7 *What's Going On? Sensors*

Knowing what is going on is a requirement for survival, not to mention for intelligent behavior. If a robot is to achieve anything at all, it must be able to sense the state of its own body (its internal state; see Chapter 3) and the state of its immediate environment (external state; also see Chapter 3). In fact, as we learned in Chapter 1, for a robot to be a robot, it must be able to sense. In this chapter you will find out how the robot's ability to sense directly influences its ability to react, achieve goals, and act intelligently.

A robot typically has two types of sensors based on the source of the information it is sensing:

PROPRIOCEPTION

1. *Proprioceptive sensors:* These sensors perceive elements of the robot's internal state, such as the positions of the wheels, the joint angles of the arms, and the direction the head is facing. The term comes from the Latin word *proprius* meaning "one's own" (also featured in "proprietor," "property," and "appropriate"). *proprioception* is the process of sensing the state of one's own body. It applies to animals as much as it does to robots.

EXTEROCEPTION

2. *Exteroceptive sensors:* These sensors perceive elements of the state of the external world around the robot, such as light levels, distances to objects, and sound. The term comes from the Latin word *extra* meaning "outside" (also featured in "extrasensory," "extraterrestrial," and "extroverted"). *Exteroception* is the process of sensing the world around the robot (including sensing the robot itself).

PERCEPTUAL SYSTEM

Proprioceptive sensors and exteroceptive sensors together constitute the *perceptual system* of a robot, as shown in figure 7.1. However, one of the main challenges of robotics is that sensors themselves do not provide convenient *state* information to the robot. For example, sensors do not say: "There is

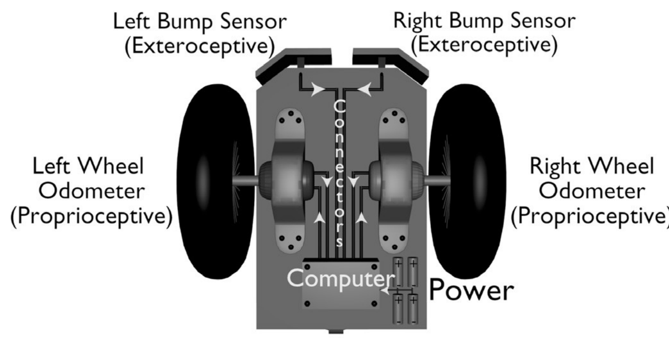

Figure 7.1 Proprioceptive and exteroceptive sensors on a simple robot.

a blue chair on your left, and your grandmother Zelda is sitting in it and looking uncomfortable." Instead, sensors may tell the robot the light levels and colors in its field of view, whether it is touching something in a particular area, whether there is a sound above some threshold, or how far away the nearest object is, and so on.

SENSORS

Rather than being magic providers of all the information the robot could possibly need, *sensors* are physical devices that measure physical quantities. Table 7.1 considers some devices and the quantities they measure.

As table 7.1 illustrates, the same physical property may be measurable with more than one type of sensor. This is very convenient, as we will find out, since sensor information is prone to noise and errors, so acquiring information from multiple sensors can provide improved accuracy.

UNCERTAINTY

Sensor noise and errors, which are inherent in physical measurement and cannot be avoided, contribute to one of the major challenges of robotics: uncertainty. *Uncertainty* refers to the robot's inability to be certain, to know for sure, about the state of itself and its environment, in order to take absolutely optimal actions at all times. Uncertainty in robotics comes from a variety of sources, including:

Physical Property	→	**Sensing Technology**
Contact	→	bump, switch
Distance	→	ultrasound, radar, infra red
Light level	→	photocells, cameras
Sound level	→	microphones
Strain	→	strain gauges
Rotation	→	encoders and potentiometers
Acceleration	→	accelerometers and gyroscopes
Magnetism	→	compasses
Smell	→	chemical sensors
Temperature	→	thermal, infra red
Inclination	→	inclinometers, gyroscopes
Pressure	→	pressure gauges
Altitude	→	altimeters

Table 7.1 Some sensors and the information they measure.

- Sensor noise and errors
- Sensor limitations
- Effector and actuator noise and errors
- Hidden and partially observable state
- Lack of prior knowledge about the environment, or a dynamic and changing environment.

Fundamentally, uncertainty stems from the fact that robots are physical mechanisms that operate in the physical world, the laws of which involve unavoidable uncertainty and lack of absolute precision. Add to that imperfect sensor and effector mechanisms and the impossibility of having total and perfect knowledge, and you can see why robotics is hard: robots must survive and perform in a messy, noisy, challenging real world. Sensors are the windows into that world, and in robotics those windows are, so far, quite small and hard to see through, metaphorically speaking.

We can think of various robot sensors in terms of the amount of information they provide. For example, a basic switch is a simple sensor that provides one bit of information, on or off. *Bit*, by the way, refers to the fundamental unit of information, which has two possible values, the binary digits 0 and 1; the word comes from "b(inary) (dig)it." In contrast, a simple camera lens (a vision sensor) is stunningly rich in information. Consider a standard camera, which has a 512 X 512 pixel lens. A *pixel* is the basic element of the

BIT

PIXEL

image on the camera lens, computer, or TV screen. Each of these 262,144 pixels may, in the simplest case, be either black or white, but for most cameras it will provide much more range. In black-and-white cameras, the pixels provide a range of gray levels, and in color cameras they provide a spectrum of colors. If that sounds like a lot of information, consider the human *retina*, the part of your eye that "sees" and passes information to the brain. The retina is a light-sensitive membrane with many layers that lines the inner eyeball and is connected to the brain via the optic nerve. That amazing structure consists of more than a hundred million photosensitive elements; no wonder robots are quite a way behind biological systems.

RETINA

Although we just mentioned this, it is worth repeating:

> *Sensors do not provide state. They provide raw measured quantities, which typically need to be processed in order to be useful to a robot.*

The more information a sensor provides, the more processing is needed. Consequently, it takes no brains at all to use a switch, but it takes a great deal of brains (in fact, a large portion of the human brain) to process inputs from vision sensors.

Simple sensors that provide simple information can be employed almost directly. For example, consider a typical mobile robot that has a switch at the front of the body. When the switch is pushed, the robot stops, because it "knows" that it has hit something; when the switch is not pushed, the robot keeps moving freely. Not all sensory data can be processed and used so simply, which is why we have brains; otherwise, we would not need them.

In general, there are two ways in which sensory information can be treated:

1. We can ask the question: "Given that sensory reading, what should I do?" or

2. We can ask the question: "Given that sensory reading, what was the world like when the reading was taken?"

Simple sensors can be used to answer the first question, which is about *action in the world*, but they do not provide enough information to answer the second question, which is about *reconstruction of the world*. If the switch on the robot indicates that it has hit something, that is all the robot knows; it cannot deduce anything more, such as the shape, color, size, or any other information about the object it has contacted.

ACTION IN THE WORLD

RECONSTRUCTION

At the other extreme, complex sensors such as vision provide a great deal more information (a lot of bits), but also require a great deal more processing to make that information useful. They can be used to answer both questions asked above. By providing more information, a sensor can allow us to attempt to *reconstruct* the world that produced the reading, at least with respect to the particular measured quantity. In a camera image we can look for lines, then objects, and finally try to identify a chair or even a grandmother in the image. For an idea of what it takes to do that, see Chapter 9.

SIGNAL-TO-SYMBOL

The problem of going from the output of a sensor to an intelligent response is sometimes called the *signal-to-symbol problem*. The name comes from the fact that sensors produce signals (such as voltage levels, current, resistance, etc.), while an action is usually based on a decision involving symbols. For example, the rule: "If grandmother is there and is smiling, approach her, otherwise go away" can be easily programmed or encoded with symbols for grandmother and smiling, but it is a lot more complicated to write the rule using raw sensor data. By using symbols, we can make information *abstract* and not sensor-specific. But getting from a sensory output (of any sensor) to such an abstract symbolic form (or encoding) of information in order to make intelligent decisions is a complex process. While it may seem obvious, this is one of the most fundamental and enduring challenges in robotics.

SENSOR PREPROCESSING

Since sensors provide signals and not symbolic descriptions of the world, they must be processed in order to extract the information the robot needs. This is usually called *sensor preprocessing* because it comes before anything else can be done in terms of using the data to make decisions and/or take action. Sensor (pre)processing can be done in different ways and draws on methods from signal processing (a branch of electrical engineering) and computation.

7.1 Levels of Processing

ELECTRONICS

Suppose that your robot has a switch sensor to detect bumping into obstacles, as described above. To figure out if the switch is open or closed, you need to measure the voltage going through the circuit. That is done with *electronics*.

SIGNAL PROCESSING

Now suppose your robot has a microphone sensor for recognizing a voice. Besides the electronic processing, it will need to separate the signal from any background noise and then compare it with one or more stored voices in order to perform recognition. That is done with *signal processing*.

Next suppose your robot has a camera for finding your grandmother in the room. Besides the electronics and signal processing, it will need to find the objects in the room, then compare them against a large database in order

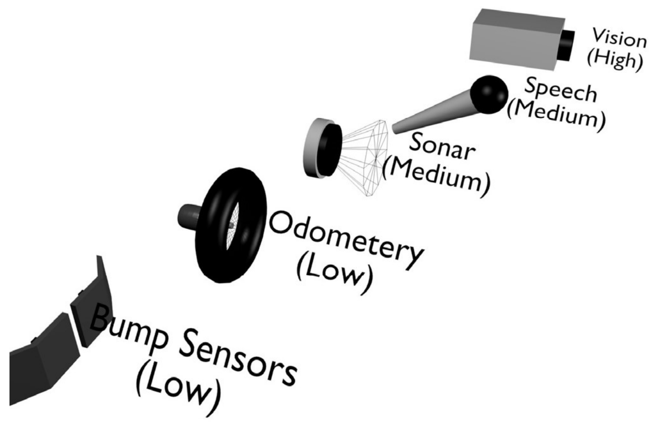

Figure 7.2 Levels of sensory processing.

COMPUTATION

to try to recognize the grandmother. That is done with *computation*.

As you can see, sensory information processing is challenging and can be computationally intensive and time consuming. Of the above processes, computation is the slowest but most general. For any specific problem, a system can be designed to solve it at a lower, and thus faster, level. For example, although computation is usually necessary for processing visual images, specialized microprocessors, so-called "vision chips" have been developed that are designed for particular vision-based tasks, such as recognizing particular faces or fruits or engine parts. They are fast but very specialized; they cannot be used to recognize anything else.

Given that a great deal of processing can be required for perception (as shown in figure 7.2), we can already see why a robot needs some type of brain. Here is what a robot needs in order to process sensory inputs:

- Analog or digital processing capabilities (i.e., a computer)
- Wires to connect everything together
- Support electronics to go with the computer
- Batteries to provide power for the whole thing.

This means that perception requires:

- Sensors (power and electronics)
- Computation (more power and electronics)
- Connectors (to connect it all).

It is generally not a good idea to separate what the robot senses, how it senses it, how it processes it, and how it uses it. If we do that, we end up with a large, bulky, and ineffective robot. Historically, perception has been studied and treated in isolation, and typically as a reconstruction problem, assuming that a robot always has to answer the second question posed above. None of these approaches has resulted in effective methods that robots can use to better sense the world and go about their business.

Instead, it is best to think about the what, why, and how of sensing as a single complete design, consisting of the following components:

- The task the robot has to perform
- The sensors best suited for the task
- The mechanical design most suited to enable the robot to get the sensory information necessary to perform the task (e.g., the body shape of the robot, the placement of the sensors, etc.).

