Epipolar Geometry and the Eight-Point Algorithm

Carlo Tomasi

March 8, 2021

The *epipolar geometry* of a pair of cameras expresses the fundamental relationship between any two corresponding points in the two image planes, and leads to a key constraint between the coordinates of these points that underlies visual reconstruction. The first Section below describes the epipolar geometry. The Section thereafter expresses the key constraint algebraically. Finally, Section 3 uses the epipolar geometry to develop an algorithm that reconstructs the relative positions of the cameras and the three-dimensional position of points in the world from two images of at least eight points.

1 The Epipolar Geometry of a Pair of Cameras

Figure 1 shows the main elements of the epipolar geometry for a pair of cameras.

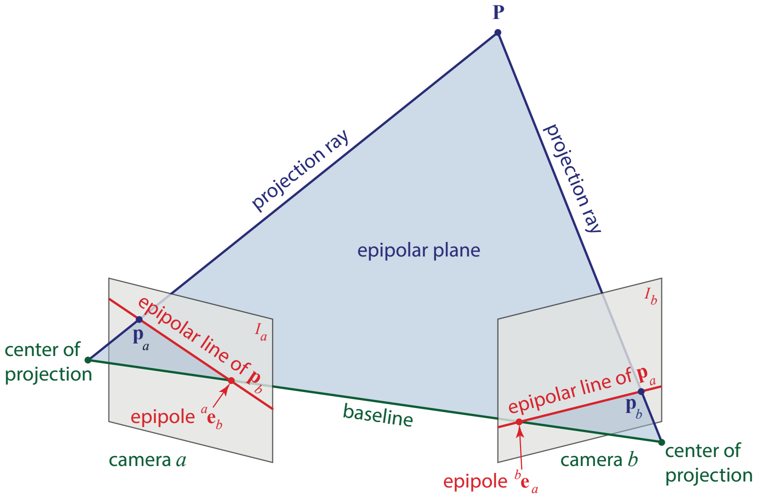

Figure 1: Essential elements of the epipolar geometry of a camera pair.

The world point $\mathbf{P}$ and the centers of projection of the two cameras identify a plane in space, the *epipolar plane* of point $\mathbf{P}$. The Figure shows a triangle of this plane, delimited by the two projection rays and by the *baseline* of the camera pair, that is, the line segment that connects the two centers of projection.[1]

[1] We use the term "baseline" for the line *segment*. However, this term is also often used for the *length* of this segment, or even for the entire *line* through the two centers of projection.

If the image planes are thought of extending indefinitely, the baseline intersects the two image planes at two points called the *epipoles* of the two images. In particular, if the cameras are arranged so that the baseline is parallel to an image plane, then the corresponding epipole is a point at infinity.

The epipoles are fixed points for a given camera pair configuration. With cameras somewhat tilted towards each other, and with a sufficiently wide field of view, the epipoles would be inside the image. Epipole $\mathbf{e}_b$ in the image I_a taken by camera a would be literally the image of the center of projection of camera b in I_a, and *vice versa.* Even if the two cameras do not physically see each other, we maintain this description in an abstract sense: each epipole is the image of one camera in the other image, even if this point is outside the field of view. Note that the epipole in image I_a is called $\mathbf{e}_b$, because it is the image *of* camera b from camera a. Similar considerations hold for $\mathbf{e}_a$.

The epipolar plane intersects the two image planes along the two *epipolar lines* of point $\mathbf{P}$, each of which passes by construction through one of the two projection points $\mathbf{p}_a$ and $\mathbf{p}_b$ and one of the two epipoles. Thus, epipolar lines come in corresponding pairs, and the correspondence is established by the single epipolar plane for the given point $\mathbf{P}$.

For a different world point $\mathbf{P}$, the epipolar plane typically changes, and with it do the image projections of $\mathbf{P}$ and the epipolar lines. However, all epipolar planes contain the baseline. Thus, the set of epipolar planes forms a *pencil* of planes supported by the line through the baseline, and the epipoles are fixed.

Suppose now that we are given the two images I_a and I_b taken by cameras a and b and a point $\mathbf{p}_a$ in I_a. If all we have is the images, we do not know where the corresponding point $\mathbf{p}_b$ is in the other image, nor where the world point $\mathbf{P}$ is, except that $\mathbf{P}$ must be somewhere along the projection ray of $\mathbf{p}_a$. However, *if in addition we know the relative position and orientation of the two cameras*, we know where the two centers of projection are relative to each other. The two centers of projection and point $\mathbf{p}_a$ identify the epipolar plane, and this in turn determines the epipolar line of point $\mathbf{p}_a$ in image I_b. The point $\mathbf{p}_b$ must be somewhere on this line. This same construction holds for any other point $\mathbf{p}_a$ on the epipolar line of $\mathbf{P}$ in image I_a. We have not pinned down $\mathbf{p}_b$, but we have narrowed down its possible positions to be somewhere on a known line.

To understand what the epipolar constraint expresses, consider that the projection rays for two arbitrary points in the two images are generically two skew lines in space. The projection rays of two *corresponding* points, on the the other hand, are coplanar with each other and with the baseline, because they belong to the same epipolar plane. The epipolar geometry captures this key constraint, and pairs of point that do not satisfy the constraint cannot possibly correspond to each other.

2 The Essential Matrix

This section expresses the *epipolar constraint* described in the previous section algebraically.

Coordinate Systems. The canonical reference system for camera a is a right-handed Cartesian coordinate system with its origin at the center of projection of a, its positive Z axis pointing towards the scene along the optical axis of the lens, and its X axis pointing to the right[2] along the rows of the camera sensor. As a consequence, the Y axis points downwards along the columns of the

[2] When the camera is upside-up and viewed from behind it, as when looking through its viewfinder.

sensor. The canonical reference system for camera b is defined similarly. Let

$$ {}^a\mathbf{p}_a = \begin{bmatrix} {}^a x_a \\ {}^a y_a \\ f \end{bmatrix} \quad \text{and} \quad {}^b\mathbf{p}_b = \begin{bmatrix} {}^b x_b \\ {}^b y_b \\ f \end{bmatrix} $$

denote the coordinates, relative to each camera's canonical reference system, of the image points that are the projections of the same world point $\mathbf{P}$. Please pay attention to this definition: ${}^a\mathbf{p}_a$ is a point on the image plane, but is here viewed as a point in three-dimensional space. Like all points on the image plane of camera a, its third (Z) coordinate in the camera's reference system is f, the camera's focal distance. Similar considerations hold for ${}^b\mathbf{p}_b$. Also, since each point is observed in its own camera, the reference system (left superscript) is that of the camera the point appears in (right subscript).

