Geometry for Computer Vision

Lecture 4a

Calibration and Oriented Epipolar Geometry

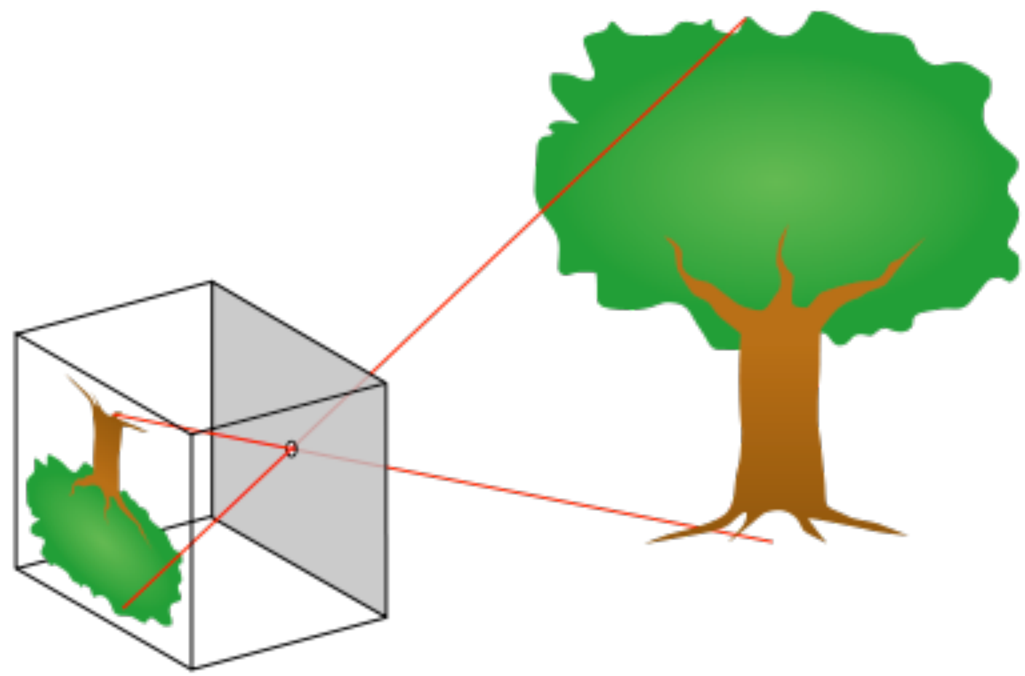
Per-Erik Forssén

Overview

1. Lens effects (distortion, vignetting)
2. Extrinsic and intrinsic camera parameters
3. Zhang’s camera calibration
4. Calibrated epipolar geometry (intro)
5. Oriented epipolar geometry

Break

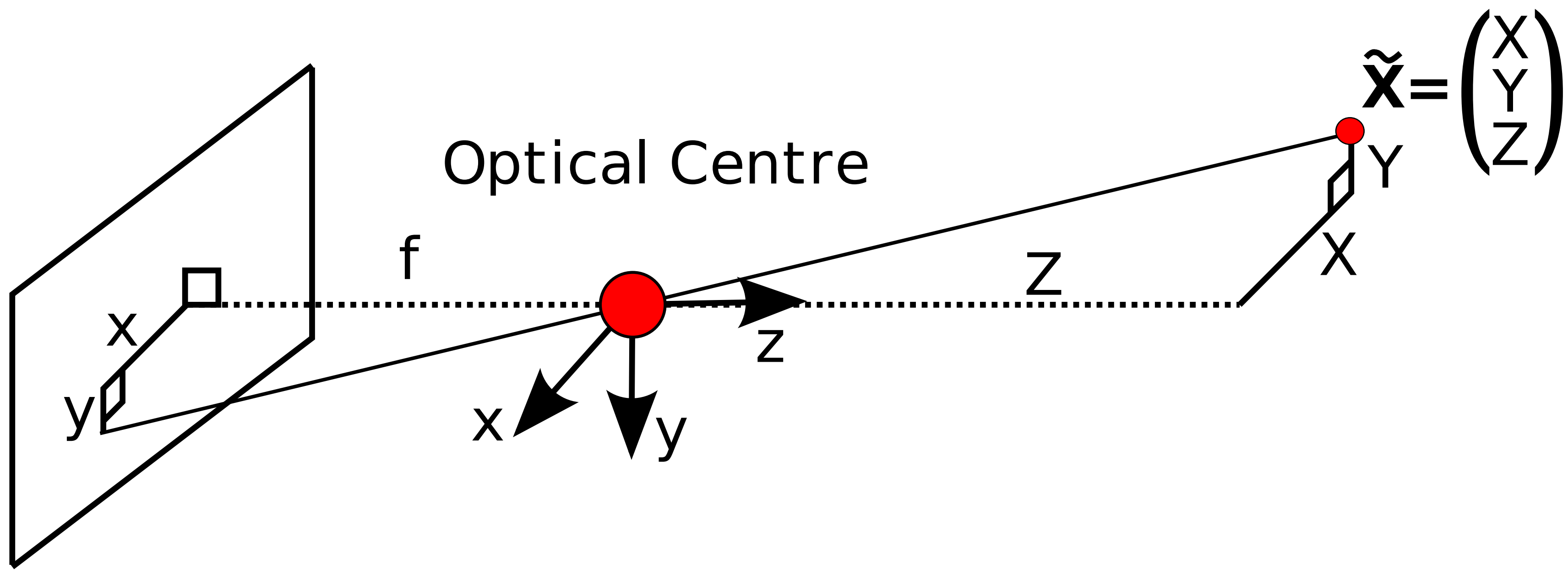
The pin-hole camera



A brightly illuminated scene will be projected onto a wall opposite of the pin-hole.

The image is rotated 180°.

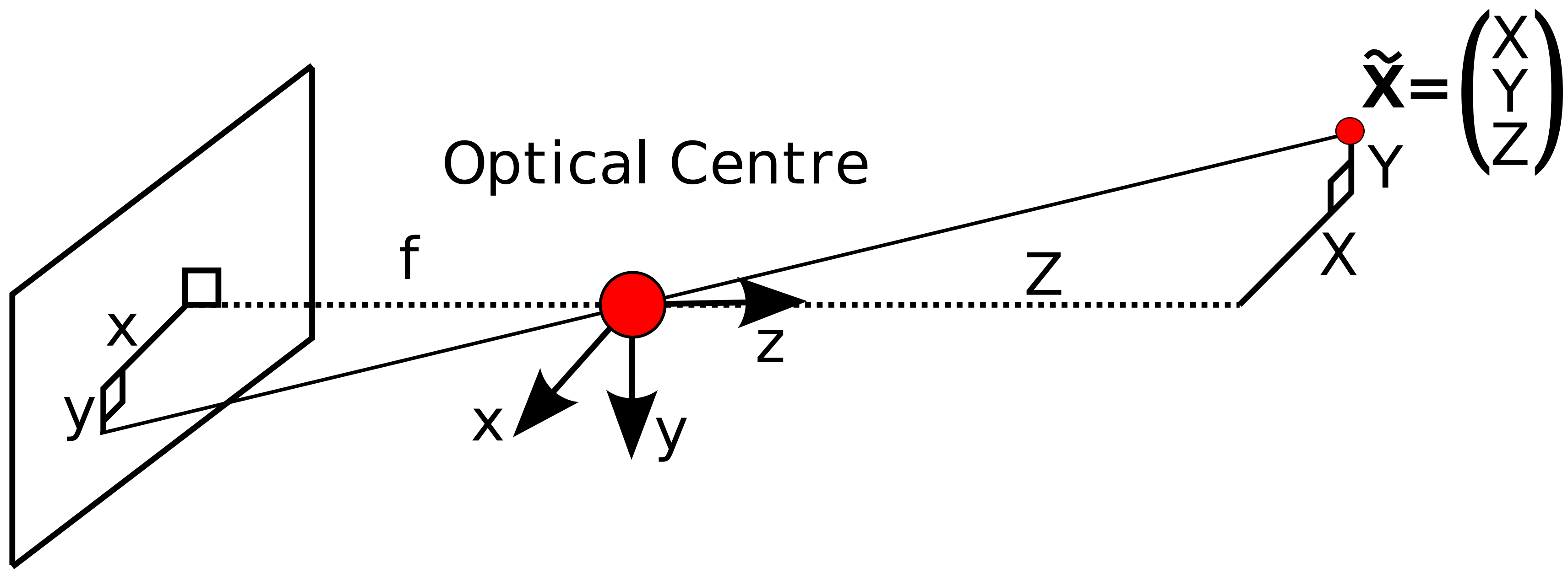
The pin-hole camera



- From similar triangles we get:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

The pin-hole camera



- From similar triangles we get:

$$\gamma \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The pin-hole camera

$$\gamma \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- More generally, we write:

$$\gamma \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & s & c_x \\ 0 & fa & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- f-focal length, s-skew, a-aspect ratio, (c_x,c_y)-projection of optical centre

The pin-hole camera

$$\gamma \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} f & s & c_x \\ 0 & fa & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\tilde{\mathbf{X}}}$$