Robotics researchers have figured out these important requirements of effective perception, and have been exploring various methods, including:

- *Action-oriented perception* (also called "active sensing"): instead of trying to reconstruct the world in order to decide what to do, the robot can use the knowledge about the task to look for particular stimuli in the environment and respond accordingly. As an example, it is very hard for robots to recognize grandmothers in general, but it is not nearly as hard to look for a particular color pattern of a grandmother's favorite dress, perhaps combined with a particular size and shape, and speed of movement.

  A clever psychologist named J. J. Gibson wrote (in 1979) about the idea that perception is naturally biased by what the animal/human needs to do (i.e., action), and is influenced by the interaction between the animal and its environment; it is not a process that retrieves absolute "truth" about the environment (i.e., reconstruction). This idea has been quite popular in action-oriented perception.

- *Expectation-based perception:* Use knowledge about the robot's environment to help guide and constrain how sensor data can be interpreted. For example, if only people can move in the given environment, we can use motion by itself to detect people. This is how simple burglar alarms work: they don't recognize burglars, or even people, just detect movement.

- *Task-driven attention:* Direct perception where more information is needed or likely to be provided. Instead of having the robot sense passively as it moves, move the robot (or at least its sensors) to sense in the direction where information is most needed or available. This seems very obvious to people, who unconsciously turn their heads to see or hear better, but most robots still use fixed cameras and microphones, missing the opportunity to perceive selectively and intelligently.

- *Perceptual classes:* Divide up (partition) the world into perceptual categories that are useful for getting the job done. That way, instead of being confronted with an unwieldy amount of information and numerous possibilities, the robot can consider manageable categories it knows how to handle. For example, instead of deciding what to do for every possible distance to an obstacle, the robot may have only three zones: too close, ok but be on the alert, and not to worry. See Chapter 14 for an example of just such a robot.

The idea that sensor function (what is being perceived) should decide sensor form (where the sensor should be, what its shape should be, if/how it should move, etc.) is employed cleverly in all biological systems. Natural evolved sensors have special geometric and mechanical properties well suited for the perceptual tasks of the creature that "wears" them. For example, flies have complex compound, faceted eyes, some birds have polarized light sensors, some bugs have horizon line sensors, humans have specially shaped ears that help us figure out where the sound is coming from, and so on. All of these, and all other biological sensors, are examples of clever mechanical designs that maximize the sensor's perceptual properties, its range and accuracy. These are very useful lessons for robot designers and programmers.

As a robot designer, you will not get the chance to make up new sensors, but you will always have the chance (and indeed the need) to design interesting ways of using the sensors you have at your disposal.

Here is an exercise: How would you detect people in an environment?

Remember the lessons from this chapter so far: Use the interaction with the world and keep in mind the task.

The obvious answer is to use a camera, but that is the least direct solution to the problem, as it involves a great deal of processing. You will learn more about this in Chapter 9. Other ways of detecting people in the environment include sensing:

- Temperature: Search for temperature ranges that correspond to human body temperature
- Movement: If everything else is static, movement means people
- Color: Look for a particular range of colors corresponding to people's skin or their clothes or uniforms
- Distance: If an otherwise open distance range becomes blocked, there is likely a moving human around.

The above are just some ways of detecting people by using sensors that are simpler than vision and require less processing. They are not perfect, but compared with vision, they are fast and easy. Often these alone, or in combination, are enough to get the task done, and even when vision sensors are available, other sensory modalities can improve their accuracy. Consider burglar alarm sensors again: they sense movement through temperature changes. While they could confuse a large dog with a human, in indoor environments canine (dog) burglars are rare, so the sensors tend to be perfectly suited to the task.

Now let's do another exercise: how would you measure distance to an object?

Here are some options:

- Ultrasound sensors provide distance measurements directly (time of flight)
- Infra red sensors can provide it through return signal intensity
- Two cameras (i.e., stereo) can be used to compute distance/depth
- A camera can compute distance/depth by using perspective (and some assumptions about the structure of the environment)
- A laser and a fixed camera can be used to triangulate distance

- A laser-based, structured light system, can overlay a grid pattern over the camera image and the processing system can use the distortions in that pattern to compute distance.

These are just some of the available means of measuring distance, and, as you saw, distance is one measure that can be used to detect other objects and people.

SENSOR FUSION

Combining multiple sensors to get better information about the world is called *sensor fusion*.

Sensor fusion is not a simple process. Consider the unavoidable fact that every sensor has some noise and inaccuracy. Combining multiple noisy and inaccurate sensors therefore results in more noise and inaccuracy and thus more uncertainty about the world. This means some clever processing has to be done to minimize the error and maximize accuracy. Furthermore, different sensors give different types of information, as you saw above in the example of detecting people. Again, clever processing is necessary to put the different types of information together in an intelligent and useful way.

As usual, nature has an excellent solution to this problem. The brain processes information from all sensory modalities - vision, touch, smell, hearing, sound - and a multitude of sensors. Eyes, ears, and nose are the obvious ones, but consider also all of your skin, the hairs on it, the strain receptors in your muscles, the stretch receptors in your stomach, and numerous others sources of body awareness (proprioception) you have at your disposal, consciously or otherwise. A great deal of our impressive brain power is involved in processing sensory information. Therefore it is not surprising that this is a challenging and important problem in robotics as well.

To Summarize

- Sensors are the robot's window into the world (through exteroception) as well as into its own body (through proprioception).

- Sensors do not provide state, but instead measure physical properties.

- Levels of sensor processing can include electronics, signal processing, and computation. Simple sensors require less processing, and thus less associated hardware and/or software.

- The what, why, and how of robot sensing, the form and the function, should be considered as a single problem.

- Nature gives us ample examples of clever sensor form and function.
- Action-oriented perception, expectation-based perception, focus of attention, perceptual classes, and sensor fusion can all be used to improve the availability and accuracy of sensor data.
- Uncertainty is a fundamental and unavoidable part of robotics.

Food For Thought

- Uncertainty is not much of a problem in computer simulations, which is why simulated robots are not very close to the real, physical ones. Can you figure out why?
- Some robotics engineers have argued that sensors are the main limiting factor in robot intelligence: if only we had more, smaller, and better sensors, we could have all kinds of amazing robots. Do you believe that is all that's missing? (Hint: If that were so, wouldn't this book be much thinner?)
- Being able to sense the self, being self-aware, is the foundation for consciousness. Scientists today still argue about what animals are conscious, and how that relates to their intelligence, because consciousness is a necessary part of higher intelligence of the kind people have. What do you think will it take to get robots to be self-aware and highly intelligent? And if some day they are both, what will their intelligence be like, similar to ours or completely different?

Looking for More?

- The Robotics Primer Workbook exercises for this chapter are found here: http://roboticsprimer.sourceforge.net/workbook/Sensors
- Jerome James Gibson (1904-1979) is considered one of the most important contributors to the field of visual perception. The ideas we discussed in the chapter come from his classic book *The Perception of the Visual World*, written in 1950, in which he put forward the idea that animals "sampled" information from their environment. In the same book he introduced the notion of "affordance", which is very important in machine vision (and

also, as it happens, in ergonomic design). To learn more, look for Gibson's book or for books on machine vision. For suggestions for those, see Chapter 9.

- You can learn about sensor fusion from the work of Prof. Robyn Murphy. In fact, after reading this book, you should consider reading her *Introduction to AI Robotics*, which covers sensor fusion and many other topics at a more advanced level.

8 *Switch on the Light* *Simple Sensors*

As we learned in Chapter 7, we can consider a sensor simple if it does not require a great deal of processing to yield useful information to the robot. In this chapter we will take a closer look at several such simple sensors, including switches, light sensors, position sensors, and potentiometers.

But first, let's consider another way in which we can categorize all sensors, both simple and complex. Sensors can be divided into two basic categories: active and passive.

8.1 Passive vs. Active Sensors

PASSIVE SENSORS
DETECTOR
ACTIVE SENSORS

Passive sensors measure a physical property from the environment. They consist of a *detector*, which perceives (detects) the property to be measured. In contrast, *active sensors* provide their own signal/stimulus (and thus typically require extra energy) and use the interaction of that signal with the environment as the property they measure. Active sensors consist of an *emitter* and a *detector*. The *emitter* produces (emits) the signal, and the detector perceives (detects) it.

EMITTER

Passive sensors can be simple or complex. In this chapter we will learn about some simple passive sensors, including switches and resistive light sensors, and in Chapter 9 we will learn about cameras, currently the most complex passive sensors. Analogously, active sensors are not necessarily complex. In this chapter we will learn about reflectance and break beam sensors, which are simple and active, and in the next chapter we will learn about ultrasound (sonar) and laser sensors, both of which are complex and active.

Remember, whether a sensor is complex is determined by the amount of

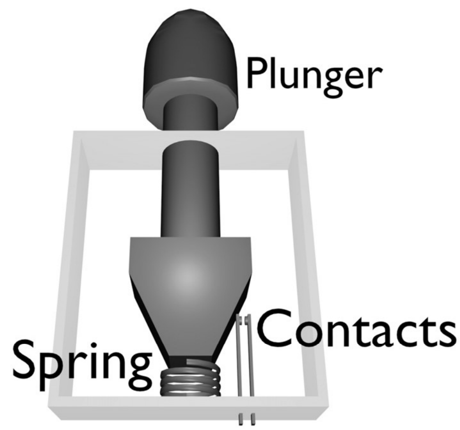

Figure 8.1 A basic switch, used for interacting with lights and robots, among other uses.

processing its data require, while whether a sensors is active is determined by its *mechanism*.

Let's start by taking a look at the simplest passive sensors: switches.

8.2 Switches

Switches, such as the one shown in figure 8.1, are perhaps the simplest sensors of all. They provide useful information at the electronics (circuit) level, since they are based on the principle of an open vs. a closed circuit. If a switch is *open*, no current can flow though the circuit; if it is *closed*, current can flow through. By measuring the amount of current flowing through the circuit, we can tell if the switch is open or closed. So, switches measure the change in current resulting from a closed circuit, which in turn results from physical contact of an object with the switch.

Depending on how you wire a switch to a circuit, it can be normally open or normally closed. Either way, the measurement of current is all you need in order to use the switch as a sensor. This simple principle is applied in a wide variety of ways to create switches, and switches are, in turn, used in a variety of clever ways for sensing, such as:

CONTACT SENSORS
- *Contact sensors* detect when the sensor has contacted another object (e.g., they trigger when a robot hits a wall or grabs an object).

LIMIT SENSORS
- *Limit sensors* detect when a mechanism has moved to the end of its range (e.g., they trigger when a gripper is wide open).

SHAFT ENCODERS
- *Shaft encoder sensors* detects how many times a motor shaft turns by having a switch click (open/close) every time the shaft turns.

You use many kinds of switches in your everyday life: light switches, computer mouse buttons, keys on keyboards (computers and electronic pianos), buttons on the phone, and others.

The simplest yet extremely useful sensor for a robot is a *bump switch*, also called a *contact switch*, which tells the robot when it has bumped into something. Knowing this, the robot can back away and get around the obstacle, or stick to it, or keep bumping it, whatever is consistent with its goals. You'll discover that even for such a simple sensor as a bump switch, there are many different ways of implementation.

As we know from numerous examples in biology, building a clever body structure around a sensor can make that sensor much more sensitive and accurate. Therefore, switches can be attached to a great variety of places and components on a robot. For instance, a switch can be attached to a large, rigid (for example, plastic) surface so that when any part of the surface contacts an object, the switch closes. This is a good way to find out if any part of a robot's front, say, or side, has hit an obstacle. Another clever way to use a switch is in the form of a whisker, as found on many animals.

