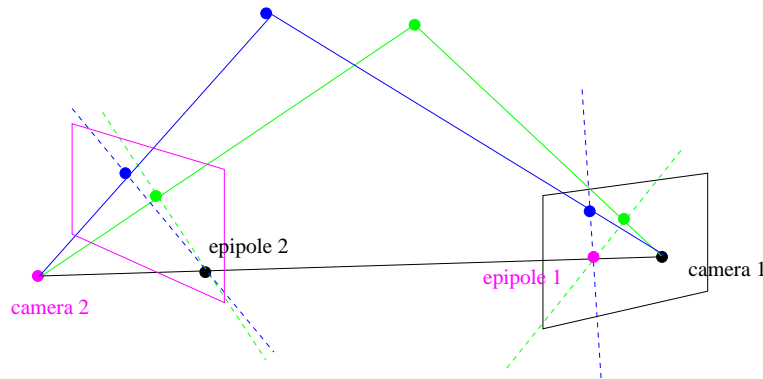


Stereo and Epipolar Geometry

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).



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.

All epipolar lines in each projection plane intersect at a single point. To see this, note that the line joining the two cameras lies on every plane Π defined above, hence the point of intersection of this line with the image plane also lies on every epipolar line. We call this point the *epipole*. (If the optical axis of a camera were perpendicular to the line joining the two cameras, then the epipole would be at infinity.)

Given two cameras, *if we know the epipolar geometry (namely the epipoles and epipolar lines) 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 blue point in the first image, to find the corresponding point in the second image, you only need to search for on the blue epipolar line.

The Essential Matrix

Let us now translate the above argument from geometry to 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 linear dependent and, in particular,

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

Recall the cross product with \mathbf{T} operation can be written as a matrix multiplication,

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

and so

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

But $\mathbf{X}_1 - \mathbf{T}_1$ can be written

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

and 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 \mathbf{X}_1 and cross it 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*, and so one writes:

$$\mathbf{X}_2^T \mathbf{E} \mathbf{X}_1 = 0. \quad (2)$$

Now we are ready to define 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, for any \mathbf{X}_1 , we get an *epipolar* line in the camera 2 projection plane, as follows. Define a 3-vector $\mathbf{l}_2 \equiv \mathbf{E} \mathbf{x}_1$. Then

$$\mathbf{x}_2^T \mathbf{E} \mathbf{x}_1 = \mathbf{x}_2^T \mathbf{l}_2 = 0$$

which is the *epipolar line* in the second image. Similarly, for any 3D point \mathbf{X}_2 , we project to \mathbf{x}_2 and define $\mathbf{l}_1^T \equiv \mathbf{x}_2^T \mathbf{E}$, then

$$\mathbf{x}_2 \mathbf{E} \mathbf{x}_1 = \mathbf{l}_1^T \mathbf{x}_1 = 0$$

which is the epipolar line in the first image, namely the line to which that 3D point must project.

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, $[\mathbf{T}_1]_{\times} \mathbf{T}_1 = \mathbf{0}$ and so $\mathbf{E} \mathbf{T}_1 = \mathbf{0}$. Thus, $\mathbf{x}_2^T \mathbf{E} \mathbf{T}_1 = \mathbf{0}$ for any \mathbf{x}_2 . Thus, the point $f_1 \frac{\mathbf{T}_1}{|\mathbf{T}_1|}$ in camera 1's projection plane must lie on all epipolar lines. It is the *epipole*.

The epipole in camera 2 will be obtained directly by writing \mathbf{T} in terms of camera 2's coordinates, namely $\mathbf{R}_2 \mathbf{R}_1^T \mathbf{T}_1$. To do it more slowly, a point \mathbf{x}_2 is the epipole in camera 2 if $\mathbf{x}_2^T \mathbf{E} = \mathbf{0}$, or equivalently, $\mathbf{E}^T \mathbf{x}_2 = \mathbf{0}$. But

$$\mathbf{E}^T \mathbf{x}_2 = -[\mathbf{T}_1]_{\times} \mathbf{R}_1 \mathbf{R}_2^T \mathbf{x}_2,$$

since $[\mathbf{T}_1]_{\times}$ is anti-symmetric. So, for the camera 2 epipole, we want $\mathbf{R}_1 \mathbf{R}_2^T \mathbf{x}_2 = \mathbf{T}_1$. So the epipole is $\mathbf{x}_2 = \mathbf{R}_2 \mathbf{R}_1^T \mathbf{T}_1$, which is just \mathbf{T}_2 , that is, the translation vector written in camera 2's coordinates.