Finally, let

$$ {}^b\mathbf{p} = {}^aR_b({}^a\mathbf{p} - {}^a\mathbf{t}_b) \tag{1} $$

be the rigid transformation between the two reference systems. As we know, the reverse transformation is

$$ {}^a\mathbf{p} = {}^bR_a({}^b\mathbf{p} - {}^b\mathbf{t}_a) \quad \text{where} \quad {}^bR_a = {}^aR_b^T \quad \text{and} \quad {}^b\mathbf{t}_a = -{}^aR_b{}^a\mathbf{t}_b \; . \tag{2} $$

The Essential Matrix. When expressed in the reference system of camera a, the directions of the projection rays through corresponding image points $\mathbf{p}_a$ and $\mathbf{p}_b$ are along the vectors

$$ {}^a\mathbf{p}_a \quad \text{and} \quad {}^bR_a{}^b\mathbf{p}_b \; , $$

and the baseline in this reference system is along the translation vector ${}^a\mathbf{t}_b$.

To simplify the notation in the manipulations that follows, we define

$$ \mathbf{a} = {}^a\mathbf{p}_a \quad , \quad \mathbf{b} = {}^b\mathbf{p}_b \quad , \quad R = {}^aR_b \quad , \quad \mathbf{t} = {}^a\mathbf{t}_b \quad , \quad \mathbf{e} = {}^a\mathbf{e}_b $$

to be the image measurements of the two corresponding points (each viewed as a three-dimensional point in its own camera's reference system), the parameters of the coordinate transformation from camera a to camera b, and the epipole of b in a. Then, the rotation and translation in the reverse direction are

$$ R^T = {}^bR_a \quad \text{and} \quad -R\mathbf{t} = {}^b\mathbf{t}_a \; . $$

Coplanarity of the projection-ray directions $\mathbf{a}$ and $R^T\mathbf{b}$ and baseline $\mathbf{t}$ can be expressed by stating that their triple product is zero:

$$ (R^T\mathbf{b})^T(\mathbf{t} \times \mathbf{a}) = 0 \quad \text{that is,} \quad \mathbf{b}^T R\,(\mathbf{t} \times \mathbf{a}) = 0 \quad \text{or} \quad \mathbf{b}^T R\,[\mathbf{t}]_\times \mathbf{a} = 0 $$

where $\mathbf{t} = (t_x, t_y, t_z)^T$ and

$$ [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} $$

is the skew-symmetric matrix that expresses the cross-product of $\mathbf{t}$ with any other vector.

In summary, for corresponding points $\mathbf{a}$ and $\mathbf{b}$ the following equation holds:

$$ \mathbf{b}^T E\,\mathbf{a} = 0 \tag{3} $$

where

$$E = R\,[\mathbf{t}]_\times \; . \tag{4}$$

Equation (3) is called the *epipolar constraint* and the matrix E is called the *essential matrix.* Equation (3) expresses the coplanarity between *any* two points $\mathbf{a}$ and $\mathbf{b}$ on the same epipolar plane for two fixed cameras.

If point $\mathbf{b}$ is fixed in image I_b, then the product

$$\boldsymbol{\lambda}^T = \mathbf{b}^T E \tag{5}$$

is a fixed row vector. If the fixed point $\mathbf{a}$ is replaced by a variable vector $\mathbf{x}$ in image I_a, then equation (3) can be written as follows:

$$\boldsymbol{\lambda}^T \mathbf{x} = 0 \; . \tag{6}$$

This is a single linear equation in the coordinates of $\mathbf{x}$, and therefore represents a line in the image plane of I_a. The point $\mathbf{a}$ satisfies this equation by equation (3). Also the translation vector $\mathbf{t}$ satisfies equation (6), because

$$\boldsymbol{\lambda}^T \mathbf{t} = \mathbf{b}^T E\mathbf{t} = \mathbf{b}^T R\,[\mathbf{t}]_\times\,\mathbf{t} = 0$$

(recall that the cross product of a vector with itself is zero). The epipole $\mathbf{e}$ in image I_a is on the baseline, and therefore its coordinates in the reference frame of camera a are proportional to those of $\mathbf{t}$, so $\mathbf{e}$ satisfies equation (6) as well. Thus, this equation represents the line through $\mathbf{a}$ and $\mathbf{e}$, that is, the epipolar line of $\mathbf{b}$ in image I_a: If we knew the essential matrix E for a pair of cameras, then we could find the equation of the epipolar line for every point $\mathbf{b}$ in I_b.

This state of affairs must of course hold the other way around as well, when the roles of the two cameras are switched. Before seeing this in more detail, however, we explore the structure of the essential matrix E.

The Structure of E. First, this matrix cannot be full rank, as the following geometric argument proves: Since the epipole in image I_a belongs to *all* epipolar lines in I_a, not just one, the vector $\mathbf{e}$ of its coordinates must satisfy equation (6) *regardless of what point* $\mathbf{b}$ *is used* in the definition (5) of $\boldsymbol{\lambda}$. This can happen only if $\mathbf{e}$ is in the null space of E, so this matrix must be degenerate.

The degeneracy of E can also be shown algebraically. More specifically, it is easy to see that the rank of E is two for any nonzero $\mathbf{t}$. To this end, note first that the matrix $[\mathbf{t}]_\times$ has rank two if $\mathbf{t}$ is nonzero, because

$$[\mathbf{t}]_\times \mathbf{t} = \mathbf{t} \times \mathbf{t} = \mathbf{0}$$

and the null space of $[\mathbf{t}]_\times$ is exactly the line through the origin and along $\mathbf{t}$. Since R is full rank, also the product $E = R\,[\mathbf{t}]_\times$ has rank 2 if $\mathbf{t} \neq \mathbf{0}$. In addition, the null space of E and that of $[\mathbf{t}]_\times$ are the same, because the solutions to the two systems

$$[\mathbf{t}]_\times \mathbf{x} = 0 \quad \text{and} \quad E\,\mathbf{x} = 0$$

are the same, since R is full rank. Therefore, *the rank of E is 2 if* $\mathbf{t}$ *is nonzero, and the null space of E is the line spanned by* $\mathbf{t}$ *(or equivalently* $\mathbf{e}$*).*

There is more to the structure of E. For any vector $\mathbf{v}$ orthogonal to $\mathbf{t}$, the definition of cross product yields

$$\|[\mathbf{t}]_\times \mathbf{v}\| = \|\mathbf{t}\|\,\|\mathbf{v}\| \; .$$

The vector $\mathbf{v}$ is orthogonal to $\mathbf{t}$ if it is in the row space of $[\mathbf{t}]_\times$, and the equation above then shows that the matrix $[\mathbf{t}]_\times$ maps all unit vectors ($\|\mathbf{v}\| = 1$) in its row space into vectors of magnitude $\|\mathbf{t}\|$. From the definition of SVD, this means that the two nonzero singular values of $[\mathbf{t}]_\times$ are equal to each other.[3] Since multiplication by an orthogonal matrix (R) does not change the matrix's singular values, we conclude that *the essential matrix E has two nonzero singular values equal to each other, and one zero singular value.* The right singular vector $\mathbf{v}_3$ corresponding to the zero singular value of E is a unit vector along the epipole and the translation vector,

$$\mathbf{v}_3 \sim \mathbf{e} \sim \mathbf{t} \; . \tag{7}$$

In these expressions, the symbol '$\sim$' means "proportional to," or "equal up to a multiplicative constant." Since the two nonzero singular values of E are equal to each other, the corresponding right singular vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are arbitrary, as long as they form an orthonormal triple with $\mathbf{v}_3$.

