Chapter 1

Perspective Geometry

When we move our bodies, heads, and eyes in space (often referred to as $\mathbb{R}^3$), the appearance of our environment changes in predictable ways. When one moves closer to an object it appears larger, turning one's head to the right moves an object to the left, and so on. These changes comprise the study of projective geometry. Quantifying the way that objects look as one moves through space is the first step in simulating the exploration of a visual environment.

1.1 Shifting Perspective

There are many different but equivalent ways to model perspective transformations; in the literature, no two researchers approach the problem in exactly the same way. My first approach is geometrically motivated rather than algebraic, but since this is a study of visual experience, this seems appropriate. The second algebraic derivation follows Blinn's Lastly, I'll compare the two results to check that there were no mishaps along the way.

Here is the problem. Imagine for a moment that you are an infinitesimal eyeball floating around in $\mathbb{R}^3$. At first, you are at (0,0,1) staring intently at the origin with some interest point I = (1,1,0) in your periphery. To you, this point appears to be just where it should be—at (1,1). But later, you decide to float up to the point (2,2,2) while continuing to look at the origin. Where does I appear to be now?

1.1.1 The Geometric Perspective

Let's solve the problem in general first, and then apply the resulting formulae to our question. The eye first views the point of interest I = (a,b,c) from the canonical initial eye position E_0 = (0,0,1) and then from a novel eye position E = (d,e,f). The goal is to find the horizontal and vertical offsets, call them u and v respectively, at which I appears in the new view, just like I appeared at (1,1) from E_0. Since we know how projection worked with the arrangement we had at E_0 (we projected points onto the xy-plane as per Fig. 1), let's try to replicate that situation at E. To that end, it will be quite convenient to define a new coordinate system that accomplishes this for us and then find u and v in these new coordinates. What will this entail? Well, we're going to need some new axes and a new origin.

Definition 1.1. A vector will be given by a point in $\mathbb{R}^3$ and has the properties of direction, equal to the direction from an origin (the implicit global origin $O = (0,0,0)$) to that point, and length, equal to the distance from an origin to that point.

Definition 1.2. An axis will be given by a vector and is defined to be the line in $\mathbb{R}^3$ containing that vector with positive direction equal to the vector's direction and length of a unit equal to the vector's length.

First, it will be easiest to find the new z axis; call it the z' axis. Since our eyeball used to look down the z axis at O, let's just assume (correctly) that the essential property of the z axis is that we're looking along it. Since the eye now looks down from E at O, it should be clear that the z' axis should just be E. Next, since we would like our projective plane to be perpendicular to the view direction, as is the case at E_0 since the z axis is perpendicular to the xy-plane, we must make both the x' and y' axes perpendicular to E. But this constraint alone doesn't give a unique set of axes—rather, it gives an entire family of coordinate systems, since the $x'y'$-plane is free to rotate, as shown in Fig. 1.1.

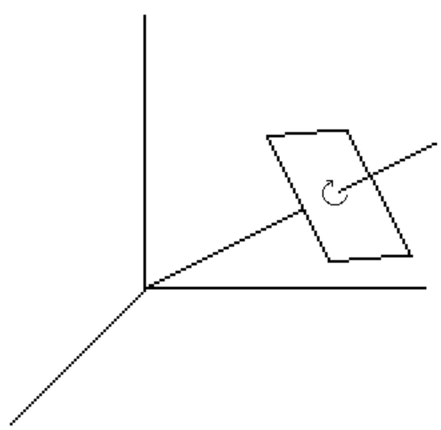

Figure 1.1: Spinning the $x'y'$-plane about the z' axis.

In order to define the axes uniquely, one more constraint must be provided. In computer graphics, this is usually done with the y' axis - we need to determine what direction we'd like to be 'up' in the projection. Often times, when one is projecting a whole image rather than a point, this is determined in some aesthetic way by the image's content. In this case, however, we can just define the y' axis to be the projection of the original (global) y-axis into the $x'y'$-plane. This will actually work for the photos later on as well, since they were all taken without any roll - i.e. with the x'-axis parallel to the ground (or xz-plane).

Given these constraints, the x' and y' axes can be calculated with just two cross products. Recall that the cross product of two vectors U and V yields a new vector that is perpendicular to the plane containing U and V and has length equal to the area of the parallelogram created by U and V. To be precise,

Definition 1.3. For $U, V \in \mathbb{R}^3$, $U \times V = (U_2V_3 - U_3V_2, U_3V_1 - U_1V_3, U_1V_2 - U_2V_1)$.

The key trick in determining these axes is that, by our construction, the y' axis will be coplanar with the z' axis *and* the y axis for any z' axis. Because each of the new axes must be pairwise perpendicular, we know that the x' axis must be perpendicular to the $y'z'$-plane, but since this is the same as the yz'-plane by our trick, the x' axis is just the cross product of the y axis and the z' axis. And now that we know the x' axis and the z' axis, the y' axis must be the cross product of these. Now the lengths of our axes will not necessarily be normalized (equal to 1) since $|E|$ is not necessarily 1, but not to worry - we will have to normalize everything later anyhow. And of course we will be keeping the right hand rule in mind since the cross product is not commutative. After these considerations, we arrive at the following formulae for our new axes:

$$\begin{aligned} x' &= y \times E \\ y' &= x' \times E \\ z' &= E. \end{aligned}$$

Next, lets find the origin O' of this new coordinate system. Keeping in mind that we're duplicating the situation at E_0, O' should be the intersection of the line containing E with the plane that is perpendicular to E and contains I. Recall that

Definition 1.4. Given a point $P \in \mathbb{R}^3$ and a direction vector $V \in \mathbb{R}^3$, a line $\mathcal{L} \subset \mathbb{R}^3$ along V that contains P may be expressed as $\mathcal{L} = \{P + tV \mid t \in \mathbb{R}\}$.

Definition 1.5. Given a point $P \in \mathbb{R}^3$ and a direction vector $V \in \mathbb{R}^3$, a plane $\mathcal{P} \subset \mathbb{R}^3$ perpendicular to V that contains P may be expressed as $\mathcal{P} = \{T \in \mathbb{R}^3 \mid V \cdot T - V \cdot P = 0\}$ where $\cdot$, the dot product, is defined as

Definition 1.6. For $U, V \in \mathbb{R}^n, U \cdot V = \sum_{i=1}^{n} U_iV_i$.

Then the intersection can be calculated by substituting the equation of the line into the equation of the plane,

$$E \cdot (E - tE) - E \cdot I = 0$$

solving for t,

$$t = 1 - \frac{E \cdot I}{E \cdot E}$$

and substituting t back into the line equation

$$\begin{aligned} O' &= E - E\left(1 - \frac{E \cdot I}{E \cdot E}\right) \\ &= E\left(\frac{E \cdot I}{E \cdot E}\right). \end{aligned}$$

Now we are in position to find u and v. After making O' the new origin, (by translating the only point relevant to the calculation, I, by $-O'$) u and v are just the components of $I - O'$ along the x' and y' axes, which can be calculated by

$$u = \frac{x' \cdot (I - O')}{|x'|} \quad \text{and} \quad v = \frac{y' \cdot (I - O')}{|y'|}.$$

Substituting in equations 1.1 and 1.2 and simplifying yields the final result

$$u = \frac{af - cd}{\sqrt{d^2 + f^2}} \quad \text{and} \quad v = \frac{b(E \cdot E) - e(E \cdot I)}{\sqrt{(d^2 + f^2)(E \cdot E)}}.$$

Well, almost the final result. We haven't quite replicated the situation at E_0 since there the distance from the eye to the plane containing I was exactly 1 - here, that distance is $E - O'$. Therefore, the very last thing we must do is to project I onto the plane where that distance is 1. This may sound pretty involved, but actually, thanks to the property of similar triangles in Fig. 1, this just comes to dividing u and v by $|E - O'|$, as is usually done when projecting points in this manner. This gives us

$$u = \frac{af - cd}{|E - O'|\sqrt{d^2 + f^2}} \quad \text{and} \quad v = \frac{b(E \cdot E) - e(E \cdot I)}{|E - O'|\sqrt{(d^2 + f^2)(E \cdot E)}}$$

and after simplifying,

$$u = \frac{(af - cd)\sqrt{E \cdot E}}{|E \cdot E - E \cdot I|\sqrt{d^2 + f^2}} \quad \text{and} \quad v = \frac{b(E \cdot E) - e(E \cdot I)}{|E \cdot E - E \cdot I|\sqrt{d^2 + f^2}}.$$

Now, we can answer our original question: where does (1,1,0) when viewed from (2,2,2)? Using the formulae above, the newly projected (1,1) appears at $\left(\frac{\sqrt{6}}{8}, \frac{\sqrt{2}}{8}\right)$. Hooray!

