

7 Machine Vision

Introduction

Machine vision (other names include computer vision and artificial vision) is an important sensor technology with potential applications in many industrial operations. Many of the current applications of machine vision are in inspection; however, it is anticipated that vision technology will play an increasingly significant role in the future of robotics.

Vision systems designed to be utilized with robot or manufacturing systems must meet two important criteria which currently limit the influx of vision systems to the manufacturing community. The first of these criteria is the need for a relatively low-cost vision system. The second criterion is the need for relatively rapid response time needed for robot or manufacturing applications, which should be typically a fraction of a second.

Nevertheless, there has been a significant influx of vision systems into the manufacturing world. The systems are used to perform tasks which include selecting parts that are randomly oriented from a bin or conveyer, parts identification, and limited inspection. These capabilities are selectively used in traditional applications to reduce the cost of part and tool fixturing, and to allow the robot program to test for and adapt to limited variations in the environment.

Advances in vision technology for robotics are expected to broaden the capabilities of robotic vision systems to allow for vision-based guidance of the robot arm, complex inspection for close dimensional tolerances, and improved recognition and part location capabilities. These will result from the constantly reducing cost of computational capability, increased speed, and new and better algorithms currently being developed.

Computer vision is a complex and multidisciplinary field and is still in its early stages of development. Advances in vision technology and related disciplines are expected within the next decade which will permit applications not only in manufacturing, but also in photointerpretation, warehousing, robotic operations in hazardous environments, autonomous navigation, cartography, and medical image analysis.

The purpose of this chapter is to explore the field of machine vision as a fundamental sensor technology in robotics.

7.1 INTRODUCTION TO MACHINE VISION

Machine vision is concerned with the sensing of vision data and its interpretation by a computer. The typical vision system consists of the camera and digitizing hardware, a digital computer, and hardware and software necessary to interface them. This interface hardware and software is often referred to as a preprocessor. The operation of the vision system consists of three functions:

1. Sensing and digitizing image data
2. Image processing and analysis
3. Application

The relationships between the three functions are illustrated in the diagram of Fig. 7.1.

The sensing and digitizing functions involve the input of vision data by means of a camera focused on the scene of interest. Special lighting techniques are frequently used to obtain an image of sufficient contrast for later processing. The image viewed by the camera is typically digitized and stored in computer memory. The digital image is called a *frame of vision data,* and is frequently captured by a hardware device called a *frame grabber*. These devices are capable of digitizing images at the rate of 30 frames per second. The frames consist of a matrix of data representing projections of the scene sensed by the camera. The elements of the matrix are called *picture elements,* or *pixels*. The number of pixels are determined by a sampling process performed on each image frame. A single pixel is the projection of a small portion of the scene which reduces that portion to a single value. The value is measure of the light intensity for that element of the scene. Each pixel intensity is converted into a digital value. (We are ignoring the additional complexities involved in the operation of a color video camera.) We will examine the details of machine vision sensing in Sec. 7.2.

The digitized image matrix for each frame is stored and then subjected to image processing and analysis functions for data reduction and interpretation of the image. These steps are required in order to permit the real-time application of vision analysis required in robotic applications. Typically, an image frame will be thresholded to produce a binary image, and then various feature measurements will further reduce the data representation of the image. This data reduction can change the representation of a frame from several hundred thousand bytes of raw image data to several hundred bytes of feature value data. The resultant feature data can be analyzed in the available time for action by the robot system.

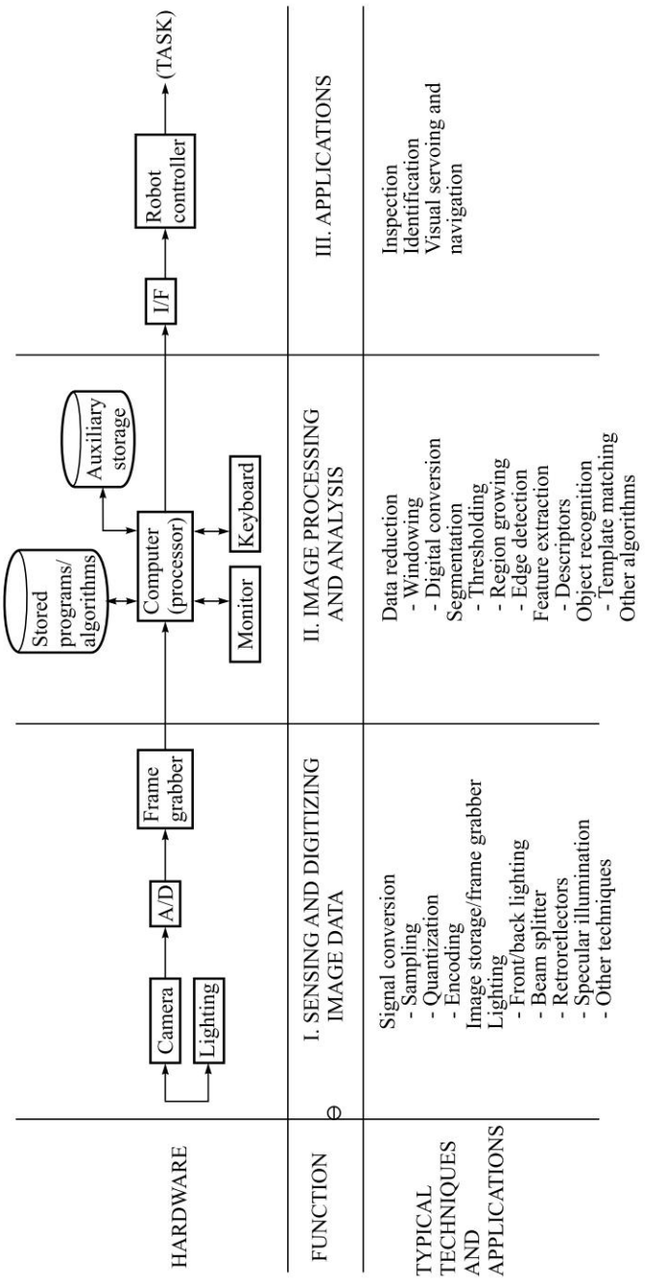


Fig. 7.1 *Functions of a machine vision system.*

Various techniques to compute the feature values can be programmed into the computer to obtain feature descriptors of the image which are matched against previously computed values stored in the computer. These descriptors include shape

and size characteristics that can be readily calculated from the thresholded image matrix. Some of the important techniques used in the vision system image processing and analysis will be discussed in Sec. 7.3.