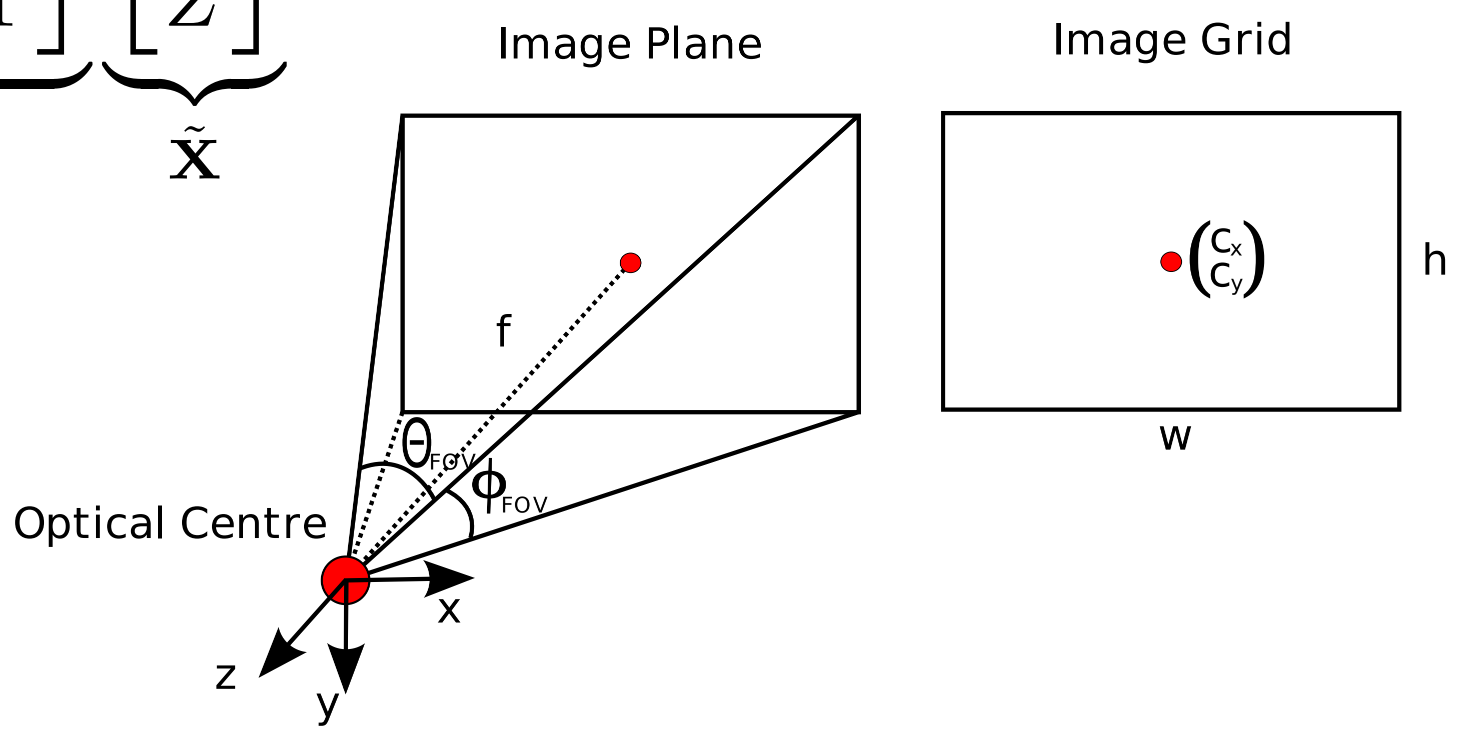
$$\mathbf{x} \sim \mathbf{K}\tilde{\mathbf{X}}$$

The pin-hole camera

$$\underbrace{\gamma\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} f & s & c_x \\ 0 & fa & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\tilde{\mathbf{X}}}$$

$$\mathbf{x} \sim \mathbf{K}\tilde{\mathbf{X}}$$

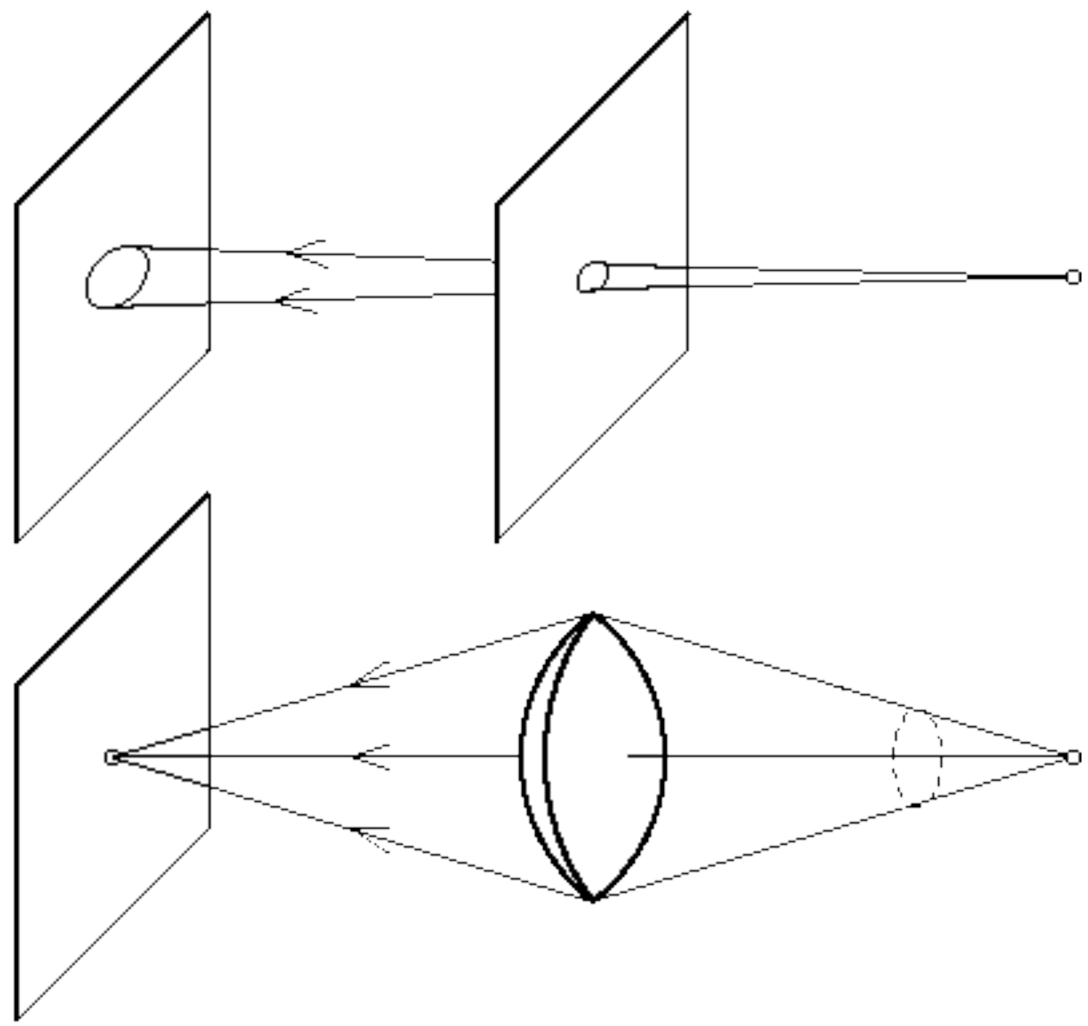
Motivation:



f-focal length, s-skew, a-aspect ratio, (c_x,c_y)-projection of optical centre

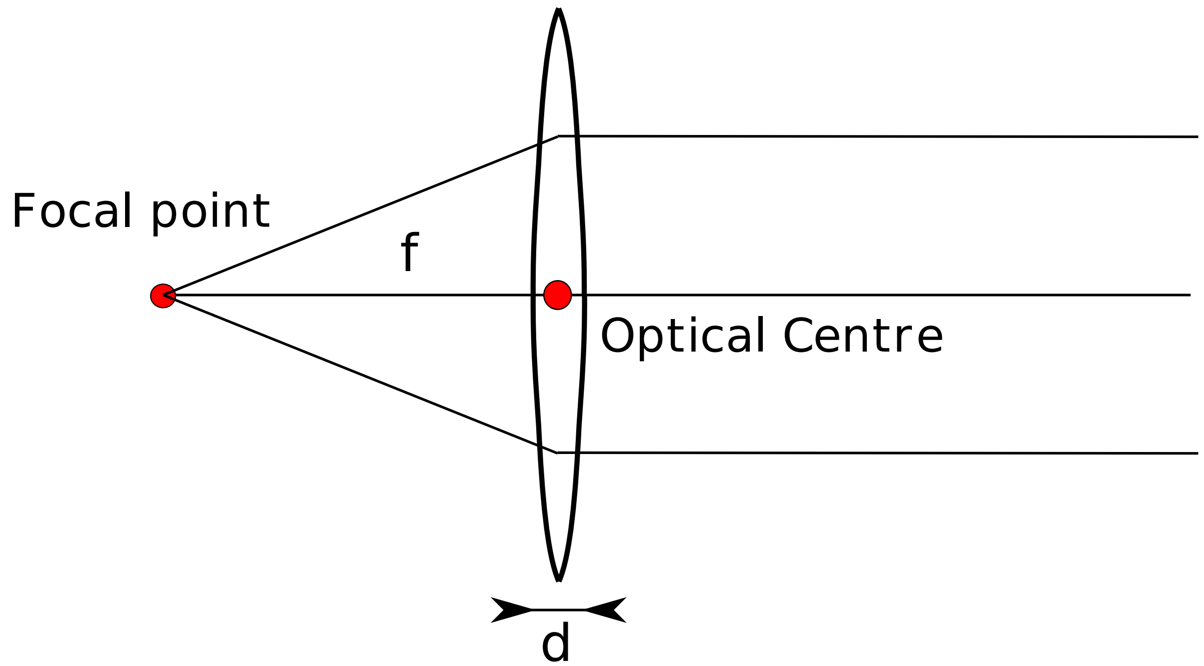
Thin Lens Camera

Real cameras use lenses, not pin-holes!



