CHAPTER 19

Image Capture

19.1 CAMERAS

19.1.1 The Pinhole Camera

A *pinhole camera* is a light-tight box with a very small hole in the front (Figure 19.1). 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 19.1). An appropriate sensor (CMOS sensor; CCD sensor; light sensitive film) at the back of the box will capture this image.

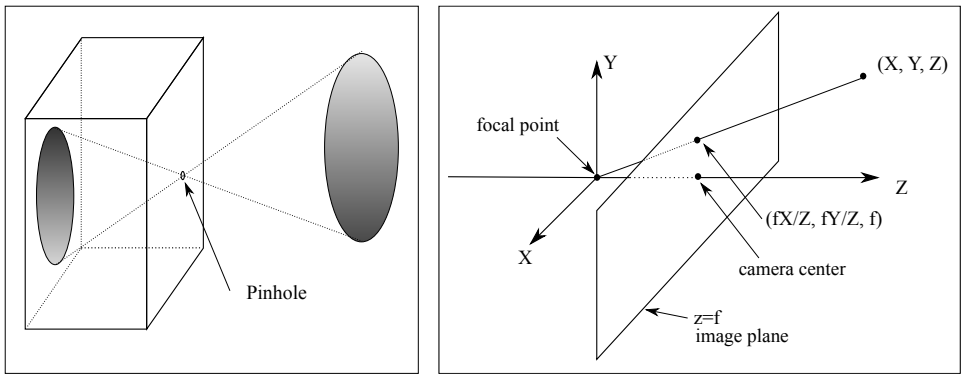


FIGURE 19.1: *The pinhole imaging model. On the* **left**, *a light-tight box with a pinhole in it views an object. The only light that a point on the back of the box sees comes through the very small pinhole, so that an inverted image is formed on the back face of the box. On the* **right**, *the usual geometric abstraction. 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 coordinate system is* left-handed, *because the camera looks down the z-axis. This is because most people's intuition is that z* increases *as one moves into the image. The text provides some more detail on this point.*

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-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. The coordinate system is left-handed even though coordinate

systems in 3D are usually right-handed coordinate systems. This is because most people's intuition is that z *increases* as one moves into the image. 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 point (X, Y, Z) in 3D is imaged to

$$(fX/Z, fY/Z, f)$$

on the sensor (Figure 19.1). 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

$$(X, Y, Z) \to (fX/Z, fY/Z).$$

The focal length just scales the image. In standard camera models, other scaling effects occur as well, and we write projection as if $f = 1$, yielding

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

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*. Remarkably, in almost every publication in computer vision the camera is expressed in left-handed coordinates and everything else works in right-handed coordinates. The exercises demonstrate that there is no real difficulty here.

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

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

Figure 19.1 shows important terminology (focal point; image plane; camera center).

19.1.2 Perspective Effects

Perspective projection has a number of important properties, summarized as:

- lines project to lines;
- more distant objects are smaller;

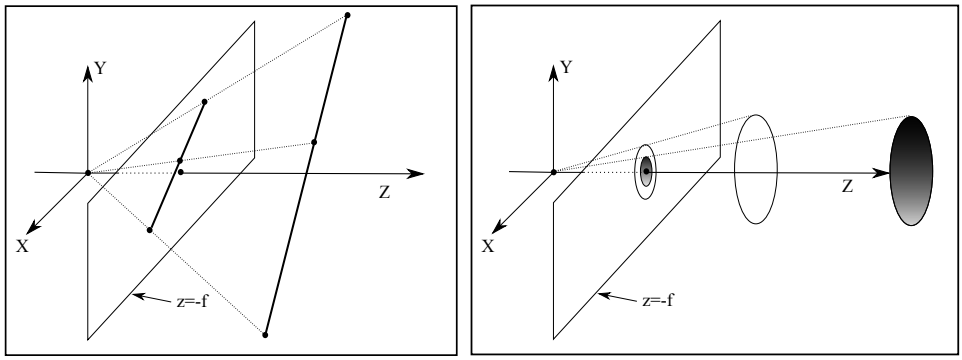


FIGURE 19.2: *Perspective projection maps almost any 3D line to a line in the image plane (***left***). Some rays from the focal point to points on the line are shown as dotted lines. The family of all such rays is a plane, and that plane must intersect the image plane in a line as long as the 3D line does not pass through the focal point. On the* **right***, two 3D objects viewed in perspective projection; the more distant object appears smaller in the image.*

- lines that are parallel in 3D meet in the image;
- planes have horizons;
- planes image as half-planes.

