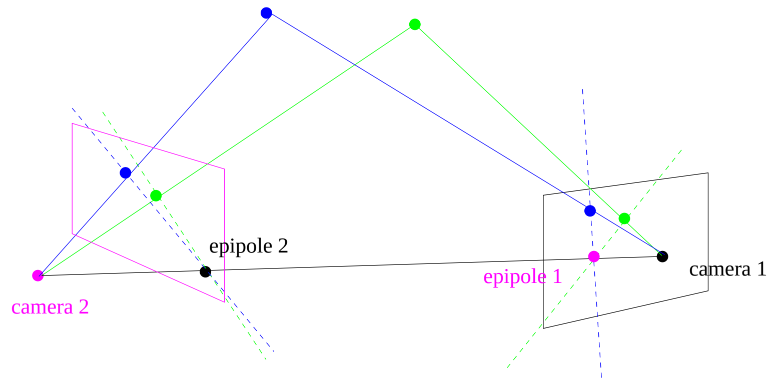
Epipolar Geometry

The figure below shows the basic setup of stereo geometry. We have two cameras, 1 and 2, with centers shown respectively in pink and black. Each camera has a position in 3D. Each camera also has a coordinate system with orthonormal axes, and projection plane illustrated using a rectangle in the figure. The region of the rectangle indicates which part of the projection plane will be sampled by the pixels.



For any 3D point (blue or green, for example), we project that point into the image projection planes of the two cameras. What can we say about this projection? Since the two cameras and the chosen 3D point define a plane Π, the image projection of the 3D point must lie on the intersection of this plane Π with the image projection plane, namely along a line. The dotted lines in the figure show these plane intersections for the two cameras and for two 3D points (blue and green). These dotted lines are called *epipolar* lines.

Where does the term "epipolar lines" come from? All of the lines just described in each projection plane intersect at a single point, and this intersection point is called the *epipole*. What is this special point? The line joining the two cameras lies on every plane Π defined above. Hence the point of intersection where the line joining the two camera intersects the image plane also lies on every epipolar line. (If the optical axis of a camera were perpendicular to the line joining the two cameras, then the epipole would be at infinity.)

Why is it useful to think of epipolar lines? Given two cameras, if we know the two epipoles and the epipolar lines through them (together called the *epipolar geometry*) then we can narrow down the possible correspondences between 2D points in the two images. For example, all 3D points on the blue plane project to the blue epipolar lines. So given a point on a blue epipolar line in the first image, to find the corresponding point in the second image, you only need to search on the corresponding blue epipolar line in the second image.

The Essential Matrix

Let us now translate the above argument from geometry to linear algebra. Suppose we have a 3D point that is written as $\mathbf{X}_1 = (X_1, Y_1, Z_1)^T$ in camera 1's coordinates and the same 3D point is written as $\mathbf{X}_2 = (X_2, Y_2, Z_2)^T$ in camera 2's coordinates. Let the position of camera 2 be $\mathbf{T}_1 = (T_X, T_Y, T_Z)^T$ when written in camera 1's coordinate system. Of course, the position of camera 2 in camera 2's coordinates is $(0,0,0)^T$ and the position of camera 1 in camera 1's coordinates is $(0,0,0)^T$.

The vectors $\mathbf{X}_1$, $\mathbf{T}_1$ and $\mathbf{X}_1 - \mathbf{T}_1$ are linearly dependent and, in particular,

$$(\mathbf{X}_1 - \mathbf{T}_1) \cdot (\mathbf{T}_1 \times \mathbf{X}_1) = 0. \tag{1}$$

Recall the cross product $\mathbf{T}_1 \times \mathbf{X}$ operation on a 3D vector $\mathbf{X}$ can be written as a matrix multiplication,

$$[\mathbf{T}_1]_\times \equiv \begin{bmatrix} 0 & -T_Z & T_Y \\ T_Z & 0 & -T_X \\ -T_Y & T_X & 0 \end{bmatrix}$$

and so we can rewrite Eq. (1)

$$(\mathbf{X}_1 - \mathbf{T}_1)^T \, [\mathbf{T}_1]_\times \, \mathbf{X}_1 = 0.$$

