Epipolar lines calculation

Tim Lenertz

October 6, 2016

1 Contents

- Extrinsic and intrinsic camera matrices
- Projection matrices with different depth conventions
- Homography matrix (gives point on epipolar line)
- Essential and fundamental matrices
- Calculation of epipolar line equation

2 Camera matrices

Camera parameters consist of two matrices, *extrinsic matrix* $\mathbf{Rt}$ and *intrinsic matrix* $\mathbf{K}$.

2.1 Extrinsic matrix $\mathbf{Rt}$

Extrinsic matrix is the pose (position & orientation) of the camera in world space. It is a rigid transformation matrix, consisting of a rotation $\mathbf{R}$ and translation $\vec{t}$. It is 4×4 matrix in homogeneous coordinates.

$$\mathbf{Rt} = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & t_0 \\ r_{1,0} & r_{1,1} & r_{1,2} & t_1 \\ r_{2,0} & r_{2,1} & r_{2,2} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$\vec{\tilde{x}} = \mathbf{Rt}\,\vec{\tilde{w}}$ (in homogeneous coordinates) is equivalent to $\vec{x} = \mathbf{R}\,\vec{w} + \vec{t}$.

$\vec{w}$ is 3D point in world space (global coordinate system). $\vec{x}$ is the same point in view space for one camera, where the camera center is at origin, and either $+\vec{z}$ or $-\vec{z}$ point in camera view direction. (depends on convention used)

$$\begin{bmatrix} wx_0 \\ wx_1 \\ wx_2 \\ w \end{bmatrix} = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & t_0 \\ r_{1,0} & r_{1,1} & r_{1,2} & t_1 \\ r_{2,0} & r_{2,1} & r_{2,2} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ 1 \end{bmatrix}$$
$$\vec{\hat{x}} = \mathbf{Rt}\,\vec{\hat{w}} \qquad (2)$$

2.1.1 MPEG convention

The extrinsic matrices in DERS and VSRS configuration files use a different convention: Instead the matrix given as

$$\begin{bmatrix} r'_{0,0} & r'_{0,1} & r'_{0,2} & t'_0 \\ r'_{1,0} & r'_{1,1} & r'_{1,2} & t'_1 \\ r'_{2,0} & r'_{2,1} & r'_{2,2} & t'_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

Represents the transformation $x = \mathbf{R}'(\vec{w} - \vec{t'})$.

To convert it into $\mathbf{R}, \vec{t}$ such that $\vec{x} = \mathbf{R}\vec{w} + \vec{t}$:

$$\mathbf{R} = \mathbf{R}' \qquad \vec{t} = -(\mathbf{R}\vec{t'}) \qquad (4)$$

2.2 Intrinsic matrix $\mathbf{K}$

Intrinsic matrix (or camera matrix) is projection from view space to pixel coordinates on the image. In pin-hole camera model, the mapping is

$$u = \frac{f_x x_0}{x_2} + t_x \qquad v = \frac{f_x x_1}{x_2} + t_y \qquad (5)$$

where $\vec{p} = (u, v)$ are pixel coordinates, $\vec{x} = (x_0, x_1, x_2)$ is 3D point in view space. (f_x, f_y) are focal lengths (usually same), and (t_x, t_y) are offsets of center point in image. Focal lengths and offsets are adjusted to image pixel size.

Intrinsic matrix $\mathbf{K}$ is 3×3 in homogeneous coordinates:

$$\vec{p} = \mathbf{K}\,\vec{x}$$
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & t_x \\ 0 & f_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \qquad (6)$$

The transformation loses depth information, and $\mathbf{K}$ is non-invertible (singular).

3 Projection matrix $\mathbf{P}$

Projection matrix $\mathbf{P}$ is 4×4 matrix derived from the intrinsic matrix $\mathbf{K}$, which keeps depth. The projected depth d is a hyperbolic function of x_2. (= The z component of view space point vector, i.e. the distance to the camera center, orthogonal to its image plane.)

Two distances $\mathrm{z_{near}}$ and $\mathrm{z_{far}}$ are defined such that for all objects of interest in the image, $\mathrm{z_{near}} < x_2 < \mathrm{z_{far}}$. (Assuming x_2 increases in camera view direction.)

Projection matrices with different conventions for depth projection are possible:

3.1 Unsigned normalized disparity

$$\vec{p} = \mathbf{P}\,\vec{x}$$
$$\begin{bmatrix} wu \\ wv \\ wd \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & t_x & 0 \\ 0 & f_y & t_y & 0 \\ 0 & 0 & -\frac{\mathrm{z_{near}}}{\mathrm{z_{far}} - \mathrm{z_{near}}} & \frac{\mathrm{z_{near}}\,\mathrm{z_{far}}}{\mathrm{z_{far}} - \mathrm{z_{near}}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \tag{7}$$

Maps the depth such that $x_2 = \mathrm{z_{near}} \rightsquigarrow d = 1$ and $x_2 = \mathrm{z_{far}} \rightsquigarrow d = 0$.

This is compatible with the depth maps of some test sequences (Poznan Blocks and Bunny), when $\mathrm{z_{near}}$ and $\mathrm{z_{far}}$ are correctly set, and the range $0 \dots 255$ in the depth map is linearly mapped to $0 \dots 1$.

