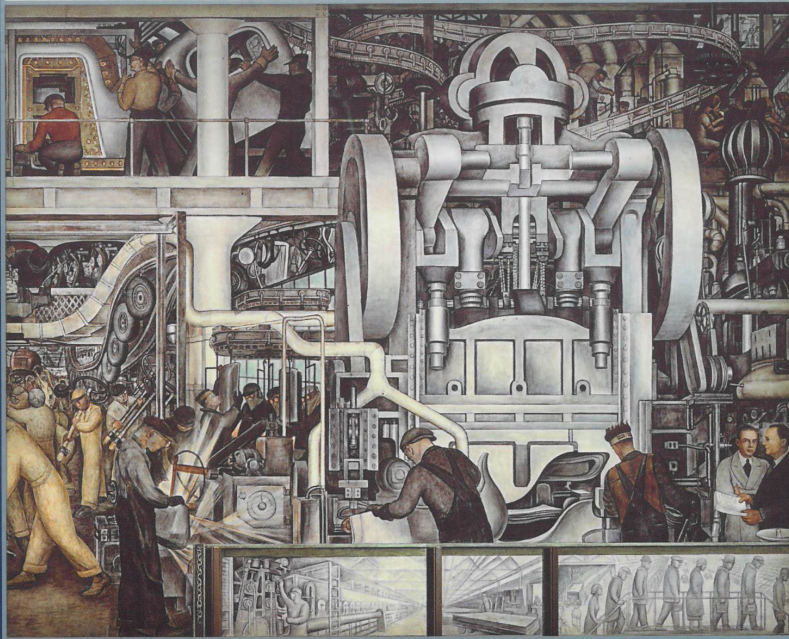


INTRODUCTION TO AI ROBOTICS



ROBIN R. MURPHY

Introduction to AI Robotics

Robin R. Murphy

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

© 2000 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

Typeset in 10/13 Lucida Bright by the author using $\text{\LaTeX}$ 2 ϵ .
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Murphy, Robin, 1957–

Introduction to AI robotics / Robin R. Murphy.

p. cm.—(Intelligent robotics and autonomous agents. A Bradford Book.)

Includes bibliographical references and index.

ISBN 0-262-13383-0 (hc. : alk. paper)

1. Robotics. 2. Artificial intelligence. I. Title. II. Series

TJ211.M865 2000

629.8'6263—dc21

00-033251

Brief Contents

I Robotic Paradigms 1

- 1 *From Teleoperation To Autonomy* 13
- 2 *The Hierarchical Paradigm* 41
- 3 *Biological Foundations of the Reactive Paradigm* 67
- 4 *The Reactive Paradigm* 105
- 5 *Designing a Reactive Implementation* 155
- 6 *Common Sensing Techniques for Reactive Robots* 195
- 7 *The Hybrid Deliberative/Reactive Paradigm* 257
- 8 *Multi-agents* 293

II Navigation 315

- 9 *Topological Path Planning* 325
- 10 *Metric Path Planning* 351
- 11 *Localization and Map Making* 375
- 12 *On the Horizon* 435

Contents

Preface xvii

I Robotic Paradigms 1

1	<i>From Teleoperation To Autonomy</i>	13
1.1	Overview	13
1.2	How Can a Machine Be Intelligent?	15
1.3	What Can Robots Be Used For?	16
1.3.1	Social implications of robotics	18
1.4	A Brief History of Robotics	19
1.4.1	Industrial manipulators	21
1.4.2	Space robotics and the AI approach	26
1.5	Teleoperation	28
1.5.1	Telepresence	32
1.5.2	Semi-autonomous control	33
1.6	The Seven Areas of AI	34
1.7	Summary	37
1.8	Exercises	37
1.9	End Notes	39

2	<i>The Hierarchical Paradigm</i>	41
2.1	Overview	41
2.2	Attributes of the Hierarchical Paradigm	42
2.2.1	Strips	44
2.2.2	More realistic Strips example	46
2.2.3	Strips summary	52

2.3	Closed World Assumption and the Frame Problem	53
2.4	Representative Architectures	54
2.4.1	Nested Hierarchical Controller	54
2.4.2	NIST RCS	57
2.4.3	Evaluation of hierarchical architectures	59
2.5	Advantages and Disadvantages	61
2.6	Programming Considerations	62
2.7	Summary	62
2.8	Exercises	63
2.9	End Notes	64
3	<i>Biological Foundations of the Reactive Paradigm</i>	67
3.1	Overview	67
3.1.1	Why explore the biological sciences?	69
3.1.2	Agency and computational theory	70
3.2	What Are Animal Behaviors?	73
3.2.1	Reflexive behaviors	74
3.3	Coordination and Control of Behaviors	75
3.3.1	Innate releasing mechanisms	77
3.3.2	Concurrent behaviors	82
3.4	Perception in Behaviors	83
3.4.1	Action-perception cycle	83
3.4.2	Two functions of perception	85
3.4.3	Gibson: Ecological approach	85
3.4.4	Neisser: Two perceptual systems	90
3.5	Schema Theory	91
3.5.1	Behaviors and schema theory	92
3.6	Principles and Issues in Transferring Insights to Robots	97
3.7	Summary	99
3.8	Exercises	100
3.9	End Notes	102
4	<i>The Reactive Paradigm</i>	105
4.1	Overview	105
4.2	Attributes of Reactive Paradigm	108
4.2.1	Characteristics and connotations of reactive behaviors	110
4.2.2	Advantages of programming by behavior	112
4.2.3	Representative architectures	113

4.3	Subsumption Architecture	113
4.3.1	Example	115
4.3.2	Subsumption summary	121
4.4	Potential Fields Methodologies	122
4.4.1	Visualizing potential fields	123
4.4.2	Magnitude profiles	126
4.4.3	Potential fields and perception	128
4.4.4	Programming a single potential field	129
4.4.5	Combination of fields and behaviors	130
4.4.6	Example using one behavior per sensor	134
4.4.7	Pfields compared with subsumption	136
4.4.8	Advantages and disadvantages	145
4.5	Evaluation of Reactive Architectures	147
4.6	Summary	148
4.7	Exercises	149
4.8	End Notes	152
5	<i>Designing a Reactive Implementation</i>	155
5.1	Overview	155
5.2	Behaviors as Objects in OOP	157
5.2.1	Example: A primitive move-to-goal behavior	158
5.2.2	Example: An abstract follow-corridor behavior	160
5.2.3	Where do releasers go in OOP?	162
5.3	Steps in Designing a Reactive Behavioral System	163
5.4	Case Study: Unmanned Ground Robotics Competition	165
5.5	Assemblages of Behaviors	173
5.5.1	Finite state automata	174
5.5.2	A Pick Up the Trash FSA	178
5.5.3	Implementation examples	182
5.5.4	Abstract behaviors	184
5.5.5	Scripts	184
5.6	Summary	187
5.7	Exercises	188
5.8	End Notes	191
6	<i>Common Sensing Techniques for Reactive Robots</i>	195
6.1	Overview	196
6.1.1	Logical sensors	197
6.2	Behavioral Sensor Fusion	198

