

Image Formation & Display

Images are a description of how a parameter varies over a surface. For example, standard visual images result from light intensity variations across a two-dimensional plane. However, light is not the only parameter used in scientific imaging. For example, an image can be formed of the *temperature* of an integrated circuit, *blood velocity* in a patient's artery, *x-ray emission* from a distant galaxy, *ground motion* during an earthquake, etc. These exotic images are usually converted into conventional pictures (i.e., light images), so that they can be evaluated by the human eye. This first chapter on image processing describes how digital images are formed and presented to human observers.

Digital Image Structure

Figure 23-1 illustrates the structure of a digital image. This example image is of the planet Venus, acquired by microwave radar from an orbiting space probe. Microwave imaging is necessary because the dense atmosphere blocks visible light, making standard photography impossible. The image shown is represented by 40,000 samples arranged in a two-dimensional array of 200 columns by 200 rows. Just as with one-dimensional signals, these rows and columns can be numbered 0 through 199, or 1 through 200. In imaging jargon, each sample is called a **pixel**, a contraction of the phrase: *picture element*. Each *pixel* in this example is a single number between 0 and 255. When the image was acquired, this number related to the amount of microwave energy being reflected from the corresponding location on the planet's surface. To display this as a *visual image*, the value of each pixel is converted into a **grayscale**, where 0 is black, 255 is white, and the intermediate values are shades of gray.

Images have their information encoded in the **spatial domain**, the image equivalent of the time domain. In other words, features in images are represented by *edges*, not *sinusoids*. This means that the spacing and number of pixels are determined by how small of features need to be seen,

rather than by the formal constraints of the sampling theorem. Aliasing *can* occur in images, but it is generally thought of as a nuisance rather than a major problem. For instance, pinstriped suits look terrible on television because the repetitive pattern is greater than the Nyquist frequency. The aliased frequencies appear as light and dark bands that move across the clothing as the person changes position.

A "typical" digital image is composed of about 500 rows by 500 columns. This is the image quality encountered in television, personnel computer applications, and general scientific research. Images with fewer pixels, say 250 by 250, are regarded as having unusually poor resolution. This is frequently the case with new imaging modalities; as the technology matures, more pixels are added. These low resolution images look noticeably unnatural, and the individual pixels can often be seen. On the other end, images with more than 1000 by 1000 pixels are considered exceptionally good. This is the quality of the best computer graphics, high-definition television, and 35 mm motion pictures. There are also applications needing even higher resolution, requiring several thousand pixels per side: digitized x-ray images, space photographs, and glossy advertisements in magazines.

The strongest motivation for using lower resolution images is that there are *fewer* pixels to handle. This is not trivial; one of the most difficult problems in image processing is managing massive amounts of data. For example, one second of digital audio requires about eight *kilobytes*. In comparison, one second of television requires about eight *Megabytes*. Transmitting a 500 by 500 pixel image over a 33.6 kbps modem requires nearly a minute! Jumping to an image size of 1000 by 1000 *quadruples* these problems.

It is common for 256 **gray levels** (quantization levels) to be used in image processing, corresponding to a single byte per pixel. There are several reasons for this. First, a single byte is convenient for data management, since this is how computers usually store data. Second, the large number of pixels in an image compensate to a certain degree for a limited number of quantization steps. For example, imagine a group of adjacent pixels alternating in value between digital numbers (DN) 145 and 146. The human eye perceives the region as a brightness of 145.5. In other words, images are very *dithered*. Third, and most important, a brightness step size of $1/256$ (0.39%) is smaller than the eye can perceive. An image presented to a human observer will not be improved by using more than 256 levels.

However, some images need to be stored with more than 8 bits per pixel. Remember, most of the images encountered in DSP represent nonvisual parameters. The acquired image may be able to take advantage of more quantization levels to properly capture the subtle details of the signal. The point of this is, don't expect to human eye to see all the information contained in these finely spaced levels. We will consider ways around this problem during a later discussion of brightness and contrast.

The value of each pixel in the digital image represents a small *region* in the continuous image being digitized. For example, imagine that the Venus

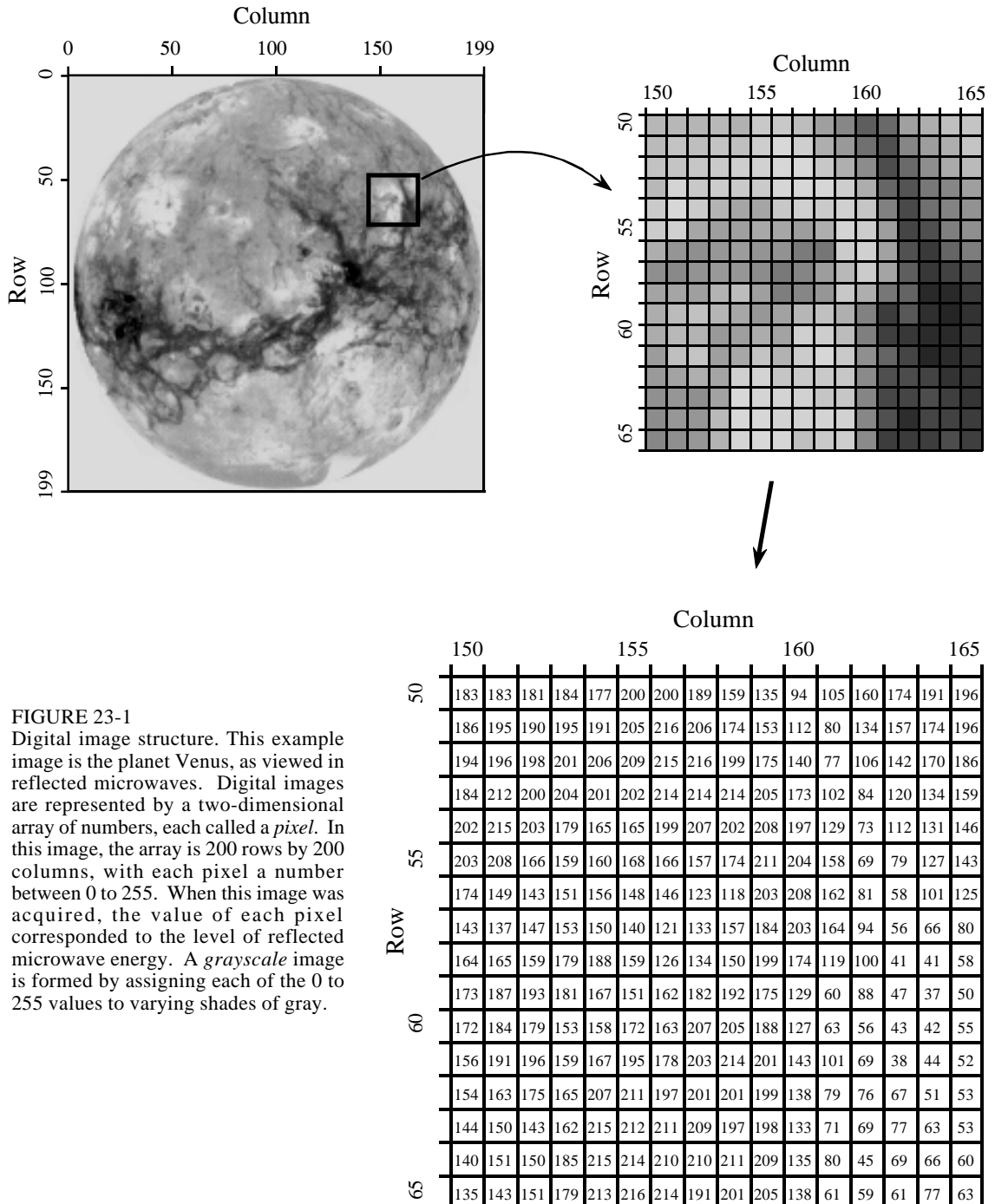


FIGURE 23-1

Digital image structure. This example image is the planet Venus, as viewed in reflected microwaves. Digital images are represented by a two-dimensional array of numbers, each called a *pixel*. In this image, the array is 200 rows by 200 columns, with each pixel a number between 0 to 255. When this image was acquired, the value of each pixel corresponded to the level of reflected microwave energy. A *grayscale* image is formed by assigning each of the 0 to 255 values to varying shades of gray.

probe takes samples every 10 meters along the planet's surface as it orbits overhead. This defines a square **sample spacing** and **sampling grid**, with each pixel representing a 10 meter by 10 meter area. Now, imagine what happens in a single microwave reflection measurement. The space probe emits

a highly focused burst of microwave energy, striking the surface in, for example, a circular area 15 meters in diameter. Each pixel therefore contains information about this circular area, regardless of the size of the sampling grid.

