INDUSTRIAL ROBOTS

Computer Interfacing and Control

WESLEY E. SNYDER

North Carolina State University

PRENTICE-HALL, INC., *Englewood Cliffs, New Jersey 07632*

13

COMPUTER VISION

In this chapter, we will introduce the basic concepts in the field of study known as *computer vision*. This discipline emphasizes the development of techniques which allow a computer to recognize or otherwise understand the content of a picture. Numerous books have been written on this subject, and we could not hope to do it justice in one chapter. Therefore, we concentrate on a subdiscipline called IMV (*industrial machine vision*), which includes robot vision.

Researchers in the field of IMV concentrate their efforts on problems appropriate to the industrial environment. In such an environment, one may be able to control the background, the lighting, the camera position, or other parameters. Such control may allow the use of techniques that would be inappropriate to a general-purpose vision system.

In this chapter, the reader will be introduced to the concept of an electronic image and its digital representation. Then, several strategies for processing such images will be covered. These strategies will allow the computer to make use of image information to guide a robot.

13.1 FUNDAMENTALS

The first section in this chapter will show the reader how digital images may be acquired and represented in the memory of a computer. The operations which the computer must perform to make use of that information for industrial applications will be presented in later sections.

13.1.1 The Formation of a Digital Image

The imaging literature is filled with a variety of imaging devices, including dissectors, flying spot scanners, videcons, orthicons, plumbicons, CCDs (charge-coupled devices), and others (Castleman, 1977; Chien and Snyder, 1975). These devices differ both in the ways in which they form images and in the properties of the images so formed. However, all these devices convert light energy to voltage in similar ways. Since our intent in this chapter is to introduce the reader to the fundamental concepts of image analysis, we will choose one device, the videcon, and discuss the way in which digital images are formed using such a device.

Image Formation With a Silicon Videcon

As shown in Figure 13.1, a lens is used to form an image on the faceplate of the videcon. When a photon of the appropriate wavelength strikes the special material of the faceplate, a quantum of charge is created (an electron-hole pair). Since the conductivity of the material is quite low, these charges tend to remain in the same general area where they were created. Thus, to a good approximation, the charge, q, in a local area of the faceplate follows

$$q = \int_0^{t_f} i dt \qquad (13.1)$$

where i is the incident light intensity, measured in photons per second. If the incident light is a constant over the integration time, then $q = it_f$, where t_f is called the frame time.

The mechanism for reading out the charge, be it electron beam, as in the videcon, or charge coupling, as in CCD cameras, is always designed so that as much of the charge is set to zero as possible. We start the integration process with zero accumulated charge, build up the charge

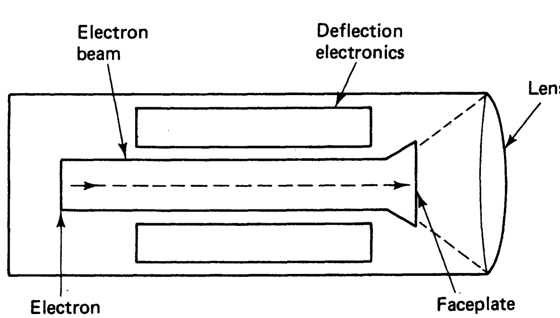


Figure 13.1 Videcon camera.

at a rate proportional to local light intensity, and then read it out. Thus, the signal read out at a point will be proportional to both the light intensity at that point and to the amount of time between read operations.

Since we are interested only in the intensities and not in the integration time, we remove the effect of integration time by making it the same everywhere in the picture. This process, called *scanning*, requires that each point on the faceplate be interrogated and its charge accumulation zeroed, repetitively and cyclically. Probably the most straightforward, and certainly the most common way in which to accomplish this is in a top-to-bottom, left-to-right scanning process called *raster scanning* (Figure 13.2).

In the videcon, an electron beam is used to neutralize the accumulated (positive) charge. Cancellation of the charge causes a current to flow in the circuit proportional to the charge neutralized. Deflection electronics steers the electron beam across the faceplate in a horizontal line. The beam is then shut off and repositioned at the left-hand end of the next, lower, line. The time the beam is off is referred to as *blanking*. This process is repeated until the entire faceplate has been swept clean of charge. However, while the beam is busy neutralizing charge at the bottom of the faceplate, charge is once again building up at the top. Since charge continues to accumulate over the entire surface of the videcon faceplate at all times, it is necessary for the beam to return immediately to the top of the faceplate and begin scanning again. The scanning process is repeated many times each second. In American television, the entire faceplate is scanned once every 33.33 ms (milliseconds) (in Europe, the frame time is 40 ms).

To compute exactly how fast the electron beam is moving, we compute

$$\frac{1 \text{ sec}}{30 \text{ frame}} \div \frac{525 \text{ lines}}{\text{frame}} = 63.5\ \mu\text{s/line} \qquad (13.2)$$

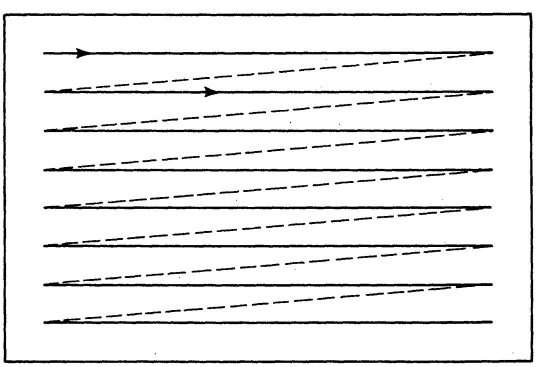


Figure 13.2 Raster scanning: Active video is indicated by a solid line, blanking (retrace) by a dashed line. This simplified figure represents noninterlaced scanning.

Sixty-three and a half microseconds (μs) to scan one line of the picture is fairly fast. (Using the European standard of 625 lines and 25 frames per second, we arrive at almost exactly the same answer, 64 μs per line.) This 63.5 μs includes not only the active video signal but also the blanking period, approximately 18 percent of the line time. Subtracting this *dead time*, we arrive at the active video time, 52 μs per line.

Figure 13.3 shows the output of a television camera as it scans three successive lines. One immediately observes that the raster scanning process effectively converts a picture from a two-dimensional signal to a one-dimensional signal, where voltage is a function of time. Figure 13.3 shows both *composite* and *noncomposite video signals*, that is, whether the signal does or does not include the sync and blanking timing pulses.

The sync signal, while critical to operation of conventional television, is not particularly relevant to our understanding of digital image processing at this time. The blanking signal, however, is the single most important timing signal in a raster scan system. *Blanking* refers to the time that the electron beam is shut off. There are two distinct blanking events: *horizontal blanking*, when the beam moves from the end of one line to the start of the next, and *vertical blanking*, when the beam moves from the bottom of the picture to the top in preparation for a new scan. In a digital system, both blanking events may be represented by pulses on separate digital wires. Composite video is constructed by shifting these special timing pulses negative and adding them to the video signal.

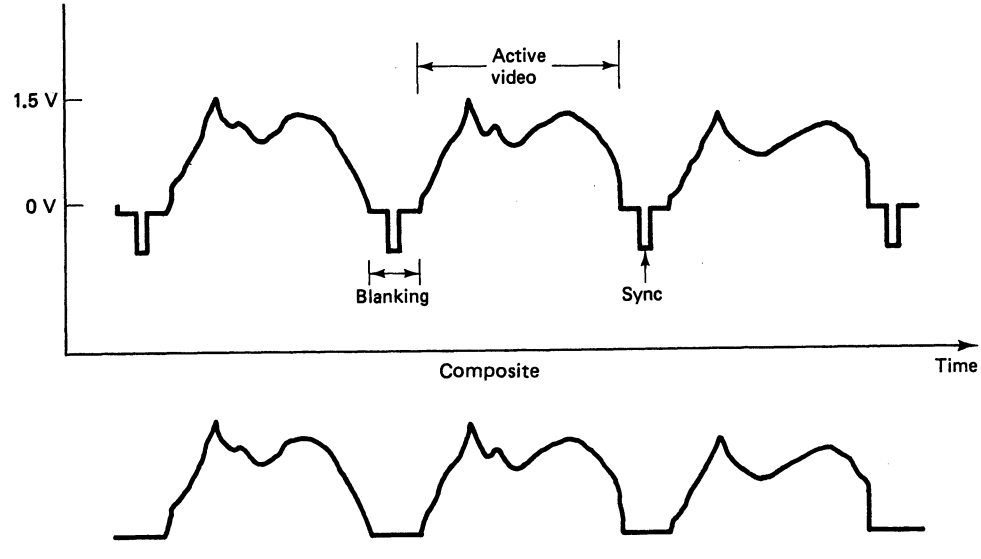


Figure 13.3 The video signal.