3.2 Signed normalized depth

$$\begin{bmatrix} f_x & 0 & t_x & 0 \\ 0 & f_y & t_y & 0 \\ 0 & 0 & \frac{\mathrm{z_{near}} + \mathrm{z_{far}}}{\mathrm{z_{far}} - \mathrm{z_{near}}} & -\frac{2\,\mathrm{z_{near}}\,\mathrm{z_{far}}}{\mathrm{z_{far}} - \mathrm{z_{near}}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{8}$$

Maps the depth such that $x_2 = \mathrm{z_{near}} \rightsquigarrow d = -1$ and $x_2 = \mathrm{z_{far}} \rightsquigarrow d = +1$.

It is compatible with projection matrices used by OpenGL, but additionally the scaling and offsets must be adjusted so that u, v and fall in $-1 \cdots +1$.

3.3 Unsigned normalized depth

$$\begin{bmatrix} f_x & 0 & t_x & 0 \\ 0 & f_y & t_y & 0 \\ 0 & 0 & \frac{\mathrm{z_{near}} + \mathrm{z_{far}}}{\mathrm{z_{far}}} & -\frac{\mathrm{z_{near}}\,\mathrm{z_{far}}}{\mathrm{z_{far}} - \mathrm{z_{near}}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

Maps the depth such that $x_2 = \mathrm{z_{near}} \rightsquigarrow d = 0$ and $x_2 = \mathrm{z_{far}} \rightsquigarrow d = +1$. (But not the same as simply reversing d from *unsigned normalized disparity*).

4 Homography

Unlike the intrinsic matrix $\mathbf{K}$, the projection matrix $\mathbf{P}$ is invertible. With this homography between images with depth can be calculated.

In homogeneous coordinates:

$$\vec{\hat{p}} = \mathbf{P}\,\mathbf{Rt}\,\vec{\hat{w}} \tag{10}$$

and so

$$\vec{\hat{w}} = (\mathbf{P}\,\mathbf{Rt})^{-1}\,\vec{\hat{p}} = \mathbf{Rt}^{-1}\,\mathbf{P}^{-1}\,\vec{\hat{p}} \tag{11}$$

4.1 Homography matrix $\mathbf{H}$

Let A and B be two camera views, with extrinsic and projection matrices $(\mathbf{Rt}_A, \mathbf{P}_A)$ and $(\mathbf{Rt}_B, \mathbf{P}_B)$. $\mathrm{z}_{\mathrm{near}}$, $\mathrm{z}_{\mathrm{far}}$ and the depth projection conventions are chosen for A and B.

Let $\vec{p_A} = (u_A, v_A, d_A)$ be the coordinates of a pixel in A, and its projected depth. The corresponding pixel coordinates $\vec{p_B} = (u_B, v_B, d_B)$ in B are calculated, in homogeneous coordinates, using:

$$\vec{\hat{p_B}} = \mathbf{H}_{A\mapsto B}\,\vec{\hat{p_A}}. \tag{12}$$

with

$$\mathbf{H}_{A\mapsto B} = \mathbf{P_B}\,\mathbf{Rt_B}\,\mathbf{Rt_A^{-1}}\,\mathbf{P_A^{-1}} \tag{13}$$

is the homography matrix from A to B. It is a 4×4 matric in homogeneous coordinates.

When d_A is correctly set and there is no occlusion, (u_B, v_B) will fall on the same scene object on B as (u_A, v_A) does in A. The homography matrix is used for image warping in VSRS.

When (u_A, v_A) is fixed and d_A is varied, (u_B, v_B) moves along the epipolar line in B.

5 Essential matrix $\mathbf{E}$

Let A and B be two camera views, with extrinsic matrices $\mathbf{Rt_A}$ and $\mathbf{Rt_B}$.

Let $\mathbf{M} = \mathbf{Rt_A}\,\mathbf{Rt_B^{-1}}$. (Rigid transformation from coordinate system of B into coordinate system of A).

$\mathbf{M}$ always consists of a rotation matrix $\mathbf{R}$ and translation vector $\vec{t}$:

$$\mathbf{M} = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & t_0 \\ r_{1,0} & r_{1,1} & r_{1,2} & t_1 \\ r_{2,0} & r_{2,1} & r_{2,2} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Let

$$\mathbf{T} = \begin{bmatrix} 0 & -t_2 & t_1 \\ t_2 & 0 & -t_0 \\ -t_1 & t_0 & 0 \end{bmatrix} \tag{15}$$

The essential matrix $\mathbf{E}_{A\mapsto B}$from A to B is

$$\mathbf{E}_{A\mapsto B} = \mathbf{T}\,\mathbf{R} \tag{16}$$

6 Fundamental matrix F

Let A and B be two camera views, with intrinsic matrices $\mathbf{K_A}$ and $\mathbf{K_B}$.

The fundamental matrix $\mathbf{F}_{A\mapsto B}$ from A to B is

$$\mathbf{F}_{A\mapsto B} = \left(\mathbf{K_A^{-1}}\right)^\intercal\ \mathbf{E}_{A\mapsto B}\,\mathbf{K_B^{-1}} \tag{17}$$

7 Epipolar line equation

For a given pixel position (u_A, v_A), the line equation $v_B = f(u_B)$ of the epipolar line in B is:

$$v_B = \frac{-\mathbf{F}_{2,2} - \mathbf{F}_{0,2}\,u_A - \mathbf{F}_{2,0}\,u_B - \mathbf{F}_{0,0}\,u_A\,u_B - \mathbf{F}_{1,2}\,v_A - \mathbf{F}_{1,0}\,u_B\,v_A}{\mathbf{F}_{2,1} + \mathbf{F}_{0,1}\,u_A + \mathbf{F}_{1,1}\,v_A} \tag{18}$$