1.1.2 The Algebraic Perspective

For this next derivation to unfold smoothly, we must begin with a few standard proofs from linear algebra that we will need later on. One of the most basic motivations of the subject is that linear transformations (which are an essential part of perspective transformations) may be represented as matrix multiplications. It will be helpful to bear this in mind as we prove the following results for matrices; these properties are meant to be understood for the corresponding linear transformations as well. This

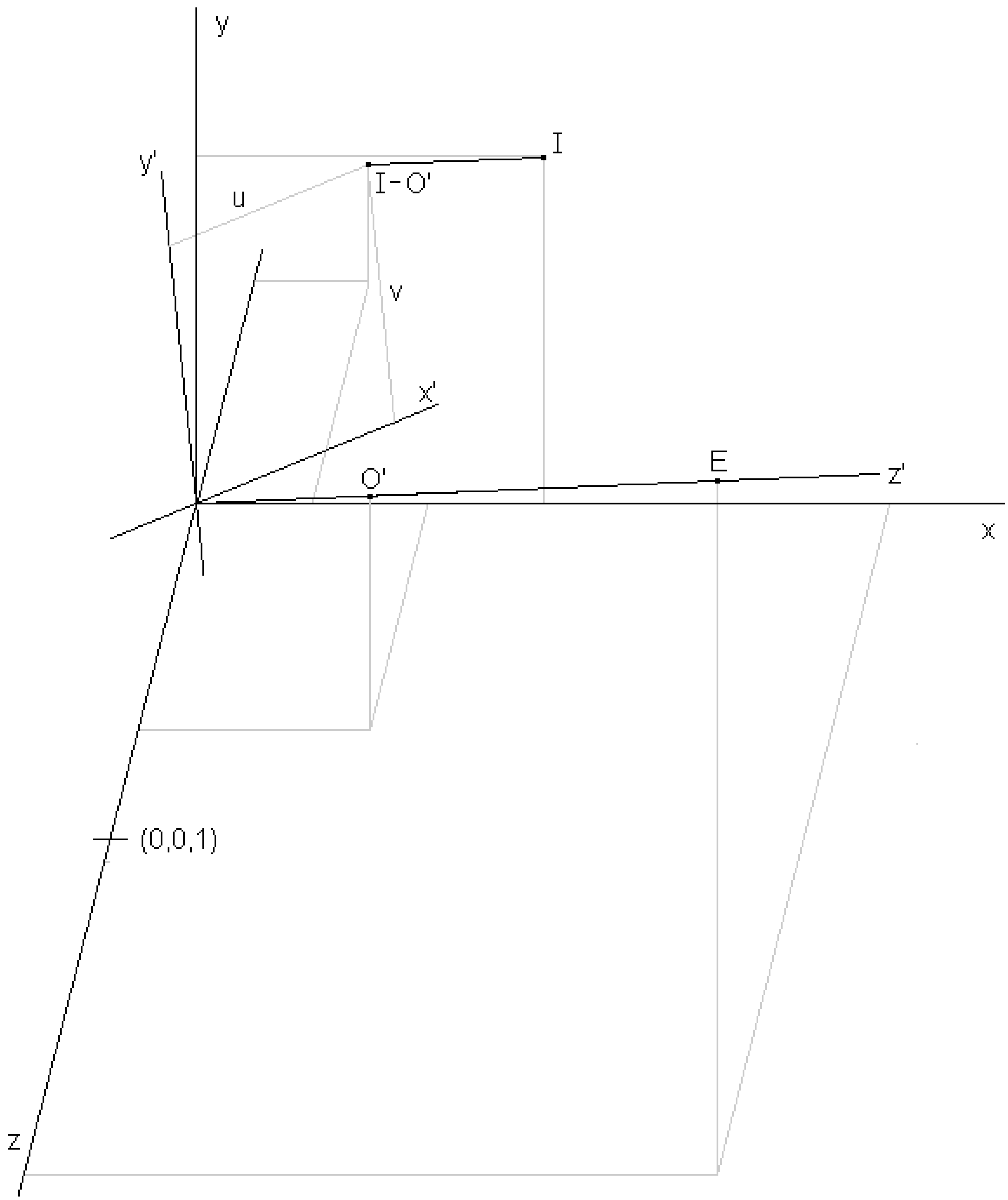

Figure 1.2: Calculating the transformation of the eye from (0,0,1) to (2,2,2) with interest point (1,1,0).

correspondence will be straightforward since the matrices we will examine, orthonormal and rotational matrices, are simply a repositioning of the standard coordinate axes in $\mathbb{R}^3$ without any deformation or change in scale. But first, a quick reminder of some basic algebraic operations that we will employ:

Transpose The transpose of the matrix, $\mathbf{M}^T$, is obtained by switching $\mathbf{M}$'s rows and columns (or flipping $\mathbf{M}$ over its diagonal) in the following way: [1]

$$\text{If } \mathbf{M} = \left[\begin{array}{c|c|c|c} V_1 & V_2 & \cdots & V_n \end{array}\right] \text{ then } \mathbf{M}^T = \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_n^T \end{bmatrix}.$$

An essential property of the transpose is the way that it interacts with matrix multiplication: $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$. Which brings us to...

Multiplication Matrices are multiplied by taking dot products of the rows of the first matrix with the respective columns of the second matrix. A relevant example of this process yields the useful identity $V^TV = V \cdot V$ for any vector V. And speaking of identities...

Identity The identity matrix, $\mathbf{I}$, has 1's down its diagonal and 0's everywhere else.

Inverse Lastly, the inverse of a matrix, $\mathbf{M}^{-1}$, is uniquely defined by the property $\mathbf{MM}^{-1} = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$. Only the identity matrix is its own inverse. Fun!

OK, basics out of the way... and we're off!

Orthonormal Matrices

Orthonormal matrices are orthogonal, meaning that each column vector is pairwise perpendicular to every other column vector, and normal, meaning that every column vector has length 1. Recall that the dot product of two vectors is 0 iff those vectors are perpendicular and that the dot product of a vector with itself is equal to its length squared. Therefore, more formally,

Definition 1.7. An orthonormal matrix is an $n \times n$ matrix $\mathbf{Q} = \left[\begin{array}{c|c|c|c} V_1 & V_2 & \cdots & V_n \end{array}\right]$ for which $V_i \cdot V_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$.

With this definition, we can prove some of the useful properties of these orthonormal matrices: that the transpose of an orthonormal matrix is its inverse, that the product of two orthonormal matrices is orthonormal, and lastly, that the dot product is invariant over orthonormal transformations.

[1] In the previous section, vectors were expressed horizontally; for instance $I = $ (a,b,c). In this section and for the remainder of the text, vectors will be vertical $n \times 1$ matrices such as $V = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$.

Theorem 1.8. *The transpose of an orthonormal matrix is its inverse.*

Proof.

$$\mathbf{Q}^T\mathbf{Q} = \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_n^T \end{bmatrix} \begin{bmatrix} V_1 \,\big|\, V_2 \,\big|\, \cdots \,\big|\, V_n \end{bmatrix} = \begin{bmatrix} V_1 \cdot V_1 & V_1 \cdot V_2 & \cdots & V_1 \cdot V_n \\ V_2 \cdot V_1 & V_2 \cdot V_2 & \cdots & V_2 \cdot V_n \\ \vdots & \vdots & \ddots & \vdots \\ V_n \cdot V_1 & V_n \cdot V_2 & \cdots & V_n \cdot V_n \end{bmatrix}$$

is both equivalent to $\mathbf{QQ}^T$ by the commutativity of the dot product, and equal to $\mathbf{I}$ by the definition of an orthonormal matrix. □

Furthermore, the converse of this theorem should be clear as well—if $\mathbf{Q}^T\mathbf{Q} = \mathbf{QQ}^T = \mathbf{I}$ then the columns of $\mathbf{Q}$ will be constrained to the precise relationships outlined in Definition 1.5 by the equivalence of the resulting matrices of dot products to the identity. Therefore, $\mathbf{Q}$ must be orthonormal.

Theorem 1.9. *The product of two orthonormal matrices is orthonormal.*

Proof. Let $\mathbf{Q}$ and $\mathbf{R}$ be orthonormal matrices. Then

$$\begin{gathered} \mathbf{QR}\left(\mathbf{QR}^T\right) = \mathbf{QR}^T\mathbf{RQ}^T = \mathbf{QQ}^T = \mathbf{I} \quad \text{and} \\ (\mathbf{QR})^T\mathbf{QR} = \mathbf{R}^T\mathbf{Q}^T\mathbf{QR} = \mathbf{R}^T\mathbf{R} = \mathbf{I}, \end{gathered}$$

so by the analysis just above, $\mathbf{QR}$ must be orthonormal as well. □

Theorem 1.10. *The dot product is invariant under orthonormal transformations.*

Proof. Let $\mathbf{Q}$ be an orthonormal matrix and U, U', V, V' be vectors such that

$$\mathbf{Q}U = U' \quad \text{and} \tag{1}$$

$$\mathbf{Q}V = V'. \tag{2}$$

By (1),

$$(\mathbf{Q}U)^T = U'^T \quad \text{yields} \quad \mathrm{U}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}} = \mathrm{U}'^{\mathrm{T}}, \tag{3}$$

and then by (2) and (3),

$$\begin{aligned} U^T\mathbf{Q}^T\mathbf{Q}V &= U'^TV' && \text{multiplying equals by equals} \\ U^TV &= U'^TV' && \text{canceling orthonormal matrices} \\ U \cdot V &= U' \cdot V'. && \text{by definition of dot product} \end{aligned}$$

□

Rotation Matrices

Since we would like our perspective transformation to be general, enabling us to view $\mathbb{R}^3$ in every possible way, we must consider perspectives from all points and in all directions. To accomplish this, the only constraint that we will impose on our transformations is that they be distance preserving or 'rigid' - in other words, that they shift but do not stretch or deform the space in any way. A basic proof from geometry establishes that every rigid transformation in $\mathbb{R}^3$ is either a rotation, a translation, or both. We'll discuss translations shortly, but first, let's explore rotations.

Definition 1.11. A rotation is a rigid transformation with a fixed point at the origin.