Scale and Epipoles at Infinity. Since the systems involving the essential matrix E are all homogeneous, the translation vector $\mathbf{t}$ and the epipole $\mathbf{e}$ can only be found up to a scale factor. This limitation is consistent with the fact that cameras fundamentally measure angles between projection rays, and cannot measure lengths. For instance, if two images show a building, it is not possible to determine from image measurements alone whether the pictures are of a real building taken from two cameras, say, three meters apart, or they are images of a miniature building perhaps a hundred times smaller, taken from two cameras that are three centimeters apart. Scale is irretrievably lost in imaging, even if multiple cameras are used and as long as only the images are available. Of course, if we knew, say, the length of the baseline, or the height of the building, then we could determine the scale factor.

While this loss of scale is generally a disadvantage of passive imaging with cameras at unknown positions, it has a positive consequence on the representation of epipoles and translation when the baseline is parallel to the image plane of either camera.

To understand this observation, consider a situation in which the angle $\theta = \theta_0$ between the optical axis of camera a and the baseline is less than 90 degrees, as illustrated in Figure 2. The orientation of camera b does not matter for this argument. Then, the baseline crosses the image plane of camera a at the epipole $\mathbf{e}$ of b in image I_a, and the translation vector from a to b is proportional to $\mathbf{e}$:

$$\mathbf{e} = \begin{bmatrix} e_{x0} \\ e_{y0} \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{t} = c\,\mathbf{e}$$

where c is some constant.

Now gradually increase the angle θ beyond θ_0 by rotating the baseline away from the optical axis. For simplicity, think of this rotation occurring in the plane that contains the optical axis and $\mathbf{e}(\theta_0)$, so that the epipole $\mathbf{e}(\theta)$ moves along the line ℓ between the principal point $\boldsymbol{\pi}_0$ of a and $\mathbf{e}(\theta_0)$.

Since the epipole is always in the image plane, its third coordinate is 1, and we have

$$\mathbf{e}(\theta) = \begin{bmatrix} e_x(\theta) \\ e_y(\theta) \\ 1 \end{bmatrix} = \begin{bmatrix} h(\theta)e_{x0} \\ h(\theta)e_{y0} \\ 1 \end{bmatrix}$$

[3] Since equation (3) is homogeneous, if E is an essential matrix then so is αE for any nonzero α. Therefore, the common magnitude of the two nonzero singular values is arbitrary.

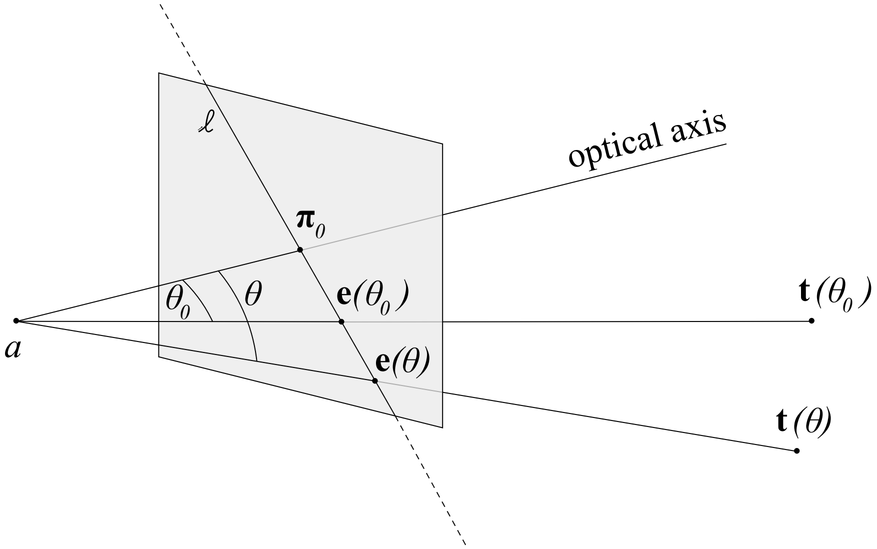

Figure 2: When the angle θ between the optical axis of camera a and the baseline approaches $\frac{\pi}{2}$, the baseline (that is, the line through a and along the translation vector $\mathbf{t}(\theta)$) becomes more and more parallel to the image plane of camera a, and the epipole $\mathbf{e}(\theta)$ tends to the point at infinity of the line ℓ through the principal point $\boldsymbol{\pi}_0$ and $\mathbf{e}(\theta_0)$.

where $h(\theta)$ is an increasing function of θ. When θ tends to $\pi/2$, the baseline becomes parallel to the image plane of camera a. The scalar $h(\theta)$ tends to infinity, and the epipole moves infinitely far away from $\boldsymbol{\pi}_0$.

However, since the third right singular vector $\mathbf{v}_3(\theta)$ of the essential matrix has unit norm, it represents the epipole $\mathbf{e}(\theta)$—and the translation $\mathbf{t}(\theta)$—only up to a constant. More specifically,

$$\mathbf{v}_3(\theta) = \frac{\mathbf{e}(\theta)}{\|\mathbf{e}(\theta)\|} = \frac{1}{\sqrt{1 + h^2(\theta)\,(e_{x0}^2 + e_{y0}^2)}} \begin{bmatrix} h(\theta)e_{x0} \\ h(\theta)e_{y0} \\ 1 \end{bmatrix}$$

and we immediately see that

$$\lim_{\theta \to \pi/2} \mathbf{v}_3(\theta) = \frac{1}{\sqrt{e_{x0}^2 + e_{y0}^2}} \begin{bmatrix} e_{x0} \\ e_{y0} \\ 0 \end{bmatrix} ,$$

a unit-norm vector as expected.

