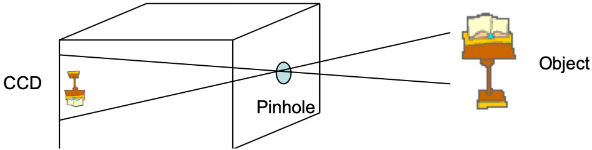
Computer vision: 3D Geometry Fundamentals

Richard Xu

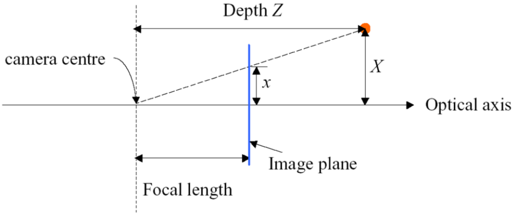
January 12, 2021

A Simple Camera Model



- It's rather odd to look at it upside down

Simpler Model



- It's rather odd to see an inverted model like this

How object location relates to an image point?

- Naturally:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- It's NOT helpful to lump the whole projection matrix into a single 3×4 matrix $\mathbf{P}$
- determine the 3D ray from 2D image point $\mathbf{x}$

$$\mathbf{X}_{\text{3D}}(\lambda) = \mathbf{P}^{+}\mathbf{x} + \lambda\mathbf{C}$$

$$\text{where } \mathbf{PP}^{+} = \mathbf{I}$$

Camera calibration

2D $\qquad\qquad\qquad\qquad$ 3D

$$s\,\mathbf{x} = \mathbf{K}\;[\mathbf{R}\,|\mathbf{t}]\;\mathbf{X}$$

4

$$\Longrightarrow s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

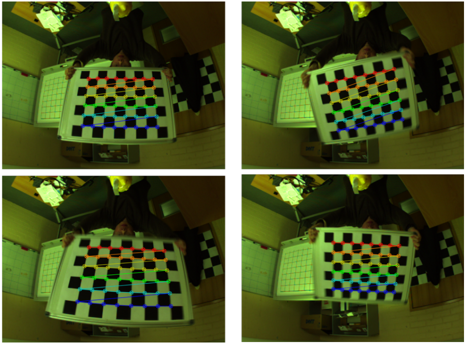
- Intrinsic parameter $\mathbf{K} = \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix}$
- Extrinsic parameter $[\mathbf{R}\,|\mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$

CHECKPOINT: Intrinsic Parameter Calibration

Intrinsic Parameter Calibration

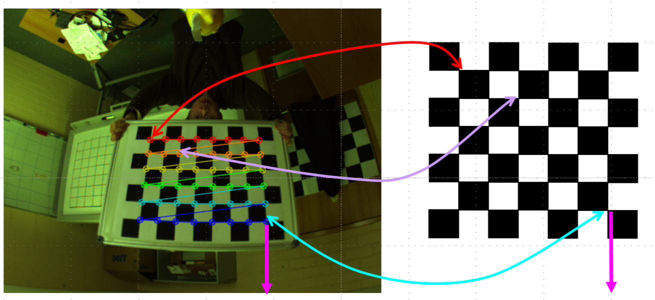
Intrinsic Camera calibration

Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," in International Conference on Computer Vision, 1999, pp. 666-673



-

Homography



$$\mathbf{x} \qquad\qquad \mathbf{M}$$

$$\mathbf{x} = \mathbf{H}\mathbf{M}$$

$$\begin{bmatrix} 34.12 \\ 65.21 \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix}$$ as an example

"Data" collection: use Homography $\mathbf{H}$ as data

- Homography $\mathbf{H}$ acts like our "data", because it can be computed **beforehand** without camera geometry
- let's define $\mathbf{M}$ to be $\mathbf{X}$ without z^{th} component

$$\mathbf{x} = \mathbf{H}\mathbf{M}$$
$$\underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\hat{\mathbf{x}}}$$

- Get 4 pair of points and we are done, yeah?
- Where is the catch? Image points have noises!

$$\sum_i \left[(\mathbf{x}_i - \hat{\mathbf{x}}_i)^\top \mathbf{\Lambda}^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i) \right]$$

- for simplicity, can just assume: $\mathbf{\Lambda} = \sigma^2 \mathbf{I}$

$$\min_{\mathbf{H}} \sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$$

Brings things to 3D

$$s\,\mathbf{x} = \mathbf{K}[\mathbf{r} \quad \mathbf{t}]\mathbf{X}$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- let's assume the board is a planar surface, and $z = 0$:

$$\begin{aligned} s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\ &= \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \mathbf{M} \end{aligned}$$

- obviousness, we need to re-arrange to **cancel** auxiliary variable **r** and **t**

Combine the two case together

- substitute $\mathbf{x} = \mathbf{H}\mathbf{M}$

$$
\begin{aligned}
s\,\mathbf{x} &= \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\mathbf{M} \\
&= \mathbf{H}\mathbf{M} \\
\implies \mathbf{H} &= \lambda\mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} \qquad \lambda = \frac{1}{s}
\end{aligned}
$$