Now that we recognize that horizontal blanking signifies the beginning of a new line of video data, we can concentrate on that line and learn how a computer might acquire the brightness information encoded in that voltage.

13.1.2 The Sampling Process

The analog voltage is converted to a digital representation using an analog-to-digital converter such as the flash converters described in Chapter 2. This device performs two functions simultaneously: sampling and quantization. Although these functions occur together, we will discuss them separately since they have different effects.

Figure 13.4(a) shows an analog voltage represented as a function of time, and Figure 13.4(b) shows the same voltage after sampling. The sampling process can be considered a mechanism for approximating the waveform. At discrete times, the waveform is interrogated and that value remembered until the next sampling time. The sampled analog waveform thus consists of a series of *steps*, with constant values between the steps. The conditions under which this process results in an accurate approximation of the waveform will be discussed later.

Resolution

The *resolution* of a system is determined in large part by the sampling process. The number of samples on a single line defines the

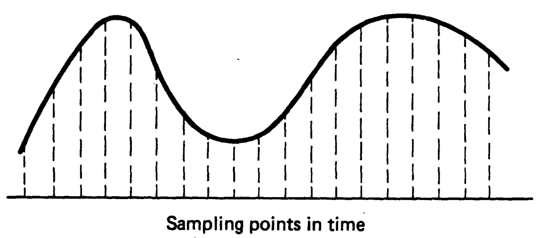


Figure 13.4(a) Analog signal.

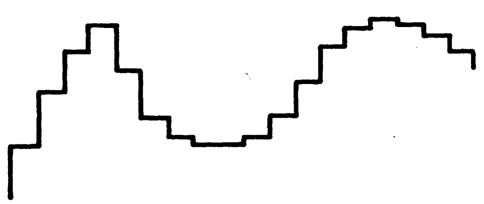


Figure 13.4(b) Sampled analog signal.

horizontal resolution of the system. Similarly, the number of lines defines the *vertical resolution*. This is demonstrated by a comparison of American and European television in which the greater vertical resolution of the European picture, with the standard of 625 lines, is obvious to the viewer.

One common sampling rate is 100 ns (nano seconds) per pixel. Such a sampling rate is easily derived from a 10-MHz clock and results in just over 512 samples on each line. The number 512 is very convenient from a hardware point of view since it is a power of 2. Using a sampling rate of 103 ns/pixel gives 512 samples on exactly one full line of video.

Dynamic Range

Once an analog signal has been sampled, it is converted to digital form by a process known as *quantization* as shown in Figure 13.5. The digital representation of a signal can have only a finite number of possible values, defined by the number of bits in the output word. For example, video signals are often encoded to 8 bits of accuracy, thus allowing a signal to be represented as one of a possible 256 values. A larger number of bits allows a signal to be represented to a greater degree of accuracy.

The accuracy (number of bits) of the digital representation is often referred to as the *dynamic range* of the imaging system. We must caution the reader that the meaning of the term dynamic range sometimes varies. An alternative definition specifies the dynamic range as the range of input signals over which a camera successfully operates. For example, if we open the iris on the camera, we may be said to alter the dynamic range of the camera. Both meanings are accepted and are in common use, but they differ according to their contexts.

Thus, we conclude that the digital image is "discrete in space and discrete in value." We also observe that there is a one-to-one relationship between time and space. That is, if we refer to the *sampling time*,

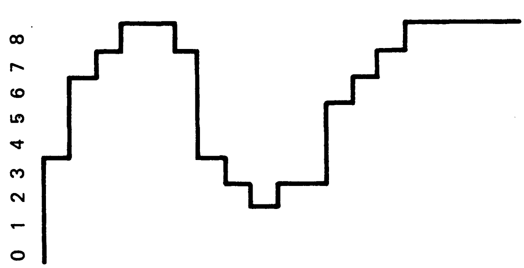


Figure 13.5 Quantized sampled signal.

we must speak of it relative to the top-of-picture signal (vertical blanking). That timing relationship identifies a unique position on the screen.

The Sampling Theorem

In 1948, Claude Shannon derived the *sampling theorem*. This theorem states, simply, that if an analog signal is filtered by an ideal low-pass filter with cutoff at a frequency f, then that (filtered) signal can be exactly reconstructed if it is sampled using a sampling rate greater than or equal to $2f$. Said another way, if we wish to sample and store an analog signal and to be able to reconstruct that signal exactly from the sampled version, our sampling rate must be at least equal to twice the highest frequency in the signal.

The sampling theorem addresses only the effects of errors due to the sampling process; it says nothing about the effects of quantization.

Black-and-white television systems are generally designed to have a 5-MHz bandwidth. Thus, a sampling rate of 10-MHz is required to completely recover the original signal. In fact, the sampling rate of 103 ns which we proposed earlier is very close to this rate. We chose 512 samples per line rather arbitrarily in the earlier discussion, presumably because it was a power of 2. Now, we can see, however, that 512 pixels is not only a power of 2, but it also results naturally from the application of the sampling theorem to the video signal.

The sampling theorem appears to lead to a fascinating contradiction. If we undersample the video signal, say, with 100 points per line, or even lower, and then reconstruct the image, when you and I look at it, we can still recognize the contents of the scene. How can this be? The sampling theorem seems to say that we must sample at 512 (or more) points per line.

In fact, there is no contradiction at all. The sampling theorem says that we must sample at twice the bandwidth to be able to reconstruct the image *without distortion*. Images are highly redundant in their information content, and the human brain is an excellent image-interpolating machine.

Example 13.1 Effects of Sampling

Show the effects of undersampling a television image in both the horizontal and vertical directions. Assume an initial image of 512 lines of 512 pixels each. Show the results of 2:1, 4:1, and 8:1 undersampling. Determine the clock rate required to produce such images, and the memory required to store them.

Solution: Figure 13.6 shows the results of the three cases. To achieve 2:1 undersampling horizontally, we use a clock period of $103 \times 2 = 206$ ns per clock. To achieve such undersampling vertically, we store only every other line. 64K bytes of memory are required (assuming that 1 byte is used to store 1 pixel).

Figure 13.6(a) Image represented by using 256 X 256 pixels.

Figure 13.6(b) Image represented by using 128 X 128 pixels.

Figure 13.6(c) Image represented by using 64 X 64 pixels.

For 4 : 1 undersampling, the clock period = 412 ns, and we store every fourth line. 16K bytes of memory are required.

For 8 : 1 undersampling, the clock period = 1.8 μs, every eighth line is stored, and 4K are required.

Quantization error is the term used to refer to the fact that information is lost whenever the continuously valued analog signal is partitioned into discrete ranges by the limited dynamic range of the A/D converter. Quantization error is observed in reconstructed images either as random noise effects or as *contouring*. Figure 13.7(a) shows an image that has been quantized to 16 grey levels; Figure 13.7(b) shows the same image quantized to 8 levels.

Experiments have shown that at a given light level, the human eye can discern only about 30 grey levels. Of course, with an overall change in brightness, the eye undergoes *dark adaptation*, a chemical process, and the iris opens; therefore, *which* 30 shades of grey are distinguishable may vary from one environment to another.

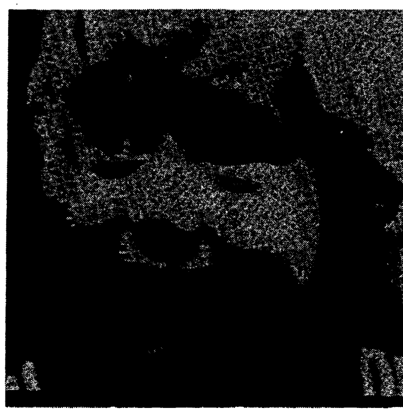


Figure 13.7(a) Sixteen grey levels.

Figure 13.7(b) Eight grey levels.

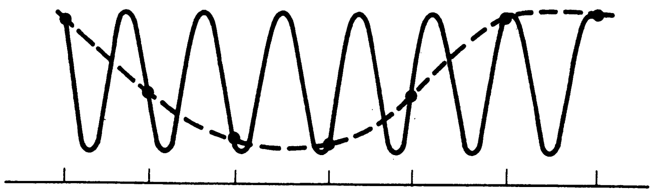


Figure 13.8 An analog sine wave sampled at too low a rate.