6.3	Designing a Sensor Suite	202
6.3.1	Attributes of a sensor	203
6.3.2	Attributes of a sensor suite	206
6.4	Proprioceptive Sensors	207
6.4.1	Inertial navigation system (INS)	208
6.4.2	GPS	208
6.5	Proximity Sensors	210
6.5.1	Sonar or ultrasonics	210
6.5.2	Infrared (IR)	216
6.5.3	Bump and feeler sensors	217
6.6	Computer Vision	218
6.6.1	CCD cameras	219
6.6.2	Grayscale and color representation	220
6.6.3	Region segmentation	226
6.6.4	Color histogramming	228
6.7	Range from Vision	231
6.7.1	Stereo camera pairs	232
6.7.2	Light stripers	235
6.7.3	Laser ranging	239
6.7.4	Texture	241
6.8	Case Study: Hors d'Oeuvres, Anyone?	242
6.9	Summary	250
6.10	Exercises	251
6.11	End Notes	254
7	<i>The Hybrid Deliberative/Reactive Paradigm</i>	257
7.1	Overview	257
7.2	Attributes of the Hybrid Paradigm	259
7.2.1	Characteristics and connotation of reactive behaviors in hybrids	261
7.2.2	Connotations of "global"	262
7.3	Architectural Aspects	262
7.3.1	Common components of hybrid architectures	263
7.3.2	Styles of hybrid architectures	264
7.4	Managerial Architectures	265
7.4.1	Autonomous Robot Architecture (AuRA)	265
7.4.2	Sensor Fusion Effects (SFX)	268
7.5	State-Hierarchy Architectures	274
7.5.1	3-Tiered (3T)	274

7.6	Model-Oriented Architectures	277
7.6.1	Saphira	278
7.6.2	Task Control Architecture (TCA)	280
7.7	Other Robots in the Hybrid Paradigm	283
7.8	Evaluation of Hybrid Architectures	284
7.9	Interleaving Deliberation and Reactive Control	285
7.10	Summary	288
7.11	Exercises	289
7.12	End Notes	291
8	Multi-agents	293
8.1	Overview	293
8.2	Heterogeneity	296
8.2.1	Homogeneous teams and swarms	296
8.2.2	Heterogeneous teams	297
8.2.3	Social entropy	300
8.3	Control	301
8.4	Cooperation	303
8.5	Goals	304
8.6	Emergent Social Behavior	305
8.6.1	Societal rules	305
8.6.2	Motivation	307
8.7	Summary	309
8.8	Exercises	310
8.9	End Notes	312
II	Navigation	315
9	Topological Path Planning	325
9.1	Overview	325
9.2	Landmarks and Gateways	326
9.3	Relational Methods	328
9.3.1	Distinctive places	329
9.3.2	Advantages and disadvantages	331
9.4	Associative Methods	333
9.4.1	Visual homing	334
9.4.2	QualNav	335

9.5	Case Study of Topological Navigation with a Hybrid Architecture	338
9.5.1	Path planning	339
9.5.2	Navigation scripts	343
9.5.3	Lessons learned	346
9.6	Summary	348
9.7	Exercises	348
9.8	End notes	350
10	Metric Path Planning	351
10.1	Objectives and Overview	351
10.2	Configuration Space	353
10.3	Cspace Representations	354
10.3.1	Meadow maps	354
10.3.2	Generalized Voronoi graphs	357
10.3.3	Regular grids	358
10.3.4	Quadrees	359
10.4	Graph Based Planners	359
10.5	Wavefront Based Planners	365
10.6	Interleaving Path Planning and Reactive Execution	367
10.7	Summary	371
10.8	Exercises	372
10.9	End Notes	374
11	Localization and Map Making	375
11.1	Overview	375
11.2	Sonar Sensor Model	378
11.3	Bayesian	380
11.3.1	Conditional probabilities	381
11.3.2	Conditional probabilities for $P(H s)$	384
11.3.3	Updating with Bayes' rule	385
11.4	Dempster-Shafer Theory	386
11.4.1	Shafer belief functions	387
11.4.2	Belief function for sonar	389
11.4.3	Dempster's rule of combination	390
11.4.4	Weight of conflict metric	394
11.5	HIMM	395
11.5.1	HIMM sonar model and updating rule	395
11.5.2	Growth rate operator	398

11.6	Comparison of Methods	403
11.6.1	Example computations	403
11.6.2	Performance	411
11.6.3	Errors due to observations from stationary robot	412
11.6.4	Tuning	413
11.7	Localization	415
11.7.1	Continuous localization and mapping	416
11.7.2	Feature-based localization	421
11.8	Exploration	424
11.8.1	Frontier-based exploration	425
11.8.2	Generalized Voronoi graph methods	427
11.9	Summary	428
11.10	Exercises	431
11.11	End Notes	434
12	<i>On the Horizon</i>	435
12.1	Overview	435
12.2	Shape-Shifting and Legged Platforms	438
12.3	Applications and Expectations	442
12.4	Summary	445
12.5	Exercises	445
12.6	End Notes	447
	<i>Bibliography</i>	449
	<i>Index</i>	459

6

Common Sensing Techniques for Reactive Robots

Chapter Objectives:

- Describe the difference between *active* and *passive* sensors, and give examples of each.
- Define each of the following terms in one or two sentences: *proprioception*, *exteroception*, *exproprioception*, *proximity sensor*, *logical sensor*, *false positive*, *false negative*, *hue*, *saturation*, *image*, *pixel*, *image function*, *computer vision*.
- List the metrics for rating individual sensors and a sensor suite, and apply these metrics to a particular application.
- Describe the problems of *specular reflection*, *cross-talk*, and *foreshortening* with an ultrasonic transducer, and if given a 2D line drawing of surfaces, illustrate where each of these problems would likely occur.
- Describe the types of *behavioral sensor fusion* and be able to apply to a real-world problem.
- Write perceptual schemas from any logically equivalent range sensor to produce a polar plot percept for obstacle avoidance behavior.
- If given a small interleaved RGB image and a range of color values for a region, be able to 1) threshold the image on color and 2) construct a color histogram.
- Write computer vision code to enable a robot to imprint on and track a color.

6.1 Overview

To review Ch. 3, perception in a reactive robot system has two roles: *to release a behavior*, and *to support or guide the actions of the behavior*. All sensing is behavior-specific, where behaviors may tap into the same sensors, but use the data independently of each other. Also, the connotation of reactive robots is that behaviors are most often stimulus-response, relying on direct perception rather than require memory. Recognition is not compatible with reactivity. In order for a reactive robot to be successful in theory, it must have reliable perception since perception and action are tightly coupled. For it to be successful in practice, a robot has to have perceptual hardware and software which updates quickly. This chapter covers the most common sensing techniques for reactive robots. The sensors and sensing techniques described in this chapter also apply to robot architectures in the Hybrid paradigm, since Hybrid architectures use reactive behaviors.

SENSOR
TRANSDUCER

Regardless of sensor hardware or application, sensing and sensors can be thought of interacting with the world and robots as shown in Fig. 6.1. The *sensor* is a device that measures some attribute of the world. The term *transducer* is often used interchangeably with *sensor*. A transducer is the mechanism, or element, of the sensor that transforms the energy associated with what is being measured into another form of energy.² A sensor receives energy and transmits a signal to a display or a computer. Sensors use transducers to change the input signal (sound, light, pressure, temperature, etc.) into an analog or digital form capable of being used by a robot. In a reactive robot, the sensor observation is intercepted by a perceptual schema which extracts the relevant percept of the environment for the behavior. This percept is then used by the motor schema, which leads to an action.

PASSIVE SENSOR
ACTIVE SENSOR

ACTIVE SENSING

A sensor is often classified as being either *passive sensor* or *active sensor*. Passive sensors rely on the environment to provide the medium for observation, e.g., a camera requires a certain amount of ambient light to produce a usable picture. Active sensors put out energy in the environment to either change the energy or enhance it. A sonar sends out sound, receives the echo, and measures the time of flight. An X-ray machine emits X-rays and measures the amount blocked by various types of tissue. Although a camera is a passive device, a camera with a flash is active. The term *active sensor* is not the same as *active sensing*. Active sensing connotes the system for using an effector to dynamically position a sensor for a "better look." A camera with a flash is an active sensor; a camera on a pan/tilt head with algorithms to direct the camera to turn to get a better view is using active sensing.

