have longer effective range than most other types of sensors.
- Visual perception plays a key role in many autonomous behaviors, some of them are critically important to robot navigation tasks.

For self-calibration purposes, navigational landmarks will be used to provide the absolute coordinates of certain calibration points, which usually designate the “points of interest”. By formulating the relation between 3-D geometrical features and their projection on the image plane, we can derive the relative position of the robot given any landmark identified by the vision system. Based on that information, the robot can plan its motion for various tasks, such as docking motion.

In order to perform accurate docking motion, which is the final stage of approaching the robot’s navigation target, a visual-servo motion control system will be developed. In general, docking can be achieved by visually tracking the landmark pattern which indicates the position of the docking port. Detailed discussions of the vision system will be provided in this dissertation.

1.6 Organization of this Dissertation

This dissertation is organized around three major topics of the autonomous mobile robot navigation systems. They are:

- 1) the multi-sensor integration technique and its application to mobile robot navigation
- 2) the adaptive navigation strategies based on obstacle density function
- 3) the vision-based mobile robot navigation, including landmark tracking using visual-servo control and robot self-positioning

The general problem of autonomous robot navigation has been defined in this chapter. Also, the classification of dynamic and static environments is included. Chapter 2 is dedicated to surveying previous work relevant to this dissertation. A brief description of our research mobile robot – SPARTA – is also covered in that chapter.

The sensing system onboard SPARTA and the algorithm for integrating redundant sensor data from different sources are presented in Chapter 3. Chapter 3 also reports the results of extensive experiments on sensing strategies and sensor data integration. Chapter 4 discusses the central idea of adaptive navigation strategies, including the navigational constraints, decision criteria, and feasible ways of measuring the clutter of the robot's environment.

The high computation cost associated with vision-based control is gradually being cancelled out by the dramatic increase in the computation power of hardware. For example, visual-servo control, which had been considered difficult before, has become more favorable in recent years. Chapter 5 gives the detailed treatment of the visual-servo control.

Given the discussion in Chapter 3, 4, and 5, a hybrid navigation system is proposed in Chapter 6 to interconnect the sensor-based dead-reckoning navigation approach and the

vision-based mobile robot guidance approach. Using this navigation system, SPARTA is capable of reaching its goal positions with an average position error of 5 cm and a heading error of 1° .

Finally, the conclusions and summary of this dissertation, followed by recommendations for future research, are presented in Chapter 7.

CHAPTER 2

Literature Background

Mobile robot navigation has drawn much attention in robotics research, not only because of the importance of its potential applications, but also due to the challenge of complexities involved. Most of the previous works can be classified into the following categories [31, 35]:

- Sensing technology / multi-sensor integration
- Map building
- Global / local path planning
- Vision
- Robot control

Each topic in the above list constitutes a sub-domain of the entire robot navigation problem. This chapter will focus on the topics of multi-sensor integration, vision, and the corresponding robot control problems.

2.1 Sensors for Mobile Robot Navigation

In order to perform various types of tasks within its environment, a mobile robot needs "eyes" and "ears" to get the information required to perceive its "world".

Dowling [6] divided sensing technologies into the following groups:

- Simple proximity sensors
- Dead-reckoning (e.g. encoders, inertial navigation devices)
- Ranging devices (e.g. Sonar, Laser range finder)
- Vision

He pointed out that the use of basic proximity sensing is not so much as a navigational aid but more as a minimal configuration to provide a level of safety to humans, the environment, and the robot itself. For long distance navigation, more sophisticated approaches should be used, such as the inertial guidance systems. Also, the acquisition of robot position is usually done by dead-reckoning methods, for example, using wheel encoders to record the incremental moves of the drive motor which translates into changes in robot's position. The problem of this method is that wheels often slip. Slippage, in addition to changes in wheel rolling radius and other external forces can all result in significant errors in calculated robot position and heading. Therefore we need a reliable way to provide accurate estimations of these parameters from the environment instead of relying on dead-reckoning only.

The sensors used in mobile robot research can be classified into Visual and Non-visual sensors. Visual sensors refer to the image acquisition devices for vision systems. This category of sensors are usually implemented as sensor arrays, such as a CCD camera. Examples of non-visual sensors are tactile sensors, infrared detectors, ultrasonic transducers, etc. The key difference between these two classes of sensors is the amount of information, i.e., the amount of data, each of them can provide. For visual sensors such as a CCD camera, the output of each scanning cycle is a 2-dimensional array of gray values (or three arrays of intensity values for color imaging). In other words, there would be more

than 16K-bytes of data for a 128 x 128 video image with 256 gray-scales. In contrast to the huge amount of data generated by visual sensors, the infrared detectors provide only 1-bit of data, where each sensor can be either *ON* (1) or *OFF* (0).

It is clear that much more information about the environment can be acquired by using a visual sensor array, however the cost of processing the large amount of information has been a major drawback, especially for real-time navigation of mobile robot. In fact, most researchers utilize at least one type of non-visual sensor to obtain the local information of the robot environment.

2.1.1 Sensor-Based Map Construction Using Occupancy Grids

Elfes [20] conducted pioneer research in sonar-based map construction. A sensor ring consisting of 24 Polaroid ultrasonic range transducers was designed to assess the robot's environment. In his approach, the environment is quantized into a grid map where each cell has two values associated with it, namely the probability of being empty (p_E) and the probability of being occupied (p_O). He defined the following probability models to describe the occupancy in the workspace (see Figure 2.1):

Definition 2.1 Given the following variables

R range measurement returned by the sonar sensor,

ϵ maximum sonar measurement error,

Ω solid angle subtending the main lobe of the sonar sensitivity function,

S, P position of the sonar sensor and the target point in the workspace, respectively,

δ distance from target point P to S ,

θ angle between the main axis of the beam and P as seen from S .

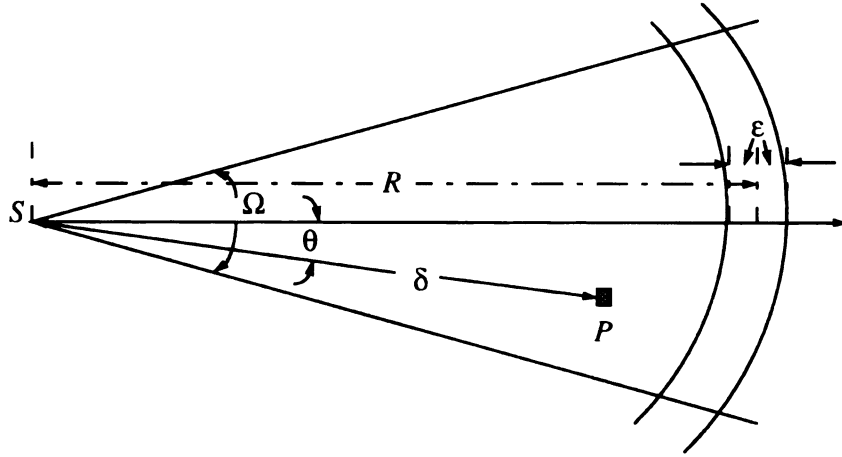


Figure 2.1. The geometry of sonar sensor model.

The empty probability density function is given by

$$p_E(P) = E_r(\delta) \cdot E_a(\theta) \quad (2.1)$$

where

$$E_r(\delta) = \begin{cases} 1 - ((\delta - R_{min}) / (R - \epsilon - R_{min}))^2 & \text{if } \delta \in [R_{min}, R - \epsilon] \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

and

$$E_a(\theta) = 1 - (2\theta/\Omega)^2 \quad \text{for } \theta \in [-\Omega/2, \Omega/2] \quad (2.3)$$

Similarly, the occupied probability density function is

$$p_O(P) = O_r(\delta) \cdot O_a(\theta) \quad (2.4)$$

where

$$O_r(\delta) = \begin{cases} 1 - ((\delta - R) / \epsilon)^2 & \text{if } \delta \in [R - \epsilon, R + \epsilon] \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

and

$$O_a(\theta) = 1 - (2\theta/\Omega)^2 \quad \text{for } \theta \in [-\Omega/2, \Omega/2] \quad (2.6)$$

Based on (2.1) and (2.4), the final map is constructed from two separate arrays derived from the empty and occupied probability distributions. He pointed out that the map construction is an iterative process which means that the map from the last sonar scanning will be updated according to current sonar data. The updating procedure proposed is as follows:

(step 1) *Initialization*: set all cells in map to unknown.

(step 2) *Superposition of empty areas*: the probabilities corresponding to the empty areas are combined using the formula:

$$p_E(\text{cell}) \leftarrow p_E(\text{cell}) + p_E(\text{reading}) - p_E(\text{cell}) \cdot p_E(\text{reading}) \quad .$$

(step 3) *Superposition of occupied areas*: the probabilities corresponding to occupied areas are initially weakened by the evidence of the empty certainty factors, that is $p_O(\text{reading}) \leftarrow p_O(\text{reading}) \cdot (1 - p_E(\text{cell}))$. Similar to the empty areas, the occupied probabilities are combined with the formula $p_O(\text{cell}) \leftarrow p_O(\text{cell}) + p_O(\text{reading}) - p_O(\text{cell}) \cdot p_O(\text{reading})$.

(step 4) *Thresholding*: the final occupation value attributed to a cell in the sonar map is given by comparing the strengths of the empty and occupied values.

This approach had been tested on several mobile robots, including the Neptune and the

Terregator at CMU.

Based on the survey of all recently developed mobile robots [24], it is clear that most of these systems are equipped with more than one type of sensor to obtain a certain degree of autonomous behavior. In other words, most researchers had realized the complexities involved in the robot's ability to perceive its environment and decided to improve the flexibility in their sensing strategies. The rationale of using multiple sensors include: 1) possibility of sensor failure, 2) uncertainties in sensor data, and 3) accuracy in feature measurements. The issues of redundant sensing and multi-sensor integration became more and more important in today's autonomous systems developments.

Flynn [23] proposed a rule-based approach to combine the results of two different types of low cost proximity sensors, namely, the sonar range finder and the infrared sensor. She pointed out the differences between the characteristics of these sensors and a possible way of constructing and refining the local map of the robot's environment. She maintained that the sonar range finder is useful in measuring the distance to an object, but its angular resolution is poor due to its wide beam width. On the other hand, the infrared sensor is able to find edges of doorways and narrow passages that would be otherwise blurred by the sonars. It is therefore desirable to use sonar for range finding and infrared sensors for detecting edges and large depth discontinuity with fine angular resolution. Flynn proposed the following rules for integrating the two types of sensor data:

- If the sonar reading is greater than the maximum range for the infrared sensor, then ignore the infrared output.
- If the sonar reading is at its maximum value, then the real distance is greater.
- Whenever the infrared sensor switches from no detection to detection, or the inverse, and the associated sonar reading is less than 10 ft., then a valid depth discontinuity has been detected.

These rules are critical in resolving the conflicts which may exist between sonar-based and infrared-based information due to aforementioned differences in sensor characteristics.

Matthies and Elfes built the robot Neptune [42] which was equipped with both sonar range finder and stereo vision system. They have successfully integrated the data produced by both types of sensors and built navigation maps based on that data. This type of map, which has also been used by many other researchers, is implemented as the *Occupancy Grid*. The values stored in each grid cell represent the probability that the cell, which corresponds to a rectangular area in the workspace, is occupied by obstacles.

Moravec [44] presented a systematic treatment of sensor integration in terms of combining different types of sensory data into a *Certainty Grid*. In particular, he presented a typical example of sensor integration where the stereo vision data is combined with the sonar ranging information. He proposed the hierarchy of scales in certainty grids, i.e. covering the near field of the robot with high resolution, and successfully larger ranges with increasingly coarse grids. This is based on the observation that each type of sensor has intrinsic uncertainties and resolution limits which make accuracy drop with range. Moravec also presented two different methods of inserting new sensor measurements into the certainty grid. One of them is actually the same as Elfes's method mentioned earlier in this section. The other method is based on Bayesian reasoning:

Given A and B as independent sources of information, we are interested in the probability that a cell in the certainty grid is occupied. The formula for map combining is

$$\frac{p(o|A \wedge B)}{p(\bar{o}|A \wedge B)} = \frac{p(o|B)}{p(\bar{o}|B)} \times \frac{p(o|A)}{p(\bar{o}|A)} \times \frac{p(\bar{o})}{p(o)} \quad (2.7)$$

where $p(o)$ represents the probability that the cell is occupied [44].

2.1.2 Map Construction Using the Sonar Histogram

Another type of grid cell navigation map is based on the histogram of sonar ranging data. Borenstein and Koren [12, 13] introduced the *histogramic in-motion mapping* (HIMM) approach, which was developed for real-time map building with a mobile robot in motion. Although a similar grid map is used, the histogramic approach is quite different from the works done by Moravec and Elfes. Instead of applying any probability model to the sonar data, the HIMM simply uses straight-scale increments /decrements to update the histogram registers associated with the grid cells on the sonar acoustic axis. Figure 2.2 illustrates the procedure for updating the sonar histogram based on sonar reading. Their method greatly reduced the overhead of probability computation, which is unavoidable in those approaches based on the probability models of occupancy. For this reason, their approach allows higher sonar scanning rate, which is more suitable for real-time mobile robot navigation. This is particularly important for those mobile robots working in dynamic environments where the motions of obstacles are unpredictable. In fact, the robot must totally rely on real-time sonar feedback for obstacle avoidance in this case. Similar to the probabilistic approaches, HIMM to an extent incorporate the uncertainty associated with the sensor data.

2.2 Vision-Based Mobile Robot Navigation

The simplest navigation strategy used in indoor applications for mobile robots is *dead reckoning*. The major drawback of dead reckoning is the accumulation of positional errors caused by environmental variations such as the road surface and temperature, etc. To overcome this problem, a mobile robot needs to calibrate itself by using sensory feedback and some prior knowledge about the environment.

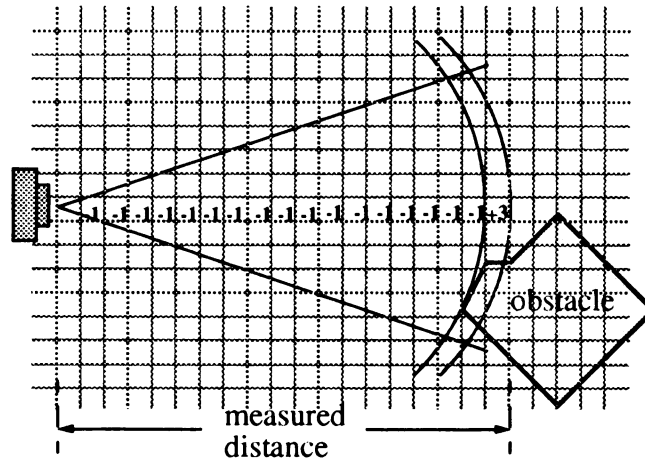


Figure 2.2. Update procedure for sonar histogram.

2.2.1 Self-Calibration/ Positioning Using Navigational Landmarks

Some researchers proposed the use of navigational landmarks for robot guidance and pose-estimation [8, 9, 18, 22, 29, 33, 45, 49, 54]. The position and orientation of each landmark are “learned” by the mobile robot as part of its prior knowledge about the workspace. By recognizing and locating a landmark, the mobile robot can calculate its own position and heading with respect to the landmark frame. This information, together with the prior knowledge about the landmark, gives the robot’s absolute position and heading with respect to the world frame. A vision system is often regarded as the best choice for above tasks because of its long effective range and the abundance of information embedded in image data. In fact, many research robots are equipped with vision systems, and some of them use binocular (or even trinocular [9]) vision systems.

In the early attempt of visual navigation, *artificial landmarks* or *guideposts* are used in the

indoor applications. Kabuka and Arenas [33] used standard patterns as navigation landmarks. The pattern proposed includes two parts: a relative displacement pattern and a set of identification codes. They chose a circle for the relative displacement measurement because the resultant projection can be approximated by an ellipse whose size, eccentricity, and orientation can be used to determine the viewing position. The geometrical relationship between the camera and the landmark circle was summarized as a set of equations. They also claimed that the parameters of the projected ellipse, namely the lengths of major and minor axis, can be calculated from the lower order position-invariant moments. In a subsequent paper, Hussain and Kabuka [29] presented a system for real-time generation of these lower order moments. They also discussed the hardware architecture for the calculation of the second- through the seventh-order moments. Results of this series of research had been applied to a flexible multiple mobile robot system (MMRS) proposed by Fok and Kabuka [24]. In MMRS environments, landmarks are fixed in the workspace to provide absolute position information to the mobile robots.

2.2.2 Tracking Motion Using Visual-Servo Control

Beside the static position verification using landmarks, some researchers explored the possibility of using vision information in the dynamic feedback loop of robot control. This is often called the *visual-servo* control. Papanikolopoulos, Khosla, and Kanade [48] presented a visual tracking system capable of following targets moving in 2-D space. Their setup includes a camera mounted on the end effector of a six-joint robot arm (eye-in-hand configuration) and a separate image processing hardware environment. The sum-of-squared differences (SSD) optical flow is used to compute the motion of the tracking target. This technique requires the real-time computation of the correlation of two subsequent images. The displacement in the image plane is then used as the feedback to the robot controller. They claimed a maximal tracking speed of 7 cm/second at the height of 68 cm.

Allen, Timcenko, Yoshimi, and Michelman [5] conducted a research on robot hand-eye system for tracking and grasping a moving object. Their work addressed three distinct problems in robot hand-eye coordination: fast computation of 3-D motion parameters from vision, predictive control of a moving robot arm to track a moving object, and interception and grasping. This setup is different from the one used by Papanikolopoulos *et al.* Their vision system consists of stereo cameras which are fixed to the robot's environment. A stereoscopic optic-flow calculation is performed at each pixel in the image in order to generate a motion energy profile. Based on that information, the 3-D position of the moving object is recovered. They used a nonlinear filter to recover the trajectory parameters which can be used for forward prediction, and the updated position is sent to a trajectory planner/arm controller. After the tracking is stable, the arm is commanded to intercept the moving object and pick it up. They claimed the robot arm can be operated at approximately the human arm movement rate.

Another work by Castano and Hutchinson [15] provided an example of visual-servo control in 3-D space. They tackle the problem by combining 2-D visual-servo control with 1-D position control. Their experimental setup includes a Puma 500 robot arm and a fixed camera staring at the target. In order to apply the vision information to robot control, the Jacobian matrix relating the perceived position of the end effector and the joint coordinates of the robot arm is derived. The motion of the end effector in the directions perpendicular to the camera axis is controlled by visual-servo in the 2-D plane. As for the direction parallel to the camera axis, a simple position control with forward kinematics is used. As a result of their strategy, the end effector complied with the 3-D line passing through the camera center and the center of the target.

Dickmanns [16] proposed a spatial-temporal world model servoed by image feature-tracking and aggregation. He made the assumption that all objects in the workspace have reasonably good dynamical models and therefore hypothesis generation with their motion

is possible. No implementation was given, however.

Ayache [9] provided a detailed discussion of the vision system for a mobile robot, including the geometrical relation, feature matching, motion, and multiple image fusion problem. Espiau, Chaumette, and Rives [22] presented a new approach to incorporate vision with servo-control system.

The major challenge for vision-based approaches is the real-time processing of a large amount of data. As a matter of fact, the speed of data processing limits the maximal speed achievable by a mobile robot. However, recent development of high performance, low cost processors seems to promise a practical solution to this problem in the near future.

2.3 Our Research Mobile Robot – SPARTA

All of the experiments conducted for this dissertation has been done on the mobile robot – SPARTA – which is the result of years of project research in the Robotics Laboratory of Electrical Engineering Department at the Michigan State University [39, 40, 41]. The mobile base of SPARTA is a Labmate robot vehicle manufactured by the Transitions Research Corporation (TRC). It has a holonomic configuration with two drive wheels and four caster wheels symmetrically installed with respect to the center of the vehicle. The microprocessor-based motor controller provides a set of low level commands for the motion control of the vehicle base. These commands can be issued by the robot controller, which is an Intel 80486-based PC, through a RS-232 serial link. A tower structure had been built on the top of the vehicle to accommodate the robot controller, a DC power supply, and two 12-Volt batteries. On the top of the tower structure, a turn table driven by a computer controlled stepper motor is installed.

The sensor subsystem of SPARTA includes:

- one Panasonic video camera,
- eight Polaroid ultrasonic sensors,

- eight Banner infrared detectors,
- two bumper switches,
- sensor boards for scheduling and activation of non-visual sensors,
- video frame grabber for storing the image taken by the camera.

All types of the sensors are mounted on the tower structure except the bumper switches, which are installed on the front and back of the vehicle base. The characteristics and functionality of each type of sensors will be discussed in Chapter 3.

2.4 Summary

Major topics of autonomous mobile robot navigation are reviewed in this chapter, including sensor categorization, occupancy grid representation, sonar histogram, and vision-based mobile robot navigation. Sensors used in robot navigation can be roughly separated into visual and non-visual types. The characteristics of each type of sensor are briefly discussed. The occupancy grid representation developed by Elfes and Moravec has been widely used in sensor-based map construction. The probability distribution that each cell in the grid map is occupied or empty is formulated. The rules-based approach proposed by Flynn for combining sonar and infrared data is discussed. Another map construction approach proposed by Borenstein and Koren is described. They use sonar histogram as the base of map construction.

A vision system can be used to perceive the environmental geometry. Together with special landmark patterns, a vision system provides a feasible solution to the position estimation problem. Some researchers have used vision system as part of the feedback loop in robot motion control systems. This chapter introduced several pioneer experiments on visual-servo control.

At the end of this chapter, our research robot – SPARTA – is introduced. Four types of

sensors are mounted on SPARTA, including ultrasonic range sensors, infrared detectors, vision camera, and bumper switches. The first three types of sensors are used to perceive the robot's environment while the bumper switches are mounted for safety consideration. All sensor data are collected by the main computer on board, which also acts as high level robot controller.

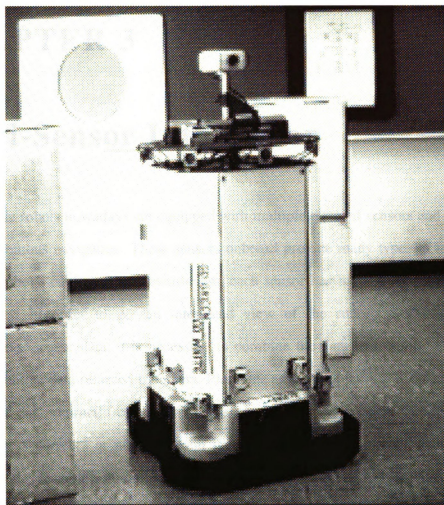


Figure 2.3. The SPARTA mobile robot.

CHAPTER 3

Multi-Sensor Integration

Most mobile robots nowadays are equipped with multiple types of sensors and capable of semi-autonomous navigation. These sensors onboard provide many types of information about the robot's environment, nevertheless, each sensor has its own characteristics and functionality. In order to get an integrated view of the robot's environment from heterogeneous sensor data, it is necessary to combine the data returned by each sensor together with the data returned by others. Problems arise when there is any inconsistency between the data returned from different sensors, the question to ask is "which source of information is correct? or none of them is correct?" Recent research on multi-sensor fusion provides feasible solutions to these problems. By applying probabilistic models on sensor data, it is possible to interpret data from different sources, sometimes even conflicting, into a uniform representation of the robot's environment. This technique helps to improve the accuracy of the robot's perception by using the data from more than one sensor to "extract" the same feature from the environment and thus "average out" possible errors.

3.1 Sensing Systems of SPARTA

Four types of sensors are equipped on SPARTA for navigational purposes. They are

- ultrasonic range sensors

- infrared detectors
- vision camera
- micro switch (on safety bumpers)

The first two types of sensors constitute the basis for proximity sensing. These sensor units are fixed to the robot's tower structure, and form a 360 degree sensor ring topologically. The total number of the ultrasonic sensors is 8, so is the total number of the infrared detectors. In particular, the ultrasonic range sensors are equiangularly mounted around the turn table on the top of the robot. This arrangement allows complete control of the orientations of these sensors without rotating the robot base.

