# Course Notes 1: Camera Models[*]

## 1 Introduction

The camera is one of the most essential tools in computer vision. It is the mechanism by which we can record the world around us and use its output - photographs - for various applications. Therefore, one question we must ask in introductory computer vision is: how do we model a camera?

## 2 Pinhole cameras

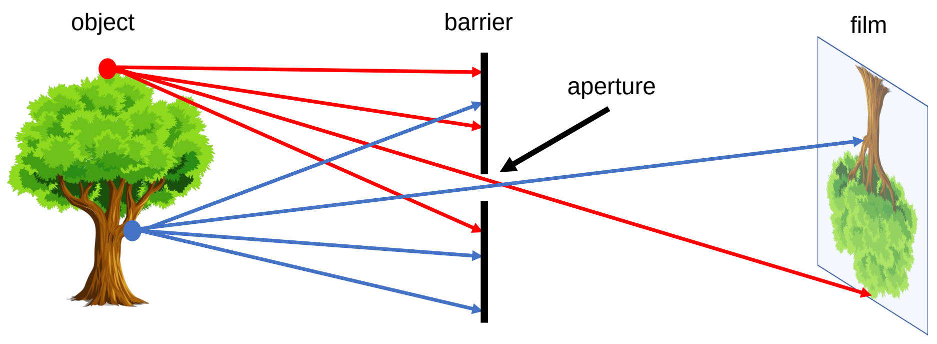

Figure 1: A simple working camera model: the pinhole camera model.

Let's design a simple camera system – a system that can record an image of an object or scene in the 3D world. This camera system can be designed by placing a barrier with a small aperture between the 3D object and a photographic film or sensor. As Figure 1 shows, each point on the 3D object emits multiple rays of light outwards. Without a barrier in place, every point on the film will be influenced by light rays emitted from every point on the 3D object. Due to the barrier, only one (or a few) of these rays of light passes through the aperture and hits the film. Therefore, we can establish a one-to-one mapping between points on the 3D object and the film. The result is that the film gets exposed by an "image" of the 3D object by means of this mapping. This simple model is known as the *pinhole camera model.*

A more formal construction of the pinhole camera is shown in Figure 2. In this construction, the film is commonly called the *image or retinal plane.* The aperture is

[*]Most contents are from
- R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision (2nd Edition)
- K. Hata and S. Savarese. Course notes of Stanford CS231A

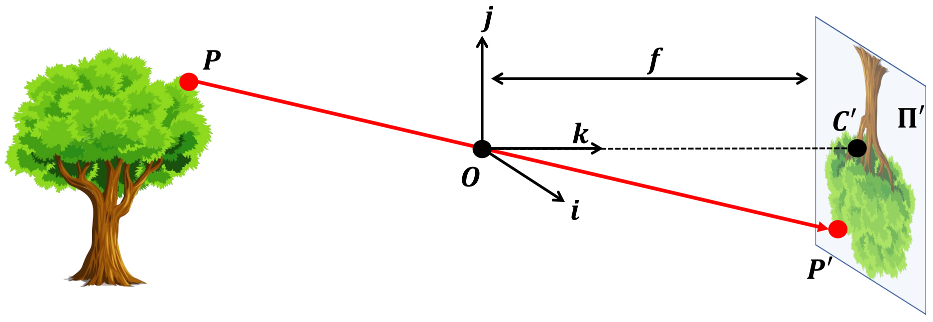

Figure 2: A formal construction of the pinhole camera model.

referred to as the *pinhole* **O** or *center of the camera*. The distance between the image plane and the pinhole **O** is the *focal length* $f$. Sometimes, the retinal plane is placed between **O** and the 3D object at a distance $f$ from **O**. In this case, it is called the *virtual image* or *virtual retinal plane*. Note that the projection of the object in the image plane and the image of the object in the virtual image plane are identical up to a scale (similarity) transformation.

Now, how do we use pinhole cameras? Let $\mathbf{P} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ be a point on some 3D object visible to the pinhole camera. **P** will be mapped or *projected* onto the image plane $\Pi'$, resulting in point[1] $\mathbf{P}' = \begin{bmatrix} x' & y' \end{bmatrix}^T$. Similarly, the pinhole itself can be projected onto the image plane, giving a new point $\mathbf{C}'$.