This region of the continuous image that contributes to the pixel value is called the **sampling aperture**. The size of the sampling aperture is often related to the inherent capabilities of the particular imaging system being used. For example, microscopes are limited by the quality of the optics and the wavelength of light, electronic cameras are limited by random electron diffusion in the image sensor, and so on. In most cases, the sampling grid is made approximately the same as the sampling aperture of the system. Resolution in the final digital image will be limited primary by the larger of the two, the sampling grid or the sampling aperture. We will return to this topic in Chapter 25 when discussing the spatial resolution of digital images.

Color is added to digital images by using three numbers for each pixel, representing the intensity of the three primary colors: red, green and blue. Mixing these three colors generates all possible colors that the human eye can perceive. A single byte is frequently used to store each of the color intensities, allowing the image to capture a total of $256 \times 256 \times 256 = 16.8$ million different colors.

Color is very important when the goal is to present the viewer with a true picture of the world, such as in television and still photography. However, this is usually not how images are used in science and engineering. The purpose here is to analyze a two-dimensional signal by using the human visual system as a *tool*. Black and white images are sufficient for this.

Cameras and Eyes

The structure and operation of the eye is very similar to an electronic camera, and it is natural to discuss them together. Both are based on two major components: a lens assembly, and an imaging sensor. The lens assembly captures a portion of the light emanating from an object, and focus it onto the imaging sensor. The imaging sensor then transforms the pattern of light into a video signal, either electronic or neural.

Figure 23-2 shows the operation of the lens. In this example, the image of an ice skater is focused onto a screen. The term *focus* means there is a one-to-one match of every point on the ice skater with a corresponding point on the screen. For example, consider a $1 \text{ mm} \times 1 \text{ mm}$ region on the tip of the toe. In bright light, there are roughly 100 trillion photons of light striking this one square millimeter area each second. Depending on the characteristics of the surface, between 1 and 99 percent of these incident light photons will be reflected in random directions. Only a small portion of these reflected photons will pass through the lens. For example, only about one-millionth of the reflected light will pass through a one centimeter diameter lens located 3 meters from the object.

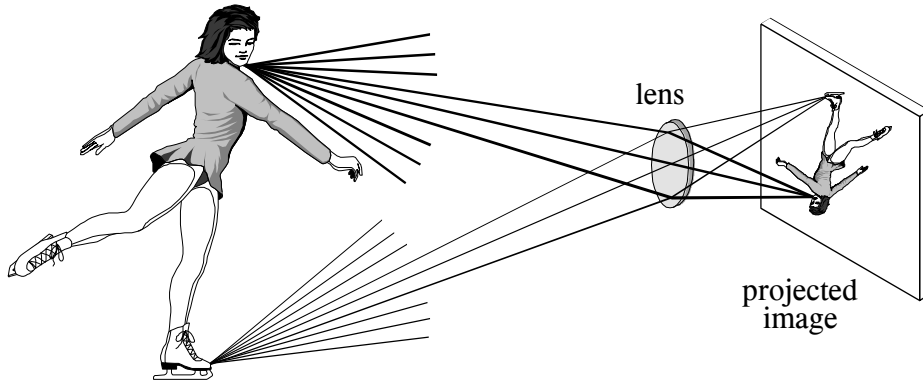


FIGURE 23-2

Focusing by a lens. A lens gathers light expanding from a point source, and force it to return to a point at another location. This allows a lens to project an image onto a surface.

Refraction in the lens changes the direction of the individual photons, depending on the location and angle they strike the glass/air interface. These direction changes cause light expanding from a single point to return to a single point on the projection screen. All of the photons that reflect from the toe *and* pass through the lens are brought back together at the "toe" in the projected image. In a similar way, a portion of the light coming from *any* point on the object will pass through the lens, and be focused to a corresponding point in the projected image.

Figures 23-3 and 23-4 illustrate the major structures in an electronic camera and the human eye, respectively. Both are light tight enclosures with a lens mounted at one end and an image sensor at the other. The camera is filled with air, while the eye is filled with a transparent liquid. Each lens system has two adjustable parameters: **focus** and **iris diameter**.

If the lens is not properly focused, each point on the object will project to a circular region on the imaging sensor, causing the image to be blurry. In the camera, focusing is achieved by physically moving the lens toward or away from the imaging sensor. In comparison, the eye contains two lenses, a bulge on the front of the eyeball called the cornea, and an adjustable lens inside the eye. The cornea does most of the light refraction, but is fixed in shape and location. Adjustment to the focusing is accomplished by the inner lens, a flexible structure that can be deformed by the action of the *ciliary muscles*. As these muscles contract, the lens flattens to bring the object into a sharp focus.

In both systems, the *iris* is used to control how much of the lens is exposed to light, and therefore the brightness of the image projected onto the imaging sensor. The iris of the eye is formed from opaque muscle tissue that can be contracted to make the *pupil* (the light opening) larger. The iris in a camera is a mechanical assembly that performs the same function.

The parameters in optical systems interact in many unexpected ways. For example, consider how the amount of available light and the sensitivity of the light sensor affects the *sharpness* of the acquired image. This is because the *iris diameter* and the *exposure time* are adjusted to transfer the proper amount of light from the scene being viewed to the image sensor. If more than enough light is available, the diameter of the iris can be reduced, resulting in a greater *depth-of-field* (the range of distance from the camera where an object remains in focus). A greater depth-of-field provides a sharper image when objects are at various distances. In addition, an abundance of light allows the exposure time to be reduced, resulting in less blur from camera shaking and object motion. Optical systems are full of these kinds of trade-offs.

An adjustable iris is necessary in both the camera and eye because the range of light intensities in the environment is much larger than can be directly handled by the light sensors. For example, the difference in light intensities between sunlight and moonlight is about one-million. Adding to this that reflectance can vary between 1% and 99%, results in a light intensity range of almost *one-hundred million*.

The **dynamic range** of an electronic camera is typically 300 to 1000, defined as the largest signal that can be measured, divided by the inherent noise of the device. Put another way, the maximum signal produced is 1 volt, and the rms noise in the dark is about 1 millivolt. Typical camera lenses have an iris that change the area of the light opening by a factor of about 300. This results in a typical electronic camera having a dynamic range of a few hundred thousand. Clearly, the same camera and lens assembly used in bright sunlight will be useless on a dark night.

In comparison, the eye operates over a dynamic range that nearly covers the large environmental variations. Surprisingly, the iris is not the main way that this tremendous dynamic range is achieved. From dark to light, the area of the pupil only changes by a factor of about 20. The light detecting nerve cells gradually adjust their sensitivity to handle the remaining dynamic range. For instance, it takes several minutes for your eyes to adjust to the low light after walking into a dark movie theater.

One way that DSP can improve images is by reducing the dynamic range an observer is required to view. That is, we do not want very light and very dark areas in the same image. A reflection image is formed from *two* image signals: the two-dimensional pattern of how the scene is *illuminated*, multiplied by the two-dimensional pattern of *reflectance* in the scene. The pattern of reflectance has a dynamic range of less than 100, because all ordinary materials reflect between 1% and 99% of the incident light. This is where most of the *image information* is contained, such as where objects are located in the scene and what their surface characteristics are. In comparison, the illumination signal depends on the light sources around the objects, but not on the objects themselves. The illumination signal can have a dynamic range of millions, although 10 to 100 is more typical within a single image. The illumination signal carries little interesting information,

FIGURE 23-3

Diagram of an electronic camera. Focusing is achieved by moving the lens toward or away from the imaging sensor. The amount of light reaching the sensor is controlled by the iris, a mechanical device that changes the effective diameter of the lens. The most common imaging sensor in present day cameras is the CCD, a two-dimensional array of light sensitive elements.

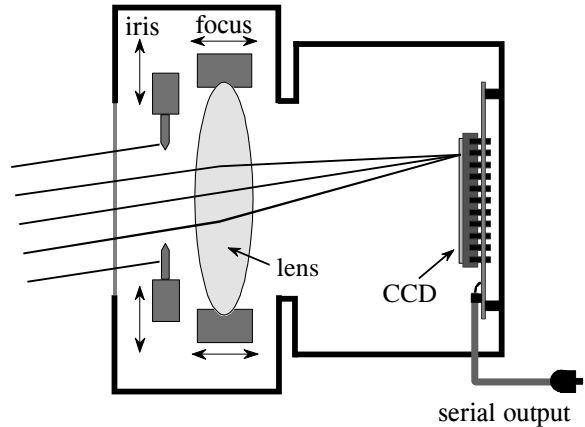
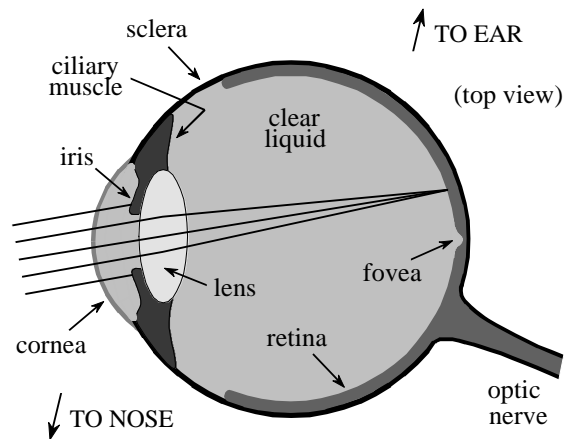


FIGURE 23-4

Diagram of the human eye. The eye is a liquid filled sphere about 3 cm in diameter, enclosed by a tough outer case called the sclera. Focusing is mainly provided by the cornea, a fixed lens on the front of the eye. The focus is adjusted by contracting muscles attached to a flexible lens within the eye. The amount of light entering the eye is controlled by the iris, formed from opaque muscle tissue covering a portion of the lens. The rear hemisphere of the eye contains the retina, a layer of light sensitive nerve cells that converts the image to a neural signal in the optic nerve.



but can degrade the final image by increasing its dynamic range. DSP can improve this situation by suppressing the illumination signal, allowing the reflectance signal to dominate the image. The next chapter presents an approach for implementing this algorithm.