The combination of ultrasonic range sensors and infrared detectors has been studied by Flynn [23]. According to her work, the ultrasonic sensors produce useful ranging information by measuring the time required for ultrasound waves to travel from the transmitter to the object in front of it then bounce back. However, the exact location of the object is uncertain due to the fact of wide transmit-receive angle and the noise problem associated with it. In contrast to the ultrasonic sensors, the infrared detectors provide little ranging information, but they are capable of detecting objects with high reliability and pinpoint accuracy in terms of detecting direction. She proposed a feasible solution to exploit the advantages of both types of sensors in order to derive an uniform representation of the robot's environment. The result is a set of rules to determine whether a point in the environment is occupied by any obstacle.

The arrangement of proximity sensors on SPARTA is based on the similar idea of combining the advantages of ultrasonic sensors and infrared detectors. However, a probability model is added to improve the overall perception results. This will be discussed later in this chapter. Figure 3.1 illustrates the layout of all proximity sensors on SPARTA.

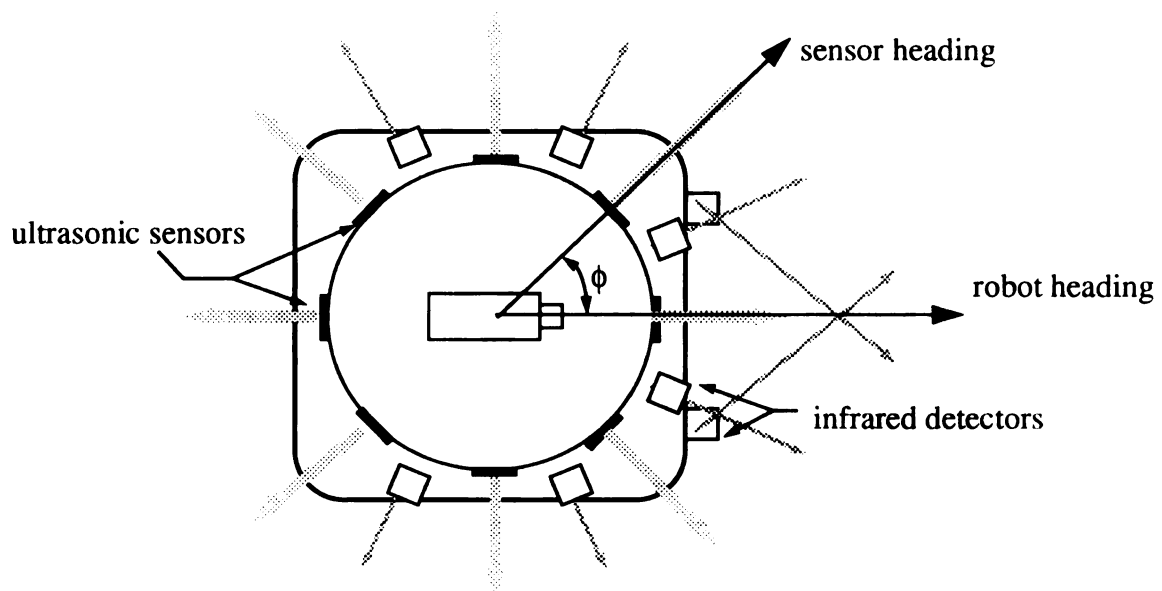


Figure 3.1. Geometrical layout of the proximity sensors on SPARTA.

A vision camera is mounted on the top of the robot's turn table such that the pan-axis of the camera shares with the z -axis of the robot frame. This reduces the coordinate transformation between the camera frame and the robot frame to one rotation plus one translation in z -axis. The geometrical relationship between the camera frame and the robot frame will be elaborated in chapter 5.

All of the proximity sensors are controlled by a sensor board which is connected to the robot controller through a RS-232 serial port. The vision camera, however, is interfaced to the robot controller through a video frame grabber.

3.2 Sensor Modeling and Occupancy Functions

As mentioned in the previous section, each ultrasonic sensor has a wide-angle transmitting and receiving envelope. It only gives the range measurement to the closest object within this envelope. The exact location of the object with respect to the sensor is not available.

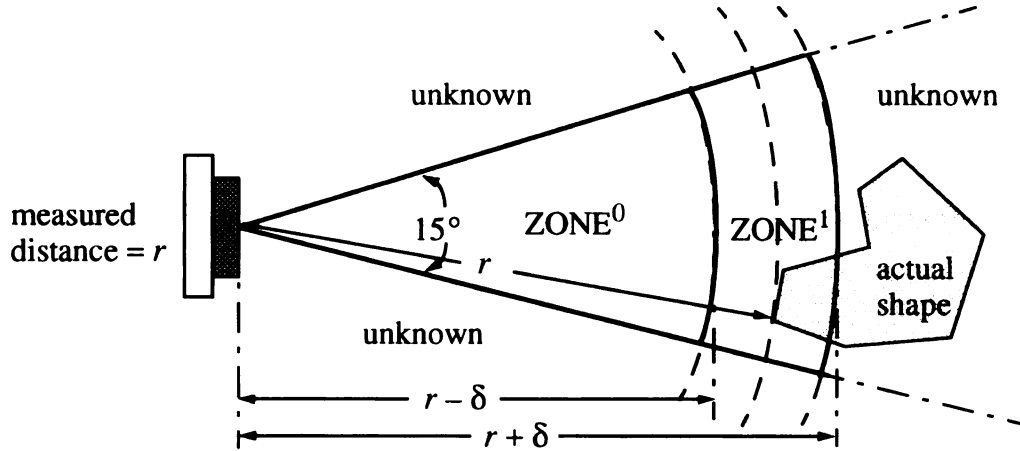


Figure 3.2. The interpretation of range data from an ultrasonic sensor.

As shown in Figure 3.2, this uncertainty leads to the geometrical representation which is a bounded sector of annular region. Note that given the measured distance equals r , the only conclusion we can draw is

- (1) ZONE^0 is empty with high probability
- (2) part of ZONE^1 is occupied with high probability

where ZONE^1 is the sector of annular region bounded by the envelope of active region and the measured distance with error margin $\pm\delta$, and ZONE^0 is the sector of circle between ZONE^1 and the sensor itself. Any other regions in the sensor's environment are unknown, i.e. inaccessible to the sensor.

Beside the uncertainty stated above, other factors such as the surface characteristics of the object and the angle of reflection also affect the output response of the sensor. In order to interpret the actual meaning of the ultrasonic ranging data, we need a probability model to

describe the relationship between the response of the sensor and the geometrical layout of the sensor's environment. This type of model can be used to generate the geometrical representation of the robot's environment perceived by the sensors. This representation is actually a map which is generally called the *Occupancy Functions* of the robot's environment.

Given the setup in Figure 3.1, the probability model associated with each ultrasonic sensor is actually a function of the following variables

- measured target range,
- sensor heading with respect to the robot frame, and
- the relative position with respect to the sensor device.

Assuming the effective range of the ultrasonic sensor is R_{ut} , the probability of occupancy for a point p , which is inside the active region and has its distance to the sensor equals x , given the measured distance r , can be assigned as follows:

$$f_{UT}(p|r) = \begin{cases} k_1 x e^{x-r} & \text{if } x \in [0, r] \\ 0.5 + k_2 (x - R_{ut}) e^{-(x-r)} & \text{if } x \in [r, R_{ut}] \\ 0.5 & \text{otherwise} \end{cases} \quad (3.1)$$

where

$$k_1 = \frac{1}{r} [0.5 + k_2 (r - R_{ut})]$$

and

$$k_2 = \frac{1 - 0.5 (R_{ut} - r) + (0.5/r) (r - 1 + e^{-r})}{(r - R_{ut} - 1 + e^{r-R_{ut}}) + (1/r) (r - R_{ut}) (r - 1 + e^{-r})}$$

Figure 3.3 illustrates the 3-D surface of this function.

Probability of occupancy based on ultrasonic ranging data: $r = 30.0$ cm

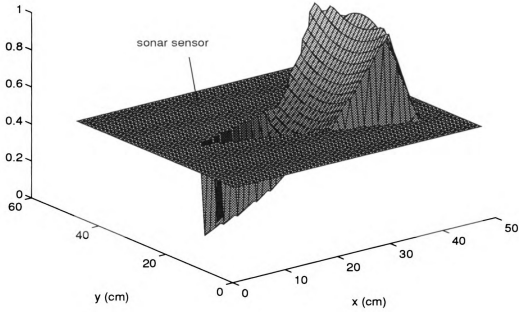


Figure 3.3. 3-D surface representation of the probability of occupancy derived from ultrasonic ranging data ($r = 30$ cm in Equation (3.1)).

In contrast to the ultrasonic sensors, the infrared detectors are capable of performing reliable detection of most types of object surfaces with narrow beam width. This provides accurate information about the presence of objects in a specific direction. However, the distance information of the object is not available from the infrared detectors. In fact, each infrared detector acts like one-bit switch which can only produce an answer of either TRUE or FALSE. Assuming the effective range of the sensor is R_{ir} and the IR beam propagates along sensor axis l , the output can be expressed as a function of the sensor's heading

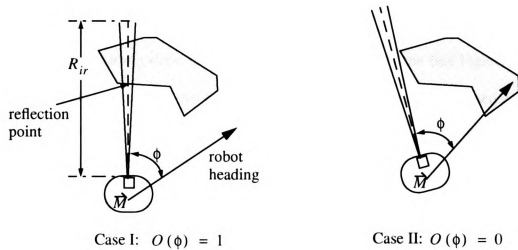


Figure 3.4. The infrared detector output function for two different cases.

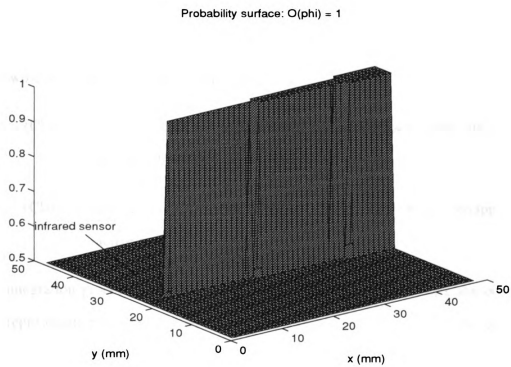


Figure 3.5. The 3-D surface representation of the probability function $O(\phi) = 1$.

$$O(\phi) = \begin{cases} 1 & \text{if } l \text{ intersects any object surface in } R_{ir} \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

where ϕ is the heading angle of l with respect to the robot frame (see Figure 3.1). The geometrical interpretation of this output function is shown in Figure 3.4. Based on (3.2), a probability model can be constructed to describe the relationship between the output of the sensor and the corresponding geometrical layout.

Given an output function $O(\phi)$, the perceived environment near the sensor axis is

$$f_{IR}(p|O(\phi)) = \begin{cases} O(\phi) & \text{if } p \text{ lies within the } \pm \delta\phi \text{ envelope of } l \\ 0.5 & \text{otherwise} \end{cases} \quad (3.3)$$

The 3-D surface of (3.3) is shown in Figure 3.5.

3.3 Constructing World Map from Multiple Sensor Data

In order to achieve autonomous navigation, one must turn multiple sensor data from different sources into a single uniform representation of the robot's environment. This is the well known multi-sensor integration problem. This type of problem can be separated into two classes based on their complexities:

(Class 1) all sensor data are uncorrelated, i.e. each sensor covers its own region without any overlap with other sensors

(Class 2) some sensor data are redundant, i.e. some sensors have overlapping active regions

The integration process for Class-1 problems is straightforward. In fact, the occupancy grid representation can be derived by superposition and normalization of the occupancy functions of all sensors. Constructing a navigation map from ultrasonic ranging data alone is a typical example of Class-1 multi-sensor integration.

The Class-2 problems, on the other hand, are more complicated because of the additional issues involved, such as conflict resolution, decision criteria, etc. However, the result of redundant sensing is usually better than the simple strategy used in Class-1 problems in terms of the representation accuracy. Clearly, this advantage is gained at the price of higher computation cost.

A typical example of Class-2 problems is the integration of ultrasonic ranging data with infrared detection results. This can be formulated as a probability distribution computation problem. Let's consider the case of SPARTA with sensor arrangement shown in Figure 3.1. In this case, each point in the workspace is "covered" by one ultrasonic sensor at any time. Meanwhile, this point may also lie on the sensor axis of some infrared detectors, depending on the geometrical layout. This problem can be defined as follows:

Definition 3.1 Given the following information:

- the robot's workspace W ,
- a set of probability functions from ultrasonic ranging data, i.e. $f_{UT}^i(p|r_i)$, $i = 1, 8$,
where r_i is the measured distance for sensor i ,
- a set of probability functions from infrared sensor outputs, i.e. $f_{IR}^i(p|O_i)$, $i = 1, 8$,
where O_i is the output for sensor i ,

find the integrated representation of the occupancy probabilities for all $p \in W$.

The integration of sensor data can be done with the algorithm listed in Table 3.1. Figure 3.6 shows a series of examples of Class-2 multi-sensor integration. Note that the environment setup shown in Figure 3.6(a) will be referred to as **workspace_1** throughout this chapter. Four observation points in **workspace_1** have been chosen to test the sensor integration algorithm. The results have been shown in Figure 3.6(b) - Figure 3.6(e).

Table 3.1. The sensor integration algorithm.

```

algorithm UT_IR_integration
begin;
{
  for all  $p \in W$ , do
  {
    compute the aspect angle of  $p$ , call it  $\theta_p$ ;

    find the ultrasonic sensor covering  $p$ , read range  $r_i$ ;

    if ( $p \in \text{ZONE}_i^1$ ) then
    {
      for  $j = 1, 8$ 
      {
        if ( $|\theta_p - \phi_j| < \delta\phi$ ) then  $f(p) = O_j(\phi_j)$ ;
        else  $f(p) = 1$ ;
      }
    }

    else if ( $p \in \text{ZONE}_i^0$ ) then  $f(p) = 0$ ;

    else  $f(p) = 0.5$ ;
  }
}
end;

```

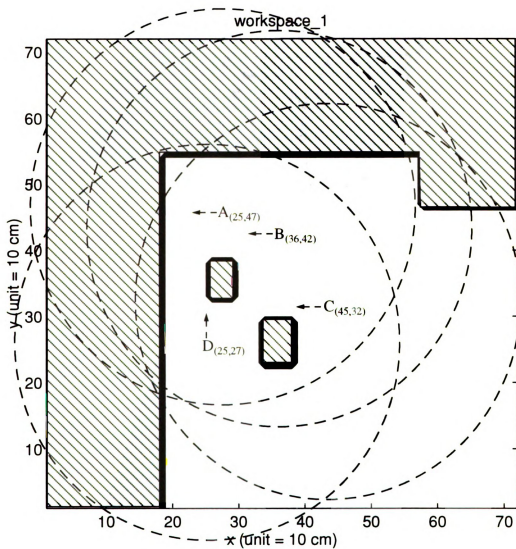


Figure 3.6(a). The layout of **workspace_1** and observation points: A, B, C, and D. (Dashed arrows represent robot headings.)

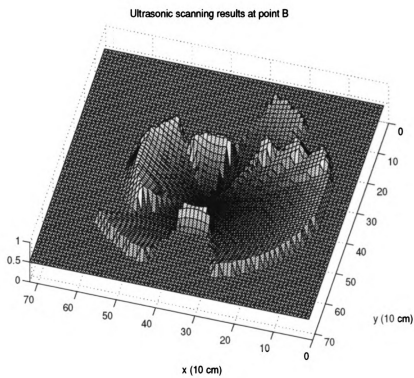
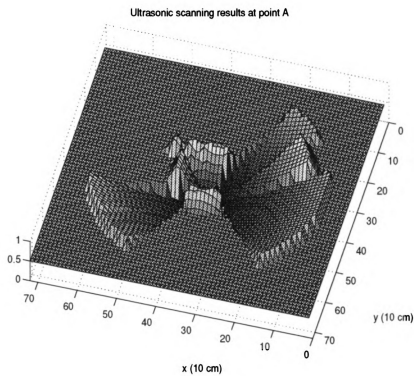


Figure 3.6.(b) The obstacle occupancy probability surface constructed from ultrasonic ranging data.

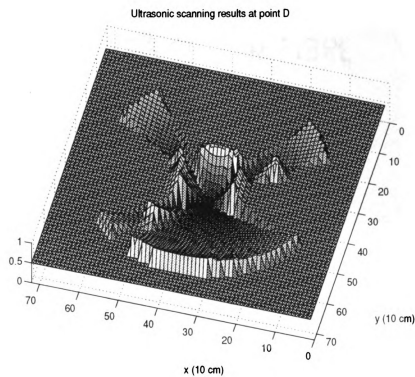
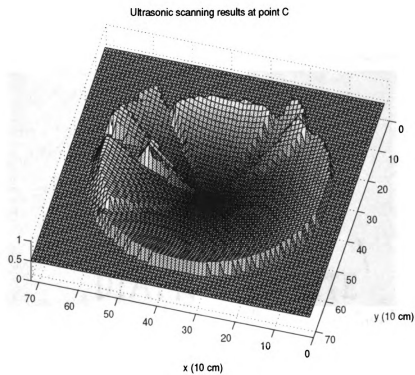


Figure 3.6.(b). (Cont.) The obstacle occupancy probability surface constructed from ultrasonic ranging data.

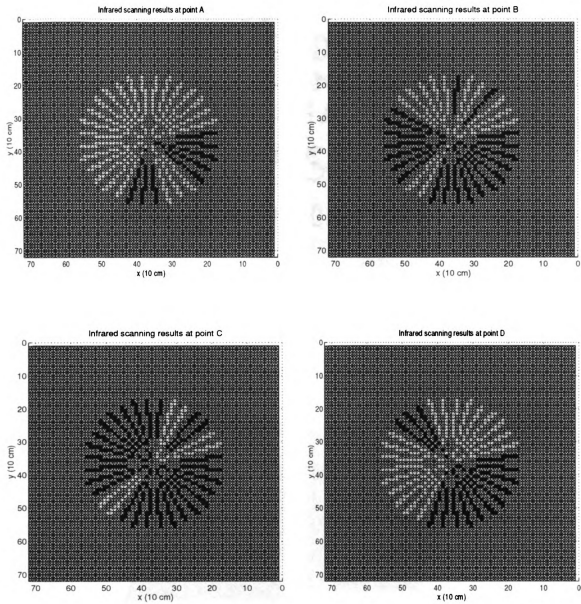


Figure 3.6.(c). The obstacle occupancy probability surface (top view) constructed from infrared detector responses.

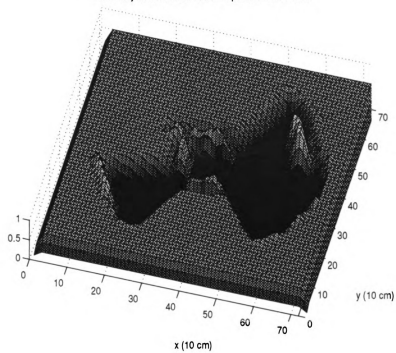
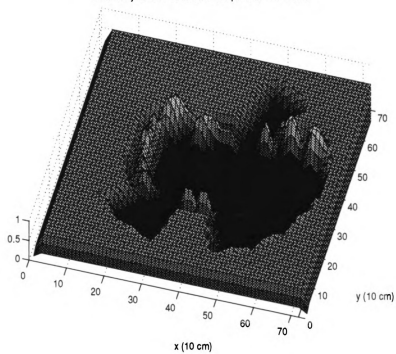
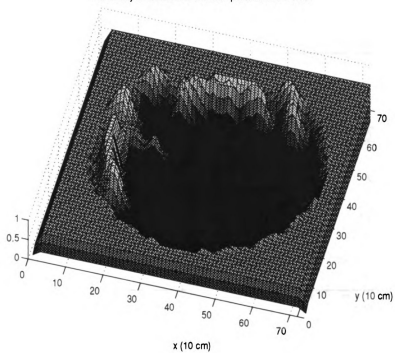
Probability surface observed from point A: $\text{Thd} = 0.78$ Probability surface observed from point B: $\text{Thd} = 0.78$ 

Figure 3.6.(d). The integrated probability surfaces (smoothed by 3×3 averaging mask).

Probability surface observed from point C: Thd = 0.78



Probability surface observed from point D: Thd = 0.78

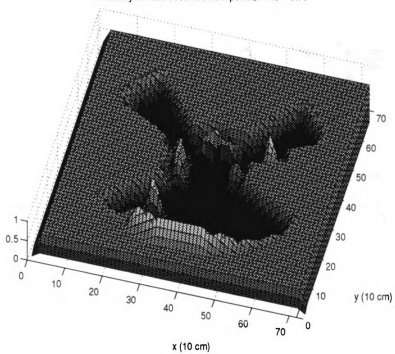


Figure 3.6.(d). (Cont.) The integrated probability surfaces (smoothed by 3 x 3 averaging mask).

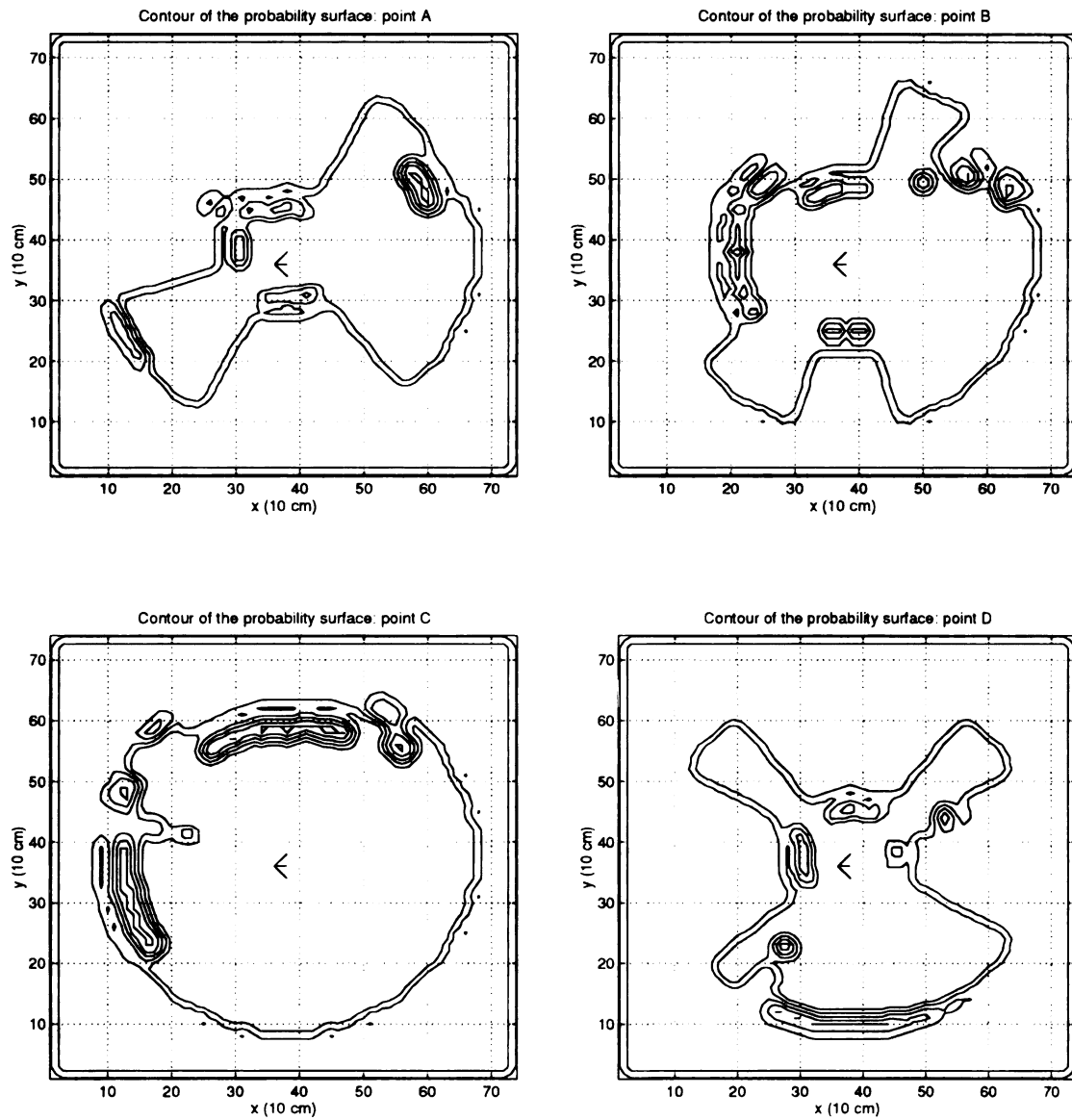


Figure 3.6.(e). The contours of probability surfaces.