Thin Lens Camera

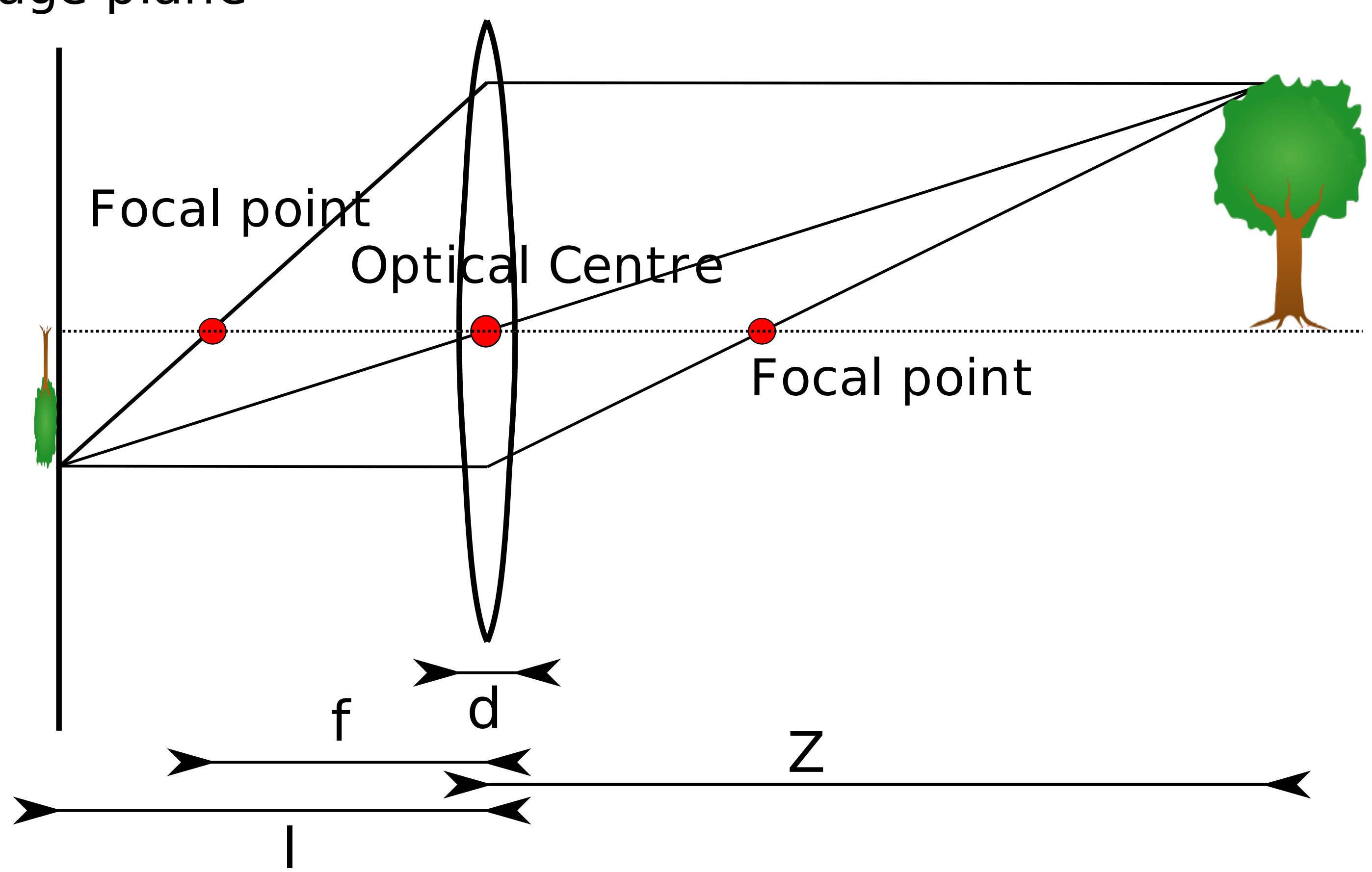
A thin lens is a (positive) lens with $d << f$



Parallel rays converge at the focal points

Rays through the optical centre are not refracted

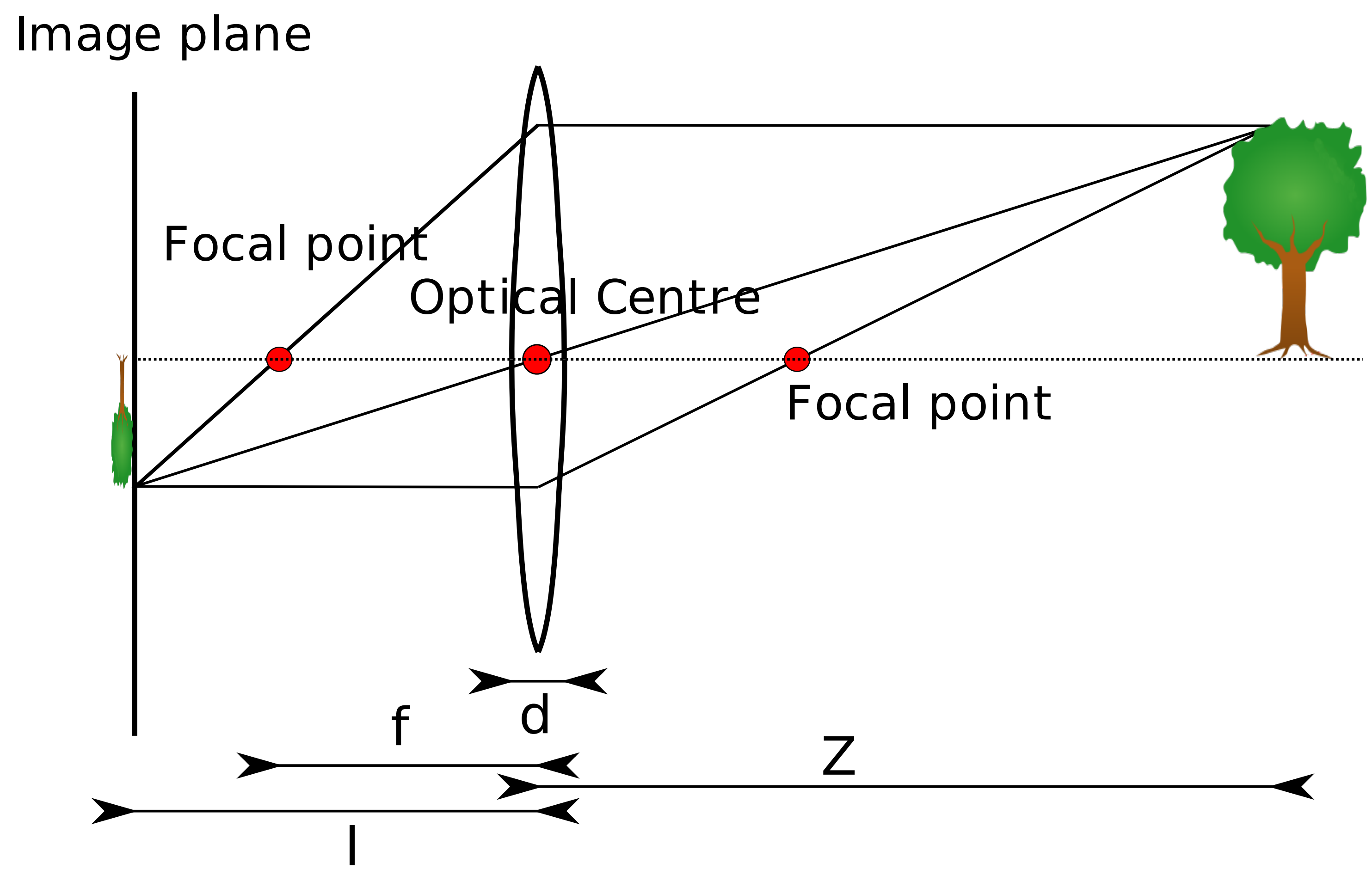
Thin Lens Camera



Thin lens relation (from similar triangles):

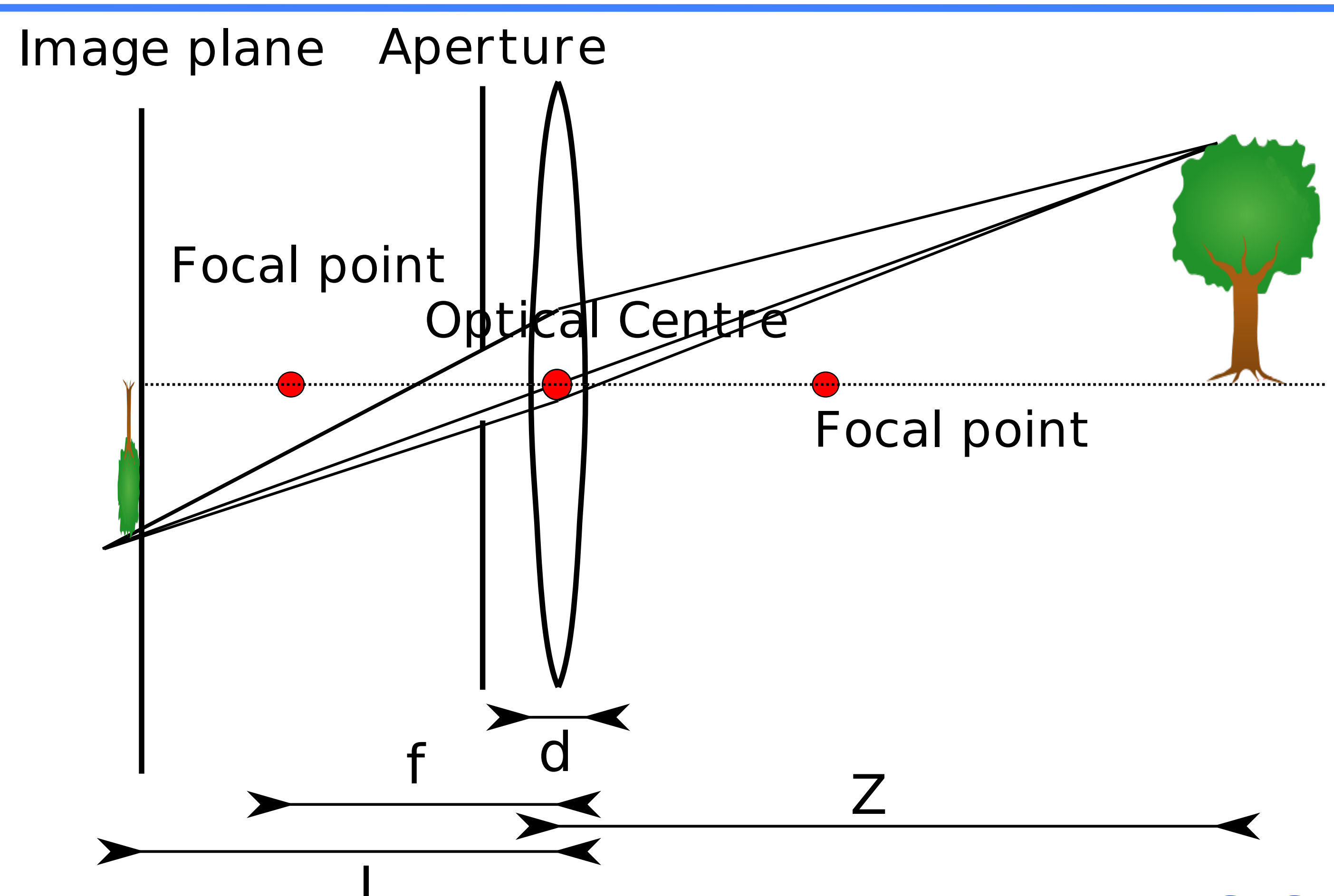
$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{l}$$

Thin Lens Camera



- Focus at one depth only.
- Objects at other depths are blurred.