Thirty shades of grey would seem to indicate that 5 bits is all the dynamic range needed. Most systems, however, use 8 bits, which allows a limited emulation of the effects of the iris. Furthermore, 8-bit memories are conveniently available.

Aliasing

Aliasing is the phenomenon that occurs when the requirements of the sampling theorem are not met. This may be best explained by considering Figure 13.8, which shows a sine wave being sampled at a sampling rate only slightly less than its own frequency. The data acquisition system is unable to correctly reconstruct the original signal because it "thinks" that it is sampling a signal much lower in frequency, as shown by the dotted line.

Example 13.2 Effects of Quantization

Show the effects of quantizing a 512 X 512 image using 16 shades of grey and 8 shades of grey. Determine the memory required for each.

Solution: See Figure 13.7. Sixteen shades of grey can be encoded into 4 bits. Therefore 512 X 512 X 4 = 1 megabits of memory is required. This would probably be stored as 2 pixels per byte.

Eight shades of grey can be encoded into 3 bits. Therefore, 512 X 512 X 3 = 786,432 bits. However, it would be difficult to store 3-bit/pixels in a byte-addressed memory. The most convenient means would probably be a 4-bit/pixel, resulting in the same memory requirements as the 16-shade case.

13.2 IMAGE PROCESSING FUNCTIONS

In Section 13.1, we discussed how digital images could be acquired, stored, and represented. Once this proces is completed, and we have a digital representation of our image stored in the memory of our computer, we can begin to operate on that image and use it in industrial applications. There are three generic classes of operations that we could ask the computer to perform on images: enhancement, restoration, and analysis. Analysis is the emphasis of the remainder of this chapter, and we will, therefore, touch only on enhancement and restoration here to a degree sufficient to define the terms.

Enhancement

Image enhancement could be most simply defined as the science of making images "look better." Generally, this means looking better from the point of view of a human observer, although enhancement techniques may also be used to preprocess an image prior to analyzing it.

In general, the computer does not need to have any knowledge of the contents of the scene to enhance it. As an example, we will describe what is probably the simplest form of enhancement, grey scale stretching, or contrast stretching.

First, we (that is, the computer) make a pass over the image, memorizing the largest intensity value in the image, and the smallest. Let us refer to them as I_{max} and I_{min}. If the maximum possible value is 255 (for 8-bit images) and the smallest possible value is 0, we can stretch the contrast by updating each pixel value in the following way:

$$I_{new} = \frac{(I_{old} - I_{min}) \cdot 255}{(I_{max} - I_{min})} \tag{13.3}$$

Thus, the new image will have a maximum value of 255 and a minimum value of 0, effectively utilizing the total dynamic range of the output (viewing) system. Similar techniques are also useful on input to normalize images to compensate for variations in lighting.

Many more sophisticated techniques exist for enhancing images, including histogram modification and edge enhancement (see Pratt, 1978; Gonzalez and Wintz, 1977).

Restoration

In the process of acquiring images, distortion always occurs (Chien and Snyder, 1975). There are many sources of distortion, including

Vignetting: Lenses transmit more efficiently near the center than near the outside, resulting in images that are darker near the outside.

Parabolic distortion: Electron beam efficiency is greater if the beam is normal to the faceplate, resulting in exactly the same effects as vignetting.

Blooming: Too much light in a single spot on the faceplate allows the accumulated charges to diffuse outward during a single scan time, resulting in bright spots that seem too large.

Lag: The electron beam is not 100 percent efficient, leaving some accumulated charge not neutralized after each scan, resulting in a "ghost" image when the scene changes.

Motion blur: If image motion is more rapid than the scan time, the image of the same point may be smeared over several pixels.

Geometric distortion: Lenses are not perfect, nor are faceplates, and the resulting image may be distorted geometrically. The image of a perfect circle, for example, may not be a perfect circle.

Blur: Improperly focused optics or an improperly focused electron beam can result in an image in which edges are not as sharp as they could be. In fact, in almost all systems of high resolution (e.g., 512 X 512), edges are never perfect step functions.

If we known the exact mathematical form of these distortions, we could write

$$g(x, y) = D[f(x, y)] \tag{13.4a}$$

where $f(x, y)$ is the "true" undistorted image, D is a distortion operator, and $g(x, y)$ is the measured image. If the D operator has an inverse, then we could recover the original image from the distorted one by simply computing

$$f(x, y) = D^{-1}[g(x, y)]. \tag{13.4b}$$

Of course, in general, D is extremely complex and may not have an inverse. The field of research that studies such distortion functions and attempts to find inverse operators is called *image restoration*. In most industrial applications, restoration techniques are not used explicitly or rigorously. However, corrections for geometric distortion (warping) are routinely done (see Pratt, 1978; Gonzalez and Wintz, 1977; Andrews and Hart, 1977).

13.3 IMAGE ACQUISITION HARDWARE

There are many ways in which to acquire and access a digital image. Here, we will describe only one technique, which allows the acquisition of a single frame of video data in real frame.

13.3.1 The Frame Buffer

Figure 13.9 shows a block diagram of a digital data acquisition system designed to acquire and store one frame of video data, quantized

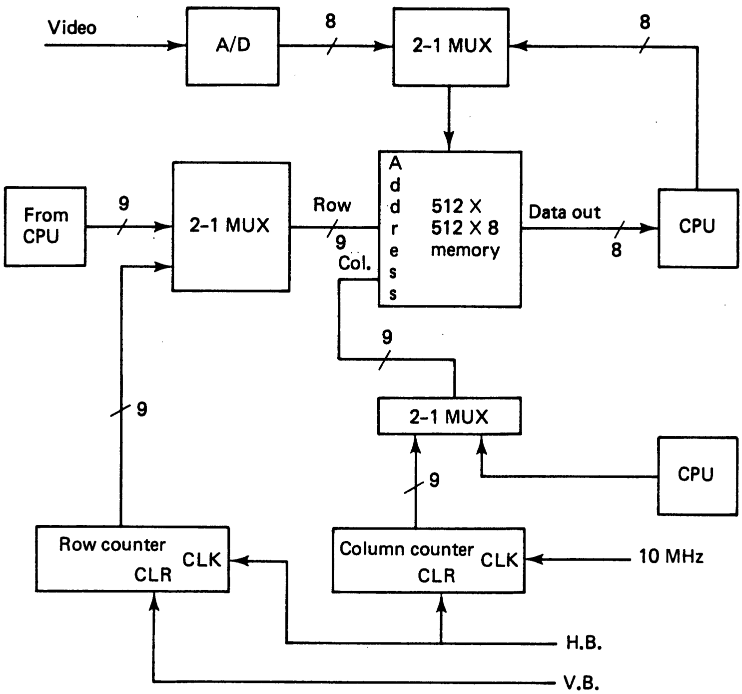


Figure 13.9 A video data acquisition system.

to 8 bits per pixel. The memory in this system, referred to as a *frame buffer*, will store a 512 X 512 picture and will acquire it in 33 ms.*

The fundamental concept behind the design is the use of counters which count in synchronism with the motion of the electron beam. The 9-bit column counter counts from 0 to 511 at a 100-ns/count rate. The column counter is cleared by the horizontal blanking pulse (HB). Hence, a zero in this counter indicates that the beam is at the left-hand end of a scan line.

Although the horizontal blanking pulse clears the column counter, it is used to increment the row counter. Thus, the row counter keeps track of which row the electron beam is currently on. The vertical blanking pulse (VB), which signifies that the beam is at the top of the picture, is used to clear the row counter.

The combination of the row and column counters composes the 18-bit address to the memory. Thus, for each position on the screen, there exists a unique address.

*In this simple design, we are assuming interlaced scanning.

To make use of the information stored in the frame buffer, the computer must be able to read it. This is accomplished via 2-1 multiplexers, one on the row address bus and one on the column address bus. During a computer access, the 18-bit address is provided by the computer rather than coming from the counters. To prevent erroneous addressing, the memory WRITE operation of data from the A/D converter must be disabled while the computer is reading.

This rather straightforward design provides data at the video rate. In the past, such high-speed configurations have not been popular due to the cost of high-speed A/D converters and memory. With the rapid decline in costs of both items, this *frame grabber* technique has now replaced virtually all the other, lower-speed techniques.

Another factor influencing the popularity of frame grabbers is the fact that the same hardware can provide raster scan graphics capabilities. In a graphics application, the memory is read in synchronism with the scan. The data are converted to analog video by a D/A converter and are transferred from there to a TV monitor. Consequently, to convert a raster scan graphics system to a frame grabber requires little more than addition of an A/D converter.