Here, we can define a coordinate system $\begin{bmatrix} i & j & k \end{bmatrix}$ centered at the pinhole **O** such that the axis **k** is perpendicular to the image plane and points toward it. This coordinate system is often known as the *camera reference system* or *camera coordinate system*. The line defined by $\mathbf{C}'$ and **O** is called the *optical axis* of the camera system.

Recall that point $\mathbf{P}'$ is derived from the projection of 3D point **P** on the image plane $\Pi'$. Therefore, if we derive the relationship between 3D point **P** and image plane point $\mathbf{P}'$, we can understand how the 3D world imprints itself upon the image taken by a pinhole camera. Notice that triangle $\mathbf{P}'\mathbf{C}'\mathbf{O}$ is similar to the triangle formed by **P**, **O**, and $(0, 0, z)$. Therefore, using the law of similar triangles we find that:

$$\mathbf{P}' = \begin{bmatrix} x' & y' \end{bmatrix}^T = \begin{bmatrix} f\frac{x}{z} & f\frac{y}{z} \end{bmatrix}^T \tag{1}$$

Notice that one large assumption we make in this pinhole model is that the aperture is a single point. In most real world scenarios, however, we cannot assume the aperture can be infinitely small. Thus, what is the effect of varying aperture size?

As the aperture size increases, the number of light rays that passes through the barrier consequently increases. With more light rays passing through, then each point on the film may be affected by light rays from multiple points in 3D space, blurring the image. Although we may be inclined to try to make the aperture as small as possible, recall that a smaller aperture size causes less light rays to pass through, resulting in crisper but darker images. Therefore, we arrive at the fundamental problem presented by the pinhole formulation: can we develop cameras that take crisp and bright images?

[1]Throughout the course notes, let the prime superscript (e.g., $\mathbf{P}'$) indicate that this point is a projected or complementary point to the non-superscript version. For example, $\mathbf{P}'$ is the projected version of **P**.

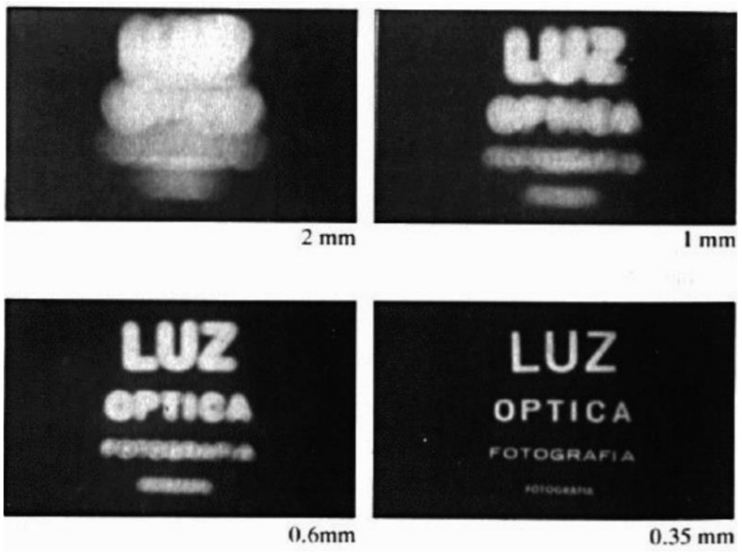

Figure 3: The effects of aperture size on the image. As the aperture size decreases, the image gets sharper, but darker.

## 3 Cameras and lenses

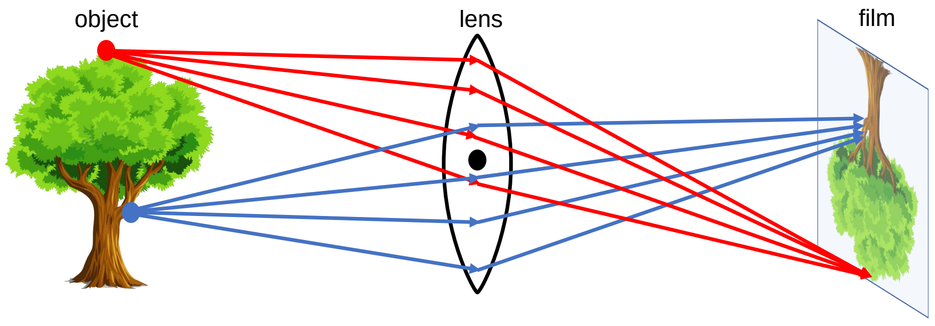

Figure 4: A setup of a simple lens model. Notice how the rays of the top point on the tree converge nicely on the film. However, a point at a different distance away from the lens results in rays not converging perfectly on the film.