> *Can you think of how to build a (simple) whisker from the principles of a switch?*

Here is one way: Attach a long wire to a switch; whenever the wire is bent enough, the switch will close and thus indicate contact. But this is not very sensitive, as the whisker has to bend quite a lot to close the switch. What can you do to fix this? Well, you can use a rigid material instead of a flexible wire for the whisker; this will make it more sensitive, and in fact more similar to the bump sensor we talked about above, for the chassis of the robot's body. But the whisker could also break if the robot does not stop. What else can you do?

A more effective way to build a whisker sensor is as follows: Use a metal (conductive) wire placed in a metal (conductive) tube. When the whisker bends, it contacts the tube, and thus closes the circuit. By adjusting the length and width of the tube, the whisker sensitivity can be tuned for the particular task and environment of the robot.

The examples above just scratch the surface of the numerous ways to cleverly design and place switches to make a robot aware of contact with objects in its environment.

Touch/contact sensors abound in biology, in much more sophisticated forms than the simple switches we have discussed so far. Whiskers and antennae are the biological inspiration for the sensors we have talked about in this section. You can consider any part of your skin to be a sensor for contact, as well as for pressure and heat, and every hair on your body as a whisker. Such abundance of sensory input is not yet available to machines; it's no wonder we are far from having truly "sensitive" robots.

8.3 Light Sensors

Besides being able to detect contact with objects, it is often useful for a robot to be able to detect areas of darkness and light in its environment. Why? Because a light can then be used to mark a special area, such as the battery recharging station or the end of a maze. With a light sensor, a robot can also find dark places and hide there.

What other uses for light sensors can you think of?

PHOTOCELL

Light sensors measure the amount of light impacting a photocell. *Photocells*, as the name indicates (*photo* means "light" in Greek), are sensitive to light, and this sensitivity is reflected in the resistance in the circuit they are attached to. The resistance of a photocell is low when it is illuminated, sensing a bright light; it is high when it is dark. In that sense, a light sensor is really a "dark" sensor. This can be made simpler and more intuitive; by simply inverting the output in the circuit, you can make it so that low means dark and high means light.

Figure 8.2 shows a photocell; the squiggly line is the photoresitive part that senses/responds to the light in the environment. Basic household night lights use the same photocells as some robots, in order to detect and respond to particular light levels. In night lights, low light causes the bulb to light up, while in the robot it may result in changing speed or direction, or some other appropriate action. Remember Braitenberg's vehicles in Chapter 2? They used this simple principle to produce all sorts of interesting robot behaviors, such as following, chasing, avoiding, and oscillating, which resulted in com-

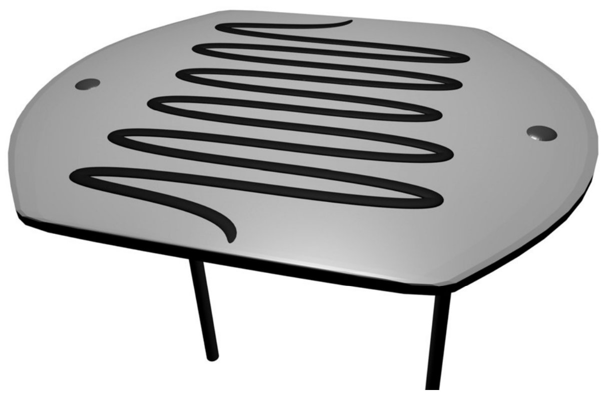

Figure 8.2 An example of a photocell.

plex interpretations by observers, that included repulsion, aggression, fear, and even love.

Light sensors are simple, but they can detect a wide range of wavelenghts, much broader than the human eye can see. For example, they can be used to detect ultraviolet and infra red light, and can be tuned to be sensitive to a particular wavelength. This is very useful for designing specialized sensors; you will see how later in this chapter.

Just as we saw with switches, light sensors can be cleverly positioned, oriented, and shielded in order to improve their accuracy and range properties. They can be used as passive or active sensors in a variety of ways, and they can measure the following properties:

- Light intensity: how light/dark it is
- Differential intensity: difference between photocells
- Break in continuity: "break beam," change/drop in intensity.

We will see examples of all of those uses in this chapter. Another property of light that can be used for sensing is polarization.

8.3.1 Polarized Light

POLARIZING FILTER

POLARIZED LIGHT

"Normal" light emanating from a light source consists of light waves that travel in all directions relative to the horizon. But if we put a *polarizing filter* in front of the light source, only the light waves with the direction of the filter will pass through it and travel away. This direction is called the "characteristic plane" of the filter; "characteristic" because it is special for that filter, and "plane" because it is planar (two-dimensional). *Polarized light* is light whose waves travel only in a particular direction, along a particular plane.

Why is this useful? Because just as we can use filters to polarize the light, we can use filters to detect light with a particular polarization. Therefore we can design *polarized light sensors*.

In fact, we don't have to limit ourselves to just one filter and thus one characteristic plane of light. We can combine polarizing filters. How does that work? Consider a light source (such as a lightbulb) covered with a polarizing filter. The resulting polarized light is only in the characteristic plane of the filter. What happens if we put another, identical filter in the path of the resulting polarized light? All of the light gets through. But what if we use a filter with a 90-degree angle of polarization instead? Now none of the light gets through, because none of it is in the characteristic plane of the second filter.

By playing around with photocells and filters and their arrangement, you can use polarized light to make specialized sensors that cleverly manipulate what and how much light is detected. These are active sensors, since they consist not only of a photocell (for detecting the light level) but also of one (or more) light source(s) (for emitting the light) and one (or more) filter(s) for polarizing the light. The general idea is that the filtering happens between the emitter and the receiver; exactly where – whether closer to the emitter, to the receiver, or both – depends on the robot, its environment, and its task.

Polarized light sensing exists in nature as well; many insects and birds use polarized light to navigate more effectively.

8.3.2 Reflective Optosensors

LIGHT-EMITTING DIODE (LED)

Reflective optosensors, as you can guess by the name, operate on the principle of reflected light. These are active sensors; they consist of an emitter and a detector. The emitter is usually made with a *light-emitting diode (LED)*, and the detector is usually a photodiode/phototransistor. You can learn about

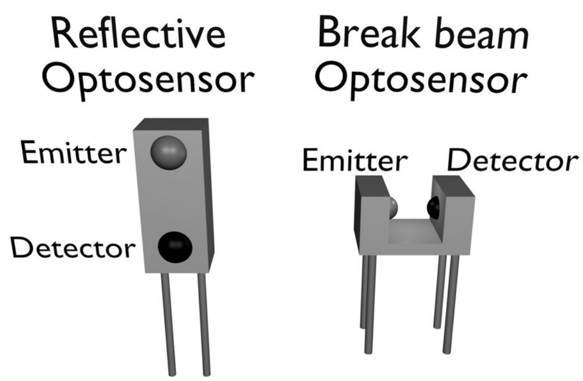

Figure 8.3 The two types of optosensor configuration: reflectance (on the left) and break beam (on the right).

the details of what these electronic components consist of as suggested in the Looking for More section at the end of this chapter.

Reflective optosensors do not use the same technology as the resistive photocells we learned about in the last section. Resistive photocells are nice and simple, but their resistive properties make them slow, because change in resistance takes time. Photodiodes and phototransistors, on the other hand, are much faster, and are therefore preferred for use in robotics.

There are two basic ways in which reflective optosensors can be arranged, based on the relative positions of the emitter and the detector:

REFLECTANCE SENSORS

1. *Reflectance sensors*: The emitter and the detector are side by side, separated by a barrier; the presence of an object is detected when the light reflects from it and back into the detector.

BREAK BEAM SENSORS

2. *Break beam sensors*: The emitter and the detector face one another; the presence of an object is detected if the beam of light between the emitter and the detector is interrupted or broken (thus the name) .

8.3.3 Reflectance Sensors

What can you do with this simple idea of measuring light reflectance/reflectivity?

Here are just a few of the many useful things you can do:

- Detect the presence of an object: Is there a wall in front of the robot?
- Detect the distance to an object: How far is the object in front of the robot?
- Detect some surface feature: Find the line on the floor (or the wall) and follow it.
- Decode a bar code: Recognize a coded object/room/location/beacon.
- Track the rotations of a wheel: Use shaft encoding; we will learn more about this later in this chapter.

Although the general idea of using reflected light is simple, the exact properties of the process are anything but. For example, light reflectivity is affected by the color, texture (smoothness or roughness), and other properties of the surface it hits. A light-colored surface reflects light better than a dark-colored one; a matte (non-shiny) black surface reflects little if any light, and so is invisible to a reflective sensor. Therefore, it may be harder and less reliable to detect dark objects using reflectance than light ones. In the case of determining the distance to an object, the same principle makes lighter objects that are far away seem closer than dark objects that are close.

This gives you part of the idea why sensing in the physical world is challenging. No sensor is perfect, and all sensors are prone to error and noise (interference from the environment). *Therefore, even though we have useful sensors, we cannot have complete and completely accurate information.* These intrinsic, unavoidable limitations of sensors are rooted in the physical properties of the sensor mechanism, and have an impact on the resulting accuracy or the sensor. Therefore, these fundamental limitations are part of *uncertainty in robotics*.

Let's talk about sensor noise as it relates to reflective light sensors. A light sensor has to operate in the presence of the light existing in the environment, which is called *ambient light*. The reflectance sensor must ignore ambient light in order to be sensitive only to its own emitter's reflected light. This is difficult if the wavelength of the ambient light is the same as that of the emitter. The sensor's mechanism has to somehow subtract or cancel out the

AMBIENT LIGHT

ambient light from the detector reading, so that it can accurately measure only the light coming from the emitter, which is what it needs.

> *How is that done? How does the detector know the amount of ambient light?*

It has to sense it. The ambient light level is measured by taking a sensor reading with the emitter *off*. To measure only the emitter's reflected light, the detector takes two (or more, for higher accuracy) readings of the sensor level, one with the emitter *on* and one with it *off*. When one is subtracted from the other (and the signs are adjusted properly so we don't end up with negative light), the difference produces the amount of the emitter's light reflected back to the sensor. This is an example of sensor calibration.

CALIBRATION

Calibration is the process of adjusting a mechanism so as to maximize its performance (accuracy, range, etc.). Sensors require calibration, some only initially, and others continually, in order to operate effectively. Calibration may be performed by the designer, the user, or the sensor mechanism itself.

Going back to our reflective optosensor, we've just seen how to calibrate it in order to subtract ambient noise. But if we do this calibration only once, when the sensor is first used, we may find that the sensor becomes very inaccurate over time. Why is that? Because ambient light levels change during the course of the day, lights get turned on and off, bright or dark objects may be present in the area, and so on. Therefore, if the environment can change, the sensor has to be calibrated repeatedly to stay accurate and useful. There is no time to sit back and relax when it comes to sensing.

As we mentioned above, ambient light is a problem if it is the same wavelength as the emitter light, and therefore interferes with it and is hard to cancel out. The general way to avoid interference is to encode the signal from the sensor in a way that the detector can easily separate it from ambient light. One way to do this is through the use of polarization filters, as we saw earlier. Another way is through adjusting the wavelength of the emitted light. Here is how.