Thin Lens Camera



An aperture increases the **depth-of-field**, the range which is sharp in the image.

A compromise between pinhole and thin lens.

Lens effects

Correct

Barrel distortion

Pin-cushion distortion

- Radial distortion
- For zoom lenses: Barrel at wide FoV pin-cushion at narrow FoV

Lens effects

Correct image

$\Rightarrow$

Distorted

- Modelling $\mathbf{x} \sim \mathbf{K} f(\mathbf{u}, \mathbf{\Theta}')$
- Used in optimisation such as BA

Lens effects

Distorted image

$\Rightarrow$

Correct

- Rectification $\quad \mathbf{x}' \sim f^{-1}(\mathbf{x}, \boldsymbol{\Theta})$
- Used in dense stereo

Distorsion polynomials

- Different models for different classes of cameras
- Radial model for normal and telecentric lenses with moderate distortion

$$r = \sqrt{x_1^2 + x_2^2} \quad \varphi = \operatorname{atan2}(x_2, x_1)$$

$$r' = \theta_1 r + \theta_2 r_3 \ldots \quad \varphi' = \varphi$$

$$x_1' = r' \cos\varphi' \quad x_2' = r' \sin\varphi'$$

- Also model centre of distortion

Distorsion polynomials

- For better accuracy:
  - Tangential distorsion
  - Rational model [Claus & Fitzgibbon 05]
- Specialised models:
  - wide-angle cameras [Kannala&Brandt 06]
  - catadioptric cameras [Micusik&Pajdla 03]
  - most simple is the FoV model [Devernay & Faugeras 2001]:

$$r' = \operatorname{atan}(r\theta_1)/\theta_1$$

Lens effects

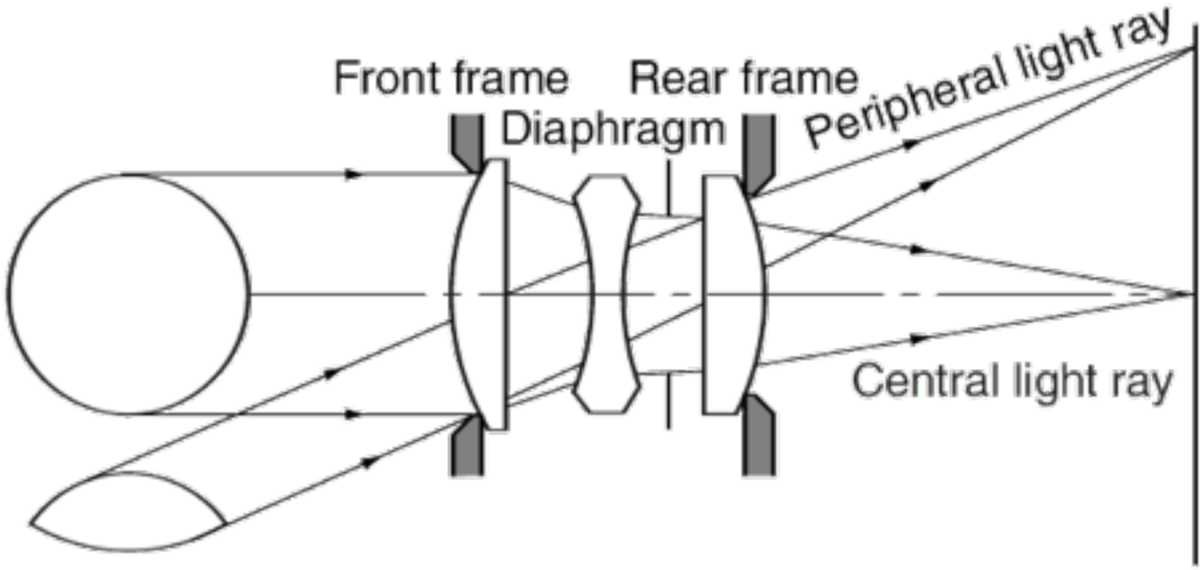
Correct

Darkened periphery

- Vignetting and $\cos^4$-law
- Stronger effects in wide FoV

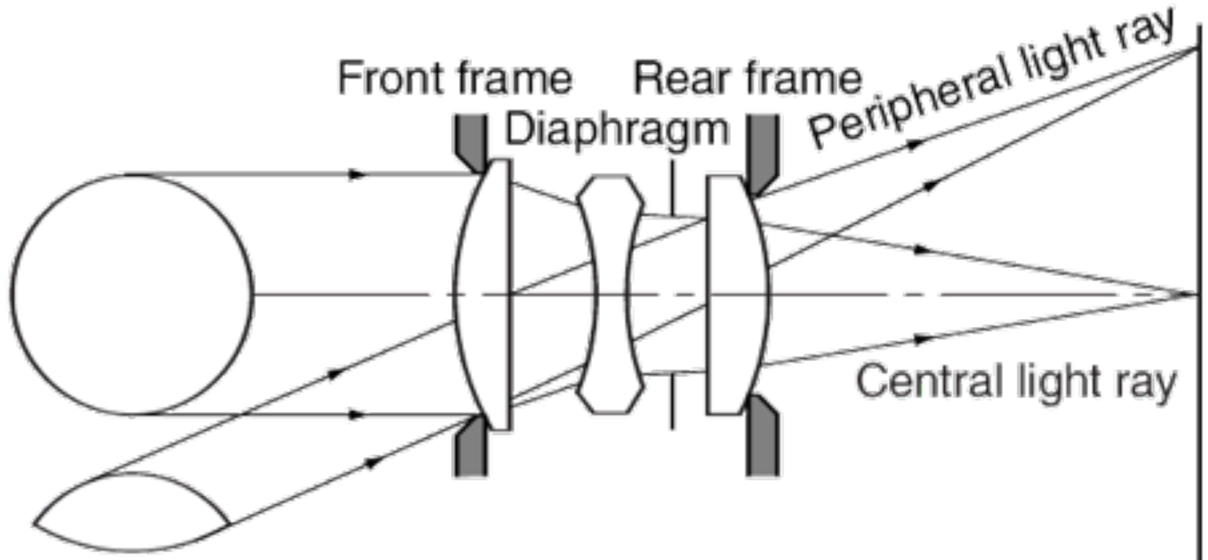
Lens effects

Vignetting



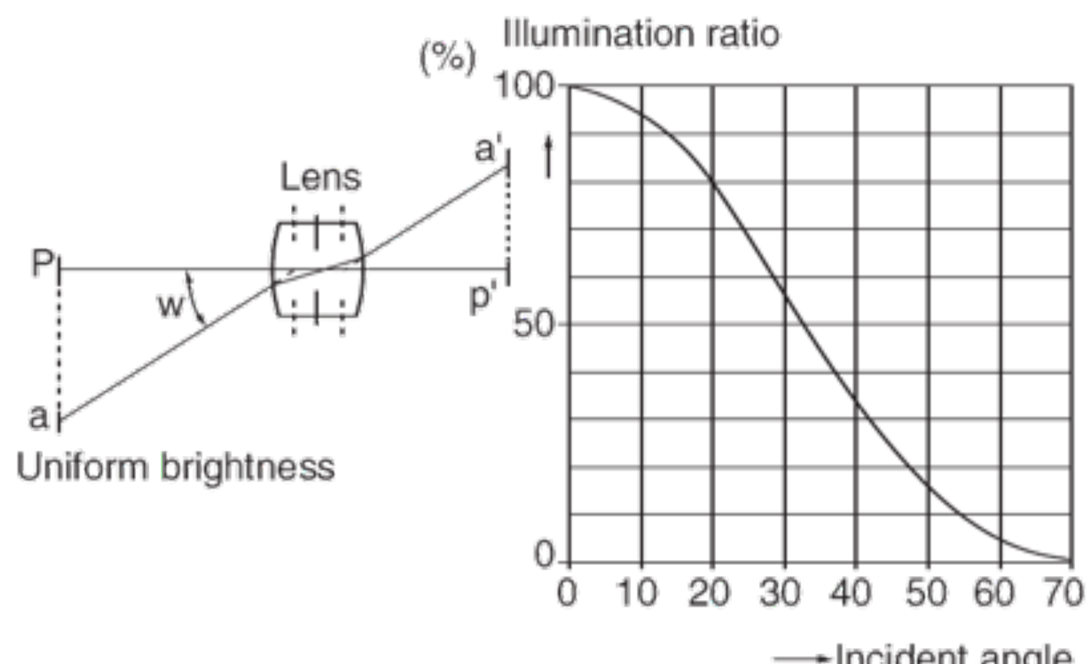
Lens effects

Vignetting



$\cos^4$-law

dampening with $\cos^4(w)$



http://software.canon-europe.com/files/documents/EF_Lens_Work_Book_10_EN.pdf

Camera parameters

For a general position of the world coordinate system (WCS) we have:

$$\mathbf{x} \sim \mathbf{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{[\mathbf{R}|\mathbf{t}]} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\mathbf{X}}$$

Camera parameters

For a general position of the world coordinate system (WCS) we have:

$$\mathbf{x} \sim \mathbf{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{[\mathbf{R}|\mathbf{t}]} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\mathbf{X}}$$

K contains the **intrinsic** parameters

[R | t] contain the **extrinsic** parameters

Camera parameters

Metric points transformed to the camera's coordinate system are called **normalised image coordinates**

$$\hat{\mathbf{x}} \sim [\mathbf{R}|\mathbf{t}]\,\mathbf{X}$$

In contrast to regular image coordinates

$$\mathbf{x} \sim \mathbf{K}\,[\mathbf{R}|\mathbf{t}]\,\mathbf{X} \qquad\qquad \mathbf{x} = \mathbf{K}\hat{\mathbf{x}}$$