The light sensitive surface that covers the rear of the eye is called the **retina**. As shown in Fig. 23-5, the retina can be divided into three main layers of specialized nerve cells: one for converting light into neural signals, one for image processing, and one for transferring information to the optic nerve leading to the brain. In nearly all animals, these layers are seemingly *backward*. That is, the light sensitive cells are in last layer, requiring light to pass through the other layers before being detected.

There are two types of cells that detect light: **rods** and **cones**, named for their physical appearance under the microscope. The rods are specialized in operating with very little light, such as under the nighttime sky. Vision appears very *noisy* in near darkness, that is, the image appears to be filled with a continually changing grainy pattern. This results from the image signal being very weak, and is not a limitation of the eye. There is so little light entering

the eye, the random detection of individual photons can be seen. This is called *statistical noise*, and is encountered in all low-light imaging, such as military night vision systems. Chapter 25 will revisit this topic. Since rods cannot detect color, low-light vision is in black and white.

The cone receptors are specialized in distinguishing color, but can only operate when a reasonable amount of light is present. There are three types of cones in the eye: red sensitive, green sensitive, and blue sensitive. This results from their containing different *photopigments*, chemicals that absorb different wavelengths (colors) of light. Figure 23-6 shows the wavelengths of light that trigger each of these three receptors. This is called **RGB encoding**, and is how color information leaves the eye through the optic nerve. The human perception of color is made more complicated by neural processing in the lower levels of the brain. The RGB encoding is converted into another encoding scheme, where colors are classified as: red *or* green, blue *or* yellow, and light *or* dark.

RGB encoding is an important limitation of human vision; the wavelengths that exist in the environment are lumped into only three broad categories. In comparison, specialized cameras can separate the optical spectrum into hundreds or thousands of individual colors. For example, these might be used to classify cells as cancerous or healthy, understand the physics of a distant star, or see camouflaged soldiers hiding in a forest. Why is the eye so limited in detecting color? Apparently, all humans need for survival is to find a *red* apple, among the *green* leaves, silhouetted against the *blue* sky.

Rods and cones are roughly 3 μm wide, and are closely packed over the entire 3 cm by 3 cm surface of the retina. This results in the retina being composed of an array of roughly $10,000 \times 10,000 = 100$ million receptors. In comparison, the optic nerve only has about one-million nerve fibers that connect to these cells. On the average, each optic nerve fiber is connected to roughly 100 light receptors through the connecting layer. In addition to consolidating information, the connecting layer enhances the image by sharpening edges and suppressing the illumination component of the scene. This biological image processing will be discussed in the next chapter.

Directly in the center of the retina is a small region called the **fovea** (Latin for *pit*), which is used for high resolution vision (see Fig. 23-4). The fovea is different from the remainder of the retina in several respects. First, the optic nerve and interconnecting layers are pushed to the side of the fovea, allowing the receptors to be more directly exposed to the incoming light. This results in the fovea appearing as a small depression in the retina. Second, only cones are located in the fovea, and they are more tightly packed than in the remainder of the retina. This absence of rods in the fovea explains why night vision is often better when looking to the *side* of an object, rather than directly at it. Third, each optic nerve fiber is influenced by only a few cones, proving good localization ability. The fovea is surprisingly small. At normal reading distance, the fovea only sees about a 1 mm diameter area, less than the size of a single letter! The resolution is equivalent to about a 20 \times 20 grid of pixels within this region.

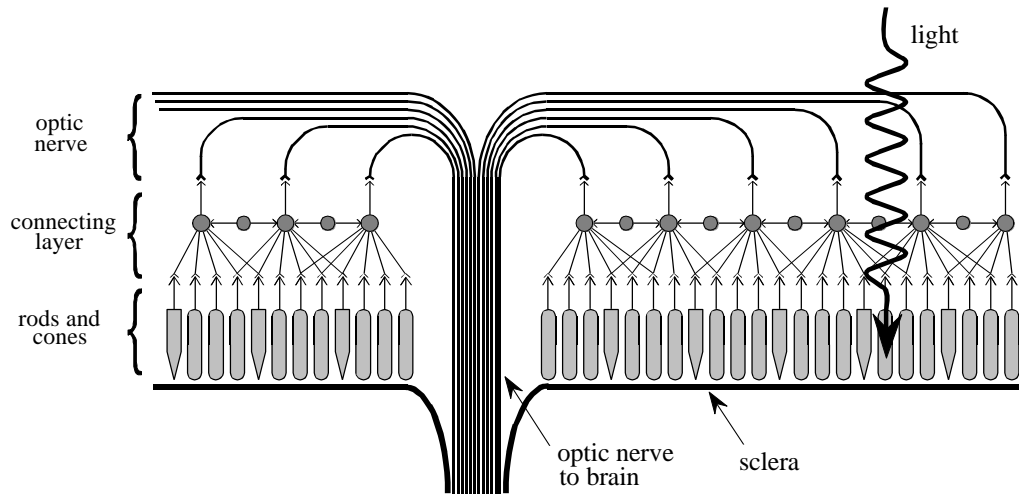


FIGURE 23-5

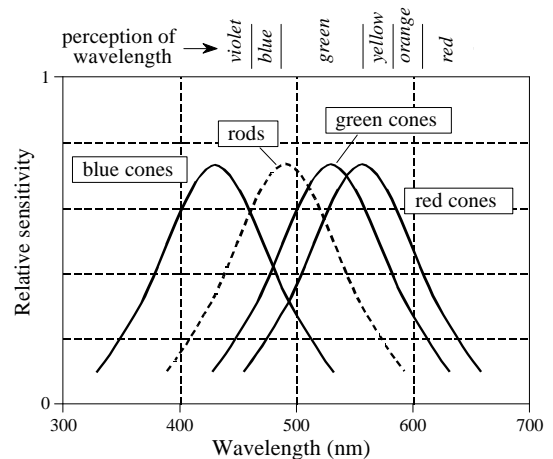
The human retina. The retina contains three principle layers: (1) the rod and cone light receptors, (2) an intermediate layer for data reduction and image processing, and (3) the optic nerve fibers that lead to the brain. The structure of these layers is seemingly *backward*, requiring light to pass through the other layers before reaching the light receptors.

Human vision overcomes the small size of the fovea by jerky eye movements called **saccades**. These abrupt motions allow the high resolution fovea to rapidly scan the field of vision for pertinent information. In addition, saccades present the rods and cones with a continually changing pattern of light. This is important because of the natural ability of the retina to adapt to changing levels of light intensity. In fact, if the eye is forced to remain fixed on the same scene, detail and color begin to fade in a few seconds.

The most common image sensor used in electronic cameras is the **charge coupled device (CCD)**. The CCD is an integrated circuit that replaced most vacuum tube cameras in the 1980s, just as transistors replaced vacuum tube amplifiers twenty years before. The heart of the CCD is a thin wafer of

FIGURE 23-6

Spectral response of the eye. The three types of cones in the human eye respond to different sections of the optical spectrum, roughly corresponding to red, green, and blue. Combinations of these three form all colors that humans can perceive. The cones do not have enough sensitivity to be used in low-light environments, where the rods are used to detect the image. This is why colors are difficult to perceive at night.



silicon, typically about 1 cm square. As shown by the cross-sectional view in Fig. 23-7, the backside is coated with a thin layer of metal connected to ground potential. The topside is covered with a thin electrical insulator, and a repetitive pattern of electrodes. The most common type of CCD is the **three phase readout**, where every third electrode is connected together. The silicon used is called *p-type*, meaning it has an excess of positive charge carriers called *holes*. For this discussion, a hole can be thought of as a positively charged particle that is free to move around in the silicon. Holes are represented in this figure by the "+" symbol.

In (a), +10 volts is applied to one of the three phases, while the other two are held at 0 volts. This causes the holes to move away from every third electrode, since positive charges are repelled by a positive voltage. This forms a region under these electrodes called a **well**, a shortened version of the physics term: *potential well*.