8.3.4 Infra Red Light

VISIBLE LIGHT

Visible light is light in the frequency band of the electromagnetic spectrum that human eyes can perceive.[1] Infra red (IR) light has a wavelength different

1. If you remember the electromagnetic spectrum from physics, that helps in this chapter, but is not necessary.

from visible light and is not in the visible spectrum. IR sensors are a type of light sensors which function in the infra red part of the frequency spectrum. They are used in the same ways that visible light sensors are: as reflectance sensors or as break beams.

Both of those uses are forms of active sensors. IRs are not usually used as passive sensors in robotics. Why is that? Because robots do not usually need to detect ambient IR. But other types of sensors, such as IR goggles, better known as "night vision" glasses, do detect ambient IR passively. They collect and enhance the light in the IR spectrum and transform it into the visible spectrum. When tuned to the frequency of human body heat, they can be (and are) used to detect human movement in the dark.

IR is very useful in robotics because it can be easily modulated, and in that way made less prone to interference. Such modulated IR can also be used for communication (i.e., for transmitting messages), which is how IR modems work. So let's learn about modulation.

8.3.5 Modulation and Demodulation of Light

MODULATED LIGHT
DEMODULATOR

Light is *modulated* by rapidly turning the emitter on and off, pulsing it. The resulting pulsed signal is then detected by a *demodulator*, a mechanism that is tuned to the particular frequency of the modulation, so it can can be decoded. The detector has to sense several "on" flashes in a row in order for the demodulator to determine its frequency and decode it.

Strobe light is a type of modulated visible light, and given how hard it is to look at a strobe light, you can see (so to speaks) why visible light is not usually used in modulated form: it's too hard on the eyes. However, modulated IR is commonly used, since IR is not in the visible spectrum. Most household remote controls are based on modulated IR, including the fought-over TV channel changer.

8.3.6 Break Beam Sensors

You probably have a good intuitive idea of how a break beam sensor works. In general, any pair of compatible emitter-detector devices can be used to produce break beam sensors, including:

- An incandescent flashlight bulb and a photocell
- Red LEDs and visible-light-sensitive phototransistors
- Infra red IR emitters and detectors.

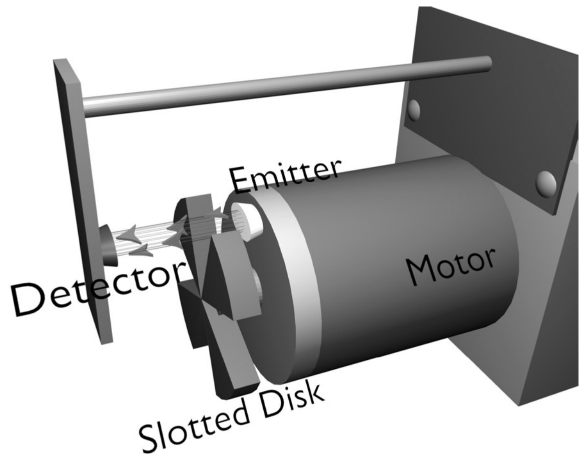

Figure 8.4 Break beam shaft encoder mechanism.

Where have you seen break beam sensors? Images from movies might come to mind, with crisscrossed laser beams and clever burglars making their way through them. More realistically, one of the most common uses of break beam sensing is not in plain sight, because it is packed inside motor mechanisms and used to keep track of shaft rotation. Here is how it works.

8.3.7 Shaft Encoders

SHAFT ENCODER

Shaft encoders measure the angular rotation of a shaft or an axle. They provide position and/or velocity information about the shaft they are attached to. For example, the speedometer measures how fast the wheels of the car are turning, and the odometer measures the number of rotations of the wheels. Both speedometers and odometers use shaft encoding as the underlying sensing mechanism.

In order to detect a turn, or a part of a turn, we have to somehow mark the thing that is turning. This is usually done by attaching a round notched disk to the shaft. If the shaft encoder uses a switch, the switch clicks every time the shaft completes a full rotation. More commonly, a light sensor is used: a light emitter is placed on one side of the disk, and a detector on the other, in a break beam configuration. As the disk spins, the light from the emitter

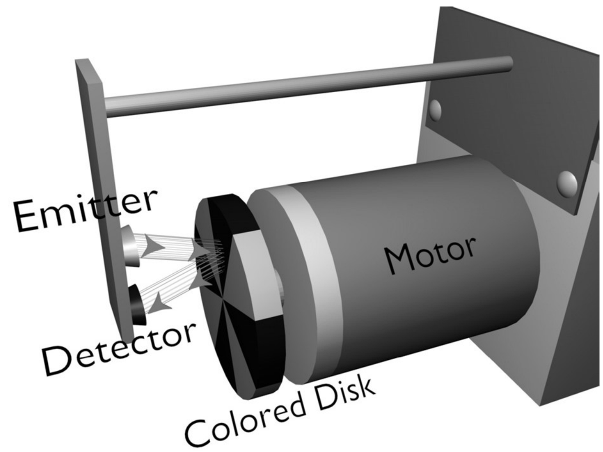

Figure 8.5 Reflectance-based shaft encoder mechanism.

reaches the detector only when the notched part of the disk passes in front of it.

If there is only one notch in the disk, then every time the notch passes between the emitter and the detector, this means the disk has completed one full rotation. This is useful, but it allows for measuring with only a low level of precision. If any noise or error is present, one or more turns might be missed, and the encoder will thus be quite inaccurate.

To make the encoder more accurate as well as more precise, many notches are cut into the disk. The break beam principle is still the same: whenever the light gets through, it is sensed by the detector and counted. Figure 8.4 shows what the mechanism looks like. You can see that it is important to have a fast sensor if the shaft turns very quickly. That is why a resistive sensor would not be appropriate; it is comparatively slow, while an optosensor works well for this propose, as we discussed earlier in this chapter. (If you forgot, just remember that optosensors use light, which travels faster than anything else.)

An alternative to cutting notches in the disk is to paint the disk with wedges of alternating, contrasting colors. The best color choices are black (absorbing, nonreflecting) and white (highly reflecting), as they provide the highest contrast and the best reflective properties. But in this case, since there are no notches in the disk, how does the break beam sensor work? It doesn't. This is not a break beam sensor, but a reflectance sensor. Instead of putting the sensor in the break beam configuration, in this case the emitter and the detector are placed on the same side of the disk, side by side, in a reflectance configuration. Figure 8.5 shows what that looks like.

Regardless of whether the shaft encoding sensor is a break beam or a reflectance sensor, the detector will output a wave function (a sequence of the signal being on and off, up and down) of the sensed light intensity of the emitter. This output is then processed, using signal processing, with hardware or a simple processor, to calculate the position and the speed by counting the peaks of the waves.

We can use encoders in at least two ways to measure the speed of a robot:

- Encode and measure the speed of a driven wheel
- Encode and measure the speed of a passive wheel (caster) that is dragged by the robot.

Why might we want to use the second of the above? One use is for robots that have legs instead of wheels. For example, the designers of Genghis, the six-legged robot mentioned in Chapter 5, had it drag a little wheel in some of the experiments, to measure the distance it had traveled, especially when it was learning how to walk (see Chapter 21 to learn about that).

We can combine the position and velocity information the encoder provides to have the robot do more sophisticated things, such as move in a straight line or turn at by exact angle. However, doing such precise movements is quite difficult, because wheels tend to slip and slide, and there is usually some backlash in the gearing mechanism (recall Chapter 4). Shaft encoders can provide feedback to correct some of the errors, but having some error remain is unavoidable. There is no perfect sensor, since uncertainty is a fact of life.

So far, we've talked about detecting position and velocity, but did not talk about direction of rotation. Suppose the robot's wheel suddenly changes the direction of rotation; it would be useful for the robot to be aware of it. If the change is intentional, the encoder can tell the robot how accurate the turn was; and if the change is unintentional, the encoder may be the first or only way the robot will know it has turned.

QUADRATURE SHAFT ENCODING

The mechanism for detecting and measuring direction of rotation is called *quadrature shaft encoding*. One place where you might have used it is inside an old-fashioned computer mouse, the kind that has a ball inside (not the more recent, optical type). In such a mouse, the direction of rotation of the ball, and thus the direction of the movement of the mouse itself, is determined through quadrature shaft encoding.

Quadrature shaft encoding is an elaboration of the basic break beam idea: instead of using only one sensor, use two. The two encoders are aligned so that their two inputs coming from the detectors are 90 degrees (one quarter of a full circle, thus the name quadrature) out of phase. By comparing the outputs of the two encoders at each time step with the output of the previous time step, we can tell if there is a direction change. Since they are out of phase, only one of them can change its state (i.e., go from on to off or vice versa) at a time. Which one does it determines in which direction the shaft is rotating. Whenever a shaft is moving in one direction, a counter is incremented in that encoder, and when it turns in the opposite direction, the counter is decremented, thus keeping track of the overall position of the mechanism.

In robotics, quadrature shaft encoding is used in robot arms with complex joints, such as the ball-and-socket joints we discussed in Chapter 4. It is also used in *Cartesian robots*, which are similar in principle to Cartesian plotter printers, and are usually employed for high-precision assembly tasks. In those, an arm moves back and forth along an axis or gear.

CARTESIAN ROBOTS

We have seen that switches and light sensors can be used in a variety of different ways, and in some cases in the same ways (as in shaft encoding). Let's talk about one more type of simple sensor.

8.4 Resistive Position Sensors

As we just learned, photocells are resistive devices that sense resistance in response to the light. Resistance of a material, it turns out, can change in response to other physical properties besides light. One such property is tension: the resistance of some devices increases as they are bent. These passive "bend sensors" were originally developed for video game controls, but have since been adopted for other uses as well.

As you might expect, repeated bending fatigues and eventually wears out the sensor. Not surprisingly, bend sensors are much less robust than light sensors, although the two use the same underlying principle of responding to resistance.

By the way, your muscles are full of biological bend sensors. These are proprioceptive sensors that help the body be aware of its position and the work it is doing.

8.4.1 Potentiometers

Potentiometers, popularly known as "pots," are commonly used for manual tuning of analog devices: they are behind every knob or slider you use on a stereo system, volume control, or light dimmer. These days, it's getting harder to find such knobs, since most devices are *digital*, meaning using discrete (from the Latin *discretus* meaning "separate") values. The word "digital" comes from the Latin *digitus* meaning finger, and most electronic devices today are tuned digitally, by pushing buttons (with fingers), and thus they use switches rather than pots. But not so long ago, tuning to radio stations and adjusting volume, among other things, were done with pots.

DIGITAL

Potentiometers are resistive sensors; turning the knob or pushing a slider effectively alters the resistance of the sensor. The basic design of potentiometers involves a tab that slides along a slot with fixed ends. As the tab is moved, the resistance between it and each of the ends of the slot is altered, but the resistance between the two ends remains fixed.

In robotics, potentiometers are used to tune the sensitivity of sliding and rotating mechanisms, as well as to adjust the properties of other sensors. For example, a distance sensor on a robot may have a potentiometer attached to it which allows you to tune the distance and/or sensitivity of that sensor manually.

You might be thinking that these simple sensors are not so very simple after all. And you are right; sensor mechanisms can get pretty complex pretty fast, but they are not much compared with biological sensors. Still, keep in mind that while these are nontrivial physical mechanisms, the resulting sensor data are quite simple and require little processing. In the next chapter we'll move on to some complex sensors that spew out much more data and give us more processing work to do.

To Summarize

- Sensors can be classified into active and passive, simple and complex.
- Switches may be the simplest sensors, but they provide plenty of variety and have a plethora of uses, including detecting contact, limits, and turning of a shaft.
- Light sensors come in a variety of forms, frequencies, and uses, includ-

ing simple photocells, reflective sensors, polarized light and infra red (IR) sensors.