To accomplish image processing and analysis, the vision system frequently must be trained. In training, information is obtained on prototype objects and stored as computer models. The information gathered during training consists of features such as the area of the object, its perimeter length, major and minor diameters, and similar features. During subsequent operation of the system, feature values computed on unknown objects viewed by the camera are compared with the computer models to determine if a match has occurred, Sec. 7.4 will discuss training of a vision system.

The third function of a machine vision system is the applications function.

The current applications of machine vision in robotics include inspection, part identification, location, and orientation. Research is ongoing in advanced applications of machine vision for use in complex inspection, guidance, and navigation. In Sec. 7.5, these applications will be discussed.

Vision systems can be classified in a number of ways. One obvious classification is whether the system deals with a two-dimensional or three-dimensional model of the scene. Some vision applications require only a two-dimensional analysis. Examples of two-dimensional vision problems include checking the dimensions of a part or verifying the presence of components on a subassembly.

Many two-dimensional vision systems can operate on a binary image which is the result of a simple thresholding technique. This is based on an assumed high contrast between the object (s) and the background. The desired contrast can often be accomplished by using a controlled lighting system.

Three-dimensional vision systems may require special lighting techniques and more sophisticated image processing algorithms to analyze the image. Some systems require two cameras in order to achieve a stereoscopic view of the scene, while other three-dimensional systems rely on the use of structured light and optical triangulation techniques with a single camera. An example of a structured light system is one that projects a controlled band of light across the object. The light band is distorted according to the three-dimensional shape of the object. The vision system sees the distorted band and utilizes triangulation to deduce the shape.

Another way of classifying vision systems is according to the number of gray levels (light intensity levels) used to characterize the image. In a binary image the gray level values are divided into either of two categories, black or white. Other systems permit the classification of each pixel's gray level into various levels, the range of which is called a gray scale.

7.2 THE SENSING AND DIGITIZING FUNCTION IN MACHINE VISION

Our description of the typical machine vision system in the preceding section identified three functions: sensing and digitizing, image processing and analysis, and application. This and the following sections in the chapter will elaborate on these functions. The present section is concerned with the sensing and digitizing aspects of machine vision.

Image sensing requires some type of image formation device such as a camera and a digitizer which stores a video frame in the computer memory. We divide the sensing and digitizing functions into several steps. The initial step involves capturing the image of the scene with the vision camera. The image consists of relative light intensities corresponding to the various portions of the scene. These light intensities are continuous analog values which must be sampled and converted into a digital form.

The second step, digitizing, is achieved by an analog-to-digital (A/D) converter. The A/D converter is either a part of a digital video camera or the front end of a frame grabber. The choice is dependent on the type of hardware in the system. The frame grabber, representing the third step, is an image storage and computation device which stores a given pixel array. The frame grabber can vary in capability from one which simply stores an image to significant computation capability. In the more powerful frame grabbers, thresholding, windowing, and histogram modification calculations can be carried out under computer control. The stored image is then subsequently processed and analyzed by the combination of the frame grabber and the vision controller.

7.2.1 Imaging Devices

There are a variety of commercial imaging devices available. Camera technologies available include the older black-and-white vidicon camera, and the newer, second-generation, solid state cameras. Solid state cameras used for robot vision include charge-coupled devices (CCD), charge injection devices (CID), and silicon bipolar sensor cameras. For our purposes, we review two such devices in this subsection, the vidicon camera and the charge-coupled device (CCD).

Figure 7.2 illustrates the vidicon camera. In the operation of this system, the lens forms an image on the glass faceplate of the camera. The faceplate has an inner surface which is coated with two layers of material. The first layer consists of a transparent signal electrode film deposited on the faceplate of the inner surface. The second layer is, a thin photosensitive material deposited over the conducting film. The photosensitive layer consists of a high density of small areas. These areas are similar to the pixels mentioned previously. Each area generates a decreasing electrical resistance in response to increasing illumination. A charge is created in each small area upon illumination. An electrical charge pattern is thus generated corresponding to the image formed on the faceplate. The charge accumulated for an area is a function of the intensity of impinging light over a specified time.

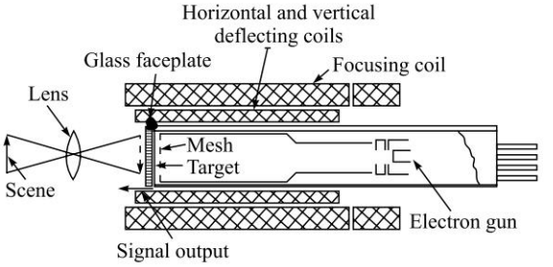


Fig. 7.2 *Cross section of a vidicon tube and its associated deflection and focusing tube. (Reprinted with permission of McGraw-Hill, Inc. [10])*

Once a light sensitive charge is built up, this charge is read out to produce a video signal. This is accomplished by scanning the photosensitive layer by an electron beam. The scanning is controlled by a deflection coil mounted along the length of the tube. For an accumulated positive charge the electron beam deposits enough electrons to neutralize the charge. An equal number of electrons flow to cause a current at the video signal electrode. The magnitude of the signal is proportional to the light intensity and the amount of time with which an area is scanned. The current is then directed through a load resistor which develops a signal voltage which is further amplified and analyzed. Raster scanning eliminates the need to consider the time at each area by making the scan time the same for all areas. Only the intensity of the impinging light is considered. In the United States, the entire faceplate is scanned approximately 30 frames per second. The European standard is 25 frames per second. Raster scanning is typically done by scanning the electron beam from left to right and top to bottom. The process is such that the system is designed to start the integration with zero-accumulated charge. For the fixed scan time, the charge accumulated is proportional to the intensity of that portion of the image being considered. The output of the camera is a continuous voltage signal for each line scanned. The voltage signal for each scan line is subsequently sampled and quantized resulting in a series of sampled voltages being stored in digital memory. This analog-to-digital conversion process for the complete screen (horizontal and vertical) results in a two-dimensional array of picture elements (pixels). Typically, a single pixel is quantized to between six and eight bits by the A/D converter.

Another approach to obtaining a digitized image is by use of the charge coupled device (CCD). In this technology, the image is projected by a video camera onto the CCD which detects, stores, and reads out the accumulated charge generated by the light on each portion of the image. Light detection occurs through the absorption of light on a photoconductive substrate (e.g., silicon). Charges accumulate under positive control electrodes in isolated wells due to voltages applied to the central electrodes. Each isolated well represents one pixel and can be transferred to output storage registers by varying the voltages on the metal control electrodes. This is illustrated in Figs. 7.3(*a*) and (*b*).