K contains the **intrinsic** parameters

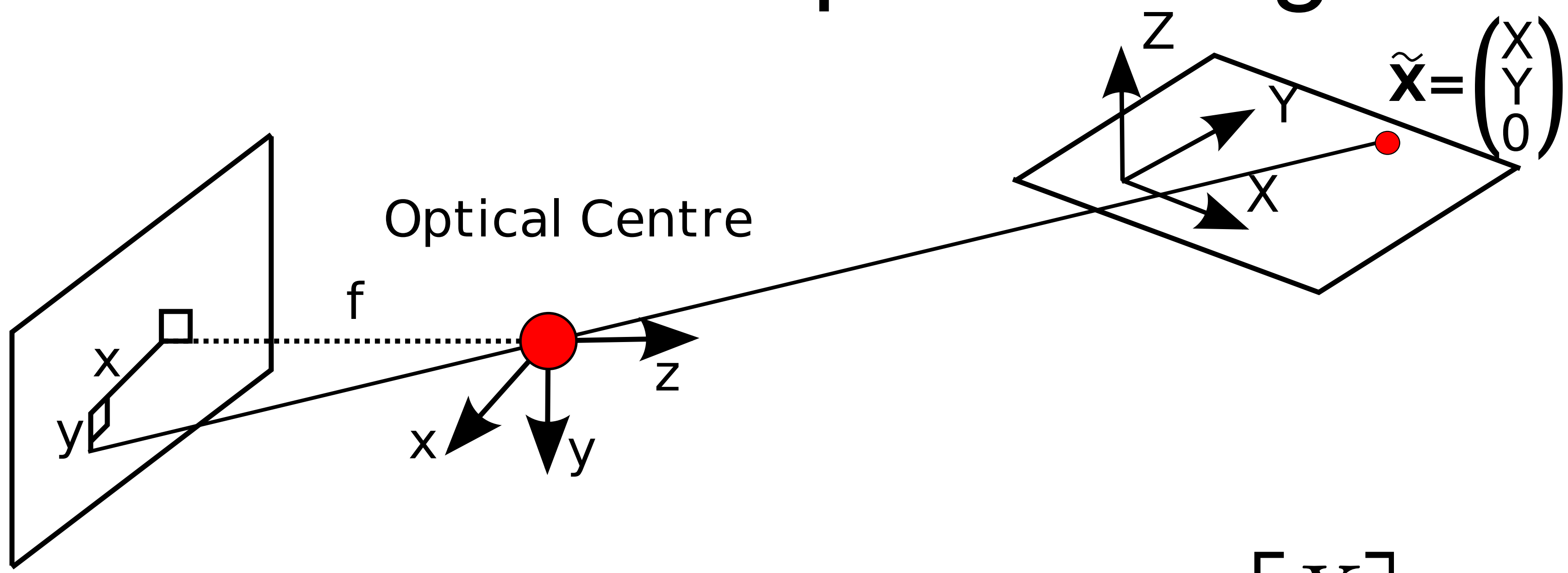
[**R** | **t**] contain the **extrinsic** parameters

Camera calibration

- Zhang's camera calibration (*A flexible new technique for camera calibration*, PAMI 2000)
- In OpenCV, and in Matlab toolbox
- Finds **K** from 3 or more photos of a planar calibration target
- Moderate lens distortion can also be estimated.

Camera calibration

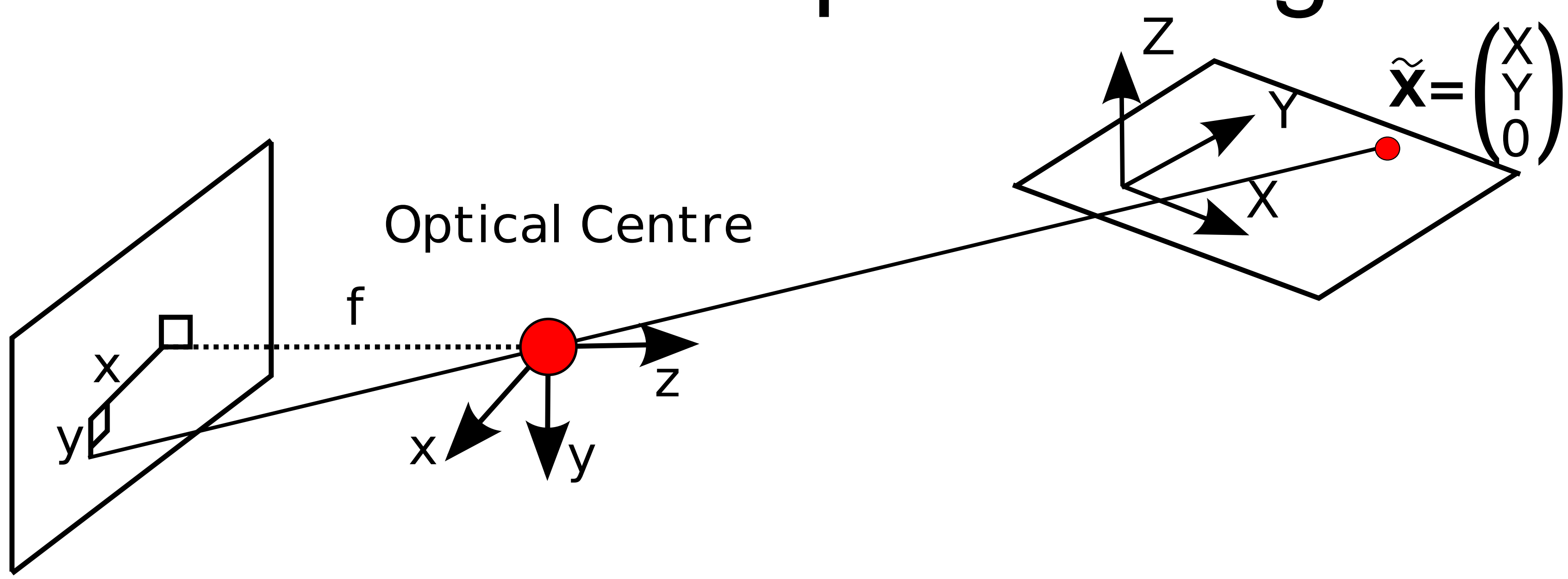
We now imagine a world coordinate system fixed to the planar target



$$\gamma \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera calibration

We now imagine a world coordinate system fixed to the planar target



$$\gamma \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Camera calibration

If we estimate a homography between the image and the model plane (lecture 3) we know **H**

$$\mathbf{H} = [\mathbf{h}_1 \;\; \mathbf{h}_2 \;\; \mathbf{h}_3] = \mathbf{K}\,[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}]$$

We also know that

$$\mathbf{r}_1^T \mathbf{r}_2 = 0 \quad \text{and} \quad \mathbf{r}_1^T \mathbf{r}_1 = \mathbf{r}_2^T \mathbf{r}_2$$

Camera calibration

If we estimate a homography between the image and the model plane (lecture 3) we know **H**

$$\mathbf{H} = [\mathbf{h}_1 \;\; \mathbf{h}_2 \;\; \mathbf{h}_3] = \mathbf{K}\,[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}]$$

We also know that

$$\mathbf{r}_1^T \mathbf{r}_2 = 0 \qquad \text{and} \quad \mathbf{r}_1^T \mathbf{r}_1 = \mathbf{r}_2^T \mathbf{r}_2$$

$$\Rightarrow \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0$$

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

Camera calibration

For a **K** of the form

$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

It can be shown that (use e.g. Maple)

$$\mathbf{K}^{-T}\mathbf{K}^{-1} = \mathbf{B} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Camera calibration

For a **K** of the form
$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

It can be shown that (use e.g. Maple)

$$\mathbf{K}^{-T}\mathbf{K}^{-1} = \mathbf{B} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Remember our constraints

$\mathbf{h}_1^T\mathbf{B}\mathbf{h}_2 = 0$ and $\mathbf{h}_1^T\mathbf{B}\mathbf{h}_1 - \mathbf{h}_2^T\mathbf{B}\mathbf{h}_2 = 0$

Camera calibration

As **B** is symmetric $$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_4 \\ b_2 & b_3 & b_5 \\ b_4 & b_5 & b_6 \end{bmatrix}$$

Camera calibration

As **B** is symmetric $\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_4 \\ b_2 & b_3 & b_5 \\ b_4 & b_5 & b_6 \end{bmatrix}$

If we now define $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix}^T$

The constraints can be written as

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0$$

$$\mathbf{v}_{ij} = [h_{i1}h_{j1},\ h_{i1}h_{j2} + h_{i2}h_{j1},\ h_{i2}h_{j2},\ h_{i3}h_{j1} + h_{i1}h_{j3},\ h_{i3}h_{j2} + h_{i2}h_{j3},\ h_{i3}h_{j3}]^T$$

Camera calibration

Each view of the plane gives us two rows in the system:

$$\mathbf{Vb} = 0$$

As **b** has 6 unknowns, we need 3 views of the plane.

Two views can also work if we require $\gamma = 0$

Camera calibration

Once **b** has been estimated, we can extract the parameters in **K** according to

$$v_0 = (b_2 b_4 - b_1 b_5)/(b_1 b_3 - b_2^2)$$
$$\lambda = b_6 - (b_3^2 + v_0(b_2 b_4 - b_1 b_5)/b_1$$
$$\alpha = \sqrt{\lambda / b_1}$$
$$\beta = \sqrt{\lambda b_1 / (b_1 b_3 - b_2^2)}$$
$$\gamma = -b_2 \alpha^2 \beta / \lambda$$
$$u_0 = \gamma v_0 \alpha - b_4 \alpha^2 / \lambda$$

Camera calibration

Once **b** has been estimated, we can extract the parameters in **K** according to

$$v_0 = (b_2 b_4 - b_1 b_5)/(b_1 b_3 - b_2^2)$$
$$\lambda = b_6 - (b_3^2 + v_0(b_2 b_4 - b_1 b_5)/b_1$$
$$\alpha = \sqrt{\lambda/b_1}$$
$$\beta = \sqrt{\lambda b_1/(b_1 b_3 - b_2^2)}$$
$$\gamma = -b_2 \alpha^2 \beta/\lambda$$
$$u_0 = \gamma v_0 \alpha - b_4 \alpha^2/\lambda$$

The H&Z book instead suggests Cholesky factorisation

Camera calibration

Cholesky factorisation of **B**(b)

$$\mathbf{B}(\mathbf{b}) = {\mathbf{K}^{-1}}^T \mathbf{K}^{-1}$$

Gives us $\mathbf{K}^{-1}$ which is invertible.

