CHAPTER 2

Obtaining Images

Your first encounter with an image as something to compute with (rather than look at) is likely as an array for an intensity image, or set of three arrays for a color image. Knowing how the image ended up in this form is important if you want to interpret it. A quite detailed model of the geometry and physics underlying images appears in Part X. A simple model will have to do for the moment.

2.1 CAMERAS

The image you see as three arrays starts as a spectral energy field – Power P moving through space. This power is a function of position in 3D $\mathbf{X}$, direction ω, time t, and wavelength λ, so you can write $P(\mathbf{X}, \omega, t, \lambda)$. This power is created by light leaving light sources, reflecting from surfaces, and eventually arriving at the entrance to the camera (Figure 2.1). This is usually but not always a lens.

2.1.1 The Pinhole Camera

A *pinhole camera* is a light-tight box with a very small hole in the front. Think about a point on the back of the box. The only light that arrives at that point must come through the hole, because the box is light-tight. If the hole is very small, then the light that arrives at the point comes from only one direction. This means that an inverted image of a scene appears at the back of the box (Figure 2.3). An appropriate sensor (CMOS sensor; CCD sensor; light sensitive film) at the back of the box will capture this image.

Pinhole camera models produce an upside-down image. This is easily dealt with in practice (turn the image the right way up). An easy way to account for this is to assume the sensor is *in front* of the hole, so that the image is not upside-

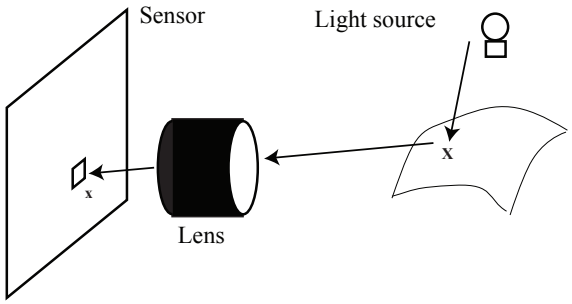

FIGURE 2.1: *A high-level model of imaging. Light leaves light sources and reflects from surfaces. Eventually, some light arrives at a camera and enters a lens system. Some of that light arrives at a photosensor inside the camera.*

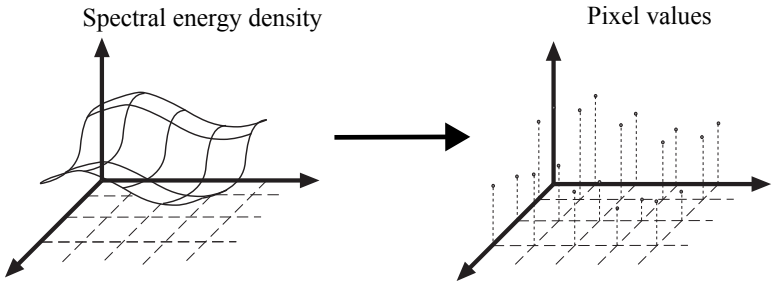

FIGURE 2.2: *Because each pixel in the sensor averages over a small range of directions and positions, the process mapping the input spectral energy distribution to pixel values can be thought of as sampling. On the* **left***, is a representation of the energy distribution as a continuous function of position. The value reported at each pixel is the value of this function at the location of the pixel (***right***).*

down. One could not build a camera like this (the sensor blocks light from the hole) but it is a convenient abstraction. There is a standard model of this camera, in a standard coordinate system (Figure 2.4). Notice that the y axis goes *down* in the image. While this is usual for image coordinate systems, there are further reasons to do this. Most people's intuition is that z *increases* as one moves into the image, and orienting the y axis downward in the image allows me to achieve this, have x in the usual direction, and use a right-handed coordinate system. The pinhole – usually called the *focal point* – is at the origin, and the sensor is on the plane $z = f$. This plane is the *image plane*, and f is the *focal length*. We ignore any camera body and regard the image plane as infinite.

Under this highly abstracted camera model, almost any point in 3D will map to a point in the image plane. We *image* a point in 3D by constructing a ray through the 3D point and the focal point, and intersecting that ray with the image plane. The focal point has an important, distinctive, property: It cannot be imaged, and it is the only point that cannot be imaged.