Lines project to lines: Almost every line in 3D maps to a line in the image. You can see this by noticing that the image of the 3D line is formed by intersecting rays from the focal point to each point on the 3D line with the image plane. But these rays form a plane, so we are intersecting a plane with the image plane, and so obtain a line (Figure 19.2). The exceptions are the 3D lines through the focal point – these project to points.

More distant objects are smaller: The further away an object is in 3D, the smaller the image of that object, because of the division by Z (Figure 19.2).

Lines that are parallel in 3D meet in the image: Now think about a set of infinitely long parallel railroad tracks. The sleepers supporting the tracks are all the same size. Distant sleepers are smaller than nearby sleepers, and arbitrarily distant sleepers are arbitrarily small. This means that parallel lines will meet in the image. The point at which the lines in a collection of parallel lines meet is known as the *vanishing point* for those lines (Figure 19.3). The vanishing point for a set of parallel lines can be obtained by intersecting the ray from the focal point and parallel to those lines with the image plane (Figure 19.3).

Planes have horizons: Now think about the image of a plane. As Figure 19.5 shows, the plane through the focal point and parallel to that plane produce a line in the image, known as the *horizon* of the plane.

Planes image as half-planes: For an abstract perspective camera, any point on the plane can be imaged to a point on the image plane. In practical cameras, we cannot image points that lie behind the camera in 3D. Now cast a ray through the focal point and some point $\mathbf{x}$ in the image plane. If $\mathbf{x}$ is on one side of

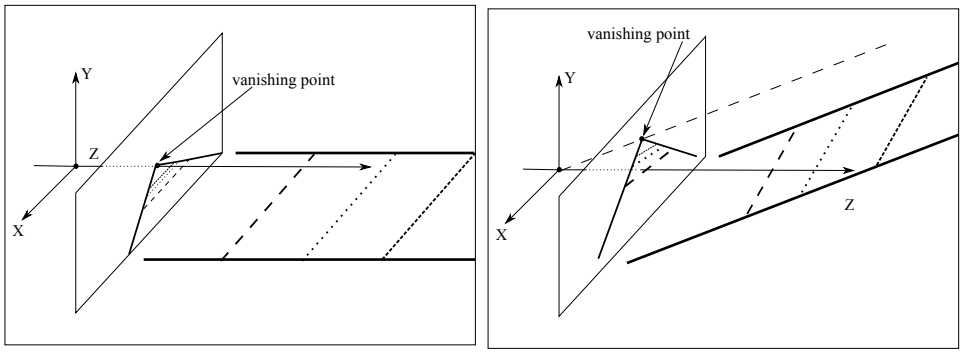


FIGURE 19.3: *Perspective projection maps a set of parallel lines to a set of lines that meet in a point. On the* **left**, *a set of lines parallel to the z-axis, with "railway sleepers" shown. As these sleepers get further away, they get smaller in the image, meaning the projected lines must meet. The vanishing point (the point where they meet) is obtained by intersecting the ray parallel to the lines and through the focal point with the image plane. On the* **right**, *a different pair of parallel lines with a different vanishing point. The figure establishes that, if there are more than two lines in the set of parallel lines, all will meet at the vanishing point.*