3.4 Errors in Sensing and Data Integration

There are various types of errors associated with sensor-based map construction. In general, these errors come from incorrect and inconsistent interpretations of the data. The causes of incorrect sensor data depends on the characteristics of the sensors used to perceive the environment. For ultrasonic sensors, the range data can be erroneous because of many reasons, including the environmental geometry and surface properties. The same problem occurs to the infrared detector outputs, although the magnitude is much smaller in this case. With the help of redundant sensing technique, some portion of the errors may be eliminated, however, other problems such as the treatment of inconsistent interpretation may also arise. Several solutions were proposed toward this problem [8, 43]. Unfortunately, there had been no clear definition of any practical measure of how accurate the map, which is constructed by data integration, matches the “real world”. Clearly, such measurement can be used to evaluate various sensor integration methods.

The first problem in defining a measurement of match is the feature-correspondence between the map and the real world. For example, when ultrasonic sensors are used for map construction, it is quite often that the perceived size and shape of objects be seriously distorted. This renders the feature extraction operations useless, not to mention the feature correspondence, unless we know exactly what types of features to look for. Therefore it is more feasible to define a match without using geometrical features like lines and curves.

For analysis purposes, a sum-of-squared differences (SSD) measurement is proposed.

Assuming the region covered by multi-sensor integration is R_{msi} , then the occupancy grid

of R_{msi} , denoted by $F_{occ}(R_{msi}) = \{f(p) | p \in R_{msi}\}$, represent the probability that

whether each cell in the grid is occupied. The value of $f(p)$ is computed by the sensor integration algorithm presented in section 3.3 or the equivalent. Based on $F_{occ}(R_{msi})$, we can formally define the term *map*.

Definition 3.2 A *map*, denoted by M , is a binary grid derived by applying a threshold to the occupancy grid, that is

$$M = \{ \beta(p) \mid p \in R_{msi} \}$$

where

$$\beta(p) = \begin{cases} 1 & \text{if } f(p) \geq \delta \\ 0 & \text{otherwise} \end{cases}$$

and $\delta \in [0, 1]$ is the threshold value.

Clearly, the selection of δ is critical to the construction of a map. This will be discussed later in the text. The mapping error inside R_{msi} can be formally defined as follows:

Definition 3.3 A *SSD error surface*, denoted by $\Xi = \{ \epsilon_{i,j} \mid (i,j) \in R_{msi} \}$, with window size $n \times n$, is a representation of map construction errors described by

$$\epsilon_{i,j} = \sum_{k=i-n/2}^{i+n/2} \sum_{l=j-n/2}^{j+n/2} \frac{1}{n^2} (M_{k,l} - W_{k,l})^2 \quad (3.4)$$

where M and W are the binary grid representations of constructed map and real world, respectively. In particular,

$$W_{k,l} = \begin{cases} 0 & \text{if cell } (k,l) \text{ is empty} \\ 1 & \text{if cell } (k,l) \text{ is occupied} \end{cases}$$

By applying a window averaging operator, the SSD surface becomes smoother. That means isolated noises will not cause high response, instead, they will be largely ignored. Clearly, the degree of smoothness in SSD surface depends on the size of the window.

Note that the SSD surface only reveals the local match information at each grid cell of the map. It is also desirable to define an index for the overall match result. One feasible way is to compute the integral of SSD in R_{msi} .

Definition 3.4 The overall *match index* of map M is the integral of SSD in R_{msi} , that is

$$I(M, R_{msi}) = 1 - \frac{1}{\text{area}(R_{msi})} \left[\sum_{(i,j) \in R_{msi}} \sum \varepsilon_{i,j}(M) \right]. \quad (3.5)$$

where $\text{area}(R_{msi})$ represents the total number of grid cells in R_{msi} .

Based on this definition, we can derive basic properties about the match index. First of all, let us inspect the special cases where the map is a perfect match to the real world geometry.

Proposition 3.1 If map M is a perfect match of the real world geometry in R_{msi} , i.e.

$M_{i,j} = W_{i,j} \forall (i,j) \in R_{msi}$, then the match index $I(M, R_{msi}) = 1.0$.

Proof> If M is a perfect match then it follows from (3.4) that

$\varepsilon_{i,j} = 0, \forall (i,j) \in R_{msi}$. Plug in (3.5), we have $I(M, R_{msi}) = 1.0$.

Q.E.D.

Proposition 3.2 If map M is the complement of the real world, i.e.

$M_{i,j} = 1 - W_{i,j}, \forall (i,j) \in R_{msi}$, then the match index $I(M, R_{msi}) = 0$.

Proof> From (3.4), it follows that $\varepsilon_{i,j} = n^2 \left(1/n^2 \right) = 1, \forall (i,j) \in R_{msi}$. Plug in (3.5), we have

$$\begin{aligned} I(M, R_{msi}) &= 1 - \frac{1}{\text{area}(R_{msi})} \left[\sum_{(i,j) \in R_{msi}} \sum (1) \right] \\ &= 1 - \frac{1}{\text{area}(R_{msi})} \cdot \text{area}(R_{msi}) \\ &= 0 \end{aligned}$$

Q.E.D.

Proposition 3.3 Given two maps, M_1 and M_2 , of the same workspace, then

$I(M_1, R_{msi}) > I(M_2, R_{msi})$ if and only if M_1 has more matched cells than M_2 .

Proof>

(i) $\leftarrow$:

If $I(M_1, R_{msi}) > I(M_2, R_{msi})$ then we have

$$1 - \frac{1}{\text{area}(R_{msi})} \left[\sum_{(i,j) \in R_{msi}} \varepsilon_{1,i,j}(M) \right] > 1 - \frac{1}{\text{area}(R_{msi})} \left[\sum_{(i,j) \in R_{msi}} \varepsilon_{2,i,j}(M) \right].$$

Since $\text{area}(R_{msi}) > 0$, it follows that

$$\sum_{(i,j) \in R_{msi}} \varepsilon_{1,i,j}(M) < \sum_{(i,j) \in R_{msi}} \varepsilon_{2,i,j}(M). \quad (3.6)$$

From (3.4),

$$\begin{aligned} \sum_{(i,j) \in R_{ms}} \sum_{k=i-n/2}^{i+n/2} \sum_{l=j-n/2}^{j+n/2} \frac{1}{n^2} (M_{1(k,l)} - W_{(k,l)})^2 \\ < \sum_{(i,j) \in R_{ms}} \sum_{k=i-n/2}^{i+n/2} \sum_{l=j-n/2}^{j+n/2} \frac{1}{n^2} (M_{2(k,l)} - W_{(k,l)})^2 \end{aligned}$$

By rearranging terms, we have

$$\frac{1}{n^2} \left(\sum_{(i,j) \in R_{ms}} \sum_k \sum_l \left[(M_{1(k,l)} - W_{(k,l)})^2 - (M_{2(k,l)} - W_{(k,l)})^2 \right] \right) < 0. \quad (3.7)$$

Now, define the following equation

$$\begin{aligned} \Delta_{(k,l)} &= (M_{1(k,l)} - W_{(k,l)})^2 - (M_{2(k,l)} - W_{(k,l)})^2 \\ &= (M_{1(k,l)} - M_{2(k,l)}) (M_{1(k,l)} + M_{2(k,l)} - 2W_{(k,l)}) \end{aligned} \quad (3.8)$$

therefore (3.7) can be rewritten as

$$\frac{1}{n^2} \left(\sum_{(i,j) \in R_{ms}} \sum_k \sum_l [\Delta_{(k,l)}] \right) < 0. \quad (3.9)$$

There exist four different cases for $M_{1(k,l)}$ and $M_{2(k,l)}$:

Case I: $M_{1(k,l)}$ is a match but $M_{2(k,l)}$ is not. In this case

$$\Delta_{(k,l)} = \begin{cases} 1 \cdot (1-2) & \text{if } W_{(k,l)} = 1 \\ -1 \cdot (1-0) & \text{if } W_{(k,l)} = 0 \end{cases} = -1 \quad (3.10)$$

Case II: $M_{2(k,l)}$ is a match but $M_{1(k,l)}$ is not. In this case

$$\Delta_{(k,l)} = \begin{cases} -1 \cdot (1-2) & \text{if } W_{(k,l)} = 1 \\ 1 \cdot (1-0) & \text{if } W_{(k,l)} = 0 \end{cases} = 1 \quad (3.11)$$

Case III: $M_{1(k,l)}$ and $M_{2(k,l)}$ are both match. In this case

$$\Delta_{(k,l)} = \begin{cases} 0 \cdot (2-2) & \text{if } W_{(k,l)} = 1 \\ 0 \cdot (0-0) & \text{if } W_{(k,l)} = 0 \end{cases} = 0 \quad (3.12)$$

Case IV: None of $M_{1(k,l)}$ and $M_{2(k,l)}$ is match. In this case

$$\Delta_{(k,l)} = \begin{cases} 0 \cdot (0-2) & \text{if } W_{(k,l)} = 1 \\ 0 \cdot (2-0) & \text{if } W_{(k,l)} = 0 \end{cases} = 0 \quad (3.13)$$

Since the value of $\Delta_{(k,l)}$ can only be 1, -1, or 0, it follows from (3.9) that the total number of occurrence of Case I is higher than Case II. Besides, Case III and IV can be ignored because they represent the cells where M_1 is identical to M_2 . Therefore, M_1 has more matched cells than M_2 .

(ii) $\rightarrow$:

If M_1 has more matched cells than M_2 , then the following inequality holds

$$\sum_{(i,j) \in R_{msi}} (M_{1(i,j)} - W_{(i,j)})^2 < \sum_{(i,j) \in R_{msi}} (M_{2(i,j)} - W_{(i,j)})^2 \quad (3.14)$$

Multiply both sides by n^2/n^2 , we have

$$\frac{1}{n^2} \sum_{(i,j) \in R_{msi}} n^2 (M_{1(i,j)} - W_{(i,j)})^2 < \frac{1}{n^2} \sum_{(i,j) \in R_{msi}} n^2 (M_{2(i,j)} - W_{(i,j)})^2 .$$

Under the assumption that all cells outside R_{msi} are matched with the real world, i.e. those

cells do not contribute to above inequality, it follows that

$$\sum_{(i,j) \in R_{msi}} \sum_{k=i-n/2}^{i+n/2} \sum_{l=j-n/2}^{j+n/2} \frac{1}{n^2} (M_{1(k,l)} - W_{(k,l)})^2$$

$$< \sum_{(i,j) \in R_{msi}} \sum_{k=i-n/2}^{i+n/2} \sum_{l=j-n/2}^{j+n/2} \frac{1}{n^2} (M_{2(k,l)} - W_{(k,l)})^2$$

From this inequality, we arrive at $I(M_1, R_{msi}) > I(M_2, R_{msi})$ by reversing the procedure of the first part of proof.

Q.E.D.

Figure 3.7 shows the SSD error surface and corresponding match index of example 1 in section 3.3. Note that the SSD surface and match index change according to the threshold value δ which has been used to convert occupancy grid into binary map. The decision of this threshold value can be incorporated as part of the sensor calibration procedure. The heuristic approach for finding suitable value of δ is:

- (1) Select a typical environment in the robot's workspace and perform a series of map constructions at, say, k different view positions. Call the resultant occupancy grids $F_{occ}^i, i = 1, k$.
- (2) For each of these occupancy grids, apply different values of δ to construct the binary map and then compute the corresponding match index. Plot the match index as a function of δ for each view position.
- (3) Compute the average of these match index curves, and find the value of δ corresponding to the maximal averaged match index. Choose that δ as the

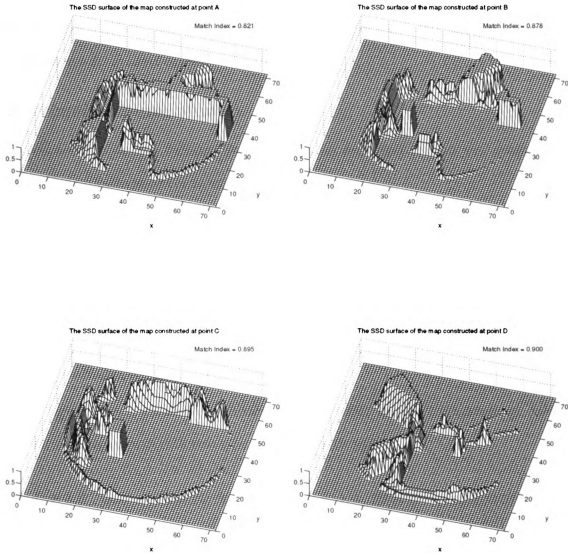


Figure 3.7. The SSD error surfaces for 4 observation points in `workspace_1`.

threshold value for generating binary map.

Figure 3.8 shows an example of this process. The value of δ selected may depend on the view position chosen, however, according to lab experiment this approach generate acceptable results in most cases.

3.5 Adaptive Sensing Strategy

The coordination among different sensors plays an important role in any autonomous system. The problem involves several topics, including:

- selecting the type and number of sensors to use
- deciding the geometrical arrangement of sensors
- defining the features of environment to extract from the sensor data
- scheduling the timing of “firing” each sensor

An intelligent sensor controller provides highly flexible *sensor configurations*, which allow the development of different types of sensing strategies.

Definition 3.5 Given k sensors, denoted by $s_i, i = 1, k$, the *sensor configuration vector* is $\Phi = [\phi_1, \phi_2, \dots, \phi_k]$ where

$$\phi_i = \begin{cases} 0 & \text{if } s_i \text{ is active} \\ 1 & \text{otherwise} \end{cases}$$

The concept of sensor configuration allows total freedom in sensor control algorithm. This also helps to reduce the amount of sensor data as well as overall power consumption by disabling sensors not useful to current navigation task. Moreover, the length of the SID cycle becomes a function of Φ , which can be modified on-line to suite different navigational purposes.

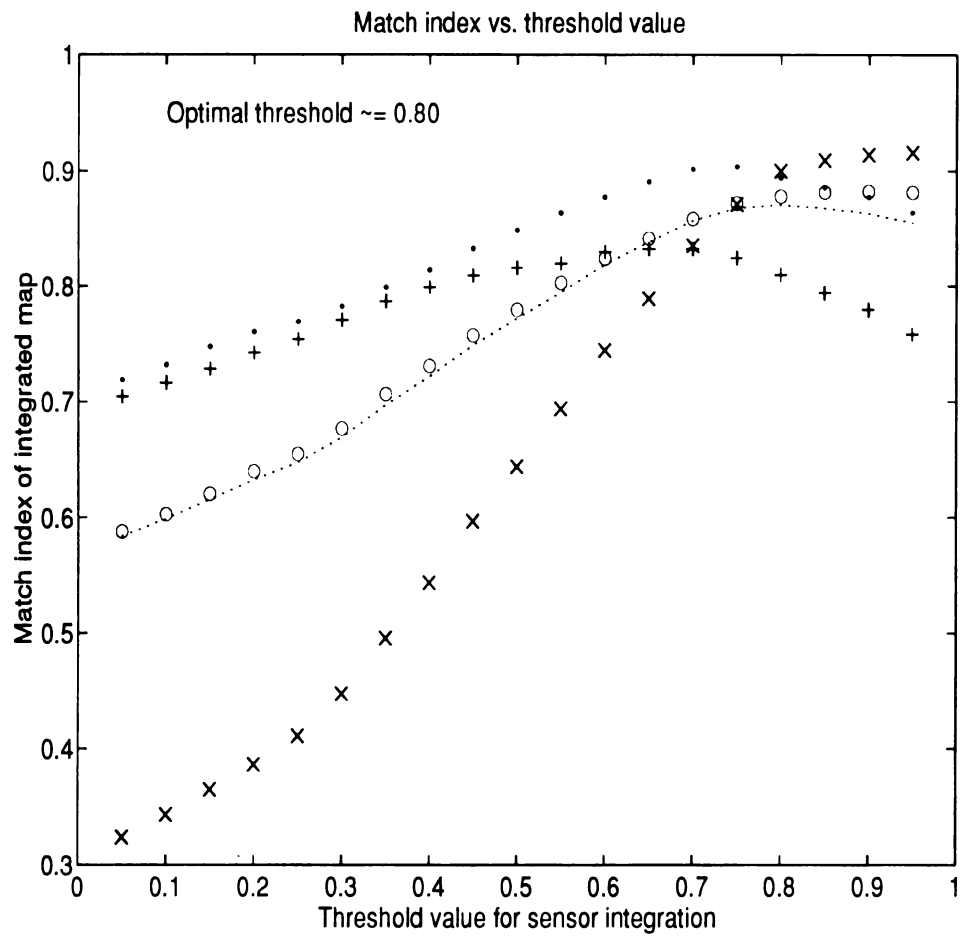


Figure 3.8. Match indices as functions of the integration threshold values.

In general, the criteria for switching from one sensor configuration to another depends on two factors, they are

- safety constraint: the mobile robot should, under no circumstances, collide into any obstacle in its workspace,
- speed constraint: the linear motion speed should be optimized when there is no obstacle present in the close vicinity of the robot.

The safety consideration actually corresponds to the capability of constructing accurate navigation map. If the robot can perceive its environment accurately, then the obstacle avoidance problem can be handled by path planning and path tracking. Obviously, the safety constraint has higher priority than the speed optimization. The relation between these two factors will be discussed in next chapter.

3.6 Summary

This chapter dealt with the theories of multi-sensor integration. First of all, the sensing system of SPARTA is described and each type of sensor is analyzed. The probability models for both ultrasonic sensors and infrared detectors are developed. By studying the characteristics of both types of sensors, we discovered the fact that the ultrasonic and infrared sensors can be used to compensate the drawbacks of each other. It is therefore desirable to integrate the occupancy probability distributions derived by each of them.

A classification of multi-sensor integration is defined. If all sensors participating the integration process provide the same type of data, then we call it Class-1 integration. On the other hand, if more than one type of sensor data is involved, then we call it Class-2 integration.

The sources of sensing errors include incorrect data and misinterpretation of sensor data. In order to evaluate the accuracy of the map constructed from sensor data, we defined the match index. This index value provides a measure of the overall mapping error which is

represented by the SSD error surface. Finally, the basic properties of the match index function are examined in detail. Several examples of multi-sensor integration performed by SPARTA are given, including the corresponding SSD surfaces and match indices.

CHAPTER 4

Adaptive Navigation Strategy

The fundamental theory of motion control for SPARTA is based on the sensing-integration-decision (SID) model introduced in section 1.2. In fact, the length of time required for SPARTA to finish each SID cycle will directly affect the maximal safety motion speed. This becomes critical when the robot's environment is dynamic and partially known, because in this case, the off-line path planning may become invalid due to unexpected obstacles moving into the way of the robot. To make things worse, some unexpected obstacles may be moving in the opposite direction of the robot's heading which further reduce the total length of the "buffer zone" available for real-time obstacle avoidance.

This chapter will focus on the relationship between sensing strategies and the corresponding motion control criteria based on the SID model.

4.1 The Sensing-Integration-Decision Cycles

In section 3.5, the concept of *sensor configuration* is developed to provide higher flexibility in the robot's sensing strategies. It is clear that each sensor configuration has its time constraint including the sensing and integration phases. That is, the total amount of time required to finish the first two phases of a SID cycle depends on the sensor configuration currently selected during that cycle. In general, the larger the amount of

sensor data the longer the SID cycle will be. So is the case when more sophisticated sensor integration model is used. In a dynamic environment with unknown moving obstacles, the length of SID cycle directly affect the maximal safety motion speed of the robot.

In general, the length of a SID cycle can be estimated as follows:

Assume that the mobile robot is equipped with sensors denoted by $S_i, i = 1, k$. The maximum range of integrated sensor data is R_s , which means any sensing information beyond this range is “unreliable”. The robot’s workspace is assumed to be partially known. This prior knowledge, together with real time sensor feedback, are used to construct and update the navigation map. To estimate the time for SID cycle, we define the following:

(i) Let $\Phi(t) = [\phi_1(t), \phi_2(t), \dots, \phi_k(t)]^T$ represents the sensor configuration vector where the value of $\phi_i(t)$ is 1 if the data from sensor S_i is used for integration at time t , and 0, otherwise.

(ii) Let $\Omega = [\omega_1, \omega_2, \dots, \omega_k]^T$ and $\omega_i = \alpha_i + \tau_i$, where α_i and τ_i represent the average data acquisition time and data processing time for sensor S_i . Note that the data processing time depends on the sensor’s characteristics and the amount of data generated. For sophisticated sensors, such as vision camera, the value of τ_i also depends on the level of features (edges, regions, etc.) the robot is interested in.

(iii) Let $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_k]^T$, where λ_i is the amount of time required to integrate the data returned by sensor i into the robot’s workspace

representation. Clearly, λ_i is a variable depending on the type of sensor i .

- (iv) Let Δ represent the total decision time for generating subsequent action plans, given the integrated workspace representation (i.e. the navigation map).

Based on above definitions, the total length of each SID cycle can be written as

$$\mu_{total} = \Psi(\Omega, \Lambda, \Phi) + \Delta \quad (4.1)$$

The mapping Ψ depends on the robot's controller architecture. For example, if the robot has one single processor for all navigational tasks, then all tasks need to be processed sequentially. In this case, (4.1) becomes

$$\mu_{total} = (\Omega + \Lambda)^T \Phi + \Delta \quad (4.2)$$

The value of μ_{total} not only indicates the time required to update the navigation map, but also provides an upper limit on robot's safe speed. First of all, let's assume the speed vectors of the robot and all obstacles are fixed during each SID cycle.

Proposition 4.1 For any SID cycle, assuming the following:

- i) the robot's position vector is $\mathbf{m}$ at the beginning of the SID cycle,
- ii) the maximum effective range of integrated sensor data is R_s ,
- iii) the initial position vector of the closest obstacle is $\mathbf{o}$,
- iv) the length of the SID cycle is μ_{total} ,

v) the minimum obstacle-free range within the region covered by integrated sensor data is γ_s , i.e. $\gamma_s = \min(R_s, \|\mathbf{m} - \mathbf{o}\|)$,

vi) all moving obstacles have bounded motion speed, i.e. $v_o \in [0, V_{obs}]$, where

$$V_{obs} < \gamma_s / \mu_{total},$$

then the highest motion speed the robot can safely take at $\mathbf{m}$ is upper bounded by

$$v_{max} \leq \frac{\gamma_s(\mathbf{m})}{\mu_{total}} - V_{obs} \quad (4.3)$$

Proof> Proof by Contradiction. Given arbitrary speed vectors $\mathbf{v}_m$ and $\mathbf{v}_o$ for the robot $\mathbf{m}$ and the closest obstacle $\mathbf{o}$, respectively, suppose a collision occurs at some $t_c \leq \mu_{total}$. That is

$$\mathbf{m} + t_c \mathbf{v}_m = \mathbf{o} + t_c \mathbf{v}_o \quad (4.4)$$

Or, equivalently,

$$\mathbf{m} - \mathbf{o} = t_c (\mathbf{v}_o - \mathbf{v}_m) \quad (4.5)$$

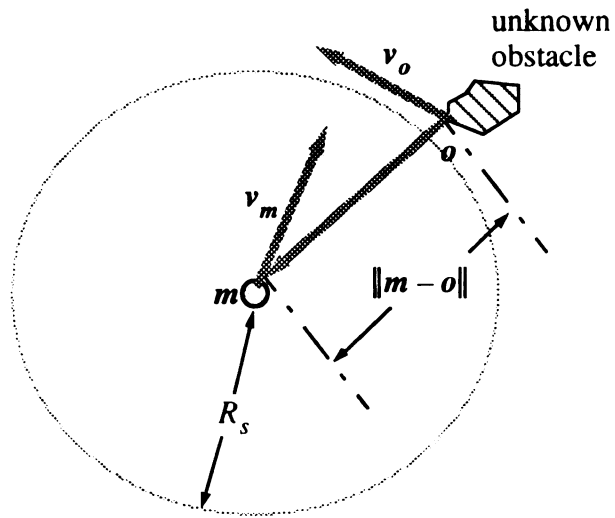
In this case, we have

$$\begin{aligned} \gamma_s \leq \|\mathbf{m} - \mathbf{o}\| &= t_c \|\mathbf{v}_o - \mathbf{v}_m\| \\ &\leq t_c (\|\mathbf{v}_o\| + \|\mathbf{v}_m\|) \\ &\leq t_c (V_{obs} + \|\mathbf{v}_m\|) \end{aligned} \quad (4.6)$$

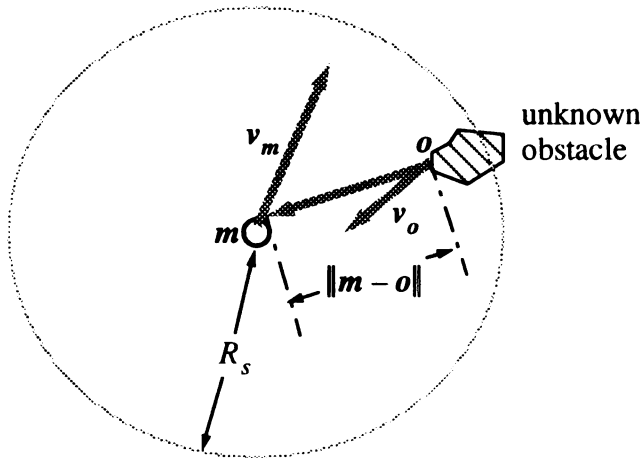
Therefore

$$\begin{aligned} \|\mathbf{v}_m\| &\geq \frac{\gamma_s}{t_c} - V_{obs} \\ &\geq \frac{\gamma_s}{\mu_{total}} - V_{obs} \end{aligned} \quad (4.7)$$

Q.E.D.