The Pixel Array

In this section, we have discussed a video acquisition system capable of sampling the video signal at a rate of one sample every 100 ns, resulting in an array of 512 samples on each line and 512 lines. Once stored in a digital memory, this array may be accessed as a conventional two-dimensional array of numbers. Each number is referred to as a *pixel*, short for "picture cell." (The term *pel*, representing "picture element," is sometimes used in the same way, especially in Europe.)

Typically a single pixel is represented by one 8-bit byte. The most important observation to be made at this point with respect to the pixel array is its size. A typical 512 X 512 array requires a quarter of a million bytes of memory. At one time, this would have been a prohibitive amount of memory. Memory cost is no longer a significant restriction, but using a serial computer on such a massive amount of data is still extremely time consuming. For this reason, much of the ongoing research in vision today is directed toward the development of algorithms that can make use of parallelism (Fu and Ichikawa, 1982).

In addition to frame grabbers, several manufacturers are now marketing frame buffer systems augmented with other special high-speed hardware for performing operations on the digital image. In the next section, we will examine a few of these operations.

13.4 SEGMENTATION

In many robot vision applications, the set of possible objects in the scene is quite limited. For example, if the camera is viewing a conveyer, there may be only one type of part which appears, and the vision task could be to determine the position and orientation of the part. In other applications, the part being viewed may be one of a small set of possible parts, and the objective is to both locate and identify each part.* Finally, the camera may be used to inspect parts for quality control.

In this section, we will assume that the parts are fairly simple and can be characterized by their two-dimensional projections, as provided by a single camera view. Furthermore, we will assume that the shape is adequate to characterize the objects. That is, color or variation in brightness is not required. We will first consider dividing the picture into connected regions.

A *segmentation* of a picture is a partitioning into connected regions, where each region is homogeneous in some sense and is identified by a unique label. For example, in Figure 13.10, region 1 is identified as the background. Although region 4 is really background also, it is labeled as a separate region since it is not connected to region 1. While there are several ways to perform segmentation, we will discuss only one here.

13.4.1 Segmentation by Thresholding

In applications where grey scale is not important, we can segment a picture into "objects" and "background" by simply choosing a threshold in brightness. We define any regions whose brightness is above the threshold as *object* and all below the threshold as *background*.

There are several different ways to choose thresholds, ranging from trivially simple techniques to quite sophisticated methods. As the sophistication of the technique increases, performance improves at the cost of increased computational complexity.

Probably the most important factor to note is the *local nature of thresholding*. That is, a single threshold is almost never appropriate for an entire scene. It is nearly always the local contrast between object and background that contains the relevant information. Since camera sensitivity drops off from the center of the picture to the edges due to parabolic distortion and/or vignetting, it is useless to attempt to establish a global threshold. A dramatic example of this effect can be seen in an

*A special case of this application occurs when only one type of part comes down the line, but that part has substantial three-dimensional structure and may be resting in one of several possible *stable states*. In this case, the view corresponding to each stable state may be treated as the view of a different object.

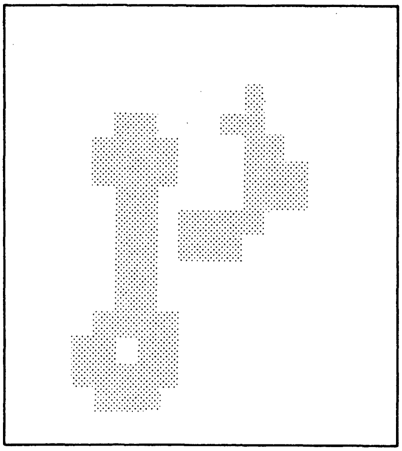


Figure 13.10(a) A picture with two foreground regions.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1
1	1	1	1	1	2	2	1	1	1	3	3	1	1	1	1	1	1
1	1	1	1	2	2	2	2	1	1	1	3	3	1	1	1	1	1
1	1	1	1	2	2	2	2	1	1	1	3	3	3	1	1	1	1
1	1	1	1	1	2	2	1	1	1	1	3	3	3	1	1	1	1
1	1	1	1	1	2	2	1	3	3	3	3	1	1	1	1	1	1
1	1	1	1	1	2	2	1	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1
1	1	1	2	2	4	2	2	1	1	1	1	1	1	1	1	1	1
1	1	1	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 13.10(b) A segmentation and labeling of the picture in Figure 13.10(a). Region 4 is a hole in region 2.

image of a chess board, in which the white squares at the corners are actually darker than the black squares in the center.

Effects such as parabolic distortion and vignetting are quite predictable and easy to correct. In fact, off-the-shelf hardware is available for just such applications. It is more difficult, however, to predict and correct effects of nonuniform ambient illumination, such as sunlight through a window, which changes radically over the day.

Since a single threshold cannot provide sufficient performance, we must choose local thresholds. The most common approach is called *block thresholding*, which the picture is partitioned into rectangular blocks and different thresholds are used on each block. Typical block sizes are 32 X 32 or 64 X 64 for 512 X 512 images. The block is first analyzed and a threshold is chosen; then that block of the image is thresholded using the results of the analysis.

Choosing a Threshold

The simplest strategy for choosing a threshold is to average the intensity over the block and choose $i_{avg} + \Delta i$ as the threshold, where Δi is some small increment, such as 5 out of 256 grey levels. Such a simple thresholding scheme can have surprisingly good results (Page, Snyder, and Rajala, 1983).

However, when the simpler schemes fail, one is forced to move to more sophisticated techniques, such as thresholding based on histogram analysis. Before we describe this technique, we will first define a histogram.

The *histogram* $h(i)$ of an image $i(x, y)$ is a function of the permissible intensity values. In a typical imaging system, intensity takes on values betwen 00 (black) and FF_{16} (white). A graph that shows, for each grey level, the number of times that level occurs in the image is called the histogram of the image. Figure 13.11 shows a typical histogram for an image of black parts on a white conveyor.

In Figure 13.11 we note two distinct peaks, one at grey level 3, almost pure black, and one at grey level 193, bright white. With the exception of noise pixels, every point in the image belongs to one of these regions. A good threshold, then, is anywhere between the two peaks.

Histograms are seldom as "nice" as the one in Figure 13.11 and some additional processing is generally needed (Chew and Kaneko, 1972; Rosenfeld and Kak, 1976). However, the philosophy of his-

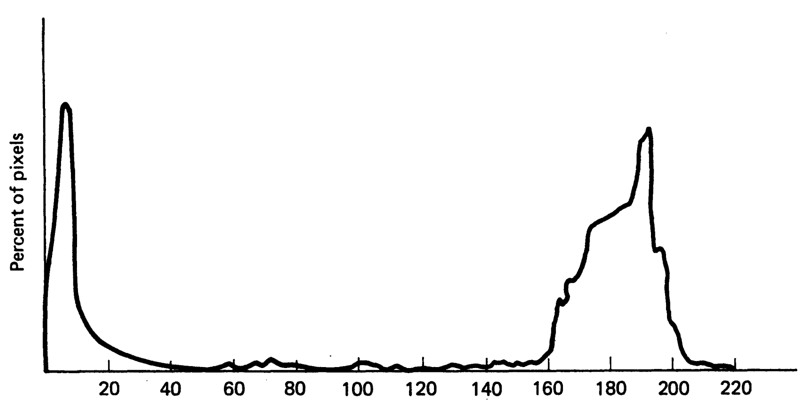


Figure 13.11 Histogram of a picture with a mixture of pure black and pure white.

togram-based thresholding is the same: find two peaks in the histogram and choose the threshold to be between them.

In general, different thresholds are used in different areas of the picture. In many industrial environments, the lighting may be extremely well controlled. With such control, the best thresholds will be constant over time and may be chosen interactively during system set up. In other, more variable, situations, the computer may be required to analyze the distribution of grey levels over the area of interest to choose an appropriate threshold.

Region Labeling

Let us assume, for now, that a good threshold has been chosen and that our picture has been partitioned into regions of pure black and pure white, as shown in Figure 13.10(a). The production of a segmented picture such as Figure 13.10(b) requires an analysis of connectedness. That is, a pixel is in region i if it is above threshold and is adjacent to a pixel in region i. Since regions may curve and fork, the analysis cannot be as simple as starting at the top and marking connected pixels going down. Instead a more sophisticated technique is needed.

One such algorithm is known as *region growing*. It utilizes a label memory corresponding to the frame buffer just as Figure 13.10(b) corresponds to 13.10(a). In this description, we will refer to "black" pixels as object and "white" as background.