In modern cameras, the above conflict between crispness and brightness is mitigated by using *lenses*, devices that can focus or disperse light. If we replace the pinhole with a lens that is both properly placed and sized, then it satisfies the following property: all rays of light that are emitted by some point $\mathbf{P}$ are refracted by the lens such that they converge to a single point $\mathbf{P}'$ in the image plane. Therefore, the problem of the majority of the light rays blocked due to a small aperture is removed (Figure 4). However, please note that this property does not hold for all 3D points, but only for some specific point $\mathbf{P}$. Take another point $\mathbf{Q}$ which is closer or further from the image plane than $\mathbf{P}$. The corresponding projection into the image will be blurred or out of focus. Thus, lenses have a specific distance for which objects are "in focus". This property is also related to a photography and computer graphics concept known as depth of field, which is the effective range at which cameras can take clear photos.

Camera lenses have another interesting property: they focus all light rays traveling

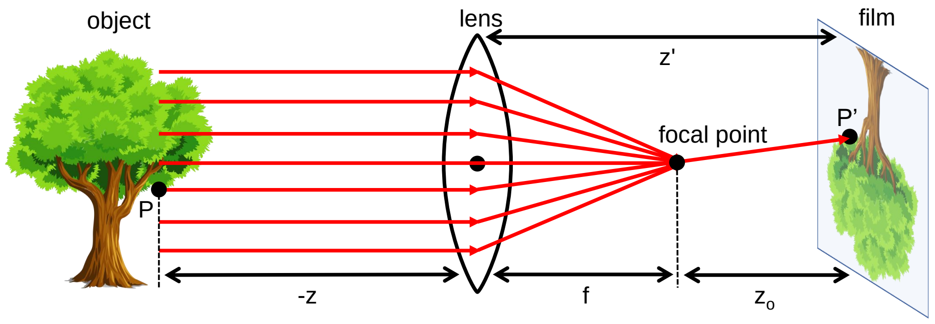

Figure 5: Lenses focus light rays parallel to the optical axis into the focal point. Furthermore, this setup illustrates the paraxial refraction model, which helps us find the relationship between points in the image plane and the 3D world in cameras with lenses.

parallel to the optical axis to one point known as the *focal point* (Figure 5). The distance between the focal point and the center of the lens is commonly referred to as the *focal length* $f$. Furthermore, light rays passing through the center of the lens are not deviated. We thus can arrive at a similar construction to the pinhole model that relates a point $\mathbf{P}$ in 3D space with its corresponding point $\mathbf{P}'$ in the image plane.

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} z'\frac{x}{z} \\ z'\frac{y}{z} \end{bmatrix} \tag{2}$$

The derivation for this model is outside the scope of the class. However, please notice that in the pinhole model $z' = f$, while in this lens-based model, $z' = f + z_0$. Additionally, since this derivation takes advantage of the paraxial or "thin lens" assumption[2], it is called the *paraxial refraction model*.

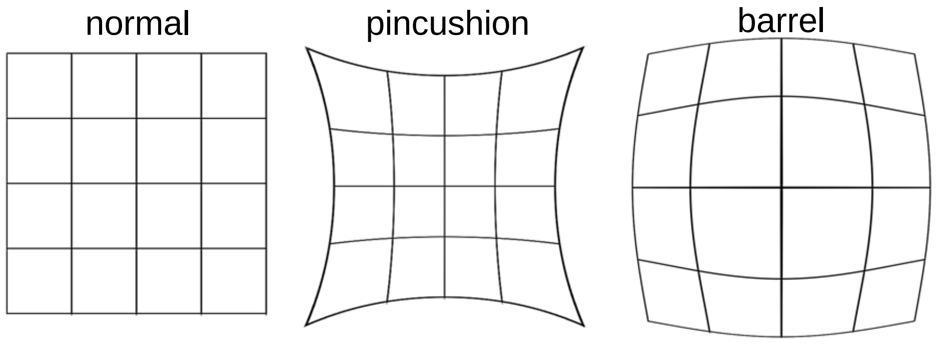

Figure 6: Demonstrating how pincushion and barrel distortions affect images.

