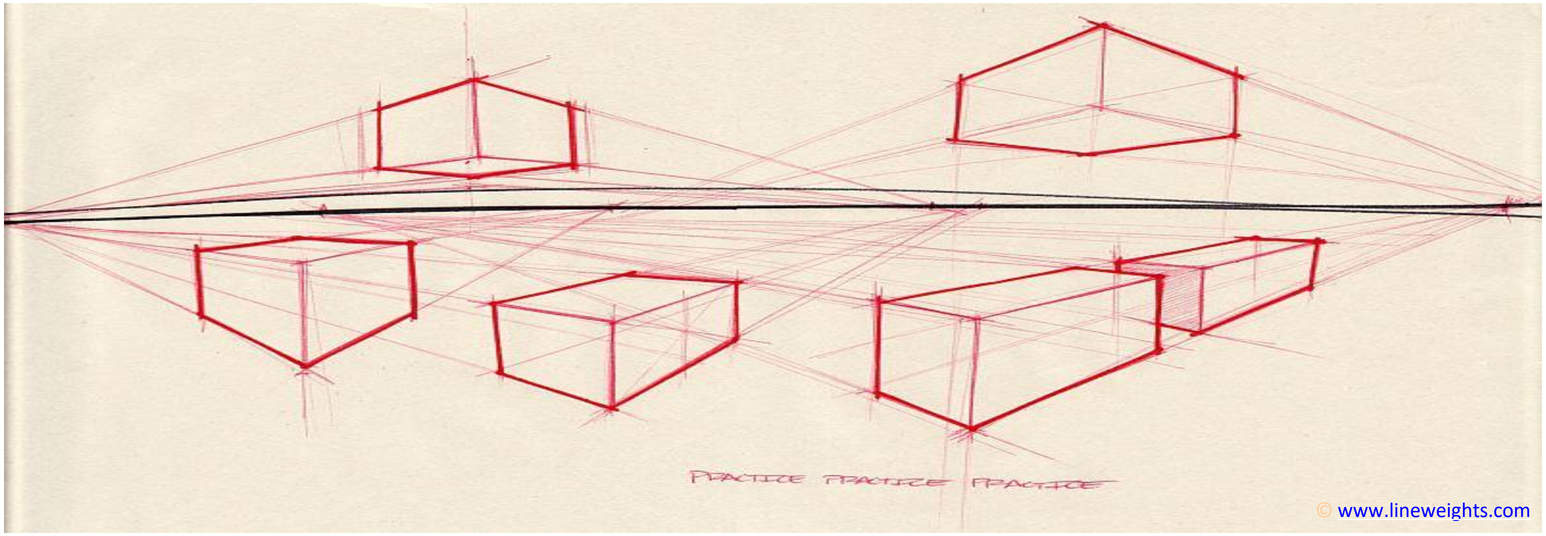


Affine and Projective Geometry
Epipolar Geometry

Computational Geometry/VGE

Bronislav Přibyl, Markéta Dubská, Brno Univ of Tech, 2016

Contents

- Revision
- From Euclidean to Projective Geometry
- Projective Transformations
- Image Formation and Camera Calibration
- Epipolar Geometry

Contents

Homogeneous Coordinates (Revision)

Cartesian to homogeneous

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ (point)} \qquad \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \text{ (vector)}$$

Homogeneous to Cartesian:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \rightarrow \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}, w \neq 0$$

Invariant to multiplication by a constant:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \simeq \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix}, k \neq 0$$

What for?

- unified representation of affine and projective transformations
- transformations can be easily composed by a linear operation (matrix multiplication)

Matrix Transforms

$$\mathbf{x}' = \mathbf{T}\mathbf{x}$$

2D ← 2D

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

3D ← 3D

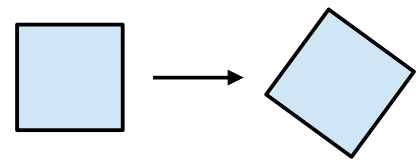
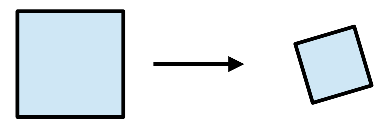
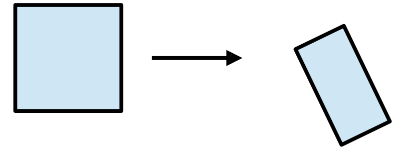
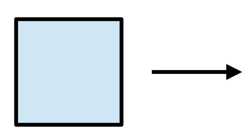
$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

2D ← 3D (projection)

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Contents

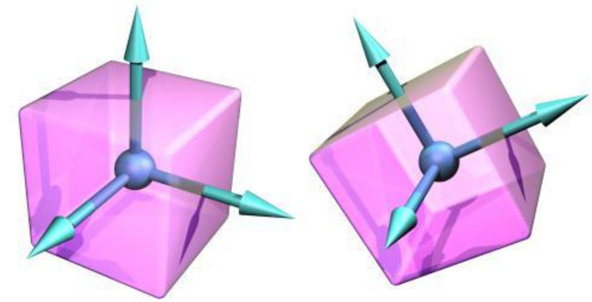
Geometric Transformations (2D)

Transform	Matrix	DoF		Invariants
Euclidean	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	3		lengths, areas, angles, ...
Similarity	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	4		angles, length ratios, parallel lines, ...
Affine	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	6		length ratios **on parallel lines**, parallel lines, area ratios, collinearity, ...
Projective	$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$	8		collinearity, cross ratio

Geometric Transformations

Source: [1, p. 37]	Euclidean	Similarity	Affine	Projective
Partial transformation				
Rotation	X	X	X	X
Translation	X	X	X	X
Uniform scaling		X	X	X
General scaling			X	X
Skew			X	X
Perspective projection				X
Invariants				
Lengths	X			
Angles	X	X		
Length ratios	X	X		
Length ratios on parallel lines	X	X	X	
Parallel lines	X	X	X	
Incidence	X	X	X	X
Cross ratio	X	X	X	X

Euclidean Geometry

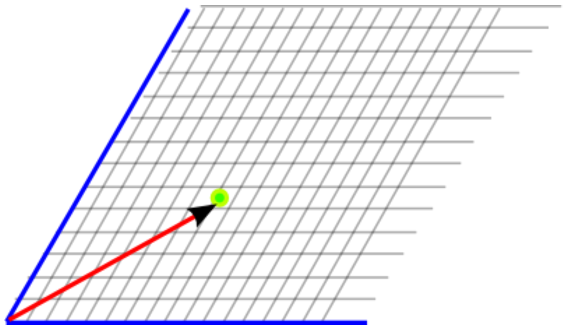


- Geometry that we know "from life"
- Objects considered identical,
  if only their location in space changes
  (by translation and rotation) → "geometry of rigid bodies"
- Transformations keep lengths and angles

Euclidean spaces:
- in 2D: Euclidean Plane = Cartesian Plane = R^2
- in 3D: Euclidean Space = Cartesian Space = R^3

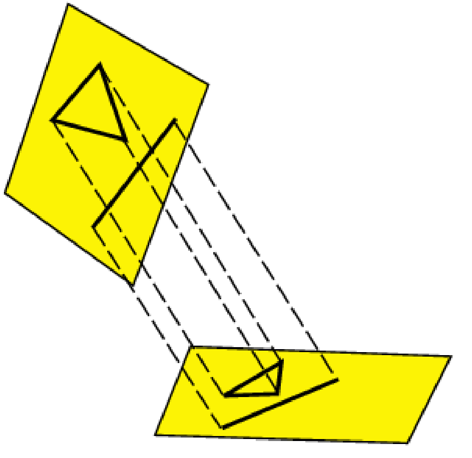
Affine Geometry

- Geometry with only lines and relation of being parallel
- Transformations preserve only length ratios on parallel lines and parallelism (i.e. also area ratios)
- Lengths and angles are **not** preserved