- kept on going:

$$
\begin{aligned}
\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} &= \lambda\mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} \\
\implies \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2\end{bmatrix} &= \lambda\mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2\end{bmatrix} \qquad \text{we do not need } \mathbf{h}_3 \text{ and } \mathbf{t} \\
\implies \mathbf{h}_1^\top\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 &= 0 \quad (1) \\
\text{also} \implies \mathbf{h}_1^\top\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 &= \mathbf{h}_2^\top\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 \quad (2)
\end{aligned}
$$

- so **r** and **t** are completely disappeared

prove $\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0$ (1)

$$
\begin{aligned}
\mathbf{H} &= \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} \\
\implies \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} &= \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} \\
\mathbf{h}_1 &= \mathbf{K}\mathbf{r}_1 \implies \mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{K}\mathbf{r}_2 \implies \mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{h}_2 \\
\mathbf{r}_1^\top \mathbf{r}_2 &= \left(\mathbf{K}^{-1}\mathbf{h}_1\right)^\top \mathbf{K}^{-1}\mathbf{h}_2 \\
&= \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0
\end{aligned}
$$

- because rotation matrix $\mathbf{R}$ is orthogonal: $\mathbf{r}_i^\top \mathbf{r}_j = 0 \forall i \neq j$
- λ won't matter:

$$
\begin{aligned}
\mathbf{h}_1 = \lambda \mathbf{K}\mathbf{r}_1 &\implies \mathbf{r}_1 = \frac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h}_1 \\
\mathbf{h}_2 = \lambda \mathbf{K}\mathbf{r}_2 &\implies \mathbf{r}_2 = \frac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h}_2 \\
&\implies \frac{1}{\lambda^2}\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{r}_1^\top \mathbf{r}_1 &= \left(\mathbf{K}^{-1}\mathbf{h}_1\right)^\top \mathbf{K}^{-1}\mathbf{h}_1 \\
&= \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = 1
\end{aligned}
$$

similarly,

$$
\begin{aligned}
\mathbf{r}_2^\top \mathbf{r}_2 &= \left(\mathbf{K}^{-1}\mathbf{h}_2\right)^\top \mathbf{K}^{-1}\mathbf{h}_2 \\
&= \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = 1
\end{aligned}
$$

together:

$$
\implies \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2
$$

- again, because rotation matrix **R** is orthogonal

now you have a linear system

- a linear system:

$$
\begin{aligned}
&\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \\
&\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \\
\implies &\mathbf{h}_1^\top \mathbf{B}\mathbf{h}_2 = 0 \\
&\mathbf{h}_1^\top \mathbf{B}\mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{B}\mathbf{h}_2 = 0 \qquad \text{let: } \mathbf{B} = \mathbf{K}^{-\top}\mathbf{K}^{-1}
\end{aligned}
$$

- knowing $\mathbf{K} = \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix}$
- you can perform python code to get expression of $\mathbf{B} = \mathbf{K}^{-\top}\mathbf{K}^{-1}$

Solve for **B**

- notice **B** is symmetrical matrix, so there are only 6 degree-of-freedom

$$\begin{bmatrix} B_{1,1} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

- so we let $\mathbf{B} = \left[B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}\right]^\top$

$$\begin{aligned} \mathbf{h}_1^\top \mathbf{B} \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^\top \mathbf{B} \mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{B} \mathbf{h}_2 &= 0 \end{aligned} \qquad \text{can be written as:}$$

$$\begin{bmatrix} h_{11}h_{21} & h_{11}h_{22} + h_{12}h_{21} & h_{12}h_{22} & h_{11}h_{23} + h_{13}h_{21} & h_{13}h_{22} + h_{12}h_{23} & h_{13}h_{23} \\ h_{11}h_{11} - h_{21}h_{21} & 2h_{11}h_{12} - 2h_{21}h_{22} & h_{12}h_{12} - h_{22}h_{22} & 2h_{11}h_{13} - 2h_{21}h_{23} & 2h_{12}h_{13} - 2h_{22}h_{23} & h_{13}h_{13} - h_{23}h_{23} \end{bmatrix} \times \begin{bmatrix} B_{11} \\ B_{12} \\ B_{21} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- then you can solve for **K** from **B**

Extrinsic Parameter Calibration, aka Camera Pose

How to calibrate extrinsic

$$
s \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} — \mathbf{p}_1 — \\ — \mathbf{p}_2 — \\ — \mathbf{p}_3 — \end{bmatrix} \begin{bmatrix} | \\ \mathbf{X} \\ | \end{bmatrix}
$$

$$
= \begin{bmatrix} \mathbf{p}_1^\top \mathbf{X} \\ \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_3^\top \mathbf{X} \end{bmatrix}
$$