Similar triangles yields that the camera maps a point $\mathbf{X}$ in 3D to a point $\mathbf{x}$ on the image plane by:

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} fX/Z \\ fY/Z \\ f \end{pmatrix} = \mathbf{x}.$$

Notice that the z-coordinate is the same for each point on the image plane, so it is quite usual to ignore it and use the model

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix} = \mathbf{x}.$$

The focal length just scales the image. In standard camera models, other scaling

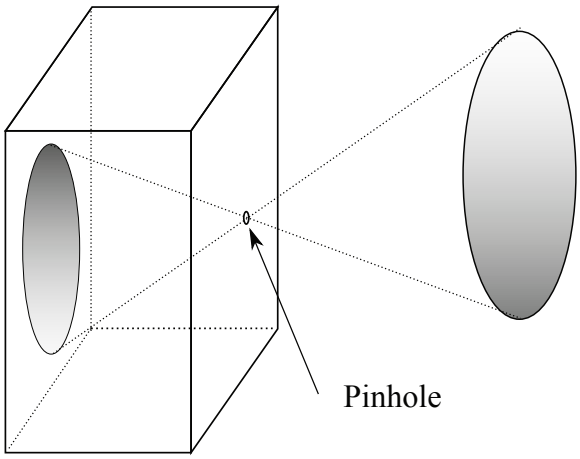

FIGURE 2.3: *In the pinhole imaging model, a light-tight box with a pinhole in it views an object. The only light that a point on the back of the box sees comes through the very small pinhole, so that an inverted image is formed on the back face of the box.*

effects occur as well, and we write projection as if $f = 1$, yielding

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix} = \mathbf{x}.$$

The projection process is known as *perspective projection*. The point where the z-axis intersects the image plane (equivalently, where the ray through the focal point perpendicular to the image plane intersects the image plane) is the *camera center*.

Remember this: *Most practical cameras can be modelled as a pinhole camera. The standard model of the pinhole camera maps*

$$(X, Y, Z) \rightarrow (X/Z, Y/Z).$$

Figure 27.1 shows important terminology (focal point; image plane; camera center).

2.1.2 Images as Sampled Functions

Various processes in lens and camera map some of the light that arrives to some *sensor* at the back of a camera, usually in a way that is very largely consistent with the pinhole camera model. The sensor is made up of a grid of *receptors*, each of which transduces the energy that arrives into a number (or some numbers). Each

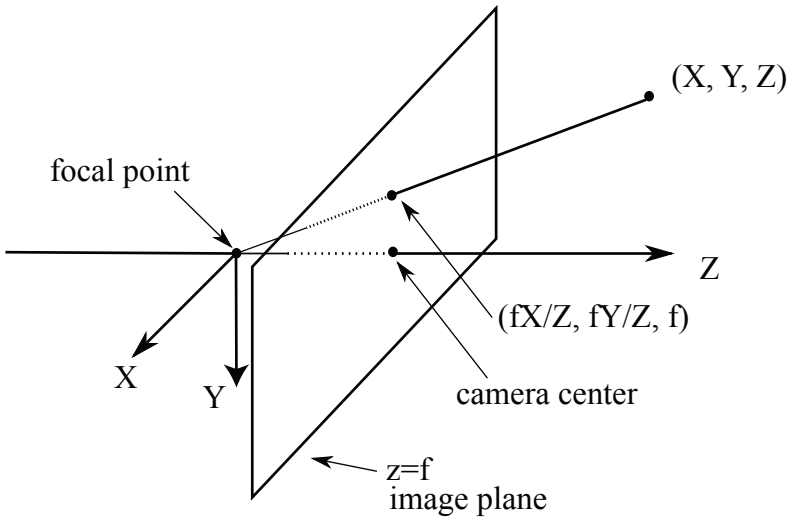

FIGURE 2.4: *The usual geometric abstraction of the pinhole model. The box doesn't affect the geometry, and is omitted. The pinhole has been moved to the back of the box, so that the image is no longer inverted. The image is formed on the plane $z = f$, by convention. Notice the y-axis goes down in the image. This allows me to use a* right handed *coordinate system and also have z* increase *as one moves into the image.*

receptor on the sensor corresponds to a single *pixel* (or spatial location) in the array that is read from the camera.