Initially, each cell in the label memory M is set to zero. We will refer to the picture memory as P. Thus, the grey scale of a point with coordinates $\langle x, y\rangle$ is $P(x, y)$, and the labeling operation can be written $M(x, y) \leftarrow N$ for some label number N.

Algorithm Grow

This algorithm implements region growing by using a push down stack on which to temporarily keep the coordinates of pixels in the region.

1. Find an unlabeled black pixel; that is, $M(x, y) = 0$. Choose a new label number for this region, call it N. If all pixels have been labeled, stop.
2. If $P(x-1, y)$ is black and $M(x-1, y) = 0$, push $\langle x-1, y\rangle$ onto the stack.
   If $P(x+1, y)$ is black and $M(x+1, y) = 0$, push $\langle x+1, y\rangle$ onto the stack.

If $P(x, y - 1)$ is black and $M(x, y - 1) = 0$, push $\langle x, y - 1 \rangle$ onto the stack.

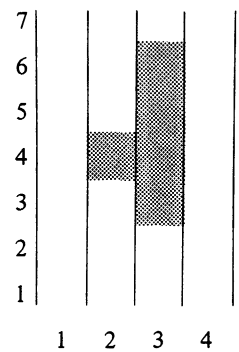
If $P(x, y + 1)$ is black and $M(x, y + 1) = 0$, push $\langle x, y + 1 \rangle$ onto the stack.

3. $M(x, y) \leftarrow N$.
4. Choose a new $\langle x, y \rangle$ by popping the stack.
5. If the stack is empty, go to 1, else go to 2.

This labeling operation results in a set of connected regions, each assigned a unique label number. To find the region to which any given pixel belongs, the computer has only to interrogate the corresponding location in the M memory and read the region number.

Example 13.3 Applying Region Growing

The figure below shows a 4×7 array of pixels. Assume the initial value of $\langle x, y \rangle$ is $\langle 2, 4 \rangle$. Apply algorithm "grow" and show the contents of the stack and M each time step 3 is executed. Let the initial value of N be 1.



Solution:

Pass 1: Immediately after execution of step 3.

Stack: $\langle 3, 4 \rangle \leftarrow$ top

$$M = \begin{array}{c|c|c|c|c|}
7 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 \\
4 & 0 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\multicolumn{1}{c}{} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4}
\end{array}$$

Pass 2:

Stack: $\langle 3, 5 \rangle \leftarrow$ top
$\langle 3, 3 \rangle$

$$M = \begin{array}{r|c|c|c|c|}
7 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 \\
4 & 0 & 1 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\multicolumn{1}{r}{} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4}
\end{array}$$

Pass 3:

Stack: $\langle 3, 6 \rangle \leftarrow$ top
$\langle 3, 3 \rangle$

$$M = \begin{array}{r|c|c|c|c|}
7 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 0 \\
4 & 0 & 1 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\multicolumn{1}{r}{} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4}
\end{array}$$

Pass 4:

Stack: $\langle 3, 3 \rangle \leftarrow$ top

$$M = \begin{array}{r|c|c|c|c|}
7 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 0 \\
5 & 0 & 0 & 1 & 0 \\
4 & 0 & 1 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\multicolumn{1}{r}{} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4}
\end{array}$$

Pass 5:

Stack: Empty

$$M = \begin{array}{c|c|c|c|c|}
7 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 0 \\
5 & 0 & 0 & 1 & 0 \\
4 & 0 & 1 & 1 & 0 \\
3 & 0 & 0 & 1 & 0 \\
2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\multicolumn{1}{c}{} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4}
\end{array}$$

This region-growing algorithm is just one of several strategies for performing *connected component analysis*. Other strategies exist which are faster than the one described, including some that run at raster scan rates (Snyder and Savage, 1982). We will now consider techniques for making use of this information.

13.5 SHAPE DESCRIPTORS

In the process of generating the segmented version of a picture, the computer performs a region-growing operation that acts on each pixel in the region. In so doing, the computer can easily keep track of the area. Area is one of many features that can help us to distinguish one type of object from another. For example, the image of a connecting rod typically occupies more area (more black pixels) than does the image of a valve. Thus, by measuring the area of a region, we may discern the type of object.

In this section we will present a few of the many other features which may be used to characterize regions (Ballard and Brown, 1983).

13.5.1 Features

Average grey value: In the case of black and white "silhouette" pictures, this is simple to compute.

Maximum grey value: Is straightforward to compute.

Minimum grey value: Is straightforward to compute.

Area: Comes directly from the region-growing algorithm.

Perimeter: Several different definitions exist. Probably the simplest is a count of all pixels in the region that are adjacent to a pixel not in the region.

Diameter: The diameter describes the maximum chord—the distance between those two points on the boundary of the region whose mutual distance is maximum (Snyder and Tang, 1980; Shamos, 1975).

Thinness: Two definitions for thinness exist: $T_A = P^2/A$ measures the ratio of the squared perimeter to the area; $T_B = D/A$ measures the ratio of the diameter to the area. Figure 13.12 compares these two measurements on example regions.

Center of gravity: The x and y coordinates of the center of gravity may be written

$$m_x = \frac{1}{N}\sum x$$

$$m_y = \frac{1}{N}\sum y$$

for all points in a region with N points.

X-Y aspect ratio: See Figure 13.13(a). The aspect ratio is the length/width ratio of the bounding rectangle. This is simple to compute.

Minimum aspect ratio: See Figure 13.13(b). Again, a length/width, but much more computation is required to find the minimum such rectangle.

Moments: A moment of order $p + q$ may by defined on a region as

$$m_{xy} = \sum x^p y^q$$

This definition assumes that the region is uniform in grey value and that grey value is arbitrarily set to 1. The area is then m_{00}, and we

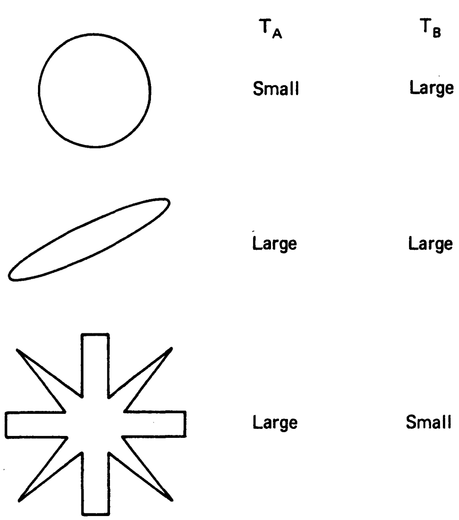


Figure 13.12 Results of applying two different thinness measurements on various regions.

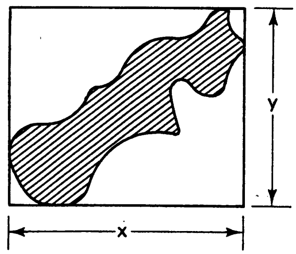


Figure 13.13(a) y/x is the aspect ratio using one definition, with horizontal and vertical sides to the bounding rectangle. This definition is very sensitive to rotations.

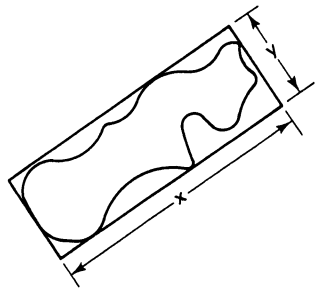


Figure 13.13(b) y/x is the minimum aspect ratio. This definition is invariant to rotation.

find that the center of gravity is

$$m_x = \frac{m_{10}}{m_{00}} \qquad m_y = \frac{m_{01}}{m_{00}}$$

We can now define as many features as we wish by choosing higher orders of moments or combinations thereof. Of particular interest are the *invariant moments* (Gonzalez and Wintz, 1977). Those moments have the characteristics that they are invariant to translation, rotation, and scale change, which means that we get the same number, even though the image may be moved, rotated, or zoomed.
The first invariant moment is

$$\frac{m_{00}m_{20} - m_{10}^2 + m_{00}m_{02} - m_{01}^2}{m_{00}^3}$$

There are six others (Gonzalez and Wintz, 1977).

Convex discrepancy: If one were to stretch a rubber band around a given region, the region that would result would be the *convex hull*

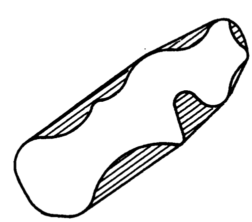


Figure 13.14 Convex hull of a region. The shaded area is the convex discrepancy.

(Figure 13.14). The difference in area between a region and its convex hull is the convex discrepancy. See Shamos (1975) for fast algorithms for computing the convex hull.

