CHAPTER 2

Single-view Geometry

When we open an eye or take a photograph, we see only a flattened, two-dimensional projection of the physical underlying scene. The consequences are numerous and startling. Size relationships are distorted, right angles are suddenly wrong, and parallel lines now intersect. Objects that were once apart are now overlapping in the image, so that some parts are not visible. At first glance, it may seem that the imaging process has corrupted a perfectly reasonable, well-ordered scene into a chaotic jumble of colors and textures. However, with careful scene representation that accounts for perspective projection, we can tease out the structure of the physical world.

In this chapter, we summarize the consequences of the imaging process and how to mathematically model perspective projection. We will also show how to recover an estimate of the ground plane, which allows us to recover some of the relations among objects in the scene.

1. Consequences of Projection

The photograph is a projection of the 3D scene onto a 2D image plane. Most cameras use a **perspective projection**. Instead of recording the 3D position of objects (X, Y, Z), we observe their projection onto the 2D image plane at (u, v). In this projection, a set of parallel lines in 3D will intersect at a single point, called the **vanishing point**. A set of parallel planes in 3D intersect in a single line, called the **vanishing line**. The vanishing line of the ground plane, called the **horizon**, is of particular interest because many objects are oriented according to the ground and gravity. The horizon is discussed further in Section 2.

Another major consequence of projection is **occlusion**. Because light does not pass through most objects, only the nearest object along a ray is visible. Take a glance at the world around you. Every object is occluding something, and most are partially occluded themselves. In some ways,

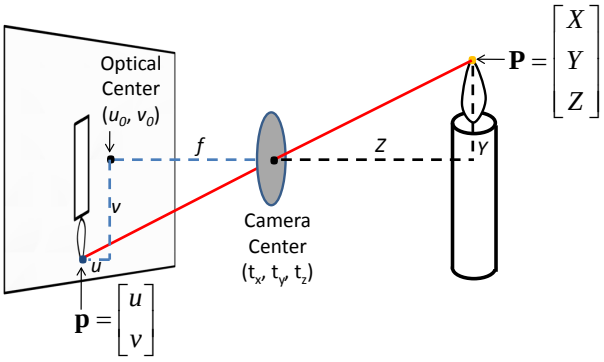

FIGURE 2.1. Illustration of pinhole camera model. The camera, with focal length f and optical center (u_0, v_0), projects from a 3D point in the world to a 2D point on the image plane.

occlusion simplifies the scene interpretation. Imagine if you could see every object for miles in a single projection! In other ways, occlusion makes scene interpretation more difficult by hiding parts and disturbing silhouettes.

2. Perspective Projection with Pinhole Camera: 3D to 2D

We often assume a **pinhole camera** to model projection from 3D to 2D, as shown in Figure 2.1. Under this model, light passes from the object through a small pinhole onto a sensor. The pinhole model ignores lens distortion and other non-linear effects, modeling the camera in terms of its intrinsic parameters (focal length f, optical center (u_0, v_0), pixel aspect ratio α, and skew s) and extrinsic parameters (3D rotation $\mathbf{R}$ and translation $\mathbf{t}$). Although the image plane is physically behind the pinhole, as shown in the figure, it is sometimes convenient to pretend that the image plane is in front, with focal length f, so that the image is not inverted. If we assume no rotation ($\mathbf{R}$ is identity), and camera translation of $(0,0,0)$, with no skew and unit aspect ratio, we can use the properties of similar triangles to show that $u - u_0 = f\frac{Y}{Z}$ and $v - v_0 = f\frac{X}{Z}$.

We can more easily write the projection as a system of linear equations in **homogeneous coordinates**. These homogeneous coordinates add a scale coordinate to the Cartesian coordinates, making them convenient for representing rays (as in projection) and direction to infinitely distant points (e.g., where 3D parallel lines intersect). To convert from Cartesian to homogeneous coordinates, append a value of 1 (e.g., $(u,v) \rightarrow (u,v,1)$ and $(X,Y,Z) \rightarrow (X,Y,Z,1)$). To convert from homogeneous to Cartesian coordinates, divide by the last coordinate (e.g., $(u,v,w) \rightarrow (u/w, v/w)$).