The lens arranges that light arriving at $\mathbf{x}$ on the sensor all arrived from one point ($\mathbf{X}$ in Figure 2.1) on a surface in 3D, assuming it is in focus. At $\mathbf{x}$, the sensor collects power P for some period Δt, then passes the result on to the camera electronics. The sensor responds to energy $P\Delta t$, so collecting more power for a shorter period or less power for a longer period will result in indistinguishable results. The value of the pixel at i, j on the grid is a *sample* of a function of position (Figure 2.2).

The vast majority of sensors in current use are linear, so doubling the amount of light arriving at the camera while fixing Δt will double the output. Linear image sensors present problems. The *dynamic range* (ratio of largest value to smallest value) of spectral energy fields can be startlingly large (1e6: 1 is often cited). Simple consumer cameras report 8 bits (256 levels) of intensity per channel. A picture from a linear camera that reports 8 bits per channel will look strange, because even relatively simple scenes have a higher dynamic range than 255. One can build cameras that can report significantly higher dynamic ranges, but this takes work (Section 27.3.4).

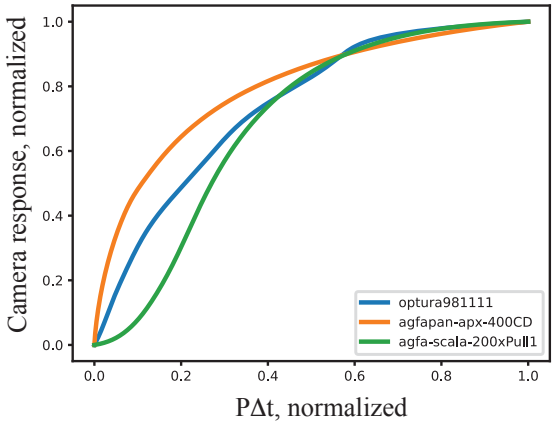

FIGURE 2.5: *Camera response functions for three different cameras, plotted from the comprehensive dataset available at* `https://cave.cs.columbia.edu/repository/DoRF`. *The horizontal axis is the "input" – the $P\Delta t$ observed by the camera, scaled to $0-1$. The vertical axis is the "output" – the response of the camera, again scaled to $0-1$. Notice that locations that would be quite dark for a linear sensor will be lighter; but as the linear sensor gets very bright, the output recorded by the camera grows slowly. This means that the range of outputs is smaller than the range of inputs, which is helpful for practical cameras. This response function is typically located deep in the camera's electronics. Typical consumer cameras apply a variety of transforms before reporting an image, though one can often persuade cameras to produce an untransformed, linear response image (a RAW file).*

2.1.3 Camera Response Functions

If the camera has a linear response and a dynamic range of 255, either a lot of the image will be too dark to be resolved, or much of the image will be at the highest value, or both will happen. This is usually fixed by ensuring that the number digitized by the camera *isn't* linearly related to brightness. Internal electronics ensures that the *camera response function* mapping the intensity arriving at the sensor to the reported pixel value looks something like Figure 2.5. This increases the response to dark values, and reduces it to light values, so that the overall distribution of pixel values is familiar. Typically, the function used approximates the response of film (which isn't linear) because people are familiar with that.

2.1.4 Color Images

Humans see color by comparing the response of different kinds of photoreceptor at nearby locations (Chapter 29). The main difference between these kinds of photoreceptor is in the sensitivity of the sensor with wavelength. Roughly, one type of sensor responds more strongly to longer wavelengths, another to medium wavelengths, and a third to shorter wavelengths (there are other kinds of sensor,

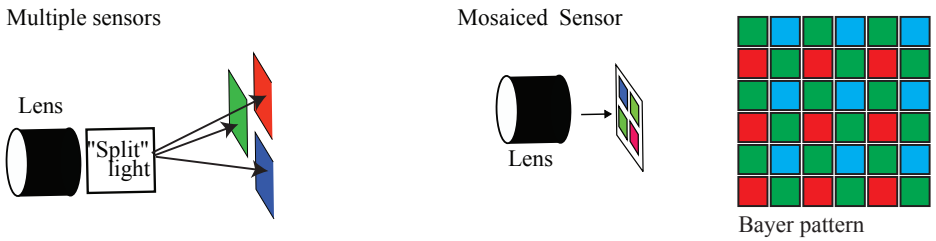

FIGURE 2.6: *There are two main ways to obtain color images. One can (as in* **left***) build a multiccd camera with three imaging sensors. Each has a different response to wavelengths. The cheaper and lighter alternative is to use one imaging sensor (*__right__*) but have a mosaic of pixels with different responses. This can be achieved by placing a small filter on each sensor location.* **Far right** *shows one traditional such pattern of filters, a* Bayer pattern.

and other differences).