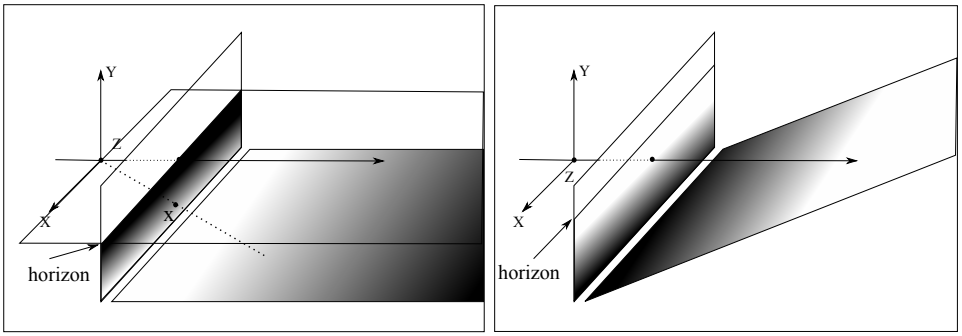


FIGURE 19.4: **Left** *shows a plane in 3D (in this case, $y = -1$). The intersection of the plane through the focal point parallel to the 3D plane (in this case, $y = 0$) and the image plane, forms an image line called the horizon. This line cuts the image plane into two parts. Construct the ray through the focal point and a point* $\mathbf{x}$ *in the image plane. For* $\mathbf{x}$ *on one side of the horizon, this ray will intersect the 3D plane in the half space $z > 0$ (and so in front of the camera, shown here). If* $\mathbf{x}$ *is on the other side of the horizon, the intersection will be in the half space $z < 0$ (and so behind the camera, where it cannot be seen).* **Right** *shows a different 3D plane with a different horizon. The gradients on the planes indicate roughly where points on the 3D plane appear in the image plane (light points map to light, dark to dark).*

the horizon, the ray will hit the plane in the $z > 0$ half space and so we can see the plane. If it is on the other side, it will hit the plane in the $z < 0$ half space, so we cannot see the plane.

Remember this: *Under perspective projection:*

- *points project to points;*
- *lines project to lines;*
- *more distant objects are smaller;*
- *lines that are parallel in 3D meet in the image;*
- *planes have horizons;*
- *planes image as half-planes.*

19.1.3 Scaled Orthographic Projection and Orthographic Projection

Under some circumstances, perspective projection can be simplified. Assume the camera views a set of points which are close to one another compared with the distance to the camera. Write $\mathbf{X}_i = (X_i, Y_i, Z_i)$ for the i'th point, and assume that $Z_i = Z(1+\epsilon_i)$, where ϵ_i is quite small. In this case, the distance to the set of points is much larger than the *relief* of the points, which is the distance from nearest to furthest point. The i'th point projects to $(fX_i/Z_i, fY_i/Z_i)$, which is approximately $(f(X_i/Z)(1-\epsilon_i), f(Y_i/Z)(1-\epsilon_i))$. Ignoring ϵ_i because it is small, we have the projection model

$$(X, Y, Z) \to (f/Z)(X, Y) = s(X, Y).$$

This model is usually known as *scaled orthograpic projection*. The model applies quite often. One important example is pictures of people. Very often, all body parts are roughly the same distance from the camera — think of a side view of a pedestrian seen from a motor car. Scaled orthographic projection applies in such cases. It is not always an appropriate model. For example, when a person is holding up a hand to block the camera's view, perspective effects can be significant (Figure **??**).

Occasionally, it is useful to rescale the camera (or assume that $f/Z = 1$), yielding $(X, Y, Z) \to (X, Y)$. This is known as *orthographic projection*.

Remember this: *Scaled orthographic projection maps*

$$(X, Y, Z) \to s(X, Y)$$

where s is some scale. The model applies when the distance to the points being viewed is much greater than their relief. Many views of people have this property.

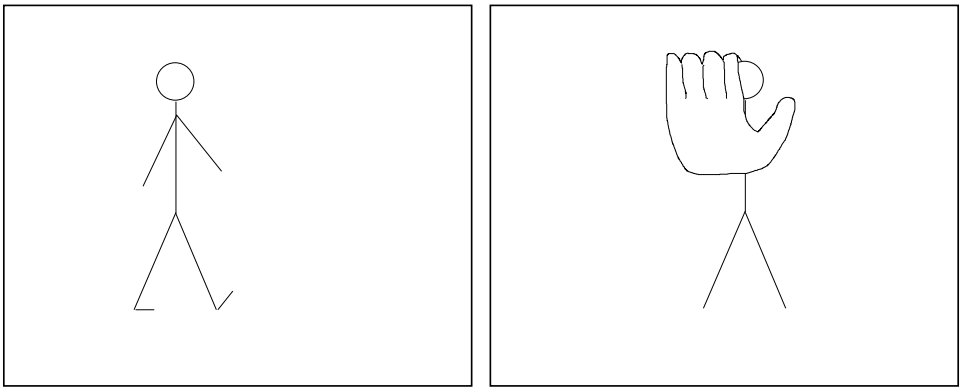


FIGURE 19.5: *The pedestrian on the* **left** *is viewed from some way away, so the distance to the pedestrian is much larger than the change in depth over the pedestrian. In this case, which is quite common for views of people, scaled orthography will apply. The celebrity on the* **right** *is holding a hand up to prevent the camera viewing their face; the hand is quite close to the camera, and the body is an armslength away. In this case, perspective effects are strong. The hand looks big because it is close, and the head looks small because it is far.*