- convert $\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \to \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ in image point

$$
\Longrightarrow u = \frac{\mathbf{p}_1^\top \mathbf{X}}{\mathbf{p}_3^\top \mathbf{X}} \qquad v = \frac{\mathbf{p}_2^\top \mathbf{X}}{\mathbf{p}_3^\top \mathbf{X}}
$$

$$
\Longrightarrow \mathbf{p}_1^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} u = 0 \qquad \mathbf{p}_2^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} v = 0
$$

another system of linear equation

- single point:

$$\mathbf{p}_1^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} u = 0 \qquad \mathbf{p}_2^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} v = 0 \implies \begin{bmatrix} \mathbf{X}^\top & \mathbf{0} & -u\mathbf{X}^\top \\ \mathbf{0} & \mathbf{X}^\top & -v\mathbf{X}^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \mathbf{0}$$

- N points:

$$\underbrace{\begin{bmatrix} \mathbf{X}_1^\top & \mathbf{0} & -u\mathbf{X}_1^\top \\ \mathbf{0} & \mathbf{X}_1^\top & -v\mathbf{X}_1^\top \\ \vdots & \vdots & \vdots \\ \mathbf{X}_N^\top & \mathbf{0} & -u\mathbf{X}_N^\top \\ \mathbf{0} & \mathbf{X}_N^\top & -v\mathbf{X}_N^\top \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^\top & \mathbf{0} & -u\mathbf{X}_1^\top \\ \mathbf{0} & \mathbf{X}_1^\top & -v\mathbf{X}_1^\top \\ \vdots & \vdots & \vdots \\ \mathbf{X}_N^\top & \mathbf{0} & -u\mathbf{X}_N^\top \\ \mathbf{0} & \mathbf{X}_N^\top & -v\mathbf{X}_N^\top \end{bmatrix} \begin{bmatrix} p_{1,1} \\ p_{1,2} \\ p_{1,3} \\ p_{1,4} \\ p_{2,1} \\ p_{2,2} \\ p_{2,3} \\ p_{2,4} \\ p_{3,1} \\ p_{3,2} \\ p_{3,3} \\ p_{3,4} \end{bmatrix} = \mathbf{0}$$

Solve this

if we to solve:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2$$

- most obvious solution is $\mathbf{p} = 0$!
- so we need a constraint, imagine let $\|\mathbf{P}\|_F = s$, i.e., Frobenius norm $= s$

$$s \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\implies s \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} sp_{1,1} & sp_{1,2} & sp_{1,3} & sp_{1,4} \\ sp_{2,1} & sp_{2,2} & sp_{2,3} & sp_{2,4} \\ sp_{3,1} & sp_{3,2} & sp_{3,3} & sp_{3,4} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- scale the matrix $\mathbf{P}$ by s won't change image points

why constraining $\|\mathbf{p}\| = 1$

- objective function:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2 \qquad \text{s.t. } \|\mathbf{p}\|^2 = 1$$

- imagine for a **vector p** s.t. $\|\mathbf{p}\| = 1$ and $\hat{\mathbf{p}} = s\mathbf{p}$
- we found the solution by constraining $\|\hat{\mathbf{p}}\| = s$:

$$\begin{aligned} \|s\mathbf{p}\| &= \sqrt{s\mathbf{p}^\top s\mathbf{p}} \\ &= s\|\mathbf{p}\| \end{aligned}$$

- **meaning**: constraining $\|\hat{\mathbf{p}}\| = s$ has the same effect of constraining $\|\mathbf{p}\| = 1$

Rayleigh quotient's view

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2 \qquad \text{s.t. } \|\mathbf{p}\|^2 = 1$$

$$\Longrightarrow \mathbf{p}^* = \arg\min_{\mathbf{p}} \left\| \mathbf{A} \frac{\mathbf{p}}{\|\mathbf{p}\|} \right\|^2 \quad \text{same as finding unconstrained } \mathbf{p}$$

$$= \arg\min_{\mathbf{p}} \left(\frac{\mathbf{p}^\top \mathbf{A}^\top \mathbf{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}} \right)$$

- a form of Rayleigh quotient:

$$R(M, x) := \frac{x^\top M x}{x^\top x} \qquad \text{where}$$

  - Rayleigh quotient reaches its min value:

$$R(M, x_{\text{min}}) = \lambda_{\text{min}}$$

    smallest eigenvalue of M, when $x = v_{\text{min}}$ the corresponding eigenvector.
  - Rayleigh quotient reaches its max value:

$$R(M, x_{\text{max}}) = \lambda_{\text{max}}$$

    largest eigenvalue of M, when $x = v_{\text{max}}$ the corresponding eigenvector.
- where have you seen this before?