- Modulation of light makes it easier to deal with ambient light and to design special-purpose sensors.
- There are various ways to set up a break beam sensor, but they are most commonly used inside motor shaft encoders.
- Resistive position sensors can detect bending and are used in a variety of analog tuning devices.

Food for Thought

- Why might you prefer a passive to an active sensor?
- Are potentiometers active or passive sensors?
- Our stomach muscles have stretch receptors, which let our brains know how stretched our stomach are, and keep us from eating endlessly. What robot sensors would you say are most similar to such stretch receptors? Are they similar in form (mechanism of how they detect) or function (what they detect)? Why might stretch receptors be useful to robots, even without stomachs and eating?

Looking for More?

- The Robotics Primer Workbook exercises for this chapter are found here: http://roboticsprimer.sourceforge.net/workbook/Sensors
- The all-around best text for learning about electronics as well as debugging hardware problems is *The Art of Electronics* by Paul Horowitz and Winfield Hill (Cambridge University Press). Every robot lab worth its beans has at least one well-worn copy.
- *Sensors for Mobile Robots: Theory and Applications* by H. R. (Bart) Everett is a comprehensive and reader-friendly textbook covering all of the sensors we have overviewed in this chapter, and quite a few more.

9 *Sonars, Lasers, and Cameras Complex Sensors*

Congratulations, you've graduated from simple to complex sensors, and now you are in for it! The sensors we have seen so far, passive or active, do not require a great deal of processing or computation in order to provide information directly useful to a robot. However, the information they provide is itself simple and limited: light levels, presence or absence of objects, distance to objects, and so on. Not only are complex processing and computation not necessary, but they would be of little use. Not so with complex sensors. In contrast to simple ones, complex sensors provide much (much, much) more information, that can yield quite a bit more useful fodder for the robot, but also requires sophisticated processing.

In this chapter, we will learn about ultrasound, laser, and vision sensors, some of the most commonly used complex sensors in robotics. But don't assume that they are the only complex sensors available; there are others (such as radar, laser radar, GPS, etc.), and new ones are always being developed.

9.1 Ultrasonic or Sonar Sensing

ULTRASOUND

Ultrasound literally means "beyond sound," from the Latin *ultra* for "beyond" (used here in the same way as in "ultraviolet" and "ultraconservative") and refers to a range of frequencies of sound that are beyond human hearing. It is also called *sonar*, from so(und) na(vigation and) r(anging). Figure 9.1 shows a mobile robot equipped with sonar sensors.

SONAR

ECHOLOCATION

The process of finding your (or a robot's) location based on sonar is called *echolocation*. Echolocation works just the way the name sounds (no pun intended): sound bounces off objects and forms echoes that are used to find one's place in the environment. That's the basic principle. But before we get into the details, let's first consider some examples.

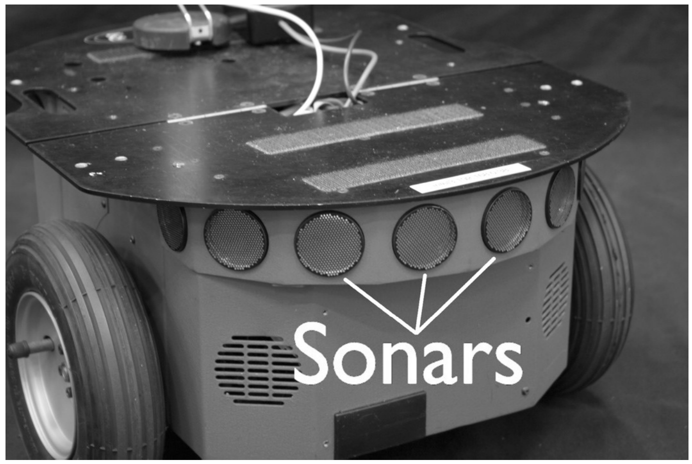

Figure 9.1 A mobile robot with sonar sensors.

The principle of echolocation comes from nature, where it is used by several species of animals. Bats are the most famous for using sophisticated echolocation; cave bats that dwell in nearly total darkness do not use vision (it would not do them much good), but rely entirely on ultrasound. They emit and detect different frequencies of ultrasound, which allows them to fly effectively in very crowded caves that have complicated structures and are packed with hundreds of other flying and hanging bats. They do all this very quickly and without any collisions. Besides flying around in the dark, bats also use echolocation to catch tiny insects and find mates. Dolphins are another species known for sophisticated echolocation. What used to be secret research is now standard in aquarium shows: blindfolded dolphins can find small fish and swim through hoops and mazes by using echolocation.

As usual, biological sensors are vastly more sophisticated than current artificial ones (also called "synthetic," because they have been "synthesized," not because they are made out of synthetics); bat and dolphin sonars are much more complex than artificial/synthetic sonars used in robotics and other applications. Still, synthetic sonars are quite useful, as you will see.

So how do they work?

TIME-OF-FLIGHT

Artificial ultrasound sensors, or sonars, are based on the *time-of-flight* principle, meaning they measure the time it takes something (in this case sound) to travel ("fly"). Sonars are active sensors consisting of an emitter and a detector. The emitter produces a chirp or ping of ultrasound frequency. The sound travels away from the source and, if it encounters a barrier, bounces off it (i.e., reflects from it), and perhaps returns to the receiver (microphone). If there is no barrier, the sound does not return; the sound wave weakens (attenuates) with distance and eventually breaks down.

If the sound does come back, the amount of time it takes for it to return can be used to calculate the distance between the emitter and the barrier that the sound encountered. Here is how it works: a timer is started when the chirp is emitted, and is stopped when the reflected sound returns. The resulting time is then multiplied by the speed of sound and divided by two. Why? Because the sound traveled to the barrier and back, and we are only trying to determine how far away the barrier is, its one-way distance.

SPEED OF SOUND

This computation is very simple, and relies only on knowing the *speed of sound*, which is a constant that varies only slightly due to ambient temperature. At room temperature, sound travels 1.12 feet per millisecond. Another way to put it is that sound takes 0.89 milliseconds to travel the distance of 1 foot. This is a useful constant to remember.

The hardware for sonar sensing most commonly used in robotics is the Polaroid Ultrasound Sensor, initially designed for instant cameras. (Instant cameras were popular before digital cameras were invented, since they provided instant photos; otherwise people had to wait for film to be developed, which took at least a day, unless they a personal film development lab.) The physical sensor is a round transducer, approximately 1 inch in diameter, that emits the chirp/ping and receives the sound (echo) that comes back. A *transducer* is a device that transforms one form of energy into another. In the case of the Polaroid (or other ultrasound) transducers, mechanical energy is converted into sound as the membrane of the transducer flexes to produce a ping that sends out a sound wave that is inaudible to humans. You can actually hear most robot sonars clicking but what you hear is the movement of the emitter (the membrane), not the sound being sent out.

TRANSDUCER

The hardware (electronics) of ultrasound sensors involves relatively high power, because significant current is needed for emitting each ping. Importantly, the amount of current required is much larger than what computer processors use. This is just one of many practical examples showing why it is a good idea to separate the power electronics of a robot's sensing and actuation mechanisms from those of its controller processor. Otherwise, the robot's brain might have to literally slow down in order for the body to sense or move.

The Polaroid ultrasound sensor emits sound that spreads from a 30-degree sound cone in all directions, and at about 32 feet attenuate to a point that they do not return to the receiver, giving the sensor a 32-foot range. The range of an ultrasound sensor is determined by the signal strength of the emitter, which is designed based on the intended use of the sensor. For robots (and for instant cameras, as it happens), the range of 32 feet is typically sufficient, especially for indoor environments. Some other uses of sonar require quite a bit less or more range, as you will see in the next section.

9.1.1 Sonar Before and Beyond Robotics

Ultrasound is used in a variety of applications besides (and before) robotics, from checking on babies in utero (inside the mother's womb) to detecting objects and attackers in submarines. When sonar is used to look into people's bodies, the result is called a sonogram, echogram, or ultrasonogram, coming from the Greek *gram* meaning "letter" and referring to writing and drawing. Sound travels well through air and water, and since the human body consists largely of water (over 90 percent by weight), ultrasound is a good technology for seeing what's going on inside. One of the most common uses of sonograms is for range (distance) measurements, just like its use in robotics. However, the Polaroid and other sonars used in robotics operate at about 50 KHz, while the medical ones operate in the higher frequency range, around

HERTZ (HZ)

3.5 to 7 MHz. Just in case you did not know or you forgot, a *Hertz (Hz)* is a unit of frequency. One Hertz means once per second, a Kilohertz (KHz) is 1000 Hz and a Megahertz (MHz) is 1,000,000 Hz.

Simply sensing the distances between the emitter and the environment may be sufficient for many robotics applications, but in medical sonar imaging, much more complex postprocessing is involved in order to create a composite image of the body part. This image is not static, but is updated in real time, allowing for what looks like real-time video of, for example, the beating heart of a baby in utero.

Since sound travels well through water, while vision is almost useless in the underwater environment, it is no surprise that sonar is the favored sensor for underwater navigation, specifically for helping submarines detect and avoid any unexpected obstacles, such as other submarines. You have no doubt seen images of large, looming, menacing, and constantly pinging submarines in the movies. Sonars used in submarines have long ranges, through the use of stronger signal intensity and narrower cones. As in depth sounders, these sonars send an intense beam of sound into the ocean (or any body of water), and wait for it to return, in order to see how deep the water is or, in other words, how far the nearest object/surface is. As you can imagine, these sonars need to reach much farther than 32 feet; thus their use of a narrower cone and stronger signal intensity. While such uses of ultrasound are inaudible and very useful to people, they are audible and, it turns out, dangerous to marine animals, such as whales and dolphins. High-strength and long-range ultrasound emissions have been shown to confuse whales and cause them to beach and die. The exact causes of this behavior are not yet understood, but the power of ultrasound should not be underestimated. When properly controlled and directed, it can be used to break up objects, such as kidney stones. Between the body and the ocean, there are other, more mundane uses of sonar. They include automated tape measures, height measures, and burglar alarms.

The principle of time-of-flight underlies all uses of sonar as a ranging and imaging device. In almost all applications, multiple sensor units are employed for increased coverage and accuracy. Most robots using sonars are equipped with several, usually a full ring, covering a cross-section of the robot's body.

Here is an easy question: what is the smallest number of standard Polaroid sonar sensors needed to cover the cross-section of a robot?

As we will see later, a dozen sonars cover the circle around a robot (unless the robot has a very wide berth, in which case more are used). Whether a dozen or more are used, they cannot all be pinged/emitted at the same time. Can you guess why?

Sonar sensors, Polaroid or others, are inexpensive and easy to incorporate into robot hardware. If only sonars always returned accurate distance readings! But they do not, since things are never that simple in the physical world. Sonar data can be tricky, and here is the reason why.

9.1.2 Specular Reflection

As we saw, sonar sensing is based on the emitted sound wave reflecting from surfaces and returning to the receiver. But the sound wave does not necessarily bounce off the nearest surface and come right back, as we might hope it would. Instead, the direction of reflection depends on several factors, including the properties of the surface (how smooth it is) and the incident angle of the sound beam and the surface (how sharp it is).