Because the paraxial refraction model approximates using the thin lens assumption, a number of aberrations can occur. The most common one is referred to as *radial distortion*, which causes the image magnification to decrease or increase as a function of the distance to the optical axis. We classify the radial distortion as *pincushion distortion* when the magnification increases and *barrel distortion*[3] when the magnification decreases. Radial

[2]For the angle $\theta$ that incoming light rays make with the optical axis of the lens, the paraxial assumption substitutes $\theta$ for any place $\sin(\theta)$ is used. This approximation of $\theta$ for $\sin\theta$ holds as $\theta$ approaches 0.

[3]Barrel distortion typically occurs when one uses fish-eye lenses.

distortion is caused by the fact that different portions of the lens have differing focal lengths.

## 4 Going to digital image space

In this section, we will discuss the details of the parameters we must account for when modeling the projection from 3D space to the digital images we know. All the results derived will use the pinhole model, but they also hold for the paraxial refraction model.

As discussed earlier, a point $\mathbf{P}$ in 3D space can be mapped (or projected) into a 2D point $\mathbf{P}'$ in the image plane $\Pi'$. This $\mathbb{R}^3 \to \mathbb{R}^2$ mapping is referred to as a *projective transformation*. This projection of 3D points into the image plane does not directly correspond to what we see in actual digital images for several reasons. First, points in the digital images are, in general, in a different reference system than those in the image plane. Second, digital images are divided into discrete pixels, whereas points in the image plane are continuous. Finally, the physical sensors can introduce non-linearity such as distortion to the mapping. To account for these differences, we will introduce a number of additional transformations that allow us to map any point from the 3D world to pixel coordinates.

### 4.1 The Camera Matrix Model and Homogeneous Coordinates

#### 4.1.1 Introduction to the Camera Matrix Model

The camera matrix model describes a set of important parameters that affect how a world point $\mathbf{P}$ is mapped to image coordinates $\mathbf{P}'$. As the name suggests, these parameters will be represented in matrix form. First, let's introduce some of those parameters.

The first parameters, $c_x$ and $c_y$, describe how image plane and digital image coordinates can differ by a translation. Image plane coordinates have their origin $\mathbf{C}'$ at the image center where the $\mathbf{k}$ axis intersects the image plane. On the other hand, digital image coordinates typically have their origin at the lower-left corner of the image. Thus, 2D points in the image plane and 2D points in the image are offset by a translation vector $\begin{bmatrix} c_x, c_y \end{bmatrix}^T$. To accommodate this change of coordinate systems, the mapping now becomes:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} + c_x \\ f\frac{y}{z} + c_y \end{bmatrix} \tag{3}$$

The second effect we must account for that the points in digital images are expressed in pixels, while points in image plane are represented in physical measurements (e.g., millimeters). In order to accommodate this change of units, we must introduce two new parameters $k$ and $l$. These parameters, whose units would be something like $\frac{\text{pixels}}{\text{mm}}$, correspond to the change of units in the two axes of the image plane. Note that $k$ and $l$ may be different because the aspect ratio of a pixel is not guaranteed to be one. If $k = l$, we often say that the camera has *square pixels*. We adjust our previous mapping to be

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} fk\frac{x}{z} + c_x \\ fl\frac{y}{z} + c_y \end{bmatrix} = \begin{bmatrix} \alpha\frac{x}{z} + c_x \\ \beta\frac{y}{z} + c_y \end{bmatrix} \tag{4}$$

Is there a better way to represent this projection from $\mathbf{P} \to \mathbf{P}'$? If this projection is a linear transformation, then it can be represented as a product of a matrix and the input vector. However, from Equation 4, we see that this projection $\mathbf{P} \to \mathbf{P}'$ is not linear, as the operation divides one of the input parameters (namely $z$). Still, representing this projection as a matrix-vector product would be useful for future derivations. Therefore, can we represent our transformation as a matrix-vector product despite its nonlinearity? Homogeneous coordinates are the solution.