Cameras parallel this process. The sensors used for the R (or red) layer of an RGB image respond more strongly to longer wavelengths; for the G (or green) layer, to medium wavelengths; and the B (or blue) to shorter wavelengths. Cameras must be engineered to produce the response of three different types of sensor *at the same place*. The usual strategy is to use one imaging sensor, and arrange that different pixels respond differently to wavelength. Typically, there are three types of pixel (R, G, and B), interleaved in a *mosaic* (Figure 2.6). This means that at many locations the camera does not measure R (or G, or B) response, and it must reconstruct this response from the value at nearby pixels. Generally, mosaic patterns have more G pixels than R or B pixels. This is because G pixels are sensitive to a wider range of visible wavelengths than R and B pixels, and so the reconstruction yields better results. Regular mosaic patterns can create effects in images, and there are *demosaicing* algorithms to remove these effects. an alternative is to use three imaging sensors and arranging for each sensor to receive the same light (lenses, mirrors, that sort of thing). Such *multiccd cameras* tend to be larger, heavier and more expensive than single sensor cameras.

2.1.5 Pointwise Image Transformations

The camera response function of Section 2.1.3 is one example of a *pointwise image transformation*. Most such transformations occur *after* the image has been read out of the camera. You take the array of pixels and apply some function to each pixel value. Simple, but useful, examples include: forming a negative (map x to $1 - x$); contrast adjustment (choose a function that makes dark pixels darker and light pixels lighter); and gamma correction (using a function that corrects for a quirk of image encoding, Figure 2.7).

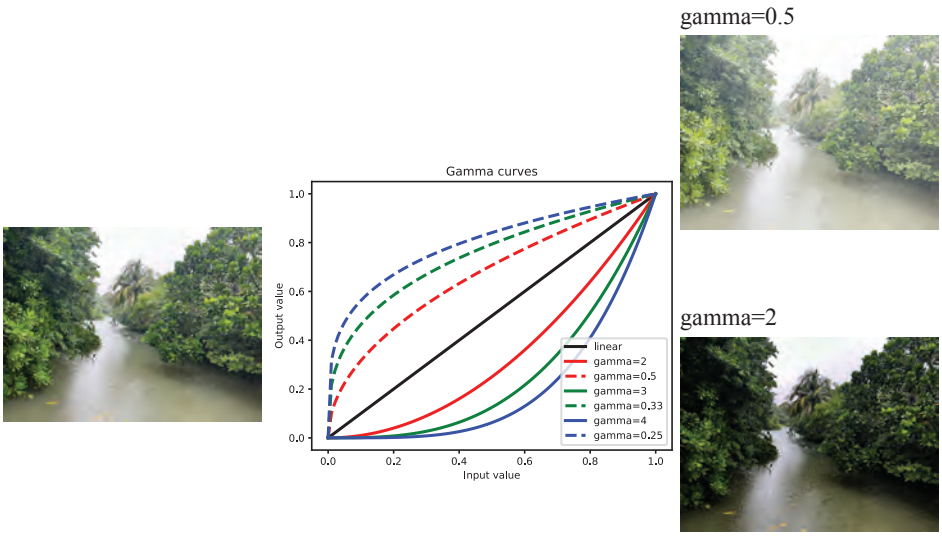

FIGURE 2.7: *Many imaging and rendering devices have a response that is a power of the input, so that output* $= C input^{\gamma}$*, where* γ *is a parameter of the device. One can simulate this effect by applying a transform like those shown in the* **center** *(curves for several values of* γ*). Note that you can remove the effect of such a transform* – gamma correct *the image – by applying another such transform with an appropriately chosen* γ*. The image on the* **left** *is transformed to the two examples on the* **right** *with different* γ *values.* Image credit: *Figure shows my photograph of a river in Singapore.*

> **Remember this:** *Cameras consist of lens systems (which arrange that light leaving a point on a surface arrives at a sensor), sensors (which sample the amount of arriving energy) and electronics (which map the sampled values into the numbers reported by the camera). Most cameras have linear sensors, but apply a camera response function to the sensor outputs. Color images can be obtained by arranging that three different sensors see the same light (heavy and expensive), or using a mosaic pattern of filters on a single sensor (cheap, but presenting reconstruction problems).*

2.2 SENSING DEPTH

It is often very useful to measure the 3D location of points directly. Methods include: stereopsis; camera projector stereo; structured light; and time of flight sensors.