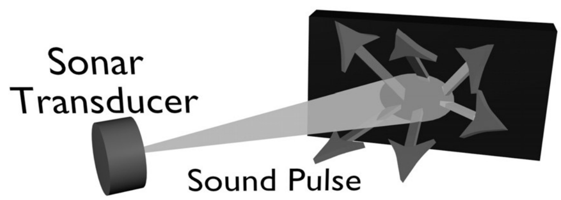

Figure 9.2 Specular reflection of an ultrasound signal.

SPECULAR REFLECTION

A major disadvantage of ultrasound sensing is its susceptibility to specular reflection. *Specular reflection* is the reflection from the outer surface of the object; this means the sound wave that travels from the emitter bounces off multiple surfaces in the environment before returning to the detector. This is likely if the surface it encounters is smooth and if the angle between the beam and the surface is small. The smaller the angle, the higher the probability that the sound will merely graze the surface and bounce off, thus not returning to the emitter but moving on to other surfaces (and potentially even more grazing bounces) before returning to the detector, if it returns at all. This bouncing sound generates a false faraway reading, one that is much longer than the straight-line distance between the robot (its sonar sensor) and the surface. The smoother the surface is, the more likely the sound is to bounce off. In contrast, rough surfaces produce more irregular reflections, which are more likely to return to the emitter. Think of it this way: as the sound hits a rough surface, it scatters, bouncing back at various angles relative to the various facets and grooves and features on the surface. At least some of the reflections are likely to go back to the emitter, and thus provide a rather accurate distance measure. In contrast, as the sound hits a uniformly smooth surface (a specular one), it may graze or bounce off it uniformly in a direction away from the detector. In figure 9.2 you can see an illustration of specular reflection. Specularity is a property of light as well as of sound, which adds to the challenges of machine vision; we'll worry about that in the next section.

In the worst case of specular reflection, the sound bounces around the environment and does not return to the detector, thus fooling the sensor into detecting no object/barrier at all, or one that is far away instead of nearby. This could happen in a room full of mirrors, as at a carnival, or full of glass cases, as at a museum. A crude but effective way to combat specular reflection is to alter the environment in which the sonar-based robot has to navigate by making the surfaces less reflective. How can we do that? Well, we can rub all smooth surfaces with sandpaper, or we can use rough wallpaper, or we can put little wood slats in the walls, basically anything to introduce features on the surfaces. Fortunately, since sonar beams are relatively focused, especially at short ranges, only the surfaces that the cone of the sonars is most likely to sense need be altered, not entire walls. For example, in research labs, roboticists have lined experimental areas with bands of corrugated cardboard, because its ridges have much better sonar reflectance properties than the smooth walls. In general, altering the environment to suit the robot is not a great idea and it is often not even possible (such as under the ocean or in space).

How else can we get around the specular reflection problem?

One solution is to use phased arrays of sensors to gain more accuracy. The basic idea is to use multiple sensors covering the same physical area, but activated out of phase. This is exactly what is used in automatic tape measure devices: these contraptions, when pointed at an object, give the distance to that object, by using multiple carefully arranged and timed sonar sensors. You can think of this as a hardware-level solution to the problem.

Can you think of some software/processing-level computational solutions?

Remember that the problem is that long sonar readings can be very inaccurate, as they may result from false reflected readings from nearby objects rather than accurate faraway readings. We can use this fact to embed some intelligence into the robot, in order to make it accept short readings but do more processing on the long ones. One idea is to keep a history of past readings, and see if they get longer or shorter over time in a reasonable, continuous way. If they do, the robot can trust them, and if not, the robot assumes they are due to specular effects. This approach is effective in some environments but not in all, and is especially challenging in unknown structures. After all, the environment may have *discontinuities*, sudden and large changes in its features. Those cannot be anticipated by the robot, so it is left having to trust its sensor readings, however unpredicatable they may be.

DISCONTINUITY

Besides postprocessing, the robot can also use action-oriented perception, which we discussed in Chapter 7. Whenever it receives an unexpected long sonar reading that seems strange/unlikely, it can turn and/or move so as to change the angle between its sonar sensor and the environment, and then take another reading, or many more readings, to maximize accuracy. This is a good example of a general principle:

> *Using action to improve sensory information is a powerful method of dealing with uncertainty in robotics.*

Ultrasound sensors have been successfully used for very sophisticated robotics applications, including mapping complex outdoor terrain and indoor structures. Sonars remain a very popular, affordable ranging sensor choice in mobile robotics.

9.2 Laser Sensing

Sonars would be great sensors if they were not so susceptible to specular reflection. Fortunately, there is a sensor that largely avoids the problem, but at some cost and trade-offs: the laser.

LASER

Lasers emit highly amplified and coherent radiation at one or more frequencies. The radiation may be in the visible spectrum or not, depending on the application. For example, when laser sensors are used as burglar detectors, they are typically not visible. In the movies, we are often shown visible laser grids, but in reality making them visible makes the burglar's job easier By the way, what principle are such sensors based on? They are break beam sensors: when a burglar (or a roaming cat, say) breaks the laser beam, the alarm goes off. When laser sensors are used for range measurements, they also are typically not visible, since it is usually not desirable to have a visible and distracting beam of light scanning the environment as the robot moves around.

Laser range sensors can be used through the time-of-flight principle, just like sonars. You can immediately guess that they are much faster, since the speed of light is quite a bit greater than the speed of sound. This actually causes a bit of a problem when lasers are used for measuring short distances:

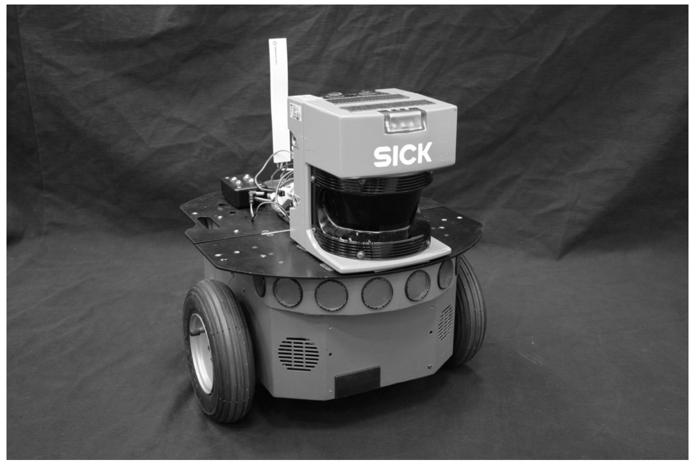

Figure 9.3 A mobile robot sporting a laser sensor. The laser is the thing that resembles a coffee maker and is labeled with the name of the manufacturer, SICK.

the light travels so fast that it comes back more quickly than it can be measured. The time intervals for short distances are on the order of nanoseconds and can't be measured with currently available electronics. As an alternative, phase-shift measurements, rather than time-of-flight, are used to compute the distance.

Robots that use lasers for indoor navigation and mapping, such as the one shown in figure 9.3, typically operate in relatively short-range environments by laser standards. Therefore, lasers used on mobile robots use phase-shift rather than time-of-flight. In both cases the processing is typically performed within the laser sensor itself, which is equipped with electronics, and so the sensor package produces clean range measurements for the robot to use.

Lasers are different from sonars in many other ways stemming from the differences in the physical properties of sound and light. Lasers involve higher-power electronics, which means they are larger and more expensive. They are also much (much, much) more accurate. For example, a popular laser sensor, the SICK LMS200 scanning laser rangefinder, has a range of 8m and a 180-degree field of view. The range and bearing of objects can be determined to within 5mm and 0.5 degrees. The laser can also be used in a long-range mode (up to 80m), which results in a reduction in accuracy of only about 10cm.

Another key distinction is that the emitted laser light is projected in a beam rather than a cone; the spot is small, about 3mm in diameter. Because lasers use light, they can take many more measurements than sonar can, thereby providing data with a higher resolution. *Resolution* refers to the process of separating or breaking something into its constituent parts. When something has high resolution, it means it has many parts. The more parts there are to the whole, the more information there is. That's why "high res" is a good thing.

RESOLUTION

So what is the resolution of the SICK sensor? Well, it makes 361 readings over a 180-degree arc at a rate of 10Hz. The true rate is much higher (again, because of the speed of light and the speed of today's electronics), but the resulting rate is imposed by the serial link for getting the data out. Ironically, the serial link is the bottleneck, but it's good enough, since a real robot does not need laser data at any rate higher than what this sensor provides via the port; it could not physically react any faster anyway.

Laser sensors sound like a dream come true, don't they? They have high resolution and high accuracy, and do not suffer nearly as much from specular effects. Of course they are not totally immune to specularities, since they are light-based and light is a wave that reflects, but specularities are not much of a problem due to the resolution of the sensor, especially compared with ultrasound sensing. So what is the downside?

Well, first of all, they are large, about the size of an electric coffee maker (and sort of look like one, too); it takes some space to house all those high-power electronics. Next, and most impressively, laser sensors are very expensive. A single SICK laser costs two orders of magnitude more than a Polaroid sonar sensor. Fortunately, price is bound to come down over time, and packaging is bound to become smaller. However, the high resolution of the laser range sensor has its downside. While the narrow beam is ideal for detecting the distance of a particular point, to cover an area, the laser needs to sweep or scan. The planar (2D) SICK laser mentioned above scans horizontally over a 180-degree range, providing a highly accurate slice of distance readings. However, if the robot needs to know about distances outside of the plane, more lasers and/or more readings are needed. This is quite doable, since lasers can scan quickly, but it does take additional sensing and processing time. 3D laser scanners also exist, but are even larger and more expensive. They are ideal, however, for accurately mapping out distances to objects, and thus the space around the sensor (or robot using the sensor).

In mobile robotics, simple sensors are usually most popular, and lasers, even planar ones, are not sufficiently affordable or portable for some applications. Can you guess which ones? For example, any that include small robots, or robots interacting with children or people who might look in the direction of the laser, which is too high-power to be considered completely safe.

Let's go back to the laser grid for detecting intruders and burglars in movies. Projecting a visible grid of laser light on the environment is part of another approach to sensing. The distortions of the grid represent the shapes of objects in the environment. But to detect the pattern and its distortions, we need another sensor: a camera. That brings us to the most complex and versatile sensory modality of all: vision.

9.3 Visual Sensing

Seeing requires a visual sensor, something akin to biological eyes. Cameras are the closest thing to natural eyes that we have available among synthetic sensors. Needless to say, any/all biological eyes are more complex than any cameras we have today. But to be fair, seeing is not done by eyes only, but largely by the brain. Eyes provide the information about the incoming light patterns, and the brain processes that information in complex ways to answer questions such as "Where are my car keys?" and make decisions such as "Stop right now; there they are, between the sofa cushions!"

Although cameras and computers are a far cry from, and much less complex than, biological eyes and brains, you will soon see that the information provided by cameras is not simple in the least, and that the associated processing is some of the most complex in robotics. The research field that deals with vision in machines, including robots, is called, appropriately, *machine vision*. Robots have particular perceptual needs related to their tasks and environments, and so some parts of machine vision research are relevant and useful for robotics, while others have proven not to be, even if they are very useful for other applications. Therefore, machine vision and robotics are two separate fields of research with some overlap of interests, problems, and uses.