from SVD perspective

$$
\begin{aligned}
\|\mathbf{A}\|_2^2 &= \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2^2 \\
& \sup_{\|\mathbf{x}\|_2=1} (\mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}) \\
&= \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^\top U \,\mathrm{diag}(\lambda_1, \dots, \lambda_n) U^\top \mathbf{x} \\
&= \max_{\|\mathbf{y}\|_2=1} \mathbf{y}^\top \mathrm{diag}(\lambda_1, \dots, \lambda_n)\mathbf{y} \qquad \text{since } U \text{ is orthogonal matrix } \|\mathbf{x}\|_2 = \|\underbrace{U\mathbf{x}}_{\mathbf{y}}\|_2 \\
&= \max_{\|\mathbf{y}\|_2=1} \lambda_1 y_1^2 + \dots + \lambda_n y_n^2 \\
&= \max\{\lambda_1, \dots, \lambda_n\} \text{ the chosen } \mathbf{y} \text{ is when } (y_1^2, \dots y_n^2) \text{ is a one hot corresponding to largest } \lambda \\
&= \lambda_{\max}(\mathbf{A}^\top \mathbf{A})
\end{aligned}
$$

- Question: what is wrong with instead finding a vector $\begin{bmatrix} y_1^2 & \dots & y_n^2 \end{bmatrix}$ that is in the same direction as $[\lambda_1 \quad \dots \quad \lambda_n]$?
- Answer: $\|\mathbf{y}\|_2 = 1 \implies [y_1 \quad \dots \quad y_n]$ is a unit vector and $\begin{bmatrix} y_1^2 & \dots & y_n^2 \end{bmatrix}$ is not!
- for example, in 2D, $\mathbf{y} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ is unit vector, and $\mathbf{y} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ is not!

Decompose further: $\mathbf{P} \rightarrow (\mathbf{R}, \mathbf{t})$

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & | & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & | & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & | & p_{3,4} \end{bmatrix} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] = \mathbf{K}[\mathbf{R} \mid \underbrace{-\mathbf{R}\mathbf{c}}_{\mathbf{t}}]$$

$$\mathbf{K}\mathbf{R} \qquad\qquad -\mathbf{K}\mathbf{R}\mathbf{c}$$

$\mathbf{c}$ is the camera center

something on change co-ordinate system

- leave out $\mathbf{K}$ for now: if we were to transform $\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$ by just the extrinsic/pose matrix $[\mathbf{R} \quad \mathbf{t}]$:

$$
\begin{aligned}
&[\mathbf{R} \quad \mathbf{t}]\begin{bmatrix}\mathbf{X}\\1\end{bmatrix} = \mathbf{RX} + \mathbf{t} \\
&= [\mathbf{R} \quad -\mathbf{Rc}]\begin{bmatrix}\mathbf{X}\\1\end{bmatrix} = \mathbf{RX} - \mathbf{Rc} = \mathbf{R}(\mathbf{X} - \mathbf{c}) \qquad \text{expression using } \mathbf{c}
\end{aligned}
$$

- making sure second expression is correct, let

  $\mathbf{X} = \mathbf{c} \implies [\mathbf{R} \quad -\mathbf{Rc}]\begin{bmatrix}\mathbf{c}\\1\end{bmatrix} = \mathbf{Rc} - \mathbf{Rc} = \mathbf{0}$

- if we to transform point $\mathbf{X}$ (defined in some "world coordinate") to the "camera coordinate" (with camera center $= \mathbf{c}$ defined by world coordinate), we need:
  1. subtract $\mathbf{X}$ by $\mathbf{c}$
  2. perform rotation $\mathbf{R}$

  alternative is to perform rotation $\mathbf{R}$ first, and then translate by $-\mathbf{Rc}$
- both are the same

3D Triangulation

2D image point $\rightarrow$ 3D point given $\mathbf{P}$

Finding a 3D point from stereo pair of images

- now we know **P**, and given a 2D image point **x**, we want to find 3D point **X**:

$$s\mathbf{x} = \mathbf{P}\mathbf{X}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\lambda}_{\frac{1}{s}} \underbrace{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} — & \mathbf{p}_1 & — \\ — & \mathbf{p}_2 & — \\ — & \mathbf{p}_3 & — \end{bmatrix} \begin{bmatrix} | \\ \mathbf{X} \\ | \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{p}_1^\top \mathbf{X} \\ \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_3^\top \mathbf{X} \end{bmatrix}$$

- here comes the cross-product trick:

$$\mathbf{x} = \lambda\mathbf{P}\mathbf{X} \quad \text{means } \mathbf{x} \text{ and } \mathbf{PX} \text{ are in same direction}$$

$$\Longrightarrow \mathbf{x} \underbrace{\times}_{\text{cross prod}} \mathbf{P}\mathbf{X} = \mathbf{0} \quad \text{cross product of the same direction } = \mathbf{0}$$