Each well in the CCD is a very efficient light sensor. As shown in (b), a single photon of light striking the silicon converts its energy into the formation of two charged particles, one electron, and one hole. The hole moves away, leaving the electron stuck in the well, held by the positive voltage on the electrode. Electrons in this illustration are represented by the "-" symbol. During the **integration period**, the pattern of light striking the CCD is transferred into a pattern of charge within the CCD wells. Dimmer light sources require longer integration periods. For example, the integration period for standard television is 1/60th of a second, while astrophotography can accumulate light for many hours.

Readout of the electronic image is quite clever; the accumulated electrons in each well are *pushed* to the output amplifier. As shown in (c), a positive voltage is placed on *two* of the phase lines. This results in each well expanding to the right. As shown in (d), the next step is to remove the voltage from the first phase, causing the original wells to collapse. This leaves the accumulated electrons in one well to the right of where they started. By repeating this pulsing sequence among the three phase lines, the accumulated electrons are pushed to the right until they reach a **charge sensitive amplifier**. This is a fancy name for a capacitor followed by a unity gain buffer. As the electrons are pushed from the last well, they flow onto the capacitor where they produce a voltage. To achieve high sensitivity, the capacitors are made extremely small, usually less than 1 pF. This capacitor and amplifier are an integral part of the CCD, and are made on the same piece of silicon. The signal leaving the CCD is a sequence of voltage levels proportional to the amount of light that has fallen on sequential wells.

Figure 23-8 shows how the two-dimensional image is read from the CCD. After the integration period, the charge accumulated in each well is moved up the column, one row at a time. For example, all the wells in row 15 are first moved into row 14, then row 13, then row 12, etc. Each time the rows are moved up, all the wells in row number 1 are transferred into the **horizontal register**. This is a group of specialized CCD wells that rapidly move the charge in a horizontal direction to the charge sensitive amplifier.

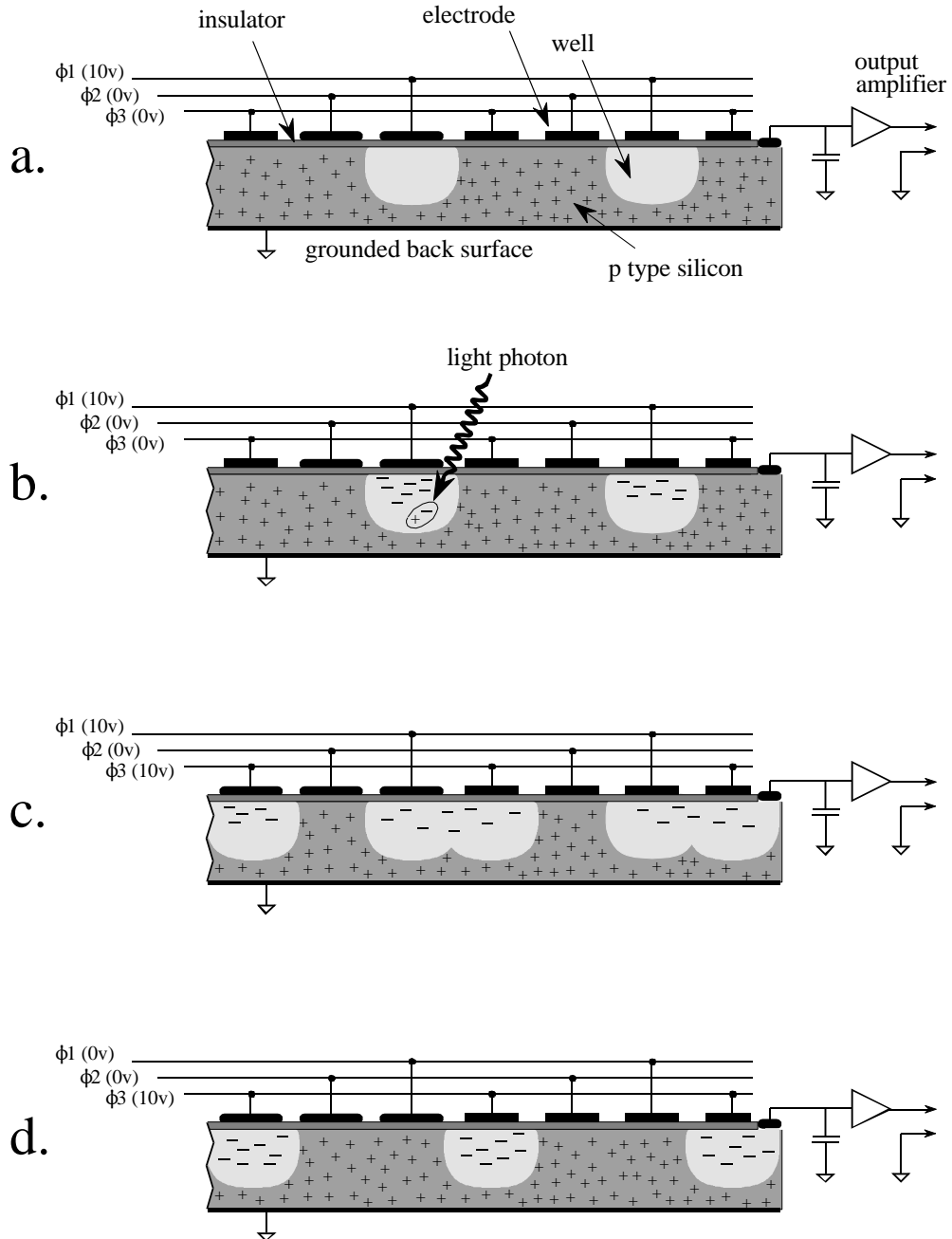


FIGURE 23-7

Operation of the charge coupled device (CCD). As shown in this cross-sectional view, a thin sheet of p-type silicon is covered with an insulating layer and an array of electrodes. The electrodes are connected in groups of three, allowing three separate voltages to be applied: ϕ_1 , ϕ_2 , and ϕ_3 . When a positive voltage is applied to an electrode, the holes (i.e., the positive charge carriers indicated by the "+") are pushed away. This results in an area depleted of holes, called a *well*. Incoming light generates holes and electrons, resulting in an accumulation of electrons confined to each well (indicated by the "-"). By manipulating the three electrode voltages, the electrons in each well can be moved to the edge of the silicon where a charge sensitive amplifier converts the charge into a voltage.

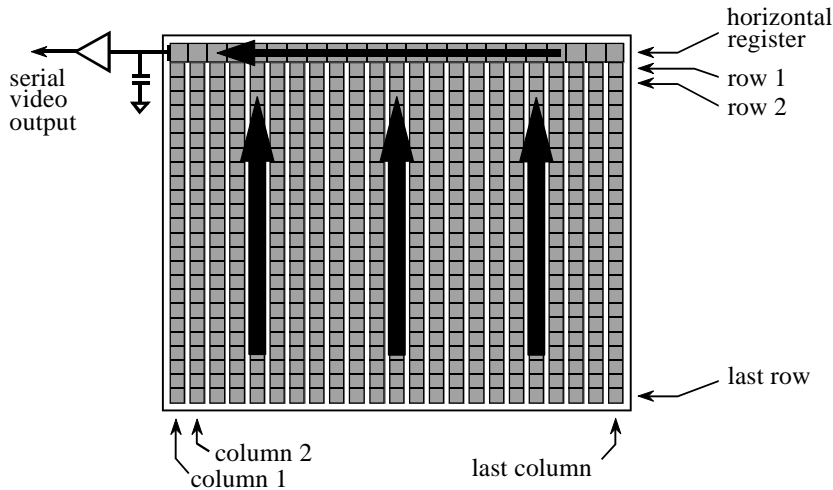


FIGURE 23-8

Architecture of the CCD. The imaging wells of the CCD are arranged in columns. During readout, the charge from each well is moved up the column into a horizontal register. The horizontal register is then readout into the charge sensitive preamplifier.

Notice that this architecture converts a two-dimensional array into a serial data stream in a particular sequence. The first pixel to be read is at the top-left corner of the image. The readout then proceeds from left-to-right on the first line, and then continues from left-to-right on subsequent lines. This is called **row major order**, and is almost always followed when a two-dimensional array (image) is converted to sequential data.

Television Video Signals

Although over 50 years old, the standard television signal is still one of the most common way to transmit an image. Figure 23-9 shows how the television signal appears on an oscilloscope. This is called **composite video**, meaning that there are vertical and horizontal synchronization (sync) pulses mixed with the actual picture information. These pulses are used in the television receiver to synchronize the vertical and horizontal deflection circuits to match the video being displayed. Each second of standard video contains 30 complete images, commonly called **frames**. A video engineer would say that each frame contains 525 **lines**, the television jargon for what programmers call *rows*. This number is a little deceptive because only 480 to 486 of these lines contain video information; the remaining 39 to 45 lines are reserved for sync pulses to keep the television's circuits synchronized with the video signal.