This definition might seem a bit obtuse for a concept so simple as rotation, but it will inform the properties of rotation which we will need. First though, let's be sure that the definition is consistent. So a rigid transformation can either be a rotation or a translation, and if a point doesn't move after the transformation, (i.e. a fixed point exists) then the transformation cannot be a translation, since translations move all points. Then it can only be a rotation. Phew.

It would also be nice if our definition of rotation described the same phenomenon that we usually refer to when we speak about rotation. This can be confirmed in the following way. Spread out the fingers on your left hand and place your right pointer finger in the middle of your left palm. Then, while keeping your right finger in the same spot on your left hand (and in space!), move your left hand about. The point of contact between your right finger and left hand is the fixed point, and the fingertips of your left hand maintain the same distance relative to each other, so this example satisfies the definition. What's more, you will find that you can rotate your left hand in any way (roll, picth, yaw, or a combination) while satisfying the constraints but cannot translate your hand, just as required!

Theorem 1.12. *Rotations are orthonormal.*

Proof. We know that rotations preserve distance between any two points, but because the origin is fixed, they also preserve length (the distance from a point to the origin). For an arbitrary rotation $\mathbf{R}$ and two arbitrary vectors U and V, these facts can be formalized as [1]

$$|\mathbf{R}U - \mathbf{R}V| = |U - V| \tag{1}$$

and

$$|\mathbf{R}U| = |U|. \tag{2}$$

Now, for the fun part...

[1] Traditionally, (in linear algebra) the length of a vector is denoted $||V||$ rather than $|V|$ and the dot product (usually the more general inner product) denoted $< U, V >$ rather than $U \cdot V$, but for continuity and ease of readership, I've chosen to keep the notation consistent.

$$
\begin{aligned}
|\mathbf{R}U - \mathbf{R}V|^2 &= |U - V|^2 && \text{by (1)} \\
|\mathbf{R}U|^2 - 2(\mathbf{R}U \cdot \mathbf{R}V) + |\mathbf{R}V|^2 &= |U|^2 - 2(U \cdot V) + |V|^2 && \text{by def of } |\,| \text{ and } \cdot \\
-2(\mathbf{R}U \cdot \mathbf{R}V) &= -2(U \cdot V) && \text{by (2)} \\
\mathbf{R}U \cdot \mathbf{R}V &= U \cdot V && \text{by simplifying} \\
\mathbf{R}^T\mathbf{R}U \cdot V &= U \cdot V && \text{by def of } \cdot \text{ and } ^T \\
\left(\mathbf{R}^T\mathbf{R}U - U\right) \cdot V &= 0 && \text{by additivity of } \cdot \\
\mathbf{R}^T\mathbf{R}U - U &= 0 && X \cdot V = 0 \text{ for all V, } X = \mathbf{0} \\
\mathbf{R}^T\mathbf{R} - \mathbf{I} &= 0 && \text{previous line holds for all } U \\
\mathbf{R}^T\mathbf{R} &= \mathbf{I} && \text{by rearranging}
\end{aligned}
$$

and thus rotations are orthonormal since their transpose is equal to their inverse. □

Now that we know that rotations are orthonormal, we can apply the previous theorems about orthonormal matrices to them. We'll be doing just that in the derivation.

Deriving the Transformation

As a refresher, let's quickly recall the motivation for the geometric derivation, since this derivation will be analogous. An infinitesimal eyeball with a canonical view and an interest point in its periphery floated up to some novel view while continuously looking at the origin; we were curious how the interest point looked from up there. To figure this out, we defined a new coordinate system that mimicked the canonical situation for the new view, calculated the position of the interest point in this new coordinate system, and finally projected the interest point onto a plane that was unit length away from the new eye. The method here will be analogous: we will calculate the transformation from the new view into the canonical view, apply this transformation to the interest point, and then project the interest point in the same way as before. However, to make the algebra work out nicely, our canonical view here will have to be a bit different: in this case, the canonical eye E_0 will sit at the origin, while the projective plane will reside at $z = 1$, rather than the other way around. This is depicted in Fig. 1.3.

Now, to find our transformation. We're still in the game of trying to recreate the canonical situation from our novel situation, but with what sort of transformation would this be accomplished? First of all, (since it's easy) we should put the eye at the origin, recreating the canonical situation as usual. For this, all we have to do is to translate all of the points relevant to our computation by $-E$. Next, another transformation should align the new view direction with the canonical one. Well, since we already went through the trouble of putting the eye at the origin, let's have this transformation fix the origin. Furthermore, since changing views doesn't change the true distance between any points in $\mathbb{R}^3$, it should be clear that the type of transformation that we need is indeed a rotation. Then this rotation $\mathbf{R}$ must

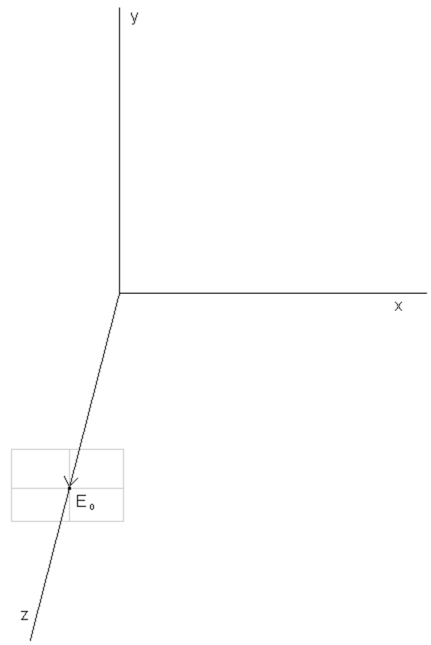

Figure 1.3: Blinn's canonical perspective view.

first align our (now normalized) view direction, call it $\hat{V}$ [1], with the canonical view direction, the z-axis. Furthermore, it must also align some global 'up' vector, call it $\hat{U}$, with the canonical 'up', which we will define to be a vector with no x-component. Formally, these relationships can be expressed as

$$\mathbf{R}\hat{V} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{R}\hat{\mathrm{U}} = \begin{bmatrix} 0 \\ g \\ h \end{bmatrix} \text{respectively,}$$

but, as is usually the case, a picture is worth a thousand words—Fig. 1.4 should help to illuminate what **R** is doing for us.

And here's where our proofs pay off—we can actually find **R** with just these two equations. Since we know that **R** is a rotation, we also know that it is orthonormal. Therefore, it follows from Thm. 1.10 that

$$\hat{V} \cdot \hat{U} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ g \\ h \end{bmatrix} = h.$$

And since $\hat{U}$ is unit length by definition, it must be the case that $g = \sqrt{1 - h^2}$. So we've got g and h. Now rearranging by Thm. 1.8, we obtain

$$\hat{V} = \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathrm{U}} = \mathbf{R}^{\mathrm{T}} \begin{bmatrix} 0 \\ g \\ h \end{bmatrix},$$

[1] In general, $\hat{X}$ will denote a normal (unit length) vector with direction X. In this case, $\hat{X} = -\frac{E}{|E|}$.

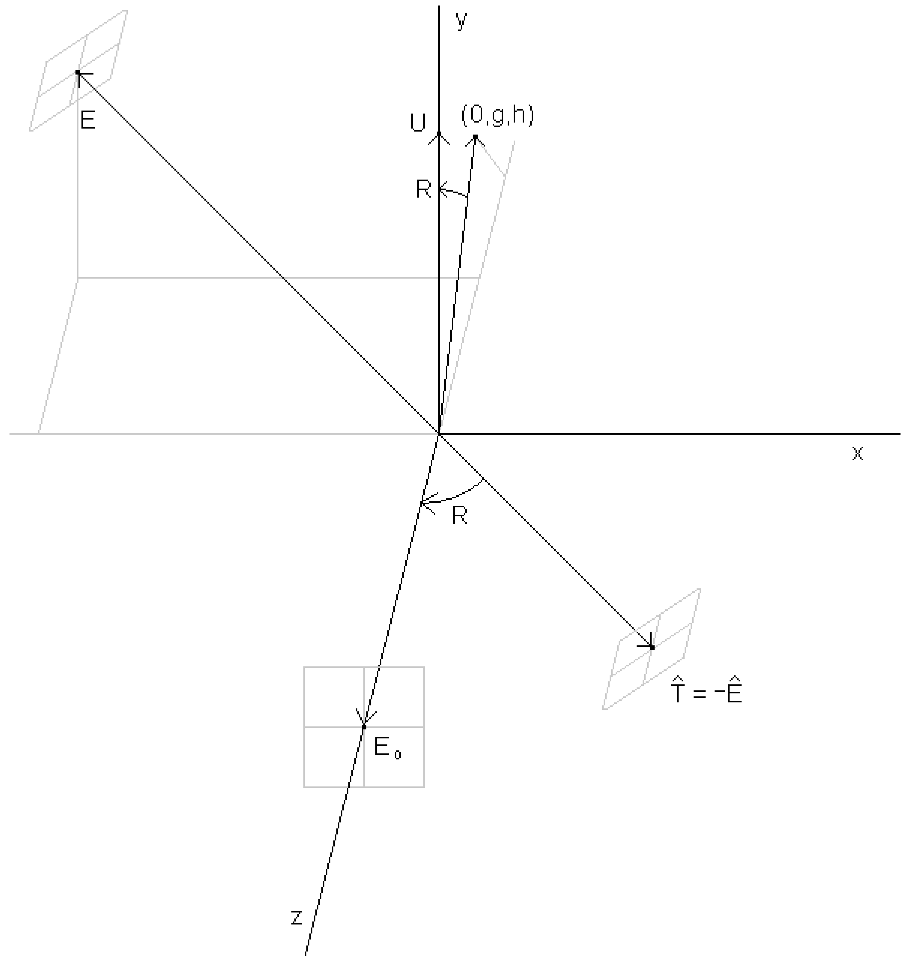

Figure 1.4: **R** acting on vectors $\hat{T}$ and $\hat{U}$.