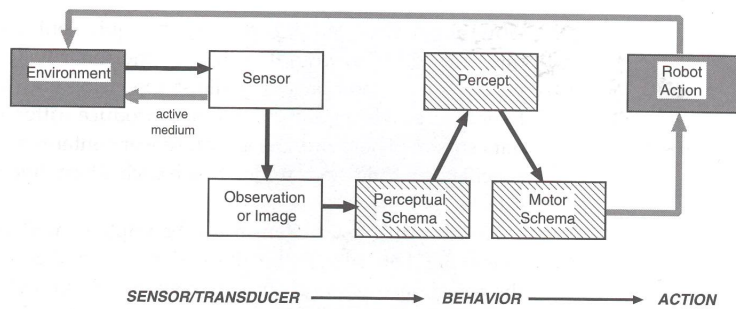


Figure 6.1 A model of sensing.

Different sensors measure different forms of energy. This in turns leads to different types of processing. Sensors which measure the same form of energy and process it in similar ways form a sensor *modality*. A sensor modality refers to what the raw input to the sensor is: sound, pressure, temperature, light, and so on. In some regards, modalities are similar to the five senses in humans. A modality can be further subdivided, for instance, vision can be decomposed into visible light, infrared light, X-rays, and other modalities.

6.1.1 Logical sensors

A powerful abstraction of sensors is *logical sensors*, first introduced by Henderson and Shilcrat.⁶⁵ A logical sensor is a unit of sensing or module that supplies a particular percept. It consists of the signal processing from the physical sensor and the software processing needed to extract the percept; it is the functional building block for perception. A logical sensor can be easily implemented as a perceptual schema.

An overlooked aspect of a logical sensor is that it contains all available alternative methods of obtaining that percept. For example, a percept commonly used for obstacle avoidance is a polar plot of range data. The logical sensor for the percept might be named `range_360` and return a data structure or object specifying the polar plot. The logical sensor would go further and list all the possible ways the robot had for constructing a polar plot of that form. The robot might be able to use sonar, a laser, stereo vision, or texture. Each of those modules would be *logically equivalent*; that is, they would return the same percept data structure so they can be used interchangeably.

However, they wouldn't necessarily be equivalent in performance or update rate. As will be seen in this chapter, the sonar is liable to produce a noisy percept in a second or two, while stereo vision may take minutes. Even different stereo vision algorithms may produce different results on the same data stream. Therefore, the logical sensor contains a selector function which specifies the conditions under which each alternative is useful and should be selected.

Notice that a logical sensor can be implemented as a perceptual schema, where the methods are the alternative means of generating the percept and the coordinated control strategy contains the knowledge as to when a particular method is appropriate. Also note that each individual method can be implemented as a perceptual schema, leading to the recursive, building-block effect.

In reactive systems, the term logical sensor has degenerated somewhat from its original usage and is essentially equivalent to a perceptual schema. "Logical sensor" is often used to connote information hiding, where the particular sensor and processing algorithm is hidden in the "package." This is useful because a robot might use the same physical sensor in two different ways. An avoid behavior might use a polar plot of sonar range data, while a panic-stop behavior might use the minimum of all the incoming sonar data. Since the perceptual schema use the raw sonar data differently, it is as if they were different sensors.

6.2 Behavioral Sensor Fusion

SENSOR FUSION

Sensor fusion is a broad term used for any process that combines information from multiple sensors into a single percept. The motivation for sensor fusion stems from three basic combinations of sensors: *redundant* (or competing), *complementary*, and *coordinated*. Although many researchers treat sensor fusion as a means of constructing a global world model in a hierarchical or deliberative system, sensor fusion can be incorporated into behaviors through *sensor fission*, *action-oriented sensor fusion*, and *sensor fashion*.

REDUNDANT COMPLEMENTARY COORDINATED

FALSE POSITIVE

FALSE NEGATIVE

In some cases multiple sensors are used when a particular sensor is too imprecise or noisy to give reliable data. Adding a second sensor can give another "vote" for the percept. When a sensor leads the robot to believe that a percept is present, but it is not, the error is called a *false positive*. The robot has made a positive identification of percept, but it was false. Likewise, an error where the robot misses a percept is known as a *false negative*. Sensors

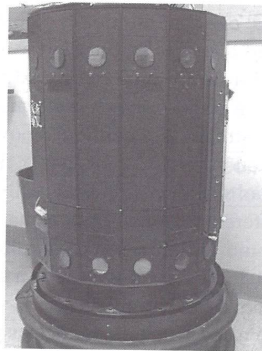


Figure 6.2 Example of redundant top and bottom sonar rings.

will often produce different false positive and false negative rates. Whether a robot can tolerate a higher false positive or false negative rate depends on the task.

When the sensors are both returning the same percept, the sensors are considered *redundant*. An example of *physical redundancy* is shown in Fig. 6.2, where a Nomad 200 has two sonar rings. The sonar software returns the minimum reading (shortest range) from the two, providing a more reliable reading for low objects which would ordinarily specularly reflect the beam from the upper sonar. Sensors can also be *logically redundant*, where they return identical percepts but use different modalities or processing algorithms. An example is extracting a range image from stereo cameras and from a laser range finder. Sometimes redundant sensors are called *competing sensors*, because the sensors can be viewed as competing to post the “winning” percept.

Complementary sensors provide disjoint types of information about a percept. In behavioral sensor fusion for urban search and rescue, a robot may search for survivors by fusing observations from a thermal sensor for body heat with a camera detecting motion. Both logical sensors return some aspect of a “survivor,” but neither provides a complete view. Coordinated sensors use a sequence of sensors, often for cue-ing or providing focus-of-attention. A predator might see motion, causing it to stop and examine the scene more closely for signs of prey.

Most of the work on sensor fusion treats it as if it were a deliberative pro-

REDUNDANT
PHYSICAL
REDUNDANCY

LOGICALLY
REDUNDANT

COMPETING SENSORS

SENSOR FISSION

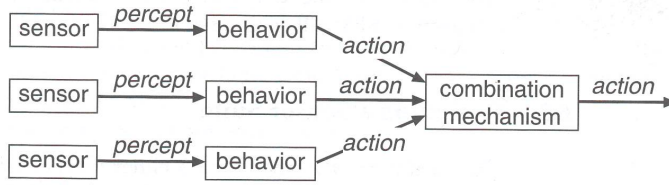
cess: one that requires a global world model. Early work in reactive systems used robots with only a few simple sensors, a sonar or sonar ring for range and a camera for color, texture, or motion affordances. As a result, there was a philosophy of one sensor per behavior. Behaviors could share a sensor stream, but without knowing it. This philosophy led to the approach taken by Brooks that sensor fusion at the behavioral level was a mirage. Instead sensor fusion was really multiple instances of the same behavior with different sensor inputs being combined together. To an outside observer, it would look like some complicated process was being enacted inside the robot, but in fact it would be a simple competition with an emergent behavior. Brooks dubbed this *sensor fission* in part as a take off on the connotations of the word “fusion” in nuclear physics. In nuclear fusion, energy is created by forcing atoms and particles together, while in fission, energy is created by separating atoms and particles. Fig. 6.3a shows a diagram of sensor fission.