Under homogeneous coordinates, our model of pinhole projection from 3D world coordinates (X,Y,Z) to image coordinates (u,v) is written as follows:

$$(2.1)\quad \begin{bmatrix} w\cdot u \\ w\cdot v \\ w \end{bmatrix} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad or \quad \begin{bmatrix} w\cdot u \\ w\cdot v \\ w \end{bmatrix} = \begin{bmatrix} f & s & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

Note that the intrinsic parameter matrix $\mathbf{K}$ has only five parameters and that the rotation and translation each have three parameters, so that there are 11 parameters in total. Even though the rotation matrix has nine elements, each element can be computed from the three angles of rotation. We commonly assume unit aspect ratio and zero skew as a good approximation for most modern cameras. Sometimes, it is also convenient to define world coordinates according to the camera position and orientation, yielding the simplified

$$(2.2)\quad \begin{bmatrix} w\cdot u \\ w\cdot v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

From this equation, we get the same result as we found using similar triangles in Figure 2.1. In Cartesian coordinates, $u = f\frac{Y}{Z} + u_0$ and $v = f\frac{X}{Z} + v_0$.

3. 3D Measurement from a 2D Image

As illustrated in Figure 2.2, an infinite number of 3D geometrical configurations could produce the same 2D image, if only photographed from the correct perspective. Mathematically, therefore, we have no way to recover 3D points or measurements from a single image. Fortunately, because our world is so structured, we *can* often make good estimates. For example, if we know how the camera is rotated with respect to the ground plane and vertical direction, we can recover the relative 3D heights of objects from their 2D positions and heights.

Depending on the application, the world coordinates can be encoded using either three orthogonal vanishing points or the projection matrix, leading to different approaches to 3D measurement. When performing estimation from a single image, it is helpful to think in terms of the horizon line (the vanishing line of the ground plane) and the vertical vanishing point.

For example, look at the 3D scene and corresponding image in Figure 2.3. Suppose the camera is level with the ground, so that the vertical vanishing point is at infinity and the horizon is a line through row v_h. Then, if a grounded object is as tall as the camera is high, the top of the object will be projected to v_h as well. We can use this and basic trigonometry to show that $\frac{Y_o}{Y_c} = \frac{v_t - v_b}{v_h - v_b}$, where Y_o is the 3D height at the top of the object (defining the ground at $Y = 0$), Y_c is the camera

Original Image | Interpretation 1: Painting on Ground

Interpretation 2: Floating Objects | Interpretation 3: Man in Field

FIGURE 2.2. Original image and novel views under three different 3D interpretations. Each of these projects back into the input image from the original viewpoint. Although any image could be produced by any of an infinite number of 3D geometrical configurations, very few are plausible. Our main challenge is to learn the structure of the world to determine the most likely solution.

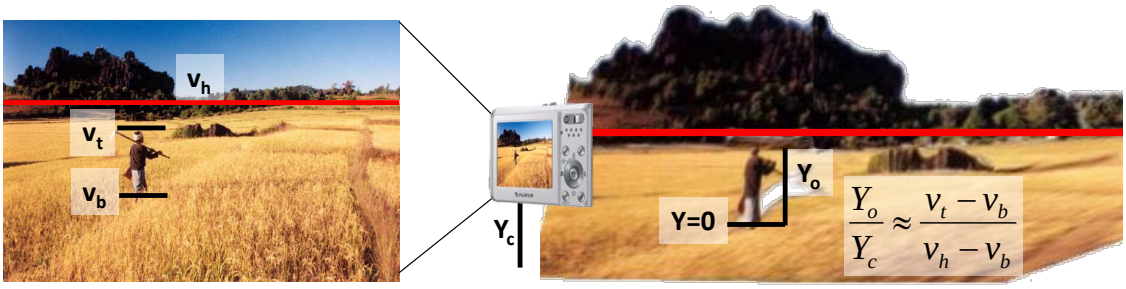

FIGURE 2.3. The measure of man. Even from one image, we can measure the object's height, relative to the camera height, if we can see where it contacts the ground.

height, and v_t and v_b are the top and bottom positions of the object in the image. This relationship is explored in more detail in Hoiem et al. [**43**] and used to aid in object recognition, a topic we will discuss in a later chapter.