19.1.4 Lenses

One practical version of a pinhole camera is a *camera obscura* – the box is built as a room, and you can stand in the room and see the view on the back wall (some examples are at `https://www.atlasobscura.com/lists/camera-obscura-places`; the internet yields amusing disputes about the correct plural form of the term). You can also build a simple pinhole camera with a matchbox, some tape, a pin, and some light sensitive film do the trick. Getting good images takes trouble, however.

A large hole in front of the camera will cause the image at the back to be brighter, but blurrier, because each point on the sensor will average light over all directions that happen to go through the hole. If the hole is smaller, the image will get sharper, but darker. In practical cameras, achieving an image that is both bright and focused is the job of the lens system. There may be one or several lenses that light passes through before reaching the sensor at the back of the camera. Each of these lenses is built from refracting materials. The shape and position of the lenses, together with the refractive index of the materials they are built of, determine the path that light follows through the lens system. Generally, the lens system is designed to collect as much light as possible at the input and produce a focused image on the image plane. Remarkably, the many or most lens systems result in an imaging geometry that can be modelled with a pinhole camera model, and lens system effects are ignored in all but quite specialized applications of computer vision.

Lens systems are designed and modelled using geometric optics, but lens designs always involve compromises. The result is that cameras with lenses differ from pinhole cameras in some ways that are worth knowing about, although they are not always important. First, in an abstract pinhole camera, all objects at what-

ever distance are in focus. Geometric optics means that a lens with this property admits very little light, so it is common to work with cameras that have a limited *depth of field* – the range of distances to the camera over which objects are in focus on the image plane. Second, manufacturing difficulties and cost considerations mean that lenses will have various *aberrations*. The net effect of most aberrations is a tendency to defocus some objects under some circumstances, but *chromatic aberrations* can cause colors to be less crisp and objects to have "halos" of color. Chromatic aberration occurs because light of different wavelengths takes slightly different paths through a refracting object. Various lens coatings can correct chromatic aberration, but the resulting lens system will be more expensive. Third, in most lens systems, the periphery of the image tends to be brighter than it would be in a pure pinhole camera. For more complex lens systems, an effect in the lens known as *vignetting* can darken the periphery somewhat. Finally, lenses may cause *geometric distortions* of the image. The most noticeable effect of these distortions is that straight lines in the world may project to curves in the image. Most common is *barrel distortion*, where a square is imaged as a bulging barrel; *pincushion distortion*, where the square bulges in rather than out, can occur (Figure **??**).

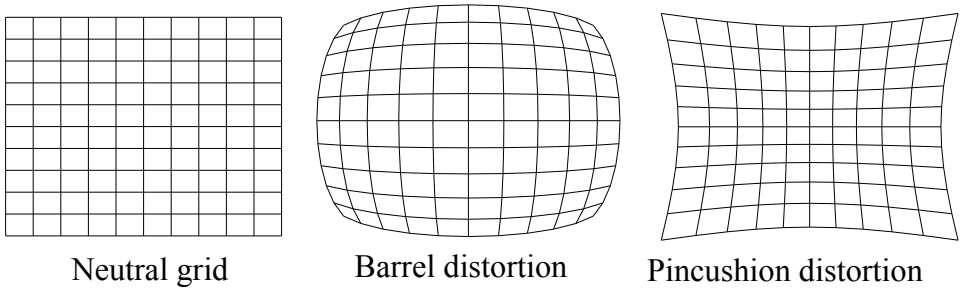


FIGURE 19.6: *On the* **left** *a neutral grid observed in a non-distorting lens (and viewed frontally to prevent any perspective distortion).* **Center** *shows the same grid, viewed in a lens that produces barrel distortion.* **Right**, *the same grid, now viewed in a lens that produces pincushion distortion.*

19.2 DEPTH MEASUREMENT

The cameras of chapter 35.2 project points in 3D to points on an image plane. Building such cameras is now very well understood (and they are extremely cheap). A lot is known about how to recover the points in 3D from the projected versions under various circumstances (some of this appears in chapters 35.2), but doing so can be inconvenient. It is often very useful to measure the 3D location of points directly.