and solving for $\mathbf{R}_3^T$ and $\mathbf{R}_2^T$, we obtain,

$$R_3^T = \hat{V}$$
$$R_2^T = \frac{1}{g}\hat{U} - \frac{h}{g}\hat{V}.$$

Lastly, we can find the first column of R^T, R_1^T, simply by the definition of orthonormality: every column is perpendicular to every other column. Therefore, $R_1^T = R_2^T \times R_3^T$ and since $\hat{V} \times \hat{V} = 0$, we obtain

$$R_1^T = \frac{1}{g}\left(\hat{U} \times \hat{V}\right).$$

Thus, in summary, our rotation matrix will be

$$\mathbf{R} = \begin{bmatrix} \frac{1}{g}\left(\hat{U} \times \hat{V}\right)^T \\ \frac{1}{g}\left(\hat{U} - h\hat{V}\right)^T \\ \hat{V}^T \end{bmatrix}.$$

Checking Ourselves

As was previously mentioned, to transform a given point of interest, we'll just translate it by $-E$, multiply it on the left by $\mathbf{R}$ and then divide out the z coordinate to project it into the image plane. As in the geometric derivation, the points we'll be working with will be labeled and defined as

$$I = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad E = \begin{bmatrix} d \\ e \\ f \end{bmatrix}, \quad \hat{U} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and } \hat{\mathrm{V}} = -\frac{\mathrm{E}}{|\mathrm{E}|}.$$

It will be easiest to carry out this calculation in two steps. In the first step we'll translate and rotate I; we'll project it second. Translating and rotating I by $-E$ and $\mathbf{R}$ will yield another point in $\mathbb{R}^3$—the new but unprojected interest point. Let's find each of the three coordinates of this point separately to preserve as much clarity as possible. Though it is relatively unenlightening, the careful reader might wish to carry out the simplifications explicitly. For the first component we have

$$\begin{aligned}
\frac{1}{g}\left(\hat{U} \times \hat{V}\right) \cdot (I - E) &= \frac{1}{g}\left(\hat{U} \times \hat{V}\right) \cdot I \\
&= \frac{-\hat{U} \times E}{|\hat{U} \times E|} \cdot I \\
&= \left(\frac{1}{\sqrt{d^2 + f^2}}\right) \begin{bmatrix} -f \\ 0 \\ d \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \\
&= \frac{cd - af}{\sqrt{d^2 + f^2}}.
\end{aligned}$$

For the second component,

$$\begin{aligned}
\frac{1}{g}\left(\hat{U} - h\hat{V}\right)^T \cdot (I - E) &= \frac{1}{g}\left(\hat{U} - h\hat{V}\right)^T \cdot I \\
&= \frac{(E \cdot E)\hat{U} - (E \cdot \hat{U})E}{\sqrt{(d^2 + f^2)(E \cdot E)}} \cdot I \\
&= \left(\frac{1}{\sqrt{(d^2 + f^2)(E \cdot E)}}\right) \begin{bmatrix} -de \\ d^2 + f^2 \\ -ef \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \\
&= \frac{b(d^2 + f^2) - e(ad + cf)}{\sqrt{(d^2 + f^2)(E \cdot E)}} \\
&= \frac{b(E \cdot E) - e(E \cdot I)}{\sqrt{(d^2 + f^2)(E \cdot E)}}.
\end{aligned}$$

And for the third,

$$\hat{V} \cdot (I - E) = \frac{-E}{|E|} \cdot (I - E) = \frac{E \cdot E - E \cdot I}{\sqrt{E \cdot E}}.$$

Hopefully, this is beginning to look a bit familiar. These are, in fact, the same equations that we obtained for the coordinates of the unprojected interest point using the geometric method. Dividing by the third coordinate will project this point in the same way as the previous method, and we do end up with the exact same solution. We did it!

So now that our derivations check out, we can move on to exploring the relationships created by multiple views, rather than just one. But before we do, a quick and happy caveat. The setup for this latter method does have one advantage over the previous geometric one: it facilitates an important and straightforward generalization of our problem. Careful readers may have noticed that in both derivations, our infinitesimal eye always looked directly at the origin. Wouldn't it be nice if, in addition to being able to move anywhere in $\mathbb{R}^3$ and calculate the correct view, our eye could look in any direction too? Well, here's how it can. Rather than taking $\hat{V}$ to be just $-\frac{E}{|E|}$, we can also specify a 'non-original' view direction with $\hat{V}$, as shown in Fig. 1.5. As you can see, this modification of $\hat{V}$ alters our rotation matrix $\mathbf{R}$ in just the right way to account for looking in arbitrary directions—not just at the origin.

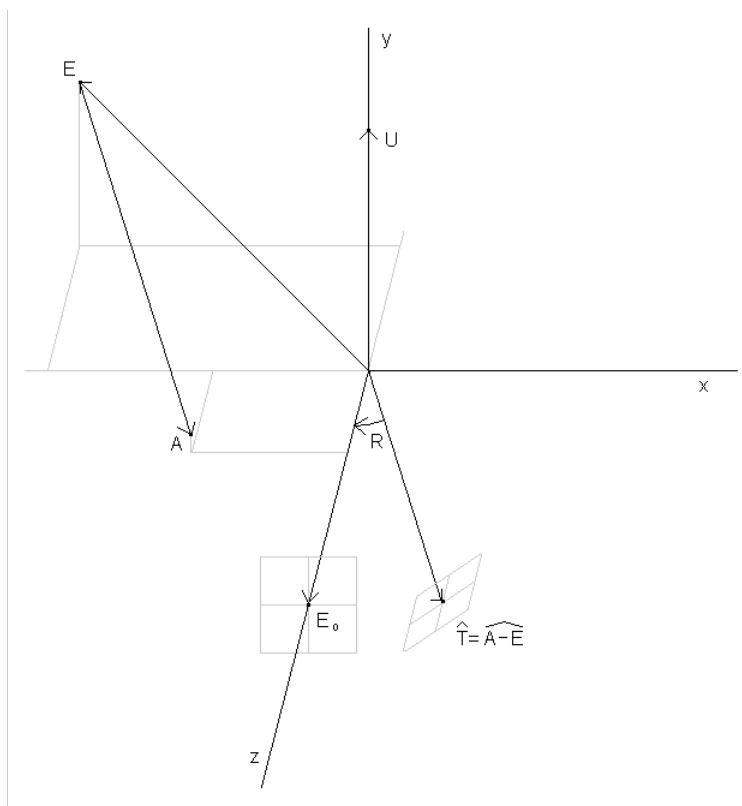

Figure 1.5: A modified $\mathbf{R}$ where the eye looks from E to A, and the view $\hat{V} = \frac{E-A}{|E-A|}$.

1.2 Relative Perspectives

In the previous section, we derived a method for calculating where an interest point appears, given some eyeball's location and view direction in $\mathbb{R}^3$. Essentially, we translated the point relative to the eye's location, rotated it to align it with the eye's view direction. Next, we will find a method to solve the reverse problem: given two different views of the same point in $\mathbb{R}^3$, we'll determine the transformation (i.e. the translation and rotation) between them. In order to accomplish this, we'll need to first review a bit more linear algebra, derive a few key equations from epipolar geometry, and use these equations to construct numerical algorithms to approximate our answer. Here, three algorithms will be considered: first, calculating the transformation between two views (known as the fundamental matrix) from known point pairs in distinct images, second, separating the implicit rotation and translation transformations embedded in the fundamental matrix, and lastly, determining the actual 3D locations of these interest points.

1.2.1 Eigenstuff, Symmetry, and the SVD

Much of numerical linear algebra is dedicated to 'decomposing' different types of matrices in different ways—many matrices have properties which allow them to be broken up into a product of other matrices that are easier to understand or manipulate. Perhaps the most basic example of this is the process of diagonalization. Given an $n \times n$ matrix $\mathbf{M}$ that meets certain conditions which we will review momentarily, we can write $\mathbf{M}$ as a product of an invertible matrix, a diagonal matrix, and the inverse of the first matrix: $\mathbf{M} = \mathbf{QDQ}^{-1}$. Before getting into the gritty details, recall the useful property that results from decomposing a matrix in this manner. Naive matrix multiplication takes $\Theta(n^3)$ time[1] time (or perhaps, one day, $\Theta(n^2)$ time to the most starry eyed researchers) for an $n \times n$ matrix. Therefore, if we needed to find the power p of a matrix the straightforward method would take $\Theta(pn^3)$ time. But if we could diagonalize that matrix before computing its power, the calculation could be simplified as follows:

$$\begin{aligned}
\mathbf{M}^p &= \left(\mathbf{QDQ}^{-1}\right)^p \\
&= \left(\mathbf{QDQ}^{-1}\right)\left(\mathbf{QDQ}^{-1}\right)\cdots\left(\mathbf{QDQ}^{-1}\right) \quad \text{(p times)} \\
&= \left(\mathbf{QD}\right)\left(\mathbf{Q}^{-1}\mathbf{Q}\right)\mathbf{D}\cdots\mathbf{D}\left(\mathbf{Q}^{-1}\mathbf{Q}\right)\left(\mathbf{DQ}^{-1}\right) \\
&= \left(\mathbf{QD}\right)\left(\mathbf{I}\right)\mathbf{D}\cdots\mathbf{D}\left(\mathbf{I}\right)\left(\mathbf{DQ}^{-1}\right) \\
&= \left(\mathbf{QD}^p\mathbf{Q}^{-1}\right).
\end{aligned}$$

Because taking the power of a diagonal matrix only requires $O(pn)$ time (since each diagonal entry only multiplies itself) and the multiplications by $\mathbf{Q}$ and $\mathbf{Q}^{-1}$ require only constant time, we've successfully progressed from a cubic time algorithm to a linear time algorithm—pretty impressive. But there's no such thing as a free lunch, and here comes the fine print.