(a) $\gamma_s = R_s$.



(b) $\gamma_s = \|m - o\|$.

Figure 4.1. Two different cases in minimal obstacle-free range.

Figure 4.1 illustrates two different cases of γ_s .

In (4.3), the value of v_{max} depends on the ratio of $\gamma_s(\mathbf{m})/\mu_{total}$, which can be calculated from the estimated SID cycle length and the real-time sensor feedback. Moreover, the robot can achieve the highest possible motion speed under safety constraints when this ratio is optimized. Although the exact value of μ_{total} is not known until the end of the corresponding SID cycle, the upper bound can be computed and used to replace μ_{total} in (4.3) for a conservative estimation of v_{max} . This gives a robot speed constraint for real-world navigation processes, which can be written as

$$v_{max} < \frac{\gamma_s(\mathbf{m})}{\mu_{est}} - V_{obs} \quad (4.8)$$

where μ_{est} is the upper bound of the length of SID cycle given a specific sensor configuration. Equation (4.8) suggests the following rule:

“If robot motion speed is the criterion of highest priority, then choose the sensor configuration which minimize the value of μ_{est} .”

Given a planned path, this rule provides a feasible way of sensor-based real-time speed control. However, the complexities embedded in a dynamic workspace often force the robot to adapt the changes in its environment, such as detouring from the original path when some un-predicted obstacle gets into its way. In some cases, the robot even needs to abandon original path and perform path planning again based on the latest navigation map constructed from sensor data. These instances implies that a more sophisticated navigation algorithm is necessary for a robot to survive in the type of workspace which is dynamic and unpredictable.

4.2 Navigation Errors

In fact, dynamic changes in the robot's workspace is not the only problem during the navigation processes, an even more serious problem results from the errors that often occur during these processes. The navigation errors come from different sources, including:

- robot position/heading errors,
- robot perception errors,

4.2.1 Errors in Robot Position and Heading

The first type of errors usually comes from the dead reckoning-based position/heading estimation. For SPARTA, the readings from both wheel encoders are used to estimate the current position/heading of the robot. However, due to wheel slippage there will be rapid accumulation of errors once the robot starts to move toward its goal. According to lab experiments, the error in linear motion accumulates at a rate of 15 – 20 cm/m which is about 15 – 20% of traveled distance. Combined with the error in robot heading, it is quite usual to have SPARTA totally lost in its workspace after several moves and turns. In some cases the perceived goal position is too much off its real position that SPARTA moves into a dead end situation. For example, if the perceived goal is actually a point inside some static obstacle then SPARTA will never be able to reach its “goal”. For this reason, any autonomous navigation system must include certain reliable self-position/calibration function to correct these accumulated errors.

4.2.2 Errors in Robot Perception

The second source of errors is caused by uncertainties of the sensing processes, including the errors in sensor data and the inconsistency of data interpretation in sensor integration. As mentioned in section 3.5, the selection of sensor configuration affects the match index

of the navigation map, which is critically important to the robot's navigation plan. Just as the errors in position and heading estimation can cause the robot to crash, an erroneous map can cause severe problem for the mobile robot. To cope with this problem, many researchers proposed multi-sensor integration for map construction. From section 3.4, it shows that in most cases the match index of a map increases when more sensor data is integrated.

It is clear from the above that the sensor configuration vector Φ is critical to the overall performance of robot navigation system. The selection of Φ is subjected to

- accuracy constraints
- optimal motion speed criteria

When accuracy is the major concern, we must choose Φ such that more sensor data can be collected in each SID cycle. However, if speed is more important than accuracy, then we can choose a Φ which minimize the length of SID cycle but still provide reasonable degree of safety. In fact, there is strong evidence that humans use the same philosophy in motion control.

4.2.3 Human Responses: Some Cases

The following cases explain the analogy between human response and the robot control criteria presented above:

- Case I: *when the environment is crowded (for human) / cluttered (for robot)*

In this case, human usually slow down his/her motion and increase the level of attention to his/her environment. These responses are primarily caused by the changes in motion criteria, namely, the decrease of free space and the uncertainty about other human's actions. For a robot, this situation translates into higher

accuracy requirements. By collecting more sensor data, the accuracy requirement can be met at the cost of longer processing time, which in turn leads to lower motion speed.

- Case II: *when the environment is free of obstacle*

In this case, human response includes increasing motion speed and reducing attention to the environment. This is due to the fact that accuracy of motion becomes less important as free space becomes large. Also the concern about unexpected obstacles is no longer necessary. For a robot, this means exactly the same. Since the accuracy in motion trajectory is not important, the robot can just move at its top speed until this situation changes. Note that, however, the minimum sensory feedback is still necessary to ensure safety during navigation process.

These observations suggest a new adaptive navigation strategy that is based on the environment's degree of clutterness. In other words, if we can make the sensor configuration vector Φ adaptive to the environment of the robot, then in all cases the robot will be able to optimize its motion speed without violating the accuracy constraint. Following the definitions in section 4.1, a new estimation of μ can be derived. The estimated μ is similar to (4.2), except that the vector Φ becomes a function of the robot workspace, specifically, the region around the mobile robot itself. This dependence of Φ on the workspace is described quantitatively by an Obstacle Density Function (ODF) which is a measure of "clutterness" of the workspace. The ODF is described next.

4.3 Obstacle Density Function

In order to measure the "clutterness" of the robot's environment, we need to formally define the obstacle density function:

Definition 4.1 Given a robot's workspace W , for any robot position $\mathbf{m} \in W$, the *obstacle density function* (ODF) is

$$G(\mathbf{m}) = \frac{1}{\text{area}(A)} \left[\int_A g(x, y) dA \right] \quad (4.9)$$

where $g(x, y)$ is the occupancy function over the workspace, and $A \subset W$ is the region of interest with respect to robot position $\mathbf{m}$. A typical selection of A is a circular disc centered at the robot with radius equal to R_s .

Given the definition of ODF, the next question is what type of occupancy function $g(x, y)$ should be used. As a matter of fact, how the occupancy function $g(x, y)$ is built is not important for this discussion, as long as it represents the obstacle occupancy of the workspace and satisfies the following condition

$$g(x, y) \in [0, 1], \forall (x, y) \in A.$$

Some examples are the HIMM algorithm proposed by Borenstein *et al.* [13], and the probability model proposed by Elfes [20].

4.3.1 Discrete Obstacle Density Function

Note that if the occupancy function $g(x, y)$ is in discrete form, i.e. $g(i, j)$ where i and j are indices to grid cell, then the ODF can be defined in discrete form.

Definition 4.2 Given the robot position $\mathbf{m}$, the *discrete obstacle density function* of a set of connected cells A is

$$G(\mathbf{m}) = \frac{1}{\text{area}(A)} \sum_{p(i, j) \in A} g(i, j) \quad (4.10)$$

where $\text{area}(A)$ is equal to the number of cells in A and $g(i, j)$ is the discrete occupancy

function at cell (i, j) .

Under the new definition, the ODF can be considered as the percentage of cells in set A which are occupied.

4.3.2 The Weighted Obstacle Density Function

As mentioned in section 4.2.3, humans react to changes in degree of clutter by modifying the level of sensory feedback as well as the motion plan. However, if we examine the human behavior in a cluttered area more closely, it is clear that the obstacles in the front are perceived more important than those in the back. This is because the current heading determines the region of high interest, i.e. the path leading to the goal. The same principle applies to mobile robot navigation. In fact, any obstacle outside the $\pm 90^\circ$ envelope of the current heading is not critical to the navigation task.

Definition 4.3 Given the circle representing the region covered by integrated sensor data, the *critical region* of a mobile robot with heading $\mathbf{h}$ is the sector of that circle bounded by the two radii which are perpendicular to $\mathbf{h}$.

Given the above, it is possible to modify the definition of ODF to assign more weight to the obstacles inside the critical region.

Definition 4.4 Given the robot position $\mathbf{m}$ and its heading vector $\mathbf{h}$. The *weighted obstacle density function* (WODF) with respect to the cell $(i, j) \in A$ is

$$G(\mathbf{m}, \mathbf{h}) = \frac{1}{\sum_{p(i,j) \in A} \sum w_{i,j}(\mathbf{m}, \mathbf{h})} \left[\sum_{p(i,j) \in A} \sum w_{i,j}(\mathbf{m}, \mathbf{h}) \cdot g_{i,j} \right] \quad (4.11)$$

$$\text{where } w_{i,j}(\mathbf{m}, \mathbf{h}) = \cos\left(\frac{\angle[\overrightarrow{(i,j)} - \mathbf{m}, \mathbf{h}]}{2}\right).$$

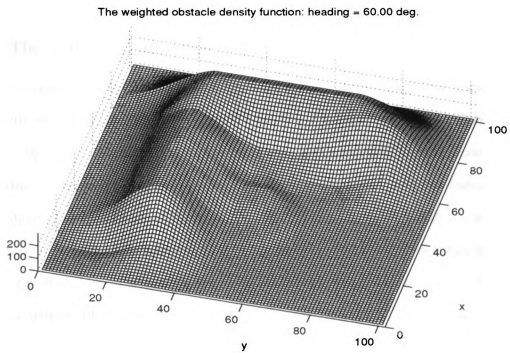
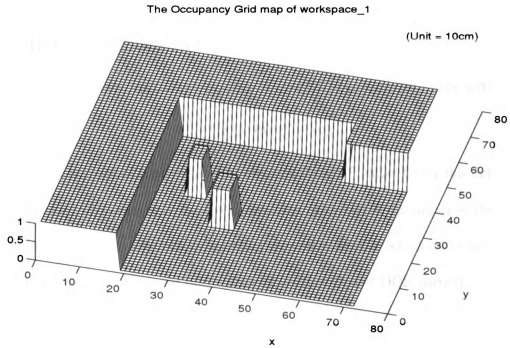


Figure 4.2. An example of WODF surface: the exact occupancy grid map and the corresponding WODF surface with robot heading = 60 degrees.

An example of ODF surface of SPARTA's workspace is shown in Figure 4.2.

4.3.3 SID Cycle Estimation Based on ODF

Once an ODF is defined over the workspace W , the SID cycle model can be written as

$$\mu(G) = \Psi(\Omega, \Lambda, \Phi(G)) + \Delta + \kappa \quad (4.12)$$

where κ is the amount of time required to measure the ODF value. Using this definition, a similar result of (4.3) can be derived as a function of the ODF. In this case, the optimal navigation speed can be achieved by selecting the vector $\Phi(G)$ which maximize the ratio of $\gamma_s(p) / \mu(G)$. Therefore, the optimal speed is a function of ODF, namely

$$v_{max}(G) < \frac{\gamma_s(m)}{\mu(G)} - V_{obs} \quad (4.13)$$

where $\gamma_s = \min(R_s, \|m - o\|)$ is the minimal obstacle-free range.

4.4 The Switching Algorithm

The key issue of adaptive navigation strategy is how should the robot respond to the geometrical changes in its environment. To address this issue, we need to understand the robot's capabilities in perceiving its environment and the possible actions it can take.

In terms of task-oriented perception, vision is by far the most powerful tool for understanding the environment because of the richness of information it can provide. However, it is well understood that some fundamental capabilities of visual perception, which can be achieved with little effort by most high level animal species, don't come easy for artifacts like mobile robots. So far, most mobile robot still heavily rely on simple sensing devices to collect critical information such as ranging data. The problem of this approach is that only sparse information can be accessed by the robot, which leads to poor resolution or even inadequate description of the environment. However, from the speed

point of view, smaller amount of sensor data implies faster processing and decision making. Thus when the accuracy of mapping is not crucial, it is feasible to reduce the number of sensing operations such that only critical data is collected. In this way, the length of SID cycles can be shortened, and according to (4.3), the robot's motion speed can be increased.

The robot's response to the changes in its environment includes the following categories:

- change of motion plan, including speed and heading:

This is the basic function of robot motion control, which is essential to all types of navigation tasks.

- change of sensor configuration:

By selecting suitable sensor configuration, it is possible to control the length of SID cycle in order to improve the overall performance of the robot.

- change of focus on geometrical features:

Different geometrical features correspond to different types of navigation methods, and they can be extracted from different sources of sensor data. For example, the detection of obstacle boundaries is the main focus of sensor-based real-time obstacle avoidance, while the landmark patterns are of major concern when the robot needs to calibrate its position/heading. Note that the combination of different features may be necessary in some cases.

Based on the above, the following conclusion can be drawn:

Autonomous robot reaction based on ODF is one way of adapting

environmental changes where neither the safety of navigation nor the speed performance will be sacrificed.

The adaptive navigation strategy can be described by the algorithm listed in Table 4.1. First of all, the range of ODF value (i.e. $[0, 1]$) is partitioned and assigned to each of the sensor configurations. The robot measures the ODF constantly while moving toward its goal, and switches among the corresponding sensor configurations dynamically. Note that the increase/decrease of sensor configuration level is determined by the combination of current ODF value and the difference between its current value and previous value. Specifically, the difference in ODF (which is denoted by ΔG) is tested with threshold values, namely Λ^+ for positive cases and Λ^- for negative cases, to determine whether a level switching is truly necessary. This is designed to prevent the “oscillation” between two adjacent levels when the ODF values bounce back and forth on the “boundary” between these two levels.

The experimental results of 3 test runs conducted on SPARTA mobile robot, including three robot trajectories and the speed / ODF curves corresponding to the third trajectory, are presented in Figure 4.3. These test runs are conducted under three different maximum speed settings, namely 450 mm/sec, 550 mm/sec, and 850 mm/sec. Note that two unknown obstacles are present in the experimental setup to test the responses of the robot during obstacle avoidance maneuver. The robot trajectories are illustrated with time marks to indicate the time-position relations during the navigation processes. From the speed and ODF curves of the third test run, we observe that the robot tends to achieve maximum speed when the ODF measurement is very low (eg. between time mark No. 5 and No. 15). On the other hand, if the ODF is high then the robot speed is greatly reduced (eg. between time mark No. 20 and No. 40). Using the ANS algorithm, SPARTA has achieved an average speed of 416.7 mm/sec under this experimental setup.

Table 4.1. The adaptive navigation based on ODF.

```

algorithm adaptive_nav_ODF

begin;
{
    do
    {
        compute robot position  $\mathbf{m}_i$ ;

        measure ODF at  $\mathbf{m}_i$ , call it  $G_i(\mathbf{m}_i)$ ;

         $\Delta G_i = G_i(\mathbf{m}_i) - G_{i-1}(\mathbf{m}_{i-1})$ 

        if ( $\Delta G_i \geq \Lambda^+$ ) then
            increase sensor configuration level;

        elseif ( $\Delta G_i \leq \Lambda^-$ ) then
            decrease sensor configuration level;

        else
            use same sensor configuration level;
        endif

        compute the estimated SID length  $\mu(G_i)$ ;

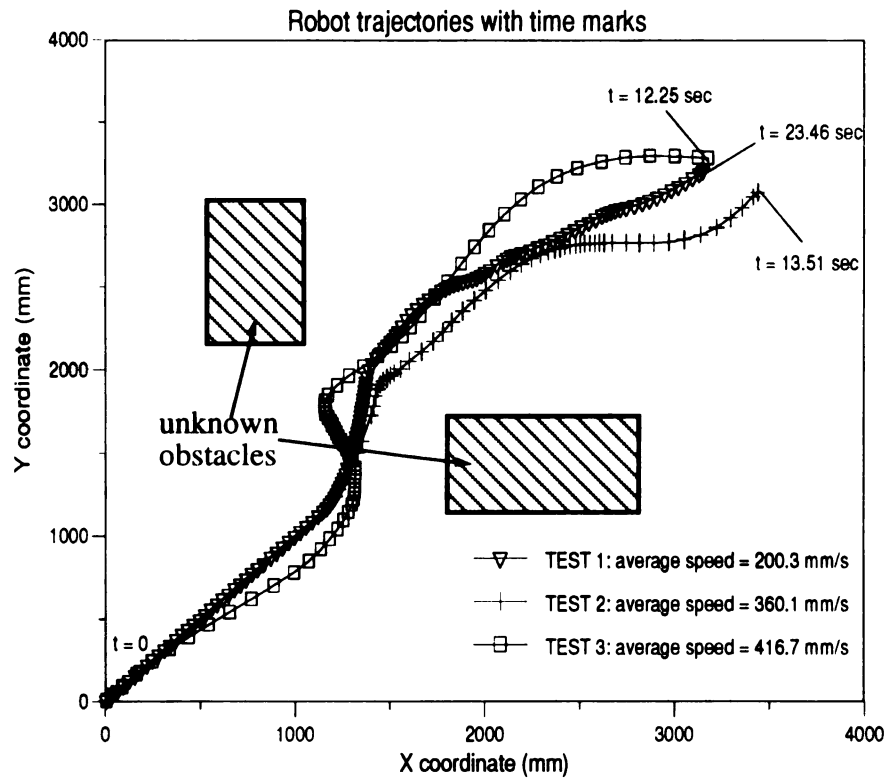
        compute the speed limit  $v_{max}$  from (4.11);

        set robot speed to  $v_{max}$ ;

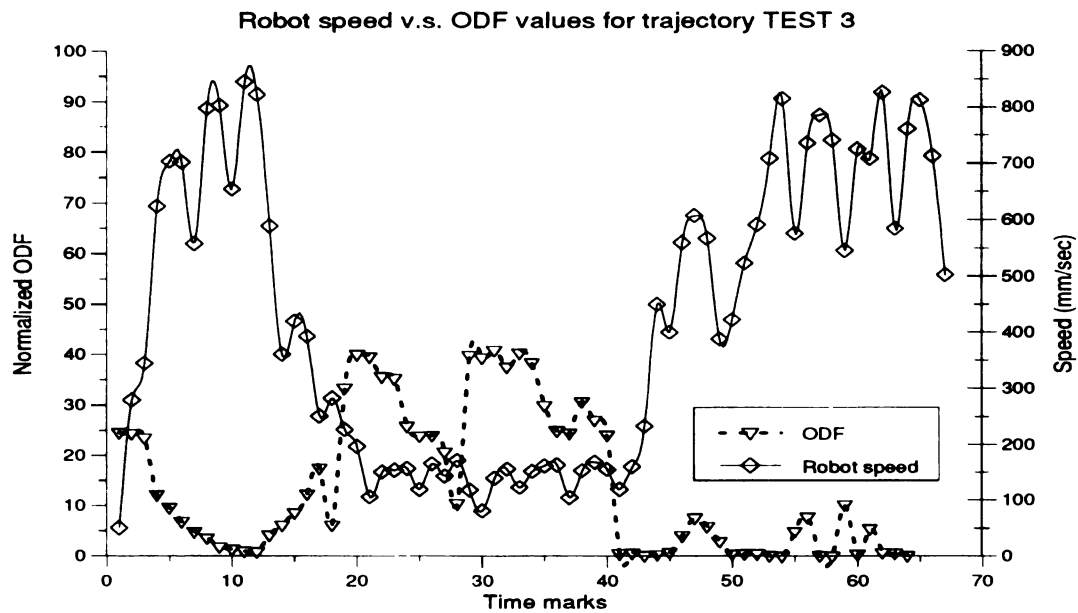
        perform next SID cycle;
    } while (NOT goal_reached);
}

end;

```



Robot trajectories of three test runs.



The speed and ODF curves of test run No. 3.

Figure 4.3. Experimental results of the Adaptive Navigation Strategy.

4.5 Summary

Adaptive navigation strategy provides a feasible solution to navigation problems in dynamic environments with unknown moving objects. First of all, the length of a SID cycle can be estimated by considering the time requirement for each individual step in that cycle. Given that estimation, we can calculate the upper limit of the motion speed under accuracy constraints.

The sources of navigation errors have been identified, their effects on the overall navigation process are discussed. As a contrast to the robot intelligence, the human responses to the changes in the environment are examined. This leads to the idea of having the robot react to the changes in clutter of its environment by adjusting its motion plan as well as sensor configuration. The clutter of the environment can be measured by the obstacle density function defined in this chapter. Several variations of the ODF's are developed for different focusing in the robot's perception. At the end of this chapter, we presented an algorithm for the dynamic switching of motion speed and sensor configuration based on real-time measurement of the ODF.

CHAPTER 5

Robot Vision System

5.1 Characteristics

Unlike the non-visual sensors commonly used in robotics systems, a visual sensor is intended for probing the robot's environment by measuring the light intensity perceived from its angle of view. The output of a visual sensor contains arrays of data, or images, where each one of them describes a specific angle of view of the robot's environment.

A typical vision system consists of visual sensor, image frame grabber / buffer, and a set of image processing functions. For robot navigation tasks, the vision data needs to be processed to provide high level geometrical information such as the location of landmark, etc. Standard image processing techniques can be applied to extract critical features from the images taken by the vision system.

5.2 Imaging Geometry of a Vision System

The imaging geometry of most cameras can be approximated by the point projection model of a pinhole camera. Through this model, it is possible to transform any point in the workspace from its world frame coordinates to its corresponding coordinates in the image plane. This transformation is often described by the projection equations.

Given the world frame coordinate system $F(i, j, k) \subset R^3$, where i, j , and k are the unit vectors of the X, Y , and Z axis. Suppose c is the camera's position with respect to the

world frame, $\mathbf{n}$ is the unit normal vector of the image plane, and $\mathbf{u}$, $\mathbf{v}$ are the unit vectors in horizontal and vertical directions, respectively, then the projection equations are

$$\frac{I}{f} = \frac{(\mathbf{p} - \mathbf{c}) \cdot \mathbf{u}}{(\mathbf{p} - \mathbf{c}) \cdot \mathbf{n}} \quad \text{and} \quad \frac{J}{f} = \frac{(\mathbf{p} - \mathbf{c}) \cdot \mathbf{v}}{(\mathbf{p} - \mathbf{c}) \cdot \mathbf{n}} \quad (5.1)$$

where $\mathbf{p} = xi + yj + zk$ is the position vector of point (x, y, z) , and (I, J) is the projection coordinates of $\mathbf{p}$ in the image plane. For the vision system of SPARTA, the camera is mounted on the top of a turntable with fixed height and pan control which allows 360 degree view. Therefore, the camera frame is related to the robot base frame by rotating about the Z axis and translating in Z direction [10].