ACTION-ORIENTED
SENSOR FUSION

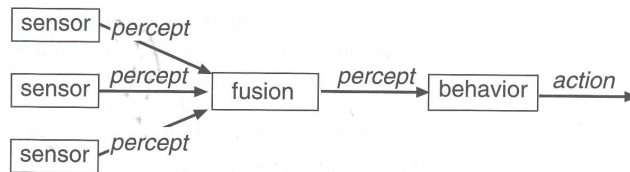
Murphy reported on studies from cognitive psychology and neurophysiology showing that behavioral sensor fusion does occur in animals, and therefore should be part of a robot’s behavioral repertoire.⁹⁹ The gist of the studies was that sensor fusion does occur in behaviors. The sensor pathways throughout the brain remain separate and can be routed to multiple behaviors in the superior colliculus. Only when the sensor signals routed to the portion of the brain associated with a particular behavior arrive at that location does there appear to be any transformation. The transformation appears to be a new common representation. Any or all of these sensor streams can be active and influence the resulting behavior. For example, consider the predation behavior in cats. If a cat hears a noise and sees a movement, it will react more strongly than if it has only a single stimulus. This type of sensor fusion is called *action-oriented sensor fusion* to emphasize that the sensor data is being transformed into a behavior-specific representation in order to support a particular action, not for constructing a world model. Fig. 6.3b shows a diagram of action-oriented sensor fusion.

SENSOR FASHION

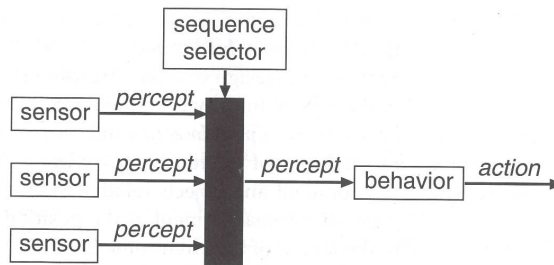
Sensor fission and action-oriented sensor fusion cover competing and complementary sensing. Sensor fission is by definition a competitive method, though complementary sensors may be used to support a particular instance of a behavior. Action-oriented sensor fusion is not restricted to either competing or complementary sensors, since the behavior makes a local transformation anyway. This leaves the category of coordinated sensors untouched. Arkin filled in the apparent gap by calling coordination *sensor fashion*, an alliterative name intended to imply the robot was changing sensors with changing circumstances just as people change styles of clothes with the seasons. A



a.



b.



c.

Figure 6.3 Three types of behavioral sensor fusion: a.) sensor fission, b.) action-oriented sensor fusion, and c.) sensor fusion.

diagram of sensor fusion is shown in Fig. 6.3c.

Murphy incorporated pathways for all three types of behavioral sensor fusion as well as other deliberative forms into the Sensor Fusion Effects (SFX) architecture. SFX will be covered in Ch. 7.

6.3 Designing a Sensor Suite

Historically, reactive robots used either inexpensive infrared (IR) or ultrasonic transducers to detect range. The earliest behaviors focused on basic navigational skills such as obstacle avoidance and wall following. The percept for these behaviors all involve knowing the distance to an occupied area of space. Now with the advent of inexpensive miniature cameras and laser range finders for consumer applications, computer vision is becoming increasingly common. In agricultural and transportation applications of reactive robots, GPS technology has become popular as well. This chapter attempts to cover the basics of these sensing modalities, and how they are used in mobile robots. Because the sensor market is rapidly changing, the chapter will focus on how to design a suite of sensors for use by a robot, rather the device details.

An artificially intelligent robot has to have some sensing in order to be considered a true AI robot. If it cannot observe the world and the effects of its actions, it cannot react. As noted in the chapter on "Action-Oriented Perception" in *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, the design of a set of sensors for a robot begins with an assessment of the type of information that needs to be extracted.¹⁴ This information can be either from *proprioception* (measurements of movements relative to an internal frame of reference), *exteroception* (measurements of the layout of the environment and objects relative to the robot's frame of reference) or *exproprioception* (measurement of the position of the robot body or parts relative to the layout of the environment).

PROPRIOCEPTION
EXTEROCEPTION
EXPROPRIOCEPTION

The Colorado School of Mines fielded an entry to the 1995 UGV competition entry discussed in Ch. 5. This provides an example of different types of sensing for a path following robot. In 1995, the follow-path behavior was expanded to track both lines of the path using a wide angle lens on the camera. follow-path could be considered exteroceptive because it acquired information on the environment. However, the camera for the robot was mounted on a panning mast, which was intended to turn to keep the line in view, no matter what direction the path turned in. Therefore, the

robot had to know where the camera was turned relative to the robot's internal frame of reference in order to correctly transform the location of a white line in image coordinates to a steering direction. This meant the information needed for follow-path had both proprioceptive and exteroceptive components, making the perception somewhat exproprioceptive. (If the robot was extracting the pose of its camera from exteroception, it would be clearly exproprioceptive.)

Due to a programming error, the follow-path behavior incorrectly assumed that the exteroceptive camera data had been transformed by the proprioceptive shaft encoder data from the panning mast into exproprioceptive data. The robot needed the exproprioception to determine where it should move next: turn to follow the path in camera coordinates, plus the compensation for the current camera pan angle. The programming error resulted in the robot acting as if the camera was aligned with the center of the robot at all times. But the camera might be turned slightly to maintain the view of both lines of the path through the pan-camera behavior. The resulting navigational command might be to turn, but too little to make a difference, or even to turn the wrong way. This subtle error surfaced as the robot went around hair pin turns, causing the robot to go consistently out of bounds.

6.3.1 Attributes of a sensor

As can be seen by the above example, robots may have dead reckoning capabilities, but will always have some type of exteroceptive sensor. Otherwise, the robot cannot be considered reactive: there would be no stimulus from the world to generate a reaction. The set of sensors for a particular robot is called a *sensor suite*. Following *Sensors for Mobile Robots*,⁵² in order to construct a sensor suite, the following attributes should be considered for each sensor:

1. **Field of view and range.** Every exteroceptive sensor has a region of space that it is intended to cover. The width of that region are specified by the sensor's *field of view*, often abbreviated as FOV. The field of view is usually expressed in degrees; the number of degrees covered vertically may be different from the number of degrees covered horizontally. Field of view is frequently used in photography, where different lenses capture different size and shape areas. A wide angle lens will often cover up to 70°, while a "regular" lens may only have a field of view around 27°. The distance that the field extends is called the range.

The *field of view (FOV)* can be thought of in terms of egocentric spherical

SENSOR SUITE

FIELD OF VIEW (FOV)

HORIZONTAL FOV
VERTICAL FOV
RANGE

coordinates, where one angle is the *horizontal FOV* and the other is the *vertical FOV*. The other aspect is the *range*, or how far the sensor can make reliable measurements. In spherical coordinates, this would be the values of r that defined the depth of the operating range.

Field of view and range are obviously critical in matching a sensor to an application. If the robot needs to be able to detect an obstacle when it's 8 feet away in order to safely avoid it, then a sensor with a range of 5 feet will not be acceptable.

RESOLUTION

2. Accuracy, repeatability, and resolution. Accuracy refers to how correct the reading from the sensor is. But if a reading for the same conditions is accurate only 20% of the time, then the sensor has little repeatability. If the sensor is consistently inaccurate in the same way (always 2 or 3 cm low), then the software can apply a bias (add 2 centimeters) to compensate. If the inaccuracy is random, then it will be difficult to model and the applications where such a sensor can be used will be limited. If the reading is measured in increments of 1 meter, that reading has less *resolution* than a sensor reading which is measured in increments of 1 cm.