### 4.1.2 Homogeneous Coordinates

One way to solve this problem is to change the coordinate systems. For example, we introduce a new coordinate, such that any point $\mathbf{P}' = (x', y')$ becomes $(x', y', 1)$. Similarly, any point $\mathbf{P} = (x, y, z)$ becomes $(x, y, z, 1)$. This augmented space is referred to as the *homogeneous coordinate system.* As demonstrated previously, to convert a Euclidean vector $(v_1, ..., v_n)$ to homogeneous coordinates, we simply append a 1 in a new dimension to get $(v_1, ..., v_n, 1)$. Note that the equality between a vector and its homogeneous coordinates only occurs when the final coordinate equals one. Therefore, when converting back from arbitrary homogeneous coordinates $(v_1, ..., v_n, w)$, we get Euclidean coordinates $(\frac{v_1}{w}, ..., \frac{v_n}{w})$. Using homogeneous coordinates, we can formulate

$$\mathbf{P}'_h = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{P}_h \tag{5}$$

From this point on, assume that we will work in homogeneous coordinates, unless stated otherwise. We will drop the $h$ index, so any point $\mathbf{P}$ or $\mathbf{P}'$ can be assumed to be in homogeneous coordinates. As seen from Equation 5, we can represent the relationship between a point in 3D space and its image coordinates by a matrix vector relationship:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{P} = M\mathbf{P} \tag{6}$$

We can decompose this transformation a bit further into

$$\mathbf{P}' = M\mathbf{P} = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \mathbf{P} = K \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \mathbf{P} \tag{7}$$

The matrix $K$ is often referred to as the *camera matrix.*

### 4.1.3 The Complete Camera Matrix Model

The camera matrix $K$ contains some of the critical parameters that describes a camera's characteristics and its model, including the $c_x, c_y, k$, and $l$ parameters as discussed above. Two parameters are currently missing this formulation: *skewness* and *distortion.* We often say that an image is skewed when the camera coordinate system is skewed, meaning that the angle between the two axes is slightly larger or smaller than 90 degrees. Most cameras have zero-skew, but some degree of skewness may occur because of sensor manufacturing errors. Deriving the new camera matrix accounting for skewness is outside the scope of this class and we give it to you below:

$$K = \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Most methods that we introduce in this class ignore distortion effects, therefore our class camera matrix $K$ has 5 degrees of freedom: 2 for focal length, 2 for offset, and 1 for skewness. These parameters are collectively known as the *intrinsic parameters*, as they are unique and inherent to a given camera and relate to essential properties of the camera, such as its manufacturing.

## 4.2 Extrinsic Parameters

So far, we have described a mapping between a point $\mathbf{P}$ in the 3D camera reference system to a point $\mathbf{P}'$ in the 2D image plane using the intrinsic parameters of a camera described in matrix form. But what if the information about the 3D world is available in a different coordinate system? Then, we need to include an additional transformation that relates points from the world reference system to the camera reference system. This transformation is captured by a rotation matrix $R$ and translation vector $\mathbf{t}$. Therefore, given a point in a world reference system $\mathbf{P}_w$, we can compute its camera coordinates as follows:

$$\mathbf{P} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{P}_w \tag{9}$$

Substituting this in equation (7) and simplifying gives

$$\mathbf{P}' = K \begin{bmatrix} R, \mathbf{t} \end{bmatrix} \mathbf{P}_w = M\mathbf{P}_w \tag{10}$$

These parameters $R$ and $\mathbf{t}$ are known as the *extrinsic parameters* because they are external to and do not depend on the camera.

This completes the mapping from a 3D point $\mathbf{P}$ in an arbitrary world reference system to the image plane. To reiterate, we see that the full projection matrix $M$ consists of the two types of parameters introduced above: *intrinsic* and *extrinsic* parameters. All parameters contained in the camera matrix $K$ are the intrinsic parameters, which change as the type of camera changes. The extrinsic paramters include the rotation and translation, which do not depend on the camera's build. Overall, we find that the $3 \times 4$ projection matrix $M$ has 11 degrees of freedom: 5 from the intrinsic camera matrix, 3 from extrinsic rotation, and 3 from extrinsic translation.

## 5 Appendix A: Rigid Transformations

The basic rigid transformations are rotation, translation, and scaling. This appendix will cover them for the 3D case, as they are common type in this class.

Rotating a point in 3D space can be represented by rotating around each of the three coordinate axes respectively. When rotating around the coordinate axes, common convention is to rotate in a counter-clockwise direction. One intuitive way to think of rotations is how much we rotate around each degree of freedom, which is often referred to as *Euler angles*. However, this methodology can result in what is known as *singularities*, or *gimbal lock*, in which certain configurations result in a loss of a degree of freedom for the rotation.