Figure 5.1 illustrates the model of point projection in a pinhole camera. The geometrical relation between the camera frame and robot frame is shown in Figure 5.2.

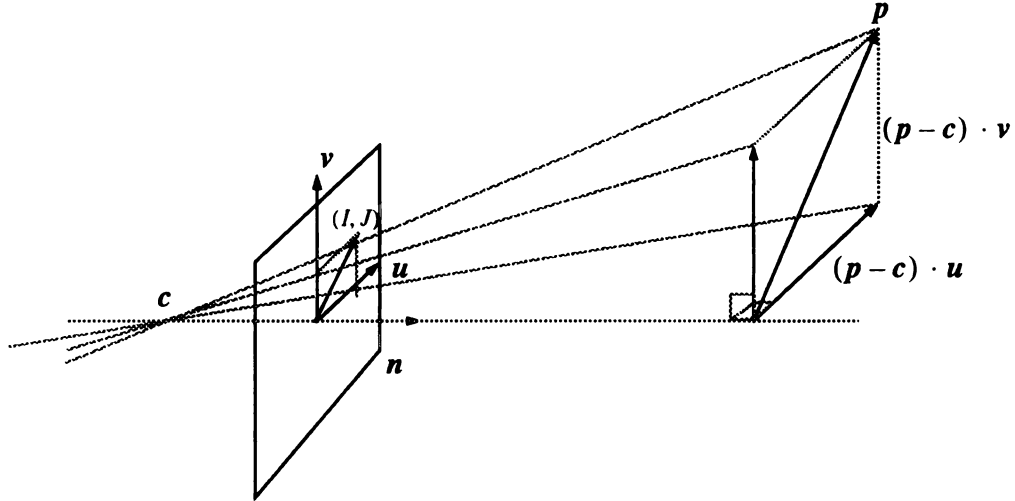


Figure 5.1. Point projection in a pinhole camera.

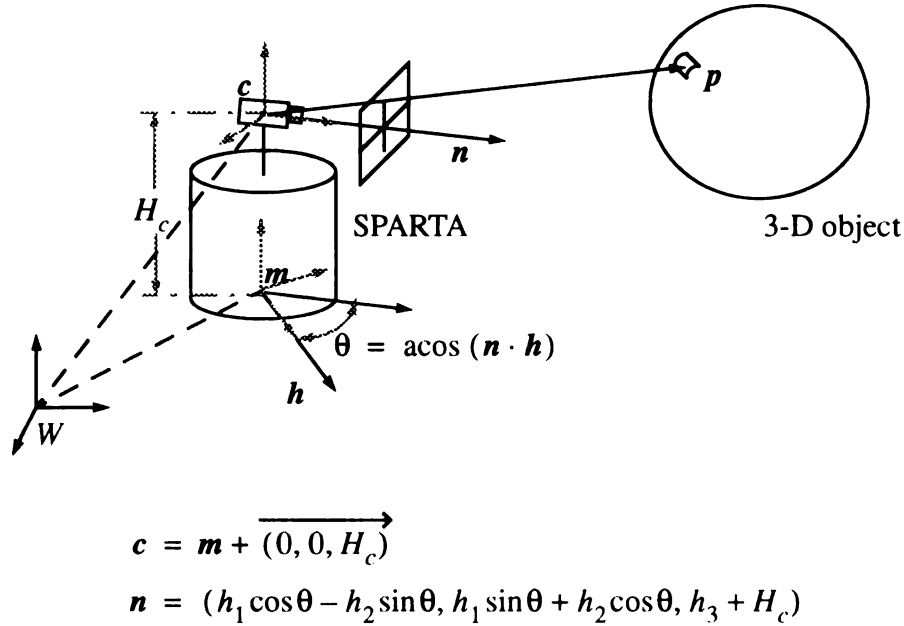


Figure 5.2. The geometry of vision system on SPARTA.

5.3 Navigation Using Landmarks

Unlike the AGV's, which rely on fixed path guidance, a mobile robot needs to “figure out” its own position and heading with respect to the world frame. Some mobile robots use the dead reckoning method to move from one position to another. The only way to estimate their positions in the workspace is to monitor the readings of the wheel encoders attached to their drive wheels. However, due to the slipping between the robot's wheels and the ground surface, the dead reckoning method often gives erroneous estimation which leads to failure of navigation task. To make things worse, the position error accumulates as the robot moves. To solve this problem, many researchers proposed the usage of navigation landmarks to help the robot calibrate itself. The mobile robot uses its vision system to recognize and locate the landmarks, whose positions in the world frame are known in advance, and then calculate its own position based on its coordinates relative to the landmarks. This approach requires the robot to perform calibration every time when it has

traveled certain distance from the last calibration point.

The problem of self-calibration can be formally defined as follows:

Definition 5.1 *The robot self-calibration problem:* Given all the variable definitions in Figure 5.2, the landmark coordinates in the world frame, the coordinates of landmark image in the image frame, and the pan angle of the camera with respect to the robot frame, find the robot's position $\mathbf{m}$ and heading $\mathbf{h}$ with respect to the world frame.

One solution to this problem involves solving the system of projection equations corresponding to at least 3 different points in the 3-D space [1, 27]. However, this approach is not efficient due to the fact that 1) ambiguity arises because usually more than one solution will be found, 2) finding one-to-one correspondence between points in the workspace and their projections in the image plane is difficult, especially when noise is present in the image. Several ad hoc solutions have been proposed to overcome the difficulties, for example, using a well defined geometric shape as the landmark pattern, setting all landmarks at the same height as the camera, adding 3-D features to the landmark to provide depth information, etc. All these topics will be discussed in the rest of this chapter.

5.4 Landmark Patterns and Recognition Strategies

A landmark usually consists of a set of simple patterns with known dimensions and geometrical layout. The design of landmark patterns must meet the following criteria:

- Use only high contrast colors, e.g. black pattern on white background.
- Choose the pattern whose parameters can be easily extracted by standard pattern matching / recognition algorithms, e.g. circular disks.
- The pattern itself should not resemble any other frequently seen objects which are not used as landmarks, e.g. rectangular boxes.

In addition to these principals, the landmarks used by SPARTA are set up at the same height as the camera onboard. This constraint greatly reduced the computation time required as well as the possibility of false recognition.

The primary pattern of SPARTA's landmarks is a black circular disk on white background.

The major reasons for choosing a circular disk are

- the projection of a circular disk in the image plane can always be approximated by an ellipse, which is easy to recognize using elliptical Hough transformation technique
- a circular pattern does not get mixed up easily with the majority of patterns frequently seen in an indoor environment, such as rectangular boxes, polygonal objects, etc.
- a circular pattern is more robust to noise and occlusion than polygonal patterns during template matching process, even though all these patterns can be detected by using Hough transformation technique

Also, a white strip of 5 cm width is painted on the vertical central axis of the primary disk. This strip will be used for locating the center of the disk, as will be discussed later in this chapter. A secondary pattern consists of a smaller white disk which shares the same center axis of the primary pattern, but has been suspended 12 cm away from the plane of primary pattern. The secondary pattern is used to determine which side of the center axis of the pattern is the robot "looking" from. All cases are illustrated in Figure 5.4. The diameters of the primary and secondary disk patterns are 50 and 10 cm, respectively. The center axis shared by both disks is parallel to the floor of the robot's workspace and set to the same height of the camera. Figure 5.3 illustrates the landmark patterns and their dimensions.

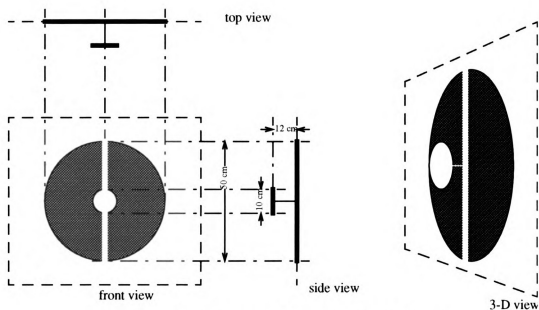


Figure 5.3. The landmark pattern used by SPARTA.

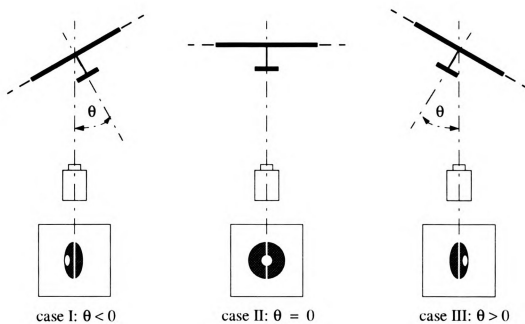


Figure 5.4. Using the secondary pattern to decide the aspect of the landmark.

5.5 Self-Positioning

As mentioned in the previous section, circular disks are used as landmark patterns to SPARTA. This type of landmarks can be detected by searching for elliptical patterns in the image frame. In fact, the projection of a circular disk on the image plane is not exactly an ellipse. However, in most cases this approximation is good enough for navigation purposes. The errors caused by this approximation will be discussed later in this chapter. Two basic assumptions are made about the landmarks used by SPARTA:

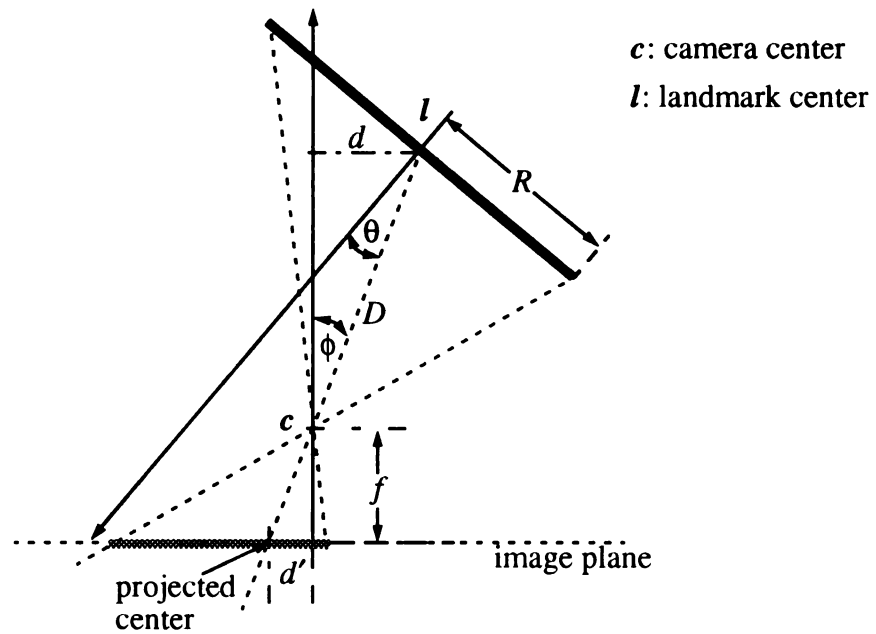
- the plane on which the primary disk pattern resides is parallel to the pan-axis of the camera
- the center of the primary disk pattern is at the same height as the camera axis

In order to find the relative position and heading of the robot, we need to find the relation between the parameters of the approximating ellipse and the aspect angle and distance of the primary disk.

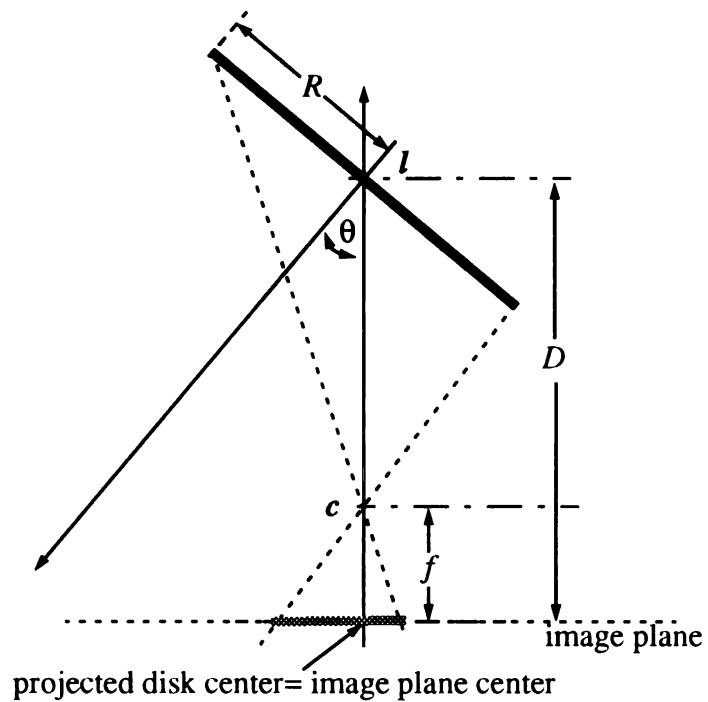
5.5.1 Finding the Center of the Primary Disk

As pictured in Figure 5.5(a), the general cases of disk projection is too complex to solve directly, instead, a more efficient model can be established if we align the camera such that the camera axis passes through the center of the disk pattern. To do so, we need to scan for the vertical strip inside the elliptical region once that ellipse has been found. This can be done by applying any edge enhancement operator to the image. Note that the strip span across the disk so that the detection is robust even if part of the ellipse is occluded. Given the horizontal displacement of the center projection point, which is d' in Figure 5.5(a), it follows that

$$\phi = \text{atan}\left(\frac{d'}{f}\right).$$



(a)



(b)

Figure 5.5. Top views of disk projection model, (a) the general cases, and (b) when camera axis passes through the disk center.

Therefore, by rotating the camera with $-\phi$, the camera axis will pass through the center of the disk. The new projection model has much simpler geometry, as shown in Figure 5.5(b).

It is clear that the major axis of the approximating ellipse is always perpendicular to the horizontal axis of the image plane. This is due to the fact that the vertical axis of the image plane is parallel to the plane of the primary disk. The longest projection of the diameter always occur in the vertical direction of the image plane.

Given the radius of the primary disk R , we can calculate the length of the major axis of the approximating ellipse, denoted by $2b$, from

$$2b = \frac{2fR}{D} \quad (5.2)$$

where $D = \|l - c\|$ is the distance between the camera and the center of the landmark pattern. It follows that $D = fR/b$ is the relative distance of the landmark.

Also, the minor axis, which is denoted by $2a$, is given by

$$\begin{aligned} 2a &= R \cos(\theta) \left(\frac{f}{D + R \sin(\theta)} \right) + R \cos(\theta) \left(\frac{f}{D - R \sin(\theta)} \right) \\ &= \frac{2fDR \cos(\theta)}{D^2 - R^2 \sin^2(\theta)} \end{aligned} \quad (5.3)$$

It may seem quite difficult to solve θ , however under the assumption of large distance, that is

$$D \gg R \sin(\theta), \quad (5.4)$$

then (5.3) can be simplified as

$$a \approx \frac{fR \cos(\theta)}{D}. \quad (5.5)$$

The aspect angle of the landmark with respect to the camera frame is, therefore

$$\theta = \arccos\left(\frac{aD}{fR}\right). \quad (5.6)$$

The next question is how does the robot know there is an ellipse, which could be the landmark's projection, in the image acquired? If the robot believes that it has found a legitimate ellipse, how does it extract the parameters, i.e. the major and minor axes, from the image? The answers to these two questions share the same idea, that is the use of *Elliptical Hough Transformation* (EHT).

5.5.2 The Modified Elliptical Hough Transformation

The EHT is quite similar to the ordinary Hough transformation used for the detection of circular patterns except the dimension of Hough array is increased by one because there is one more feature axis for the elliptical patterns. Figure 5.6 demonstrates the EHT approach. Note that by assuming the landmark is at the same height as the camera, it is possible to further reduce the dimension of the Hough array from four to three.

Beside the above assumption, if the gradient information from edge detection stage is used, then the volume needed to be updated in the Hough array will be greatly reduced. To do so, SPARTA uses Sobel edge detector to extract the edge information, and the gradient vector at each edge pixel is computed by finding the normal of the tangent line passing through that pixel. By using the least square line fitting method, it is possible to find a close approximation of the tangent line passing through each edge pixel. We choose to apply a 5 x 5 window centered at each edge pixel through which the slope of the tangent line will be calculated. Finding the slope of the tangent line has twofold meanings:

- it provides information about the possible range of the center of the ellipse;
- it also gives the possible range of the length of the major axis.

These points will be discussed later in this chapter. Let us focus on finding the slope of tangent lines for now. First of all, the tangent line at each point on the ellipse can be found

by using the LSSD (*Least-Sum-of-Squared-Distance*) approach.

5.5.3 LSSD Line-Fitting

The problem of LSSD line fitting can be formulated as follows:

Given an edge pixel p_0 , a window of size $n \times n$ centered at p_0 , and all of the k edge pixels inside the window except p_0 itself, call them $p_i, i = 1, k$, find a line L through p_0 that minimize the sum of squared distance

$$\sum_{i=1}^k [dist(L, p_i)]^2. \quad (5.7)$$

Since p_0 is at the center of the window, it is convenient to assign the coordinate $(0, 0)$ to p_0 and (x_i, y_i) to $p_i, i = 1, k$, where x_i and y_i are the pixel displacement of p_i w.r.t p_0 in the horizontal and vertical directions, respectively.

Assuming the slope of L is s , then the equation for L is $sx - y = 0$, and the distance from any point $p_i = (x_i, y_i)$ to L is

$$d_i = \left(\frac{1}{s^2 + 1} \right) \sqrt{(s^2 x_i - s y_i)^2 + (y_i - s x_i)^2}. \quad (5.8)$$

The sum of squared distance is

$$\begin{aligned} \eta &= \sum_{i=1}^k d_i^2 \\ &= \sum_{i=1}^k \frac{(s^2 x_i - s y_i)^2 + (y_i - s x_i)^2}{(s^2 + 1)^2} \end{aligned} \quad (5.9)$$

To find the minimum of squared distance, we differentiate η w.r.t s , that is

$$\begin{aligned}
\frac{d\eta}{ds} &= \sum_{i=1}^k \frac{2(s^2+1)}{(s^2+1)^4} [x_i y_i s^4 + (x_i^2 - y_i^2) s^3 + (x_i^2 - y_i^2) s - x_i y_i] \\
&= \sum_{i=1}^k \frac{2}{(s^2+1)^2} [x_i y_i s^2 + (x_i^2 - y_i^2) s - x_i y_i] \\
&= \frac{2}{(s^2+1)^2} \left[\left(\sum_{i=1}^k x_i y_i \right) s^2 + \left(\sum_{i=1}^k (x_i^2 - y_i^2) \right) s - \left(\sum_{i=1}^k x_i y_i \right) \right]
\end{aligned} \tag{5.10}$$

Setting $\frac{d\eta}{ds} = 0$, we have

$$s = \frac{-\beta \pm \sqrt{\beta^2 + 4\alpha^2}}{2\alpha}, \tag{5.11}$$

where $\alpha = \sum_{i=1}^k x_i y_i$ and $\beta = \sum_{i=1}^k (x_i^2 - y_i^2)$.

The values of s in (5.11) correspond to the maximal and minimal values of the squared distance. To solve the ambiguity, we find the second derivative of η

$$\begin{aligned}
\frac{d^2\eta}{ds^2} &= \frac{2}{(s^2+1)^4} \left[(2\alpha s + \beta) (s^2+1)^2 - 4s (s^2+1) (\alpha s^2 + \beta s - \alpha) \right] \\
&= \frac{2}{(s^2+1)^3} [-2\alpha s^3 - 3\beta s^2 + 6\alpha s + \beta]
\end{aligned} \tag{5.12}$$

Defining the test equation $\delta(s) = -2\alpha s^3 - 3\beta s^2 + 6\alpha s + \beta$, it is straightforward to find which value of s corresponds to the minimal η by examining the sign of $\delta(s)$.

For each edge pixel, given the slope of the tangent line L , the gradient vector is simply the normal vector of L . By simple calculation, we know that the intersection point of the gradient vector emitting from the edge pixel and the x axis is $(I + sJ, 0)$. The intersection point of the x axis and the major axis of any legitimate ellipse lies in the

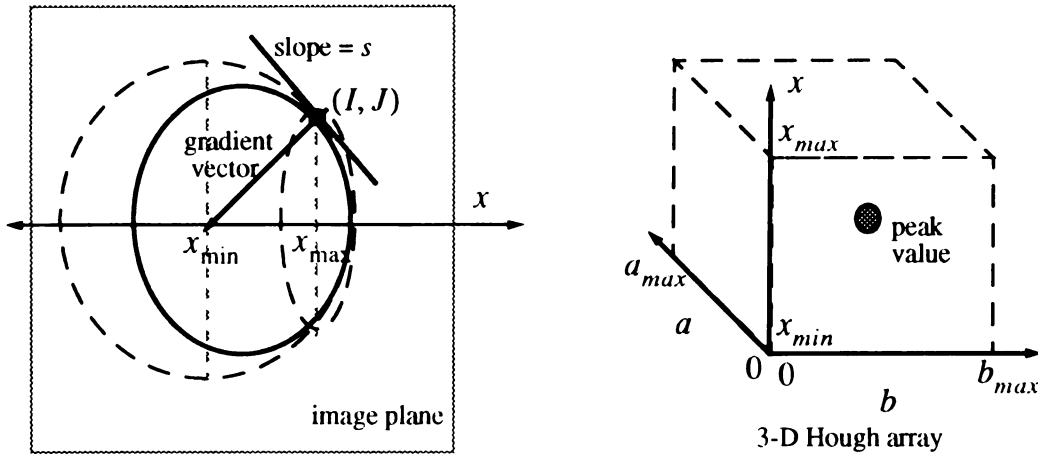


Figure 5.6. The Modified Elliptical Hough Transformation with gradient information and the corresponding Hough array.

closed interval $[I + sJ, I]$. This is shown in Figure 5.6.

Until now, only the following information is used to decide which portion of the Hough array needs to be updated:

- the coordinates of the edge pixel currently considered in the image
- the gradient vector at the edge pixel

However, it is possible to further reduce the number of updates in the Hough array by applying the constraints on the major and minor axis. Consider the class of ellipse which satisfy the following conditions

- the major axis is perpendicular to the x axis
- the center of the ellipse lies on the x axis

The general equation for this class of ellipse can be written as

$$\frac{(x-x_0)^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (5.13)$$

Differentiating w.r.t x , we have

$$\frac{2(x-x_0)}{a^2} + \frac{2y}{b^2} \cdot \frac{dy}{dx} = 0 \quad (5.14)$$

Clearly, $\frac{dy}{dx}$ is the slope of the line tangent to the ellipse at (x, y) , therefore

$$\frac{a}{b} = \sqrt{\frac{(x_0-x)}{sy}}. \quad (5.15)$$

This is a necessary condition for the point (x, y) to lie on the ellipse with center $(x_0, 0)$ and the ratio of major/minor axis equals to a/b . In other words, we only need to update the cells with index $[x_0, a, b]$ that satisfy the above equation. The next question is “what is the possible range of a (or b , since these two values are related by (5.15))?” A simple estimation gives the possible range of a as $[y, y - s(x - x_0)]$.