3. Responsiveness in the target domain. Most sensors have particular environments in which they function poorly. Another way of viewing this is that the environment must allow the signal of interest to be extracted from noise and interference (e.g., have a favorable signal-to-noise ratio). As will be seen below, sonar is often unusable for navigating in an office foyer with large amounts of glass because the glass reflects the sound energy in ways almost impossible to predict. It is important to have characterized the ecological niche of the robot in terms of what will provide, absorb, or deflect energy.

4. Power consumption. Power consumption is always a concern for robots. Since most robots operate off of batteries, the less power they consume, the longer they run. For example, the battery life on a Nomad 200, which carries five batteries, was improved from four hours to six by shutting off all sensors. Power is so restricted on most mobile robots that many robot manufacturers will swap microprocessor chips just to reduce the power drain (which was part of the motivation for the Transmeta Crusoe chip). Sensors which require a large amount of power are less desirable than those which do not. In general, passive sensors have less power demands than active sensors because they are not emitting energy into the environment.

HOTEL LOAD

The amount of power on a mobile robot required to support a sensor package (and any other electronics such as a microprocessor and communications links) is sometimes called the *hotel load*. The sensor suite is the "guest" of the

LOCOMOTION LOAD

platform. The power needed to move the robot is called the *locomotion load*. Unfortunately, many robot manufacturers focus on only the locomotion load, balancing power needs with the desire to reduce the overall weight and size. This leads to a very small hotel load, and often prevents many sensors from being added to platform.

5. Hardware reliability. Sensors often have physical limitations on how well they work. For example, Polaroid sonars will produce incorrect range reading when the voltage drops below 12V. Other sensors have temperature and moisture constraints which must be considered.

6. Size. The size and weight of a sensor does affect the overall design. A microrover on the order of a shoebox will not have the power to transport a large camera or camcorder, but it may be able to use a miniature "Quick-Cam" type of camera.

The above list concentrated on considerations for the physical aspects of the sensor. However, the sensors only provide observations; without the software perceptual schemas, the behaviors cannot use the sensors. Therefore, the software that will process the information from a sensor must be considered as part of the sensor selection process.

7. Computational complexity. Computational complexity is the estimate of how many operations an algorithm or program performs. It is often written as a function O , called the "order," where $O(x)$ means the number of operations is proportional to x . x is often a function itself. Lower orders are better. An algorithm that executes with $O(n)$ equally consuming operations is faster than one with $O(n^2)$ operations. (If you doubt this, see if you can find a positive, whole number value of n such that $n > n^2$.) Computational complexity has become less critical for larger robots, with the rapid advances in processors and miniaturization of components. However, it remains a serious problem for smaller vehicles.

8. Interpretation reliability. The designer should consider how reliable the sensor will be for the ecological conditions and for interpretation. The robot will often have no way of determining when a sensor is providing incorrect information. As a result the robot may "hallucinate" (think it is seeing things that are not there) and do the wrong thing. Many sensors produce output which are hard for human to interpret without years of training; medical X-rays are one example, and synthetic aperture radar (SAR) which produces polar plots is another. If a sensor algorithm was not working properly in these modalities, the designer might not be skilled enough to notice it. Therefore, the algorithms themselves must be reliable.

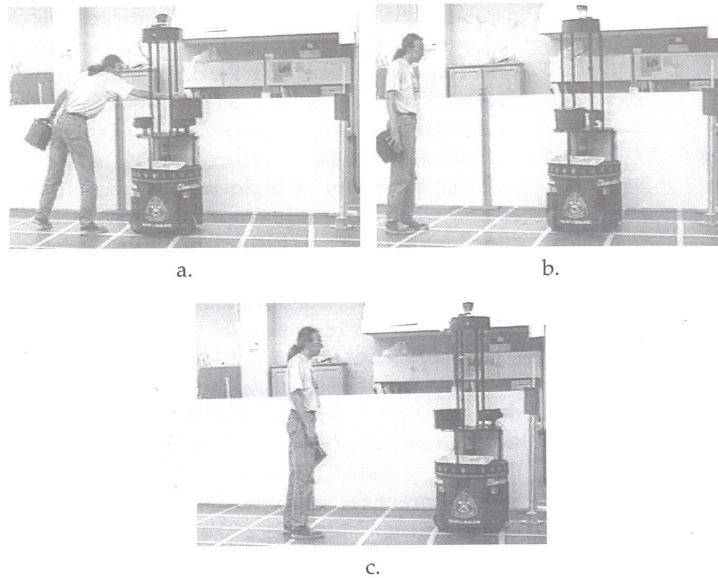


Figure 6.4 Sequence showing a Denning mobile robot with redundant cameras, responding to sensing failures introduced by Dave Hershberger.

6.3.2 Attributes of a sensor suite

Everett⁵² recommends that the following attributes should be considered for the *entire sensing suite*:

1. **Simplicity.** As with any design, simple sensors which have straightforward hardware and operating principles are more desirable than sensors which are complex and require constant maintenance.

2. **Modularity.** In order to reconfigure a sensor suite (add new sensors, remove ones that aren't appropriate for the task), the designer must be able to remove one sensor and/or its perceptual schema without impacting any other sensing capability.

3. **Redundancy.** In most military and industrial applications, it is imperative that the robot function correctly, since it would put people at risk to try

to retrieve the robot and fix it. Sensors are especially important because a faulty sensor can cause the robot to “hallucinate.” As a result, sensor suites may offer some sort of redundancy.

PHYSICAL REDUNDANCY

There are two types of redundancy. *Physical redundancy* means that there are several instances of physically identical sensors on the robot. Fig. 6.4 shows a robot with redundant cameras. In this case, the cameras are mounted 180° apart, and when one sensor fails, the robot has to “drive backwards” in order to accomplish its task.

LOGICAL REDUNDANCY

Logical redundancy means that another sensor using a different sensing modality can produce the same percept or releaser. For example, the Mars Sojourner mobile robot had a laser striping system for determining the range to obstacles, which emitted a line of light. If the surface was flat, a camera would see a flat line, whereas if the surface was rough the line would appear jagged. The robot also carried a second camera which could do obstacle detection via stereo triangulation. The laser striping sensor and the stereo cameras are logically redundant. They are not physically redundant, but they produce the same overall information: the location of obstacles relative to the robot. However, logically redundant sensors are not necessarily equivalent in processing speed or accuracy and resolution. A stereo range sensor and algorithm computes the range much slower than a laser striping system.

FAULT TOLERANCE

Physical redundancy introduces new issues which are the area of active research investigation. Possibly the most intriguing is how a robot can determine that a sensor (or algorithm) has failed and needs to be swapped out. Surviving a failure is referred to as *fault tolerance*. Robots can be programmed in most cases to tolerate faults as long as they can identify when they occur.

6.4 Proprioceptive Sensors

SHAFT ENCODER

Proprioception is dead reckoning, where the robot measures a signal originating within itself. In biology, this is usually some measure of how far an arm or leg has been extended or retracted. In robotics, actuators are generally motors. Many motors come with a *shaft encoder* which measures the number of turns the motor has made. If the gearing and wheel size is known, then the number of turns of the motor can be used to compute the number of turns of the robot's wheels, and that number can be used to estimate how far the robot has traveled.

Proprioception is often only an estimate. This is due to the impact of the environment on the actual movement of the robot. Arkin in his PhD thesis⁸ showed that the same wheeled robot, HARV, traveled different distances for the same encoder count on a sidewalk, grass, and wet grass. The texture of the different surfaces caused the wheels of HARV mobile to slip to varying degrees. A robot on a tiled floor might slip twice as much as on dry grass.