One way to prevent this is to use rotation matrices, which are a more general form of representing rotations. Rotation matrices are square, orthogonal matrices with determinant one. Given a rotation matrix $R$ and a vector $\mathbf{v}$, we can compute the resulting vector $\mathbf{v}'$ as

$$\mathbf{v}' = R\mathbf{v}$$

Since rotation matrices are a very general representation of matrices, we can represent a rotation $\alpha, \beta, \gamma$ around each of the respective axes as follows:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Due to the convention of matrix multiplication, the rotation achieved by first rotating around the z-axis, then y-axis, then x-axis is given by the matrix product $R_x R_y R_z$.

Translations, or displacements, are used to describe the movement in a certain direction. In 3D space, we define a translation vector $\mathbf{t}$ with 3 values: the displacements in each of the 3 axes, often denoted as $t_x, t_y, t_z$. Thus, given some point $\mathbf{P}$ which is translated to some other point $\mathbf{P}'$ by $\mathbf{t}$, we can write it as:

$$\mathbf{P}' = \mathbf{P} + \mathbf{t} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

In matrix form, translations can be written using homogeneous coordinates. If we construct a translation matrix as

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

then we see that $\mathbf{P}' = T\mathbf{P}$ is equivalent to $\mathbf{P}' = \mathbf{P} + \mathbf{t}$.

If we want to combine translation with our rotation matrix multiplication, we can again use homogeneous coordinates to our advantage. If we want to rotate a vector $\mathbf{v}$ by $R$ and then translate it by $\mathbf{t}$, we can write the resulting vector $\mathbf{v}'$ as:

$$\begin{bmatrix} \mathbf{v}' \\ 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$$

Finally, if we want to scale the vector in certain directions by some amount $S_x, S_y, S_z$, we can construct a scaling matrix

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}$$

Therefore, if we want to scale a vector, then rotate, then translate, our final transformation matrix would be:

$$T = \begin{bmatrix} RS & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

Note that all of these types of transformations would be examples of affine transformations. Recall that projective transformations occur when the final row of $T$ is not $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$.

## 6 Appendix B: Different Camera Models

We will now describe a simple model known as the *weak perspective model*. In the weak perspective model, points are first projected to the reference plane using orthogonal projection and then projected to the image plane using a projective transformation.

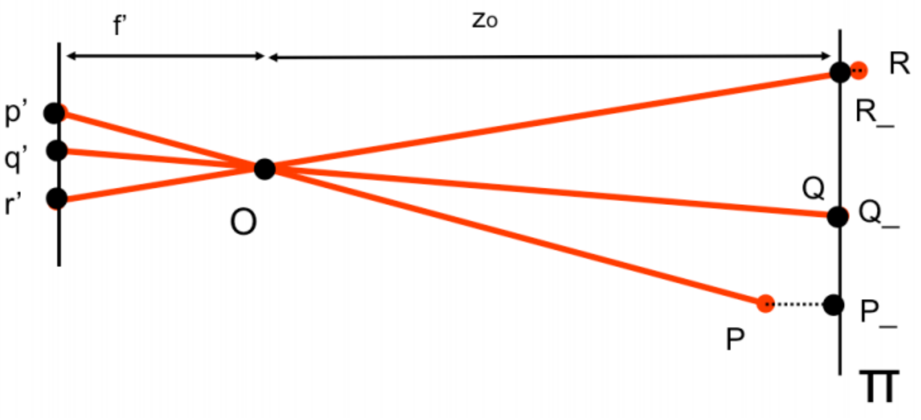

Figure 7: The weak perspective model: orthogonal projection onto reference plane

As Figure 7 shows, given a reference plane Π at a distance $z_o$ from the center of the camera, the points $\mathbf{P}$,$\mathbf{Q}$, and $\mathbf{R}$ are first projected to the plane Π using an orthogonal projection, generating points $\mathbf{P}_{-}$, $\mathbf{Q}_{-}$, and $\mathbf{R}_{-}$. This is a reasonable approximation when deviations in depth from the plane are small compared to the distance of the camera.

