Estimation of *F* or *E*

The technique is similar to that of calibrating the perspective projection matrix ***P***.

Denote the homogeneous coordinates of left and right pixels by

$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1)^T$$

For one pair of image correspondence

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

Since we want to find F_{ij}, rearrange the previous equation to

$$\Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

This is a homogeneous equation; can only solve up to a scale. Hence, set, say, $F_{33} = 1$.

$$(uu', uv', u, vu', vv', v, u', v') \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = -1$$

Concatenate eight correspondence pairs :

$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

With 8 unknowns in 8 equations, theoretically all the $\boldsymbol{F}_{ij}$ are found. For robustness, use more image pairs and then apply the technique of pseudo-inverse or SVD (see previous notes).

Motion Recovery from *E*

If we want to recover ***R*** and ***t*** from ***E***, how can this be done ? This applies to recovering the motion of a camera.

And with a slight modification to the formulation, we can also apply this method to recover the motion of a translating/rotating object or camera. First solve for E from at least eight tracked image points. Then recover ***R*** and ***t*** from ***E*** [Longguet-Higgins].

[K Kanatani. "Geometric Computation for machine vision". Oxford Press, 1993].

Determining Translation from Essential Matrix

Assuming Essential matrix E has been obtained.

First, determine **t** (up to a scale) from

$$\mathbf{E}^T\mathbf{t} = \mathbf{0}$$

$$p_r^T \underbrace{E\, p_\ell}_{=0} = 0$$

There are two solutions **t** and **-t**

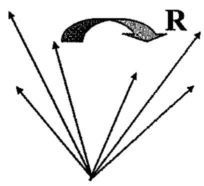
$$E = RS, \quad S = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix} \quad t = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}$$

$$R = ES^{-1}$$

$$RE^T = S^{-1}$$

$$R\,[e_1\ e_2\ e_3] = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix},$$

Next, find the rotation matrix **R**.



$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

$$\mathbf{E}\mathbf{R}^{-1} = [\mathbf{t}]_\times \mathbf{R}\mathbf{R}^{-1}$$

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

$$\mathbf{R}\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix} = \begin{bmatrix}\mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3\end{bmatrix}^{\mathrm{T}}$$

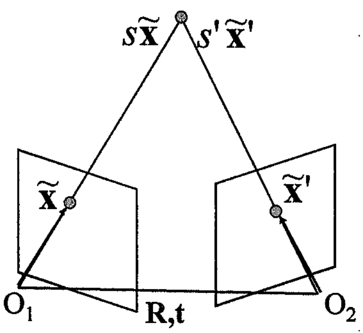
$$\begin{bmatrix}\mathbf{r}_i^{T}\mathbf{e}_1 & \mathbf{r}_i^{T}\mathbf{e}_2 & \mathbf{r}_i^{T}\mathbf{e}_3\end{bmatrix} = \mathbf{t}_{\mathrm{i}}^{\mathrm{T}} \quad \text{for } i = 1, 2 \text{ and } 3$$

We can consider the problem as equivalent to determining **R** from 3 pairs of vectors before and after the rotation. This linear method may not be accurate. There are other non-linear methods of estimating **F** or **E**.

Recovery of the 3D Points

Finally, after having found **R** and **t**, for each image pair, **x** and **x**', one can determine how far the world point is from the cameras. Let s and s' be the scales such that the two lines "meet" in the 3D space.

$$sx = s'\mathbf{R}x' + t$$



There will be four possible solutions.

- s>0, s'>0 in front of both cameras
- s>0, s'<0
- s<0, s'>0
- s<0, s'<0

Choose the one in front of the camera.

Recall :

And so $\tilde{u} \cong \mathbf{M}_{\text{int}} \tilde{x}_1$ and $\tilde{x}_1 = \begin{bmatrix} R & t \\ 0_{3\times1} & 1 \end{bmatrix} \tilde{x}_W$

$$\tilde{\mathbf{u}} \cong \mathbf{M}_{\text{int}} \mathbf{M}_{ext} \tilde{\mathbf{x}}_W$$

$$\begin{aligned} \boldsymbol{p}_R^T \mathrm{F} \boldsymbol{p}_L &= \boldsymbol{x}_R^T \mathbf{M}_{\text{int}R}^T \mathrm{F} \mathbf{M}_{\text{int}L}^T \boldsymbol{x}_L \\ &= \boldsymbol{x}_R^T (\mathbf{M}_{\text{int}R}^T \mathrm{F} \mathbf{M}_{\text{int}L}^T) \boldsymbol{x}_L \\ &= \boldsymbol{x}_R^T \mathbf{E} \boldsymbol{x}_L \end{aligned}$$

$$\mathbf{E} = \mathbf{M}_{\text{int}R}^T \mathrm{F} \mathbf{M}_{\text{int}L}^T$$

Here $\boldsymbol{p}_R$ and $\boldsymbol{p}_L$ are the pixel coordinates (instead of $\boldsymbol{u}_R$ and $\boldsymbol{u}_L$).

Homography

Planar homography If the scene is from a plane, then the camera model takes on a special form known as a homography.

Plane seen in first camera coordinate frame

$$aX + bY + cZ + d = 0$$

$$\tfrac{1}{d} N^T \mathrm{X} = 1$$

$$\lambda_2 \mathrm{x}_2 = R \lambda_1 \mathrm{x}_1 + T$$

$$\lambda_2 \mathrm{x}_2 = (R + \tfrac{1}{d} T N^T) \lambda_1 \mathrm{x}_1$$

$$\boxed{\begin{aligned} \mathrm{x}_2 &\sim H \mathrm{x}_1 \\ H &= (R + \tfrac{1}{d} T N^T) \end{aligned}}$$