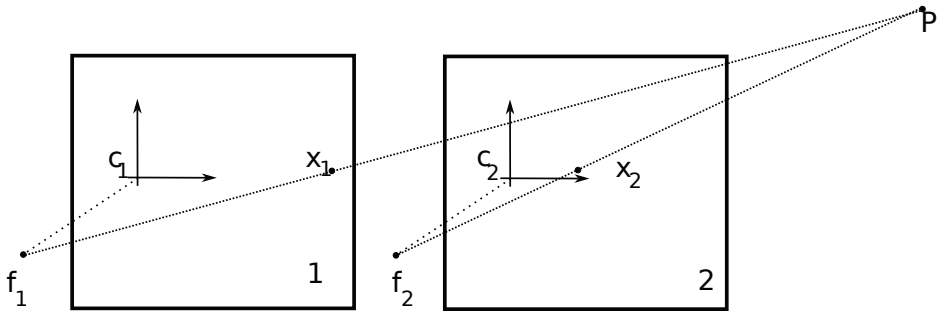

FIGURE 2.8: *When two pinhole cameras view a point, the 3D coordinates of the point can be reconstructed from the two images of that point. This applies for almost every configurations of the cameras. It is an elementary exercise in trigonometry (***exercises** *) to determine* $\mathbf{P}$ *from the positions of the two focal points, the locations of the point in the two images, and the distance between the focal points. Considerable work can be required to find appropriate matching points, but the procedures required are now extremely well understood (Chapters 34). One can now buy camera systems that use this approach to report 3D point locations (often known as RGBD cameras). Here we show a specialized camera geometry, chosen to simplify notation. The second camera is translated with respect to the first, along a direction parallel to the image plane. The second camera is a copy of the first camera, so the image planes are parallel. In this geometry, the point being viewed shifts somewhat to the left in the right camera.*

2.2.1 Sensing Depth with Stereo

Stereo uses two pinhole cameras somewhat offset from one another. Figure 2.8 sketches this idea. The key is that if you know where the cameras are with respect to one another, and where a 3D point projects to in each of two perspective images, simple trigonometry will reveal where it is in 3D. Calibrating the relative geometry of the cameras is now well understood (Chapter 32), as is determining which (if any) point in the first image corresponds to which in the second (Chapter 34), and recovering a good depth model from this information (Chapter 34). Stereo rigs can be very cheap and accurate, and they have the great advantage that measurement is passive – one does not have to send signals into the environment.

But there are limits to stereopsis. Measuring large depths with two cameras that are close together requires highly accurate estimates of point positions in images. Figure 2.8 shows a simple geometry that illustrates the problem. The point $\mathbf{P}$ projects to $\mathbf{x}_1$ in camera 1, and to $\mathbf{x}_2$ in camera 2. Notice because of the carefully chosen camera geometry, the y-coordinates of $\mathbf{x}_1$ and $\mathbf{x}_2$ are the same; only the x-coordinates differ. Write x_1 for the x-coordinate of $\mathbf{x}_1$; X for the x-coordinate of P, and so on. From the triangles in that figure, we have