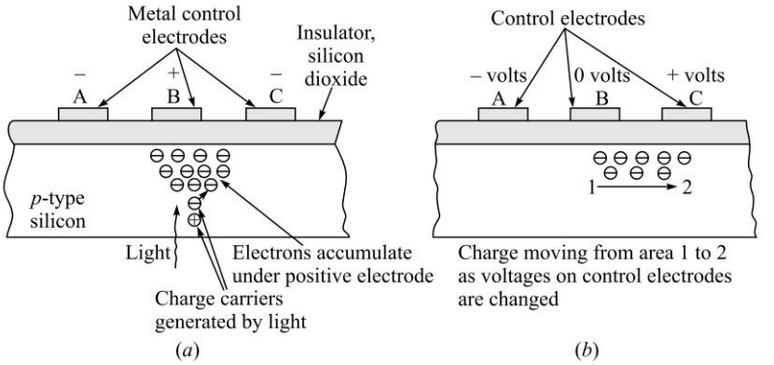


Fig. 7.3 *Basic principle of charge-coupled device: (a) Accumulation of an electron charge in a pixel element: (b) Movement of accumulated charge through the silicon by changing the voltages on the electrodes A, B, and C. (Reprinted with permission of McGraw-Hill, Inc. [10])*

Figure 7.4 indicates one type of CCD imager. Charges are accumulated for the time it takes to complete a single image after which they are transferred line by line into a storage register. For example, register A in Fig. 7.4 accumulates the pixel charge produced by the light image. Once accumulated for a single picture, the charges are transferred line by line to register B. The pixel charges are read out line by line through a horizontal register C to an output amplifier. During readout, register A is accumulating new pixel elements. The complete cycle is repeated approximately every $\frac{1}{60}$ th of a second.

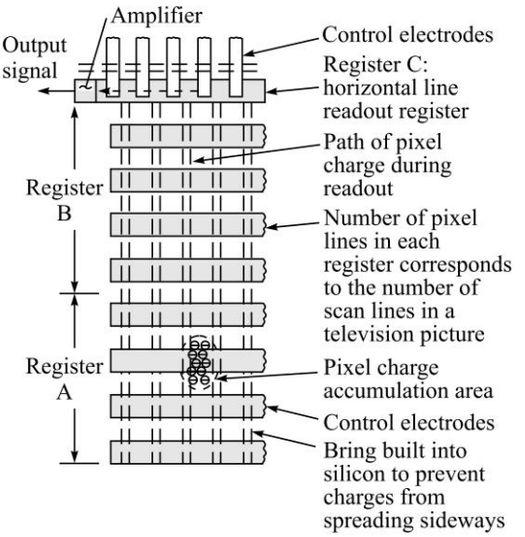


Fig. 7.4 *One type of charge-coupled device imager. Register A accumulates the pixel charges produced by photoconductivity generated by the light image. The B register stores the lines of pixel charges and transfers each line in turn into register C. Register C reads out the charges laterally as shown into the amplifier. (Reprinted with permission of McGraw-Hill, Inc. [10])*

7.2.2 Lighting Techniques

An essential ingredient in the application of machine vision is proper lighting. Good illumination of the scene is important because of its effect on the level of complexity of image-processing algorithms required. Poor lighting makes the task of interpreting the scene more difficult. Proper lighting techniques should provide high contrast and minimize specular reflections and shadows unless specifically designed into the system.

The basic types of lighting devices used in machine vision may be grouped into the following categories:

1. **Diffuse surface** *devices* Examples of diffuse surface illuminators are the typical fluorescent lamps and light tables.

2. **Condenser projectors** A condenser projector transforms an expanding light source into a condensing light source. This is useful in imaging optics.
3. **Flood or spot projectors** Flood lights and spot lights are used to illuminate surface areas.
4. **Collimators** Collimators are used to provide a parallel beam of light on the subject.
5. **Imagers** Imagers such as slide projectors and optical enlargers form an image of the target at the object plane.

Various illumination techniques have been developed to use these lighting devices. Many of these techniques are presented in Table 7.1. The purpose of these techniques is to direct the path of light from the lighting device to the camera so as to display the subject in a suitable manner to the camera.

Table 7.1 *Illumination techniques.*

	Technique	*Function/use*
A.	Front light source	
1.	Front illumination	Area flooded such that surface is defining feature of image.
2.	Specular illumination (dark field)	Used for surface defect recognition (background dark).
3.	Specular illumination (light field)	Used for surface defect recognition; camera in-line with reflected rays (background light).
4.	Front imager	Structured light application; imaged light superimposed on object surface–light beam displaced as function of thickness.
B.	Back light source	
1.	Rear illumination (lighted field)	Uses surface diffusor to silhouette features; used in parts inspection and basic measurements.
2.	Rear illumination (condenser)	Produces high-contrast images; useful for high magnification application.
3.	Rear illumination (collimator)	Produces parallel light ray source such that the features of object do not lie in same plane.
4.	Rear offset illumination	Useful to produce feature highlights when feature is in transparent medium.
C.	Other miscellaneous devices	
1.	Beam splitter	Transmits light along same optical axis as sensor; advantage is that it can illuminate difficult-to-view objects.
2.	Split mirror	Similar to beam splitter but more efficient with lower intensity requirements.
3.	Non-selective redirectors	Light source is redirected to provide proper illumination
4.	Retroreflector	A device that redirects incident rays back to sensor; incident angel capable of being varied; provides high contrast for object between source and reflector.
5.	Double density	A technique used to increase illumination intensity at sensor; used with transparent media and retroreflector.

As shown in Table 7.1, there are two basic illumination techniques used in machine vision: front lighting and back lighting. *Front lighting* simply means that the light source is on the same side of the scene as the camera. Accordingly, reflected light is used to create the image viewed by the camera. In *back lighting,* the light source is directed at the camera and is located behind the objects of interest. The image seen by the camera is a silhouette of the object under study. Back lighting is suitable for applications in which a silhouette of the object is sufficient for recognition or where there is a need to obtain relevant measurements. The table also lists other miscellaneous techniques that may be used to provide illumination.

7.2.3 Analog-to-Digital Signal Conversion

For a camera utilizing, the vidicon tube technology it is necessary to convert the analog signal for each pixel into a digital form. The analog-to-digital (A/D) conversion process involves taking an analog input voltage signal and producing an output that represents the voltage, signal in the digital memory of a computer. A/D conversion consists of three phases: sampling, quantization, and encoding.