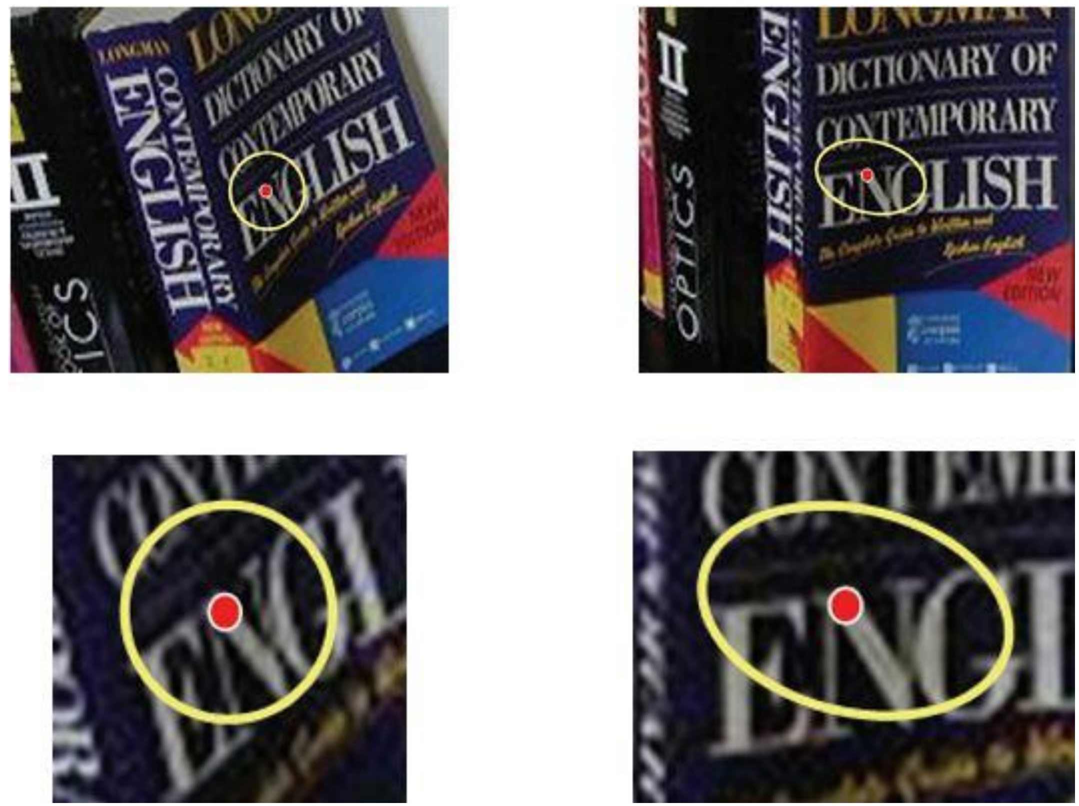
- **Affine plane** – Euclidean plane with no meaning of angles

- Example of Affine transform: parallel projection from plane to plane (sun lights through a pattern on window → casts shadow)



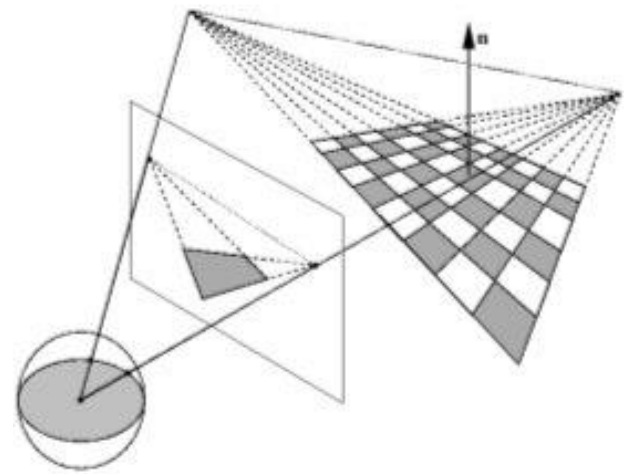
Affine Geometry – example

- Some interest point detectors are affine invariant
- The idea: Projective transforms can be locally approximated by an affine transform
  - MSER
  - ASIFT
  - Harris-Affine
  - Hessian-Affine
  - Laplacian-Affine
  - …



Projective Geometry

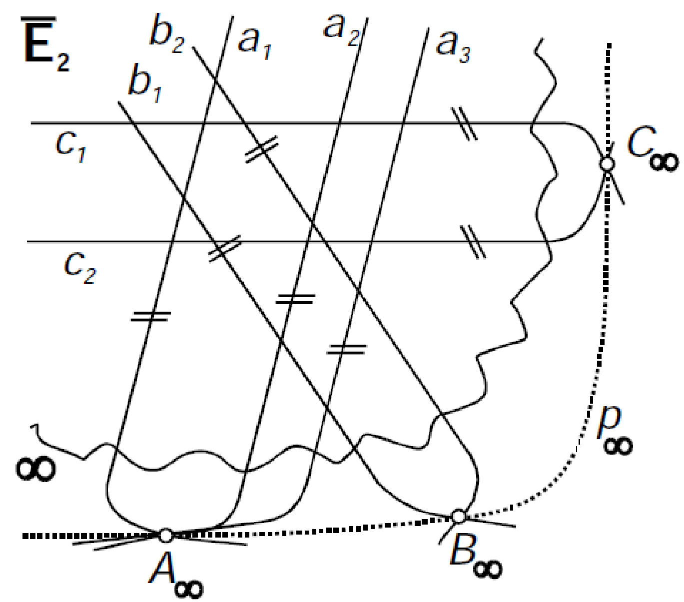
- **Euclidean Geometry** describes objects “as they really are”
- **Projective Geometry** describes objects “as we see them”
- → Describes the image formation process
- Transformations preserve only collinearity
  (lines remain straight) and so called cross-ratio
- Angles, lengths, length raitos
  or even parallelness (!) **are not preserved**



- **2 axioms:**
  I. Every two points are connected by a single line
  II. a) in 2D: Every two lines intersect in a single point
  III. b) in 3D: Every two different planes intersect in a single line

Projective Plane P^2

- Projective Plane = Affine Plane enriched by
  - **a point in infinity** (*ideal point*, $[x, y, 0]$) for every group of parallel lines, which is their intersection, and
  - **a line in infinity** (*ideal line*, $[0, 0, 1]$) which includes all the ideal points
  - These new entities are not different from other points and lines – they are simply points and a line in the plane
- **Every** 2 lines intersect



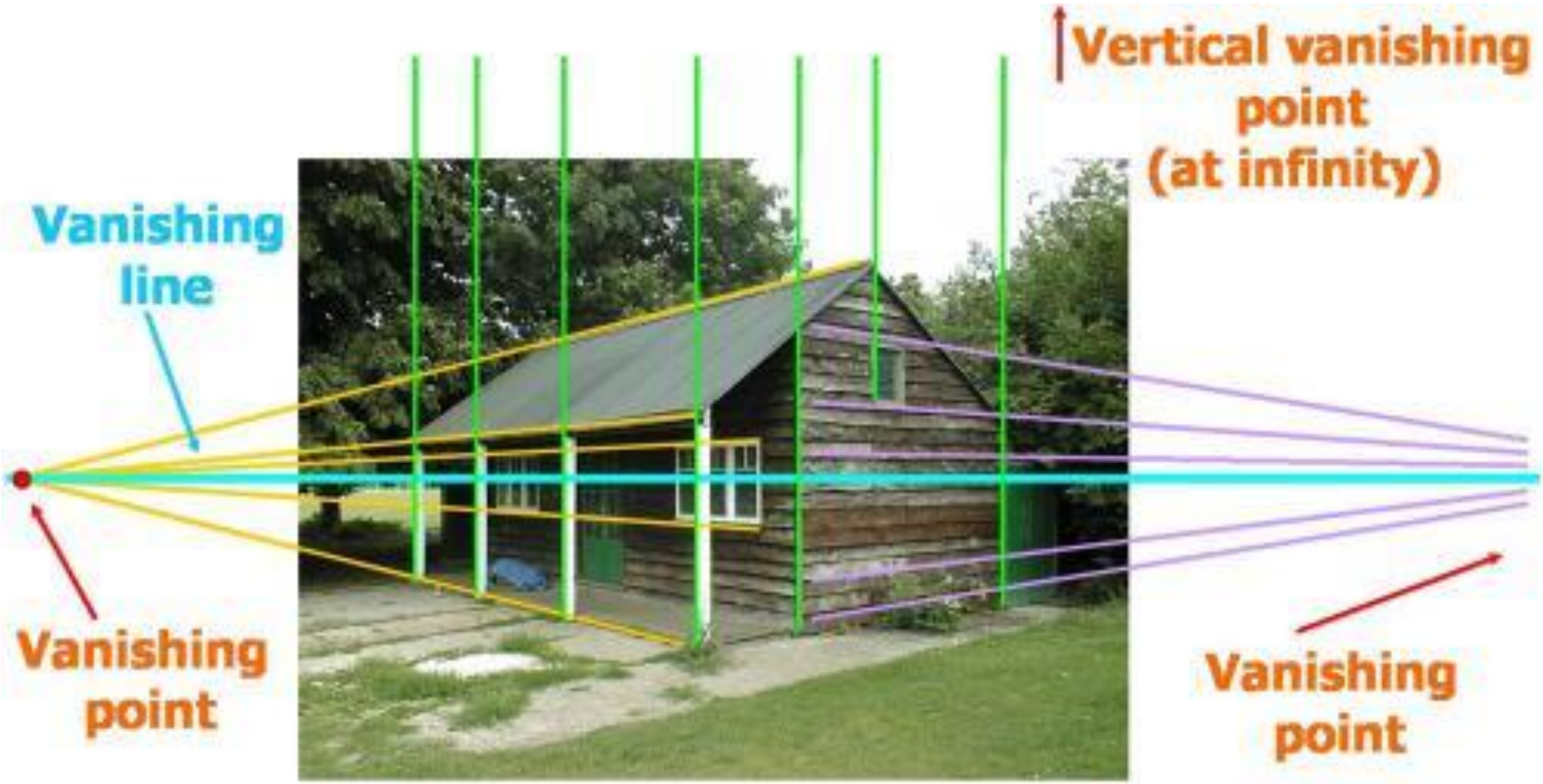
Projective Space P^3

- Projective Space = Affine Space enriched by
  - **a point in infinity** (*ideal point*, $[x, y, z, 0]$)
    for each group of parallel lines, their intersection,
  - **a line in infinity** (*ideal line*)
    for each group of parallel planes, their intersection, and
  - **a plane in infinity** (*ideal plane*, $[0, 0, 0, 1]$),
    which is the combination of all ideal lines (and also points) in infinity
  - These new entities are not different from others
    – they are simply points and lines and a plane in the space