Number of holes: One final feature that is very descriptive and reasonably easy to compute is the number of holes in a region.

In this section, several features were defined that could be used to quantify the shape of a region. Some, like the moments, are easy measurements to make. Others, such as the diameter or the convex discrepancy, require development of fairly sophisticated algorithms to avoid extremely long computation times. Space does not permit a discussion of those algorithms here, but the reader may find adequate direction in the sources cited in the reference list.

13.6 USING SHAPE DESCRIPTORS

The features described in the previous section can be used in many different ways to identify, locate, and orient parts. A thorough discussion of those techniques is the basis of entire books. In this section we simply introduce the reader to some of the basic concepts of machine vision and to whet his or her intellectual appetite for more knowledge about this potentially very productive field. For that reason, we will consider only the problem of recognizing parts and leave the issues of location and orientation for another text. As before, we consider only parts which can be recognized by their two-dimensional silhouettes.

We define a *feature vector* **x** as an ordered d-tuple $\langle x_1 x_2 \cdots x_d \rangle$ in which each element is a scalar feature, as discussed in Section 13.5.

The *event* w_i means that the object being viewed by the camera belongs to class $w_i (i = 1 \cdots c)$. Typical classes of objects would be valves, rods, bolts, covers, and so on.

$P(w_i)$ is the probability that the object being viewed belongs to class i, before any measurements have been made. $P(w_i)$ is the a priori probability.

Given a measurement **x** made on the object, we can define the conditional probability $P(w_i|\mathbf{x})$ that the object is in class i, given that the measurement vector has value **x**.

Finally, the *conditional probability density* $p(\mathbf{x}|w_i)$ is a function of the feature that represents the probability that a member of class w_i will have feature value **x**. Note that a different function $p(\mathbf{x}|w)$ exists for each different w. Figure 13.15 shows the probability density functions for one feature and different classes. Since **x** is a vector quantity, $p(\mathbf{x}|w)$ is typically a vector function. For simplicity, Figure 13.15 has

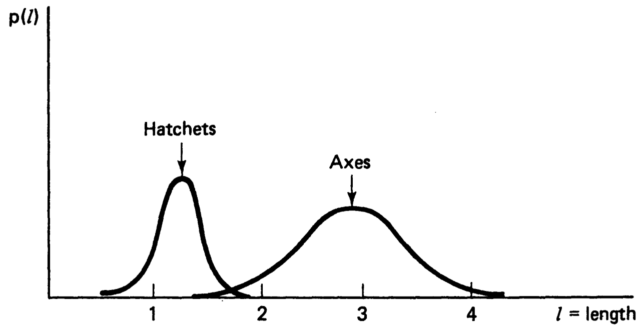


Figure 13.15 Probability density function for length of hatchets and axes. From this graph, we see that an object of length 2 is more likely to be an axe than a hatchet.

treated only one of the components of the feature vector, so that the corresponding density functions may be shown as graphs of only two dimensions.

Now, we can state *Bayes' rule*, which relates these probabilities.

$$P(w_i|\mathbf{x}) = \frac{p(\mathbf{x}|w_i)\,P(w_i)}{\sum_j p(\mathbf{x}|w_j)\,P(w_j)} \tag{13.5}$$

This equation relates the conditional probability of class w_i (which is what we want) to the conditional density of the measurement (which we know from past experience) and includes the a priori probabilities of each class (also known from prior experience).

We will assume all the classes are equally likely and, therefore, simply ignore $P(w_i)$. Furthermore, we observe that the denominator is a normalizing factor and does not help us to distinguish one class from another. Hence we can define a decision rule:

$$\text{Decide class } w_i \text{ if, for all } j \neq i$$

$$p(\mathbf{x}|w_i) > p(\mathbf{x}|w_j) \tag{13.6}$$

Exactly how to determine and represent $p(\mathbf{x}|w_j)$ can be a subject unto itself. In the most straightforward approach, it is determined experimentally and is stored in a tabular form. While convenient for scalars, this approach is awkward for $\mathbf{x}$ vectors of high dimensionality. In these cases, we usually approximate the experimental $p(\mathbf{x}|w)$ with a continuous function. A Gaussian function is often chosen because of its mathematical simplicity and because the Gaussian approximates naturally occurring densities quite well.

In the Gaussian approximation, we use

$$p(\mathbf{x}|w_i) = \frac{1}{(2\pi)^{d/2}\,|C_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\, C_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right] \tag{13.7}$$

where d is the dimension of the vector $\mathbf{x}$. $\boldsymbol{\mu}_i$ is the mean vector of a *training set* X_i of samples, all known to belong to class i, and N_i is the number of samples in X_i:

$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{\mathbf{x}\in \mathrm{X}_i} \mathbf{x} \tag{13.8}$$

The covariance matrix C_i is determined, using that Gaussian assumption, by

$$C_i = \frac{1}{N_i} \sum_{\mathbf{x}\in \mathrm{X}_i} (\mathbf{x}-\boldsymbol{\mu}_i)\,(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \tag{13.9}$$

With this introduction, we can define a strategy developing a statistical pattern recognition system for recognizing parts.

1. Define features and develop algorithms for measuring features.
2. Choose a large set of objects of type A and measure the features, determine mean and variance for the features for class A. Repeat for classes B, C, and so on.
3. Given an unknown x, a part on the conveyor, measure the feature vector $\mathbf{x}$, and apply Eq. 13.7 once for each class. The application returning the highest value is the most likely class.

This discussion has, of course, glossed over most of the field of statistical pattern recognition. Many books are available which go into more detail, including Duda and Hart (1973), Ballard and Brown (1983).

Example 13.4: Object Recognition Using Shape Descriptors

An unknown object is measured by a vision system to have a length of 2.5 units.

A large number of flanges, all of the same type, have been measured and found to have an average (mean) length of 2.1 units and a variance in length of 0.8. Similarly, a large number of gaskets have been measured and found to have an average length of 2.8 units and a variance in length of 1.3.

Is the unknown most likely a flange or a gasket?

Solution: Using the Gaussian assumption described in Eq. 13.7, we find

$$p(2.5\,|\,\text{flange}) = \frac{1}{\sqrt{2\pi(0.8)}} \exp\frac{-(2.5-2.1)^2}{2\cdot(0.8)} = 0.404$$

$$p(2.5\,|\text{gasket}) = \frac{1}{\sqrt{2\pi(1.3)}} \exp \frac{-(2.5 - 2.8)^2}{2 \cdot 1.3} = 0.338$$

Using Eq. 13.6, we make the decision that the unknown is more likely to be a flange, since that decision has the higher probability of being correct. However, two observations are in order. First, the two results were very close; therefore, our confidence in the accuracy of our decision is not very high. We need to make another measurement, perhaps the perimeter of the object image, to improve our decision-making capabilities. Second, this analysis has assumed that the Gaussian distribution modeled in Eq. 13.7 accurately reflects the shape of the probability density function. This assumption may not be correct for the length of flanges and gaskets and should be tested before being put into operation.

13.7 STRUCTURED ILLUMINATION

For a number of years, popular fantasy painted the picture of the robot with human vision or better as being just around the corner. Today, we know better. Computer vision is a very difficult problem that is not likely to be solved in general in the foreseeable future. However, in the industrial environment, general-purpose vision is not required and probably not even desired. We need only to locate and/or identify a very restricted set of objects. Furthermore, in the industrial environment, we have one other tremendous advantage: we can control the lighting.

The strategy that takes advantage of control of lighting to make the vision problem easier is called *structured illumination*, and it takes many forms. In this section, we will provide three examples, each of which uses controlled lighting to considerable advantage to solve a different problem. One technique eliminates problems with object reflectivity, a second finds objects by triangulation, and a third uses controlled lighting for inspection.

13.7.1 Silhouetting

Many industrial parts have the property that they are uniquely identifiable by their silhouette. That is, neither three-dimensional nor grey scale information is required for the part to be uniquely identified and its orientation determined. In these cases, particularly simple techniques exist for extracting and processing the object's silhouette.

For example, in General Motors' CONSIGHT system (Ward et al., 1979) a linear sensor array is used rather than a two-dimensional camera. Such arrays typically have more resolution per line than a camera; 2000 pixels on a line is not uncommon. To form a two-dimensional picture, either the part or the camera may be moved or a mirror may move the image of the part. In CONSIGHT, the part is moved, a trivial task since the sensor is focused on a conveyor.