The algorithm listed in Table 5.1 summarizes the MEHT used by SPARTA:

Once the ellipse is found in the image plane, the corresponding parameters, namely the lengths of major and minor axes, are also given. The next step is to find the relative position of the robot with respect to the landmark. This can be done by calculating the distance and the aspect angle of the landmark. The distance between the center of the camera and the center of the landmark is

$$D = \frac{fR}{b} \quad (5.16)$$

Given the distance, the aspect angle of the landmark with respect to the camera frame can be found by solving (5.3) where $\theta = (\pi/2) - \alpha$. However, if $D \gg R \sin(\theta)$, then the

Table 5.1. The Modified Elliptical Hough Transformation (MEHT).

```

algorithm MEHT          /* the Modified Elliptical Hough Transformation */
/*
    assuming the following:
    (i)  $I_{edge}(x, y)$  is the edge enhanced image by applying the Sobel operator to
        the intensity image
    (ii) the legitimate range of major and minor axis length are  $B_{range}$  and  $A_{range}$ 
*/
begin
for all edge pixels  $p(x, y) \in I_{edge}$  , do
{
    calculate the slope of the tangent line
        at  $p(x, y)$  , call it  $s$  ;

    calculate the gradient vector at  $p(x, y)$  ;
    if  $s \geq 0$  then       $X_e = [x, x + sy]$  ;
    else                  $X_e = [x + sy, x]$  ;

    for all  $x_0 \in X_e$  , do
    {
         $A_e = [y, y - s(x - x_0)]$  ;

        for all  $a \in A_e$  , do
        {

```

Table 5.1. (continued)

```


$$b = a \frac{sy}{(x_0 - x)} \sqrt{\frac{x_0 - x}{sy}};$$

if  $b \in B_{range}$  then      /* increment the Hough entry
                               corresponding to  $x_0$ ,  $a$ , and  $b$  by
                               1 */
     $H[x_0][a][b]++;$ 
}
}

}

find  $\text{Max}(H[x_0][a][b])$  for all  $x_0 \in X_{range}$  ,  $a \in A_{range}$  ,
    and  $b \in B_{range}$  ;

/*    an ellipse is found at  $(x_0, 0)$  with major and minor axis equal to  $a$  and  $b$  ,
    respectively */
end

```

solution can be approximated by

$$\theta \approx \arccos\left(\frac{a}{b}\right) \quad (5.17)$$

The ambiguity associated with the sign of θ can be resolved by scanning along the minor axis of the projection ellipse on the image plane to locate the projection of the secondary disk pattern. This has been shown in Figure 5.4. Given θ , D , and the “absolute position” of the landmark, the robot can calibrate its position and heading with respect to the world frame.

5.5.4 MEHT Robustness Test

The MEHT approach has demonstrated its robustness under the situations where the landmark patterns are partially occluded. Table 5.2 and Table 5.3 summarize the experimental results under different landmark aspect angles and percentages of occlusion. The measured aspect angle error in Table 5.2 is defined as

$$\text{Measured aspect angle error} = |\text{Actual aspect angle} - \text{Measured aspect angle}|$$

where the measured aspect angle can be calculated from the parameters of the approximating ellipse. The values of $2b/h$ and $2a/w$ are listed in Table 5.3.

The test result shows that the MEHT is reliable even when the landmark is 50% occluded. For higher percentages of occlusion, however, the MEHT might give errors in the aspect angles of up to 15 degrees. In particular, for Case 7 in Table 5.3, the estimated dimensions of the landmark projection tend to be much smaller than the actual values. This problem is caused by the fact that the resulted landmark projection is very similar to a smaller ellipse which consequently produced higher response in the Hough array than the original landmark pattern. An example of landmark recognition, including the landmark image and the corresponding MEHT array, is presented in Figure 5.7 - Figure 5.9.

Table 5.2. Errors in the measured aspect angle of the landmark due to occlusion.

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Measured aspect angle error (°)	0% occlusion 	25% occlusion 	25% occlusion 	50% occlusion 	50% occlusion 	75% occlusion 	75% occlusion 	75% occlusion 
$\theta = -30^\circ$	0.5	1.5	2.9	1.5	2.3	2.5	11.4	12.8
$\theta = -45^\circ$	3.6	1.6	4.3	2.3	6.9	2.4	10.1	14.4
$\theta = -60^\circ$	0.1	4.0	4.0	5.2	4.2	12.4	7.3	7.5

$$\text{Measured aspect angle error} = |\text{Actual aspect angle} - \text{Measured aspect angle}|$$

Table 5.3. Estimation of landmark projections using measured elliptic parameters.

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
$\left(\frac{2b}{h}, \frac{2a}{w}\right)$	0% occlusion 	25% occlusion 	25% occlusion 	50% occlusion 	50% occlusion 	75% occlusion 	75% occlusion 	75% occlusion 
$\theta = -30^\circ$	1.00, 1.03	0.92, 0.96	0.87, 0.87	0.95, 0.97	0.97, 1.03	0.89, 0.90	0.78, 1.43	0.84, 0.73
$\theta = -45^\circ$	1.00, 1.06	0.92, 0.94	0.92, 0.98	0.95, 0.98	0.92, 1.02	0.87, 0.82	0.78, 0.63	1.00, 1.22
$\theta = -60^\circ$	0.95, 1.00	0.81, 0.95	0.81, 0.95	0.78, 0.95	0.76, 0.90	0.95, 1.35	0.90, 1.15	0.81, 0.65

(ideal case = 1.00, 1.00)

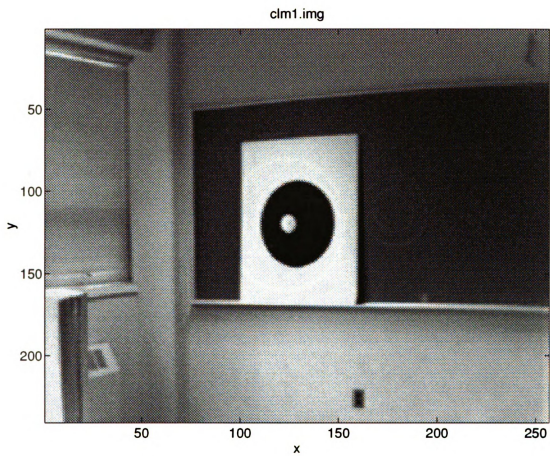


Figure 5.7. A sample landmark image.

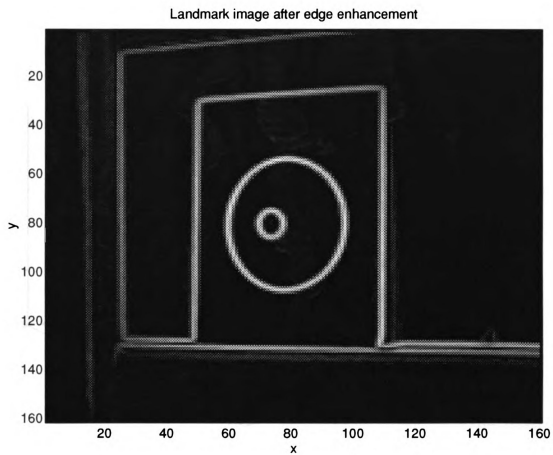


Figure 5.8. The edge enhanced landmark image.

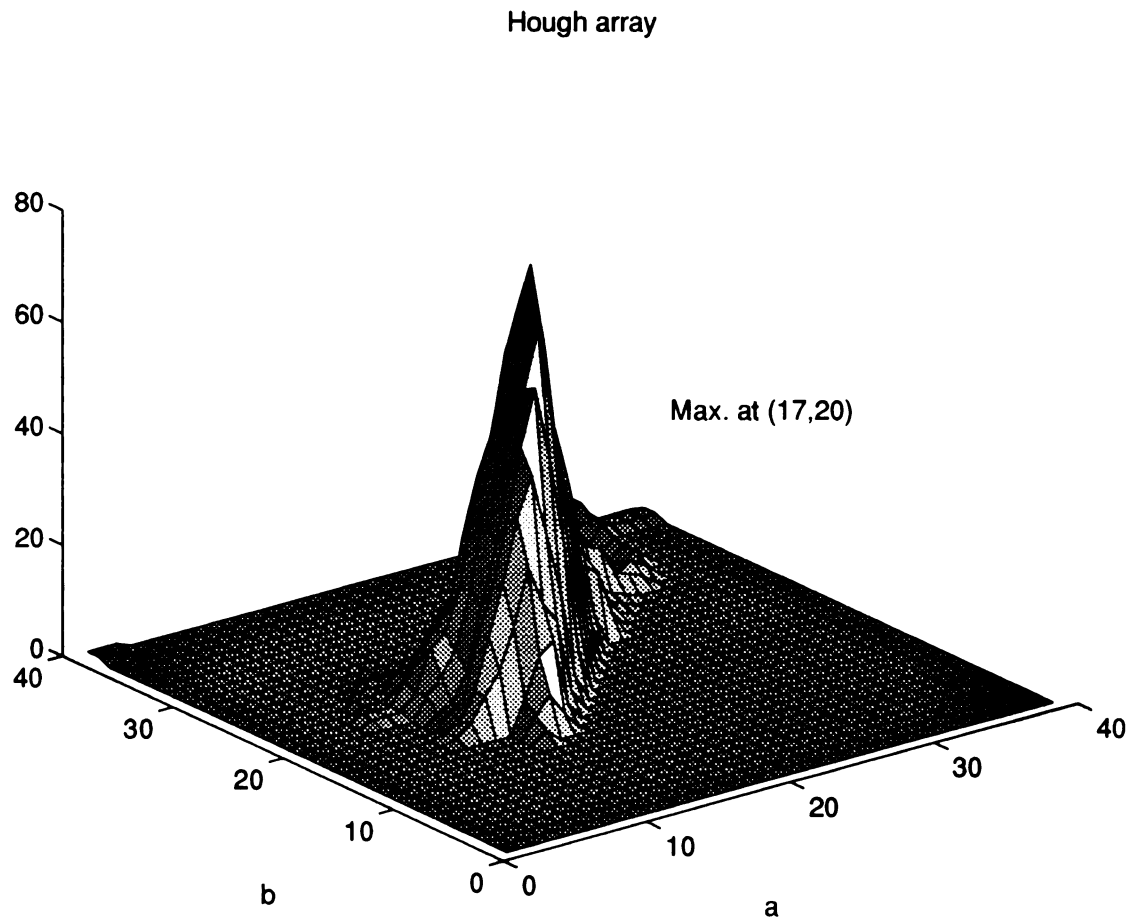


Figure 5.9. The peak value in the corresponding EHT array indicates the dimensions of the ellipse approximating the landmark projection.

5.6 Error Analysis

As mentioned in section 5.5, the projection of a circular disk is not an ellipse in the pinhole model. The only case where the projection is a circle in the image plane is when the circular disk lies in a plane perpendicular to the camera axis (i.e. $\theta = 0$), moreover, the camera axis must pass through the center of the disk. In any other case, the projection appears like an ellipse, but actually is asymmetrical except to the central horizontal axis. Looking at Figure 5.10, it is clear that the two portions of the circular disk separated by its vertical diameter are projected quite differently. In particular, the equivalent geometrical features, such as the lengths of the radii in different directions, produce different projection sizes. This phenomenon is called the perspective distortion in z direction. The z coordinate of the feature being projected with respect to the camera frame determines the size factor of the projection image.

Considering the landmark recognition problem described in previous sections, an elliptical pattern produced by on-line EHT algorithm is used to approximate the projection of a circular disk. The ideal case is that the EHT algorithm generates a matching ellipse whose major and minor axis is same as the height and width of the exact projection, i.e. $h = 2b$ and $w = 2a$ (see Figure 5.10). If this is the case, then we can use (5.2) to compute the distance of the landmark. Also, (5.17) gives an estimation of the aspect angle when the distant of landmark is relatively large. However, according to lab experiments the EHT usually produces ellipse with about $\pm 10\%$ of the actual dimensions of the exact projection.

To understand the effectiveness of this approximation, two questions need to be answered. They are:

- (i) “what is the magnitude of errors in the aspect angle and distance of the landmark caused by the EHT approximation?”

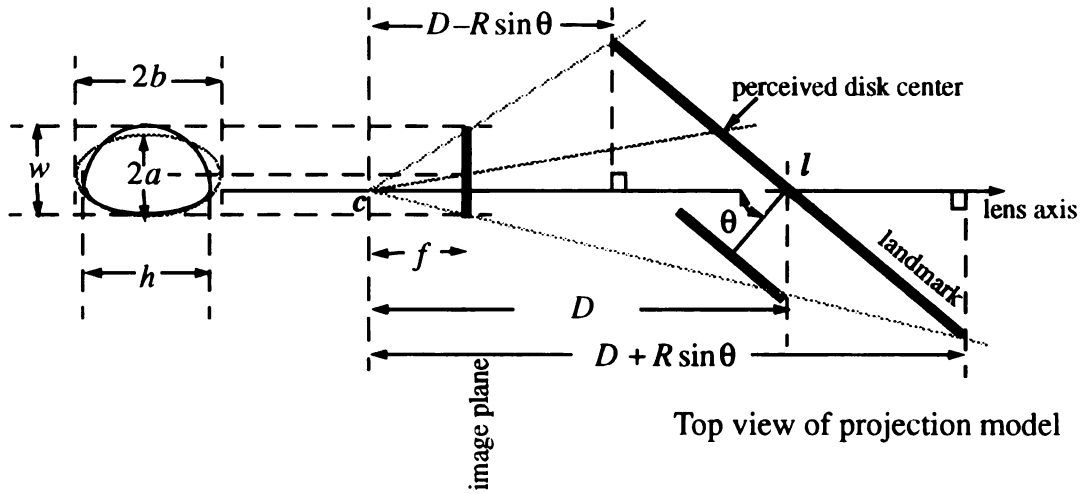


Figure 5.10. The exact projection model of the primary landmark pattern.

- (ii) “what is the magnitude of errors in the horizontal position of perceived disk center?”

To answer both questions, it is necessary to derive the exact model of landmark projection first.

Given the exact projection in the image plane, the width of the projection is

$$w = \frac{2fDR \cos \theta}{D^2 - R^2 \sin^2 \theta} \quad (5.18)$$

Also, the height of the projection can be derived as

$$h = \frac{2fR}{D} \quad (5.19)$$

5.6.1 Errors in Aspect Angle and Distance of Landmark

Substituting $\sin^2 \theta$ in (5.18) with $1 - \cos^2 \theta$, we have

$$\cos^2 \theta - \frac{2fD}{wR} \cos \theta + \left(\frac{D^2}{R^2} - 1 \right) = 0 \quad (5.20)$$

Solve for $\cos \theta$, we have

$$\cos \theta = \frac{1}{2} \left[\frac{2fD}{wR} \pm \sqrt{\left(\frac{2fD}{wR} \right)^2 - 4 \left(\left(\frac{D}{R} \right)^2 - 1 \right)} \right] \quad (5.21)$$

Testing with the condition $\theta = 0$, which corresponds to the special case where $w = 2fR/D$, (5.21) becomes

$$\begin{aligned} 1 &= \frac{1}{2} \left[\left(\frac{D}{R} \right)^2 \pm \sqrt{\left(\frac{D}{R} \right)^4 - 4 \left(\left(\frac{D}{R} \right)^2 - 1 \right)} \right] \\ &= \frac{1}{2R^2} \left[D^2 \pm \sqrt{(D^2 - 2R^2)^2} \right] \end{aligned} \quad (5.22)$$

The errors in the aspect angle of the landmark caused by approximation can be calculated by consider the following two cases:

(i) $D \geq \sqrt{2}R$:

In this case, the exact aspect angle is

$$\theta = \arccos \left(\frac{1}{2} \left[\frac{2fD}{wR} - \sqrt{\left(\frac{2fD}{wR} \right)^2 - 4 \left(\left(\frac{D}{R} \right)^2 - 1 \right)} \right] \right) \quad (5.23)$$

Substituting $D = 2fR/h$ into this equation, we have the solution for θ as a function of w and h

$$\theta(w, h) = \arccos \left[\frac{1}{wh} \left(2f^2 - \sqrt{4f^2(f^2 - w^2) + w^2 h^2} \right) \right] \quad (5.24)$$

The error in aspect angle caused by approximation is

$$\tilde{\theta} = \theta(w, h) - \arccos \left[\frac{1}{2ab} \left(f^2 - \sqrt{f^4 - 4a^2 f^2 + 4a^2 b^2} \right) \right] \quad (5.25)$$

(ii) $D < \sqrt{2}R$:

In this case, the aspect angle is

$$\theta(w, h) = \arccos \left[\frac{1}{wh} \left(2f^2 + \sqrt{4f^2(f^2 - w^2) + w^2 h^2} \right) \right] \quad (5.26)$$

and the error is

$$\tilde{\theta} = \theta(w, h) - \arccos \left[\frac{1}{2ab} \left(f^2 + \sqrt{f^4 - 4a^2 f^2 + 4a^2 b^2} \right) \right] \quad (5.27)$$

The error surface is plotted in Figure 5.11 for $\pm 10\%$ perception errors in both w and h .

For the error in landmark distance D , it follows from (5.19) that

$$D = \frac{2fR}{h} = \frac{fR}{b} \quad (5.28)$$

because $h = 2b$.

The only source of error comes from the EHT, i.e. when the length of the major axis is incorrect. From (5.28), we have

$$dD = -\left(\frac{fR}{b^2}\right) \cdot db \quad (5.29)$$

This gives an estimation of distance error.

5.6.2 Errors in Perceived Disk Center

From the exact projection equation (5.18), we know the offset of true disk center projection from the perceived center in the image plane is

$$c = \frac{fR^2 \sin \theta \cos \theta}{D^2 - R^2 \sin^2 \theta} \quad (5.30)$$

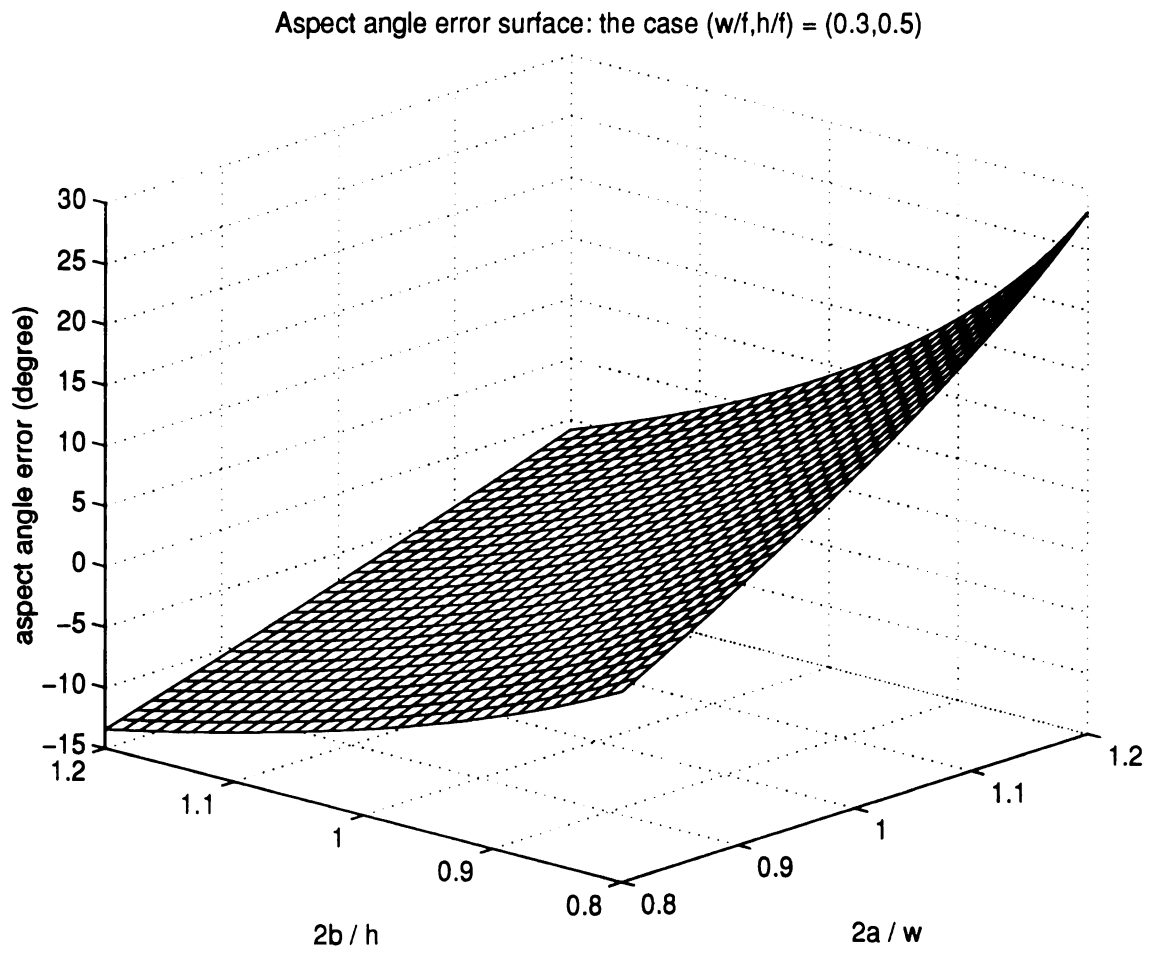


Figure 5.11. The error surface of landmark aspect angle.

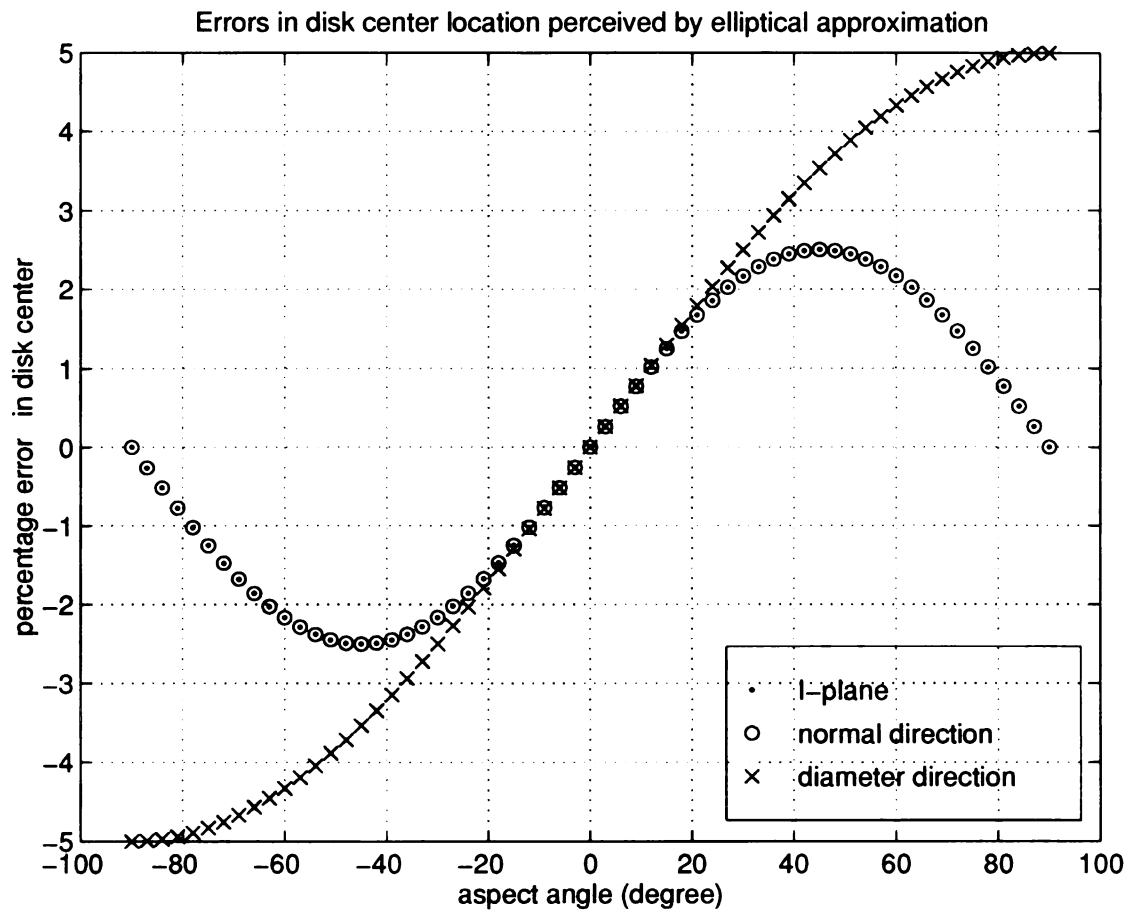


Figure 5.12. Errors in the perceived disk center position.

Using the inverse projection formula, the real world offset distance of the disk center in the normal direction of camera axis is

$$C_{\text{normal}} = \frac{DR^2 \sin \theta \cos \theta}{D^2 - R^2 \sin^2 \theta}, \quad (5.31)$$

or, equivalently, in the horizontal diameter direction

$$C_{\text{l-plane}} = \frac{C_{\text{normal}}}{D \cos \theta + C_{\text{normal}} \sin \theta} \quad (5.32)$$

In Figure 5.12, the errors in (5.30), (5.31), and (5.32) are plotted as functions of the aspect angle θ .