Sampling A given analog signal is sampled periodically to obtain a series of discrete time analog signals. This process is illustrated in Fig. 7.5. By setting a specified sampling rate, the analog signal can be approximated by the sampled digital outputs. How well we approximate the analog signal is determined by the sampling rate of the A/D converter. The sampling rate should be at least twice the highest frequency in the video signal if we wish to reconstruct that signal exactly.[14]

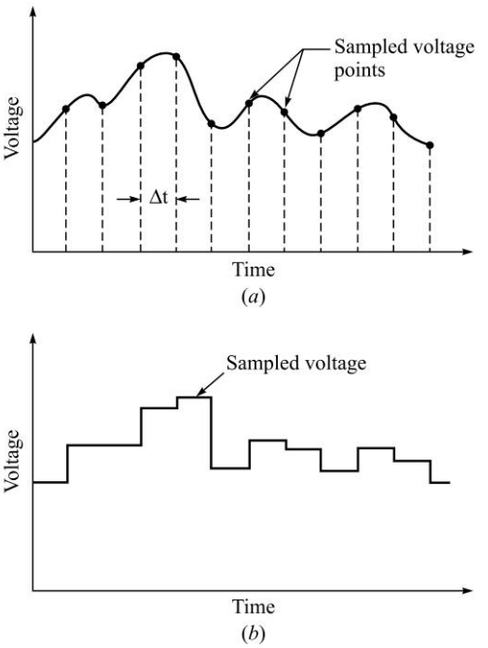


Fig. 7.5 *Sampling and digitizing an analog waveform: (a) Analog waveform indicating sampling interval, t, and sampled voltage points, (b) Digital approximation to analog signal.*

Example 7.1 Consider a vision system using a vidicon tube. An analog video signal is generated for each line of the 512 lines comprising the faceplate. The sampling capability of the A/D converter is 100 nanoseconds (100×10^{-9} s). This is the cycle time required to complete the A/D conversion process for one pixel. Using the American standard of 33.33 milliseconds $\left(\frac{1}{30}\text{ s}\right)$ to scan the entire faceplate consisting of 512 lines, determine the sampling rate and the number of pixels that can be processed per line.

The cycle time per pixel is limited by the A/D conversion process to 100×10^{-9} s

$$= 0.1 \times 10^{-6} \text{ s/pixel}$$

The scanning rate for the 512 lines in the faceplate is $\frac{1}{30}\text{ s} = 33.33 \times 10^{-3}$ s

Accordingly, the scanning rate for each line is = $(33.33 \times 10^{-3}\text{ s})/512$ lines

$$= 65.1 \times 10^{-6} \text{ s/line}$$

The number of pixels that can be processed per line is therefore

$$= \frac{65.1 \times 10^{-6}\text{s/line}}{0.1 \times 10^{-6}\text{s/pixel}}$$

$$= 651 \text{ pixels/line.}$$

In practice, an allowance would have to be made for the time required when the electron beam is shut off during its raster from one line to the next. This dead time would decrease the number of pixels used in the vidicon system. For example, the number of pixels that the system could handle might be reduced from 651 pixels per line to 512 pixels per line, which is similar to the number of pixels per line used in home television. For a 512×512 pixel image, there are a total of 262,144 pixels to consider during the faceplate scanning period. Each pixel must be processed and some type of image processing function must be carried out during the sampling period. This results in substantial demand on the digital computer performing the function. The requirement on the capability of the vision system computer has been one of the limiting factors in the development of machine vision. Systems with a smaller number of pixels ($128 \times 128 = 16{,}384$ pixels or $256 \times 256 = 65{,}536$ pixels) impose lower computational requirements: however, the image resolution of these systems is much lower than for systems possessing a greater pixel density. For a given video signal representing one line, the number of samples taken determines the horizontal resolution of the imaging system. The total number of lines determines the vertical resolution.

Quantization Each sampled discrete time voltage level is assigned to a finite number of defined amplitude levels. These amplitude levels correspond to the gray scale used in the system. The predefined amplitude levels are characteristic to a particular A/D converter and consist of a set of discrete values of voltage levels. The number of quantization levels is defined by

Number of quantization levels = 2^n

where n is the number of bits of the A/D converter. A large number of bits enables a signal to be represented more precisely. For example, an 8 bit converter would

allow us to quantize at $2^8 = 256$ different values whereas 4 bits would allow only $2^4 = 16$ different quantization levels.

Encoding The amplitude levels that are quantized must be changed into digital code. This process, termed encoding, involves representing an amplitude level by a binary digit sequence. The ability of the encoding process to distinguish between various amplitude levels is a function of the spacing of each quantization level. Given the full-scale range of an analog video signal, the spacing of each level would be defined by

$$\text{Quantization level spacing} = \frac{\text{Full-scale range}}{2^n}$$

The quantization error resulting from the quantization process can be defined as

$$\text{Quantization error} = \pm \frac{1}{2} (\text{quantization level spacing})$$

Example 7.2 A continuous video voltage signal is to be converted into a discrete signal. The range of the signal after amplification is 0 to 5 V. The A/D converter has an 8 bit capacity. Determine the number of quantization levels, the quantization level spacing, the resolution, and the quantization error.

For an 8 bit capacity, the number of quantization levels is $2^8 = 256$. The A/D converter resolution = $\frac{1}{256} = 0.0039$ or 0.39 per cent. For the 5 V range,

$$\text{Quantization level spacing} = \frac{(5\text{ V})}{(2^8)} 0.0195\text{ V}$$

$$\text{Quantization error} = \pm \frac{1}{2} (0.0195\text{ V}) = 0.00975\text{ V}$$

To represent the voltage signal in binary form involves the process of encoding. This is accomplished by assigning the sequence of binary digits to represent increasing quantization levels.

Example 7.3 For the A/D converter of Example 7.2, indicate how the voltage signal might be indicated in binary form.

Of the eight bits available, we can use these bits to represent increasing quantization levels as fellows:

Voltage range, V	*Binary number*	*Gray scale*
0–0.0195	0000 0000	0 (black)
0.0195–0.0390	0000 0001	1 (dark gray)
0.0390–0.0585	0000 0010	2
⋮	⋮	⋮
4.9610–4.9805	1111 1110	254 (light gray)
4.9805–5.0	1111 1111	255 (white)

If we had a voltage signal which, for example, varied over a ± 5 V range, we could then assign the first bit to indicate polarity and use the succeeding 7 bits to identify specific voltage ranges. The number of quantization levels would be $2^7 = 128$ and the quantization level spacing would be 5 V divided by 128 or 0.039 V.