6.4.1 Inertial navigation system (INS)

INERTIAL NAVIGATION SYSTEMS (INS)

Aircraft, submarines, and missiles use *inertial navigation systems (INS)*. INS measure movements electronically through miniature accelerometers. As long as the movements are smooth, with no sudden jarring, and the samples are taken frequently, an INS can provide accurate dead reckoning to 0.1 percent of the distance traveled.⁵² However, this technology is unsuitable for mobile robots for several reasons. The cost of an INS is prohibitive; units run from \$50,000 to \$200,000 USD. The cost is due in part to having to stabilize the accelerometers with gyroscopes, as well as the nature of precision electronics. Mobile robots often violate the constraint that motion must be smooth. A hard bump or sudden turn can exceed the accelerometers' measurement range, introducing errors. INS systems are typically big; smaller devices have less accuracy. Sojourner, the Mars rover, carried an INS system. In one trek, it would have stopped 30 cm from a rock it was to sample if it had just used proprioception. Instead, by using exteroception, it got within 4 cm.

6.4.2 GPS

PRECISION AGRICULTURE

GPS, or Global Positioning System, is becoming more common on robots, especially those used to automate farm equipment (an effort called *precision agriculture*). GPS systems work by receiving signals from satellites orbiting the Earth. The receiver triangulates itself relative to four GPS satellites, computing its position in terms of latitude, longitude, altitude, and change in time. GPS isn't a proprioceptive sensor per se since the robot must receive signals from the satellites, external to the robot. However, they are not exteroceptive sensors either, since the robot isn't computing its position relative to its environment. Since they tend to be used in place of dead reckoning on outdoor robots, only GPS will be covered here.

Currently the only sets of GPS satellites that a receiver can triangulate itself against are the Navstar "constellation" maintained by the United States Air Force Space Command or the Russian counterpart, GLONASS, main-

tained by the Russian Federation Ministry of Defense. There are two types of channels on Navstar, one public, called the Standard Positioning System, and an encrypted signal, the Precise Positioning System. Until early in the year 2000, the U.S. military actually introduced an error in the satellite message as to where the satellite actually is, which could result in triangulation errors of up to 100 meters. The error was called *selective availability*, because it made accurate positioning available only to those users selected by the U.S. military. This was intended to prevent a hostile country from putting a GPS receiver on a guided missile and precisely targeting where the President is giving a talk. Selective availability was turned off in part because of the rise of civilian uses of GPS, and because it led to interoperability with groups working with the U.S. military who were using commercial, not military, GPS.

Many inexpensive hand-held receivers sold to hunters and hikers attempt to improve on localization by averaging or filtering the readings. This can reduce the error down to 10-15 meters. Surveyors and GPS specialty companies such as Trimble and Rockwell have found a way to subtract the error in the public channel and get performance near the Y-code's rumored accuracy of centimeters. The method is called *differential GPS (DGPS)*, where two GPS receivers are used. One remains stationary, while the other is put on the robot. If the two receivers are observing the same satellites, then any sudden change in position on the stationary "base" receiver is due to the induced error and can be subtracted from the readings at the robot GPS. The ultimate fix to the induced error will probably come in a few years due to the commercial sector. A consortium of private companies is planning to launch a new constellation of GPS satellites, Teledesic, which will emit accurate data at all times in a format that can be decoded only by chips licensed by the consortium. Teledesic is scheduled to go on-line in 2004.

GPS and DGPS are not complete solutions to the dead reckoning problem in mobile robots for at least two reasons. First, GPS does not work indoors in most buildings, especially offices or factories with large amounts of steel-reinforced concrete. As with cellular phones, these structures interrupt the reception of radio signals. Likewise, GPS may not work outdoors in major cities where skyscrapers act as *urban canyons* and interfere with reception. Second, commercial DGPS systems cost on the order of \$30,000 USD, which is prohibitively high. Several web sites now offer free "do-it-yourself" DGPS code to create a DGPS from two inexpensive receivers.

6.5 Proximity Sensors

Proximity sensors measure the relative distance (range) between the sensor and objects in the environment. Since the sensor is mounted on the robot, it is a straightforward computation to translate a range relative to the sensor to a range relative to the robot at large. Most proximity sensors are active. Sonar, also called ultrasonics, is the most popular proximity sensor, with infrared, bump, and feeler sensors not far behind.

6.5.1 Sonar or ultrasonics

Sonar refers to any system for using sound to measure range. Sonars for different applications operate at different frequencies; for example, a sonar for underwater vehicles would use a frequency appropriate for traveling through water, while a ground vehicle would use a frequency more suited for air. Ground vehicles commonly use sonars with an ultrasonic frequency, just at the edge of human hearing. As a result the terms “sonar” and “ultrasonics” are used interchangeably when discussing extracting range from acoustic energy.

TIME OF FLIGHT

Ultrasonics is possibly the most common sensor on commercial robots operating indoors and on research robots. They are active sensors which emit a sound and measure the time it takes for the sound to bounce back. The *time of flight* (time from emission to bounce back) along with the speed of sound in that environment (remember, even air changes density with altitude) is sufficient to compute the range of the object.

Ultrasonics is common for several reasons. Its evolution paralleled the rise of the Reactive Paradigm. In the mid-1980's, Hans Moravec did impressive robot navigation with a ring of sonars. The ring configuration gave a 360° coverage as a polar plot. This ring was developed by one of the first mobile robot manufacturers, Denning Robotics, and since then sonar rings are often referred to as “Denning rings,” regardless of manufacturer. Besides providing direct range measurements, the transducers were cheap, fast, and had terrific coverage. In the early 1980's, the Polaroid Land Corporation had developed small, inexpensive sonars for use as camera range finders. A bigger version, the Polaroid Lab Grade ultrasonic transducer, costs on the order of \$30 USD and can measure ranges from 1 to 25 feet with inch resolution over a field of view of 30°. Furthermore, the measurement time was on the order of seconds versus hours for computer vision. Ultrasonics became the sensor of choice for behavior-based robots.

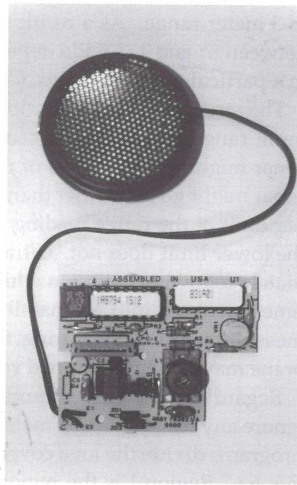


Figure 6.5 Polaroid ultrasonic transducer. The membrane is the disk.

A robotic sonar transducer is shown in Fig. 6.5. The transducer is about the size and thickness of a dollar coin, and consists of a thin metallic membrane. A very strong electrical pulse generates a waveform, causing the membrane on the transducer to produce a sound. The sound is audible as a faint clicking noise, like a crab opening and closing its pinchers. Meanwhile a timer is set, and the membrane becomes stationary. The reflected sound, or *echo*, vibrates the membrane which is amplified and then thresholded on return signal strength; if too little sound was received, then the sensor assumes the sound is noise and so ignores it. If the signal is strong enough to be valid, the timer is tripped, yielding the time of flight.

The key to how useful the data is requires understanding how the sound wave is generated by the transducer. In reality, the sound beam produces multiple secondary sound waves which interact over different regions of space around the transducer before dissipating. Secondary sound waves are called *side lobes*. Most robot systems assume that only sound from the main, or centermost, lobe is responsible for a range measurement. The width of the main lobe is often modeled as being 30° wide at about 5 meters away. However, in practice, reactive robots need to respond to obstacles in the 0.3

to 3 meter range. As a result many algorithms only treat the lobe as being between 8° and 15° wide depending on how reliable the range readings are in a particular environment. Ch. 11 will go over this in more detail.