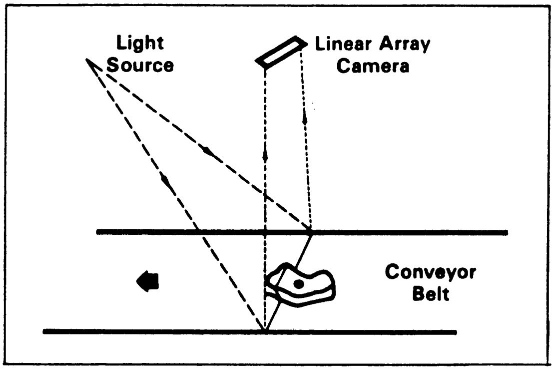


Figure 13.16 (Ward) The camera is positioned to image a line across the conveyor belt. (*Robotics Today, used with permission*)

The structured illumination concept applies to CONSIGHT in the way in which the part is illuminated. A narrow stripe of light is focused at an angle on the conveyor at exactly the point of focus of the sensor. With an empty conveyor, the sensor detects a bright bar. When a part moves into the field of view, however, the stripe of light shines on the part (Figure 13.16). Since the part has depth and the light is at an angle, the light stripe is displaced horizontally and the sensor detects darkness. By taking a series of stripe images as the part moves, the system can build up a profile of the part.

The system as implemented (Figure 13.17) actually uses two light projectors. This eliminates the problem of shadows being erroneously identified as parts.

The reflectivity of parts, especially metallic ones, is one of the most difficult problems in image analysis. To see that this is a problem, we should recognize that a shiny metallic part is really a mirror. Consider the intrinsic difficulty in analyzing the image of a mirror. Since the CONSIGHT system makes use of only the displacement of the light

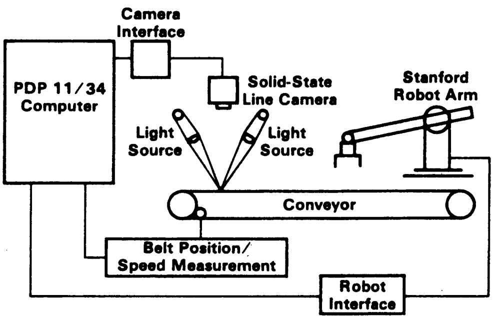


Figure 13.17 (Ward) Schematic diagram of CONSIGHT.

stripe, it has the additional advantage that it is independent of the surface reflectivity of the part.

13.7.2 Range Triangles

In many systems, the concept of the range triangle has been used to determine three-dimensional position and orientation of parts. In Page, Snyder, and Rajala (1983), a system is described which determines the position and orientation (often these two terms are lumped together and are referred to as *pose*) of steam turbine blades.

Such blades have shiny metallic surfaces and, therefore, reflect most of the light in undesirable ways. However, the surface scatters enough light back to the camera to detect a horizontal bar, if such a bar is projected on the blades.

Figure 13.18 shows a light source projecting a narrow (0.2 centimeters) stripe of light across a scene containing a blade. The scene detected by the camera is shown in Figure 13.19. The horizontal light stripe appears curved to the camera when it is reflected off the blade. The curved stripe is easy to detect in the image by using some of the thresholding strategies described earlier in this chapter.

The three-dimensional position of the two corners of the blade are determined by making use of a *range triangle*, as shown in Figure 13.20. The camera position is known, as is the position of the light projectors. The angle of the projector with respect to the scene is also known by a calibration procedure. From the camera image, the computer finds the angle α, which is all that is needed to determine the distance d to the part.

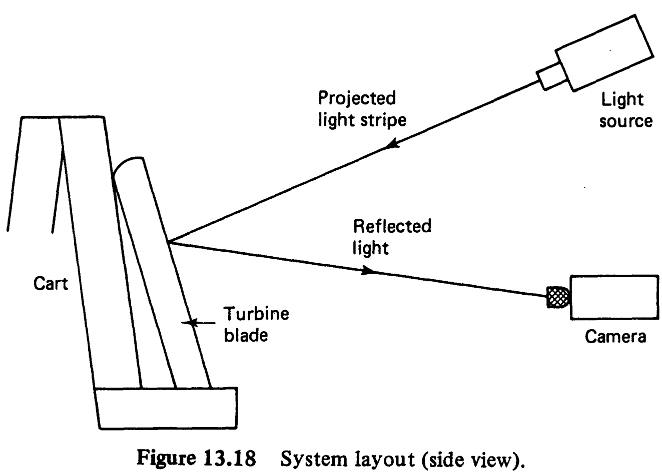


Figure 13.18 System layout (side view).

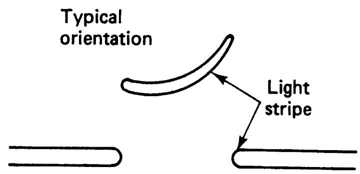


Figure 13.19(a) Camera's view of a single blade.

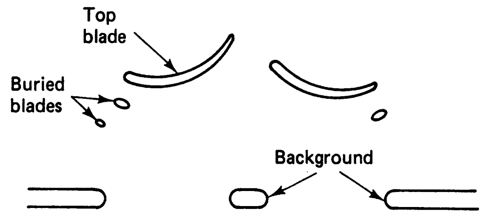


Figure 13.19(b) Camera's view of multiple blades in a cart.

Again, the combination of controlled lighting and simple image processing has provided a reasonable operational industrial vision system.

13.7.3 Range Sensors

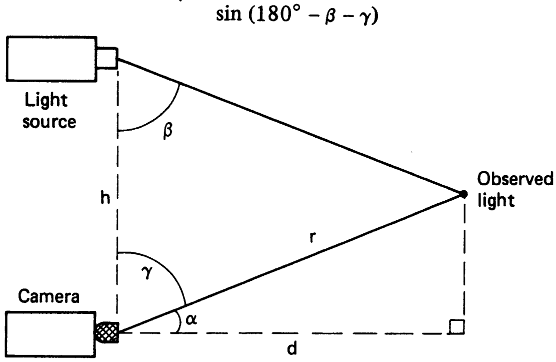
One of the most exciting prospects of current research in image analysis is the development of special sensors which explicitly determine a *range image*. By range image we mean a two-dimensional array of pixels, exactly like a conventional "luminance" image. However, the number stored in each pixel location represents the Euclidean distance from the sensor to the surface being viewed, rather than the relative brightness of the surface at that point. Such sensors may employ various structured lighting strategies, such as those described earlier, or

Figure 13.20 Range triangle. The depth of the point was derived from the following trigonometic relationships:

$$\gamma = 90° - \alpha \qquad \frac{\sin(\beta)}{r} = \frac{\sin(180° - \beta - \gamma)}{h}$$

Recall that h and β were known and that α was measured by the camera. Then,

$$r = \frac{[h \times \sin(\beta)]}{\sin(180° - \beta - \gamma)}$$



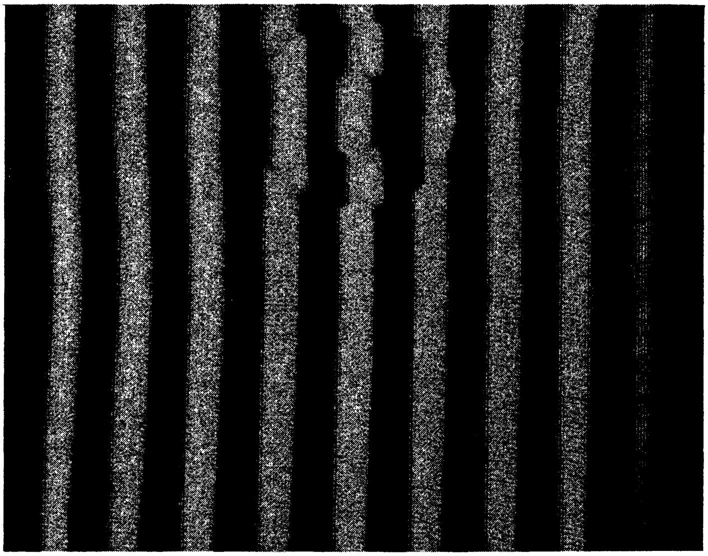


Figure 13.21 (Porter & Mundy) Sinusoidally modulated light. (*Courtesy 1982 IEEE*)

even more sophisticated techniques, such as pulsed lasers. Independent of the internal strategy of the sensor, the result is a range picture. Such a picture gives surface pose information immediately and, thus, can readily provide the three-dimensional information needed by robots for grasping.

Such sensors are still under development, as are techniques for making use of such information. Porter and Mundy (1982) describe one such sensor that uses sinusoidally modulated light patterns (Figure 13.21). The sensor makes use of the difference of two patterns projected at different wavelengths to reduce the undesirable effects of surface reflectivity. The system reported is used in an inspection application to identify small defects on parts.