19.2.1 Stereoscopic Depth Measurement

Stereo uses two cameras somewhat offset from one another. Figure 35.2 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 35.2), as is determining which (if any) point in the first image corresponds to which in the second (Chapter 35.2), and recovering a good depth model from this information (Chapter 35.2). 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.

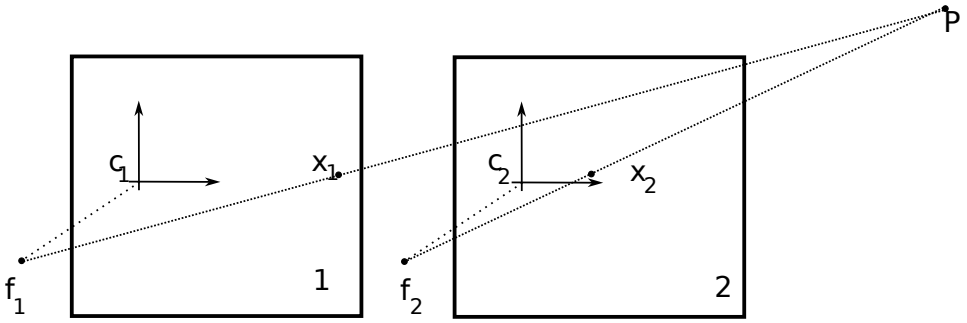
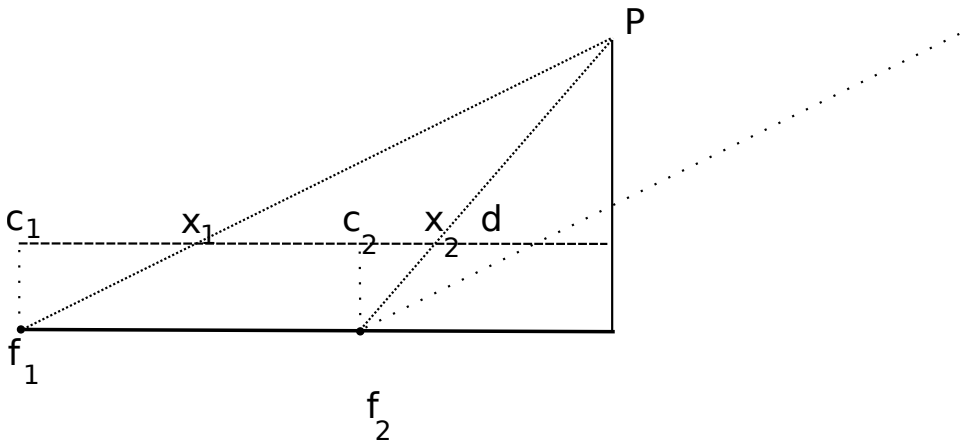


FIGURE 19.7: *When two pinhole cameras view a point, the 3D coordinates of the point can be reconstructed from the two images of that point. This applies for almost every configurations of the cameras. It is an elementary exercise in trigonometry (exercises) to determine* $\mathbf{P}$ *from the positions 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 35.2). 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.*

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 19.7 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

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

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 35.2) is large (and so the equipment is bulky) or one can't reliably measure large depths.



ht

FIGURE 19.8: **Right**: shows two important triangles in the plane spanned by the two focal points ($\mathbf{f}_1$, $\mathbf{f}_2$) and the point being viewed ($\mathbf{P}$). The extent of the shift leftwards (the *disparity*, d in the figure) reveals the depth to the point. Comparing triangle $\mathbf{f}_1$, $\mathbf{p}$, $\mathbf{R}$ with triangle $\mathbf{f}_2$, $\mathbf{p}$, $\mathbf{R}$ yields the relationship between depth, disparity and *baseline* (the distance between the two focal points.

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 19.9), and so their depth cannot be measured by stereo. The result is quite characteristic "holes" in depth maps obtained from stereo cameras .

19.2.2 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. One could use 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. Modulation tricks mean the light through each different pixel location is uniquely identifiable. The geometry of Figure 19.7 still applies, but now the ray from $\mathbf{f}_1$ to $\mathbf{P}$ is a ray of emitted light.

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 19.7, 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 19.9 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.

Camera projector stereo uses the same geometry as two camera stereo, so that large depths are hard to measure without large baselines, and there will still be holes in depth maps.