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 from 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 (x_1, y_1) where

$$\begin{bmatrix} w_1 x_1 \\ w_1 y_1 \\ w_1 \end{bmatrix} = \mathbf{K}_1 \mathbf{X}_1$$

and its pixel position in the second camera is (x_2, y_2) where:

$$\begin{bmatrix} w_2 x_2 \\ w_2 y_2 \\ w_2 \end{bmatrix} = \mathbf{K}_2 \mathbf{X}_2$$

and (x_1, y_1) and (x_2, y_2) are pixel coordinates, not projection plane positions.

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

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0.$$

or

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

where

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

and \mathbf{F} is called the *fundamental matrix*. It relates corresponding pixels (x_1, y_1) and (x_2, y_2) in the two cameras. Note that \mathbf{K}^{-T} denotes the transpose of the inverse of \mathbf{K} (or equivalently, the inverse of the transpose).

Epipolar lines (and epipoles) are defined in exactly the same way as was done for the essential matrix. Any pixel $\mathbf{x}_1 = (x_1, y_1, 1)^T$ in the first image defines a vector

$$\mathbf{l}_2 = \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \mathbf{x}_1$$

such that $\mathbf{x}_2 \cdot \mathbf{l}_2 = 0$, which is an epipolar line in the second image. Similarly, any pixel $\mathbf{x}_2 = (x_2, y_2, 1)$ in the second image defines a vector

$$\mathbf{l}_1 = \mathbf{x}_2^T \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1}$$

such that $\mathbf{l}_1 \cdot \mathbf{x}_1 = 0$, which is an epipolar line in the first image.

\mathbf{F} has rank 2, since the essential matrix \mathbf{E} has rank 2. The epipole \mathbf{e}_1 in the camera 1 image is in the null space of \mathbf{F} , namely it is the intersection of all the epipolar lines. Similarly, the epipole

\mathbf{e}_2 in the camera 2 image is in the null space of \mathbf{F}^T . Note that the epipoles might not lie within the image domain (which is finite). Indeed they could even be at infinity.

I emphasize that if you know the fundamental matrix \mathbf{F} , 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 lines. 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.

Estimation of Fundamental matrix (8 point algorithm)

How can we estimate the fundamental matrix that relates two images? Suppose we find a pair of corresponding points (x_1, y_1) and (x_2, y_2) in the first and second camera's image, respectively. We then would have the following constraint on the nine \mathbf{F}_{ij} elements.

$$(x_1x_2, y_1x_2, x_2, x_1y_2, y_1y_2, y_2, x_1, y_1, 1) \cdot (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}) = 0$$

If we have eight pairs of corresponding points, we get a system of 8 equations with 9 unknowns. This defines an 8×9 matrix. The null space of this system of equations would give us an exact estimate of \mathbf{F} . This is called the *eight point algorithm* for estimating \mathbf{F} .

The positions of the eight corresponding points will typically be noisy. To get a more accurate estimate, we can use $N \gg 8$ corresponding points and take the SVD of an $N \times 9$ matrix. This amounts to solving the least squares problem, of finding an \mathbf{F} that minimizes:

$$\sum_{i=1}^N (\mathbf{x}_2^{iT} \mathbf{F} \mathbf{x}_1^i)^2$$

subject to a constraint such as $\|\mathbf{F}\| = 1$. For the solution, we take the last column of \mathbf{V}^T , which corresponds to the smallest eigenvalue.