Camera calibration

Once **K** is computed we can also find the extrinsic camera parameters **R**,**t** for each image:

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1}\mathbf{h}_1 \quad \mathbf{r}_2 = \lambda \mathbf{K}^{-1}\mathbf{h}_2 \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{R} = [\mathbf{r}_1 \ \ \mathbf{r}_2 \ \ \mathbf{r}_3] \quad \mathbf{t} = \lambda \mathbf{K}^{-1}\mathbf{h}_3$$

$$(\,\lambda = 1/||\mathbf{K}^{-1}\mathbf{h}_1|| = 1/||\mathbf{K}^{-1}\mathbf{h}_2||\ )$$

Camera calibration

Once **K** is computed we can also find the extrinsic camera parameters **R**,**t** for each image:

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1}\mathbf{h}_1 \quad \mathbf{r}_2 = \lambda \mathbf{K}^{-1}\mathbf{h}_2 \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{R} = [\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{r}_3] \quad \mathbf{t} = \lambda \mathbf{K}^{-1}\mathbf{h}_3$$

Finally, **K**, $\mathbf{R}_i$, $\mathbf{t}_i$ are refined using ML (minimising the cost function)

$$\arg\min \sum_{i=1}^{n}\sum_{j=1}^{m} ||\mathbf{x}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)||^2$$

Camera calibration

Once **K** is computed we can also find the extrinsic camera parameters **R**,**t** for each image:

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1}\mathbf{h}_1 \quad \mathbf{r}_2 = \lambda \mathbf{K}^{-1}\mathbf{h}_2 \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix} \quad \mathbf{t} = \lambda \mathbf{K}^{-1}\mathbf{h}_3$$

Optionally, all of **K**,$\boldsymbol{\Theta}$,**R**$_i$, **t**$_i$ are refined using ML:

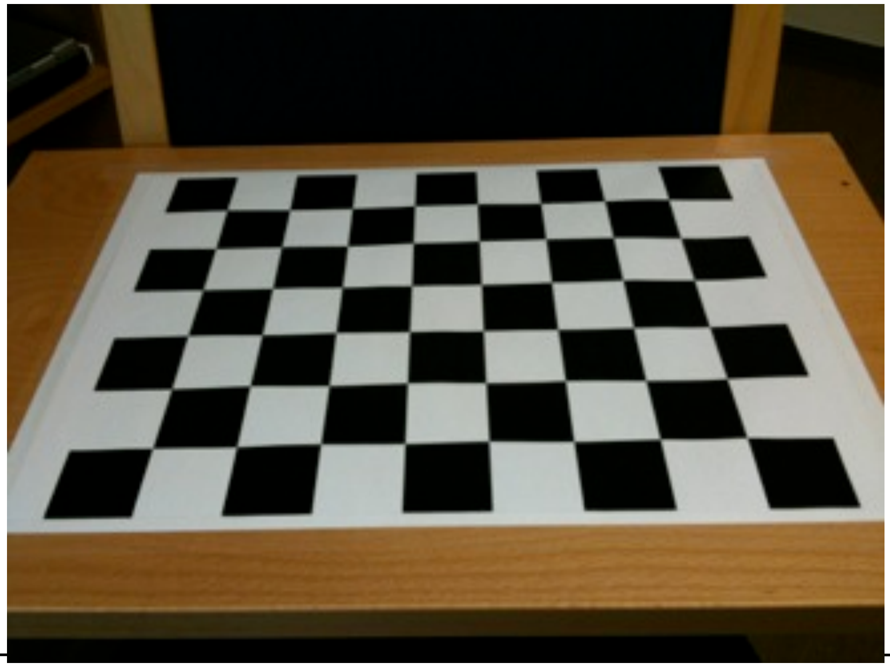
$$\arg\min \sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{x}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, \boldsymbol{\Theta}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)\|^2$$

Camera calibration

So what about the initial homographies?

$$\mathbf{H} = \mathbf{K}\left[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}\right]$$

Assign each point a WCS value $\mathbf{X} = [x\ y\ 0]^T$



Camera calibration

So what about the initial homographies?

$$\mathbf{H} = \mathbf{K}\left[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}\right]$$

Assign each point a WCS value $\mathbf{X} = [x\ y\ 0]^T$

Do we need to know which point is the upper left one on the checker-board? Why not?

Camera calibration

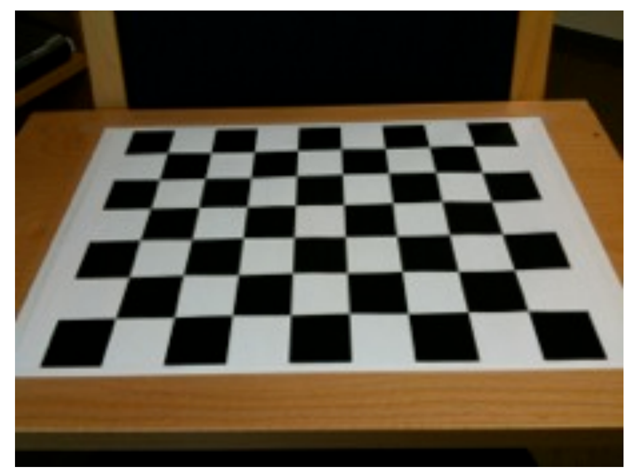
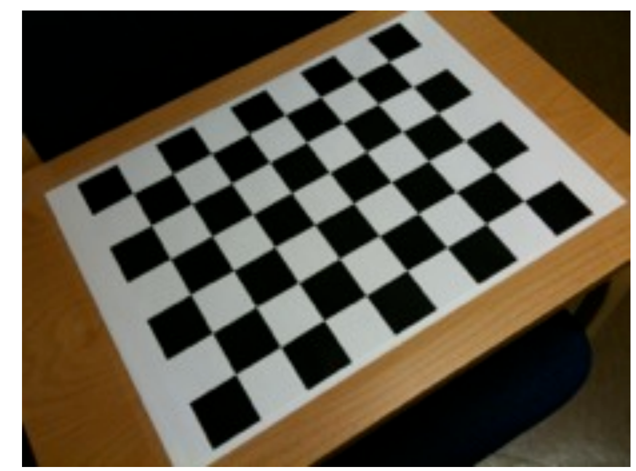
Can we use any combination images of the calibration plane?




Camera calibration

Can we use any combination images of the calibration plane?



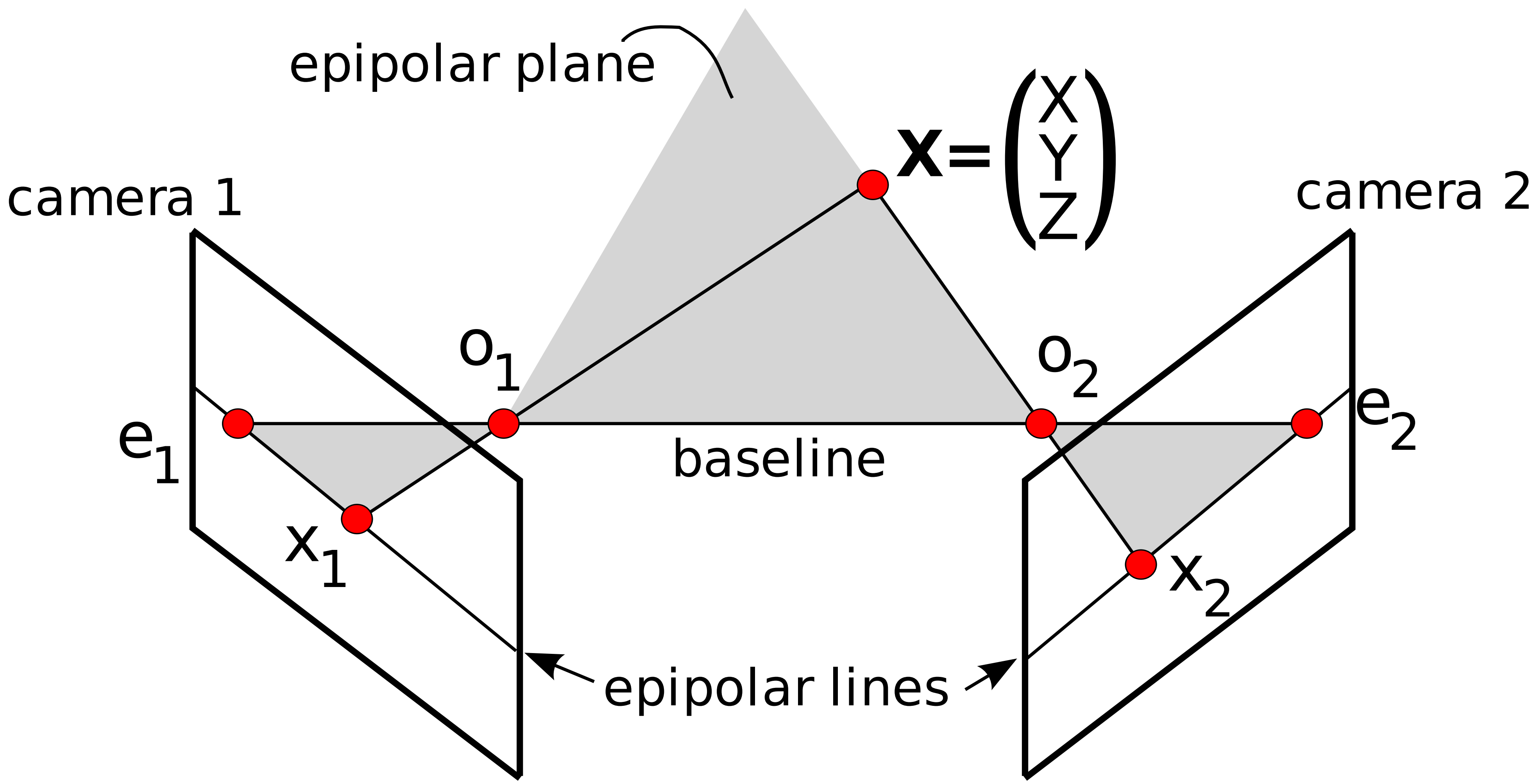



The constraints used: $\mathbf{r}_1^T \mathbf{r}_2 = 0$ and $\mathbf{r}_1^T \mathbf{r}_1 = \mathbf{r}_2^T \mathbf{r}_2$ have to be linearly independent.