7.2.4 Image Storage

Following A/D conversion, the image is stored in computer memory, typically called a frame buffer. This buffer may be part of the frame grabber or in the computer itself. Various techniques have been developed to acquire and access digital images. Ideally, one would want to acquire a single frame of data in real time. The frame grabber is one example of a video data acquisition device that will store a digitized picture and acquire it in = $\frac{1}{30}$ s. Digital frames are typically quantized to 8 bits per pixel. However, a 6 bit buffer is adequate since the average camera system cannot produce 8 bits of noise-free data. Thus the lower-order bits are dropped as a means of noise cleaning. In addition, the human eye can only resolve about $2^6 = 64$ gray levels. A combination of row and column counters are used in the frame grabber which are synchronized with the scanning of the electron beam in the camera. Thus, each position on the screen can be uniquely addressed. To read the information stored in the frame buffer, the data is 'grabbed' via a signal sent from the computer to the address corresponding to a row–column combination. Such frame grabber techniques have become extremely popular and are used frequently in vision systems.

7.3 IMAGE PROCESSING AND ANALYSIS

The discussion in the preceding section described how images are obtained, digitized, and stored in a computer. For use of the stored image in industrial applications, the computer must be programmed to operate on the digitally stored image. This is a substantial task considering the large amount of data that must be analyzed. Consider an industrial vision system having a pixel density of 350 pixels per line and 280 lines (a total of 98,000 picture elements), and a 6-bit register for each picture element to represent various gray levels; this would require a total of $98{,}000 \times 6 = 588{,}000$ bits of data for each $\frac{1}{30}$ s. This is a formidable amount of data to be processed in a short period of time and has led to various techniques to reduce the magnitude of the imageprocessing problem. These techniques include:

1. Image data reduction
2. Segmentation
3. Feature extraction
4. Object recognition

These techniques of image data analysis are discussed in the following subsections.

7.3.1 Image Data Reduction

In image data reduction, the objective is to reduce the volume of data. As a preliminary step in the data analysis, the following two schemes have found common usage for data reduction:

1. Digital conversion
2. Windowing

The function of both schemes is to eliminate the bottleneck that can occur from the large volume of data in image processing.

Digital conversion reduces the number of gray levels used by the machine vision system. For example, an 8-bit register used for each pixel would have $2^8 = 256$ gray levels. Depending on the requirements of the application, digital conversion can be used to reduce the number of gray levels by using fewer bits to represent the pixel light intensity. Four bits would reduce the number of gray levels to 16. This kind of conversion would significantly reduce the magnitude of the image-processing problem.

Example 7.4 For an image digitized at 128 points per line and 128 lines, determine (a) the total number of bits to represent the gray level values required if an 8-bit A/D converter is used to indicate various shades of gray, and (b) the reduction in data volume if only black and white values are digitized.

(a) For gray scale imaging with $2^8 = 256$ levels of gray

Number of hits = $128 \times 128 \times 256 = 4{,}194{,}304$ bits.

(b) For black and white (binary bit conversion)

Number of bits = $128 \times 128 \times 2 = 16{,}384$ bits

Reduction in data volume = $4{,}194{,}304 - 16{,}384$

$= 4{,}177{,}920$ bits.

Windowing involves using only a portion of the total image stored in the frame buffer for image processing and analysis. This portion is called the *window*. For example, for inspection of printed circuit boards, one may wish to inspect and analyze only one component on the board. A rectangular window is selected to surround the component of interest and only pixels within the window are analyzed. The rationale for windowing is that proper recognition of an object requires only certain portions of the total scene.

7.3.2 Segmentation

Segmentation is a general term which applies to various methods of data reduction. In segmentation, the objective is to group areas of an image having similar characteristics or features into distinct entities representing parts of the image. For example, boundaries (edges) or regions (areas) represent two natural segments of an image. There are many ways to segment an image.

Three important techniques that we will discuss are:

1. Thresholding
2. Region growing
3. Edge detection

In its simplest form, *thresholding* is a binary conversion technique in which each pixel is converted into a binary value, either black or white. This is accomplished by utilizing a frequency histogram of the image and establishing what intensity

(gray level) is to be the border between black and white. This is illustrated for an image of an object as shown in Fig. 7.6. Fig. 7.6(a) shows a regular image with each pixel having a specific gray tone out of 256 possible gray levels. The histogram of Fig. 7.6(b) plots the frequency (number of pixels) versus the gray level for the image. For histograms that are bimodal in shape, each peak of the histogram represents either the object itself or the background upon which the object rests. Since we are trying to differentiate between the object and background, the procedure is to establish a threshold (typically between the two peaks) and assign, for example, a binary bit 1 for the object and 0 for the background. The outcome of this thresholding technique is illustrated in the binary-digitized image of Fig, 7.6(c). To improve the ability to differentiate, special lighting techniques must often be applied to generate a high contrast.

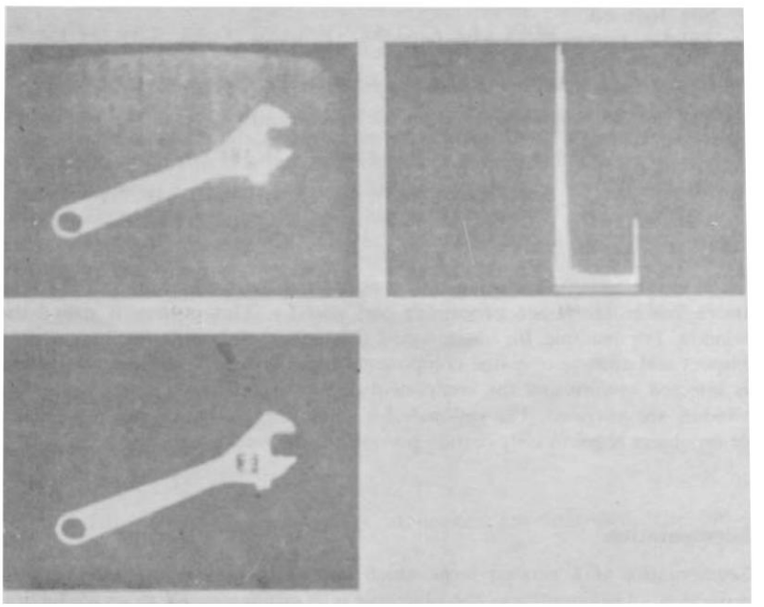


Fig. 7.6 *Obtaining a binary image by thresholding: (a) Image of object with all gray-levels present, (b) Histogram of image, (c) Binary image of object after thresholding. (Photos courtesy: Robotics Laboratory, Lehigh University)*

It should be pointed out that the above method of using a histogram to determine a threshold is only one of a large number of ways to threshold an image. It is however the method used by many of the commercially available robot vision systems today. Such a method is said to use a global threshold for the entire image. In some cases, this is not possible and a local thresholding method as described below may be employed.