5.7 Tracking the Landmarks Using Visual Servo Control

One of the most challenging problem in mobile robot navigation is the docking motion control. Toward the solution of this problem, a visual servo control algorithm has been developed and implemented on SPARTA.

Due to the holonomic configuration of the mobile base, the motion dynamics of SPARTA can be easily described by two variables, namely the angular speed in the change of the robot heading, which is

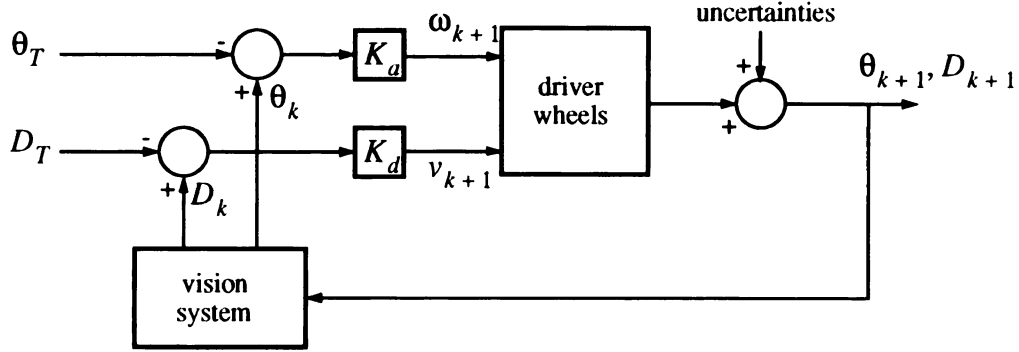
$$\omega = \frac{d\theta}{dt},$$

and the linear speed in the direction of robot heading

$$v = \frac{dD}{dt}.$$

Written in discrete time model, we have

$$\omega_k = \frac{\Delta \theta_k}{\Delta T}, \text{ and } v_k = \frac{\Delta D_k}{\Delta T}$$



$$\omega_{k+1} = K_a (\theta_k - \theta_T)$$

$$v_{k+1} = K_d (D_k - D_T)$$

Figure 5.13. The block diagram of visual servo motion control system.

where ΔT represents the sampling period of system variables. This leads to the following setup of visual servo control:

Given the vision system capable of finding the landmark aspect angle and distance, the visual servo control of SPARTA can be modelled by the discrete-time linear system

$$\frac{1}{\Delta T} \begin{bmatrix} \Delta \theta_{k+1} \\ \Delta D_{k+1} \end{bmatrix} = \begin{bmatrix} \omega_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} K_a & 0 \\ 0 & K_d \end{bmatrix} \cdot \left(\begin{bmatrix} \theta_k \\ D_k \end{bmatrix} - \begin{bmatrix} \theta_T \\ D_T \end{bmatrix} \right) \quad (5.33)$$

where θ and D are the aspect angle and the distance to the center of the landmark, K_a and K_d are some constants, and the vector $\begin{bmatrix} \theta_T & D_T \end{bmatrix}^T$ is the target of the control. The block diagram of this system is shown in Figure 5.13.

In real-world applications, the final approaching stage of a docking motion is one of the

most important part of the entire navigation process. Most frequently, the robot needs to adjust its position and heading such that it directly faces the target before it enters the final approaching range of the docking station. Considering SPARTA's model, this is actually the case where $\theta_T = 0$ in (5.33) if we assume that the landmark pattern has been set up on the docking port. In fact, during the phase of recognizing landmark projection patterns, $\theta_T = 0$ corresponds to the special case where the projection of the primary disk is also a circular disk in the image plane. This implies the possibility of avoiding the use of EHT because in this case the projection pattern can be effectively approximated by a circle. Instead, the ordinary Hough transformation can be used without losing track of the landmark. This causes significant improvement in recognition speed because the Hough array is 2-dimensional instead of 3-dimensional in the case of EHT. In fact, according to experiments the improvement is so critical that landmark tracking can be done at an average rate of 3.2 frames per second. Beside the change in the dimension of search space, a tracking window is also defined to limit the domain of Hough transformation. This is based on the assumption of continuity in the changes of the circular projection pattern, namely the center and the radius of the circle. The use of tracking window also reduces the possibility of "jumping" away from original pattern to any other false pattern which might generate better response in certain cases.

The experimental results of a sample visual-servo docking process are presented in Figure 5.14 - Figure 5.15.

5.8 Selecting Landmark Positions

For calibration purposes mentioned in the previous section, the positions of landmarks in the workspace need to be determined in advance. This process is governed by the following rules:

- (Rule 1) The total number of landmarks should be as small as possible.

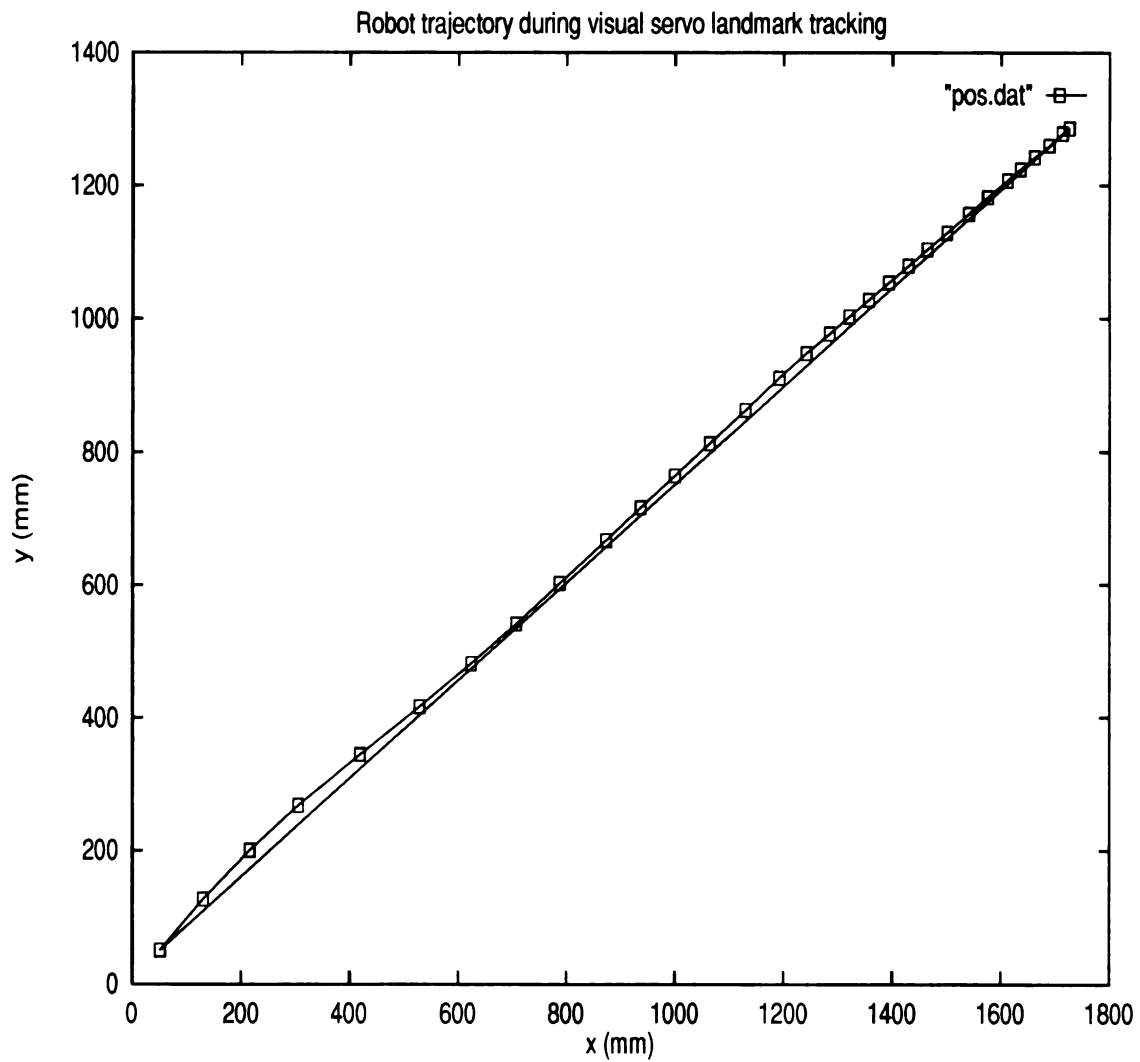


Figure 5.14. Robot trajectory during visual-servo tracking of a landmark (initial heading error = 8 degrees).

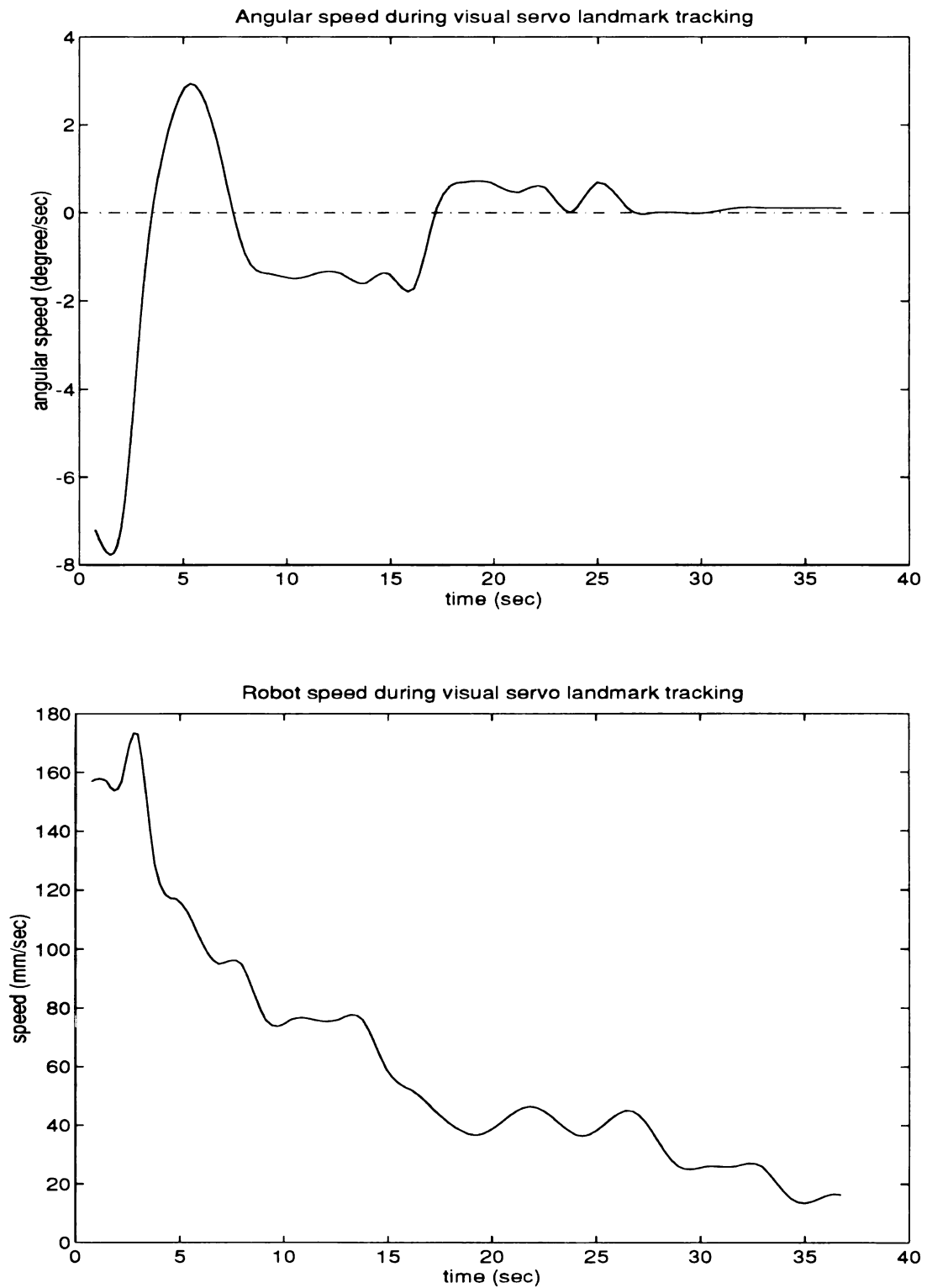


Figure 5.15. The robot's angular and linear speed profile during landmark tracking.

(Rule 2) At least one other landmark should be visible from any landmark.

(Rule 3) Landmarks should be as close to the points of interest as possible.

The reason for Rule 1 is self-explanatory. Rule 2 is required for continuous path tracking for the case of longer path which consists of more than one calibration point. Rule 3 is important to the docking motion when the mobile robot arrives at each point of interest.

With the inspiration from graph theory, a simplistic solution to the landmark selection problem is developed. In general, the positions of landmarks can be determined by the following steps:

(step 1) Construct a set V , which includes

- (i) all of the points of interest in the workspace,
- (ii) all convex vertices in the workspace.

$$\text{i.e. } V = \{v \in W \mid v \text{ is point of interest} \vee v \text{ is convex vertex}\}$$

(step 2) Using V as the vertex set, generate a visibility graph based on the polygonal representation of the workspace, call this graph G .

(step 3) Find the minimum order subgraph of G which

- (i) includes all of the point of interest in its vertex set, and
- (ii) is connected

call this subgraph $G' = (V', E')$.

(step 4) In G' , find the minimum vertex cut set V_{cut} . Add V_{cut} to landmark list, then remove V_{cut} from V' . This produces k subgraphs ($k \geq 1$), call them $G''_i, i = 1, \dots, k$. For each of these subgraphs, repeat (step 4). This continues recursively until every subgraph reduces to a single vertex.

(step 5) Set up a landmark at each vertex in the landmark list. The information associated with each landmark includes

- (i) the ID's of all its neighbor landmarks
- (ii) the distances to its neighbor landmarks, sorted from the one which is nearest to the one most far away
- (iii) the orientation to each neighbor landmark

In fact, the problem of choosing landmark positions is difficult, however, it can be done off-line and only needs to be done for once. Figure 5.16 shows an example of landmark arrangement in a typical indoor environment.

5.9 Vision-Based Navigation Planning

Given the discussions in section 5.5 and section , it is possible to integrate the vision-based navigation tools, i.e. the self-calibration and landmark tracking, to perform motion control with high accuracy.

Suppose the robot needs to move from its initial position to a goal and perform docking motion at the goal configuration. Also, assuming a landmark is used to indicate the goal position and the home-in target for docking motion, then we can design the navigation plan as follows:

- (i) Scan for target landmark, if possible. Use the landmark to calibrate initial heading.
- (ii) Move toward the goal using dead-reckoning, until the target zone is entered.
- (iii) Perform self-calibration to acquire the landmark aspect angle and distance. Move to the landmark's center axis and align the heading with that axis so the robot faces the target of docking.
- (iv) Use visual-servo motion control to approach the docking port until the proximity sensors report the commencement of docking.

This procedure is illustrated in Figure 5.17 with an example. At the initial position, the robot performs self-calibration to compute the value of θ and D . The next step is to move to the position suitable for visual-servo motion. To do so, the robot rotates $\phi = \pi/2 - \theta$ and moves a distance of $D \sin \theta$, and rotates $-\pi/2$ to face the landmark. Starting from this new position, the robot can use visual-servo control for the final approaching stage of the docking motion.

5.10 Summary

This chapter focused on vision-based mobile robot navigation. The theory of projection, as well as the complete formulation of corresponding geometry, are developed as the basis of subsequent discussion. The usage of landmark patterns to calibrate robot position/heading was described in detail, including the pattern design, recognition strategy, corresponding geometries, and calibration procedure. The landmark used by SPARTA consists of two high contrast circular disks, one for position estimation, and the other for identifying the sign of aspect angle. A modified elliptical Hough transformation was developed for detecting and measuring the projection image of circular disks. This

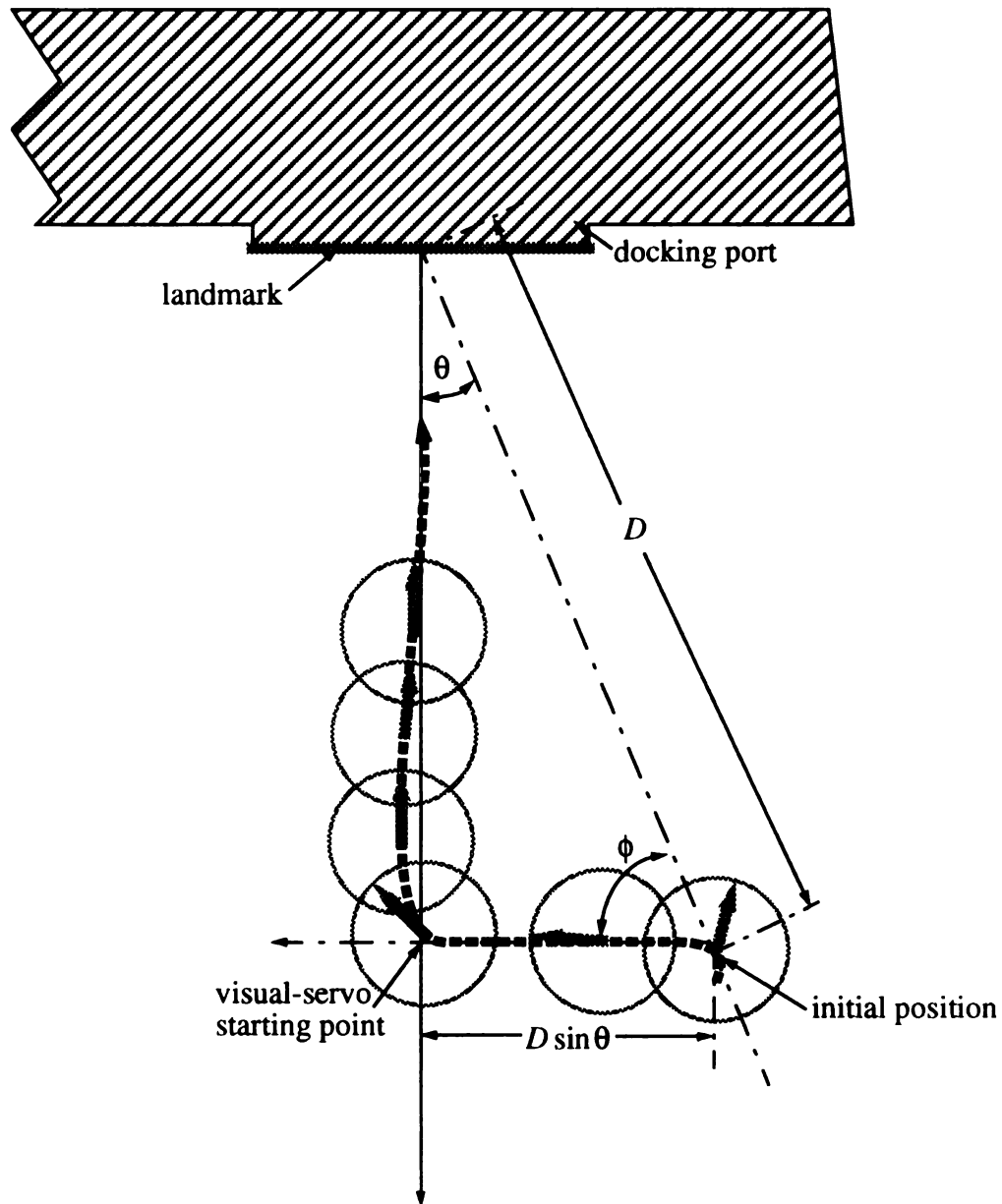


Figure 5.17. Vision-based navigation: a docking motion example.

approach provides robust pattern detection under the situations of noisy or incomplete images. It has been pointed out that, in fact, the projection of a circular disk is not an ellipse in the image plane. For that reason, error analysis was performed to provide an understanding of the effects of elliptical approximation.

In the second part of this chapter, we presented the visual servo control system which had been used for tracking landmarks. The control system is identified and described by systems of difference equations. Under the assumption of small aspect angle, this approach demonstrated encouraging results in servo tracking of circular patterns.

A heuristic algorithm was proposed for choosing the locations of landmarks in a structured building environment.

At the end, we presented vision-based navigation planning which integrates the results of self-calibration and visual-servo control. This approach allows the robot to: (i) start from any position close to the landmark, (ii) adjust its position and heading such that it faces the landmark directly, and (iii) home-in the target with visual-servo motion control. A docking problem is presented as an example to demonstrate this approach.

CHAPTER 6

Systems Integration

In previous chapters we have developed the concepts of:

- on-line navigation planning with sensor-based map construction
- adaptive navigation strategy based on real-time ODF measuring
- vision-based navigation planning

This chapter will address the problem of combining these technologies into a general purpose mobile robot navigation system for the environments which are, dynamic, structured, partially known, and consisting of moving obstacles.

6.1 High Level Navigational Constraints

The major concerns of our robot navigation system are speed and accuracy. As mentioned in chapter 4, the safety constraints can actually be treated as the subset of accuracy constraints. These issues, from high level control point of view, translate into a minimum set of rules in human reasoning:

(Rule 1) If there exists no sign of unsafe situation, then move at top speed.

(Rule 2) If accuracy is critical, then construct better map at the cost of lower speed.

(Rule 3) If the goal is close, then start the vision-based navigation.

The first two rules can be implemented by using ODF as the decision criteria. This has been extensively discussed in chapter 4. The last rule corresponds to the final approach of the robot towards the goal. These rules suggest a multi-stage hybrid navigation plan.

6.2 The 3-Stage Hybrid Navigation Plan

To integrate different navigation strategies, we propose a new hybrid approach with the setup of navigation landmarks to indicate navigation goals. In the robot's workspace, each path consists of the starting position, final goal position, and a series of intermediate goal positions. Each of these goal positions is indicated by a landmark and a target zone around it. The selection of the positions of intermediate goals and their corresponding landmarks has been discussed in section 5.8. The maximal size of the target zone is determined such that any point inside that zone can be used as a calibration point with respect to the landmark associated with the goal. In general, a circular region centered at the landmark can be used as the target zone of the corresponding goal position. This will be further discussed later in this chapter.

The hybrid navigation plan consists of the following stages:

Stage 1: *Initialization*

The robot receives a navigation task, specifying the position of next goal and the path to take. Usually the size of navigation task should be designed to allow the robot to take a straight path to move from its current position to the next intermediate goal without worrying about any turns. Therefore, the robot can simply rotate to face the goal position directly.

Stage 2: *Dead-reckoning navigation*

The robot moves toward the goal in using dead-reckoning approach. Proximity sensors are used for on-line obstacle avoidance and ODF measuring. The speed and sensor configuration of the robot are adjusted dynamically according to the changes in ODF value. This process continues until the robot's estimated position coincide with the goal position, which should be inside the circular boundary of the target zone.

Stage 3: *Vision-based final approach*

Once the robot “thinks” it has entered the target zone, then the vision system is used for recognizing and locating the landmark associated with the goal. The accumulated position/heading errors should be cleared by performing self-calibration. After that, the robot follows the standard visual-servo procedure to complete the final approach.

This navigation plan is based on the assumption that the robot will fall inside the target zone after Stage 2, despite of the accumulation of errors in position and heading. Under normal situation, the robot's real world position can be modeled by the uncertainty ellipse suggested by Leonard and Durrant-Whyte [38]. That means starting from the initial position, the robot's real world position is bounded by an ellipse, as shown in Figure 6.1. This model can be used to determine the minimal size of the target zone associated with any goal position in the workspace, given the distance the robot has to travel to reach that goal. The hybrid navigation plan has been successfully tested on SPARTA.