[1]Please see section 2.4.2 for the definition of Θ- notation.

The Eigenstuff

Not every matrix is diagonalizable. There are two conditions that our matrix must meet for this decomposition to work—its characteristic polynomial must split and the dimension of each eigenspace must equal the multiplicity of its associated eigenvalue. Wow, that's a lot to unravel. Let's start with some definitions and then try to put them together.

Definition 1.13. A non-zero $n \times 1$ vector V is an eigenvector of an $n \times n$ matrix $\mathbf{M}$ whenever $\mathbf{M}V = \lambda V$ for some scalar $\lambda \in \mathbb{R}$. λ is called the eigenvalue of $\mathbf{M}$ associated with the eigenvector V.

Definition 1.14. The characteristic polynomial of a matrix $\mathbf{M}$ is given by the expression $\chi_{\mathbf{M}}(t) = \det(\mathbf{M} - \mathrm{t}\mathbf{I})$, where 'det' denotes the determinant of a matrix.

To begin to get a sense (or recollection) of these constructions, let's prove a quick but essential theorem.

Theorem 1.15. *An eigenvalue λ of a matrix $\mathbf{M}$ is a root of $\chi_{\mathbf{M}}$, i.e. $\chi_{\mathbf{M}}(\lambda) = 0$.*

Proof. Since λ is an eigenvalue of $\mathbf{M}$, there is some eigenvector V of $\mathbf{M}$ such that $\mathbf{M}V = \lambda V$. Then by the following derivation,

$$\begin{aligned} \mathbf{M}V &= \lambda V \\ \mathbf{M}V &= \lambda \mathbf{I} V \\ \mathbf{M}V - \lambda \mathbf{I} V &= \mathbf{0} \\ (\mathbf{M} - \lambda \mathbf{I})V &= \mathbf{0} \end{aligned}$$

$(\mathbf{M} - \lambda \mathbf{I})$ must not be invertible, or else we could multiply by its inverse on both sides of the final equation above, and V would be the trivial zero vector, in contradiction with the definition of an eigenvector. To ensure that $(\mathbf{M} - \lambda \mathbf{I})$ is not invertible, we require that its determinant be zero, giving us the constraint on λ that $\det(\mathbf{M} - \lambda \mathbf{I}) = 0$. And so, we must have $\chi_{\mathbf{M}}(\lambda) = 0$, as required. □

Now we are in a position to explain the first condition of diagonalizability. The notion of a polynomial 'splitting' is simply that it can be factored (split) into linear terms with real coefficients, which in turn means that the polynomial's roots are real numbers. For example, $x^2 - 1$ splits because it factors into $(x+1)(x-1)$, but $x^2 + 1$ doesn't split because there are no factorizations over the real numbers, and therefore its roots are complex.[1] Since our eigenvalues (and consequently our eigenvectors as well) will only be real numbers, we must be concerned that our characteristic polynomials split over the reals.

[1] If we allowed our eigenvalues to range over the complex numbers, then every characteristic polynomial would split, but then we would obtain complex solutions for some problems a bit further down the road for which we truly need real valued solutions.

Before we get to the second condition for diagonalizability, it will be quite helpful to prove one further theorem regarding the diagonal decomposition itself.

Theorem 1.16. *For the diagonalization of a matrix* $\mathbf{M} = \mathbf{QDQ}^{-1}$*, each column of* $\mathbf{Q}$*,* Q_i *for some index* i*, is an eigenvector of* $\mathbf{M}$*, and each diagonal element in* $\mathbf{D}$*,* d_{ii}*, is the eigenvalue associated with that eigenvector* Q_i*.*

Proof. Let all eigenvalue/eigenvector equations of the form $\mathbf{M}V = \lambda V$ be indexed to obtain $\mathbf{M}V_i = \lambda_i V_i$ where i ranges over all of the eigenvalue/eigenvector pairs for $\mathbf{M}$. By the structure of matrix multiplication, we can express all of these relationships at once by making the eigenvectors columns of a single matrix in the following way:

$$\begin{aligned}
\mathbf{M}\left[\begin{array}{c|c|c|c} V_1 & V_2 & \cdots & V_n \end{array}\right] &= \left[\begin{array}{c|c|c|c} \lambda_1 V_1 & \lambda_2 V_2 & \cdots & \lambda_n V_n \end{array}\right] \\
&= \left[\begin{array}{c|c|c|c} V_1 & V_2 & \cdots & V_n \end{array}\right] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.
\end{aligned}$$

After rewriting the above equation as matrices, $\mathbf{MQ} = \mathbf{QD}$, it should be clear that so long as we assume that $\mathbf{Q}$ is invertible, we are able to obtain the desired result, $\mathbf{M} = \mathbf{QDQ}^{-1}$, where $\mathbf{Q}$ is the column matrix of $\mathbf{M}$'s eigenvectors and $\mathbf{D}$ is the diagonal matrix of $\mathbf{M}$'s eigenvalues. □

And now for the second criterion of diagonalizability, that the dimension of the eigenspace be the multiplicity of the associated eigenvalue. Happily, one can see the necessity for this fact in the above theorem without really having to define bases, null spaces, span, or even eigenspaces at all. Basically, this criterion requires that if there is some number of equal eigenvalues in $\mathbf{D}$, that there be just as many distinct associated eigenvectors in $\mathbf{Q}$. Suppose for a moment that we were to use the same eigenvector in $\mathbf{Q}$ for some of the repeated eigenvalues in $\mathbf{D}$. Then $\mathbf{Q}$ would no longer be invertible (recall that invertible matrices may not have a column which can be obtained by a scalar multiplication of another column) and our theorem wouldn't work. Therefore, if we obtain repeated eigenvalues as roots of our characteristic polynomial, we must find a distinct eigenvector for each repetition of that eigenvalue.

Symmetry and Skew

In order to get more familiar with this notion of diagonalization, and because we're going to need the results later, let's diagonalize two types of matrices as examples (both of which will be denoted $\mathbf{S}$, for clarity). We'll look at real symmetric matrices, with the property that $\mathbf{S} = \mathbf{S}^T$, and real skew-symmetric matrices, for which $-\mathbf{S} = \mathbf{S}^T$, and show that both types are actually diagonalized by orthonormal matrices (which will be quite handy a bit later). Let's begin with the symmetric case. The usual route by which symmetric matrices are diagonalized is with another variety of decomposition known as Schur's Theorem, and here it is.

Theorem 1.17. *A square matrix with a characteristic polynomial that splits may be decomposed into a product of the matrices* $\mathbf{M} = \mathbf{R}\mathbf{U}\mathbf{R}^T$*, where* $\mathbf{R}$ *is orthonormal and* $\mathbf{U}$ *is upper triangular.*

Proof. First, recall that an upper triangular matrix is a square matrix with the property that every element below the diagonal must be zero. Ok, now we're going to have to proceed by induction, so hang on to your hats. As per usual, the base case is trivial. Let $\mathbf{M}$ be an arbitrary 1×1 matrix $[m]$. Then $[m] = [1][m][1]$ where $[1]$ is orthonormal and $[m]$ is technically upper triangular, so the theorem holds for all 1×1 matrices.

Now for the inductive case. Let $\mathbf{M}$ be an $n \times n$ matrix where $\chi_{\mathbf{M}}$ splits. This condition guarantees that there is at least one eigenvector of $\mathbf{M}$; calling it V, we know that it satisfies the equation $\mathbf{M}V = \lambda_V V$. Without loss of generality, assume that V is normal, and let $\mathbf{R}$ be an orthonormal matrix with V as its left-most column. Consider

$$\mathbf{R}^T\mathbf{M}\mathbf{R} = \left[\begin{array}{c|c} \lambda_V & \cdots \\ \hline 0 & \\ \vdots & \left[\mathbf{R}^T\mathbf{M}\mathbf{R}\right]_{n-1} \\ 0 & \end{array}\right].$$

First, let's show that $\chi_{[\mathbf{R}^T\mathbf{M}\mathbf{R}]_{n-1}}$ splits, and then we can use our inductive hypothesis.

$$\begin{aligned}
\chi_{\mathbf{M}} &= \det(\mathbf{M} - \lambda\mathbf{I}) & \\
&= \det\left(\mathbf{R}^T\right)\det(\mathbf{M} - \lambda\mathbf{I})\det(\mathbf{R}) & \det\left(\mathbf{R}^T\right) = \det\left(\mathbf{R}\right) = 1 \\
&= \det\left(\mathbf{R}^T(\mathbf{M} - \lambda\mathbf{I})\mathbf{R}\right) & \det\left(\mathbf{A}\mathbf{B}\right) = \det\left(\mathbf{A}\right)\det\left(\mathbf{B}\right) \\
&= \det\left(\mathbf{R}^T\mathbf{M}\mathbf{R} - \lambda\mathbf{I}\right) & \text{distributing and canceling} \\
&= (\lambda_V - \lambda)\det\left(\left[\mathbf{R}^T\mathbf{M}\mathbf{R}\right]_{n-1} - \lambda\mathbf{I}\right) & \text{expanding the determinant} \\
&= (\lambda_V - \lambda)\,\chi_{[\mathbf{R}^T\mathbf{M}\mathbf{R}]_{n-1}}. &
\end{aligned}$$