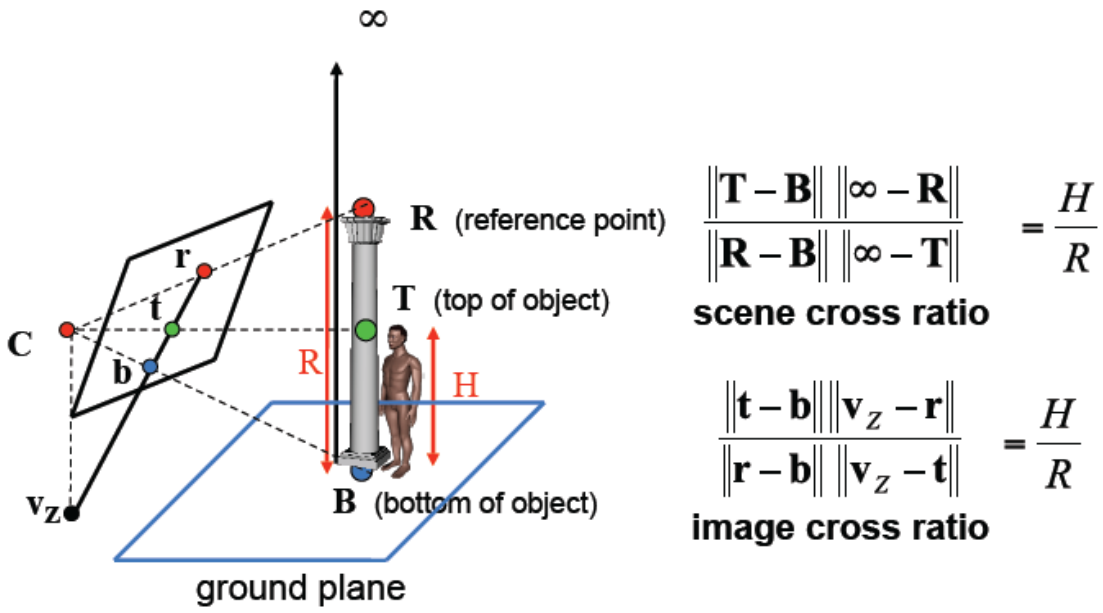

FIGURE 2.4. This figure by Steve Seitz illustrates the use of the cross-ratio to make world measurements from an image. Need to get permission to use this figure.

As shown by Criminisi et al. [**16**], the **cross ratio invariant** can be used to relate the 3D heights of objects under more general conditions. The cross ratio of 4 collinear points (e.g., $\frac{\|\mathbf{p}_3-\mathbf{p}_1\|\|\mathbf{p}_4-\mathbf{p}_2\|}{\|\mathbf{p}_3-\mathbf{p}_2\|\|\mathbf{p}_4-\mathbf{p}_1\|}$, for any collinear points $\mathbf{p}_{1..4}$) does not change under projective transformations. If one of those points is a vanishing point, then its 3D position is at infinity, simplifying the ratio. Using the vertical vanishing point, we can recover the relative heights of objects in the scene (see Figure 2.4 for an illustration).

We have a special case when the camera is upright and not slanted so that the vertical vanishing point is at infinity in the image plane. In this case, $v_Z = \infty$, so we have $\frac{Y_t-Y_b}{Y_c-Y_b} = \frac{v_t-v_b}{v_h-v_b}$, where Y_t is the height of the top of the object, Y_b is the height at the bottom, Y_c is the height of the camera, and v_t, v_b, and v_h, are the image row coordinates of the object and horizon.

4. Automatic Estimation of Vanishing Points

Recall that all lines with the same 3D orientation will converge to a single point in an image, called a vanishing point. Likewise, sets of planes converge to a vanishing line in the image plane. If a set of parallel lines is also parallel to a particular plane, then their vanishing point will lie on the vanishing line of the plane.