Thus, a singular vector $\mathbf{v}_3$ that has a third component equal to zero can be viewed as pointing to an epipole $\mathbf{e}$ that is the point at infinity on the line ℓ. Since $\mathbf{t}$ is proportional to $\mathbf{v}_3$ as well, we see that $\mathbf{t}(\frac{\pi}{2})$ is also parallel to the image plane, consistently with the fact that for $\theta = \frac{\pi}{2}$ camera b is to the side of camera a, that is, in the plane $z = 0$ in the reference system of camera a.

In summary, the solution $\mathbf{e}$ or $\mathbf{t}$ provided by $\mathbf{v}_3$ is correct even when the baseline is parallel to the image plane, as long as the epipole $\mathbf{e}$ is then interpreted as a point at infinity on the image plane of camera a.

Switching Cameras. Suppose now that we fix $\mathbf{a}$ in image I_a but replace $\mathbf{b}$ by a varying vector in I_b. Then we can repeat all the considerations above for the left null space and the left row space of E. In particular, the product $E\mathbf{a}$ for fixed $\mathbf{a}$ is a column vector, and equation (3) becomes the equation of the epipolar line in image I_b. The third *left* singular vector $\mathbf{u}_3$ of E is the direction of the epipole $\mathbf{e}_a$ in I_b *in the reference frame of camera b*. Rather than showing this through a separate argument, we prove that E^T is the essential matrix that would be obtained if the roles of cameras a and b were reversed.

To this end, Table 1 shows the results both ways using full subscripts, to make sure we do not confuse the two reference systems. To justify these results in the reverse direction, we then need to show that

$$ {}^aE_b^T = {}^bE_a \; , $$

that is, that transposing one essential matrix yields the essential matrix in the opposite direction. This result is a straightforward consequence of the invariance of the cross product to rotation,

$$ (R\mathbf{x}) \times (R\mathbf{y}) = R\,(\mathbf{x} \times \mathbf{y}) $$

which can be restated as follows for cross-product matrices, thinking of $\mathbf{x}$ as fixed and $\mathbf{y}$ as variable:

$$ [R\mathbf{x}]_\times R = R\,[\mathbf{x}]_\times \; . \tag{8} $$

Because $[{}^a\mathbf{t}_b]_\times$ is skew-symmetric,

$$ {}^aE_b^T = ({}^aR_b\;[{}^a\mathbf{t}_b]_\times)^T = -[{}^a\mathbf{t}_b]_\times\,{}^aR_b^T \; . $$

From our discussion of rigid transformations, we also know that if

$$ {}^b\mathbf{p} = {}^aR_b({}^a\mathbf{p} - {}^a\mathbf{t}_b) $$

then

$$ {}^a\mathbf{p} = {}^bR_a({}^b\mathbf{p} - {}^b\mathbf{t}_a) \quad \text{where} \quad {}^aR_b = {}^bR_a^T \quad \text{and} \quad {}^a\mathbf{t}_b = -{}^bR_a{}^b\mathbf{t}_a \; . $$

Therefore,

$$ {}^aE_b^T = [{}^bR_a{}^b\mathbf{t}_a]_\times\,{}^bR_a $$

and from equation (8)

$$ {}^aE_b^T = {}^bR_a\;[{}^b\mathbf{t}_a]_\times = {}^bE_a $$

as promised.

Use of the Epipolar Constraint. The epipolar constraint (3) is repeated here for convenience, using full notation for the essential matrix:

$$ \mathbf{b}^T\,{}^aE_b\,\mathbf{a} = 0 $$

where $\mathbf{a}$ and $\mathbf{b}$ are corresponding points. This constraint is used in two different contexts. In stereo vision, ${}^aR_b$ and ${}^a\mathbf{t}_b$ and therefore ${}^aE_b$ are known. Given a point $\mathbf{a}$ in I_a, the epipolar constraint then allows restricting the search for a corresponding point $\mathbf{b}$ to the epipolar line of $\mathbf{a}$.

In visual reconstruction, on the other hand, several pairs $(\mathbf{a}_i, \mathbf{b}_i)$ of corresponding points are given, and ${}^aE_b$ is unknown. Equation (3) for each pair of points yields a linear equation in the entries of ${}^aE_b$. From this, ${}^aE_b$ and then ${}^aR_b$ and ${}^a\mathbf{t}_b$ can be found, as we will see in Section 3.

For two cameras a and b with nonzero baseline, let

$$ {}^b\mathbf{p} = {}^aR_b\,({}^a\mathbf{p} - {}^a\mathbf{t}_b) $$

be the coordinate transformation between points ${}^a\mathbf{p}$ in a and points ${}^b\mathbf{p}$ in b, and let

$$ {}^a\mathbf{p} = {}^bR_a\,({}^b\mathbf{p} - {}^b\mathbf{t}_a) \quad \text{with} \quad {}^aR_b = {}^bR_a^T \quad \text{and} \quad {}^a\mathbf{t}_b = -{}^bR_a\,{}^b\mathbf{t}_a $$

be the transformation in the reverse direction.
The *essential matrix* of the camera pair (a, b) is the matrix

$$ {}^aE_b = {}^aR_b\,[{}^a\mathbf{t}_b]_\times \quad \text{where} \quad [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} $$

and the essential matrix of the camera pair (b, a) is

$$ {}^bE_a = {}^aE_b^T \ . $$

The *epipole* ${}^a\mathbf{e}_b$ is the image of the center of projection of camera b in image I_a and the epipole ${}^b\mathbf{e}_a$ is the image of the center of projection of camera a in image I_b. They satisfy

$$ {}^aE_b\,{}^a\mathbf{e}_b = {}^bE_a\,{}^b\mathbf{e}_a = \mathbf{0} \quad \text{and also} \quad {}^aE_b\,{}^a\mathbf{t}_b = {}^bE_a\,{}^b\mathbf{t}_a = \mathbf{0} \ . $$

A point ${}^a\mathbf{p}_a$ in image I_a and its corresponding point ${}^b\mathbf{p}_b$ in image I_b, both written as 3D vectors in their camera's canonical reference system, satisfy the *epipolar constraint*

$$ {}^b\mathbf{p}_b^T\,{}^aE_b\,{}^a\mathbf{p}_a = 0 \ . $$

This equation can also be written as follows:

$$ \boldsymbol{\lambda}_b^T\ {}^a\mathbf{p}_a = \boldsymbol{\lambda}_a^T\ {}^b\mathbf{p}_b = 0 $$

where

$$ \boldsymbol{\lambda}_b = {}^bE_a\,{}^b\mathbf{p}_b \quad \text{and} \quad \boldsymbol{\lambda}_a = {}^aE_b\,{}^a\mathbf{p}_a $$

are the vectors of coefficients of the *epipolar line* of $\mathbf{p}_b$ in image I_a and that of $\mathbf{p}_a$ in image I_b respectively.
Up to a nonzero and otherwise arbitrary multiplicative constant, the singular value decomposition of ${}^aE_b$ is

$$ {}^aE_b \sim U\Sigma V^T = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix} \operatorname{diag}(1,1,0) \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^T $$

where

$$ \mathbf{v}_3 \sim {}^a\mathbf{e}_b \sim {}^a\mathbf{t}_b \quad \text{and} \quad \mathbf{u}_3 \sim {}^b\mathbf{e}_a \sim {}^b\mathbf{t}_a $$

and $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{v}_1$, $\mathbf{v}_2$ are any vectors for which U and V become orthogonal.