- **Every** 2 planes intersect

Projective Space P^3

- Image of a point in infinity is called **vanishing point**
- Image of a line in infinity is called **vanishing line**

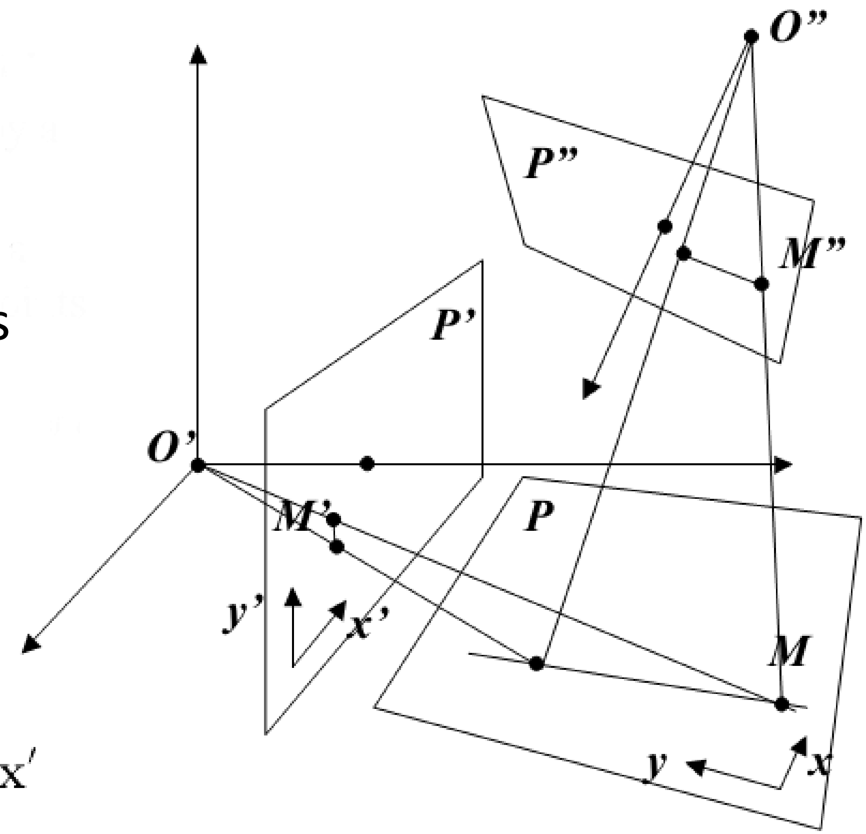


Contents

Projective Transform in a Plane = Homography

- Maps points in one plane again to a plane [1, p. 33]

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

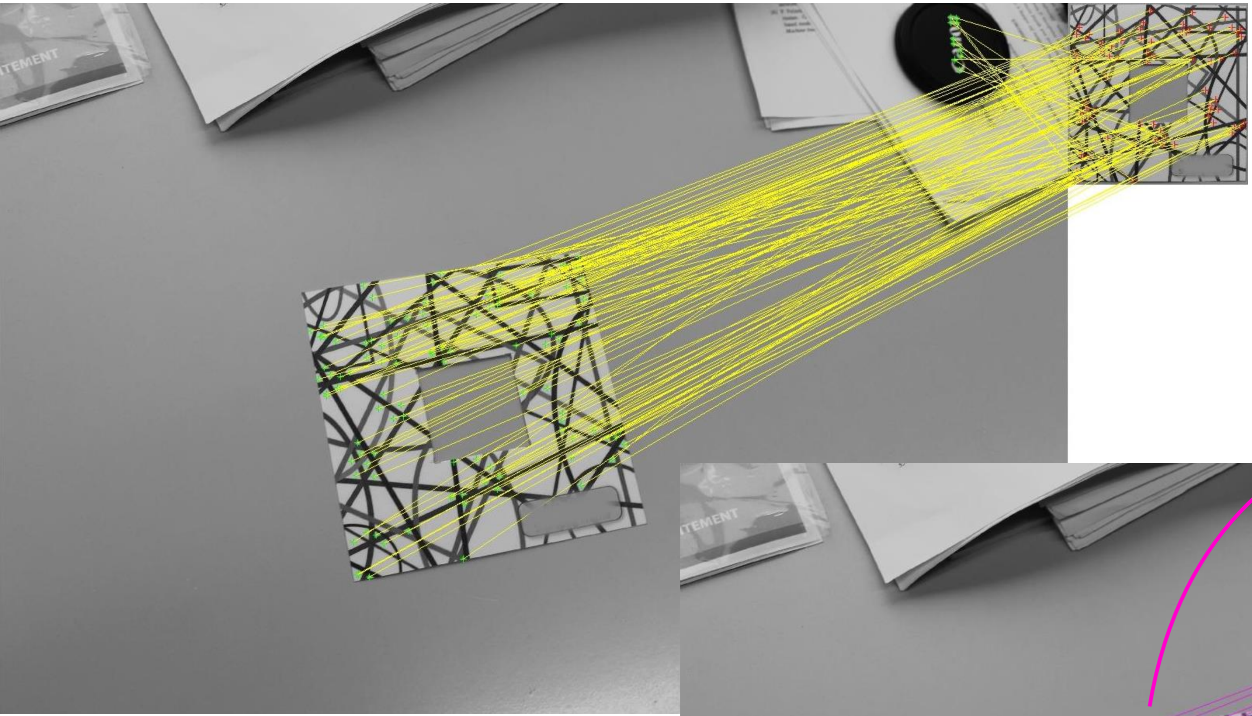


- Matrix $\mathbf{H}$ is homogeneous
  - Important are only length ratios of individual sections
  - It can always be adjusted so that e.g. h_{33}= 1.

$$\mathbf{x}' = \mathbf{H}'\mathbf{x}, \quad \mathbf{x}'' = \mathbf{H}''\mathbf{x} \quad \Rightarrow \quad \mathbf{x}'' = \mathbf{H}''\mathbf{H}'^{-1}\mathbf{x}'$$

$\mathbf{H} = \mathbf{R} - \dfrac{\mathbf{t}\mathbf{n}^\top}{d}$ Homography between 2 cameras (**R**, **t**) induced by a plane with normal **n** [1, s. 327]

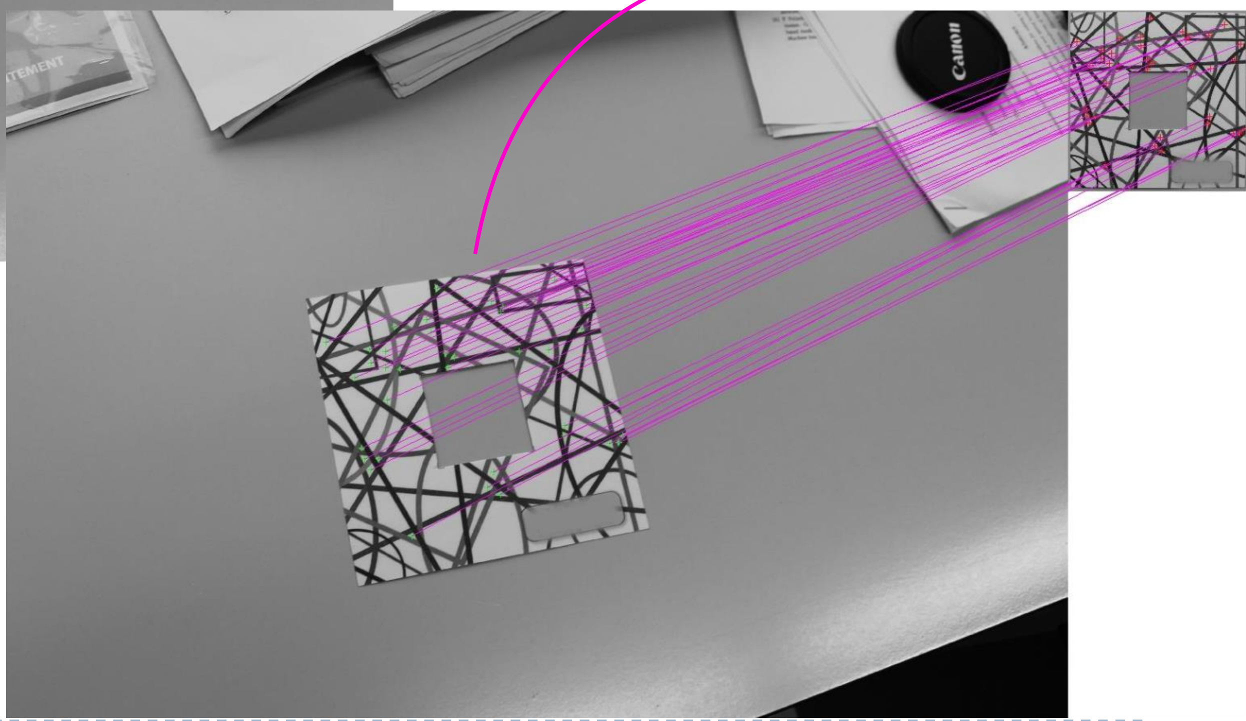
Homography – example



note: photo resolution 1375 x 915 px

$$\mathbf{H} = \begin{bmatrix} 0{,}6222 & -0{,}1125 & -614{,}2588 \\ 0{,}1006 & 0{,}6659 & -886{,}1327 \\ 0{,}0000 & 0{,}0001 & 1{,}0000 \end{bmatrix}$$