When it is not possible to find a single threshold for an entire image (for example, if many different objects occupy the same scene, each having different levels of intensity), one approach is to partition the total image into smaller rectangular areas and determine the threshold for each window being analyzed.

Thresholding is the most widely used technique for segmentation in industrial vision applications. The reasons are that it is fast and easily implemented and that the lighting is usually controllable in an industrial setting.

Once thresholding is established for a particular image, the next step is to identify particular areas associated with objects within the image. Such regions usually possess uniform pixel properties computed over the area. The pixel properties may be multidimensional; that is, there may be more than a single attribute that can be used to characterize the pixel (e.g., color and light intensity). We will avoid this complication and confine our discussion to single pixel attributes (light intensity) of a region.

Region growing is a collection of segmentation techniques in which pixels are grouped in regions called *grid elements* based on attribute similarities. Defined regions can then be examined as to whether they are independent or can be merged to other regions by means of an analysis of the difference in their average properties and spatial connectiveness. For instance, consider an image as depicted in Fig. 7.7(*a*). To differentiate between the objects and the background, assign 1 for any grid element occupied by an object and 0 for background elements. It is common practice to use a square sampling grid with pixels spaced equally along each side of the grid. For the two-dimensional image of a key as shown, this would give the pattern indicated in Fig. 7.7(*b*). This technique of creating 'runs' of 1s and 0s is often used as a first-pass analysis to partition the image into identifiable segments or 'blobs.' Note that this simple procedure did not identify the hole in the key of Fig. 7.7(*a*), This could be resolved by decreasing the distance between grid points and increasing the accuracy with which the original image is represented.

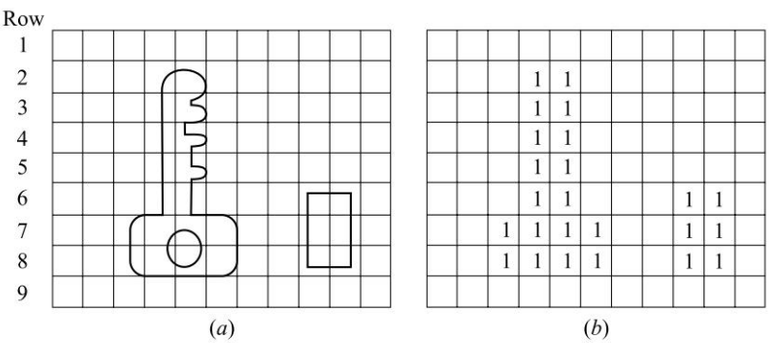


Fig. 7.7 *Image segmentation: (a) Image pattern with grid, (b) Segmented image after runs test.*

For a simple image such as a dark blob on a light background, a runs technique can provide useful information. For more complex images, this technique may not provide an adequate partition of an image into a set of meaningful regions. Such regions might contain pixels that are connected to each other and have similar attributes, for example, gray level. A typical region-growing technique for complex images could have the following procedure:

1. Select a pixel that meets a criterion for inclusion in a region. In the simplest case, this could mean select white pixel and assign a value of 1.

2. Compare the pixel selected with all adjacent pixels. Assign an equivalent value to adjacent pixels if an attribute match occurs.
3. Go to an equivalent adjacent pixel and repeat process until no equivalent pixels can be added to the region.

This simple procedure of 'growing' regions around a pixel would be repeated until no new regions can be added for the image.

The region growing segmentation technique described here is applicable when images are not distinguishable from each other by straight thresholding or edge detection techniques. This sometimes occurs when lighting of the scene cannot be adequately controlled. In industrial robot vision systems, it is common practice to consider only edge detection or simple thresholding. This is due to the fact that lighting can be a controllable factor in an industrial setting and hardware/computational implementation is simpler.

Edge detection considers the intensity change that occurs in the pixels at the boundary or edges of a part. Given that a region of similar attributes has been found but the boundary shape is unknown, the boundary can be determined by a simple edge following procedure. This can be illustrated by the schematic of a binary image as shown in Fig. 7.8. For the binary image, the procedure is to scan the image until a pixel within the region is encountered. For a pixel within the region, turn left and step, otherwise, turn right and step. The procedure is stopped when the boundary is traversed and the path has returned to the starting pixel. The contour-following procedure described can be extended to gray level images.

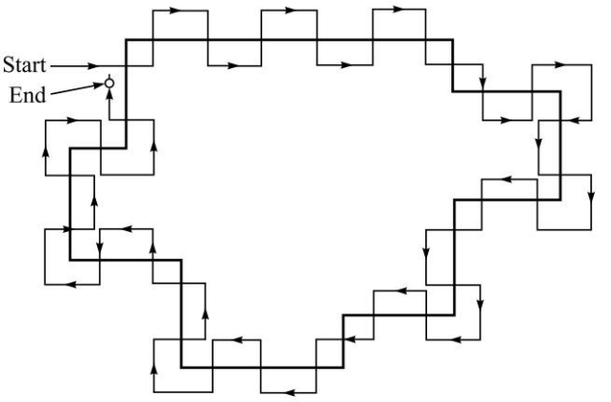


Fig. 7.8 *Edge following procedure to detect the edge of a binary image.*

7.3.3 Feature Extraction

In machine vision applications, it is often necessary to distinguish one object from another. This is usually accomplished by means of features that uniquely characterize the object. Some features of objects that can be used in machine vision include area, diameter, and perimeter. A feature, in the context of vision systems, is a single parameter that permits ease of comparison and identification. A list of

some of the features commonly used in vision applications is given in Table 7.2. The techniques available to extract feature values for two-dimensional cases can be roughly categorized as those that deal with boundary features and those that deal with area features. The various features can be used to identify the object or part and determine the part location and/or orientation.

The region-growing procedures described before can be used to determine the area of an object's image. The perimeter or boundary that encloses a specific area can be determined by noting the difference in pixel intensity at the boundary and simply counting all the pixels in the segmented region that are adjacent to pixels not in the region, that is, on the other side of the boundary. An important objective in selecting these features is that the features should not depend on position or orientation. The vision system should not be dependent on the object being presented in a known and fixed relationship to the camera.