Table 1: Definition and properties of the essential matrix.

3 The Eight-Point Algorithm

This Section describes a method for computing estimates of the rigid transformation ${}^aT_b = ({}^aR_b, {}^a\mathbf{t}_b)$ between two cameras a and b and estimates of the coordinates ${}^a\mathbf{P}_1, \ldots, {}^a\mathbf{P}_n$ of a set of n points in the reference system of one of the two cameras from the n pairs $({}^a\mathbf{p}_{a,1}, {}^b\mathbf{p}_{b,1}), \ldots ({}^a\mathbf{p}_{a,n}, {}^b\mathbf{p}_{b,n})$ of noisy measurements of their corresponding images. The transformation ${}^aT_b$ is called camera *motion*, and the point coordinates ${}^a\mathbf{P}_1, \ldots, {}^a\mathbf{P}_n$ are collectively called the scene *structure*. The image points ${}^a\mathbf{p}_{a,i}$ and ${}^b\mathbf{p}_{b,i}$ are regarded as 3D points with their third coordinate equal to 1, the standard focal distance. This means that rather than measuring all distances in, say, meters or millimeters, they are all measured in units of focal distance.

The classic method described below is called the *eight-point* algorithm and is was invented by Hugh Christopher Longuet-Higgins in 1981 [3]. Its main goal is to find ${}^aT_b$. Triangulation, that is, the calculation of structure from the image points and ${}^aT_b$, is outlined in Appendix B.

To simplify notation in the manipulations that follow, we again let

$$\mathbf{a} = {}^a\mathbf{p}_a \quad , \quad \mathbf{b} = {}^b\mathbf{p}_b \quad , \quad \mathbf{A} = {}^a\mathbf{P} \quad , \quad \mathbf{B} = {}^b\mathbf{P} \quad , \quad R = {}^aR_b \quad , \quad \mathbf{t} = {}^a\mathbf{t}_b \ ,$$

adding a subscript to **a**, **b**, or **A** when necessary to distinguish different points.

Since cameras fundamentally measure angles, both structure and motion can be estimated only up to a common nonzero multiplicative scale factor. The resulting degree of freedom is eliminated by assuming that

$$\|\mathbf{t}\| = \text{ 1 focal distance} \ . \tag{9}$$

The method described below is often called the *eight-point algorithm*, because it requires a *minimum* of $n = 8$ pairs of corresponding image points. More than 8 point pairs are typically used for better noise rejection.

The epipolar constraint described in Section 2,

$$\mathbf{b}^T E \, \mathbf{a} = 0 \ ,$$

can be spelled out as follows:

$$e_{11}a_1b_1 + e_{12}a_2b_1 + e_{13}a_3b_1 + e_{21}a_1b_2 + e_{22}a_2b_2 + e_{23}a_3b_2 + e_{31}a_1b_3 + e_{32}a_2b_3 + e_{33}a_3b_3 \ = \ 0$$

where $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$, $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^T$, and

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \ .$$

For easier programmatic manipulation, this expression can in turn be packaged in the following form:

$$\mathbf{c}^T \boldsymbol{\eta} = 0 \quad \text{where} \quad \mathbf{c} = \mathbf{b} \otimes \mathbf{a} = \begin{bmatrix} b_1 \, \mathbf{a} \\ b_2 \, \mathbf{a} \\ b_3 \, \mathbf{a} \end{bmatrix} \tag{10}$$

is the Kronecker product[4] of $\mathbf{b}$ and $\mathbf{a}$, and

$$\boldsymbol{\eta} = \left[\begin{array}{ccccccccc} e_{11} & e_{12} & e_{13} & e_{21} & e_{22} & e_{23} & e_{31} & e_{32} & e_{33} \end{array} \right]^T$$

is the stack of entries in E read by rows. Equation (10) can be replicated n times, one per image point pair,

$$\mathbf{c}_m^T \boldsymbol{\eta} = 0 \quad \text{for} \quad m = 1, \ldots, n \ ,$$

to yield a linear system

$$C\boldsymbol{\eta} = \mathbf{0} \quad \text{where} \quad C = \left[\begin{array}{ccc} \mathbf{c}_1 & \cdots & \mathbf{c}_n \end{array} \right]^T$$

is an $n \times 9$ matrix. The homogeneous nature of this system reflects the fact that translation $\mathbf{t}$ and therefore the essential matrix E are defined up to a nonzero multiplicative scale factor. As we know from a previous note, to prevent the trivial solution $\boldsymbol{\eta} = \mathbf{0}$ and at the same time solve the system above in the least-squares sense to account for measurement inaccuracies, one computes

$$\boldsymbol{\eta} = \arg \min_{\|\boldsymbol{\eta}\|=1} \|C\boldsymbol{\eta}\| = \mathbf{v}_9 \quad \text{where} \quad C = U_C \Sigma_C V_C^T$$

is the Singular Value Decomposition (SVD) of C and $\mathbf{v}_9$ is the last column of V_C. The resulting vector $\boldsymbol{\eta}$ is then reshaped into an estimate E of the essential matrix.[5]

As we know, the null space of E is the one-dimensional space spanned by $\mathbf{t}$, which also spans the null space of the skew matrix $[\mathbf{t}]_\times$. So an estimate of $\mathbf{t}$ is

$$\mathbf{t}_{1,2} = \pm \mathbf{v}_3$$

where $\mathbf{v}_3$ is the last column of V in the SVD $E = U\Sigma V^T$ of E. The ambiguity in the sign of $\mathbf{t}$ will be resolved later.