Standard television uses an **interlaced** format to reduce *flicker* in the displayed image. This means that all the odd lines of each frame are transmitted first, followed by the even lines. The group of odd lines is called the **odd field**, and the group of even lines is called the **even field**. Since

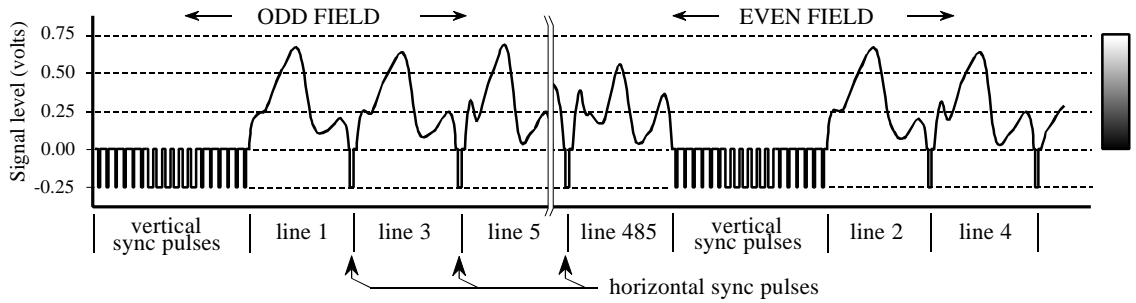


FIGURE 23-9

Composite video. The NTSC video signal consists of 30 complete frames (images) per second, with each frame containing 480 to 486 lines of video. Each frame is broken into two fields, one containing the odd lines and the other containing the even lines. Each field starts with a group of vertical sync pulses, followed by successive lines of video information separated by horizontal sync pulses. (The horizontal axis of this figure is not drawn to scale).

each frame consists of two fields, the video signal transmits 60 fields per second. Each field starts with a complex series of vertical sync pulses lasting 1.3 milliseconds. This is followed by either the even or odd lines of video. Each line lasts for 63.5 microseconds, including a 10.2 microsecond horizontal sync pulse, separating one line from the next. Within each line, the analog voltage corresponds to the grayscale of the image, with brighter values being in the direction *away* from the sync pulses. This places the sync pulses beyond the black range. In video jargon, the sync pulses are said to be *blacker than black*.

The hardware used for analog-to-digital conversion of video signals is called a **frame grabber**. This is usually in the form of an electronics card that plugs into a computer, and connects to a camera through a coaxial cable. Upon command from software, the frame grabber waits for the beginning of the next frame, as indicated by the vertical sync pulses. During the following two fields, each line of video is sampled many times, typically 512, 640 or 720 samples per line, at 8 bits per sample. These samples are stored in memory as one row of the digital image.

This way of acquiring a digital image results in an important difference between the vertical and horizontal directions. Each row in the digital image corresponds to one line in the video signal, and therefore to one row of wells in the CCD. Unfortunately, the columns are not so straightforward. In the CCD, each row contains between about 400 and 800 wells (columns), depending on the particular device used. When a row of wells is read from the CCD, the resulting line of video is filtered into a smooth analog signal, such as in Fig. 23-9. In other words, the video signal does not depend on how many columns are present in the CCD. The resolution in the horizontal direction is limited by how rapidly the analog signal is allowed to change. This is usually set at 3.2 MHz for color television, resulting in a risetime of about 100 nanoseconds, i.e., about 1/500th of the 53.2 microsecond video line.

When the video signal is digitized in the frame grabber, it is converted back into columns. However, these columns in the digitized image have *no relation* to the columns in the CCD. The number of columns in the digital image depends solely on how many times the frame grabber samples each line of video. For example, a CCD might have 800 wells per row, while the digitized image might only have 512 pixels (i.e., columns) per row.

The number of columns in the digitized image is also important for another reason. The standard television image has an **aspect ratio** of 4 to 3, i.e., it is slightly wider than it is high. Motion pictures have the wider aspect ratio of 25 to 9. CCDs used for scientific applications often have an aspect ratio of 1 to 1, i.e., a perfect square. In any event, the aspect ratio of a CCD is fixed by the placement of the electrodes, and cannot be altered. However, the aspect ratio of the digitized image depends on the number of samples per line. This becomes a problem when the image is displayed, either on a video monitor or in a hardcopy. If the aspect ratio isn't properly reproduced, the image looks squashed horizontally or vertically.

The 525 line video signal described here is called **NTSC** (National Television Systems Committee), a standard defined way back in 1954. This is the system used in the United States and Japan. In Europe there are two similar standards called **PAL** (Phase Alternation by Line) and **SECAM** (Sequential Chrominance And Memory). The basic concepts are the same, just the numbers are different. Both PAL and SECAM operate with 25 interlaced frames per second, with 625 lines per frame. Just as with NTSC, some of these lines occur during the vertical sync, resulting in about 576 lines that carry picture information. Other more subtle differences relate to how color and sound are added to the signal.

The most straightforward way of transmitting color television would be to have three separate analog signals, one for each of the three colors the human eye can detect: red, green and blue. Unfortunately, the historical development of television did not allow such a simple scheme. The color television signal was developed to allow existing black and white television sets to remain in use without modification. This was done by retaining the same signal for brightness information, but adding a separate signal for color information. In video jargon, the brightness is called the *luminance signal*, while the color is the *chrominance signal*. The chrominance signal is contained on a 3.58 MHz carrier wave added to the black and white video signal. Sound is added in this same way, on a 4.5 MHz carrier wave. The television receiver separates these three signals, processes them individually, and recombines them in the final display.

Other Image Acquisition and Display

Not all images are acquired an entire frame at a time. Another very common way is by **line scanning**. This involves using a detector containing a one-dimensional array of pixels, say, 2048 pixels long by 1 pixel wide. As an object is moved past the detector, the image is acquired line-by-line. Line

scanning is used by fax machines and airport x-ray baggage scanners. As a variation, the object can be kept stationary while the detector is moved. This is very convenient when the detector is already mounted on a moving object, such as an aircraft taking images of the ground beneath it. The advantage of line scanning is that *speed* is traded for detector *simplicity*. For example, a fax machine may take several seconds to scan an entire page of text, but the resulting image contains thousands of rows and columns.

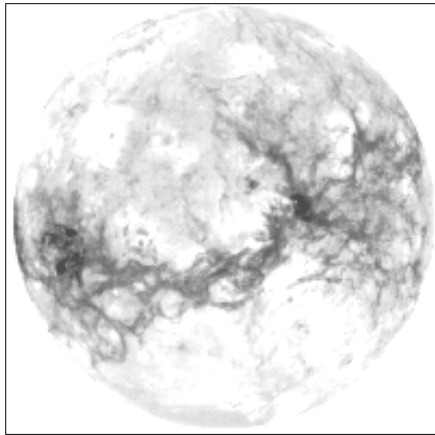
An even more simplified approach is to acquire the image **point-by-point**. For example, the microwave image of Venus was acquired one pixel at a time. Another example is the *scanning probe microscope*, capable of imaging individual atoms. A small probe, often consisting of only a single atom at its tip, is brought exceedingly close to the sample being imaged. Quantum mechanical effects can be detected between the probe and the sample, allowing the probe to be stopped an exact distance from the sample's surface. The probe is then moved over the surface of the sample, keeping a constant distance, tracing out the peaks and valleys. In the final image, each pixel's value represents the elevation of the corresponding location on the sample's surface.

Printed images are divided into two categories: **grayscale** and **halftone**. Each pixel in a grayscale image is a shade of gray between black and white, such as in a photograph. In comparison, each pixel in a halftone image is formed from many individual *dots*, with each dot being completely black or completely white. Shades of gray are produced by alternating various numbers of these black and white dots. For example, imagine a laser printer with a resolution of 600 dots-per-inch. To reproduce 256 levels of brightness between black and white, each pixel would correspond to an array of 16 by 16 printable dots. Black pixels are formed by making all of these 256 dots black. Likewise, white pixels are formed making all of these 256 dots white. Mid-gray has one-half of the dots white and one-half black. Since the individual dots are too small to be seen when viewed at a normal distance, the eye is fooled into thinking a grayscale has been formed.

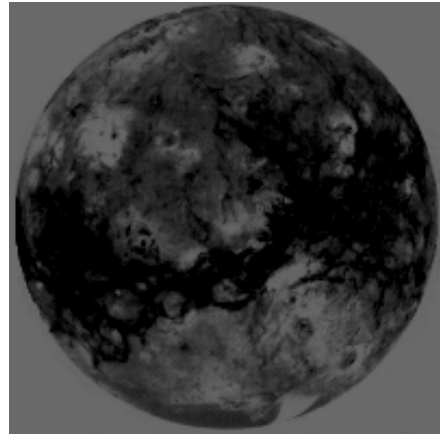
Halftone images are easier for printers to handle, including photocopy machines. The disadvantage is that the image quality is often worse than grayscale pictures.