Preliminary correspondences of keypoints



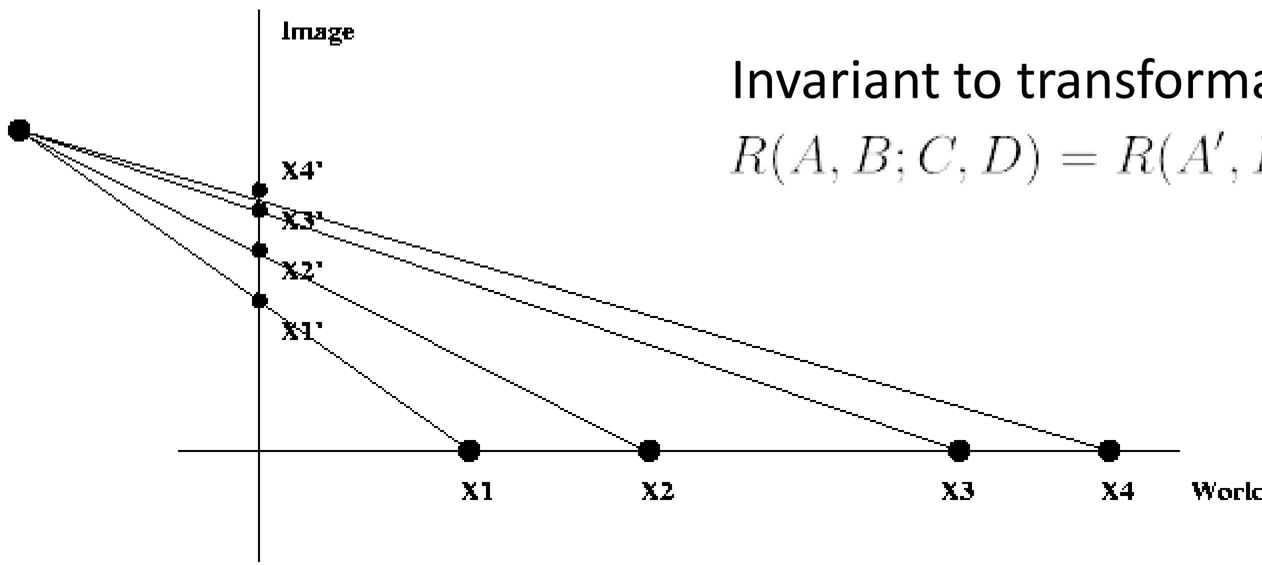
Correct correspondences after filtration of inconsistencies

Cross-Ratio

- or **anharmonic ratio**
- Ratio of distances of collinear points

$$R(A,B;C,D) = \frac{|AC|}{|BC|} \div \frac{|AD|}{|BD|} = \frac{|AC|\cdot|BD|}{|BC|\cdot|AD|}$$

The order of the 4 points can be arbitrary (but must be kept)



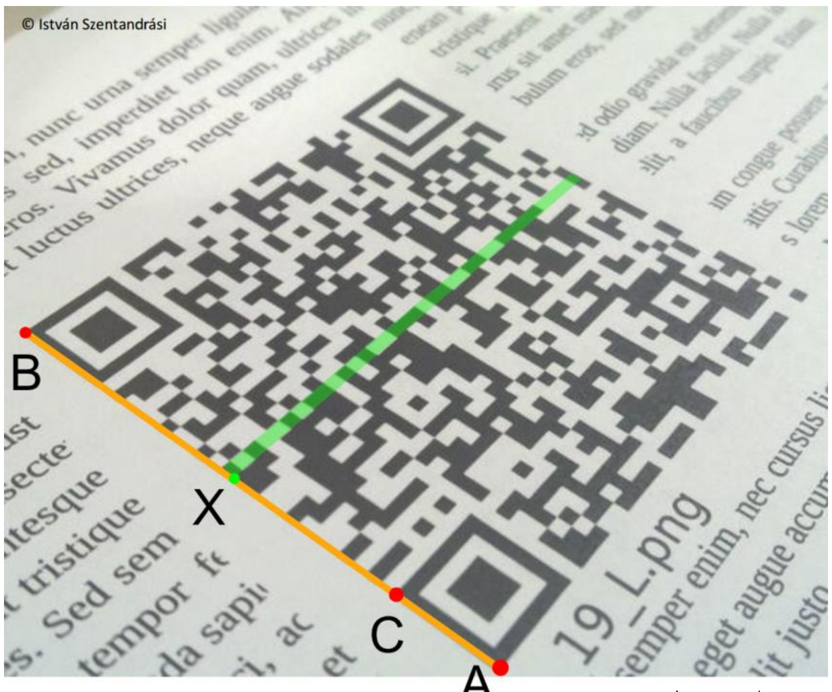
Invariant to transformation! i.e.

$$R(A,B;C,D) = R(A',B';C',D')$$

[1, s. 45]

Cross-Ratio – example

- QR code 37x37 elements, corner markers 7x7 elements.
  We know the screen coordinates of the QR code's corners (A, B).
  On which screen coordinates should we start reading 20th line of the code?
- Real length of the QR code's side $|ab| = 37$ elm.
  Length of the corner marker $|ac| = 7$ elm.
  What is the image distance $|AX|$ of the beginning of the 20th line from point A?



Solution:

- From the point's coordinates a, b, c, x compute the cross-ratio.

$$R(a,b;c,x) = \frac{|ac| \cdot |bx|}{|bc| \cdot |ax|} = \frac{7 \cdot 17}{30 \cdot 20} = 0,1983$$

- The same ratio must hold also for points *A, B, C, X* in the image, i.e.

$$R(a,b;c,x) = R(A,B;C,X) = 0,1983$$

$$0,1983 = \frac{|AC| \cdot |BX|}{|BC| \cdot |AX|} = \frac{|AC| \cdot (|AB| - |AX|)}{|BC| \cdot |AX|}$$

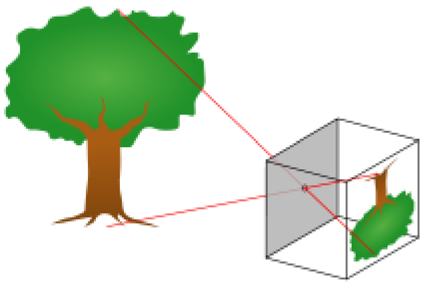
$$|AX| = \frac{|AC| \cdot |AB|}{\left(0,1983 + \frac{|AC|}{|BC|}\right) \cdot |BC|} = \frac{93 \cdot 423}{\left(0,1983 + \frac{93}{329}\right) \cdot 329} \doteq 249\,\mathrm{px}$$

Contents

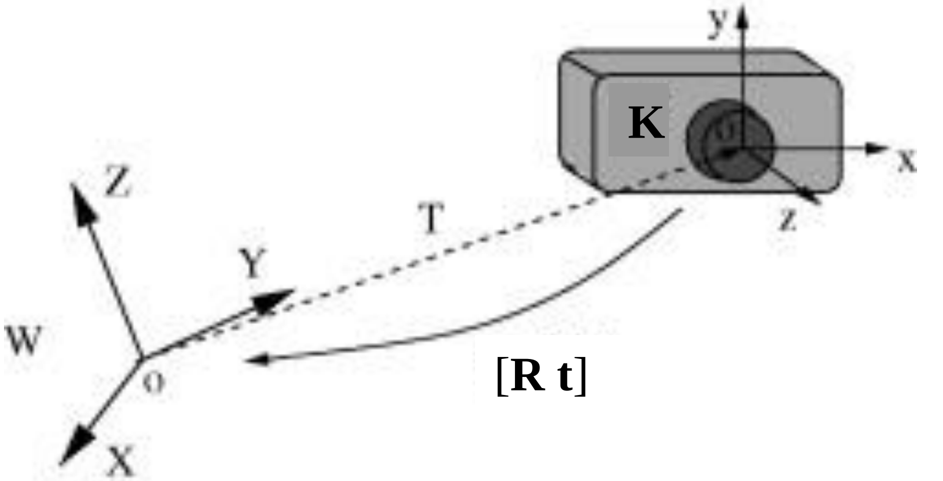
Image Formation

- Consider a pinhole camera
- Perspective (central) projection of the 3D world to 2D image:

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



- Outer camera parameters
  - $\mathbf{R}$ – rotation matrix
  - $\mathbf{t}$ – translation vector
- Inner camera parameters
  - $\mathbf{K}$ – camera matrix



Camera Calibration

- photometric calibration = mapping of colors
- **geometric calibration** (*camera resectioning*)

= determining internal and external camera parameters

$$\mathbf{x} = \overbrace{K}^{\text{internal}}\ \overbrace{[R\ \mathbf{t}]}^{\text{external}}\ \mathbf{X}$$

rotation matrix

translation vector

$$\overbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}^{\text{2D point in image coordinates}} = \underbrace{\overbrace{\begin{bmatrix} f & k & u_0 \\ 0 & rf & v_0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{camera matrix}} \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}^{\text{transformation matrix}}}_{\text{camera calibration matrix}} \overbrace{\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}}^{\text{3D point in scene coordinates}}$$

relativní ohnisková vzdálenost [px]

poměr stran pixelů

zešikmení pixelů = $r f (\tan \alpha)$

souřadnice hlavního bodu (principal point)

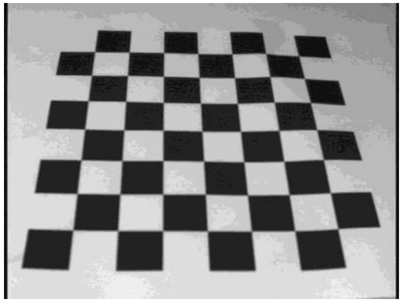
$$\mathbf{x} = P\mathbf{X}$$

[1, p. 153]

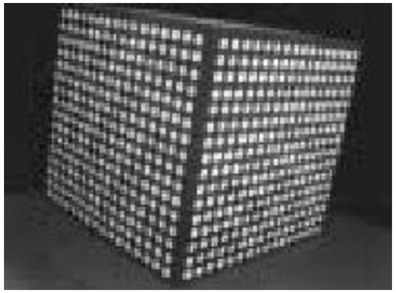
Camera Calibration - algorithms

[1, p. 178]

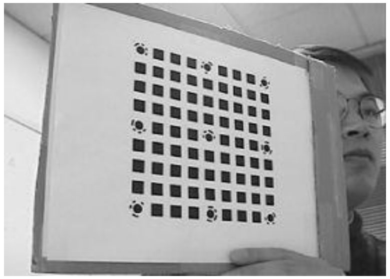
- General principle:
  - Capture an object of known dimensions (usually a checkerboard), and
  - based on correspondences of 3D coords on the object and their 2D projections in the image estimate internal + external camera parameters



- Algorithms for estimation of cam parameters:
  - R. Y. **Tsai** (1986): *An efficient and accurate camera calib. technique*
  - J. **Heikkila**, O. Silven (1997): *A four-step camera calibration procedure with implicit image correction*