$$d = x_2 - x_1 = f\frac{(X - B) - X}{Z} = -f\frac{B}{Z}$$

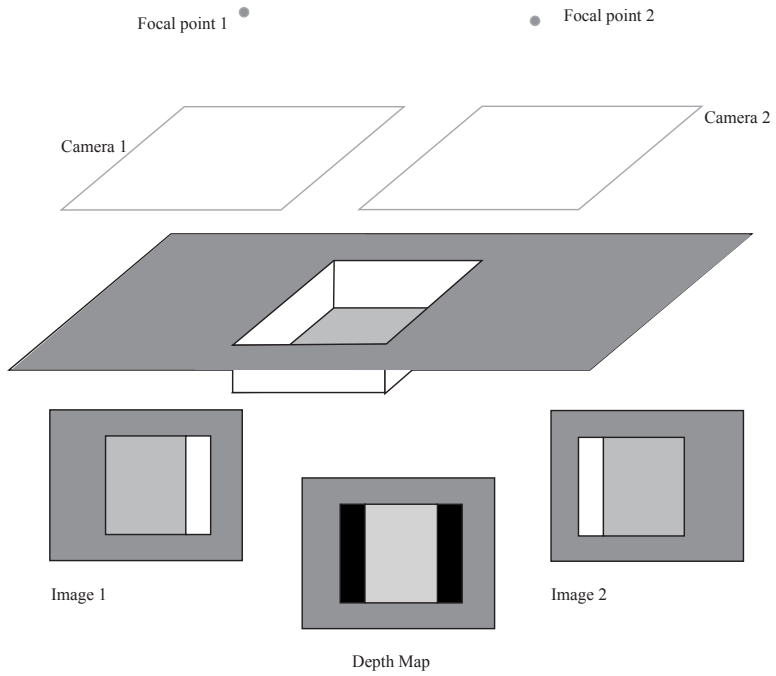

FIGURE 2.9: **Top** *shows two pinhole cameras viewing a rectangular depression in a flat surface. As the images show, camera on the left can see the right wall, and that on the right can see the left wall. This means that these walls cannot be reconstructed directly using trigonometry, and so the depth map will have holes in it. The depth map here is shown with a fairly common convention, where nearer surfaces are lighter, farther surfaces are darker, and holes are "infinitely far away".*

meaning that as $\mathbf{P}$ gets further away, the *disparity* (difference between projected positions in left and right cameras) gets smaller, and so gets harder to measure. Resolving small differences in large depths is going to be hard. This means that either the *baseline* (distance between camera focal points, B in Figure 21.3) is large (and so the equipment is bulky) or one can't reliably measure large depths.

A second important limit is that some points will appear in one camera, but not in the other (an effect known as *Da Vinci stereopsis*, illustrated in Figure 2.9), and so their depth cannot be measured by stereo. The result is quite characteristic "holes" in depth maps obtained from stereo cameras .

2.2.2 Camera-Projector Stereo

The key difficulty in stereo is establishing which point in the left image corresponds to which in the right. This can be tricky even now for some kinds of object. In *camera projector systems*, one uses one camera and one projector. This projector is constructed to have geometry like that of a camera. Light leaves an analog of the focal point, and travels along rays through pixel locations.

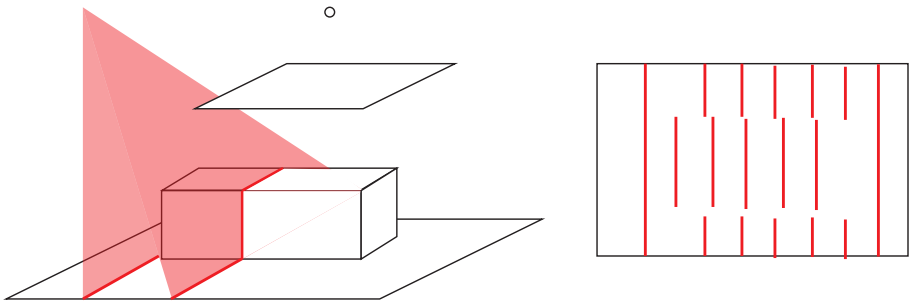

FIGURE 2.10: *The projector on the* **left** *casts planes of light into the scene. These are viewed by a camera. Their shape in the image (***right***) is a cue to the depth in the scene.*

The simplest projector casts planes of light (Figure 2.10) which are swept across the scene. In these *structured light systems*, the pattern of the curves made by those planes reveals depth.

An alternative is to modulate the light through each pixel. In this case, the light through each different pixel location is uniquely identifiable. The geometry of Figure 2.8 still applies, but now the ray from $\mathbf{f}_1$ to $\mathbf{P}$ is a ray of emitted light. Because the geometry hasn't changed, large depths are hard to measure without large baselines, and there will still be holes in depth maps.

A natural modulation trick is for the projector to display a sequence of (say) 8 patterns. Each pixel in each pattern is either dark or light. If the patterns are properly chosen, and if the camera observes all of them, you can think of each ray through the projector focal point as being tagged with eight bits. These eight bits identify the ray. Many rays will have the same bit pattern. If depth limits are known for the scene, and if the patterns are appropriately chosen, this ambiguity is not important.