Hopefully it is clear that $\chi_{[\mathbf{R}^T\mathbf{M}\mathbf{R}]_{n-1}}$ splits whenever $\chi_{\mathbf{M}}$ splits, which was one of our assumptions. And now we can use our inductive hypothesis—that the theorem holds for all matrices of size $(n-1) \times (n-1)$. Then for some $(n-1) \times (n-1)$ orthonormal matrix $\mathbf{P}$ and upper triangular matrix $\mathbf{U}$, $\mathbf{P}^T\left[\mathbf{R}^T\mathbf{M}\mathbf{R}\right]_{n-1}\mathbf{P} = \mathbf{U}$. Define $\mathbf{P}^+$ to be the $n \times n$ matrix

$$\left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \mathbf{P} & \\ 0 & & & \end{array}\right], \text{ and lastly, consider } \mathbf{P}^{+T}\mathbf{R}^T\mathbf{M}\mathbf{R}\mathbf{P}^+ = \left[\begin{array}{c|c} \lambda & \cdots \\ \hline 0 & \\ \vdots & \mathbf{U} \\ 0 & \end{array}\right].$$

Since $\mathbf{P}^+$ is orthonormal, $\mathbf{M}$ is similar to the final upper triangular matrix above by an orthonormal matrix, $\mathbf{R}\mathbf{P}^+$, as required. □

Hmmm, if we could show that the characteristic polynomials of symmetric matrices split, then we could probably apply Schur's Theorem to get diagonal matrices out of them... shall we begin?

Theorem 1.18. *A real symmetric matrix* $\mathbf{S}$ *may be diagonalized as* $\mathbf{RDR}^T$ *where* $\mathbf{R}$ *is orthonormal.*

Proof. First, to show that $\chi_{\mathbf{S}}$ splits. Let $\bar{x}$ denote the complex conjugate of x. Then all eigenvalues of $\mathbf{S}$ must be real since

$$\begin{aligned}
\mathbf{S}V &= \lambda V && \text{eigenvalue equation}\\
\bar{\mathbf{S}}\bar{V} &= \bar{\lambda}\bar{V} && \text{taking complex conjugates}\\
\mathbf{S}\bar{V} &= \bar{\lambda}\bar{V} && \mathbf{S} \text{ is real}\\
(\mathbf{S}\bar{V})^T &= (\bar{\lambda}\bar{V})^T && \text{taking transposes}\\
\bar{V}^T\mathbf{S} &= \bar{\lambda}\bar{V}^T && \mathbf{S} \text{ is symmetric}\\
\bar{V}^T\mathbf{S}V &= \bar{\lambda}\bar{V}^T V && \text{multiplying by V}\\
\lambda\bar{V}^T V &= \bar{\lambda}\bar{V}^T V && \mathbf{S}V = \lambda V\\
\lambda &= \bar{\lambda} && \text{canceling } \bar{V}^T V
\end{aligned}$$

Thus, $\chi_{\mathbf{S}}$ must split since all eigenvalues of $\mathbf{S}$ are real. And now, since the condition for Schur's theorem is satisfied for symmetric matrices, we can apply it in the following sneaky way:

$$\begin{aligned}
\mathbf{S} &= \mathbf{S}^T && \text{by definition of symmetry}\\
\mathbf{RUR}^T &= \left(\mathbf{RUR}^T\right)^T && \text{by Schur}\\
\mathbf{RUR}^T &= \mathbf{RU}^T\mathbf{R}^T && \text{transposing the right side}\\
\mathbf{U} &= \mathbf{U}^T && \text{canceling } \mathbf{R} \text{ and } \mathbf{R}^T
\end{aligned}$$

Since $\mathbf{U}$ is equal to its own transpose, it must be symmetric, but since it is also upper triangular, only diagonal elements may be non-zero. Therefore $\mathbf{S} = \mathbf{RUR}^T$ where $\mathbf{U}$ is actually diagonal and $\mathbf{R}$ is orthonormal, as required. □

So, we've successfully diagonalized a symmetric matrix. Cool! But for the next example, a skew-symmetric matrix, this same set of tricks won't work. This can be seen by running through the proof that symmetric matrices have all real eigenvalues, but using skew-symmetric matrices instead. The last line will be $\lambda = -\bar{\lambda}$ rather than $\lambda = \bar{\lambda}$. This reveals that the eigenvalues of skew-symmetric matrices are either purely imaginary or zero; we can't use Schur's Theorem to diagonalize these guys. What's more, when we do diagonalize these matrices, we're going to get complex eigenvectors to pair with the complex eigenvalues when we really need everything to be real. Jeez, this is starting to sound *too* complex. Maybe we'll just try 3×3 matrices first, and see how that goes.

For a real skew-symmetric matrix, we know that there will be at least one non-zero eigenvalue, (assuming, as usual, that we're working with a non-zero matrix) and we know that eigenvalue will be purely imaginary. But if we restrict ourselves to the 3×3 matrices, we can calculate the eigenvalues explicitly. A skew-symmetric 3×3 matrix $\mathbf{S}$ has the form $\begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$, and thus $\chi_{\mathbf{S}}(\lambda) = 0$ reduces to $-\lambda^3 = \lambda\left(a^2 + b^2 + c^2\right)$, meaning that the eigenvalues are precisely 0 and $\pm i\sqrt{a^2 + b^2 + c^2}$. And now that we know the eigenvalues, we can actually diagonalize the matrix, bringing us much of the way toward a useful decomposition of this type of matrix.

Theorem 1.19. *A 3×3 skew-symmetric matrix $\mathbf{S}$ may be diagonalized as $\lambda\mathbf{RDR}^T$, where $\lambda \in \mathbb{R}$, $\mathbf{R}$ is orthonormal, and $\mathbf{D} = \begin{bmatrix} i & 0 & 0 \\ 0 & -i & 0 \\ 0 & 0 & 0 \end{bmatrix}$.*

Proof. First, let's check that $\mathbf{S}$ meets the criteria for diagonalizability. It's characteristic polynomial clearly splits since we're working over the complex numbers, so that's fine. And we know by the previous discussion that $\mathbf{S}$ has three distinct eigenvalues, so we don't have to worry about finding different eigenvectors for repetitions of the same eigenvalue. Great! We also know from the previous discussion that $\mathbf{S}$ diagonalizes to the matrix described in the theorem since we actually found its eigenvalues, and that $\lambda \in \mathbb{R}$ because a, b, and c are real. Then all that remains is to show that $\mathbf{R}$ is orthonormal. To that end, let $\mathbf{S}V_1 = \lambda_1 V_1$ and $\mathbf{S}V_2 = \lambda_2 V_2$, where $\lambda_1 \neq \lambda_2$. Then

$$
\begin{aligned}
V_1^T \mathbf{S}^T &= \lambda_1 V_1^T && \text{taking transposes} \\
V_1^T \mathbf{S}^T V_2 &= \lambda_1 V_1^T V_2 && \text{multiplying by } V_2 \\
-V_1^T \mathbf{S} V_2 &= \lambda_1 V_1^T V_2 && \mathbf{S} \text{ is skew-symmetric} \\
-V_1^T \lambda_2 V_2 &= \lambda_1 V_1^T V_2 && \mathbf{S}V_2 = \lambda_2 V_2 \\
\lambda_2 V_1^T V_2 &= \lambda_1 V_1^T V_2 && \text{remembering that } U \cdot V = \sum_{i=1}^{n} u_i \bar{v}_i \text{ over } \mathbb{C}^n \\
V_1^T V_2 &= 0 && \lambda_1 \neq \lambda_2
\end{aligned}
$$

and so V_1 and V_2 are orthogonal since their dot product is zero. Moreover, they can be normalized since eigenvectors are only constrained up to scalar multiplication. □

So we've diagonalized our skew-symmetric matrix into a pretty nice situation, where the diagonal matrix is flanked by orthonormal matrices, just like the symmetric version. But don't forget that we need the eigenvalues and eigenvectors to be real valued, and they are still very much complex. What can we do now? Well, we can always ask our magical crystal ball (the internet)[1] to give us the answer. And it says,

[1] A complete treatment of the diagonalization of skew-symmetric matrices can apparently be found in **??**, according to **??** on p. 581, if the reader is proficient enough to find exactly where in **??** this proof is located.

"Diagonalize the matrix $\tilde{\mathbf{D}} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and the truth shall be revealed."

Alright, let's give it a shot. By the discussion prior to the thoerem, the eigenvalues of this matrix $\tilde{\mathbf{D}}$ are clearly i, $-i$, and 0. The eigenvectors are going to take a bit more work, however. The (non-normalized) matrix of eigenvectors of our general skew-symmetric matrix, $\mathbf{R}$, can be computed by

$$\mathbf{R} = \begin{bmatrix} bc - ia\sqrt{a^2+b^2+c^2} & bc - ia\sqrt{a^2+b^2+c^2} & c \\ a^2+c^2 & a^2+c^2 & -b \\ ab - ic\sqrt{a^2+b^2+c^2} & ab - ic\sqrt{a^2+b^2+c^2} & a \end{bmatrix}.$$

So in our specific case where $a = 1$ and $b = c = 0$, we obtain the (non-normalized) diagonalization of our magical matrix,

$$\tilde{\mathbf{D}} = \tilde{\mathbf{R}}\mathbf{D}\tilde{\mathbf{R}}^* = \begin{bmatrix} -i & i & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i & 0 & 0 \\ 0 & -i & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i & 1 & 0 \\ -i & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $*$, the conjugate transpose of a matrix, is just the complex version of the usual transpose for real matrices.