Finding a 3D point from stereo pair of images

$$\mathbf{x} \times \mathbf{P}\mathbf{X} = \mathbf{0}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{p}_1^\top \mathbf{X} \\ \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_3^\top \mathbf{X} \end{bmatrix} = \begin{bmatrix} v\mathbf{p}_3^\top \mathbf{X} - \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_1^\top \mathbf{X} - u\mathbf{p}_3^\top \mathbf{X} \\ u\mathbf{p}_2^\top \mathbf{X} - v\mathbf{p}_1^\top \mathbf{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- notice that the last row is a linear combination of the first two:

$$\begin{aligned} & u\left(v\mathbf{p}_3^\top \mathbf{X} - \mathbf{p}_2^\top \mathbf{X}\right) + v\left(\mathbf{p}_1^\top \mathbf{X} - u\mathbf{p}_3^\top \mathbf{X}\right) \\ &= uv\mathbf{p}_3^\top \mathbf{X} - u\mathbf{p}_2^\top \mathbf{X} + v\mathbf{p}_1^\top \mathbf{X} - uv\mathbf{p}_3^\top \mathbf{X} \\ &= -u\mathbf{p}_2^\top \mathbf{X} + v\mathbf{p}_1^\top \mathbf{X} \end{aligned}$$

- so we ignore it and to use only the first two equations:

$$\begin{bmatrix} v\mathbf{p}_3^\top & - \mathbf{p}_2 \\ \mathbf{p}_1^\top & - u\mathbf{p}_3 \end{bmatrix} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

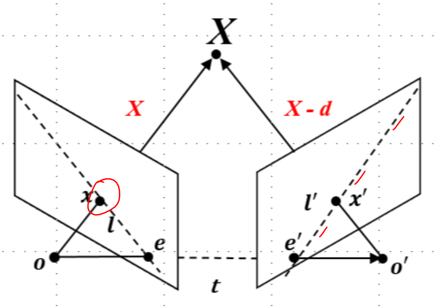
adding another camera

- adding second pair of camera having $\mathbf{P}'$ projection matrix:

$$\underbrace{\begin{bmatrix} v\mathbf{p}_3^\top - \mathbf{p}_2 \\ \mathbf{p}_1^\top - u\mathbf{p}_3 \\ v'\mathbf{p}_3'^\top - \mathbf{p}_2' \\ \mathbf{p}_1'^\top - u'\mathbf{p}_3' \end{bmatrix}}_{\mathbf{A}} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

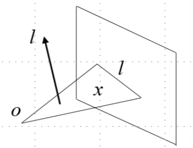
Epi-polar Geometry

FX



in this section, we use **o** for camera centre, instead of **c**

Line equation



- First, let's look at **line equation**:
  1. a line on a "2D image plane" is an intersection between:
     - "image plane" and
     - a particular "plane defined by its normal"
  2. the same normal $\mathbf{l}$ also defines such a line in that image plane $\mathbf{l}$
- algebraically:

$$
\begin{aligned}
ax + by + c &= 0 \\
\Longrightarrow \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} &= 0 \\
\Longrightarrow \mathbf{x}^\top \mathbf{l} &= 0
\end{aligned}
$$

- points $\mathbf{x}$ of plane satisfy $\mathbf{x}^\top \mathbf{l}$ forms a particular "plane defined by its normal" $\mathbf{l}$

Bring it to the camera setting

- if we design the image plane to be $z = 1$
  $\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ and $\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ 1 \end{bmatrix}$ intersects with this image plane
- obviously by definition $\mathbf{e}^\top \mathbf{l} = 0$
- $\mathbf{l}$ is not unique: infinite planes can intersect with image plane $z = 1$ and contain points $\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ and $\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ 1 \end{bmatrix}$ - think what if change the angle of plane intersecting with image plane
- however, since $\mathbf{o}$ is also co-plane with $\mathbf{x}$ and $\mathbf{e}$, making $\mathbf{l}$ unique, and since $\mathbf{l}$ is normal to all points in this plane:

$$\mathbf{l}^\top \mathbf{o} = \mathbf{l}^\top \mathbf{x} = \mathbf{l}^\top \mathbf{e} = 0$$

- in the second camera system, we have:

$$\mathbf{l}'^\top \mathbf{o}' = \mathbf{l}'^\top \mathbf{x}' = \mathbf{l}'^\top \mathbf{e}' = 0$$

Essential Matrix

- our desire is to have an essence matrix $\mathbf{E}$ to link the pair $(\mathbf{x}, \mathbf{l}')$ and $(\mathbf{x}', \mathbf{l})$, each defined in their respective co-ordinate systems

$$\mathbf{E}\mathbf{x} = \mathbf{l}' \quad \text{similarly} \quad \mathbf{E}\mathbf{x}' = \mathbf{l}$$

- if point $\mathbf{x}$ is on epi-polar line $\mathbf{l}$ by definition:

$$\mathbf{x}^\top \mathbf{l} = 0 \quad \text{similarly} \quad \mathbf{x}'^\top \mathbf{l}' = 0$$

- now putting things together:

$$\begin{aligned} \mathbf{E}\mathbf{x} &= \mathbf{l}' \\ \mathbf{x}'^\top \mathbf{E}\mathbf{x} &= \mathbf{x}'^\top \mathbf{l}' \\ &= 0 \end{aligned}$$

- $\mathbf{E}$ encodes epipolar geometry, maps a point to a line
- we can learn values of $\mathbf{E}$ by using $\mathbf{x}$ and $\mathbf{x}'$ as data
- then, question is, what is the physcial meaning of $\mathbf{E}$?