Table 7.2 *Basic features and measures for object identification for two-dimensional objects.*

Gray level (maximum, average, or minimum)

Area

Perimeter length

Diameter

Minimum enclosing rectangle

Center of gravity—For all pixels (n) in a region where each pixel is specified by (x, y) coordinates, the x and y coordinates of the center of gravity are defined as

$$\text{C.G.}_x = \frac{1}{n}\sum_x x$$

$$\text{C.G.}_y = \frac{1}{n}\sum_y y$$

Eccentricity: A measure of 'elongation' Several measures exist of which the simplest is

$$\text{Eccentricity} = \frac{\text{Maximum chord length } A}{\text{Maximum chord length } B}$$

where maximum chord length B is chosen perpendicular to A.

Aspect ratio—The length-to-width ratio of a boundary rectangle which encloses the object. One objective is to find the rectangle which gives the minimum aspect ratio.

Thinness—This is a measure of how thin an object is. Two definitions are in use

(a) $$\text{Thinness} = \frac{(\text{Perimeter})^2}{\text{Area}}$$

This is also referred to as compactness.

Diameter (b) $$\text{Thinness} = \frac{\text{Diameter}}{\text{Area}}$$

The diameter of an object, regardless of its shape, is the maximum distance obtainable for two points on the boundary of an object.

Contd.

Holes—Number of holes in the object.

Moments—Given a region, R, and coordinates of the points (x, y) in or on the boundary of the region, the pqth order moment of the image of the region is given as

$$M_{pq} = \sum_{x, y} x^p y^q$$

The preceding measures provide some basic methods to analyze images in a two-dimensional plane. Various other measures exist for the three-dimensional case as well. To illustrate some of the two-dimensional definitions and measures, the following example is provided:

Example 7.5 Consider the schematic of the image in Fig. 7.9. Determine the area, the minimum aspect ratio, the diameter, the centroid, and the thinness measures of the image.

For the image, the minimum boundary rectangle is as shown. For ease of calculation, the origin is translated to O' with x', y' coordinates. The area may be determined from the moment, $M_{o'o'}$ as

$$M_{o'o'} = \sum_{x', y'} x'y' = 24 \text{ pixels}$$

The minimum aspect ratio would use the minimum boundary rectangle and would simply be

$$\text{Minimum aspect ratio} = \frac{\text{Length}}{\text{Width}} = \frac{9}{4}$$

with the diameter = 9 pixels. The centroid position for the $n = 24$ pixels in the region would be calculated from

$$\text{C.G.}_{x'} = \frac{1}{n}\sum_{x'} x'$$

$$\text{C.G.}_{y'} = \frac{1}{n}\sum_{y'} y'$$

where pixel location is taken at the midpoint of each pixel cell. (C.G. =

$$\text{C.G.}_{x'} = \frac{1}{24}\left[4\left(\frac{1}{2}\right) + 4\left(\frac{3}{2}\right) + 4\left(\frac{5}{2}\right) + 2\left(\frac{7}{2}\right) + \ldots + 2\left(\frac{17}{2}\right)\right]$$

$$= \frac{1}{24}(90) = \frac{15}{4} \text{ units}$$

$$\text{C.G.}_{y'} = \frac{1}{24}\left[3\left(\frac{1}{2}\right) + 9\left(\frac{3}{2}\right) + 9\left(\frac{5}{2}\right) + 3\left(\frac{7}{2}\right)\right]$$

$$= \frac{96}{48} = 2 \text{ units}$$

center of gravity.) The centroid is indicated in Fig. 7.9. Of the two thinness measures defined, the calculations would result in

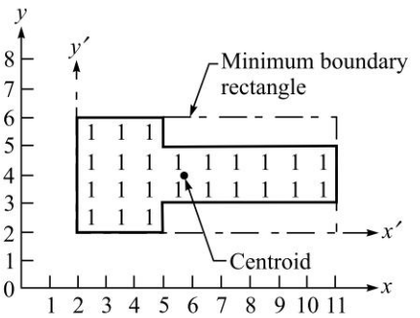


Fig. 7.9 *Schematic of pixel pattern for Example 7.5.*

$$\text{Compactness} = \frac{(\text{Perimeter})^2}{\text{Area}} = \frac{26^2}{24} = 28.17$$

$$\text{Thinness} = \frac{\text{Diameter}}{\text{Area}} = \frac{9}{24} = \frac{3}{8}$$

7.3.4 Object Recognition

The next step in image data processing is to identify the object the image represents. This identification problem is accomplished using the extracted feature information described in the previous subsection. The recognition algorithm must be powerful enough to uniquely identify the object. Object recognition techniques used in industry today may be classified into two major categories:

1. Template-matching techniques
2. Structural techniques

Template-matching techniques are a subset of the more general statistical pattern. Recognition techniques that serve to classify objects in an image into predetermined categories. The basic problem in template matching is to match the object with a stored pattern feature set defined as a model template. The model template is obtained during the training procedure in which the vision system is programmed for known prototype objects. These techniques are applicable if there is not a requirement for a large number of model templates. The procedure is based on the use of a sufficient number of features to minimize the frequency of errors in the classification process. The features of the object in the image (e.g., its area, diameter, aspect ratio, etc.) are compared to the corresponding stored values. These values constitute the stored template. When a match is found, allowing for certain statistical variations in the comparison process, then the object has been properly classified.

Structural techniques of pattern recognition consider relationships between features or edges of an object. For example, it the image of an object can be

subdivided into four straight lines (the lines are called primitives) connected at their end points, and the connected lines are at right angles, then the object is a rectangle. This kind of technique, known *as* syntactic pattern recognition, is the most widely used structural technique. Structural techniques differ from decision-theoretic techniques in that the latter deals with a pattern on a quantitative basis and ignores for the most part interrelationships among object primitives. A detailed discussion of pattern recognition techniques is the subject of complete books and is beyond the scope of this text.

It can be computationally time consuming for complete pattern recognition. Accordingly, it is often more appropriate to search for simpler regions or edges within an image. These simpler regions can then be used to extract the required features. The majority of commercial robot vision systems make use of this approach to the recognition of two-dimensional objects. The recognition algorithms are used to identify each segmented object in an image and assign it to a classification (e.g., nut, bolt, flange, etc.)

7.4 TRAINING THE VISION SYSTEM

The purpose of vision system training is to program the vision system with known objects. The system stores these objects in the form of extracted feature values which can be subsequently compared against the corresponding feature values from images of unknown objects. Typical features used in machine vision were discussed in the previous section and summarized in Table 7.2.