Brightness and Contrast Adjustments

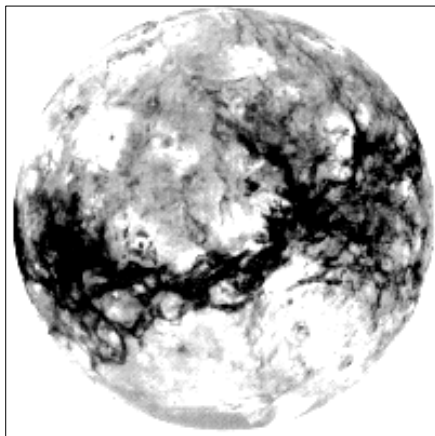
An image must have the proper **brightness** and **contrast** for easy viewing. Brightness refers to the overall lightness or darkness of the image. Contrast is the *difference* in brightness between objects or regions. For example, a white rabbit running across a snowy field has *poor* contrast, while a black dog against the same white background has *good* contrast. Figure 23-10 shows four possible ways that brightness and contrast can be misadjusted. When the brightness is too high, as in (a), the whitest pixels are saturated, destroying the detail in these areas. The reverse is shown in (b), where the brightness is set too low, saturating the blackest pixels. Figure (c) shows



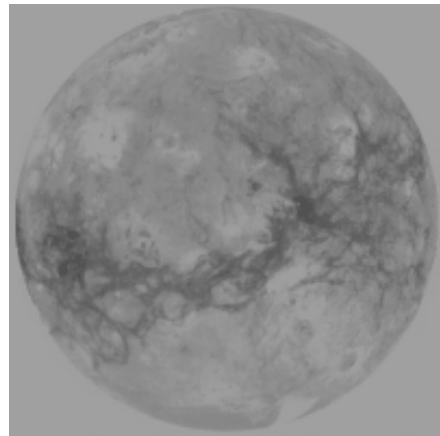
a. Brightness too high



b. Brightness too low



c. Contrast too high



d. Contrast too low

FIGURE 23-10

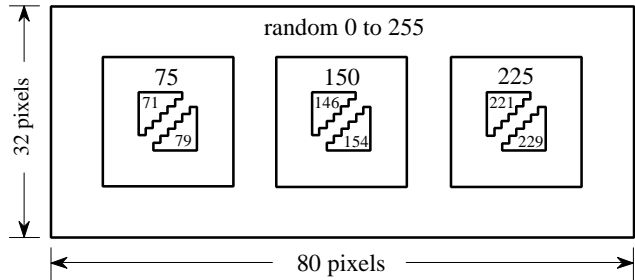
Brightness and contrast adjustments. Increasing the *brightness* makes every pixel in the image becomes lighter. In comparison, increasing the *contrast* makes the light areas become lighter, and the dark areas become darker. These images show the effect of misadjusting the brightness and contrast.

the contrast set to high, resulting in the blacks being too black, and the whites being too white. Lastly, (d) has the contrast set too low; all of the pixels are a mid-shade of gray making the objects fade into each other.

Figures 23-11 and 23-12 illustrate *brightness* and *contrast* in more detail. A test image is displayed in Fig. 23-12, using six different brightness and contrast levels. Figure 23-11 shows the construction of the test image, an array of 80×32 pixels, with each pixel having a value between 0 and 255. The background of the test image is filled with random noise, uniformly distributed between 0 and 255. The three square boxes have pixel values of 75, 150 and 225, from left-to-right. Each square contains two triangles with pixel values only slightly different from their surroundings. In other

FIGURE 23-11

Brightness and contrast test image. This is the structure of the digital image used in Fig. 23-12. The three squares form dark, medium, and bright objects, each containing two low contrast triangles. This figure indicates the digital numbers (DN) of the pixels in each region.



words, there is a dark region in the image with faint detail, this is a medium region in the image with faint detail, and there is a bright region in the image with faint detail.

Figure 23-12 shows how adjustment of the contrast and brightness allows different features in the image to be visualized. In (a), the brightness and contrast are set at the *normal* level, as indicated by the **B and C slide bars** at the left side of the image. Now turn your attention to the graph shown with each image, called an **output transform**, an **output look-up table**, or a **gamma curve**. This controls the hardware that displays the image. The value of each pixel in the stored image, a number between 0 and 255, is passed through this look-up table to produce another number between 0 and 255. This new digital number drives the video intensity circuit, with 0 through 255 being transformed into black through white, respectively. That is, the look-up table maps the stored numbers into the displayed brightness.

Figure (a) shows how the image appears when the output transform is set to do *nothing*, i.e., the digital output is identical to the digital input. Each pixel in the noisy background is a random shade of gray, equally distributed between black and white. The three boxes are displayed as dark, medium and light, clearly distinct from each other. The problem is, the triangles inside each square cannot be easily seen; the contrast is too low for the eye to distinguish these regions from their surroundings.

Figures (b) & (c) show the effect of changing the brightness. Increasing the brightness shifts the output transform to the *left*, while decreasing the brightness shifts it to the *right*. Increasing the brightness makes *every* pixel in the image appear lighter. Conversely, decreasing the brightness makes *every* pixel in the image appear darker. These changes can improve the viewability of excessively dark or light areas in the image, but will **saturate** the image if taken too far. For example, all of the pixels in the far right square in (b) are displayed with full intensity, i.e., 255. The opposite effect is shown in (c), where all of the pixels in the far left square are displayed as blackest black, or digital number 0. Since all the pixels in these regions have the same value, the triangles are completely wiped out.

Also notice that *none* of the triangles in (b) and (c) are easier to see than in (a). Changing the brightness provides little (if any) help in distinguishing low contrast objects from their surroundings.

Figure (d) shows the display optimized to view pixel values around digital number 75. This is done by turning up the *contrast*, resulting in the output transform increasing in *slope*. For example, the stored pixel values of 71 and 75 become 100 and 116 in the display, making the contrast a factor of four greater. Pixel values between 46 and 109 are displayed as the blackest black, to the whitest white. The price for this increased contrast is that pixel values 0 to 45 are saturated at black, and pixel values 110 to 255 are saturated at white. As shown in (d), the increased contrast allows the triangles in the left square to be seen, at the cost of saturating the middle and right squares.

Figure (e) shows the effect of increasing the contrast even further, resulting in only 16 of the possible 256 stored levels being displayed as nonsaturated. The brightness has also been decreased so that the 16 usable levels are centered on digital number 175. The details in the center square are now very visible; however, almost everything else in the image is saturated. For example, look at the noise around the border of the image. There are very few pixels with an intermediate gray shade; almost every pixel is either pure black or pure white. This technique of using high contrast to view only a few levels is sometimes called a **grayscale stretch**.

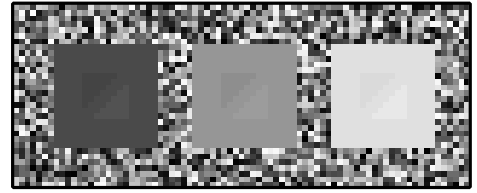
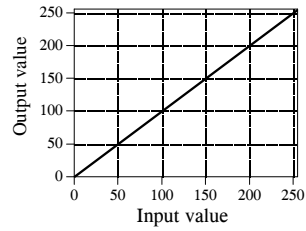
The contrast adjustment is a way of *zooming in* on a smaller range of pixel values. The brightness control *centers* the zoomed section on the pixel values of interest. Most digital imaging systems allow the brightness and contrast to be adjusted in just this manner, and often provide a graphical display of the output transform (as in Fig. 23-12). In comparison, the brightness and contrast controls on television and video monitors are *analog circuits*, and may operate differently. For example, the contrast control of a monitor may adjust the gain of the analog signal, while the brightness might add or subtract a DC offset. The moral is, don't be surprised if these analog controls don't respond in the way you think they should.

Grayscale Transforms

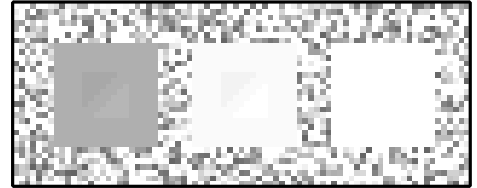
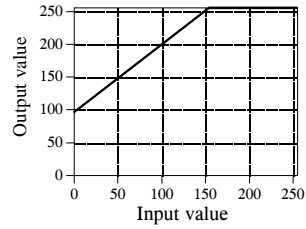
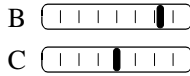
The last image, Fig. 23-12f, is different from the rest. Rather than having a slope in the curve over *one* range of input values, it has a slope in the curve over *two* ranges. This allows the display to simultaneously show the triangles in both the left and the right squares. Of course, this results in saturation of the pixel values that are *not* near these digital numbers. Notice that the slide bars for contrast and brightness are not shown in (f); this display is beyond what brightness and contrast adjustments can provide.

Taking this approach further results in a powerful technique for improving the appearance of images: the **grayscale transform**. The idea is to increase the contrast at pixel values of interest, at the expense of the pixel values we don't care about. This is done by defining the relative importance of each of the 0 to 255 possible pixel values. The more important the value, the greater its contrast is made in the displayed image. An example will show a systematic way of implementing this procedure.