But vector $\mathbf{X}_1 - \mathbf{T}_1$ can be written also as $\mathbf{X}_2$ if that vector is written in camera 1's coordinate axes:

$$\mathbf{X}_1 - \mathbf{T}_1 = \mathbf{R}_1 \mathbf{R}_2^{-1} \mathbf{X}_2$$

So by substitution we get

$$\mathbf{X}_2^T \, \mathbf{R}_2 \mathbf{R}_1^T \, [\mathbf{T}_1]_\times \mathbf{X}_1 = 0.$$

This says: if you take a 3D point $\mathbf{X}_1$ which is in camera 1's coordinates, and you take its cross product with $\mathbf{T}_1$, and apply a rotation so it written in terms of camera 2's axes, you get a vector that is perpendicular to $\mathbf{X}_2$. The matrix

$$\mathbf{E} \equiv \mathbf{R}_2 \mathbf{R}_1^T [\mathbf{T}_1]_\times$$

is called the *essential matrix.* One writes:

$$\mathbf{X}_2^T \, \mathbf{E} \mathbf{X}_1 = 0. \tag{2}$$

We use this equation to define the epipolar lines and epipoles. From Eq. (2), note that we can scale the vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ to $\mathbf{x}_1 = (x_1, y_1, f_1)$ and $\mathbf{x}_2 = (x_2, y_2, f_2)$, respectively, so that they lie in the image projection planes of their respective cameras. So we can write

$$\mathbf{x}_2^T \, \mathbf{E} \mathbf{x}_1 = 0. \tag{3}$$

So, for any point $\mathbf{x_1}$ in the first image, we get an *epipolar* line in the second image by defining the 3-vector $\mathbf{l}_2 \equiv \mathbf{E}\mathbf{x}_1$, namely

$$\mathbf{x}_2^T \mathbf{l}_2 = 0.$$

Similarly, for any point $\mathbf{x}_2$ in the second image, we get an epipolar line in the first image by defining $\mathbf{x}_2^T \mathbf{E} \equiv \mathbf{l}_1^T$, then from (3),

$$\mathbf{l}_1^T \mathbf{x}_1 = 0$$

When the $\mathbf{x}_1$ and $\mathbf{x}_2$ in Eq. 3 are the projections of the same 3D point in the scene, then the epipolar lines both contain the projection of that 3D point.

What about the intersections of the epipolar lines? The essential matrix is of rank 2, since $[\mathbf{T}_1]_\times$ is of rank 2. In particular,

$$[\mathbf{T}_1]_\times \mathbf{T}_1 = \mathbf{0}.$$

Thus $\mathbf{E}\mathbf{T}_1 = \mathbf{0}$ and so $\mathbf{x}_2^T \mathbf{E} \mathbf{T}_1 = \mathbf{0}$ for any $\mathbf{x}_2$.

[The rest of this subsection was updated Nov 12]

Letting $\mathbf{T_1} = (T_x, T_y, T_z)$, it follows that the point $\frac{f_1}{T_z}\mathbf{T}_1$ in camera 1's projection plane must lie on all epipolar lines. This point is the *epipole* $\mathbf{e_1}$. Note that if $T_z = 0$ then the epipole would be a point at infinity.

To obtain the epipole in camera 2, we first write $\mathbf{T}_1$ in terms of camera 2's coordinates, namely

$$\mathbf{T}_2 = \mathbf{R}_2\mathbf{R}_1^T\mathbf{T}_1.$$

We can then project this onto camera 2's projection plane just as we did with camera 1 above, namely divide by the Z coordinate of $\mathbf{T}_2$ and multiply by f_2. This gives $\mathbf{e_2}$.

Substituting $\mathbf{T_2}$ (which is just a scaled version of $\mathbf{e_2}$) into $\mathbf{x_2}$ in (3) you can verify for yourself that

$$\mathbf{e}_2^T\mathbf{E}\mathbf{x_1} = 0$$

and so $\mathbf{e}_2^T\mathbf{l_2} = 0$ and so the epipole in image 2 lies on all epipolar lines.

The Fundamental Matrix

The above arguments relied on points on image projection planes, which is fine if we know the camera internal matrices $\mathbf{K}$ since we can go back and forth from projection plane to pixel coordinates. However, in many cases we don't know the camera internals. What can we say then?