Given $\mathbf{t}$, one can construct the skew matrix $[\mathbf{t}]_\times$, and then estimate R by solving the following *Procrustes problem* [1]:

$$E \approx R\,[\mathbf{t}]_\times \ . \tag{11}$$

where the approximation is in the Frobenius norm. That is,

$$R = \arg \min_R \|E - R\,[\mathbf{t}]_\times\|_F = \sqrt{\sum_{i,j} d_{ij}^2} \quad \text{where} \quad D = [d_{ij}] = E - R\,[\mathbf{t}]_\times \ .$$

Appendix A shows[6] that if E and $[\mathbf{t}]_\times$ were full rank, the solution to problem (11) would be

$$R = Q \det(Q) \quad \text{where} \quad Q = U_F V_F^T \quad \text{and} \quad F = U_F \Sigma_F V_F^T$$

[4] More generally, the *Kronecker product* of two matrices F and G where F is $m \times n$ is defined as follows:

$$F \otimes G = \left[\begin{array}{ccc} f_{11}G & \ldots & f_{1n}G \\ \vdots & & \vdots \\ f_{m1}G & \ldots & f_{mn}G \end{array} \right] .$$

[5] As we found out in Section 2, the two nonzero singular values of the essential matrix are equal to each other, and the matrix $\tilde{E}$ that satisfies this constraint and is closest to E in the Frobenius norm is $\tilde{E} = U\text{diag}([1, 1, 0])V^T$ where $E = U\Sigma V^T$ is the SVD of E. However, the singular values of $\tilde{E}$ are not needed in the computation that follows, so this correction is unnecessary.

[6] Let $A = E$ and $B = [\mathbf{t}]_\times$ in that proof, so that $p = n = 3$.

is the SVD of the 3 × 3 matrix

$$F = E\,[\mathbf{t}]_\times^T \ ,$$

and where the multiplication by $\det(Q)$ ensures that the resulting orthogonal matrix is a rotation. This multiplication is allowed, because $\mathbf{t}$, and therefore E, is defined up to a multiplicative nonzero constant. In particular, if E is an essential matrix then so is $-E$.

However, the two matrices E and $[\mathbf{t}]_\times$ have rank 2. Since their third singular value is therefore zero, the third singular vectors (both left and right) of these two matrices are defined up to a sign. Recall that the third right singular vector is the direction of the translation $\mathbf{t}$ from camera a to camera b in the reference frame of a. Similarly, the third left singular vector is the direction of the translation $\mathbf{s} = -R^T\mathbf{t}$ from camera b to camera a in the reference frame of b. Because of this sign ambiguity in the solution, the Procrustes problem has two solutions:

$$R_{1,2} = Q_{1,2}\det(Q_{1,2}) \quad \text{where} \quad Q_{1,2} = \boldsymbol{\alpha}_1\boldsymbol{\beta}_1^T + \boldsymbol{\alpha}_2\boldsymbol{\beta}_2^T \pm \boldsymbol{\alpha}_3\boldsymbol{\beta}_3^T$$

where

$$U_F = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_3 \end{bmatrix} \quad , \quad V_F = \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \boldsymbol{\beta}_3 \end{bmatrix} \ .$$

Combining the twofold ambiguity in $\mathbf{t}$ with that in R yields four solutions, each corresponding to a different essential matrix:

$$(\mathbf{t}, R_1)\ ,\ (-\mathbf{t}, R_2)\ ,\ (\mathbf{t}, R_2)\ ,\ (-\mathbf{t}, R_1)\ .$$

Appendix C shows that only one of these solutions places all reconstructed world points in front of both cameras. The correct solution can then be identified by computing structure for all four cases by triangulation, and choosing the one solution that enforces structure to be in front of both cameras. Allowing for reconstruction errors, a safer approach is to chose the solution with a *majority* of points in front of the camera. Appendices B and C show the details of this calculation and a separate HTML file shows Python code for 3D reconstruction with two cameras. This HTML file has links to the code and data used in two simple test of the algorithm.

References

[1] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[2] R. I. Hartley. Chirality. *International Journal of Computer Vision*, 26(1):41–61, 1998.

[3] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

Appendices

A Solving the Procrustes Problem

This proof is adapted from a classical text on matrix computations [1], and applies to any two matrices A and B of size $p \times n$ that encode two sets of n data points in p dimensions.

Theorem A.1. *Let corresponding columns of the two matrices $A, B \in \mathbb{R}^{p\times n}$ encode n pairs of corresponding points in $\mathbb{R}^p$ with $p \leq n$. The following algorithm finds an orthogonal matrix $Q \in \mathbb{R}^{p\times p}$ that minimizes the Frobenius norm of $\|A - QB\|_F$.*

$$\begin{aligned} C &= AB^T \\ [U, \Sigma, V] &= \text{svd}(C) \\ Q &= UV^T \end{aligned}$$

Proof. The trace $\text{tr}(C)$ of a matrix C is the sum of its diagonal entries, and from the definition of Frobenius norm of a matrix C,

$$\|C\|_F^2 = \sum_{i,j} c_{ij}^2 = \text{tr}(CC^T) \ .$$

Then,

$$\|A - QB\|_F^2 = \text{tr}[(A - QB)(A - QB)^T] = \text{tr}(AA^T) + \text{tr}(BB^T) - 2\,\text{tr}(AB^TQ^T)$$

where we used the fact that Q is orthogonal and that the trace of the sum of several matrices is the sum of their traces.

The first two terms in the right-hand side of the equation above do not depend on Q, so minimizing $\|A - QB\|_F^2$ is the same as maximizing $\text{tr}(AB^TQ^T)$. If

$$AB^T = U\Sigma V^T$$

is the SVD of AB^T, then we want to find the maximum of

$$\text{tr}(U\Sigma V^TQ^T) = \text{tr}(U\Sigma V^TQ^TUU^T) = \text{tr}(U\Sigma ZU^T) \quad \text{where} \quad Z = V^TQ^TU$$

is an orthogonal matrix. It is easy to verify that if matrix G has the same size as matrix F^T then

$$\text{tr}(FG) = \text{tr}(GF) \ ,$$

so that

$$\text{tr}(U\Sigma ZU^T) = \text{tr}(\Sigma ZU^TU) = \text{tr}(\Sigma Z) = \sum_{i=1}^{p} \sigma_i z_{ii} \ .$$

Since Z is the product of orthogonal matrices, it is itself orthogonal. The rows of orthogonal matrices have unit norm, so no entry in an orthogonal matrix can have magnitude greater than 1. So the sum in the last term above is maximized when

$$z_{11} = \ldots = z_{pp} = 1 \ ,$$

which occurs when Z is the $p \times p$ identity matrix I_p. So one solution is achieved when

$$Z = I_p \quad \text{that is,} \quad V^T Q^T U = I_p \quad \text{or} \quad Q^T = V U^T \; .$$

The last equation was obtained by multiplying the previous one by V on the left and by U^T on the right. Thus,

$$Q = U V^T$$

as promised.