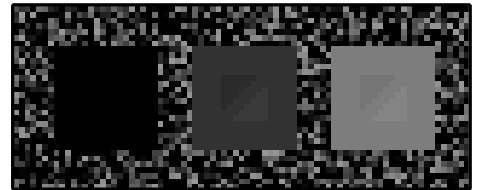
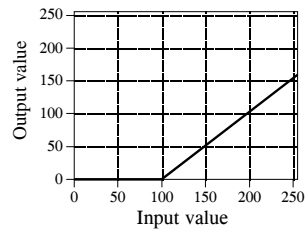
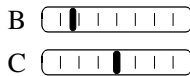
a. Normal



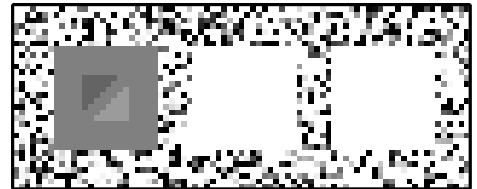
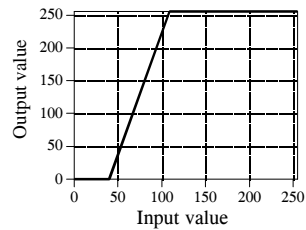
b. Increased brightness



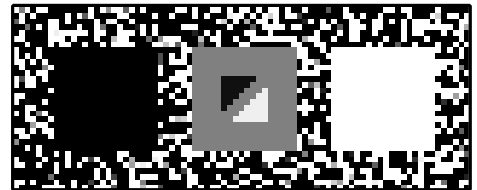
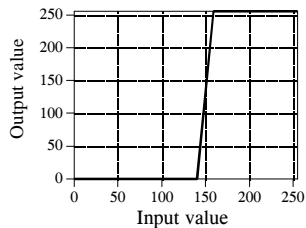
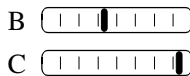
c. Decreased brightness



d. Slightly increased contrast at DN 75



e. Greatly increased contrast at DN 150



f. Increased contrast at both DN 75 and 225

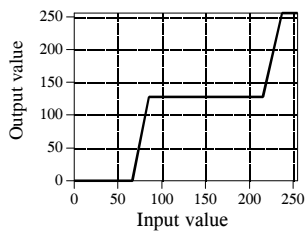
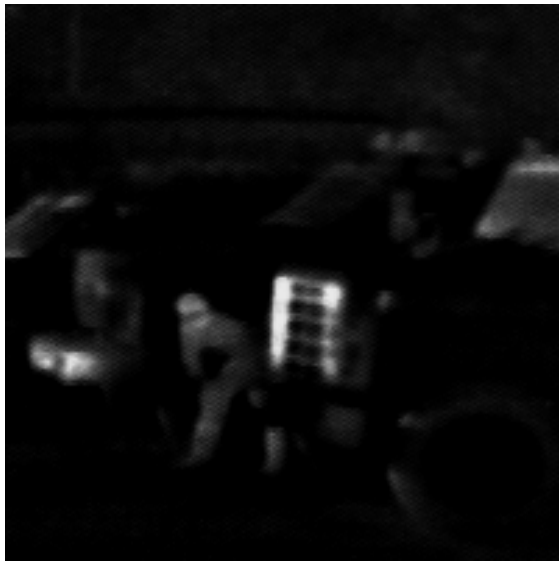


FIGURE 23-12



a. Original IR image



b. With grayscale transform

FIGURE 23-13

Grayscale processing. Image (a) was acquired with an infrared camera in total darkness. Brightness in the image is related to the temperature, accounting for the appearance of the warm human body and the hot truck grill. Image (b) was processed with the manual grayscale transform shown in Fig. 23-14c.

The image in Fig. 23-13a was acquired in total darkness by using a CCD camera that is sensitive in the far infrared. The parameter being imaged is *temperature*: the hotter the object, the more infrared energy it emits and the brighter it appears in the image. This accounts for the background being very black (cold), the body being gray (warm), and the truck grill being white (hot). These systems are great for the military and police; you can see the other guy when he can't even see himself! The image in (a) is difficult to view because of the uneven distribution of pixel values. Most of the image is so dark that details cannot be seen in the scene. On the other end, the grill is near white saturation.

The histogram of this image is displayed in Fig. 23-14a, showing that the background, human, and grill have reasonably separate values. In this example, we will increase the contrast in the background and the grill, at the expense of everything else, including the human body. Figure (b) represents this strategy. We declare that the lowest pixel values, the background, will have a relative contrast of twelve. Likewise, the highest pixel values, the grill, will have a relative contrast of six. The body will have a relative contrast of one, with a staircase transition between the regions. All these values are determined by trial and error.

The grayscale transform resulting from this strategy is shown in (c), labeled *manual*. It is found by taking the running sum (i.e., the discrete integral) of the curve in (b), and then normalizing so that it has a value of 255 at the

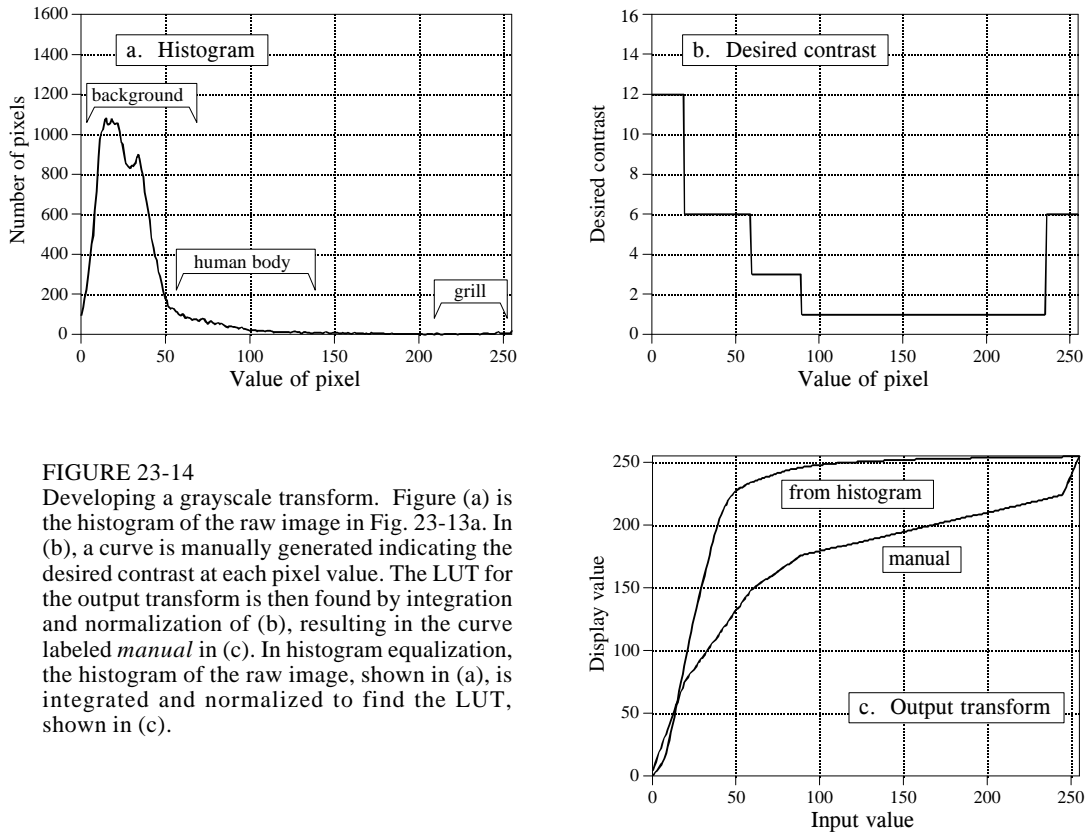


FIGURE 23-14

Developing a grayscale transform. Figure (a) is the histogram of the raw image in Fig. 23-13a. In (b), a curve is manually generated indicating the desired contrast at each pixel value. The LUT for the output transform is then found by integration and normalization of (b), resulting in the curve labeled *manual* in (c). In histogram equalization, the histogram of the raw image, shown in (a), is integrated and normalized to find the LUT, shown in (c).

right side. Why take the *integral* to find the required curve? Think of it this way: The contrast at a particular pixel value is equal to the slope of the output transform. That is, we want (b) to be the derivative (slope) of (c). This means that (c) must be the integral of (b).

Passing the image in Fig. 23-13a through this manually determined grayscale transform produces the image in (b). The background has been made *lighter*, the grill has been made *darker*, and both have better contrast. These improvements are at the expense of the body's contrast, producing a less detailed image of the intruder (although it can't get much worse than in the original image).