Suppose the two cameras have calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$. If a 3D point (X, Y, Z) is written in the first camera's coordinates, then its pixel position is $(\tilde{x}_1, \tilde{y}_1)$ where

$$\begin{bmatrix} w_1\tilde{x}_1 \\ w_1\tilde{y}_1 \\ w_1 \end{bmatrix} = \mathbf{K}_1\mathbf{X}_1$$

and its pixel position in the second camera is $(\tilde{x}_2, \tilde{y}_2)$ where:

$$\begin{bmatrix} w_2\tilde{x}_2 \\ w_2\tilde{y}_2 \\ w_2 \end{bmatrix} = \mathbf{K}_2\mathbf{X}_2$$

Multiplying both sides of the above matrix by their respective $\mathbf{K}^{-1}$ matrices, and dropping the scalars w_1 and w_2, we can rewrite Eq. (2) in terms of the pixel coordinates

$$\begin{bmatrix} \tilde{x}_2 & \tilde{y}_2 & 1 \end{bmatrix} \mathbf{K}_2^{-T}\ \mathbf{E}\ \mathbf{K}_1^{-1} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ 1 \end{bmatrix} = 0$$

where $\mathbf{K}^{-T}$ denotes the transpose of the inverse of $\mathbf{K}$ (or equivalently, the inverse of the transpose).

Defining

$$\mathbf{F} \equiv\ \mathbf{K}_2^{-T}\ \mathbf{E}\ \mathbf{K}_1^{-1}$$

we get

$$\tilde{\mathbf{x}}_2^T\ \mathbf{F}\tilde{\mathbf{x}}_1 = 0.$$

$\mathbf{F}$ is called the *fundamental matrix.* It relates pixel positions $(\tilde{x}_1, \tilde{y}_1)$ and $(\tilde{x}_2, \tilde{y}_2)$ in the two cameras which are the projections of the same 3D scene point.

Epipolar lines and epipoles are defined in exactly the same way as for the essential matrix. For any pixel $\tilde{\mathbf{x}}_1 = (\tilde{x}_1, \tilde{y}_1, 1)^T$ in the first image , we define

$$\mathbf{l_2} \equiv \ \mathbf{F} \ \tilde{\mathbf{x}}_1$$

and then $\tilde{\mathbf{x}}_2^T \mathbf{l_2} = 0$ is an epipolar line in the second image. Similarly, for any pixel $\tilde{\mathbf{x}}_2 = (\tilde{x}_2, \tilde{y}_2, 1)$ in the second image, we define

$$\mathbf{l_1} \equiv \tilde{\mathbf{x}}_2^T \ \mathbf{F}$$

and so $\mathbf{l}_1^T \tilde{\mathbf{x}}_\mathbf{1} = 0$ is an epipolar line in the first image.

Since the essential matrix $\mathbf{E}$ has rank 2, so does $\mathbf{F}$ since the $\mathbf{K}$ matrices are invertible. In particular, the the epipole $\tilde{\mathbf{e}}_1 = \mathbf{K_1 e_1}$ in pixel coordinates in the camera 1 image is in the null space of $\mathbf{F}$, namely $\mathbf{F}\tilde{\mathbf{e}}_\mathbf{1} = \mathbf{0}$. The epipole is the intersection of all the epipolar lines. Similarly, the epipole $\tilde{\mathbf{e}}_\mathbf{2} = \mathbf{K_2 e_2}$ in the camera 2 image is in the (left) null space of $\mathbf{F}$, namely $\tilde{\mathbf{e}}_\mathbf{2}\mathbf{F} = 0$. Note that the epipoles might not lie within the image domain (which is finite). Indeed they could even be at infinity.

That completes our mathematical discussion of the essential matrix and fundamental matrix. Admittedly, the notation can be challenging. I think one of the main challenges is that choosing a point in image 1 defines a line in image 2 (and vice versa). Make sure you understand both the geometry and linear algebra statements of how this point/line combination works.

Lastly, it is important to keep in mind why we care about the fundamental matrix. The reason is this: given two camera images of one scene, if you know the fundamental matrix that relates the two images, then the correspondence problem (which we will discuss next class) of finding which points in one image correspond to which points in the other image is restricted to searching along lines rather than searching anywhere. Thus, if we can estimate $\mathbf{F}$, but we do not know the camera calibration matrices (internal) or the rotation and translation between cameras (external), then we can still greatly simplify the correspondence problem. More on this next class...