MACHINE VISION

Traditionally, machine vision has concerned itself with answering the questions "What is that?", "Who is that?" and "Where is that?" To answer these, the approach has been to reconstruct what the world was like when the camera took its picture, in order to understand the picture, so that the question can be answered. We already talked about vision as reconstruction in Chap-

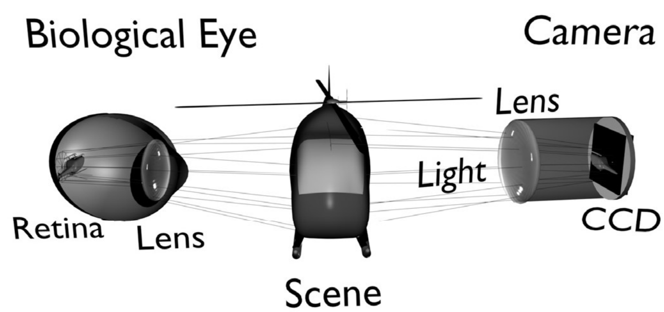

Figure 9.4 Components of biological and synthetic vision systems.

ter 7.

After decades of research in machine vision, we know that visual reconstruction is an extremely difficult problem. Fortunately for robotics, it is not the problem that robots really need to solve. Instead, robots are typically concerned with acting so as to achieve their goals or, put more simply, with doing the right thing. Instead of answering the machine vision questions above, they need to answer action-related questions of the type: "Should I keep going or turn or stop?", "Should I grab or let go?", "Where do I turn?", and so on. It is usually not necessary to reconstruct the world in order to answer those questions. To begin to understand how robots use vision, let's first see (so to speak) how cameras, the basic visual sensors, do their work.

9.3.1 Cameras

BIOMIMETIC

Cameras are *biomimetic* meaning they imitate biology, in that they work somewhat the way eyes do. But as usual, synthetic and natural vision sensors are quite different. Figure 9.4 shows a simple comparison, marking the key components of the vision system.

Here are those components. Light, scattered from objects in the environ-

SCENE
IRIS
IMAGE PLANE
PHOTOSENSITIVE
EARLY VISION
HIGH-LEVEL VISION

ment (which are collectively called the *scene*), goes through an opening (the *iris*, which is in the simplest case a pinhole, but usually is a lens) and hits the image plane. The *image plane* corresponds to the retina of the biological eye, which is attached to numerous light-sensitive (*photosensitive*) elements called rods and cones. These in turn are attached to nerves that perform *early vision*, the first stages of visual image processing, and then pass information on to other parts of the brain to perform *high-level vision* processing, everything else that is done with the visual input. As we have mentioned before, a very large portion of the human (and other animal) brain activity is dedicated to visual processing, so this is a highly complex endeavor. Instead of rods and cones, film cameras use silver halides on photographic film, and digital cameras use silicon circuits in charge-coupled devices (CCD). In all cases, some information about the incoming light (e.g., its intensity, color) is detected by the photosensitive elements on the image plane.

LENS
FOCUS
DEPTH OF FIELD

In machine vision, the computer needs to make sense out of the information on the image plane. If the camera is very simple, and uses a tiny pinhole, then some computation is required to determine the projection of the objects from the environment onto the image plane (note that they will be inverted). If a *lens* is involved (as in vertebrate eyes and real cameras), then more light can get in, but at the price of being focused; only objects a particular range of distances from the lens will be in *focus*. This range of distances is called the camera's *depth of field*.

PIXEL

The image plane is usually subdivided into equal parts called *pixels*, typically arranged in a rectangular grid. As we saw in Chapter 7, in a typical camera there are 512×512 pixels on the image plane. For comparison, there are 120×10^6 rods and 6×10^6 cones in the human eye.

IMAGE
SURFACE PATCH

The projection of the scene on the image plane is called, not surprisingly, the *image*. The brightness of each pixel in the image is proportional to the amount of light that was reflected into the camera by the part of the object or surface that projects to that pixel, called the *surface patch*.

DIFFUSE REFLECTION

Since you already know that reflectance properties vary, you can guess that the particular reflectance properties of the surface patch, along with the number and positions of light sources in the environment, and the amount of light reflected from other objects in the scene onto the surface patch all have a strong impact on what the pixel brightness value ends up being. All those influences that affect the brightness of the patch can be lumped into two kinds of reflections: specular (off the surface, as we saw before) and diffuse. *Diffuse reflection* consists of the light that penetrates into the object, is absorbed, and then comes back out. To correctly model light reflection and reconstruct the scene, all these properties are necessary. No wonder visual reconstruction is hard to do. It's a good thing that robots usually don't need to do it.

TIME SERIES
FRAME
FRAME GRABBER
DIGITAL IMAGE
IMAGE PROCESSING

We need to step back for just a second here and remember that the camera, just like the human eye, observes the world continually. This means it captures a video, a series of images over time. Processing any *time series* information over time is pretty complicated. In the case of machine vision, each individual snapshot in time is called a *frame*, and getting frames out of a time series is not simple. In fact, it involves specialized hardware, called a *frame grabber*, a device that captures a single frame from a camera's analog video signal and stores it as a *digital image*. Now we are ready to proceed with the next step of visual processing, called *image processing*.

9.3.2 Edge Detection

EDGE DETECTION

The typical first step (early vision) in image processing is to perform *edge detection*, to find all the edges in the image.

How do we recognize edges? What are edges, really?

EDGE

In machine vision, an *edge* is defined as a curve in the image plane across which there is a significant change in the brightness. More intuitively, finding edges is about finding sharp changes in pixel brightness. Finding changes mathematically is done by taking derivatives. (Calculus has its uses.) A simple approach to finding edges, then, is to differentiate the image and look for areas where the magnitude of the derivative is large, which indicates that the difference in the local brightness values is also large, likely due to an edge. This does find edges, but it also finds all sorts of other things that produce large changes, such as shadows and noise. Since it is impossible to distinguish "real" edges from those resulting from shadows simply by looking at pixel brightness/intensity in the image, some other method has to be used to do better.

What about noise – how do we deal with it?

Unlike shadows, noise produces sudden and spurious intensity changes that do not have any meaningful structure. This is actually a good thing, since noise appears as peaks in the intensities, and those peaks can be taken away by a process called "smoothing."

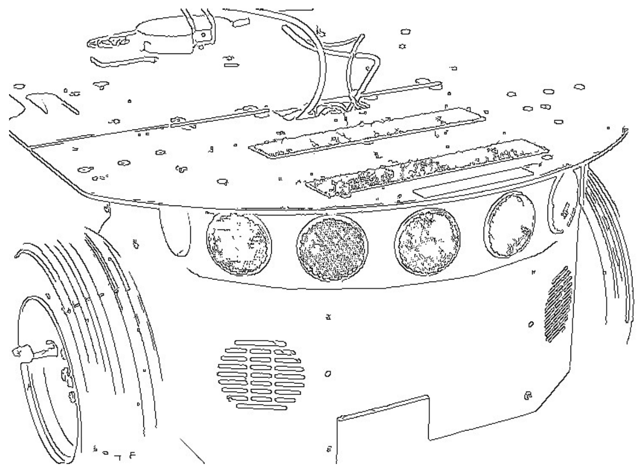

Figure 9.5 A camera image that has been processed through edge detection and then segmentation.

How is smoothing done automatically?

SMOOTHING

CONVOLUTION

Again, math comes to the rescue. To perform *smoothing*, we apply a mathematical procedure called convolution, which finds and eliminates the isolated peaks. *Convolution* applies a filter to the image; this is called *convolving* the image. This type of mathematical filter is really the same, in principle, as a physical filter, in that the idea is to filter out the unwanted things (in this case the spurious peaks that come from visual noise) and let through the good stuff (in this case the real edges). The process of finding real edges involves convolving the image with many filters with different orientations. Think back to the previous chapter, when we talked about polarized light filters. Those were physical filters, and here we are talking about mathematical filters, but both have the same function: to separate out a particular part of the signal. In the case of the polarized filters, we were looking for a particular frequency of light; in the case of edge detection filters, we are looking for intensities with particular orientations.

Edge detection used to be a very popular difficult problem in machine vision, and many algorithms were written for it, tested, and published. Eventually, researchers developed the best algorithms possible for the task, so now edge detection is not considered an interesting research problem, but it is still a very real practical problem in machine vision. Whenever one has to do edge detection, there is an "off-the-shelf" algorithm to use, and in some cases specialized hardware, such as edge detection processors, can be used to speed up the visual processing.

SEGMENTATION

Once we have edges, the next thing to do is try to find objects among all those edges. *Segmentation* is the process of dividing or organizing the image into parts that correspond to continuous objects. Figure 9.5 shows an image that has been processed by edge detection and segmentation.

But how do we know which lines correspond to which objects? And what makes an object?

The next few sections describe several cues and approaches we can use to detect objects.

9.3.3 Model-Based Vision

Suppose that your robot has a bunch of line drawings of chairs in its memory. Whenever it sees an object in the environment, it performs edge detection, which produces something like a very bad line drawing, and compares the outcome with those stored drawings to see if any of them match what it saw in the environment, which would indicate it saw a chair.

MODEL-BASED VISION

Those stored drawings are called *models* and the process is called *model-based vision*. It is a part of a philosophy about how the brain may recognize familiar objects, or how we might get robots to do it effectively. Model-based vision uses models of objects, and some prior information or knowledge about those objects, represented and stored in a way that can be used for comparison and recognition.

Models can be stored in a variety of forms; line drawings are just one form. Even though 2D line drawings are relatively simple and intuitive, using model matching for recognizing them is still a complex process. Here is why: it is not enough just to compare what the robot sees (even after edge detection) with the stored model. The robot may be looking at the object from any angle and any distance. Therefore, to compare it effectively with the model, it has to properly scale (change the size of) the model, and rotate the model to try different orientations. Also, since any edge in the image may correspond to any edge in the model, all those combinations have to be considered and evaluated. Finally, since the robot does not know what it is looking at, it needs to consider all models it has stored in its memory, unless it can somehow cleverly eliminate some of them as unlikely. All of this is computationally intensive, taking plenty of memory (to store the models) and processor power (to do scaling, rotation, and comparisons).

Models can vary from simple 2D line drawings to weirdly processed, mathematically distorted images that combine all the various views of the object to be recognized in a mathematical way. For example, some very successful face recognition systems use only a few views of the person's face, and then do some interesting math to produce a model that can then recognize that person from many more points of view. Face recognition is a very popular problem in machine vision, and model-based approaches seem very well suited for it, since faces do have repeatable features, such as two eyes, a nose, and a mouth, with relatively constant ratios of distances in between those features (for most people, at least). Nevertheless, we are still far from being able to reliably and efficiently recognize a particular "face in the crowd," whether it be your mother who has come to pick you up at the airport or a known criminal trying to flee the country.

Face recognition is one of the important things that your brain performs very effectively, and is in fact fine-tuned through evolution to do very well. This is so because humans are such *social animals*, for whom it is very important to know who is who in order to establish and maintain social order. Imagine if you could not recognize faces, what would your life be like? There is a neurological disorder that produces that very deficit in some people; it is called *prosopagnosia*, from the Greek *prosop* for "face" and *agnosia* for "not knowing." It is rare and, so far, incurable.

PROSOPAGNOSIA

Face recognition is likely to be useful in robotics, especially for robots that are going to interact with people as their companions, helpers, nurses, coaches, teachers, or pets. But there are many faces to learn and recognize, so this will remain an interesting research challenge, not just for machine vision but also for the area of robotics that deals with *human-robot interaction* and which we will discuss in Chapter 22.

HUMAN-ROBOT INTERACTION

9.3.4 Motion Vision

Visual systems are often, though not always, attached to things that move (such as people and robots, for example). The movement of the body and the camera on it makes vision processing more challenging and *motion vision* is a set of machine vision approaches that uses motion to facilitate visual processing.