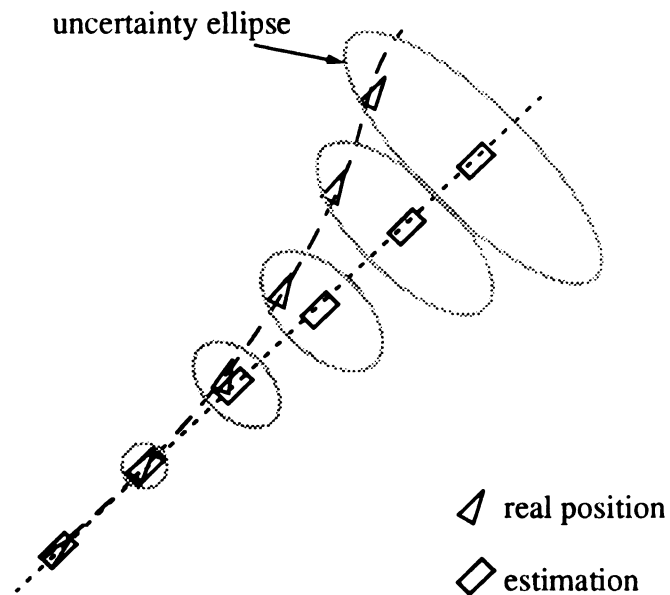


Figure 6.1. The position uncertainty grows as the robot moves in dead-reckoning.

6.3 System Architecture

The basic components of our mobile robot system includes:

- robot motion controller
- position/heading estimator
- ultrasonic ranging system
- infrared detector array
- proximity sensor controller
- vision system
- central robot controller/main computer

The central robot controller is in charge of the integration/interpretation of all sensor data. It is also responsible for making navigational decisions based on the latest version of the map constructed from sensor feedback. The measuring of ODF, which is a routine task for the robot, provide the controller with information about the clutter of the environment in real-time. Based on that information, the robot can adjust the sensor configuration through the sensor control module. Separated from the ultrasonic and infrared sensors, the vision system of SPARTA is activated only when necessary by the robot controller. The block diagram of SPARTA mobile robot system is shown in Figure 6.2.

6.4 Summary

This chapter described a hybrid navigation system for the integration of different navigation strategies into a general purpose mobile robot navigation system. Assuming the robot's workspace is structured as described in Chapter 5, the navigation tasks usually take the form of "moving from point A , which is indicated by landmark L_A , to point B , which is indicated by landmark L_B ."

In the hybrid navigation approach, the robot starts from its current position with initial heading pointing at the goal position. Only dead-reckoning method based on the robot's wheel positions is used in this stage. As the robot moves toward its goal, the uncertainty in the robot's position and heading accumulates until the robot 'thinks' that it has reached a valid calibration point, which should be inside the target zone. The robot then perform vision-based self-calibration to find its relative position with respect to the goal landmark. Based on that information, the robot can complete the final approach of the goal by visual-servo motion control.

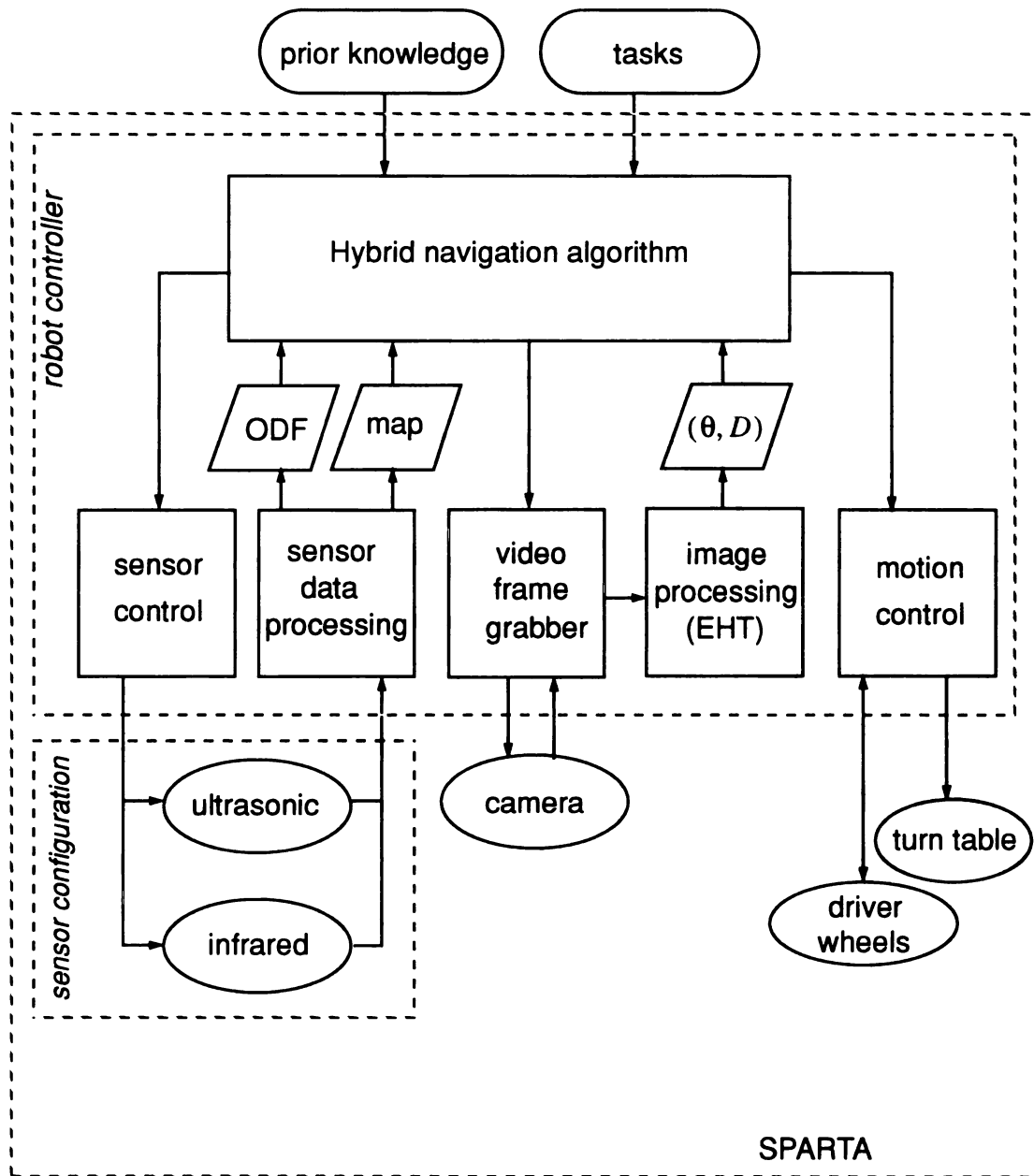


Figure 6.2. The block diagram of SPARTA mobile robot system.

CHAPTER 7

Summary and Conclusions

Autonomous mobile robot navigation represents a new generation of machine intelligence which, unlike the traditional fixed-path robot programming, combines sensor-based environmental perception, decision theories, and robot motion planning. A high degree of flexibility is provided through the concept of on-line sensing-decision navigation model. The major topics in mobile robot research include: sensing technologies, sensor-based map construction, path planning, real-time obstacle avoidance, and self-calibration. The first half of this dissertation dealt with the topics related to sensor-based mobile robot navigation in dynamic environments with unknown moving obstacles. The second half has focused on the vision-based navigational applications including landmark-based self-positioning and visual-servo control for landmark tracking.

7.1 Sensor-Based Robot Navigation

The characteristics of two types of proximity sensors, namely the ultrasonic range finders and infrared detectors, are discussed in detail. Moreover, the probability models used to interpret the output data generated by each type of sensor are presented. Based on these models, an algorithm is developed to integrate the results into a single, uniform representation of the occupancy information in the region covered by both types of sensors. This representation is called the occupancy grid of the environment. A binary

map can be constructed by applying a suitable threshold value to the occupancy grid.

7.1.1 The SID Cycle Model

Due to the dynamic nature of the robot's workspace, it is critical to refresh, or update, the navigation map constantly. Together with the motion planning phase, we have an iterative process which is defined as the *sensing-integration-decision* (SID) cycle. This actually provides an estimation of the shortest response time a robot has against the changes in its environment. As an immediate result of this physical constraint, the maximal speed of the robot is bounded for the safety/accuracy criteria.

The length of SID cycle is not fixed since most robots have multiple types of sensors which can be controlled independently. This leads to the concept of an *adaptive sensing strategy*, which means choosing different combination of sensors according to the needs of current navigation task. If more sensory feedback is necessary, then the robot can activate more sensors at the cost of a longer SID cycle, which also means lower speed limit. It is pointed out that the safety/accuracy constraints and the maximum motion speed are competing factors in robot navigation systems.

7.1.2 Sensor Configuration and Obstacle Density Function

Multi-level sensor configurations play the key role in the robot's capability to adapt to dynamic changes. It provides tremendous flexibility by allowing the robot to decide how much sensory information to take, and most importantly, what features to extract from the sensor data.

The safety/accuracy constraints are only the static part of navigational requirements. Observations in human responses to environmental changes also suggest the feasibility of incorporating the degree of workspace clutter as a factor in the robot's decision making. To do so, we defined the *obstacle density function* (ODF) for the on-line measuring of obstacle clutter in the region around the robot. Based on ODF, it is possible to adjust the

sensor configuration in reaction to the dynamic changes. In this way, the trade-off between the amount of sensor data and the motion speed becomes useful tool in optimizing the overall system performance. This new approach is called the *adaptive navigation strategy*. Several examples have been presented in this dissertation to address the significance of this new approach.

7.2 Navigational Errors

Errors in navigation processes come from uncertainties associated with: (i) robot position/heading estimation, and (ii) environmental perception. The first source refers to the error accumulation due to the nature of the dead-reckoning approach. Since our robot only has wheel encoders as a tool of position/heading estimation, errors accumulate at an unacceptable rate. For that reason, we must design a self-calibration system to clear these errors periodically. The second source of errors is the result of incorrect sensor data and/or poor interpretation of sensor data. Sensing errors can only be reduced by redundant sensing and sophisticated sensor integration algorithms. A *match index* based on SSD error model is defined for evaluating the accuracy of the map constructed from sensor data.

7.3 Vision

The vision system of SPARTA has been used for: (i) recovering the errors in its position and heading, and (ii) visual-servo control for final approach and/or docking motions. Both types of tasks require the use of landmarks.

7.3.1 Landmark-Based Self-Calibration

The theory of 3-D projection was elaborated in the second half of this dissertation. A pair of high contrast circular disks of 10 and 50 cm diameters were used as the patterns of each landmark. Also, a white strip is painted on the vertical central axis of the larger disk to

indicate the horizontal location of the disk center. Since the projection of a circular disk in the image plane can be effectively approximated by an ellipse, we can use the *elliptical Hough transformation* (EHT) to extract the parameters of that projection image. Using these parameters, we can compute the distance and aspect angle of the landmark. Finally, the robot's position and heading with respect to the world frame are recovered.

7.3.2 Landmark Tracking Using Visual-Servo Control

A visual-servo control system was developed on SPARTA to perform accuracy-demanding tasks such as docking motion. Unlike the self-calibration task, the visual-servo can only be established when the aspect angle of the landmark is small. In this case, the projection of a circular disk is also a circular disk in the image plane. Therefore, the regular Hough transformation can be used instead of the EHT, which means the dimension of parameter space is reduced by one. Meanwhile, a tracking window was used in the image plane to further reduce the amount of computation required. The use of tracking windows also reduced the chance of locking onto a wrong target if there existed any other high contrast pattern which resemble a legitimate landmark pattern. Using these techniques, a much higher frame rate can be achieved. Experiments on SPARTA have shown excellent results in visual-servo motion.

The results of self-calibration/positioning and visual-servo control are further integrated to provide a powerful tool of vision-based navigation. Given the robot's initial position inside the perceivable range of a landmark, the robot is capable of recovering its position and heading, adjusting its position and heading to face the landmark, and finally, starting the visual-servo control to home-in the target. This allows us to decompose each navigation task of the form "move from point *A* to point *B*" into three stages, namely: initialization, dead-reckoning navigation with sensor-based on-line obstacle avoidance, and finally, the vision-based navigation for docking. This approach provides a feasible solution to the robot position/heading uncertainty problem caused by dead-reckoning

navigation.

7.4 Systems Integration – The Hybrid Navigation Strategy

A hybrid navigation system was proposed for the environments that are dynamic, structured, partially known, and consisting of moving obstacles. This system combines the results of sensor-based adaptive navigation strategy and the vision-based navigation planning. Based on the uncertainty ellipse model for robot position estimation, we assume the robot's real world position is bounded by an ellipse which grows in size as the robot moves. Thus, we can assign a target zone around the goal and plan the path that will assure the robot's position will fall inside that region. Once the robot enters the target zone, it starts the vision-based navigation to perform the following: (i) calibrating its position and heading, (ii) adjusting its position and heading to face the goal landmark, and (iii) using visual-servo control to complete the final approach.

The system block diagram of SPARTA was presented and analyzed.

7.5 Conclusions

Autonomous mobile robot navigation depends on the use of real-time sensor feedback to perceive the changes in the robot's environment and adapt to those changes. These environmental changes can be translated into various navigation constraints, by adjusting sensor configuration and the motion speed. This allows the robot to deal with dynamic changes, which are often "unexpected", in its environment.

For mobile robots equipped with multiple types of sensors, the multi-sensor integration technique provides a feasible way of automatic map construction. However, the high computation cost associated with this approach has prevented it from realtime implementation, especially when a vision system is involved.

Inspired by human reaction, we developed a new approach to control the cost of sensor data integration by monitoring the clutter of the environment in close range and modifying

the robot's sensor configuration. This adaptive sensing strategies exploits the flexibility of the sensing mechanism, in particular, for mobile robots equipped with multiple types of sensors. By measuring the changes of clutter in the environment, the robot decides how much, and which types of sensor data is needed. Based on that information, the length of SID cycle can be estimated and then translated into the robot's speed constraint.

Despite the relatively higher computation cost, vision systems provide a reliable way of recognizing certain geometrical features in the environment. In our experiments, a vision system has been used to calibrate the robot's position and heading by finding its relative position/orientation with respect to fixed landmarks. In addition, the feature tracking capability of the vision setup allows the mobile robot to perform visual-servo motion control. This capability is extremely important to perform docking motion, which is usually the final approaching stage of a navigation process.

7.6 Recommendations for Future Research

The work presented in this dissertation have focused on sensor-based navigation with human-like environmental reactions. However, there is still much to be done before a higher degree of autonomy can be achieved on mobile robots. The following observations can serve as suggestions for future research in this area:

- The landmark system proposed in this work can be improved for more general purpose robotics applications. A new algorithm for selecting landmark positions also needs to be developed for the robustness of navigation processes.
- Use the Match Index to test different map construction algorithms or sensor arrangements. Use the test results to find the best sensing strategy for the robot.
- Improve the Elliptical Hough Transformation to give better estimation of the dimensions of the ellipses approximating the landmark projection patterns.

- New sensing technologies can be used to create new dimensions in robot navigation strategy. For example, the use of stereo vision started a new direction in the area of robot navigation. Also, the rearrangement of existing sensors may result in better modeling of the robot's environment. For instance, the ranging information provided by an ultrasonic sensor only tells us the possibility of existence of a certain obstacle at that range. However, the exact dimensions and location of the obstacle are not clear due to the wide-angle nature of the ultrasonic sensor. This problem might be solved by allowing an infrared detector to sweep across the 12° envelope of an ultrasonic sensor in order to extract the boundaries of the obstacle.
- Visual sensing has become more important in mobile robot navigation. In particular, the recognition and interpretation of basic geometrical features can be effectively used to navigate a mobile robot in structured environments. Currently, we have been using a vision system for special purposes such as landmark finding and visual tracking of certain geometrical features. However, more understanding in visual perception and the corresponding robotics applications is still necessary.
- Multi-robot environments need to be studied for more complex robot operations. This includes the possibility of mounting robot arms on the mobile robot platform.
- The interface between mobile robots and humans should be emphasized as the purpose of developing intelligent machines is to improve the quality of human lives. Technologies such as voice recognition and special command input devices can be incorporated into the robot controller.

BIBLIOGRAPHY

- [1] M. Abidi and T. Chandra. "Pose Estimation for Camera Calibration and Landmark Tracking," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [2] M. D. Adams, Huosheng Hu, and P. J. Probert. "Towards a Real-Time Architecture for Obstacle Avoidance and Path Planning in Mobile Robots," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [3] R. S. Ahluwalia and E. Y. Hsu. "Sensor-Based Obstruction Avoidance Technique for a Mobile Robot," *Journal of Robotic Systems*, Vol. 1, No. 4, 1984.
- [4] Robert S. Alexander and Neil C. Rowe. "Path Planning by Optimal-Path-Map Construction for Homogeneous-Cost Two-Dimensional Regions," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [5] P. K. Allen, A. Timcenko, B. Yoshimi, and Paul Michelman, "Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 2, April 1993.
- [6] Ronald Arkin and Robin R. Murphy. "Autonomous Navigation in a Manufacturing Environment," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 4, August 1990.
- [7] Ronald Arkin and Douglas MacKenzie, "Temporal Coordination of Perceptual Algorithms for Mobile Robot Navigation," *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 3, June 1994.
- [8] Sami Atiya and Gregory D. Hager, "Real-Time Vision-Based Robot Localization," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, December 1993.
- [9] Nicholas Ayache, *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*, The MIT Press, 1991.
- [10] Dana H. Ballard and Christopher M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 1982.
- [11] Martin Beckerman. "Treatment of Systematic Errors in the Processing of Wide-Angle Sonar Data for Robotic Navigation," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 2, April 1990.

- [12] Johann Borenstein and Yoram Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [13] Johann Borenstein and Yoram Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, August 1991.
- [14] David J. Braunegg, "MARVEL: A System that Recognizes World Locations with Stereo Vision," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 3, June 1993.
- [15] Andres Castano and Seth Hutchinson, "Visual Compliance: Task-Directed Visual Servo Control," *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 3, June 1994.
- [16] Ernst Dieter Dickmanns, "Object Recognition and Real-Time Relative State Estimation Under Egomotion," *Real-Time Object Measurement and Classification*, Springer-Verlag, 1988.
- [17] Kevin J. Dowling. "Sensing and Guidance Technologies for Autonomous Vehicles," *Japan U.S.A. Symposium on Flexible Automation*.
- [18] Keith C. Drake, E. S. McVey, and R. M. Inigo, "Sensing Error for a Mobile Robot Using Line Navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, No. 4, July, 1985.
- [19] Keith C. Drake, Eugene S. McVey, Rafael M. Inigo, "Experimental Position and Ranging Results for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 1, February 1987.
- [20] H. F. Durrant-Whyte, B. Y. S. Rao, and H. Hu, "Toward a Fully Decentralized Architecture for Multi-Sensor Data Fusion," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [21] Alberto Elfes. "Sonar-Based Real-World Mapping and Navigation," *IEEE Journal Of Robotics and Automation*, Vol. RA-3, No. 3, June 1987.
- [22] Bernard Espiau, Francois Chaumette, and Patrick Rives, "A New Approach to Visual Servoing in Robotics," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992.
- [23] A. M. Flynn, "Combining Sonar and Infrared Sensors for Mobile Robot Navigation," *International Journal of Robotics Research*, Vol. 7, No. 6, December 1988.
- [24] Kun-Chee H. Fok and Mansur R. Kabuka, "A Flexible Multiple Mobile Robots System," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 5, October 1992.

- [25] J. K. Hackett and M. Shah, "Multi-Sensor Fusion: A Perspective," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [26] Tony Hague, Mike Brady, and Stephen Cameron. "Using Moments to Plan Paths for the Oxford AGV," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [27] Robert M. Haralick, "Determining Camera Parameters from the Perspective Projection of a Rectangle," *Journal of Pattern Recognition*, Vol. 22, No. 3, 1989.
- [28] Scott Y. Harmon, "The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, June 1987.
- [29] Basit Hussain and Mansur R. Kabuka, "Real-Time System for Accurate Three-Dimensional Position Determination and Verification," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 1, February 1990.
- [30] Hiroshi Ishiguro, Patrick Stelmazyk, and Saburo Tsuji, "Acquiring 3-D Structure by Controlling Visual Attention of a Mobile Robot," *Proc. IEEE Int. Conf. Robotics and Automation*, 1990.
- [31] S. Sitharama Iyengar and Rangasami L. Kashyap, "Autonomous Intelligent Machines," *IEEE Computer Magazine*, June 1989.
- [32] P. Kielczynski, W. Pajewski, and M. Szalewski, "Ultrasonic Transducers with Bessel Function Distribution of vibrational Amplitude on their Surfaces," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, December 1993.
- [33] Mansur R. Kabuka, Alvaro E. Arenas, "Position Verification of a Mobile Robot Using Standard Pattern," *IEEE Journal Of Robotics and Automation*, Vol. RA-3, No. 6, December 1987.
- [34] Jin-Oh Kim and Pradeep K. Khosla, "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992.
- [35] Richard D. Klafter, "Mobile Robots, Research and Development," *International Encyclopedia of Robotics*, Wiley, 1988.
- [36] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [37] Xavier Lebegue and J. K. Aggarwal, "Significant Line Segments for an Indoor Mobile Robot," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, December 1993.
- [38] John J. Leonard and Hugh F. Durrant-Whyte, "Mobile Robot Localization by Tracking Geometric," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991.

- [39] Cheng-Chih Lin and R. L. Tummala, "Mobile Robot Navigation Using Sensory Feedback," *Proc. IEEE Int'l Conf. Systems Engineering*, Dayton, OH, August, 1991.
- [40] Cheng-Chih Lin and R. L. Tummala, "Adaptive Sensor Integration for Mobile Robot Navigation," *Proc. IEEE Conf. Multisensor Fusion and Intelligent Systems*, Las Vegas, NV, October 1994.
- [41] Cheng-Chih Lin and R. L. Tummala, "Adaptive Sensor Integration for Mobile Robots in Partially Known Environment," *Proc. ISCA Int'l Conf. on Computers and Their Applications*, Long Beach, CA, March 1994.
- [42] L. H. Matthies, A. E. Elfes, "Sensor Integration for Robot Navigation: Combining Sonar and Stereo Range Data in a Grid-Based Representation," *Proc. 26th IEEE Decision and Control Conference*, Los Angeles, December 1987.
- [43] E. S. Mc Vey, K. C. Drake, and R. M. Inigo, "Range Measurements by a Mobile Robot Using a Navigation Line," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, January, 1986.
- [44] Hans P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *Annual Research Review of the Robotics Institute*, 1987.
- [45] Lars Nielsen, "Automated Guidance of Vehicles using Vision and Projective Invariant Marking," *Automatica*, Vol. 24, No. 2, pp. 135-148, 1988.
- [46] K. Onoguchi, M. Watanabe, Y. Okamoto, Y. Kuno, and H. Asada, "A Visual Navigation System Using a Multi-Information Local Map," *Proc. Int. Conf. Robotics and Automation*, 1990.
- [47] Tai-Jee Pan and Ren C. Luo. "Motion Planning for Mobile Robots in a Dynamic Environment," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [48] Nikolaos P. Papanikolopoulos, Pradeep K. Khosla, and Takeo Kanade, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision," *IEEE Transactions on Robotics and Automation*, Vol. 9 No. 1, February 1993.
- [49] R. Safae-Rad, I. Tchoukanov, K. C. Smith, and B. Benhabib, "Three-Dimensional Location Estimation of Circular Features for Machine Vision," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 5, October 1992.
- [50] A. Sankaranarayanan and M. Vidyasagar. "A New Path Planning Algorithm for Moving a Point Object Amidst Unknown Obstacles in a Plane," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [51] Rolf Schuster, Nirwan Ansari, and Ali Bani-Hashemi, "Steering a Robot with Vanishing Points," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, August 1993.

- [52] K. Storjohann, Th. Zielke, H. A. Mallot, and W. von Seelen, "Visual Obstacle Detection for Automatically Guided Vehicles," *Proc. Int. Conf. Robotics and Automation*, 1990.
- [53] Robert B. Tilove, "Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials," *Proc. IEEE Conf. Robotics and Automation*, Cincinnati, OH, 1990.
- [54] Stephen J. Walsh, *Indoor Robot Navigation Using a Symbolic Landmark Map*, Ph. D. Dissertation, Michigan State University, 1992.
- [55] Juyang Weng, Paul Cohen, and Nicolas Rebibo, "Motion and Structure Estimation from Stereo Image Sequences," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992.
- [56] Qiuming Zhu. "Hidden Markov Model for Dynamic Obstacle Avoidance for Mobile Robot Navigation," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991.