For any baseline, there will be some practical limit to the largest and smallest depths that can be measured. This has an interesting consequence. In the geometry of Figure **??**, imagine we fire a ray of modulated light from $\mathbf{f}_1$ through $\mathbf{x}_1$. If it is observed in camera 2 (it might not be, because the geometry of Figure 2.9 also still applies), we have a very good idea *where* it will be observed. The y-coordinate will not have changed and the disparity is limited by the depth range. This means we can use the same code for rays through two different points in camera 1 as long as they are sufficiently far apart.

2.2.3 Time of Flight Sensors

Time of flight sensors send a pulse of light out from a laser source, then wait for the pulse to return to a sensor. The time from flash to return yields the depth to the surface along the direction of the flash. A moving mirror ensures that depth can be measured along many rays (a scanning time of flight sensor; Figure 2.11). Accuracy in this class of sensor depends on very accurate measurements of short

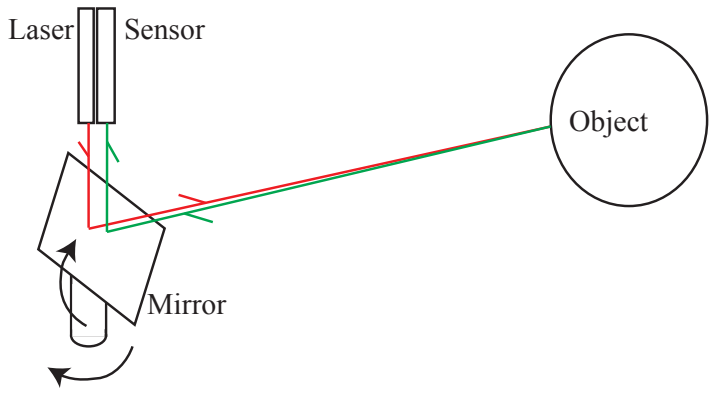

FIGURE 2.11: *A laser flashes a brief pulse of light onto a mirror. The light travels into a scene, is reflected from an object and returns to the mirror, and is reflected into a sensor. The time from flash to sensing reveals the distance tot he object. The mirror can be tilted or rotated to flash the light in different directions, and so to measure depth along different rays.*

time intervals; very accurate timing of the pulses of light. Depth along each ray is measured at a slightly different time, so the mirror needs to scan rays fast enough to ensure that moving objects are captured satisfactorily. Such systems are often called *lidar* (for LIght Detection And Ranging), by analogy with radar.

Remember this: *Depth sensors can be built in a variety of ways. The main kinds are stereo sensors; camera projector sensors; and time of flight sensors. Two camera geometry underpins stereo sensors and camera projector sensors, so that for each kind of sensor, accurate measurements of large depths require large sensors. Time of flight sensors require quite complex engineering, and tend to be more expensive.*

2.3 YOU SHOULD

2.3.1 remember these facts:

2.3.2 be able to:

- Give a brief account of what a camera does.
- Remember how a pinhole camera model works.
- Give a brief account of how depth sensors of each kind work.

EXERCISES

QUICK CHECKS

2.1. As the pinhole in a pinhole camera gets larger, the image on the image plane (a) gets brighter and (b) is less focused. Why?

2.2. For a sufficiently small pinhole, diffraction effects cause the image to be defocused. Explain.

2.3. Why do cameras have lenses?

2.4. For smaller Δt, images will be darker; for larger Δt, moving objects may have blurred outlines. Why?

2.5. A common form of camera response function is $\text{Intensity}(P\Delta t) = Ce^{\gamma P \Delta t}$. Do you expect γ to be positive or negative? Why?

2.6. Why are mosaiced color CCD cameras more common than multiccd cameras?

2.7. You want to increase the contrast in an image by a pointwise image transformation that (a) makes dark pixels darker *and* (b) makes bright pixels brighter. Why will $f(I) = Ce^{\gamma I}$ not achieve this? Sketch a transformation that will.

2.8. Why is it hard to build a stereo camera that is (a) small and (b) good at measuring large depths?

2.9. What happens in a scanning time of flight sensor if the mirror moves slowly and objects in the scene move quickly?

2.10. An object is 30m away from a scanning time of flight sensor. How long does it take for the pulse to travel from camera to object to sensor? You can take the speed of light to be 3×10^8 meters per second.

LONGER PROBLEMS

2.11. Show that, in the geometry of Figure 2.8, $d = -f\frac{B}{Z}$.