Essential Matrix

- we can learn values of $\mathbf{E}$ by using $\mathbf{x}$ and $\mathbf{x}'$ as data,
- note that in here, $\mathbf{x}$ and $\mathbf{x}'$ are normalized image co-ordinates
- then, question is, what is the physcial meaning of $\mathbf{E}$?
- it turns out that:

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

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

- remember in epipolar geometry:
  1. $\mathbf{x}$ is defined to coordinate $\mathbf{o}$
  2. $\mathbf{x}'$ is defined to coordinate $\mathbf{o}'$
- but both are the same object!
  so $\mathbf{x}$ can be transformed from systems: $\mathbf{o} \to \mathbf{o}'$
- we assume that in camera system $\mathbf{o}' = \mathbf{o} + \mathbf{t}$, meaning from $\mathbf{o}$, $\mathbf{o}' = \mathbf{t}$

$$\begin{aligned} \mathbf{x}' &= \mathbf{R}(\mathbf{x} - \mathbf{t}) \\ \implies \mathbf{R}^\top \mathbf{x}' &= (\mathbf{x} - \mathbf{t}) \qquad \text{multiple by } \mathbf{R}^\top \end{aligned}$$

- we know $(\mathbf{x} - \mathbf{t})^\top (\mathbf{t} \times \mathbf{x}) = 0$

$$\begin{aligned} &(\mathbf{x} - \mathbf{t})^\top (\mathbf{t} \times \mathbf{x}) = 0 \\ \implies &(\mathbf{R}^\top \mathbf{x}')^\top (\mathbf{t} \times \mathbf{x}) = 0 \\ \implies &\underbrace{(\mathbf{x}'^\top \mathbf{R})}_{\text{row vector}} (\mathbf{t} \times \mathbf{x}) = 0 \end{aligned}$$

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

$$
\begin{aligned}
(\mathbf{x}'^\top \mathbf{R})(\mathbf{t} \times \mathbf{x}) &= 0 \\
(\mathbf{x}'^\top \mathbf{R})([\mathbf{t}_\times]\mathbf{x}) &= 0 \text{ change cross-product to matrix multiplication} \\
\mathbf{x}'^\top \underbrace{\mathbf{R}[\mathbf{t}_\times]}_{\mathbf{E}} \mathbf{x} &= 0 \\
\mathbf{x}'^\top \mathbf{E}\mathbf{x} &= 0 \text{ where } \mathbf{E} \text{ is essential matrix}
\end{aligned}
$$

$$
\mathbf{E} = \mathbf{R} \underbrace{[\mathbf{t}_\times]}_{\text{rank 2!}}
$$

- how to change **cross product** to **matrix multiplication**:

$$
\begin{aligned}
\mathbf{a} \times \mathbf{b} &= \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} \\
\Longrightarrow [\mathbf{a}_\times]\mathbf{b} &= \underbrace{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}}_{[\mathbf{a}_\times]} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}
\end{aligned}
$$

Essential Matrix apply to Epipoles

$$\mathbf{x}'^{\top}\mathbf{E}\mathbf{x} = 0 \qquad \mathbf{x}^{\top}\mathbf{E}^{\top}\mathbf{x}' = 0 \qquad \mathbf{x}'^{\top}\mathbf{l}' \qquad = 0 \qquad \mathbf{x}^{\top}\mathbf{l} = 0$$

look at epi-poles $\mathbf{e}$ and $\mathbf{e}'$:

- what would happen is you let $\mathbf{x} = \mathbf{e}$? where should its $\mathbf{l}'$ be?
- $\mathbf{e}$ lies on **all** epipolar lines in left image, so:

$$\begin{aligned} \mathbf{e}^{\top}\mathbf{l} &= 0 \\ \implies \mathbf{e}^{\top}\mathbf{E}^{\top}\mathbf{x}' &= 0 \\ \implies \mathbf{x}'^{\top}(\mathbf{E}\mathbf{e}) &= 0 \\ \implies \mathbf{E}\mathbf{e} &= \mathbf{0} \end{aligned}$$

  look at the diagram atgin, when $\mathbf{x} = \mathbf{e} \implies \mathbf{X}$ moves to the line joining $\mathbf{o}$ and $\mathbf{o}'$
- similarly:

$$\implies \mathbf{e}'^{\top}\mathbf{E} = \mathbf{0}$$

- knowing $\mathbf{E}\mathbf{e} = \mathbf{0}$ means that rank$(\mathbf{E}) = 2$
- alternative fact is that rank$([\mathbf{t}_{\times}] = 2$