$\Longrightarrow$ Planes must not be parallel!

Calibrated epipolar geometry

Recall the epipolar constraint $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$



Calibrated epipolar geometry

Recall the epipolar constraint $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$

...and the normalised image coordinates

$$\mathbf{x} = \mathbf{K}\hat{\mathbf{x}}$$

We can instead express the epipolar constraint in normalised coordinates

$$\hat{\mathbf{x}}_1^T \mathbf{K}_1^T \mathbf{F} \mathbf{K}_2 \hat{\mathbf{x}}_2 = 0 \quad \text{or} \quad \hat{\mathbf{x}}_1^T \mathbf{E} \hat{\mathbf{x}}_2 = 0$$

The matrix **E** is called the **essential matrix**. It has some interesting properties...

Calibrated epipolar geometry

In lecture 2 we saw that for cameras $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\mathbf{F} = [\mathbf{e}_{12}]_\times \mathbf{P}_1 \mathbf{P}_2^+ \qquad \mathbf{e}_{12} = \mathbf{P}_1 \mathbf{O}_2$$

Now, if $\mathbf{P}_2 = \mathbf{K}_2 \left[\mathbf{I}|\mathbf{0}\right]$ and $\mathbf{P}_1 = \mathbf{K}_1 \left[\mathbf{R}|\mathbf{t}\right]$

We get $\mathbf{P}_2^+ = \begin{bmatrix} \mathbf{K}_2^{-1} \\ \mathbf{0}^T \end{bmatrix}$ and

$$\mathbf{F} = \left[\mathbf{K}_1 \mathbf{t}\right]_\times \mathbf{K}_1 \mathbf{R} \mathbf{K}_2^{-1}$$

Calibrated epipolar geometry

Using the cross-product-commutator rule:

(A4.3) $$[\mathbf{b}]_\times \mathbf{A} = \det(\mathbf{A})\mathbf{A}^{-T}\left[\mathbf{A}^{-1}\mathbf{b}\right]_\times$$

on $$\mathbf{F} = [\mathbf{K}_1\mathbf{t}]_\times \mathbf{K}_1\mathbf{R}\mathbf{K}_2^{-1}$$

...we may express **F** as either of

$$\mathbf{F} = \mathbf{K}_1^{-T}[\mathbf{t}]_\times \mathbf{R}\mathbf{K}_2^{-1} \qquad \mathbf{F} = \mathbf{K}_1^{-T}\mathbf{R}\left[\mathbf{R}^T\mathbf{t}\right]_\times \mathbf{K}_2^{-1}$$

$$\mathbf{F} = \mathbf{K}_1^{-T}\mathbf{R}[\mathbf{t}_2]_\times \mathbf{K}_2^{-1}$$

Calibrated epipolar geometry

This gives us the essential matrix expressions:

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

E has only 5 dof (3 from **R**, 2 from **t**) recall that **F** has 7

A necessary and sufficient condition on **E** is that it has the singular values [a,a,0] (see 9.6.1 in the H&Z book for proof)

Calibrated epipolar geometry

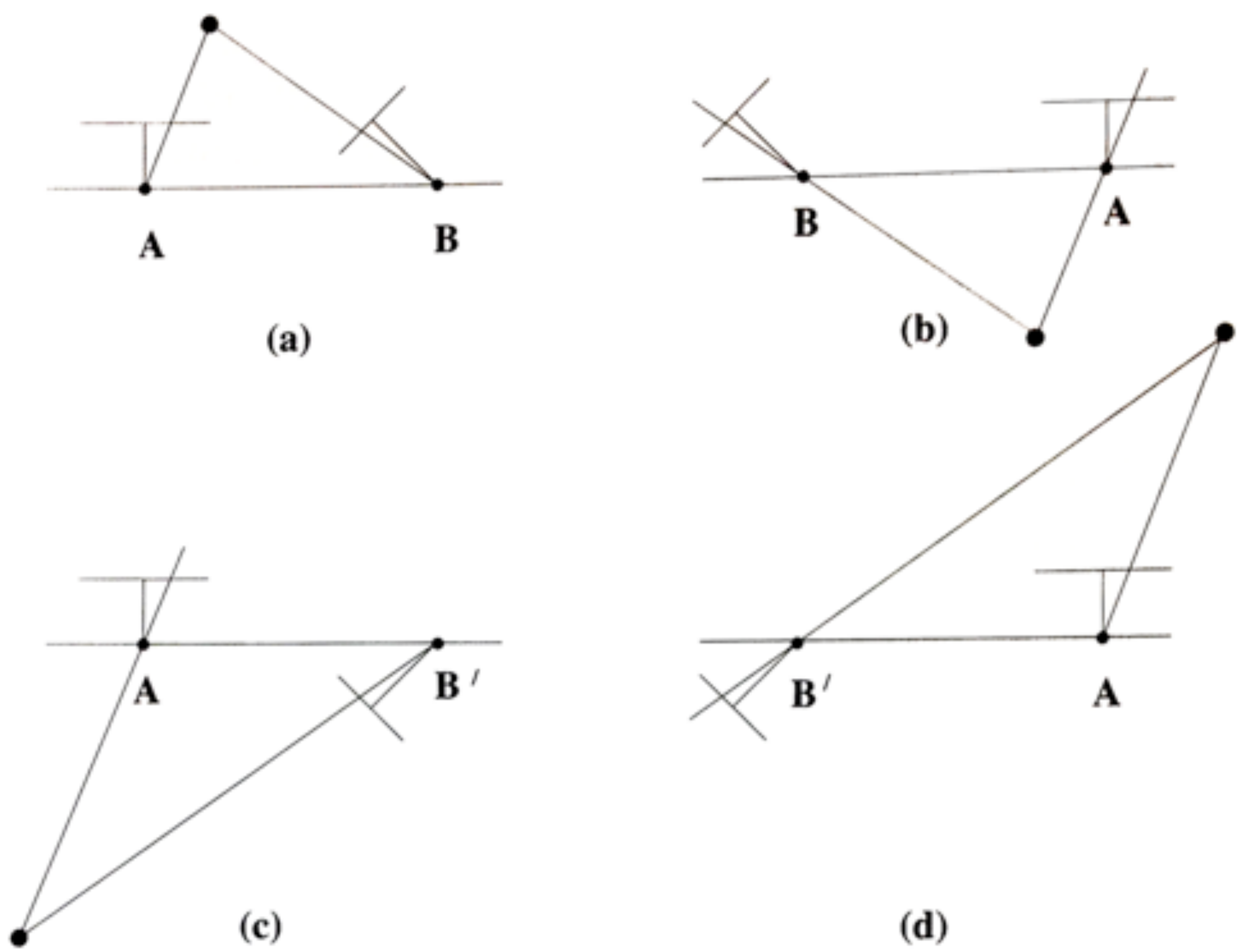
This gives us the essential matrix expressions:

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

We can extract **R** and **t** (up to scale) from **E** if we also make use of one point correspondence (a 3D point known to be in front of both cameras). See 9.6.2 in the H&Z book.

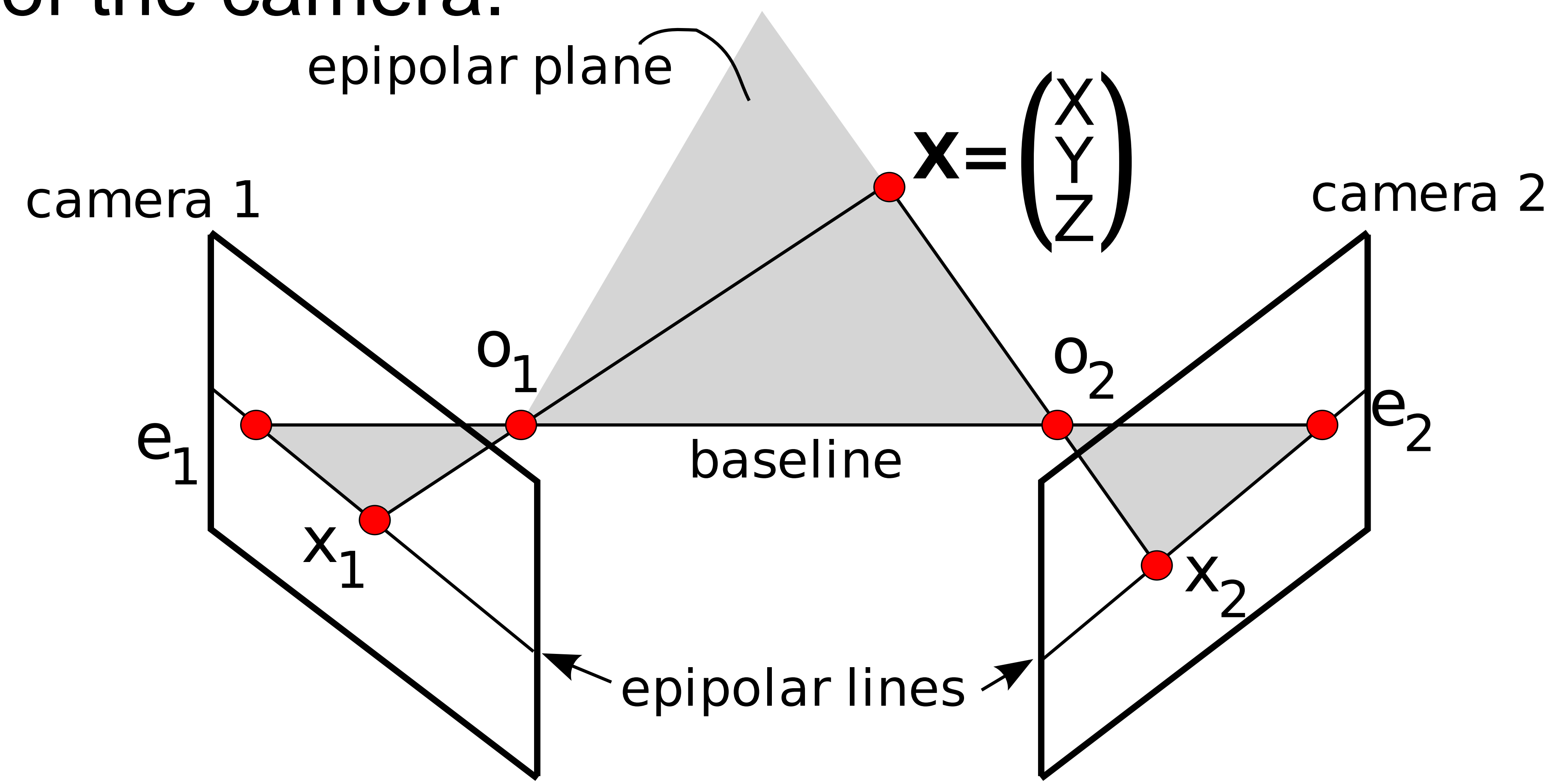
Calibrated epipolar geometry

4 cases for **R** and **t**, just one has point in front of both cameras.