19.2.3 Structured light

Structured light uses

19.2.4 Time of flight sensors and Lidar

could fire light out from a location, then wait till it returns. The length of the wait and the speed of light reveal the depth to the point (Figure 35.2).

PROBLEMS

19.1. Use Figure 35.2, and write B for the distance between $\mathbf{f}_L$ and $\mathbf{f}_R$ and $\mathbf{v}_L$ for the unit vector between $\mathbf{f}_L$ and $\mathbf{X}_L$.

(a) Show that the point

$$\mathbf{P} = \mathbf{f}_L + \mathbf{v}_L \left[\frac{B}{\cos\theta_L + \cos\theta_R \left(\frac{\sin\theta_L}{\sin\theta_R} \right)} \right]$$

(b) Show that the point

$$\mathbf{P} = \mathbf{f}_R + \mathbf{v}_R \left[\frac{B}{\cos\theta_R + \cos\theta_L \left(\frac{\sin\theta_R}{\sin\theta_L} \right)} \right]$$

(c) Under what circumstances could these two expressions produce different results? (hint: $\mathbf{f}_L$, $\mathbf{f}_R$, $\mathbf{X}_L$, $\mathbf{X}_R$ and $\mathbf{P}$ are coplanar, but what happens if $\mathbf{X}_L$ and $\mathbf{X}_R$ are measured with small errors?)

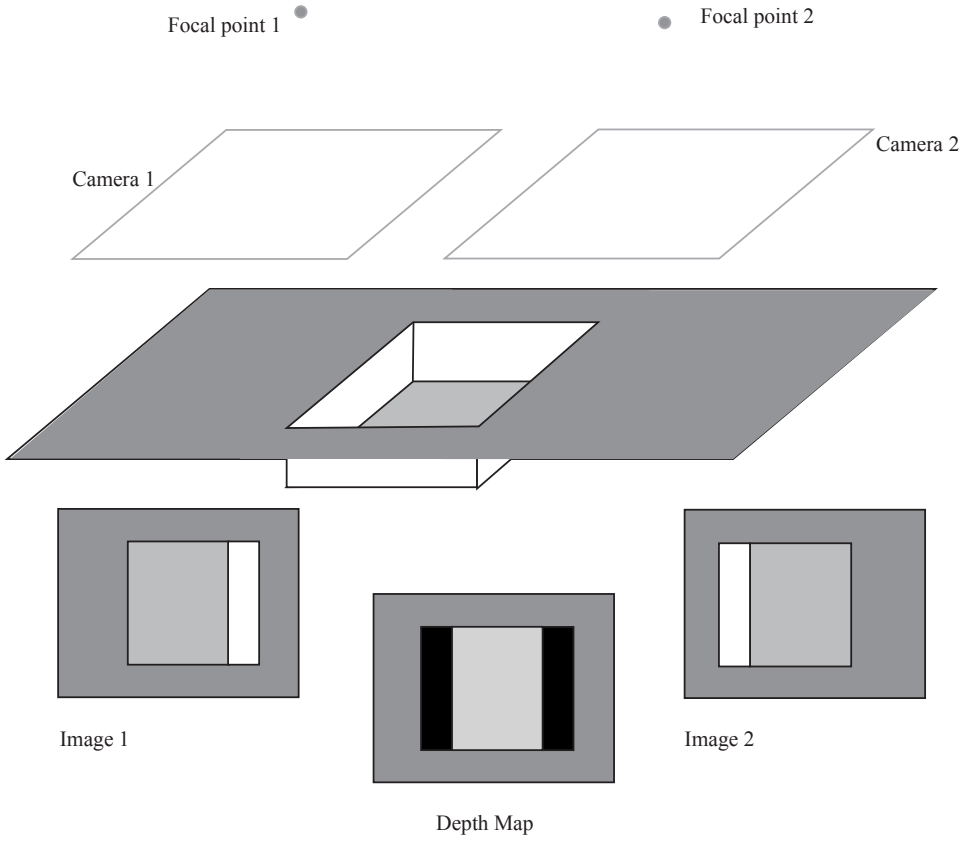


FIGURE 19.9: **Top** *shows two pinhole cameras viewing a rectangular depression in a flat surface. As the images show, camera on the left can see the right wall, and that on the right can see the left wall. This means that these walls cannot be reconstructed directly using trigonometry, and so the depth 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".*