So, can you see the trick? If we flip around the last equation, we can get $\mathbf{D} = \tilde{\mathbf{R}}^*\tilde{\mathbf{D}}\tilde{\mathbf{R}}$, and if we substitute this into the original diagonalization of the general skew-symmetric matrix, we obtain $\mathbf{S} = \mathbf{R}\tilde{\mathbf{R}}^*\tilde{\mathbf{D}}\tilde{\mathbf{R}}\mathbf{R}^T$, where the almost diagonal matrix $\tilde{\mathbf{D}}$ is real instead of complex. If only $\mathbf{R}\tilde{\mathbf{R}}^*$ was real valued too... which it is! The more compulsive readers are encouraged to double-check this result, and there we have it. Now that we have diagonalized two difficult and general matrices, we are ready for the last section of linear algebra—in a sense, the final and best type of diagonalization.

Singular Value Decomposition

Although it's great for computations when we can diagonalize a matrix, at times it is the case that we are unable to do so, as was intimated by our skew-symmetric example. But sometimes we can't even come close—some matrices have no eigenvalues at all, and diagonalization isn't even defined for matrices that aren't square. We need an alternative in these cases, and singular value decomposition is often the answer. This type of decomposition can be used on any matrix whatsoever, and even uses orthonormal matrices to decompose them. Ok, enough hype; let's see it already.

Theorem 1.20. *A real $m \times n$ matrix $\mathbf{A}$ (where, without loss of generality, $m \geq n$) may be decomposed as $\mathbf{U\Sigma V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0 \in \mathbb{R}$, and*

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \hline & & \mathbf{0} & \end{bmatrix}.$$

Proof. Since $\mathbf{A}$ is so general, we will begin by considering the nicer matrices $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$, which are square ($m \times m$ and $n \times n$, respectively) and symmetric. Then we know by our previous work that $\mathbf{A}^T\mathbf{A}V_i = \lambda_i V_i$ in the usual way. But from here, $\mathbf{AA}^T\mathbf{A}V_i = \lambda_i \mathbf{A}V_i$, so $\mathbf{A}Vi$ is actually an eigenvector of $\mathbf{AA}^T$. Moreover,

$$|\mathbf{A}V_i|^2 = (\mathbf{A}V_i)^T(\mathbf{A}V_i) = V_i^T\mathbf{A}^T\mathbf{A}V_i = V_i^T\lambda_i V_i = \lambda_i,$$

so let $\sigma_i = \sqrt{\lambda_i} = |\mathbf{A}V_i|$ and define U_i to be the normal vector $\frac{\mathbf{A}V_i}{\sigma i}$. Now, consider

$$U_i^T\mathbf{A}V_j = \left(\frac{\mathbf{A}V_i}{\sigma_i}\right)^T\mathbf{A}V_j = \frac{V_i^T\mathbf{A}^T\mathbf{A}V_j}{\sigma_i} = \frac{\lambda_j}{\sigma_i}V_i^TV_j,$$

which is zero when $i \neq j$, and σ_i when $i = j$, since V is orthonormal. We can then combine all of these equations into matrix form to obtain

$$\begin{bmatrix} U_1^T \\ U_2^T \\ \vdots \\ U_n^T \end{bmatrix} \mathbf{A} \left[\begin{array}{c|c|c|c} V_1 & V_2 & \cdots & V_n \end{array}\right] = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}.$$

We must make one final adjustment to this equation in order to complete the proof. Since the number of eigenvectors of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{AA}^T$ is equal, there must be exactly n of them—equal to the smaller of $\mathbf{A}$'s dimensions. Hence, the matrix of eigenvectors corresponding to the larger dimension of $\mathbf{A}$, the eigenvector matrix of $\mathbf{AA}^T$, is not square; it contains only n eigenvectors, not m. Fortunately, this problem is easily resolved. We can just pick $m - n$ more vectors perpendicular to the vectors already in $\mathbf{U}$, and put them at the bottom of the matrix. This will result in the the extra rows of zeros at the bottom of $\mathbf{\Sigma}$ seen in the theorem statement, but now both $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and by shifting them to the other side of the equation, the theorem is proved. □

Before delving into the geometric prerequisites for our view-finding algorithms, let's have a sneak preview of how we're going to apply the SVD; it turns out that the SVD will give the best approximate solution to a homogeneous system of linear equations. More formally, given a matrix $\mathbf{A}$, our goal is to find the vector X that minimizes the vector $|\mathbf{A}X|$. Clearly the zero vector would accomplish this, but as usual, we will be interested only in non-trivial solutions, so we will also impose the constraint that $|X| = 1$. Now let's sit back and let the SVD do the work for us. We get

$$|\mathbf{A}X| = |\mathbf{U\Sigma V}^TX| = |\mathbf{\Sigma V}^TX|$$

by the SVD and since $|\mathbf{U}| = 1$. Then instead of minimizing $|\mathbf{A}X|$ where $|X| = 1$, we can minimize $|\mathbf{\Sigma V}^TX|$ where $|\mathbf{V}^TX| = 1$. It is easy to see by the structure of $\mathbf{\Sigma}$, since the singular values (σ_i) decrease down the diagonal, that the minimizing vector

$$\mathbf{V}^TX = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad \text{and so} \quad X = V\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

So the vector that minimizes $|\mathbf{A}X|$ is simply the last column of V, the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$. We will make use of this key result soon, but first let's have a short break from the algebra with pretty pictures and some geometry.

1.2.2 Epipolar Geometry

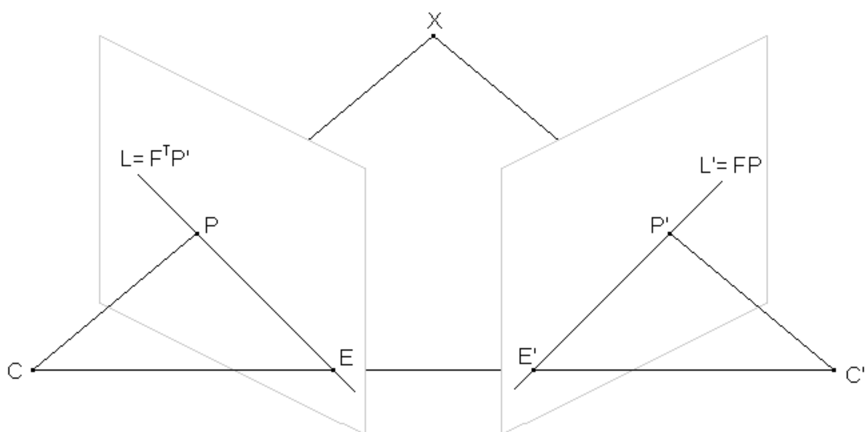

Figure 1.6: The basic relationships of epipolar geometry.

The primary function of interest in epipolar geometry, the fundamental matrix, represents the relationship between projected points seen from one view to the same projected points seen from a different view. However, the mapping is a bit different from the sorts of functions that we are used to seeing, since it does not relate points with points, but rather, points with lines. More precisely, given two views from cameras $\mathbf{C}$ and $\mathbf{C}'$ with the projection of an interest point X from $\mathbf{C}$ being P, the projection of X from $\mathbf{C}'$, call it P', is constrained to lie on the projection of the line $\overleftrightarrow{CX}$ from $\mathbf{C}'$. This line can be computed from P with a 3×3 matrix $\mathbf{F}$ called the fundamental matrix. In order to work out exactly what this equation entails, we're going to have to come up with some rather unique definitions for cameras and for lines, but rest assured, it will all work out in the end.

The Matrices of Cameras and Lines

Definition 1.21. A camera may be understood as a view direction and a location, and will be represented as a 3×4 matrix $\mathbf{C} = [\,\mathbf{R}\,|\,C\,]$, where $\mathbf{R}$ is the camera's view (a rotation from the canonical z-axis) and C is the camera's location (a translation from the origin).

Recall that both of our calculations of novel views of interest points in $\mathbb{R}^3$ had three steps. First we translated the interest point, then we rotated it, then projected it. What this representation of camera matrices is doing for us is essentially just combining the first two steps. See, if we represent interest points I in $\mathbb{R}^3$ as 'homogeneous' four dimensional points of the form

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} \quad \text{instead of the usual} \quad \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

then when we multiply by a camera matrix of the aforementioned form we obtain

$$\mathbf{C}I = [\,\mathbf{R}\,|\,C\,] \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + C;$$

the same result as the first two steps of our other two view calculation methods, but combined into a single matrix multiplication. The only catch is that from now on, we'll have to represent points in $\mathbf{R}^3$ as these homogenous four dimensional points with a 1 as the final entry. Similarly, projected points will now be homogenous three dimensional points with 1 as the final entry, but this is essentially what we've been doing already. When we projected points, we divided out by the third coordinate and concerned ourselves only with the first two. Now, we'll do exactly the same thing, but leave a 1 in the third coordinate as a place holder for further matrix multiplications, in case we need it. And that's it!