Training of the vision system should be carried out under conditions as close to operating conditions as possible. Physical parameters such as camera placement, aperture setting, part position, and lighting are the critical conditions that should be simulated as closely as possible during the training session.

Vision system manufacturers have developed application software for each individual system marketed. The software is typically based on a high-level programming language. For example, Object Recognition Systems Inc. uses the 'C' language, and Automatix Inc. uses their internally developed language called RAIL (for Robot Automatic Incorporated Language). There are two versions of RAIL, one for automated vision systems and the other for robot programming. We will present portions of the robot programming language in the appendix to Chap. 9.

7.5 ROBOTIC APPLICATIONS

Many of the current applications of machine vision are inspection tasks that do not involve the use of an industrial robot. A typical application is where the machine vision system is installed on a high-speed production line to accept or reject parts made on the line. Unacceptable parts are ejected from the line by some mechanical device that is communicating with the vision system.

Machine vision applications can be considered to have three levels of difficulty. These levels depend on whether the object to be viewed is controlled in position and/

or appearance. Controlling the position of an object in a manufacturing environment usually requires precise fixturing. Controlling the appearance of an object is accomplished by lighting techniques. Also, appearance is influenced by the object's surface texture or coloration. The three levels of difficulty used to categorize machine vision applications in an industrial setting are:

1. The object can be controlled in both position and appearance.
2. Either position or appearance of the object can be controlled but not both.
3. Neither position nor appearance of the object can be controlled.

The third level of difficulty requires advanced vision capabilities. The objective in engineering the vision application is to lower the level of difficulty involved, thereby reducing the level of sophistication of the vision system required in the application. For example, one problem that occurs in object recognition is that the recognition process is facilitated if the object is in a known position and orientation. Parts in a factory are typically not positioned and oriented in this manner. This problem can be reduced from a third level to a first level of difficulty by fixturing the parts and using techniques such as structured lighting to control the appearance.

In this section, the use of machine vision in robotic applications will be emphasized. Robotic applications of machine vision fall into the three broad categories listed below:

1. Inspection
2. Identification
3. Visual servoing and navigation

The first category is one in which the primary function is the inspection process. This is carried out by the machine vision system, and the robot is used in a secondary to support the application. The objectives of machine vision inspection include checking for gross surface defects, discovery of flaws in labeling (during final inspection of the product package), verification of the presence of components in assembly, measuring for dimensional accuracy, and checking for the presence of holes and other features in a part. When these kinds of inspection operations are performed manually, there is a tendency for human error. Also, the time required in most manual inspection operations requires that the procedures be accomplished on a sampling basis. With machine vision, these procedures are carried out automatically, using 100 percent inspection, and usually in much less time. In Chap. 15, we will return to robotic inspection in which many of the applications of robots make use of machine vision.

The second category, *identification,* is concerned with applications in which the purpose of the machine vision system is to recognize and classify an object rather than to inspect it. Inspection implies that the part must be either accepted or rejected. Identification involves a recognition process in which the 'part itself, or its position and/or orientation, is determined. This is usually followed by a subsequent decision and action taken by the robot. Identification applications of machine vision include part sorting, palletizing and depalletizing, and picking parts that are randomly oriented from a conveyer or bin.

In the third application category, *visual servoing and navigational control,* the purpose of the vision system is to direct the actions of the robot (and other devices in the robot cell) based on its visual input. The generic example of robot visual servoing is where the machine vision system is used to control the trajectory of the robot's end effector toward an object in the workspace. Industrial examples of this application include part positioning, retrieving parts moving along a conveyor, retrieving and reorienting parts moving along a conveyor, assembly, bin picking, and seam tracking in continuous arc welding. The use of machine vision in arc welding is described in Chap. 14 when the application of welding robots will be discussed in detail. An example of navigational control would be in automatic robot path planning and collision avoidance using visual data. Clearly, the visual data are just an important input in this type of task and a great deal of intelligence is required in the controller to use the data for navigation and collision avoidance. This and the visual servoing tasks remain important research topics and are not now viable applications of robot vision systems.

The *bin-picking* application is an interesting and complex application of machine vision in robotics which involves both identification and servoing. Bin picking involves the use of a robot to grasp and retrieve randomly oriented parts out of a bin or similar container. The application is complex because parts will be overlapping each other. The vision system must first recognize a target part and its orientation in the container, and then it must direct the end effector to a position to permit grasping and pickup. The difficulty is in the fact that the target part is jumbled together with many other parts (probably identical parts), and the conditions of contrast between the target and its surroundings are far from ideal for part recognition.

Solution of the bin-picking problem owes much to the pioneering work in vision research at the University of Rhode Island. At the time of writing, there are two commercially available bin-picking systems, one offered by Object Recognition Systems, Inc. (ORS) called the i-bot 1 system, and the other by General Electric Co. called Bin Vision. We will describe the ORS product to illustrate the operation of these systems. Fig. 7.10 shows the i-bot system. It utilizes a Unimation PUMA robot to pick parts from the bin. The system is capable of identifying the position and orientation of three parts in a 5-s image-processing cycle. However, the parts must have a relatively large aspect ratio (length-to-width ratio) for the system to operate effectively. Also, the objects in the bin must be the same in terms of features such as size, shape, and texture.

Seam tracking in continuous arc welding is another example of visual servoing and navigation in robotics vision systems. Because the vision system must operate in the presence of an arc-welding torch, special problems in the interpretation of the vision image must be solved. A typical solution uses a form of structured light to project a known pattern on the parts to be welded and uses the observed geometric distortion to determine the path and other parameters required for a successful welding operation. We will discuss this and other sensor technologies used in robotic arc welding in Chap. 14.



Fig. 7.10 *i-bot robot-vision system. (Photo courtesy: Object Recognition Systems, Inc.)*

In addition to the three main categories of machine vision applications in robotics, there are other special purpose vision applications. An example of these special applications is one, in which the vision system functions as a range finder using triangulation or other techniques to determine the distance to the object.

Machine vision, coupled with the force and torque sensors discussed in Chap. 6, can immensely enhance a robot's applicability in manufacturing. Representative tasks in which such highly sensored systems could be applied include manipulation of parts during deburring, flash removal, and assembly. We will discuss the particular technical problems encountered in robotic assembly and the need for sensors in this application in Chap. 15.

Problems

7.1 Consider a vision system which provides one frame of 256 lines every $\frac{1}{2}$ s. The system is a raster scan system. Assume that the time for the electron beam to move from one line to the next takes 15 per cent of the time to scan a single line. Determine the sampling rate for the system if it is specified that there will be 320 pixels on each line.