13.8 CONCLUSION

Vision provides the robot with its most flexible and powerful capability. In this chapter we have presented a few of the currently relevant research and development topics in the area of industrial machine vision. With the introduction and terminology provided here, the reader can more easily follow the developments in this exciting and growing field.

13.9 SYNOPSIS

Vocabulary

You should know the definition and application of the following terms:

aliasing
aspect ratio
blanking
block thresholding
blur
composite video
contouring
convex discrepancy
convex hull
decision rule
dynamic range
enhancement
feature
feature vector
frame buffer
frame grabber
frame time
Gaussian density
geometric distortion
histogram
industrial machine vision
invariant moment
lag
moment
motion blur
parabolic distortion
pel
pixel
quantization
range sensor
raster scan
region growing
region labeling
resolution
restoration
sampling
sampling theorem

segmentation
shape descriptor
silhouetting
structured illumination
thinness
thresholding
undersampled
videcon
vignetting
warping

Notation

Symbols	Meanings
i	Incident light intensity
t_f	Frame time
q	Charge
$f(x, y)$	The true image
$g(x, y)$	The image perceived by the sensor system
D	Distortion operator
m_{xy}	A moment
T_A	One definition of thinness
T_B	Another definition of thinness
$\mathbf{x}$	A feature vector
$P(w_i)$	Probability of event w_i
$p(\mathbf{x}\|w_i)$	Probability density of making measurement $\mathbf{x}$, given that event w_i has occurred
$\boldsymbol{\mu}_i$	Mean of a Gaussian (normal) distribution
C_i	Covariance of a Gaussian distribution
$\mathbf{X}_i$	Training set

13.10 REFERENCES

ANDREWS, H. C., and HUNT, B. R. *Digital image restoration*. Englewood Cliffs, NJ.: Prentice-Hall, 1977.

BALLARD, D. H., and BROWN, C. M. *Computer vision*. Englewood Cliffs, N.J.: Prentice-Hall, 1983.

CASTLEMAN, R. R. *Digital image processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1977.

CHIEN, R. T., and SNYDER, W. E. "Hardware for Visual Image Processing." *IEEE Transactions on Circuits and Systems*, *22*, June 1975.

CHOW, C. K., and KANEKO, T. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, August 1972.

COOLEY, J. W., and TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. *Math. of Computation*, Vol. 19, April 1965, pp. 299–301.

DUDA, R. O., and HART, P. E. *Pattern classification and scene analysis.* New York: John Wiley, 1973.

FU, K. S., and ICHIKAWA, T. *Special computer architectures for pattern processing.* Boca Raton, Fl.: CRC Press, 1982.

FU, K. S., ed. *Applications of Pattern Recognition.* Boca Raton, Fl.: CRC Press, 1980.

GONZALEZ, R. C., and WINTZ, P. *Digital image processing.* Reading, Mass.: Addison-Wesley, 1977.

PAGE, N., SNYDER, W. E., and RAJALA, S. A. Turbine blade image processing system. First International Conference on Advanced Software in Robotics, Liège, Belgium, May, 1983.

PORTER, G. B., and MUNDY, J. L. A noncontact profile sensing system for visual inspection. 6th International Conference on Pattern Recognition, Munich, Oct. 1982.

PRATT, W. K. *Digital image processing.* New York: John Wiley, 1978.

ROSENFELD, A., and KAK, A. C. *Digital picture processing.* New York: Academic Press, 1976.

SHAMOS, M. I. Geometric complexity. *Proceedings of the 7th ACM Symposium on Theory of Computing*, May 1975.

SNYDER, W. E., and SAVAGE, C. "Content-Addressable Read/Write Memories for Image Analysis," pp. 963–968. *IEEE Transactions on Computers*, October 1982.

——, and TANG, D. A. "Finding the extrema of a region. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 3, pp 266–269, May 1980.

WARD, M. R., ROSSOL, L., HOLLAND, S. W., and DEWAR, R. Consight: An adaptive robot with vision. pp. 26–32. *Robotics Today*, Summer 1979.

13.11 PROBLEMS

1. One type of slow scan television produces one frame of 525 lines every second. Determine the sampling rate necessary to produce an image with 525 lines and 256 pixels of video on each line. Assume that blanking occupies 18 percent of of the line time.
2. Assume that you are given a memory which can acquire data at the rate of 1 pixel every 200 ns. Discuss the options you have in trading off dynamic range and resolution, given that you must acquire data at video rates.

3. For the system described in problem 2, design a circuit that will acquire 2-bit pixels and pack them into 8-bit memory words to be read into the memory. Assume that the memory will accept data when it is given a MSTROBE signal, which you must provide.
4. A pattern consisting of 500 vertical black bars on a white background is scanned by a conventional, black and white, NTSC TV camera. The width of the white and black areas is equal, and the array of bars encompasses the entire frame. The TV signal is sampled by a clock scanning at 103 ns per sample. This results in an undersampled signal, resulting in an aliased signal at a lower frequency. What will be this frequency (or the dominant frequency component of the aliased signal). Note: The solution of this problem may require research into, or knowledge of, signal processing literature not included in this text.
5. The equation for contrast stretching (Section 13.2) could be written $I_{new'} = (I_{old} - I_{min})\,\alpha$. A picture is determined to have a maximum brightness of 200 and a minimum brightness of 60. Determine α. Show that this equation will stretch the contrast to between 0 and 255.
6. Determine the memory capacity required to store a 512 X 512 array of 6-bit pixels. Discuss alternative memory organizations.
7. The figure below shows a 7 X 4 array of pixels. Determine a "good" threshold to separate black from white pixels. Justify your decision based on a histogram.

61	55	60	61
62	63	57	58
56	58	38	32
55	35	36	37
37	34	39	35
40	39	38	39
35	34	39	40

8. A bimodal histogram (that is, a histogram with two peaks) can be modeled as the sum of two Gaussian density functions.
   (a) Simplify Eq. 13.7 for the case where $d = 1$. In this case, the vectors become scalars and multiplication by the inverse of a matrix becomes division by a constant. (You may wish to look up the form of the normal density function.)
   (b) Represent a bimodal histogram as the sum of two such functions.
   (c) Suppose that we have an experimental histogram such as that illustrated in Figure 13.11, which we wish to model analytically by the equation derived in part 2. What are the unknowns? Discuss how one might go about solving for these unknowns.
9. Algorithm *grow* (Section 13.4.1) gives one approach to labeling of connected components. This approach requires two frame buffer memories: one to hold the image and one to hold the labels. In addition, use of the stack is rather

slow. An alternative algorithm would be to label pixels as they come from the camera, in raster scan order. That is, a black pixel at $\langle x, y \rangle$ would be assigned the same label as the pixel at $\langle x - 1, y \rangle$ if that pixel were black, or assigned the same label as the pixel at $\langle x, y - 1 \rangle$ if that pixel were black.

(a) Develop a flow chart for assigning labels in this way

(b) Discuss any problems which arise. For example, consider U-shaped regions.

10. The following figure shows a 4 X 7 region of an image, containing a black figure on a white background.

		**	
		**	
	**	**	
		**	

(a) What is the area of this region?

(b) What is the diameter of the region. Discuss any ambiguities in the definition.

(c) What is the perimeter? Propose at least two definitions.

(d) Determine and compare the two definitions of thinness when applied to this region.

(e) Determine the center of gravity.

(f) Find the smallest rectangle that encloses the region.

(g) Determine the convex hull and convex discrepancy.

11. Example 13.4 presents a technique for distinguishing between flanges and gaskets using only length. In this problem, we again distinguish flanges from gaskets, but using both length and weight. We define

$$\mathbf{x} = \begin{bmatrix} \text{length} \\ \text{weight} \end{bmatrix}$$

Training sets have been used, and statistics developed as follows:

$$\boldsymbol{\mu}_{\text{gasket}} = \begin{bmatrix} 2.8 \\ 4.0 \end{bmatrix} \quad \boldsymbol{\mu}_{\text{flange}} = \begin{bmatrix} 2.1 \\ 3.0 \end{bmatrix}$$

$$C_{\text{gasket}} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \quad C_{\text{flange}} = \begin{bmatrix} 0.02 & 0.03 \\ 0.03 & 0.06 \end{bmatrix}$$

Apply Eq. 13.7 and distinguish whether the measurement

$$\mathbf{x} = \begin{bmatrix} 2.2 \\ 3.8 \end{bmatrix}$$

is most likely a flange or a gasket.