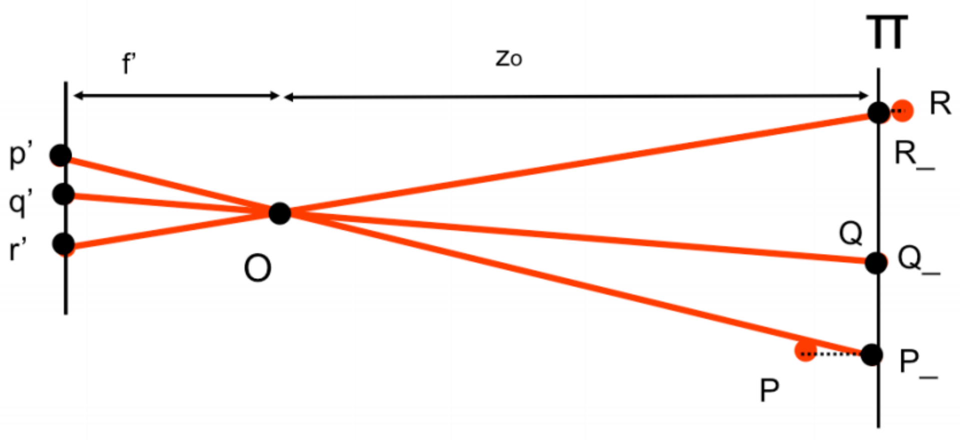

Figure 8: The weak perspective model: projection onto the image plane

Figure 8 illustrates how points $\mathbf{P}_{-}$, $\mathbf{Q}_{-}$, and $\mathbf{R}_{-}$ are then projected to the image plane using a regular projective transformation to produce the points $\mathbf{p}'$, $\mathbf{q}'$, and $\mathbf{r}'$. Notice, however, that because we have approximated the depth of each point to $z_o$ the projection has been reduced to a simple, constant magnification. The magnification is equal to the focal length $f'$ divided by $z_o$, leading to

$$x' = \frac{f'}{z_0}x \qquad y' = \frac{f'}{z_0}y$$

This model also simplifies the projection matrix

$$M = \begin{bmatrix} A & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}$$

As we see, the last row of $M$ is $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$ in the weak perspective model, compared to $\begin{bmatrix} \mathbf{v} & 1 \end{bmatrix}$ in the normal camera model. We do not prove this result and leave it to you as an exercise. The simplification is clearly demonstrated when mapping the 3D points to the image plane.

$$\mathbf{P}' = M\mathbf{P} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \mathbf{P} = \begin{bmatrix} \mathbf{m}_1^T\mathbf{P} \\ \mathbf{m}_2^T\mathbf{P} \\ 1 \end{bmatrix} \tag{11}$$

where $\mathbf{m}_1^T$, $\mathbf{m}_2^T$, and $\mathbf{m}_3^T$ denote the three rows of the projection matrix $M$. Thus, we see that the image plane point ultimately becomes a magnification of the original 3D point, irrespective of depth. The nonlinearity of the projective transformation disappears, making the weak perspective transformation a mere magnifier.

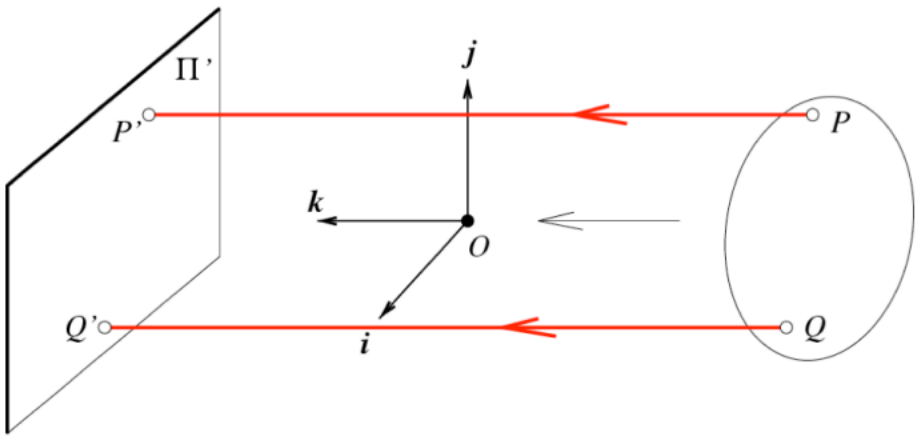

Figure 9: The orthographic projection model

Further simplification leads to the *orthographic (or affine) projection model.* In this case, the optical center is located at infinity. The projection rays are now perpendicular to the retinal plane. As a result, this model ignores depth altogether. Therefore,

$$x' = x$$
$$y' = y$$

Orthographic projection models are often used for architecture and industrial design.

Overall, weak perspective models result in much simpler math, at the cost of being somewhat imprecise. However, it often yields results that are very accurate when the object is small and distant from the camera.