Several observations should be made. First, the fundamental matrix is a 3×3 matrix of rank 2, and it has seven degrees of freedom. To see this, note that the rank 2 constraint implies that any column is a linear combination of the other two columns e.g. the third column is a linear combination of the first two columns. So the first two elements of the third column specify the linear combination and hence specify the third element of the third column. This suggests there are eight degrees of freedom. However, recall that we are working with homogeneous coordinates, so you can scale \mathbf{F} by any constant without changing the $\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0$ constraint. This removes one of the eight degrees of freedom, leaving seven.

Second, if there is any error/noise in the positions of the points then the ninth singular value will not be zero, and most likely the estimated \mathbf{F} will be of rank 3 rather than rank 2. If we use this estimated \mathbf{F} , then the epipolar lines will not intersect exactly at the epipoles, and the epipolar constraints will not hold exactly. In this case, we find a rank 2 approximation of \mathbf{F} . We write

$$\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T$$

and one finds a rank 2 matrix that is close to \mathbf{F} in the least squares sense. (Recall this technique was used in the factorization method last class.) We obtain such a rank 2 matrix by setting the third singular value to 0. (That is, set Σ_{33} to 0.) This forces the \mathbf{F} matrix to be rank 2. It forces

the corresponding points to line on epipolar lines and all epipolar lines in an image pass through an epipole. (Of course, since we are making an approximation in the presence of noise, we are not guaranteed that the true corresponding points will lie on corresponding epipolar lines. But it is the best we can do.)

Finally, if we are not sure about corresponding points, then we can use a method like RANSAC to protect ourselves against matching outliers. This is similar to fitting a homography, so I don't need to go through the details here. (I elaborated a bit more in the slides.)

Rectification (based on \mathbf{F})

Suppose we have estimated a fundamental matrix that relates two images. We would like to find corresponding points in the two images. For each point in the left image, the fundamental matrix gives us the line in the right image where the corresponding point must lie. Similarly, for each point in the right image, the fundamental matrix gives us the line in the left image where the corresponding point must lie. Thus, finding the corresponding points really only involves a 1D search, rather than a 2D search.

Next lecture I will sketch out an algorithm for solving the correspondence problem. This algorithm attempts to match points on corresponding epipolar lines. One could find corresponding epipolar lines by brute force: take a point \mathbf{x}_1 in camera 1. The line from the epipole through \mathbf{x}_1 is an epipolar line in camera 1. The corresponding line in camera 2 is just $\mathbf{F}\mathbf{x}_1$. So we can easily find corresponding epipolar lines.

Rather than doing the brute force procedure just described, it one can use a homography to transform the two images such that, for any scene point, the epipolar lines containing (the projection of) that scene point are the same row in the two transformed (rectified) images. Here I will only sketch out the basic idea.

Assuming we have \mathbf{F} and thus we know \mathbf{e}_1 and \mathbf{e}_2 , we compute homographies \mathbf{H}_1 and \mathbf{H}_2 which map the respective epipoles to the point at infinity, $(\pm 1, 0, 0)$, and also is such that corresponding points lie in the same row i.e. have the same y value. So, for example, let $\mathbf{e}_1 = (e_u, e_v, 1)$. Then we could choose

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{e_v}{e_u} & 1 & 0 \\ -\frac{1}{e_u} & 0 & 1 \end{bmatrix}$$

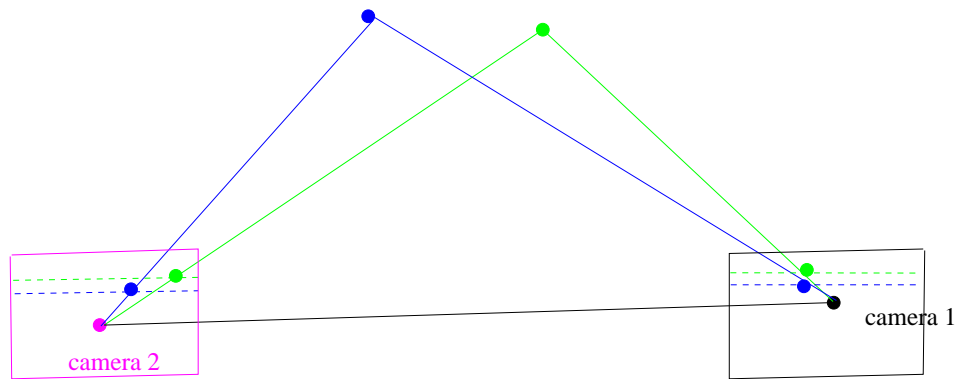
which clearly maps the epipole in camera 1 to the right place.