The strength of the main lobe in the environment determines the maximum range that the sonar can extract reliability. In ideal indoor venues, a sonar might return ranges of up to 25 feet, while in the outdoors, the same sonar might go no further than 8 feet with any repeatability. So while the upper limit of the range reading depends on the sensor and the environment, the lower limit does not. Ultrasonic transducers have a "dead time" immediately following emission while the membrane vibration decays. The decay time translates into an inability to sense objects within 11 inches because measurements made during this period are unreliable because the membrane may not have stopped ringing.

Regardless of the maximum allowed range return (i.e., does the program ignore any reading over 3 meters?) and the width of the lobe, most computer programs divide the area covered by a sonar into the three regions shown in Fig. 6.6. Region I is the region associated with the range reading. It is an arc, because the object that returned the sound could be anywhere in the beam. The arc has a width, because there are some resolution and measurement errors; the width of Region I is the tolerance. Region II is the area that is empty. If that area was not empty, the range reading would have been shorter. Region III is the area that is theoretically covered by the sonar beam, but is unknown whether it is occupied or empty because it is in the shadow of whatever was in Region I. Region IV is outside of the beam and not of interest.

SPECULAR REFLECTION

Although they are inexpensive, fast, and have a large operating range, ultrasonic sensors have many shortcomings and limitations which a designer should be aware of. Ultrasonic sensors rely on reflection, and so are susceptible to *specular reflection*. Specular reflection is when the wave form hits a surface at an acute angle and the wave bounces away from the transducer. Ideally all objects would have a flat surface perpendicular to the transducer, but of course, this rarely happens. To make matters worse, the reflected signal may bounce off of a second object, and so on, until by coincidence return some energy back to the transducer. In that case, the time of flight will not correspond to the true relative range.

Even with severely acute angles, the surface is usually rough enough to send some amount of sound energy back. An exception to this is glass, which is very common in hospitals and offices where mail robots operate, but induces serious specular reflection. Fortunately this energy is often sufficiently

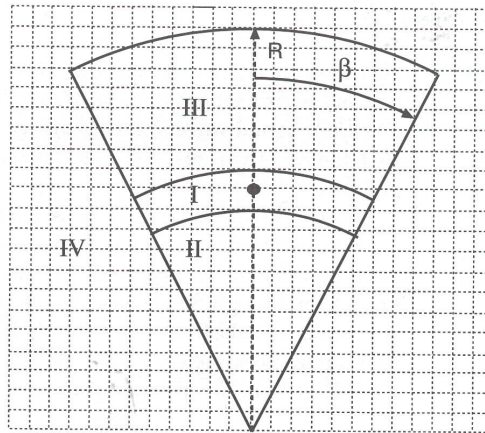


Figure 6.6 The regions of space observed by an ultrasonic sensor.

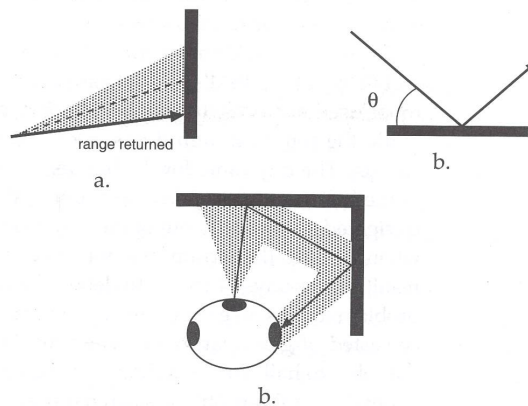


Figure 6.7 Three problems with sonar range readings: a.) foreshortening, b.) specular reflection, and c.) cross-talk.

FORESHORTENING

strong to pass the thresholding in the transducer circuit. However, a new problem, *foreshortening*, may occur. Recall that a sonar has a 30° field of view. This means that sound is being broadcast in a 30° wide cone. If the surface is not perpendicular to the transducer, one side of the cone will reach the object first and return a range first. Most software assumes the reading is along the axis of the sound wave. If it uses the reading (which is really the reading for 15°), the robot will respond to erroneous data. There is no solution to this problem.

SPECULAR REFLECTION

Specular reflection is not only by itself a significant source of erroneous readings; it can introduce a new type of error in rings of sonars. Consider a ring of multiple sonars. Suppose the sonars fire (emit a sound) at about the same time. Even though they are each covering a different region around the robot, some specularly reflected sound from a sonar might wind up getting received by a completely different sonar. The receiving sonar is unable to tell the difference between sound generated by itself or by its peers. This source of wrong reading is called *cross-talk*, because the sound waves are getting crossed. Most robot systems stagger the firing of the sonars in a fixed pattern of four sonars, one from each quadrant of the ring) at a time. This helps some with cross-talk, but is not a complete or reliable solution. If the sonar sound frequency and firing rate can be changed (which is generally not the case), then sophisticated aliasing techniques can be applied. These techniques are outside the scope of this book.

CROSS-TALK

One researcher, Monet Soldo, told a story of when she developed a reactive mobile robot for IBM's T.J. Watson Laboratories during the late 1980's. The robot used sonar as its primary sensors, and she had written behaviors to guide the robot through doors, rooms, and hall successfully at reasonable speeds. The day came for the big demonstration, which was to be held not in the hallways of the laboratory but in the front reception area. The robot navigated successfully out of the lab, down the halls, and then went berserk when it got to the atrium. She rebooted, and tried again, but with the same result. After days of trying to debug the code, she realized it wasn't a code problem, it was an environment problem: most of the atrium reception area consisted of glass partitions. The specular reflection and cross-talk caused the robot to hallucinate, although in different ways each time.

I had a similar problem when my robot started navigating in an office environment. In that environment, the robot was expected to navigate among office cubes, or work areas delimited by partitions. The partitions were covered with cloth to dampen the sound from the workers. Unfortunately, the cloth also absorbed the sound from the sonars! These stories emphasize the

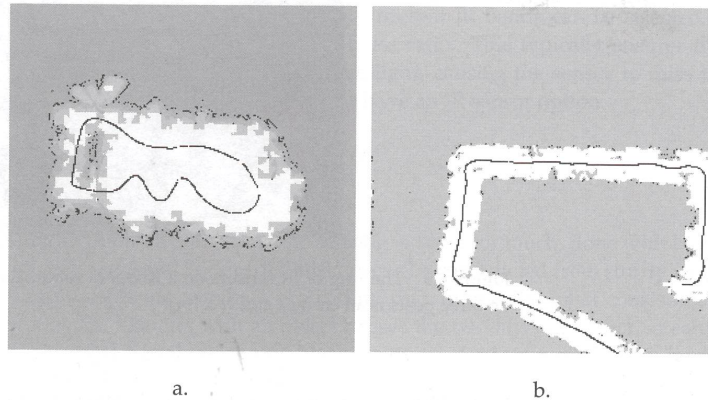


Figure 6.8 Maps produced by a mobile robot using sonars in: a.) a lab and b.) a hallway. (The black line is the path of the robot.)

need to consider the operating environment for the sensor and its impact on the signal.

The impact of the problems of specular reflection and cross talk become easier to see with plots of sonar returns overlaid with the perimeter of the area they were taken in; see Fig. 6.8. Some walls are invisible, others are too close. Clearly sonar readings have to be taken with a grain of salt.