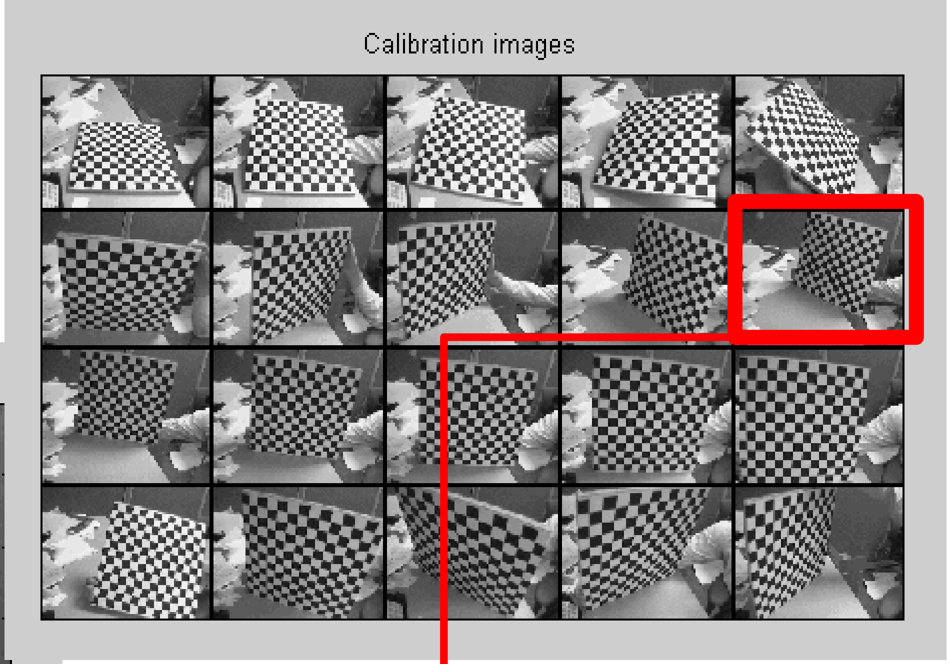


  - Z. **Zhang** (2000): *A flexible new technique for camera calib.*



Camera Calibration – example

Camera Calibration Toolbox for Matlab



Set of images with calibration pattern

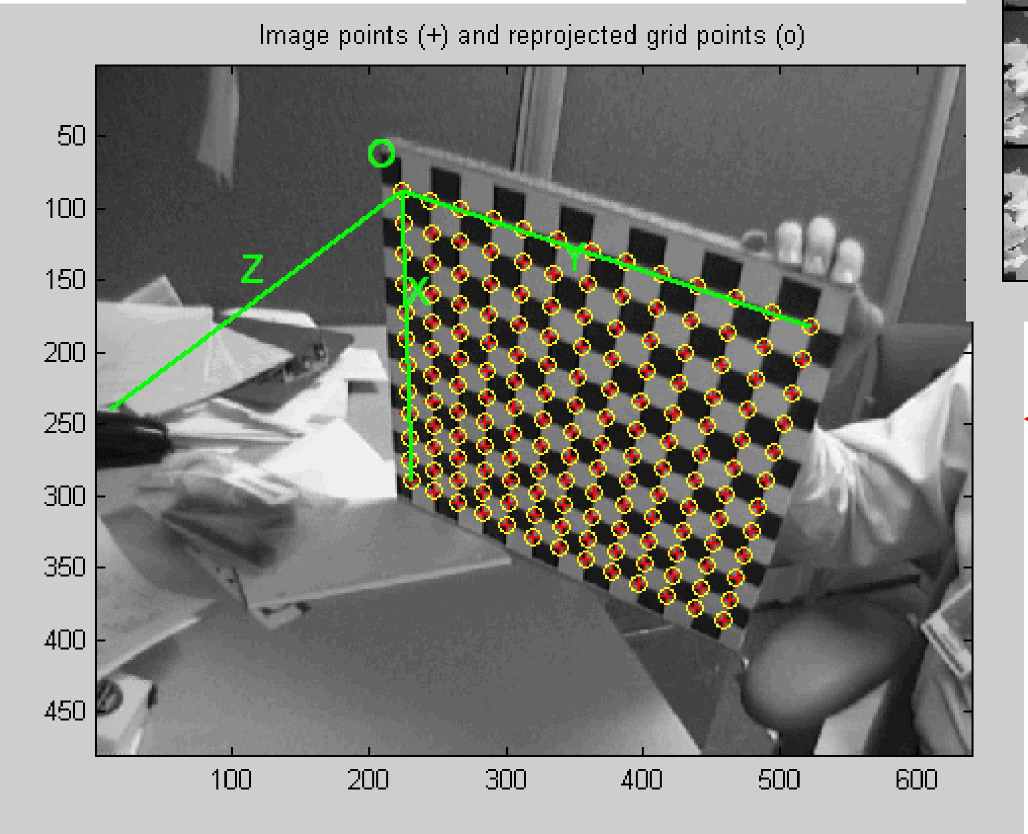
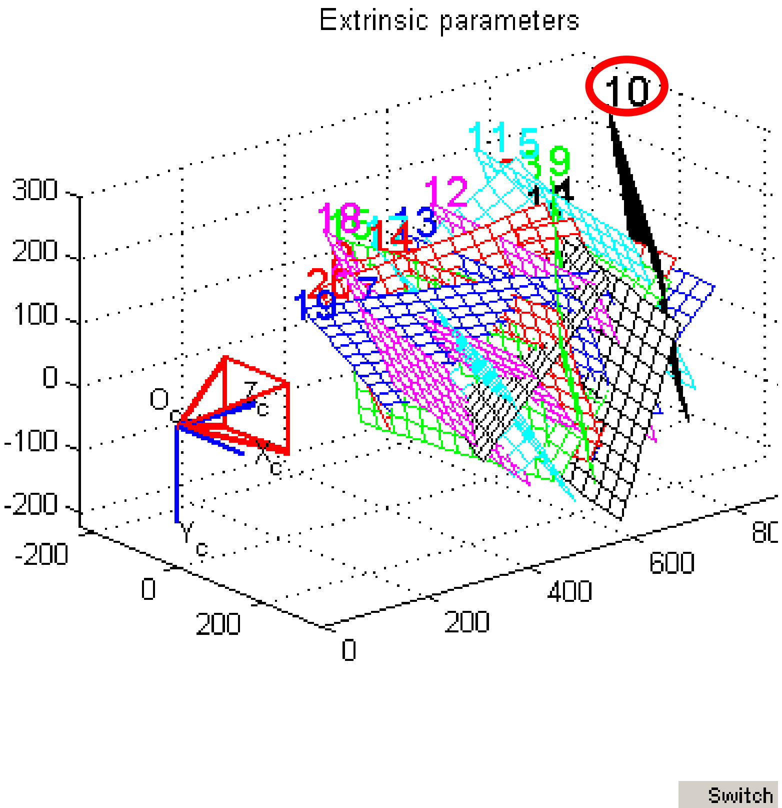
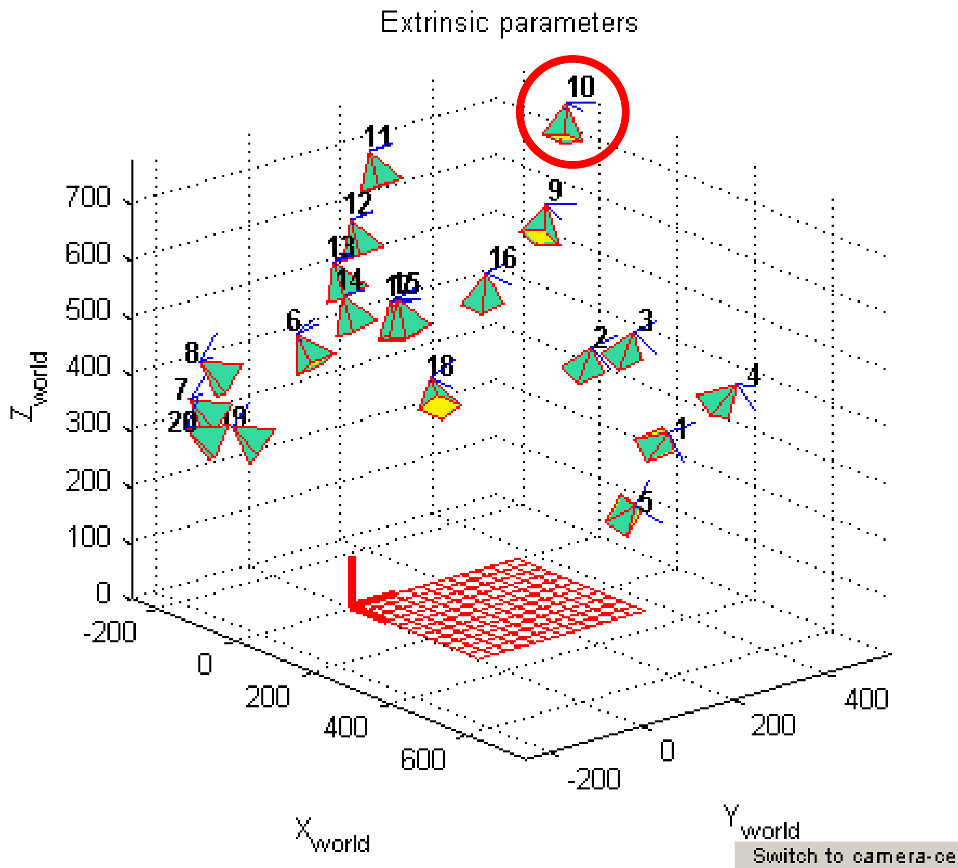


Image no. 10 with detected corner points

Camera Calibration – example

$$s\mathbf{x} = \begin{bmatrix} 665,7 & 0 & 319,5 \\ 0 & 665,7 & 239,5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0,044 & 0,086 & -0,480 & -94,6 \\ 0,766 & 0,338 & 0,547 & -184,0 \\ 0,641 & -0,344 & -0,686 & 766,2 \end{bmatrix} \mathbf{X}$$





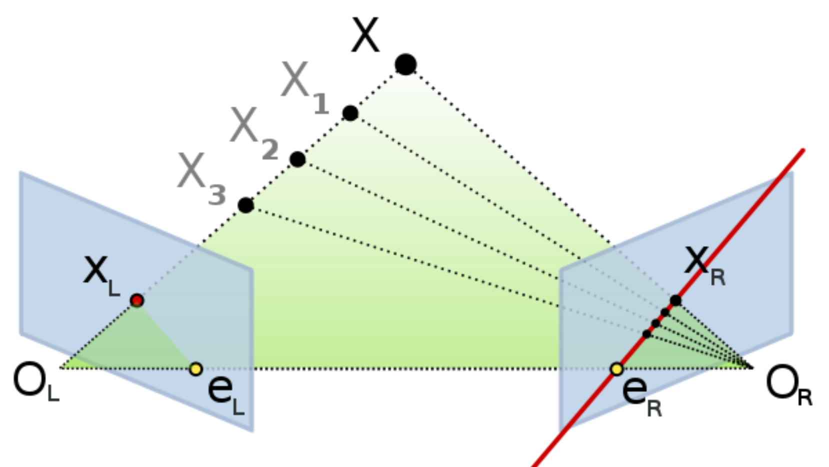
Contents

Epipolar Geometry

[1, s. 239]

- Geometry of **two views** (two different images of the same scene)
- If the mutual positions of the two cameras are not known, they can be computed
- It is possible to find the positions of the points in the 3D space:
  - Find corresponding points in the two images,
  - cast rays through these points,
  - real points are at the intersections of the rays.