MOTION VISION

If the vision system is trying to recognize static objects, it can take advantage of its own motion. By looking at an image at two consecutive time steps, and moving the camera in between, continuous solid objects (at least those that obey physical laws we know about) will move as one, and their brightness properties will be unchanged. Therefore, if we subtract two consecutive images from one another, what we get is the "movement" between the two, while the objects stay the same. Notice that this depends on knowing exactly how we moved the camera relative to the scene (the direction and distance of the movement), and on not having anything else moving in the scene.

As we mentioned in Chapter 7, by using active perception a robot can use movement to get a better view of something. However, moving about to go places and moving to see better are not necessarily the same, and may be in conflict in some cases. Therefore, a mobile robot using vision has to make some clever decisions about how it moves, and has to subtract its own movement from the visual image in order to see what it can see.

If other objects also move in the environment, such as other robots and people, the vision problem becomes much harder. We will not go into any more details about it here, but you can pursue it in the readings at the end of this chapter.

9.3.5 Stereo Vision

BINOCULAR VISION
STEREO VISION

Everything we have discussed so far about vision has assumed a single camera. Yet in nature, creatures have two eyes, giving them *binocular vision*. The main advantage of having two eyes is the ability to see *in stereo*. *Stereo vision*, formally called *binocular stereopsis*, is the ability to use the combined points of view from the two eyes or camers to reconstruct three-dimensional solid objects and to perceive depth. The term *stereo* comes from the Greek *stereos* meaning "solid," and so it applies to any process of reconstructing the solid from multiple signals.

In stereo vision, just as in motion vision (but without having to actually move), we get two images, which we can subtract from one another, as long as we know how the two cameras or eyes are positioned relative to each other. The human brain "knows" how the eyes are positioned, and similarly we, as robot designers, have control over how the cameras on the robot are positioned as well and can reconstruct depth from the two images. So if you can afford two cameras, you can get depth perception and reconstruct solid objects.

This is the way 3D glasses, which make images in movies look solid, work. In normal movies, the images come from a single projector and both of your eyes see the same image. In 3D movies, however, there are two different images from two different projectors. That's why when you try to watch a 3D movie without the special glasses, it looks blurry. The two images do not come together, on the screen or in your brain. But when you put on the glasses, the two come together in your brain and look 3D. How? The special glasses let only one of the projected images into each of your eyes, and your brain fuses the images from the two eyes, as it does for everything you look at. This does not work with normal movies because the images in your two eyes are the same, and when they are fused, they still look the same. But the 3D movie images are different, by clever design, and when brought together, they look better.

You might wonder how it is that those simple glasses with colored foil lenses manage to make just one of the projected images go into each of your eyes. That's simple: one of the projected images is blue (or green) and one is red, and the glasses let in one color each. Pretty easy and neat, isn't it? The first 3D movie using this method was made in 1922, and you can still see the same technology used for TV, movies, and books. Today there are more sophisticated ways of achieving the same result, with better color and sharpness. These involve the use of something else you already know about: polarized light, which we learned about in Chapter 8. The movies are shown from two projectors that use different polarization, and the 3D glasses use polarizing filters instead of the simpler color filters.

The ability to perceive in 3D using stereo is fundamental to realistic human/animal vision, and so it is involved in a variety of applications from video games to teleoperated surgery. If you lose the use of one of your eyes, you will lose the ability to see depth and 3D objects. To see how important depth perception is, try catching a ball with one eye closed. It's hard, but not impossible. That is because your brain compensates for the loss of depth for a little while. If you put a patch over one eye for several hours or longer, you will start to trip and fall and reach incorrectly toward objects. Fortunately, once you take the patch off, your brain will readjust to seeing in 3D. But if you lose an eye permanently, it won't. So eyes and cameras are to be treated with care.

9.3.6 Texture, Shading, Contours

What other properties of the image can we find and use to help in object detection?

Consider texture. Sandpaper looks quite a bit different from fur, which looks quite a bit different from feathers, which look quite a bit different from a smooth mirror, and so on, because all reflect the light in very different ways. Surface patches that have uniform texture have consistent and almost identical brightness in the image, so we can assume they come from the same object. By extracting and combining patches with uniform and consistent texture, we can get a hint about what parts of the image may belong to the same object in the scene.

Somewhat similarly, shading, contours, and object shape can also be used to help simplify vision. In fact, anything at all that can be reliably extracted from a visual image has been used to help deal with the object recognition problem. This is true not only for machines but also (and first) for biological vision systems, so let's consider those now.

9.3.7 Biological Vision

The brain does an excellent job of quickly extracting the information we need from the scene. We use *model-based vision* to recognize objects and people we know. Without it, we find it hard to recognize entirely unexpected objects or novel ones, or to orient ourselves, as in the typical example of waking up and not knowing where you are. Biological model-based vision is of course different from machine vision, and it is still poorly understood, but it works remarkably well, as you can tell when you effortlessly recognize a face in the crowd or find a lost object in a pile of other stuff.

We use *motion vision* in a variety of ways in order to better understand the world around us, as well as to be able to move around while looking and not having it all result in a big blur. The latter is done through the *vestibular ocular reflex (VOR*, in which your eyes stay fixed even though your head is moving, in order to stabilize the image. (Go ahead and try it, move your head around as you read the rest of this paragraph.) There has been a great deal of research on VOR in neuroscience and machine vision and robotics, since that ability would be very useful for robots, but it is not quite simple to implement.

VESTIBULAR OCULAR REFLEX (VOR)

We are "hard-wired" to be sensitive to movement at the periphery of our field of view, as well as to looming objects, because they both indicate potential danger. Like all carnivores, we have *stereo vision*, because it helps to find and track prey. In contrast, herbivores have eyes on the sides of their heads, pointing in different directions, which are effective at scanning for (carnivorous) predators, but whose images do not overlap or get fused together in the way carnivore stereo vision works.

We are very good at recognizing shadows, textures, contours, and various other shapes. In a famous experiment performed by a scientist named Johansson in the 1970s, a few dots of light were attached to people's clothes and the people were videotaped as they moved in the dark, so only the movement of the dots was visible. Any person watching the dots could immediately tell that they were attached to moving humans, even if only very few light dots were used. This tells us that our brains are wired to recognize human motion, even with very little information. Incidentally, this depends on seeing the dots/people from the side. If the view is from the top, we are not able to readily recognize the activity. This is because our brains are not wired to observe and recognize from a top-down view. Birds' brains probably are, so somebody should do that experiment.

We have gone from machine vision to human vision to bird vision, so let's return to robot vision now.

9.3.8 Vision for Robots

Robot vision has more stringent requirements than some other applications of machine vision, and only slightly less demanding requirements than biological vision. Robot vision needs to inform the robot about important things: if it's about to fall down the stairs, if there is a human around to help/track/avoid, if it has finished its job, and so on. Since vision processing can be a very complex problem, responding quickly to the demands of the real world based on vision information is very difficult. It is not only impractical to try to perform all the above steps of image processing before the robot gets run over by a truck or falls down the stairs, but fortunately it may be unnecessary. There are good ways of simplifying the problem. Here are some of them:

1. Use color; look for specifically and uniquely colored objects, and recognize them that way (such as stop signs, human skin, etc.).

BLOB TRACKING

SALIENT

2. Use the combination of color and movement; this is called color *blob tracking* and is quite popular in mobile robotics. By marking important objects (people, other robots, doors, etc.) with *salient* (meaning "noticeable," "attention-getting"), or at least recognizable colors, and using movement to track them, robots can effectively get their work done without having to actually recognize objects.

3. Use a small image plane; instead of a full 512 $\times$ 512 pixel array, we can

reduce our view to much less, for example, just a line (as is used in linear CCD cameras). Of course there is much less information in such a reduced image, but if we are clever and know what to expect, we can process what we see quickly and usefully.

4. Combine other, simpler and faster sensors with vision. For example, IR cameras isolate people by using body temperature, after which vision can be applied to try to recognize the person. Grippers allow us to touch and move objects to help the camera get a better view. Possibilities are endless.

5. Use knowledge about the environment; if the robot is driving on a road marked with white or yellow lines, it can look specifically for those lines in the appropriate places in the image. This greatly simplifies following a road, and is in fact how the first, and still some of the fastest, robot road and highway driving is done.

Those and many other clever techniques are used in robot vision to make it possible for robots to see what they need to see quickly enough for doing their task.

Consider the task of autonomous or at least semiautonomous driving. This robotics problem is gaining popularity with the auto industry, as a potential means of decreasing the number of auto accidents. Automakers would be glad to have cars that can make sure the driver does not swerve off the road or into oncoming traffic. But in that task environment, everything is moving very quickly, and there is no time for slow vision processing. This is in fact a very exciting area of machine vision and robotics research. In 2006, several robot cars (regular cars and vans with robotic control of the steering) managed to drive completely autonomously from Los Angeles to Las Vegas, a very long trip. The cars used vision and laser sensing to obtain the information needed for performing the task. The next challenge being pursued is to do something similar in urban environments.

Complex sensors imply complex processing, so they should be used selectively, for tasks where they are required or truly useful. As we have said before (and will say again):

> *To design an effective robot, it is necessary to have a good match between the robot's sensors, task, and environment.*

To Summarize

- Sensor complexity is based on the amount of processing the data require. Sensors may also have complex mechanisms, but that is not what we are as concerned with in robotics.
- Ultrasound (sonar) sensing uses the time-of-flight principle to measure the distance between the transducer and the nearest object(s).
- Ultrasound sensing is relatively high-power and is sensitive to specular reflections.
- Ultrasound is used not only by robots and other machines (from submarines to medical imaging equipment) but also by animals (dolphins, whales).
- Lasers are used in ways similar to sonars, but are much faster and more accurate, as well as much more expensive.
- Vision is the most complex and sophisticated sensory modality, both in biology and in robotics. It requires by far the most processing and provides by far the most useful information.
- Machine vision has traditionally concerned itself with questions of recognition such as "Who is that?" and "What is that?", while robot vision has concerned itself with questions related to action, such as "Where do I go?" and "Can I grab that?"
- Object recognition is a complex problem. Fortunately, it can often be avoided in robot vision.
- Motion vision, stereo vision, model-based vision, active vision, and other strategies are employed to simplify the vision problem.

Food for Thought

- What is the speed of sound in metric units?
- How much greater is the speed of light than the speed of sound? What does this tell you about sensors that use one or the other?
- What happens when multiple robots need to work together and all have sonar sensors? How might you deal with their sensor interference? In Chapter 20 we will learn about coordinating teams of robots.

- Besides using time-of-flight, the other way to use sonars is to employ the Doppler shift. This involves examining the shift in frequency between the sent and reflected sound waves. By examining this shift, one can very accurately estimate the velocity of an object. In medical applications, sonars are used in this way to measure blood flow, among other things. Why don't we use this in robotics?

- Since two eyes are much better than one, are three eyes much better, or even any better, than two?

Looking for More?

- Check out *Directed Sonar Sensing for Mobile Robot Navigation* and other work by Hugh Durrant-Whyte (Australian Center for Field Robotics) and John Leonard (MIT) on complex sonar processing for navigation, in the lab, outdoors, and even underwater.

- You can learn more about 3D glasses here:
http://science.howstuffworks.com/3-d-glasses2.htm.

- If you to discover the nitty gritty of the mathematics behind robot vision, read *Robot Vision* by Berthold Klaus Paul Horn. A good book on machine vision to check out is *Computer Vision* by Dana Ballard and Christopher Brown. There are quite a few other books on machine/robot/computer vision out there, so do a search on the Internet and browse.