To choose \mathbf{H}_2 , we need to satisfy the constraint that $\mathbf{H}_2\mathbf{e}_2$ also is a point at infinity in the x -direction, so we take

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

and we need to choose the h values such that the rows of the mapped images are in alignment. This is not difficult to do¹ but there are many further details. For the sake of saving time, I'll omit them and move on.

¹e.g. if you want the details, see "Projective rectification from the fundamental matrix" by J. Mallon and P.F. Whelan. *Image and Vision Computing* (2005) and references therein



Projective Reconstruction Theorem

It is important to appreciate that if you don't know the camera internals and externals, then there are strong limitations on how much you can say about the true geometry of scene points, *even if you can find correspondences between all points in the left image and all points in the right image*. Here is an example of these limitations.

Suppose the two cameras are \mathbf{P}_1 and \mathbf{P}_2 . Then for any 3D point in the scene, $\mathbf{X} = (X, Y, Z)$ – note it is written in the scene coordinate system – we get image pixel positions:

$$\begin{bmatrix} w_1 x_1 \\ w_1 y_1 \\ w_1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} w_2 x_2 \\ w_2 y_2 \\ w_2 \end{bmatrix} = \mathbf{P}_2 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

But notice that, for any 4×4 invertible matrix \mathbf{M} , exactly the same image position would be produced by cameras $\mathbf{P}_1 \mathbf{M}$ and $\mathbf{P}_2 \mathbf{M}$, and by 3D points $\mathbf{M}^{-1}(X, Y, Z, 1)^T$ since

$$\begin{bmatrix} w_1 x_1 \\ w_1 y_1 \\ w_1 \end{bmatrix} = (\mathbf{P}_1 \mathbf{M})(\mathbf{M}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix})$$

and similarly for camera 2. If we only have correspondences between pixels in the two images, there are many possible cameras and scene points that would produce such corresponding points. Thus, we say that stereo for uncalibrated cameras can only be solved up to a “projective ambiguity”. This is often called the *projective reconstruction theorem*.

Depth estimation

Let's suppose we have rectified two images, as described above. Let's take this situation as our new starting point. We pretend that we have a pair of images (a stereo pair) that was obtained by two

cameras with the same (unknown) internal parameters and whose external parameters are simply related: the cameras have the same orientation (i.e. parallel coordinate axes) and are separated by an unknown baseline translation $(T, 0, 0)$ and have a projection plane at unknown $Z = 1$. So, we *assume*

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & -T \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(Here I am being a bit sloppy since I am not distinguishing pixel units from mm units. However, since all internals are unknown anyhow, there is not much point in being careful with the units.)

With the above assumptions, any 3D point in the scene can be represented as (X_1, Y_1, Z_1) in camera 1's rectified coordinates and $(X_2, Y_2, Z_2) = (X_1 - T, Y_1, Z_1)$ in camera 2's rectified coordinates, and this 3D point will project to image positions in the two cameras:

$$(x_1, y_1) = \left(\frac{X_1}{Z_1}, \frac{Y_1}{Z_1} \right)$$

$$(x_2, y_2) = \left(\frac{X_1 - T}{Z_1}, \frac{Y_1}{Z_1} \right).$$

(Again, I am not worry about units here.)

We now define the *binocular disparity* to be

$$d = x_1 - x_2 = \frac{T}{Z}.$$

If we can now find corresponding points in the two images, then we can trivially estimate the binocular disparity of these points and estimate the depths Z to the corresponding 3D scene point (up to an unknown constant T , and an unknown projective transform).