Fundamental Matrix

- Essential matrix require **normalized co-ordinates** $(\hat{\mathbf{x}}', \hat{\mathbf{x}})$ i.e., not image points directly
- it require knowledge of $\mathbf{K}$ and $\mathbf{K}'$

$$
\begin{aligned}
\hat{\mathbf{x}}'^{\top}\mathbf{E}\hat{\mathbf{x}} &= 0 \\
\text{where} \quad \hat{\mathbf{x}} &= \mathbf{K}^{-1}\mathbf{x} \qquad \hat{\mathbf{x}'} = \mathbf{K}'^{-1}\mathbf{x}' \\
\implies (\mathbf{K}'^{-1}\mathbf{x}')^{\top}\mathbf{E}\mathbf{K}^{-1}\mathbf{x} &= 0 \\
\implies \mathbf{x}'^{\top}\underbrace{\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}}_{\mathbf{F}}\mathbf{x} &= 0 \\
\implies \mathbf{x}'^{\top}\mathbf{F}\mathbf{x} &= 0
\end{aligned}
$$

- Fundamental matrix does **not** require $\mathbf{K}$ and $\mathbf{K}'$
- remember $\mathbf{E} = \mathbf{R}[\mathbf{t}_{\times}]$:

$$
\begin{aligned}
\mathbf{x}'^{\top}\underbrace{\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}}_{\mathbf{F}}\mathbf{x} &= 0 \\
\implies \mathbf{F} &= \mathbf{K}'^{-\top}\mathbf{R}[\mathbf{t}_{\times}]\mathbf{K}^{-1}
\end{aligned}
$$

Essential matrix from Fundamental matrix

- can be computed in reverse: $\mathbf{F} \rightarrow \mathbf{E}$

$$\begin{aligned} \mathbf{F} &= \mathbf{K}'^{-\top}\mathbf{R}[\mathbf{t}_\times]\mathbf{K}^{-1} \\ \implies \mathbf{K}'^\top\mathbf{F} &= \mathbf{R}[\mathbf{t}_\times]\mathbf{K}^{-1} \\ \implies \mathbf{K}'^\top\mathbf{F}\mathbf{K} &= \mathbf{R}[\mathbf{t}_\times] = \mathbf{E} \end{aligned}$$

- the reverse equation is:

$$\mathbf{E} = \mathbf{K}'^\top\mathbf{F}\mathbf{K}$$

- we leave the recovering of **R** and **t** in next section

8-point algorithm

- $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$:

$$\Longrightarrow \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

- bring parameters into unknowns:

$$\Longrightarrow \begin{bmatrix} x_1 x_1' & x_1 y_1' & x_1 & y_1 x_1' & y_1 y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1 x_1' & x_1 y_1' & x_1 & y_1 x_1' & y_1 y_1' & y_1 & x_1' & y_1' & 1 \end{bmatrix} \begin{bmatrix} f_{1,1} \\ f_{1,2} \\ f_{1,3} \\ f_{2,1} \\ f_{2,2} \\ f_{2,3} \\ f_{3,1} \\ f_{3,2} \\ f_{3,3} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{0}$$

3D Reconstruction

Reconstruction Framework

- **input** $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ image points of n different views of the **same 3D object**
- **output** $\{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ and $\mathbf{X}$
  obviously there is only a single static $\mathbf{X}$

How do we perform reconstruction?

1. compute $\mathbf{F}$: $\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$
2. $\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [\mathbf{P}|\ \mathbf{0}]$ use first camera as the reference

   $\mathbf{P}' = \left[[\mathbf{e}'_{\times}]\mathbf{F} \mid \mathbf{e}' \right]$ for the rest of the poses

   when we know $\mathbf{K}$ and $\mathbf{K}'$ (or $\mathbf{K} = \mathbf{K}'$ if same camera used), we can also obtain $(\mathbf{R}_n, \mathbf{t}_n)$ next page
3. apply triangulation to solve for $\mathbf{X}$:

Decomposing **F** into **R** and **t** when **K** is known

$$
\begin{aligned}
\mathbf{E} &= [\mathbf{t}_\times]\mathbf{R} \\
&= \mathbf{K}'^\top \mathbf{F}\mathbf{K} \qquad \text{(or } \mathbf{K} = \mathbf{K}' \text{ if same camera used)}
\end{aligned}
$$

- we need to find some $[\mathbf{t}_\times]$ and $\mathbf{R}$ such that their product is $\mathbf{E}$
- from $\mathbf{E}$, it's difficult to perform factorization directly, as $[\mathbf{t}_\times]$ and $\mathbf{R}$ have special properties, i.e., can not freely decompose
- But we can make the factorization on SVD of $\mathbf{E}$ instead:

$$
\mathbf{E} = \mathbf{U}\,\Sigma\,\mathbf{V}^\top \qquad \text{i.e., SVD}
$$

Decomposing **F** into **R** and **t** when **K** is known

- according to internal constraints of **E**, Σ must consist of two identical and one zero:

$$\Sigma = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- define

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{W}^{-1} = \mathbf{W}^{\top} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- we claim:

$$[\mathbf{t}_{\times}] = \mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^{\top} \qquad \mathbf{R} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^{\top} \quad \text{is one of a solution}$$