Oriented epipolar geometry

The regular epipolar constraint $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$ ignores the knowledge that points are in front of the camera.



Oriented epipolar geometry

In **oriented projective geometry** a (visible) point in front of the camera is defined as having a projection

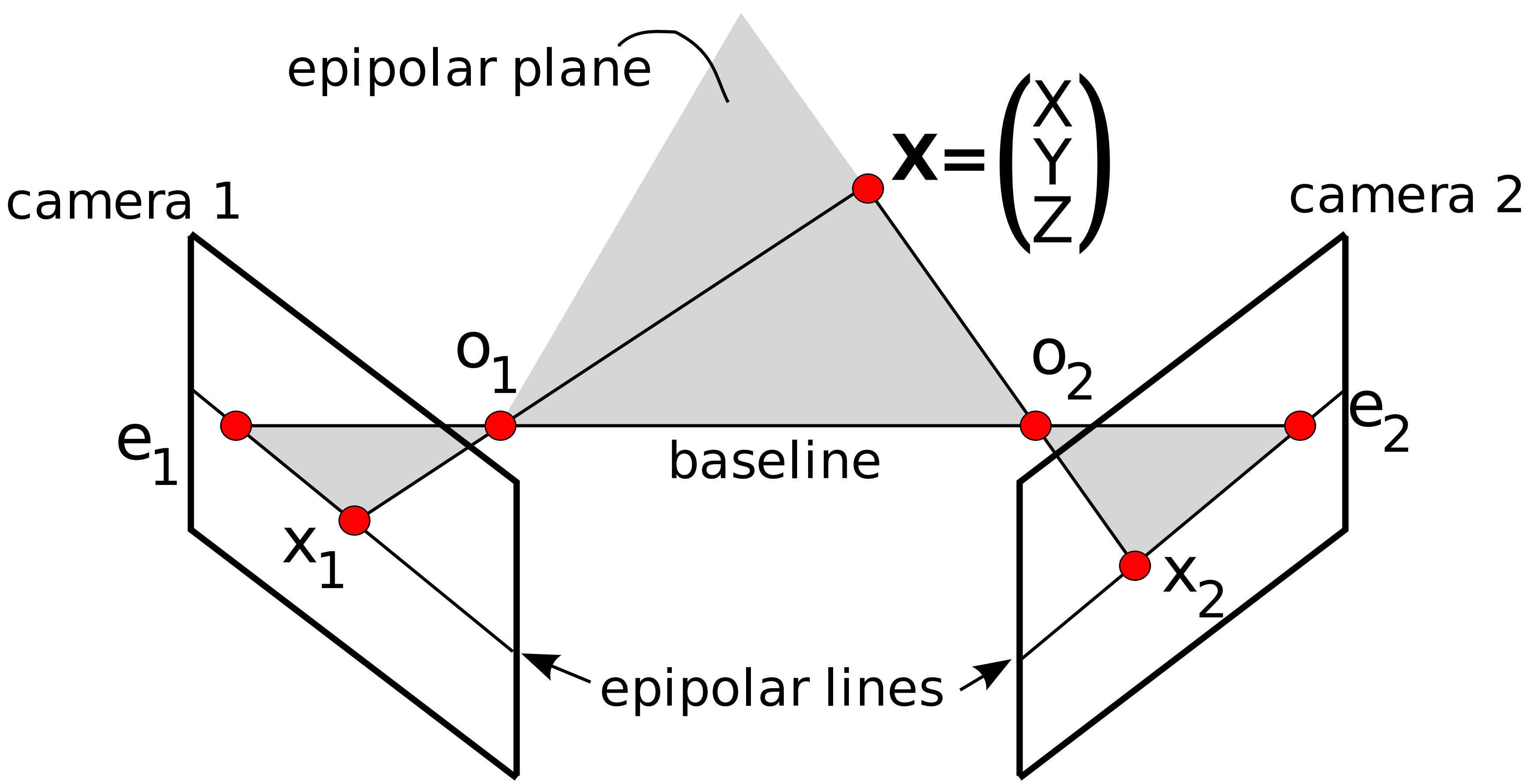
$$\mathbf{x} = \lambda \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}^T \quad \text{with} \quad \lambda > 0$$

and a (hidden) point behind the camera has a projection

$$\mathbf{x} = \lambda \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}^T \quad \text{with} \quad \lambda < 0$$

Oriented epipolar geometry

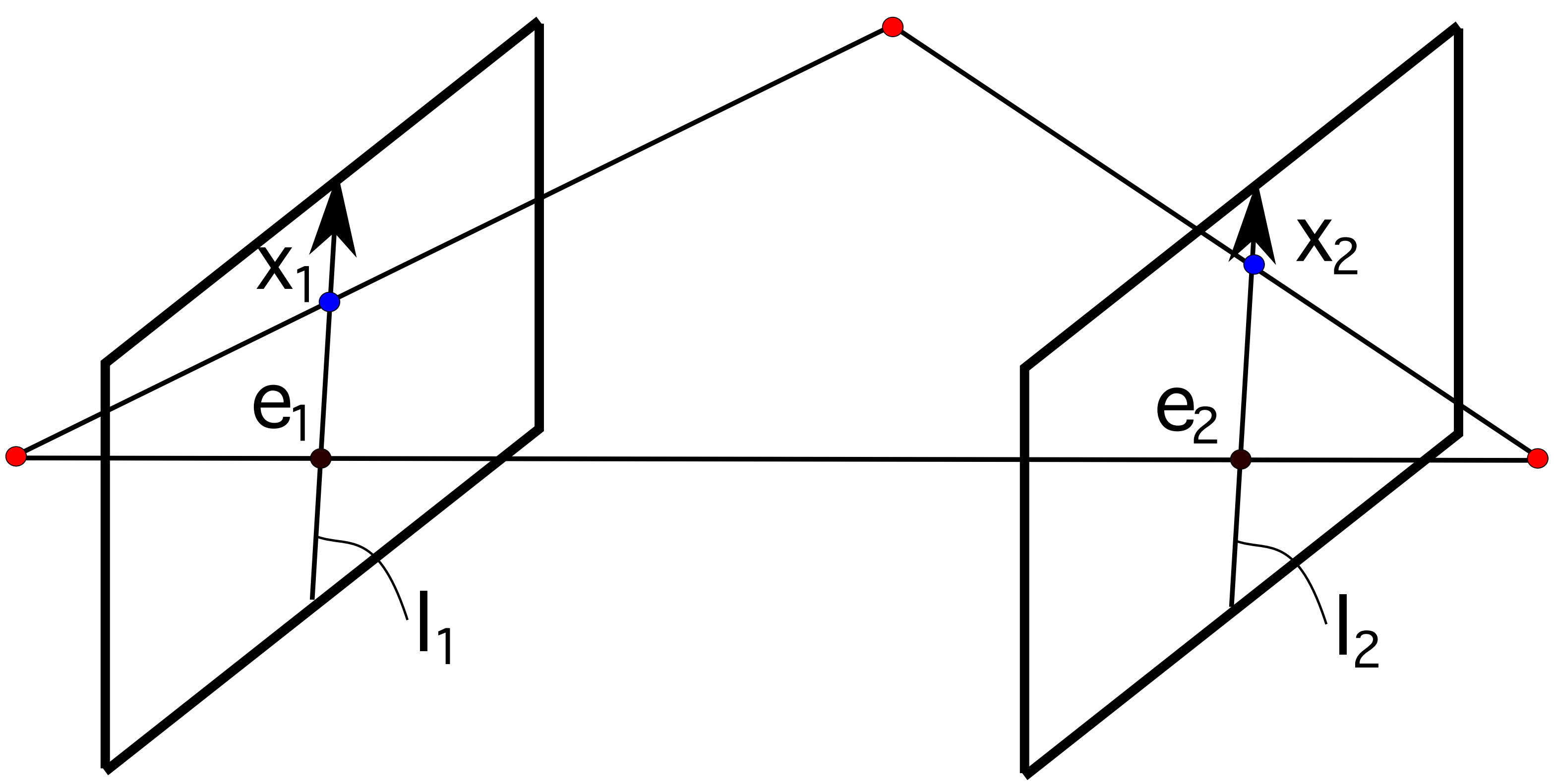
The **oriented epipolar constraint** properly distinguishes points in front of and behind the camera $\lambda \mathbf{e}_1 \times \mathbf{x}_1 = \mathbf{F}\mathbf{x}_2 \ , \quad \lambda \in \mathbb{R}^+$



Oriented epipolar geometry

The **oriented epipolar constraint** can be interpreted as comparing oriented lines

$\lambda \mathbf{e}_1 \times \mathbf{x}_1$ and $\mathbf{F}\mathbf{x}_2$



(NB! image planes drawn in front of cameras)

Oriented epipolar geometry

Line normalisation is not unique

$$\mathrm{norm}_D(\mathbf{l}) = \begin{bmatrix}\cos\alpha & \sin\alpha & -\rho\end{bmatrix}^T$$

$$\mathrm{norm}_D(-\mathbf{l}) = \begin{bmatrix}-\cos\alpha & -\sin\alpha & \rho\end{bmatrix}^T$$

The extra information in the sign can be used to encode the **line orientation**.

Oriented epipolar geometry

Usage:

The oriented epipolar constraint can be used to quickly reject a hypothesized **F** inside a RANSAC loop.

See today’s paper: Chum, Werner and Matas, *Epipolar Geometry Estimation via RANSAC benefits from the Oriented Epipolar Constraint*, ICPR04