If the matrix C is full rank (so that both A and B are full rank), then this is the only solution. Otherwise, this is just *a* solution, because some of the σ_i are zero, so the corresponding values z_{ii} do not matter. The case in which $\text{rank}(C) = p - 1$ is both simple and relevant to the eight-point algorithm. In that case, the null space of C has dimension 1, so the only ambiguity in U and V that pertains to the last singular value is the sign of its last singular vectors $\mathbf{u}_p$ and $\mathbf{v}_p$. Changing the sign of both vectors leaves the product UV^T unaltered, because

$$UV^T = \left[\begin{array}{ccc} \mathbf{u}_1 & \ldots & \mathbf{u}_p \end{array} \right] \left[\begin{array}{ccc} \mathbf{v}_1 & \ldots & \mathbf{v}_p \end{array} \right]^T = \sum_{i=1}^{p} \mathbf{u}_i \mathbf{v}_i^T \; .$$

So if UV^T is one solution, then the other one is

$$\left[\begin{array}{ccc} \mathbf{u}_1 & \ldots & -\mathbf{u}_p \end{array} \right] \left[\begin{array}{ccc} \mathbf{v}_1 & \ldots & \mathbf{v}_p \end{array} \right]^T$$

which is the same as

$$\left[\begin{array}{ccc} \mathbf{u}_1 & \ldots & \mathbf{u}_p \end{array} \right] \left[\begin{array}{ccc} \mathbf{v}_1 & \ldots & -\mathbf{v}_p \end{array} \right]^T \; .$$

Δ

B Approximate Triangulation

Triangulation is the process of computing the coordinates $\mathbf{A}$ of each point in space from its projections in the two images, given that the transformation $(R, \mathbf{t})$ between the two cameras is known. This Appendix shows a simple triangulation method obtained by solving the two projection equations for $\mathbf{A}$. In this derivation, the two image projections are represented by vectors $\mathbf{a}_2$ and $\mathbf{b}_2$, which are the coordinates of the two projections of $\mathbf{A}$ in the canonical *image* reference system. If the focal distances of the two cameras are f_a and f_b, then $\mathbf{a}_2$ and $\mathbf{b}_2$ relate to the coordinates $\mathbf{a}$ and $\mathbf{b}$ (which are measured in the canonical *camera* reference system) by

$$\mathbf{a} = f_a \left[\begin{array}{c} \mathbf{a}_2 \\ 1 \end{array} \right] \quad \text{and} \quad \mathbf{b} = f_b \left[\begin{array}{c} \mathbf{b}_2 \\ 1 \end{array} \right] \; .$$

There are four scalar projection equations (one for each point coordinate in the two images) in three unknowns (the coordinates of $\mathbf{A}$), so the resulting linear system in $\mathbf{A}$ is over-constrained. In this Appendix, this system is solved in the sense of least squares, by minimizing the norm of the discrepancy between the left-hand side and the right-hand side of this system. The least-squares solution is optimal when this discrepancy, called the *algebraic error*, is Gaussian and isotropic.

However, this is typically not the case: What is likely Gaussian and sometimes isotropic is the *image reprojection error*, that is, the norm of the difference between the measured image point coordinates $\mathbf{a}_2$ and $\mathbf{b}_2$ and the coordinates obtained by projecting the solution $\mathbf{A}$ onto the two images.

Because of this, the solution to triangulation given here is not optimal. However, the solution found by using Longuet-Higgins's algorithm is typically used to initialize *bundle adjustment*, a computation that refines both motion $(R, \mathbf{t})$ and structure $(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ to minimize the image reprojection error—a nonlinear function of the unknowns. As a consequence, the approximate triangulation method described here is typically adequate, both as an initializer for bundle adjustment and to resolve the sign ambiguity discussed in Appendix C.

The projection equations for each point $\mathbf{A}$ can be written as follows for the two cameras:

$$\mathbf{a}_2 = \frac{1}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \quad \text{and} \quad \mathbf{b}_2 = \frac{1}{\mathbf{k}^T(\mathbf{A}-\mathbf{t})} R_2(\mathbf{A}-\mathbf{t}) \quad \text{where} \quad R_2 = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \end{bmatrix}$$

and where $\mathbf{i}^T$, $\mathbf{j}^T$, $\mathbf{k}^T$ are the rows of the rotation matrix R. The vector $\mathbf{A} = (X, Y, Z)^T$ collects the unknown coordinates of the point in space. Multiplying each equation by the denominator in its right-hand side and rearranging terms yield the following over-constrained 4×3 system of linear equations in $\mathbf{A}$:

$$\left[\begin{array}{c|c} I & -\mathbf{a}_2 \\ \hline \multicolumn{2}{c}{\mathbf{b}_2\,\mathbf{k}^T - R_2} \end{array}\right] \mathbf{A} = \left[\begin{array}{c} \mathbf{0} \\ \hline (\mathbf{b}_2\mathbf{k}^T - R_2)\mathbf{t} \end{array}\right]$$

where I is the 2×2 identity matrix and $\mathbf{0}$ is a column vector with two zeros. The solution $\mathbf{A}$ to this system can be found by the Least Squares method, and $\mathbf{B}$ can be computed by transforming $\mathbf{A}$ to the reference system of camera b:

$$\mathbf{B} = R\,(\mathbf{A} - \mathbf{t})\ .$$

This procedure is to be repeated for each of the image-point pairs.

C Resolving the Sign Ambiguity

Because of the sign ambiguity in $\mathbf{s}$ and $\mathbf{t}$, the Procrustes problem has two solutions:

$$R_{1,2} = W_{1,2}\det(W_{1,2}) \quad \text{where} \quad W_{1,2} = \boldsymbol{\alpha}_1\boldsymbol{\beta}_1^T + \boldsymbol{\alpha}_2\boldsymbol{\beta}_2^T \pm \mathbf{s}\mathbf{t}^T$$

where

$$U_B = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & -\mathbf{s} \end{bmatrix} \quad , \quad V_B = \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \mathbf{t} \end{bmatrix}\ .$$

Equivalently, if U_B and V_B are first replaced by their rotation versions $U_B\det(U_B)$ and $V_B\det(V_B)$ (so that their determinants are equal to 1), we have

$$R_1 = \boldsymbol{\alpha}_1\boldsymbol{\beta}_1^T + \boldsymbol{\alpha}_2\boldsymbol{\beta}_2^T - \mathbf{s}\mathbf{t}^T \quad \text{and} \quad R_2 = -\boldsymbol{\alpha}_1\boldsymbol{\beta}_1^T - \boldsymbol{\alpha}_2\boldsymbol{\beta}_2^T - \mathbf{s}\mathbf{t}^T\ . \tag{12}$$

These equations reveal that R_1 and R_2 relate to each other through a 180-degree rotation of either camera reference system around the baseline. To see this, write the transformation between these two frames of reference as a transformation from frame 1 to the world frame composed with one from world frame to frame 2:

$$R_2 R_1^T = (-\boldsymbol{\alpha}_1 \boldsymbol{\beta}_1^T - \boldsymbol{\alpha}_2 \boldsymbol{\beta}_2^T - \mathbf{s}\mathbf{t}^T)(\boldsymbol{\beta}_1 \boldsymbol{\alpha}_1^T + \boldsymbol{\beta}_2 \boldsymbol{\alpha}_2^T - \mathbf{s}\mathbf{t}^T) = -\boldsymbol{\alpha}_1 \boldsymbol{\alpha}_1^T - \boldsymbol{\alpha}_2 \boldsymbol{\alpha}_2^T + \mathbf{s}(\mathbf{s})^T \ ,$$

and this rotation maps $\boldsymbol{\alpha}_1$ to $-\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ to $-\boldsymbol{\alpha}_2$, and $\mathbf{s}$ (or $\mathbf{t}$) to itself, as promised.