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



Left view

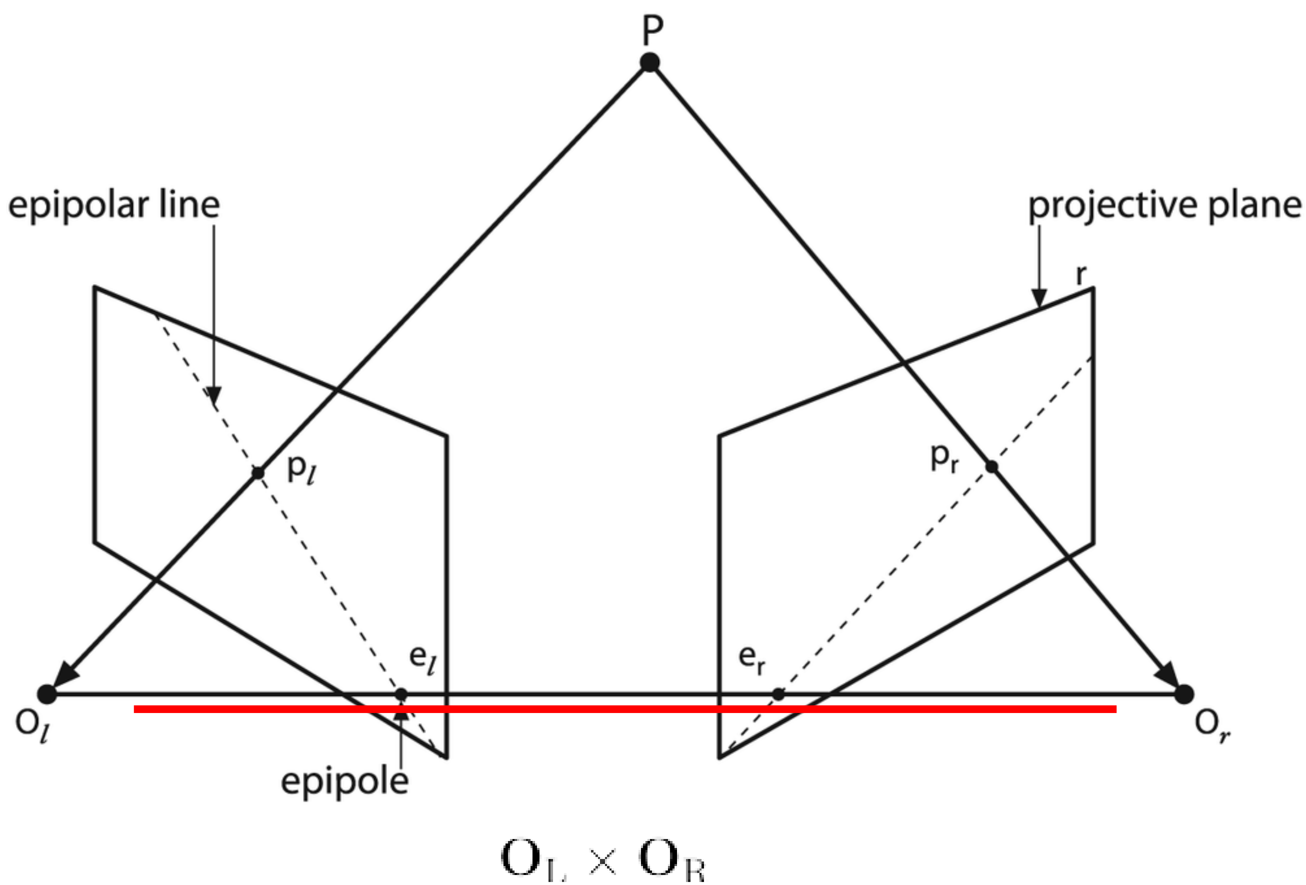
Right view

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

Epipolar Geometry – basics

Baseline

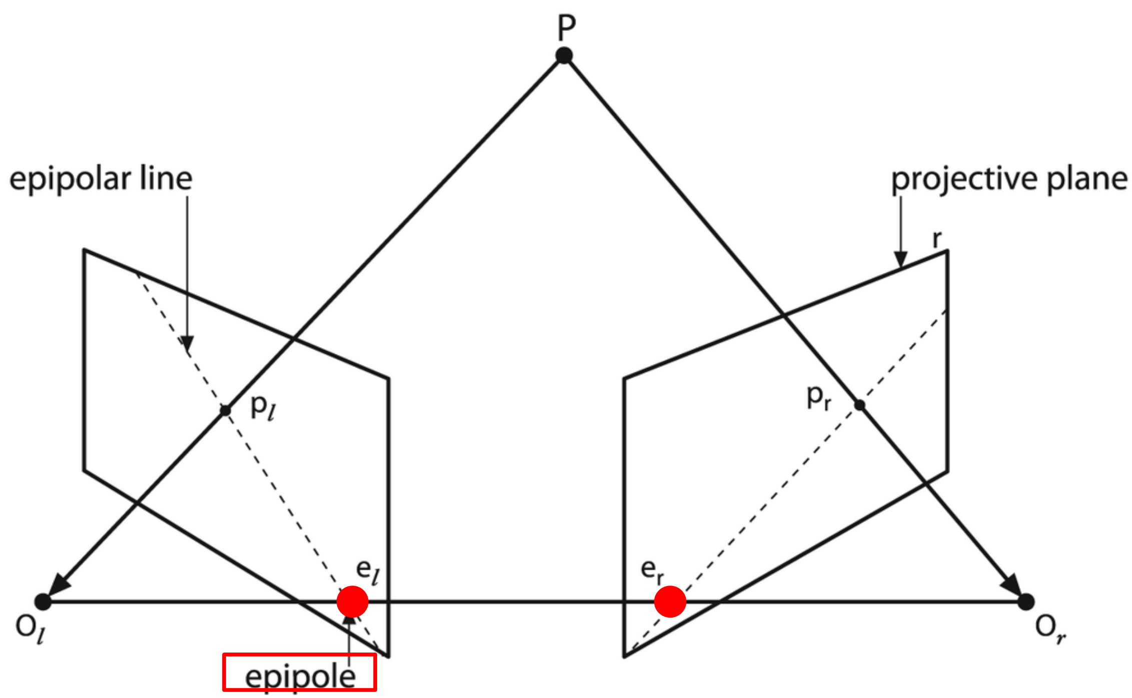
- connects the centers of projection of both cameras



Epipolar Geometry – basics

Epipole – $\mathbf{e}_L$, $\mathbf{e}_R$

- Projection of the center of projection of the other camera to the image plane

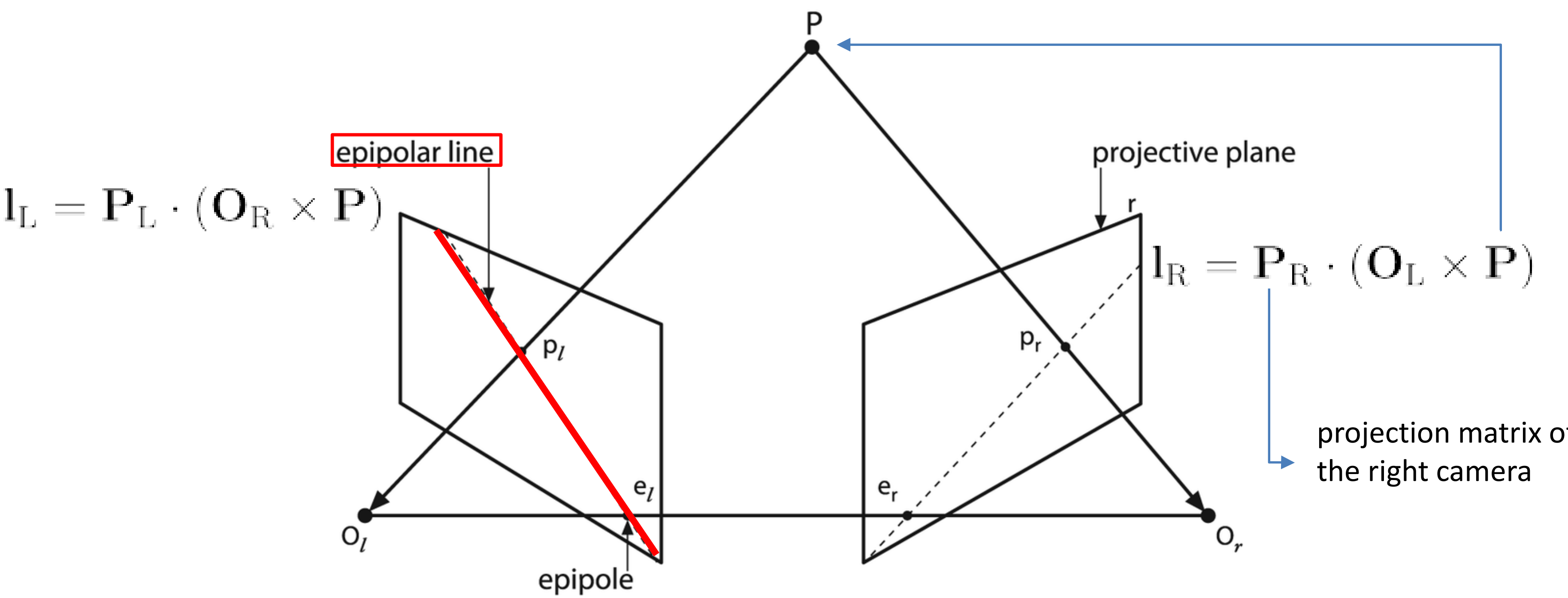


$$e_L = P_L O_R \qquad e_R = P_R O_L$$

Epipolar Geometry – basics

Epipolar line – $\mathbf{l}_L$, $\mathbf{l}_R$

- Epipolar line $\mathbf{l}$ of point $\mathbf{P}$ is the projection of the ray which connects the center of projection $\mathbf{O}$ of the other camera with point $\mathbf{P}$, to the image plane
- Projection $\mathbf{p}$ of point $\mathbf{P}$ always lays on the epipolar line $\mathbf{l}$
- Epipolar line always passes through the epipole (→ epipole is the intersection of all epipolar lines)

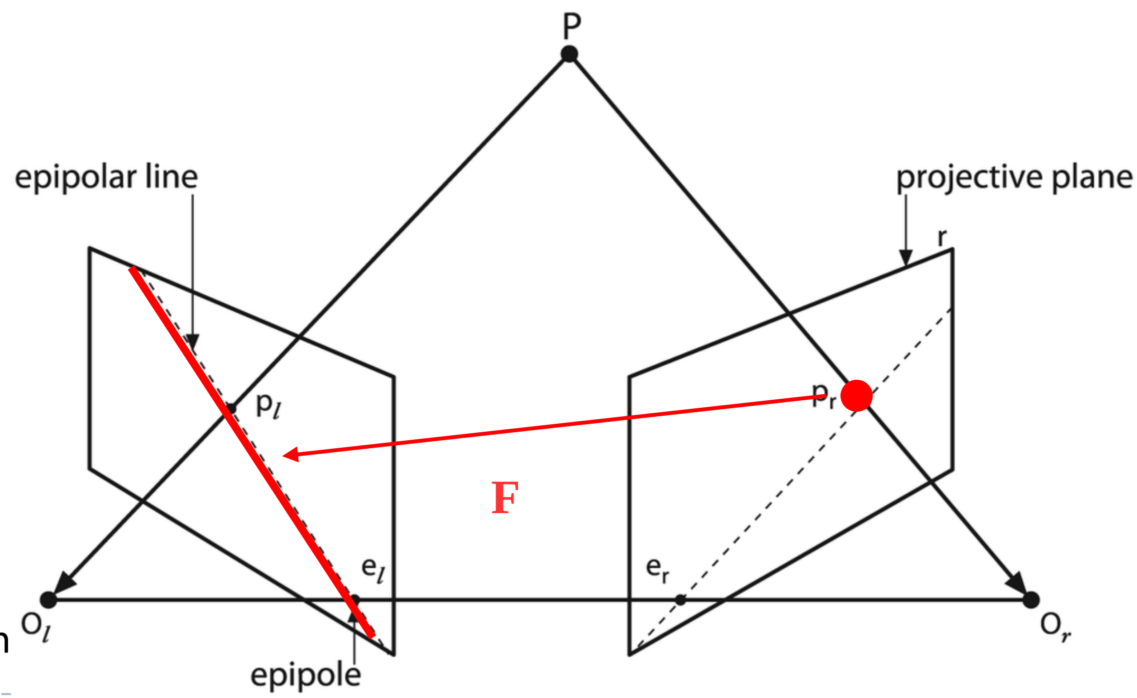


Fundamental Matrix F

- Maps points on lines (points from one image to epipolar lines in the other image)
- Matrix 3x3, order 2, $\det(\mathbf{F}) = 0$.
- 7 degrees of freedom (2x2 parameters for epipoles + 3 parameters for mapping points on the epipolar lines)
- Can be computed as:

$$\mathbf{F} = \begin{bmatrix} 0 & -e_{Lz} & e_{Ly} \\ e_{Lz} & 0 & -e_{Lx} \\ -e_{Ly} & e_{Lx} & 0 \end{bmatrix} \mathrm{P_L P_R^+}$$

$$\mathbf{F} = \mathbf{e}_{L\times} \mathrm{P_L P_R^+}$$

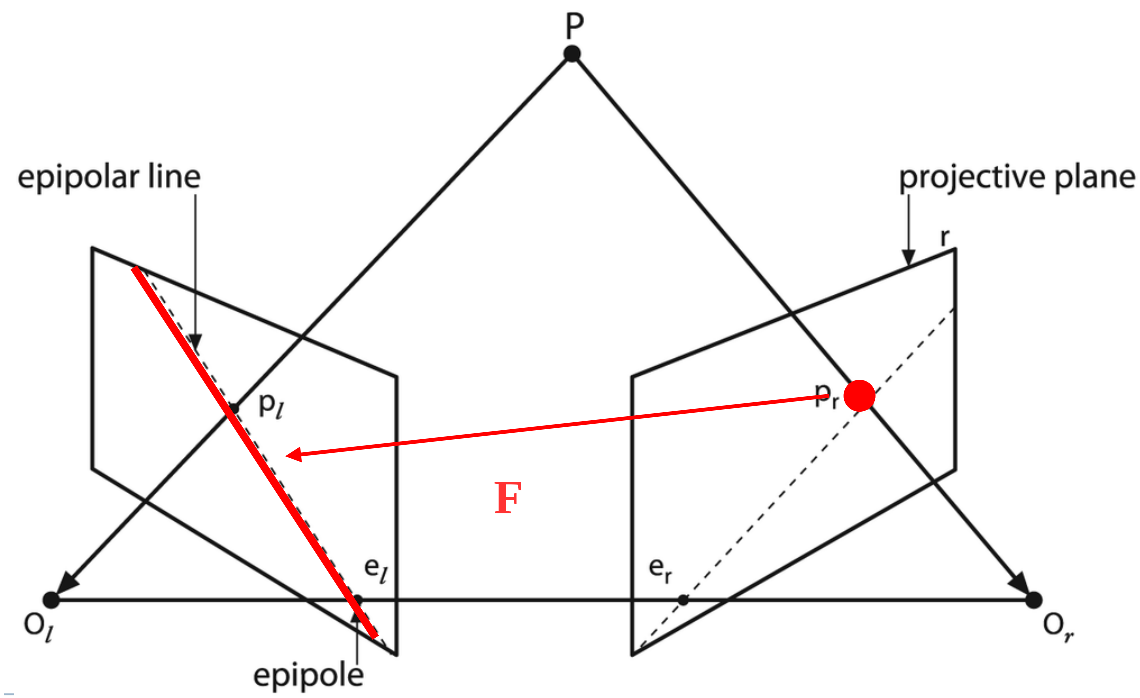


note: $^+$ denotes matrix pseudoinversion

Fundamental matrix F

- If $\mathbf{F}$ is fundamental matrix for $\mathbf{P}_{\mathrm{L}}$,$\mathbf{P}_{\mathrm{R}}$,
  then $\mathbf{F}^{\mathrm{T}}$ is the fundamental matrix for $\mathbf{P}_{\mathrm{R}}$,$\mathbf{P}_{\mathrm{L}}$
- For each pair of corresponding points: $\mathbf{x}_{\mathrm{L}}^{T}\mathbf{F}\mathbf{x}_{\mathrm{R}} = 0$
- Epipolar lines:

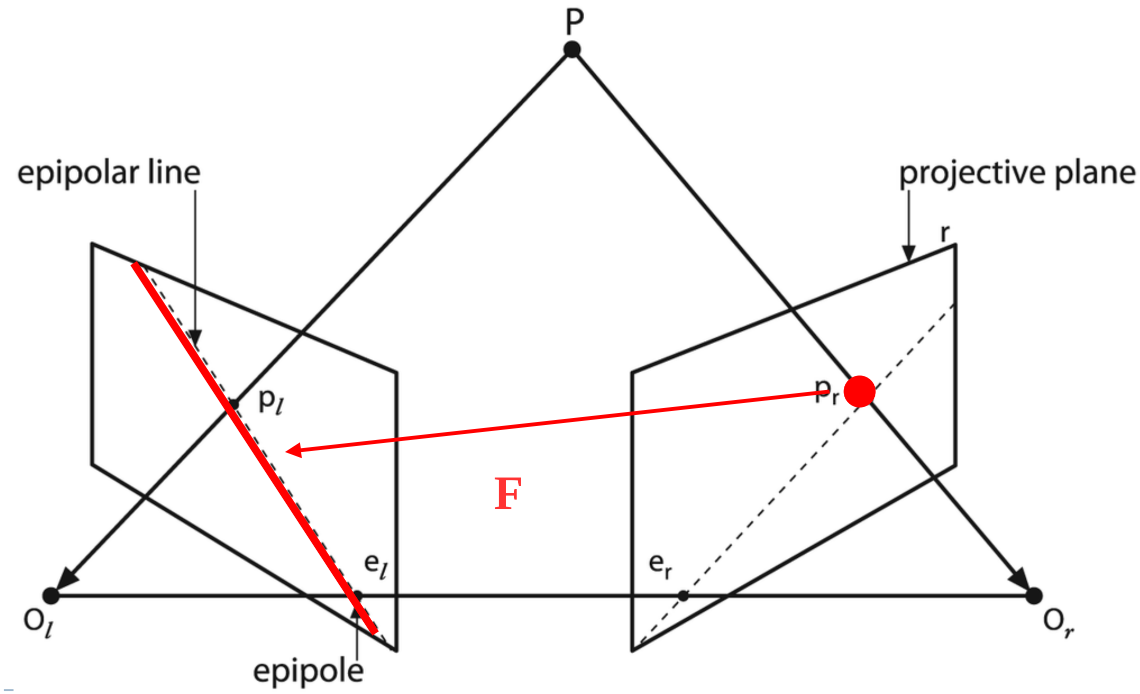
$$\mathbf{l}_{\mathrm{L}} = \mathbf{F}\mathbf{x}_{\mathrm{R}} \quad \mathbf{l}_{\mathrm{R}} = \mathbf{F}^{T}\mathbf{x}_{\mathrm{L}}$$



Fundamental matrix F

Computing $\mathbf{F}$ when we don't have calibrated cameras
(i.e. we don't know $\mathrm{P_R}$ nor $\mathrm{P_L}$):

- From corresponding points in the image
- $\mathbf{F}$ has 7 degrees of freedom, i.e.
  - 7 point pairs are ok, but need to used non-linear constraint $\det(\mathbf{F}) = 0$
  - or 8 point pairs → system of linear equations $\mathbf{x}_\mathrm{L}^T \mathbf{F} \mathbf{x}_\mathrm{R} = 0$



Essential Matrix E

- Special case of $\mathbf{F}$ in case when image coords are *normalized* (cameras must be calibrated) *:*
  - decompose $\mathbf{P}$: $\mathbf{P} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}]$,
  - instead of image coords $\mathbf{x} = \mathbf{PX}$, use coordinates $\mathbf{x}' = \mathbf{K}^{-1}\mathbf{x}$ (like if the camera is $\mathbf{K} = \mathbf{I}$).
- The same properties like $\mathbf{F}$, except that:
  - it has only 5 degrees of freedom

Sources

1 Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2. vydání, 2004. ISBN 9780521540513.

2 Birchfield, S.: *An Introduction to Projective Geometry (for computer vision)*. 1998. <http://robotics.stanford.edu/~birch/projective/>

3 Wildberger, N. J.: *Rational Trigonometry*. CosmoLearning, online video kurz. Přednášky č. 34 – 39.

4 <http://www.cosmolearning.com/courses/rational-trigonometry-551/>

5 Zisserman, A.: *MATLAB Functions for Multiple View Geometry*.

6 <http://www.robots.ox.ac.uk/~vgg/hzbook/code/>