Grayscale transforms can significantly improve the viewability of an image. The problem is, they can require a great deal of trial and error. **Histogram equalization** is a way to automate the procedure. Notice that the histogram in (a) and the contrast weighting curve in (b) have the same general shape. Histogram equalization blindly uses the histogram as the contrast weighing curve, eliminating the need for human judgement. That is, the output transform is found by integration and normalization of the *histogram*, rather than a manually generated curve. This results in the greatest contrast being given to those values that have the greatest number of pixels.

Histogram equalization is an interesting mathematical procedure because it maximizes the *entropy* of the image, a measure of how much information is transmitted by a fixed number of bits. The fault with histogram equalization is that it mistakes the sheer *number* of pixels at a certain value with the *importance* of the pixels at that value. For example, the truck grill and human intruder are the most prominent features in Fig. 23-13. In spite of this, histogram equalization would almost completely ignore these objects because they contain relatively few pixels. Histogram equalization is quick and easy. Just remember, if it doesn't work well, a manually generated curve will probably do much better.

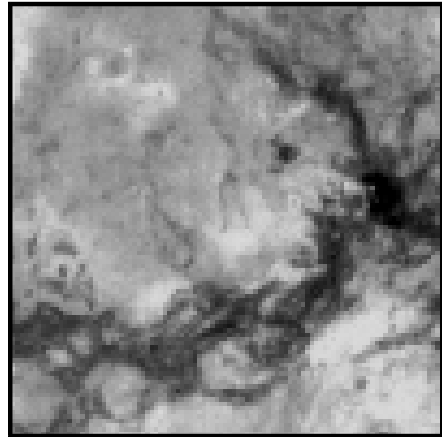
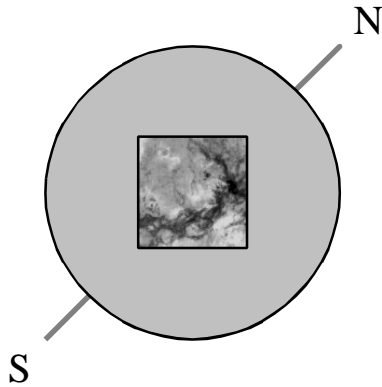
Warping

One of the problems in photographing a planet's surface is the distortion from the curvature of the spherical shape. For example, suppose you use a telescope to photograph a square region near the center of a planet, as illustrated in Fig. 23-15a. After a few hours, the planet will have rotated on its axis, appearing as in (b). The previously photographed region appears highly distorted because it is curved near the horizon of the planet. Each of the two images contain complete information about the region, just from a different perspective. It is quite common to acquire a photograph such as (a), but really want the image to look like (b), or vice versa. For example, a satellite mapping the surface of a planet may take thousands of images from straight above, as in (a). To make a natural looking picture of the entire planet, such as the image of Venus in Fig. 23-1, each image must be distorted and placed in the proper position. On the other hand, consider a weather satellite looking at a hurricane that is not directly below it. There is no choice but to acquire the image obliquely, as in (b). The image is then converted into how it would appear from above, as in (a).

These spatial transformations are called **warping**. Space photography is the most common use for warping, but there are others. For example, many vacuum tube imaging detectors have various amounts of spatial distortion. This includes night vision cameras used by the military and x-ray detectors used in the medical field. Digital warping (or *dewarping* if you prefer) can be used to correct the inherent distortion in these devices. Special effects artists for motion pictures love to warp images. For example, a technique called **morphing** gradually warps one object into another over a series of frames. This can produce illusions such as a child turning into an adult, or a man turning into a werewolf.

Warping takes the *original image* (a two-dimensional array) and generates a *warped image* (another two-dimensional array). This is done by looping through each pixel in the warped image and asking: What is the proper pixel value that should be placed here? Given the particular row and column being calculated in the warped image, there is a corresponding row and column in the original image. The pixel value from the original image is transferred to the warped image to carry out the algorithm. In the jargon of image processing, the row and column that the pixel *comes from* in the

a. Normal View



b. Oblique View

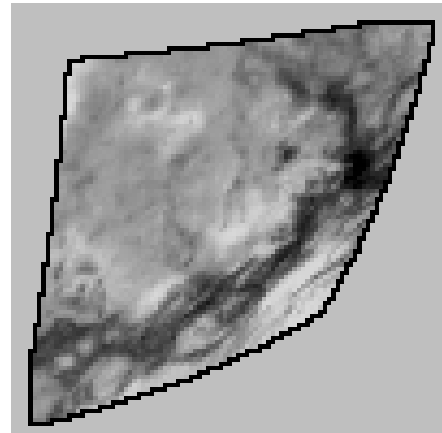
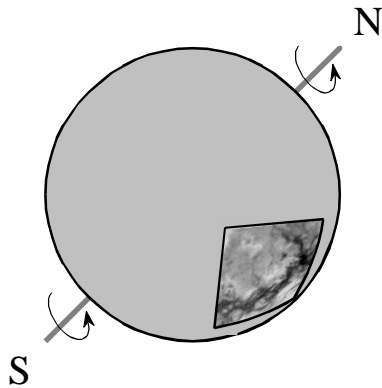


FIGURE 23-15

Image warping. As shown in (a), a normal view of a small section of a planet appears relatively distortion free. In comparison, an oblique view presents significant spatial distortion. *Warping* is the technique of changing one of these images into the other.

original image is called the **comes-from address**. Transferring each pixel from the original to the warped image is the easy part. The hard part is calculating the *comes-from address* associated with each pixel in the warped image. This is usually a pure math problem, and can become quite involved. Simply stretching the image in the horizontal or vertical direction is easier, involving only a multiplication of the row and/or column number to find the comes-from address.

One of the techniques used in warping is **subpixel interpolation**. For example, suppose you have developed a set of equations that turns a row and column address in the warped image into the comes-from address in the original

image. Consider what might happen when you try to find the value of the pixel at row 10 and column 20 in the warped image. You pass the information: *row = 10, column = 20*, into your equations, and out pops: *comes-from row = 20.2, comes-from column = 14.5*. The point being, your calculations will likely use floating point, and therefore the comes-from addresses will not be integers. The easiest method to use is the **nearest neighbor** algorithm, that is, simply round the addresses to the nearest integer. This is simple, but can produce a very grainy appearance at the edges of objects where pixels may appear to be slightly misplaced.

Bilinear interpolation requires a little more effort, but provides significantly better images. Figure 23-16 shows how it works. You know the value of the four pixels *around* the fractional address, i.e., the value of the pixels at row 20 & 21, and column 14 and 15. In this example we will assume the pixels values are 91, 210, 162 and 95. The problem is to interpolate between these four values. This is done in two steps. First, interpolate in the *horizontal* direction between column 14 and 15. This produces two intermediate values, 150.5 on line 20, and 128.5 on line 21. Second, interpolate between these intermediate values in the vertical direction. This produces the bilinear interpolated pixel value of 139.5, which is then transferred to the warped image. Why interpolate in the horizontal direction *and then* the vertical direction instead of the reverse? It doesn't matter; the final answer is the same regardless of which order is used.

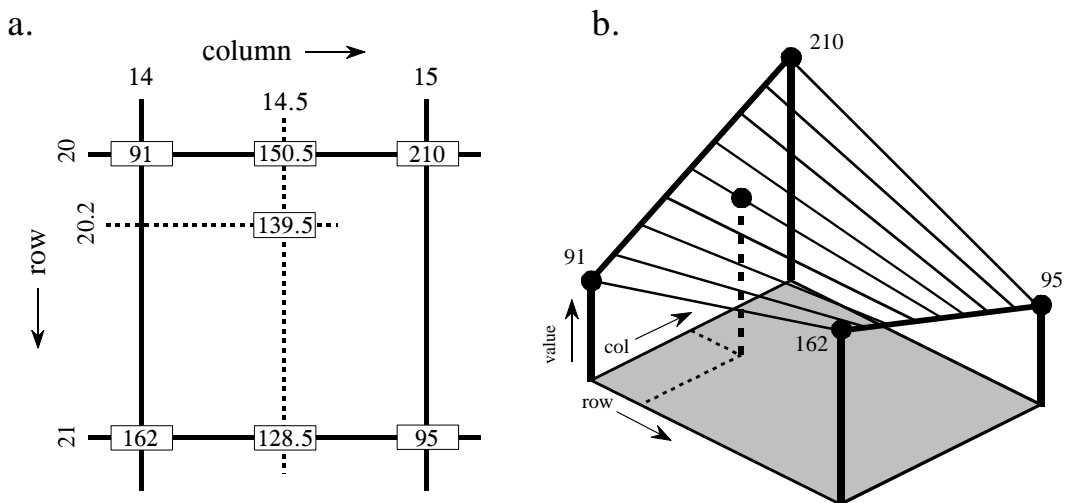


FIGURE 23-16

Subpixel interpolation. Subpixel interpolation for image warping is usually accomplished with bilinear interpolation. As shown in (a), two intermediate values are calculated by linear interpolation in the horizontal direction. The final value is then found by using linear interpolation in the vertical direction between the intermediate values. As shown by the three-dimensional illustration in (b), this procedure uniquely defines all values between the four known pixels at each of the corners.