Our representation of lines is going to be just a tad more involved. Here, we'll only be concerned with lines in $\mathbb{R}^2$, since the lines we'll be working with will all reside in camera projections of $\mathbb{R}^3$, not in the full space. The definition will be a good place to start.

Definition 1.22. A line in $\mathbb{R}^2$ may be written in standard form as $ax + by + c = 0$. We'll condense this information into a 3×1 matrix of the form $L = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$.

With this particular representation of lines, employing the usual matrix operations (the dot product and cross product) yield quite useful and perhaps unexpected information regarding the relationships between these lines and homogeneous points in $\mathbb{R}^2$. Let's take a moment to explore, shall we?

Theorem 1.23. *A homogenous point P lies on a line L iff $P \cdot L = 0$.*

Proof. Let $P = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$ and $L = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$. Then

$$\begin{aligned} P \in L \quad &\text{iff} \quad a\frac{p}{r} + b\frac{q}{r} + c = 0 \\ &\text{iff} \quad px + qy = -rz \\ &\text{iff} \quad P \cdot L = 0. \end{aligned}$$

□

Theorem 1.24. *The line defined by two non-zero homogeneous points is equal to the cross product of those points. Furthermore, the homogeneous point at the intersection of two lines is equal to the cross product of those lines.*

Proof. For a line defined by two points, let the line L contain the two points P and Q. Then by the previous theorem, $L \cdot P = 0$ and $L \cdot Q = 0$. We know by the usual properties of dot products that two vectors[1] have a dot product of 0 iff they are perpendicular or one of the vectors is zero. We assume that neither point is zero, so L is perpendicular to P and Q. There are only two vectors in $\mathbb{R}^3$ that are perpendicular to both P and Q: $P \times Q$ and $-P \times Q$. Fortunately, it is easy to see by the definition of a line that $L = -L$, so our line through P and Q is well-defined and the first part of the theorem is proved. The proof of the second part is almost exactly the same as the first, and is left to the reader. □

Epipolar Lines and the Fundamental Matrix

Now, to make use of our new notions of cameras and lines, let's figure out what's depicted in Fig. 1.6. First of all, it should be clear that the cameras $\mathbf{C}$ and $\mathbf{C}'$ are projecting the interest point X onto the points P and P', respectively. But the cameras are also projecting each other: $\mathbf{C}$ projects C' onto E and $\mathbf{C}'$ projects C onto E'. The projected point of a camera center by another camera called an epipole, and any line in a camera's projective plane that passes through an epipole is called an epipolar line. Hence, L and L', which are defined to be $E \times P$ and $E' \times P'$ as per the previous theorem, are epipolar lines.

The nice thing about epipolar lines is that they describe the relationship between different projections of the same interest point—no easy task. The problem with projection is that it is not bijective; in Fig. 1.6, $\mathbf{C}$ projects every point on the line $\overleftrightarrow{CX}$ to P, not just X. To put it differently, the inverse of the projection induced by the camera $\mathbf{C}$—call it the 'unprojection,' or, more formally, $\mathbf{C}^\dagger P$—is not well-defined because it doesn't yield the point X, but rather, an arbitrary point on the line $\overleftrightarrow{CX}$. This presents a problem because we're going to need to be able to find $\mathbf{C}'$ given $\mathbf{C}$, P, and P', but we can't do it without knowing exactly where X lies on $\overleftrightarrow{CX}$. The only thing that we know is that since X must be somewhere on $\overleftrightarrow{CX}$, P' lies on the projection of $\overleftrightarrow{CX}$, also known as the epipolar line L'.

That last part is the key, so let's restate and formalize it. Obviously X lies on the line $\overleftrightarrow{CX}$, which could also be defined as the line joining the camera center C and the unprojection of P by $\mathbf{C}$, $\mathbf{C}^\dagger P$. If we were to project those two points by the camera

[1] One great part about these homogenous representations is that we can switch willy-nilly between them and the usual non-homogeneous vectors that we mean when we write arrays of numbers like this; then we can apply all of the usual properties of vectors to the lines and homogeneous points as we see fit since the computational properties of an array of numbers remains the same, no matter what it represents.

$\mathbf{C}'$, we would obtain the points $\mathbf{C}'C$ and $\mathbf{C}'\mathbf{C}^\dagger P$. Now, by the previous theorem, the line between these two projected points is $\mathbf{C}'C \times \mathbf{C}'\mathbf{C}^\dagger P$, which is clearly the projection of the line $\overleftrightarrow{CX}$ by the camera $\mathbf{C}'$ and is also the epipolar line L'. And here we arrive at the desired formulation, a fundamental relationship between camera matrices:

$$L' = FP, \text{ where } F = \mathbf{C}'C \times \mathbf{C}'\mathbf{C}^\dagger \text{ is known as the fundamental matrix.}$$

By Thm. **??**, we have another formulation for the fundamental matrix as well,

$$L' = FP \text{ implies } P'^T FP = 0.$$

We'll use both of these relationships in the upcoming algorithms, but first, let's do a simple example to get a sense of how they work and for the homogeneous points upon which they operate.

We'll use the formula $F = \mathbf{C}'C \times \mathbf{C}'\mathbf{C}^\dagger$ on the simplest conceivable situation and see what we get for F. Let $\mathbf{C} = [\,\mathbf{I}\,|\,0\,]$, the canonical situation a la Blinn, and let the other camera be arbitrary, i.e. $\mathbf{C}' = [\,\mathbf{R}\,|\,t\,]$. Then

$$\begin{aligned} F &= [\,\mathbf{R}\,|\,t\,] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \times [\,\mathbf{R}\,|\,t\,] \begin{bmatrix} \mathbf{I} \\ \hline 0 \end{bmatrix} \\ &= t \times \mathbf{R}. \end{aligned}$$

Now there is a bit of an issue here since we've only defined the cross product between two vectors, not between a vector and a matrix. But it turns out that we can define the cross product in a more general way that encompasses both situations. From here on, for any 3×1 matrix $A = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$, let $A \times B$ refer to the product of the skew matrix $\mathbf{S}_A = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$ with B, i.e. $A \times B = \mathbf{S}_A B$. One may check that this definition preserves the usual behavior of the cross product, and hopefully this will resolve any issues the reader may have with our little abuse of notation. And now, at long last, we come to the three algorithms that we'll use to find the relative positions of different views of the same scene.

1.2.3 Computing F and X from Point Pairs

The Normalized 7-Point Algorithm

normalization: why and how
non-point constraints: why det(F)=0 and —F—=1
7pt alg
cubic solution lottery

The Essential Decomposition

mention $K^T F K = E$
svd to find R and t from E p. 258; points in front of camera? p. 162

Triangulation

derive X's from x,x',E using optimal method (p.315)

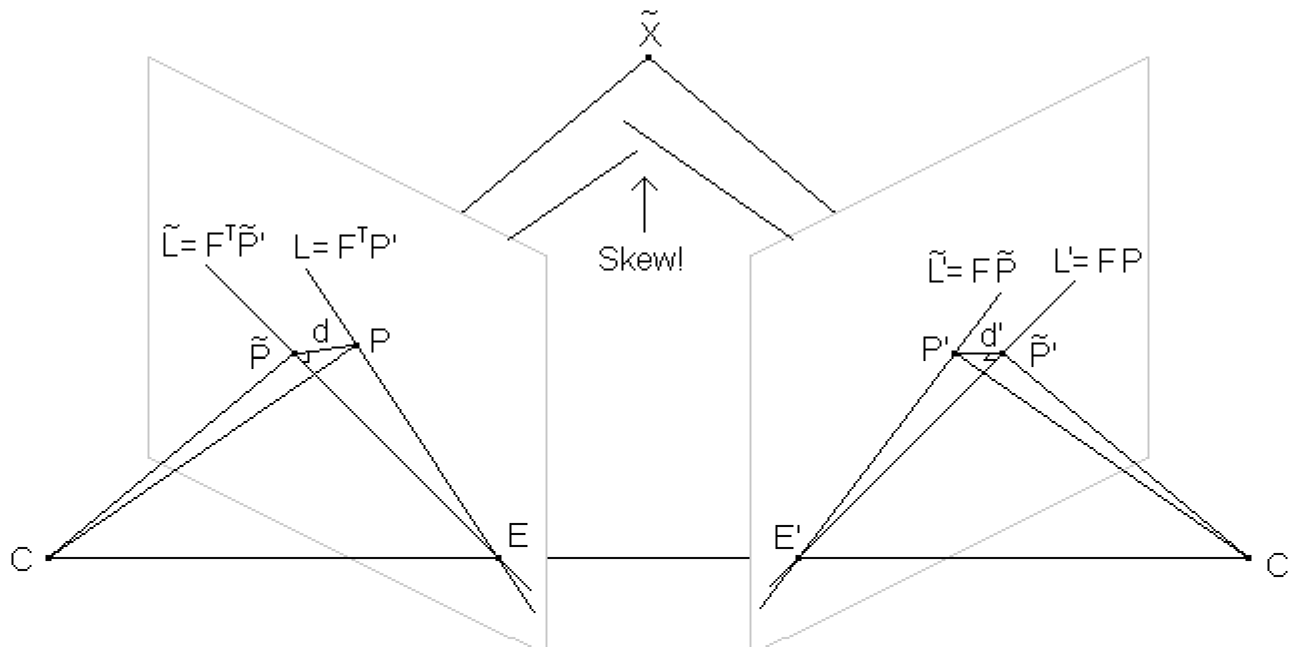

Figure 1.7: Finding proper epipolar relationships in a model with error.