The 30° cone also creates resolution problems. While sonars often have excellent resolution in depth, they can only achieve that at large distances if the object is big enough to send back a significant portion of the sound wave. The further away from the robot, the larger the object has to be. Most desk chairs and table tops present almost no surface area to the sensor and so the robot will often not perceive their presence and run right into them.

In practice, another problem leads to spurious sonar readings: power. The generation of a sound wave requires a significant pulse of energy. If the robot is operating at low power levels, the correct waveform will not be generated and the return signal worthless. This problem is often difficult to debug

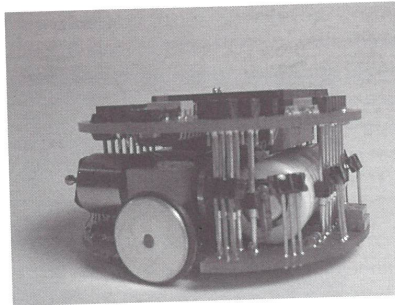


Figure 6.9 The ring of IR sensors on a Khepera robot. Each black square mounted on three posts is an emitter and receiver.

by looking at the sonar returns, which often suggest specular reflection or crosstalk. One team in a robot competition attached a voltmeter next to the emergency stop button on their robot to ensure that this problem would not go undetected.

One method of eliminating spurious readings, regardless of cause, is to take the average of three readings (current plus the last two) from each sensor. This method is fairly common on purely reactive robots, but is ad hoc. As will be seen in later chapters, other approaches treat the reading as being uncertain and apply formal evidential reasoning techniques to refine the reading. These uncertainty techniques are employed by architectures operating under the Hybrid Paradigm, and will be covered in Ch. 11.

6.5.2 Infrared (IR)

Infrared sensors are another type of active proximity sensor. They emit near-infrared energy and measure whether any significant amount of the IR light is returned. If so, there is an obstacle present, giving a binary signal. IR sensors have a range of inches to several feet, depending on what frequency of light is used and the sensitivity of the receiver. The simplest IR proximity sensors can be constructed from LEDs, which emit light into the environment and have a range of 3-5 inches. Fig. 6.9 shows the IR emitters and receivers placed side by side in a single rectangular package on a Khepera robot. These often fail in practice because the light emitted is often “washed out” by bright

ambient lighting or is absorbed by dark materials (i.e., the environment has too much noise).

In more sophisticated IR sensors, different IR bands can be selected or modulated to change the signal-to-noise ratio. This typically ensures that an object in range doesn't absorb the light, causing the sensor to miss the presence of the object. Nomad robots have an IR sensor option.

6.5.3 Bump and feeler sensors

TACTILE

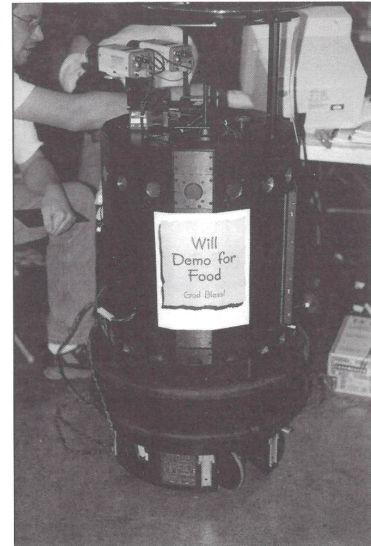
Another popular class of robotic sensing is *tactile*, or touch, done with bump and feeler sensors. Feelers or whiskers can be constructed from sturdy wires. Bump sensors are usually a protruding ring around the robot consisting of two layers. Contact with an object causes the two layers to touch, creating an electrical signal. In theory, the sensitivity of a bump sensor can be adjusted for different contact pressures; some robots may want a "light" touch to create a signal rather than a "heavier" touch. In practice, bump sensors are annoyingly difficult to adjust. In the "Hors d'Oeuvres, Anyone?" event of the 1997 AAAI Mobile Robot Competition, humans were served finger food by robot "waiters." The robot waiters were various research robots with serving trays attached. Humans were supposed to communicate to the Colorado School of Mines' Nomad 200 robot waiter (seen in Fig. 6.10) that they were done eating by kicking the bump sensor. The sensitivity of the bump sensor was so low that it often required many kicks, producing a very comical scene with Bruce Lee overtones.

Placement of bump sensors is a very important issue. The bump sensors on a Nomad 200 base clearly protect the robot only from low obstacles not perceivable by sonar. The Denning mobile robot platforms built in the 1980's used a bump sensor that looked much like a thick piece of gray tape. Denning mobile robots look like a fat version of the Nomad 200's, and the bump sensor is wrapped around the cowling of the robot at the waist level. Unfortunately, in certain turning configurations, the wheels extend beyond the cowling. In those situations, the bump sensor is totally useless in preventing the expensive synchro-drive mechanism from being damaged in a collision.

Feeler sensors are whiskers or antennae, only not as sensitive as those on animals. Contact of a whisker and an object will trip a binary switch, whereas there is reason to believe that an insect or animal can extract much more information. Bump and feeler sensors are actually tactile sensors since they require the robot to be touching something in order to generate a reading.



a.



b.

Figure 6.10 CRISbot, the Colorado School of Mines' entry in the 1997 AAAI Mobile Robot Competition Hors d'Oeuvres, Anyone? event. a.) Interacting with the audience. b.) Audience members "communicated" by kicking the two protruding bumpers near the bottom of the robot. (Photographs courtesy of AAAI.)

However, they are often interchangeable with IR sensors because IR sensors often operate over the short range (inches) with less reliability.

6.6 Computer Vision

COMPUTER VISION
IMAGE

PIXELS

Computer vision refers to processing data from any modality which uses the electromagnetic spectrum which produces an image. An *image* is essentially a way of representing data in a picture-like format where there is a direct physical correspondence to the scene being imaged. Unlike sonar, which returns a single range reading which could correspond to an object anywhere within a 30° cone, an image implies multiple readings placed in a two-dimensional array or grid. Every element in the array maps onto a small region of space. The elements in image arrays are called *pixels*, a contraction

INTRODUCTION TO AI ROBOTICS

ROBIN R. MURPHY

This text covers all the material needed to understand the principles behind the AI approach to robotics and to program an artificially intelligent robot for applications involving sensing, navigation, planning, and uncertainty. Robin Murphy is extremely effective at combining theoretical and practical rigor with a light narrative touch. In the overview, for example, she touches upon anthropomorphic robots from classic films and science fiction stories before delving into the nuts and bolts of organizing intelligence in robots.

Following the overview, Murphy contrasts AI and engineering approaches and discusses what she calls the three paradigms of AI robotics: hierarchical, reactive, and hybrid deliberative/reactive. Later chapters explore multiagent scenarios, navigation and path-planning for mobile robots, and the basics of computer vision and range sensing. Each chapter includes objectives, review questions, and exercises. Many chapters contain one or more case studies showing how the concepts were implemented on real robots. Murphy, who is well known for her classroom teaching, conveys the intellectual adventure of mastering complex theoretical and technical material.

Robin R. Murphy is Associate Professor in the Department of Computer Science and Engineering, and in the Department of Psychology, at the University of South Florida, Tampa.

Intelligent Robotics and Autonomous Agents series
A Bradford Book

Cover art: Mural, Detroit Industry, South Wall (detail), 1932–1933. Diego M. Rivera. Gift of Edsel B. Ford.
Photograph © 1991 The Detroit Institute of Arts

The MIT Press
Massachusetts Institute of Technology • Cambridge, Massachusetts 02142 • <http://mitpress.mit.edu>

MURIH 0-262-13383-0