- multiply together and see:

$$\begin{aligned} [\mathbf{t}]_{\times}\,\mathbf{R} &= \mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^{\top}\mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^{\top} \\ &= \mathbf{U}\,\underbrace{\mathbf{W}\,\Sigma\mathbf{W}^{-1}}\,\mathbf{V}^{\top} \\ &= \mathbf{U}\,\Sigma\,\mathbf{V}^{\top} \\ &= \mathbf{E} \end{aligned}$$

why is $[\mathbf{t}_\times] = \mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^\top$ valid?

look at a cross product matrix: $[\mathbf{a}_\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

1. **property one** $[\mathbf{a}_\times]$'s diagonal = 0: plug-in definition of $(\mathbf{W}, \Sigma)$ and a random $\mathbf{U}$ will verify
2. **property two** $[\mathbf{a}_\times]$ is skew-symmetric:

$$\Longrightarrow [\mathbf{a}_\times]^\top = -[\mathbf{a}_\times]$$

- let's check for condition to make: $[\mathbf{t}_\times]^\top = -[\mathbf{t}_\times]$:

$$\begin{aligned}[\mathbf{t}_\times]^\top &= \left(\mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^\top\right)^\top \\ &= \mathbf{U}\Sigma^\top\,\mathbf{W}^\top\,\mathbf{U}^\top = \mathbf{U}\,(\mathbf{W}\,\Sigma)^\top\,\mathbf{U}^\top\end{aligned}$$

$$(\mathbf{W}\,\Sigma)^\top = \left(\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix}\right)^\top = \begin{pmatrix} 0 & -s & 0 \\ s & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^\top = -\begin{pmatrix} 0 & s & 0 \\ -s & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = -\mathbf{W}\,\Sigma$$

$$\Longrightarrow [\mathbf{t}_\times]^\top = \mathbf{U}\,(\mathbf{W}\,\Sigma)^\top\,\mathbf{U}^\top = -\mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^\top = -[\mathbf{t}_\times]$$

why is $\mathbf{R} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^\top$ valid?

- need to show $\mathbf{R} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^\top$ is a rotation matrix.
- product of three orthogonal matrices $\implies$ **R** too is orthogonal or

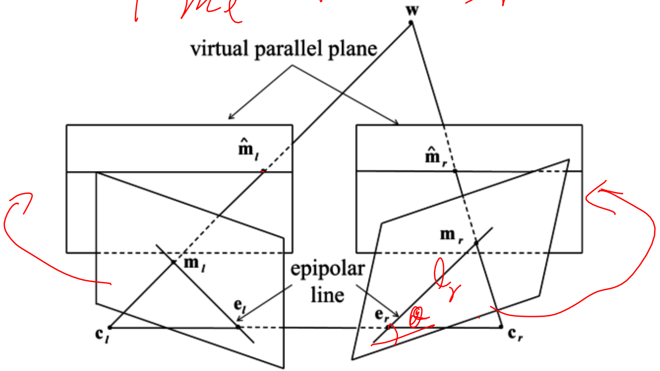
$$\det(\mathbf{R}) = \pm 1$$

- a rotation matrix must satisfy $\det(\mathbf{R}) = 1$
- Since, in this case, **E** is seen as a projective element this can be accomplished by reversing the sign of **E** if necessary.

Stereo Disparities

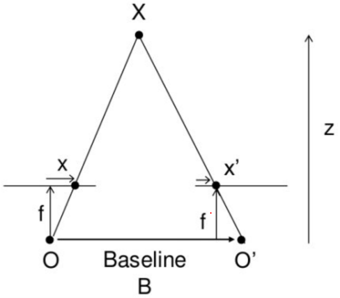
rectified stereo images

$F\hat{m}_l = \hat{l}_r$



- figure sourced from: *H. Ko, H. S. Shim, O. Choi and C.-C. J. Kuo, "Robust uncalibrated stereo rectification with constrained geometric distortions (USR-CGD)", Image Vis. Comput., vol. 60, pp. 98-114, Apr. 2017.*

estimating Stereo Disparities



- $\mathbf{X}$ is horizontal distance between $O \to X$, $\mathbf{B}$ is signed distance between $O \to O'$
- using similar triangles:

$$\mathbf{x} = f\frac{\mathbf{X}}{Z} \qquad \mathbf{x}' = f\frac{\mathbf{X} - \mathbf{B}}{Z} \quad \text{allow negative distance}$$

$$\implies \text{disparity} = \mathbf{x} - \mathbf{x}' = f\frac{\mathbf{X}}{Z} - f\frac{\mathbf{X} - \mathbf{B}}{Z} = \frac{f\mathbf{X} - f\mathbf{X} + f\mathbf{B}}{Z} = \frac{\mathbf{B}f}{Z}$$