If we can recover vanishing points, we can use them to make a good guess at the 3D orientations of lines in the scene. As we will see, if we can recover a triplet of vanishing points that correspond to orthogonal sets of lines, we can solve for the focal length and optical center of the camera. Typically, two of the orthogonal vanishing points will be on the horizon and the third will be in the vertical direction. If desired, we can solve for the rotation matrix that sets the orthogonal directions as the X, Y, and Z directions.

Most vanishing point detection algorithms work by detecting lines and clustering them into groups that are nearly intersecting. If we suspect that the three dominant vanishing directions will be mutually orthogonal, e.g., as in architectural scenes, we can constrain the triplet of vanishing points to provide reasonable camera parameters. However, there are a few challenges that make the problem of recovering vanishing points robustly much more tricky than it sounds. In the scene, many lines, such as those on a tree's branches, point in unusual 3D orientations. Nearly all pairs of lines will intersect in the image plane, even if they are not parallel in 3D, leading to many spurious candidates for vanishing points. Further, inevitable small errors in line localization cause sets of lines that really are parallel in 3D not to intersect at a single point. Finally, an image may contain many lines, and it is difficult to quickly find the most promising set of vanishing points.

Simple algorithm to estimate three mutually orthogonal vanishing points:

1. Detect straight line segments in an image, represented by center (u_i, v_i) and angle θ_i in radians. Ignore lines fewer than L pixels (e.g., L=20), because these are less likely to be correct detections or to have large angular error. Kosecka and Zhang [**52**] describe an effective line detection algorithm (see Derek Hoiem's software page for an implementation).
2. Find intersection points of all pairs of lines to create a set of vanishing point candidates. It's easiest to represent points in homogeneous coordinates, $\mathbf{p}_i = [w_i u_i \; w_i v_i \; w_i]^T$, and lines in the form $\mathbf{l}_i = [a_i b_i c_i]^T$ using the line equation $ax + by + c = 0$. The intersection of lines $\mathbf{l}_i$ and $\mathbf{l}_j$ is given by their cross product: $\mathbf{p}_{ij} = \mathbf{L}_i \times \mathbf{L}_j$. If the lines are parallel, then $w_{ij} = 0$. Likewise, the line formed by two points $\mathbf{p}_i$ and $\mathbf{p}_j$ is given by their cross product: $\mathbf{L}_{ij} = \mathbf{p}_i \times \mathbf{p}_j$.
3. Compute a score for each vanishing point candidate: $s_j = \sum_i |l_i| \exp\left(-\frac{|\alpha_i - \theta_i|}{2\sigma^2}\right)$, where $|l_i|$ is the length of line segment i, α_i is the angle from the center of the line segment (u_i, v_i) to (u_j, v_j) in radians, and σ is a scale parameters (e.g., $\sigma = 0.1$). In the distance between angles, note that $-2\pi = 0 = 2\pi$.

4. Choose the triplet with the highest total score that also leads to reasonable camera parameters. For a given triplet, $\mathbf{p}_i$, $\mathbf{p}_j$, $\mathbf{p}_k$, the total score is $s_{ijk} = s_i + s_j + s_k$. Orthogonality constraints specify: $\left(\mathbf{K}^{-1}\mathbf{p}_i\right)^T \left(\mathbf{K}^{-1}\mathbf{p}_j\right) = 0$; $\left(\mathbf{K}^{-1}\mathbf{p}_i\right)^T \left(\mathbf{K}^{-1}\mathbf{p}_k\right) = 0$; and $\left(\mathbf{K}^{-1}\mathbf{p}_j\right)^T \left(\mathbf{K}^{-1}\mathbf{p}_k\right) = 0$, where $\mathbf{K}$ is the 3x3 matrix in Eq. 2.2 and the points are in homogeneous coordinates. With three equations and three unknowns, we can solve for u_0, v_0, and f. We can consider (u_0, v_0) within the image bounds and $f > 0$ to be reasonable parameters.

The algorithm described above is a simplified version of the methods in Rother [**81**] and Hedau et al. [**36**]. Many other algorithms are possible, such as [**52, 95, 7**], that improve efficiency, handle vanishing points that are not mutually orthogonal, incorporate new priors, or have different ways of scoring the candidates. Implementations from Tardif [**95**] and Barinova et al. [**7**] are available online.