The transformation between the first and the last of the four solutions above places camera 2 on the opposite side of camera 1 along the baseline.[7] This transformation can equivalently described as leaving the cameras where they are, pointing in the same way, but replacing all structure vectors $\mathbf{A}_i$ and $\mathbf{B}_i$ by their opposites $-\mathbf{A}_i$ and $-\mathbf{B}_i$. This transformation is said to change the *chirality* of structure in the literature [2], because superposing the original structure with the transformed one requires a change of handedness of the reference system (that is, a mirror flip). This transformation has the effect of placing the scene *behind* the two cameras if it is in front of them to begin with. With some abuse of terminology, a change of chirality in computer vision means merely changing whether structure is in front or behind a camera. In this sense, structure has two values of chirality, one per camera. A 180-degree rotation around the baseline—obtained by replacing R_1 with R_2 or *vice versa*—changes chirality once more, but only for the camera being rotated.

The four motion solutions given earlier correspond to using top right, top left, bottom right, and bottom left camera pairs in Figure 3, in this order. The two top pairs in the figure are said to form a *twisted pair*, and so are the two bottom pairs.

Only one of these solutions puts the scene points in front of both cameras. So the correct solution can be identified by computing structure for all four cases by triangulation, as shown in Appendix B, and choosing the one solution that enforces most of the structure solution (allowing for a few reconstruction errors) to be in front of both cameras:

$$\mathbf{e}_3^T \mathbf{A}_i > 0 \quad \text{and} \quad \mathbf{e}_3^T \mathbf{B}_i > 0 \quad \text{for} \quad i = 1, \ldots, n \quad \text{where} \quad \mathbf{e}_3^T = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right] \ .$$

[7] Of course, the same transformation can be described as a displacement of camera 1 relative to camera 2.

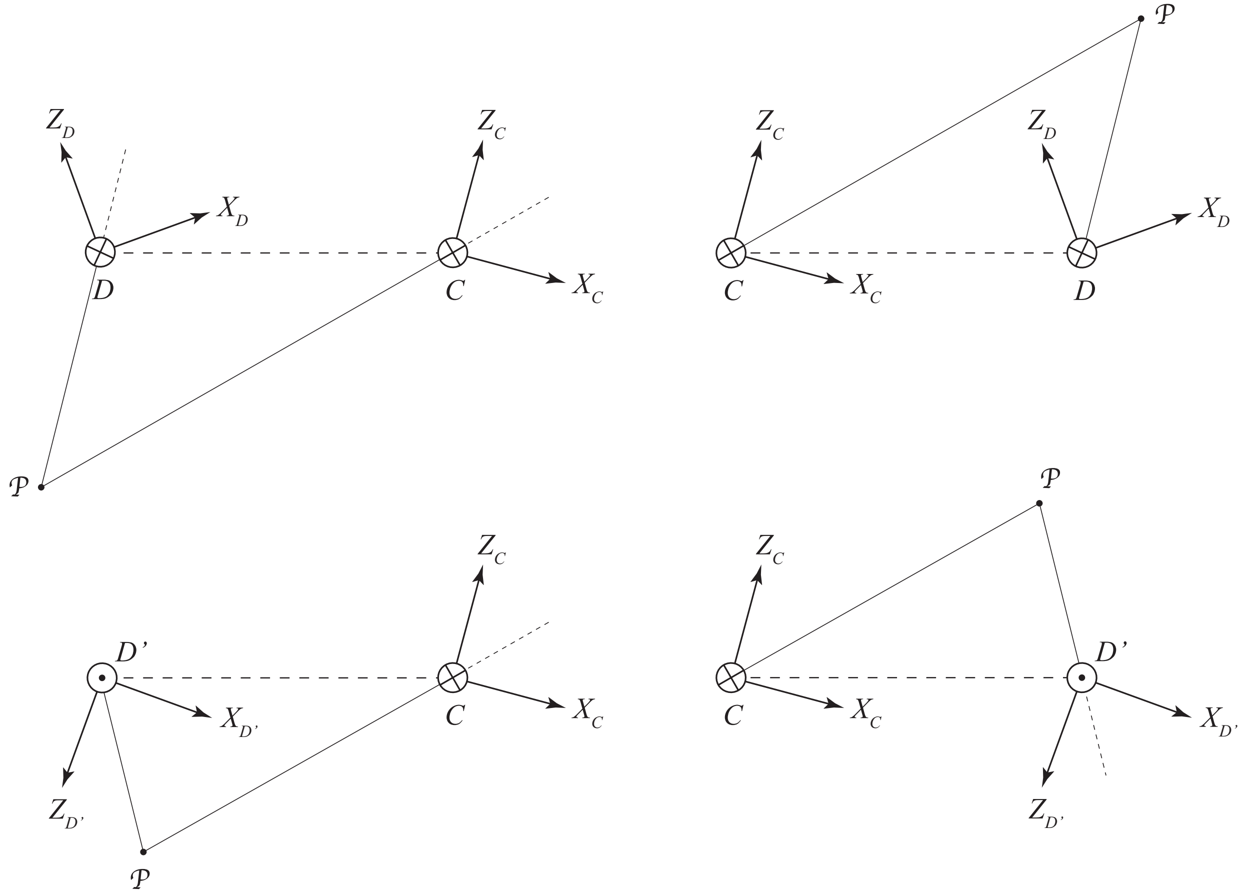

Figure 3: The fourfold ambiguity of reconstruction corresponds to the two ways to pick the sign of $\mathbf{t}$ (left or right diagrams) and the two ways to choose the rotation matrix R (top or bottom diagrams). A circle with a cross (a dot) denotes a Y axis pointing into (out of) the page. Only the arrangement in the top right has the scene structure (represented by the single point $\mathbf{P}$ and its two projection rays) in front of both cameras.