INFO - H - 501

Pattern recognition and image analysis

Vision

Stereovision

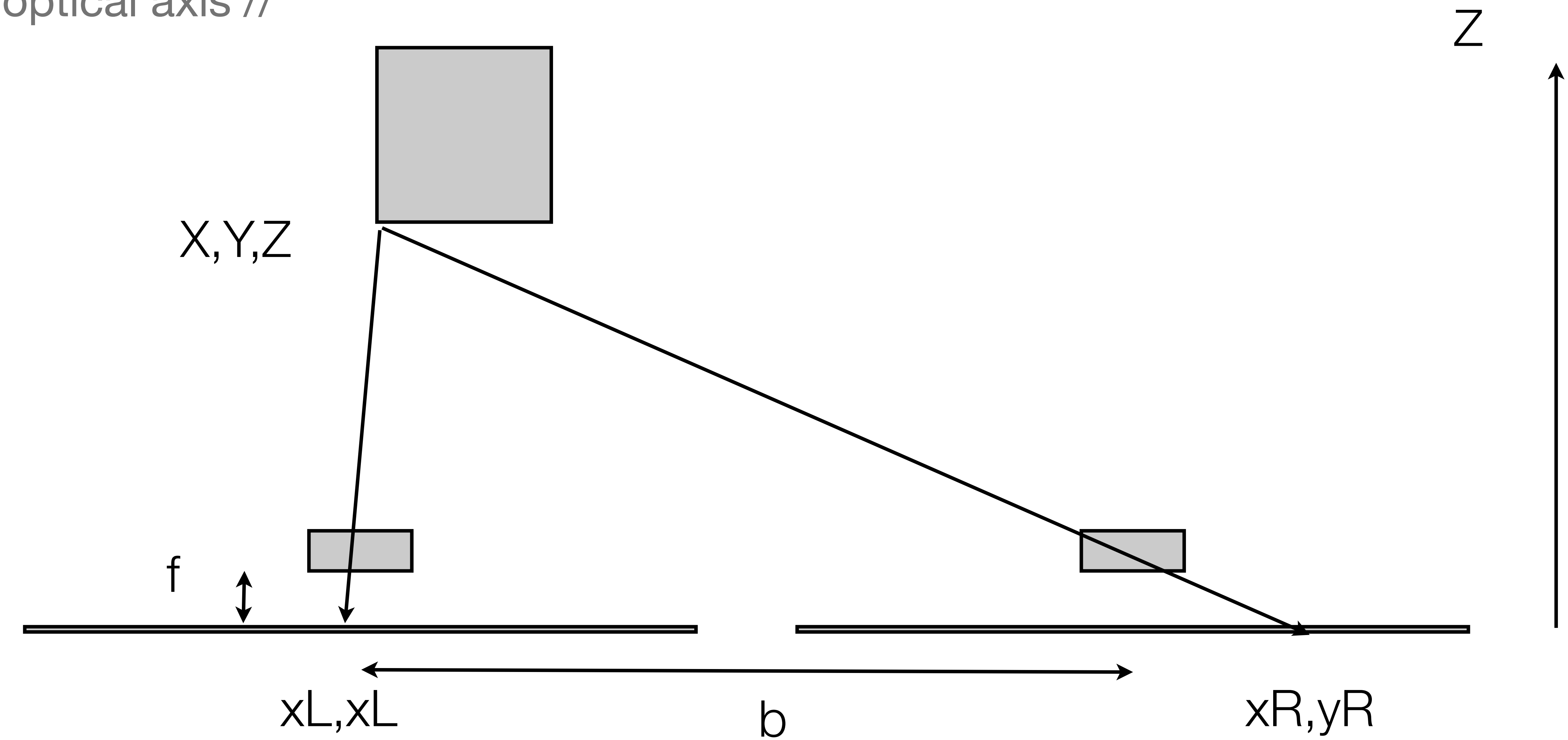
- digital elevation model
- obstacle avoidance
- 3D model scanner
- human machine interface (HMI)
- ...

Stereovision

- image of the same point seen by two cameras = conjugated pair
- both points are on a same line : epipolar line
- if axes are // and both images share a same base
- epipolar are // to X (special case)

Ideal case

- optical axis //



Ideal case

- optical axis //
- P global space (X,Y,Z)
- common base line, distance between origins = b , yL=yR
- left (l) and right (r) projection coordinate
  (xL,yL,zL) = (X-b/2, Y, Z) et (xR,yR,zR) = (X+b/2, Y, Z)
  xL = (X+b/2)f/Z
  xR = (X-b/2)f/Z

  where

  Z = bf/(xL- xR)

  and

  X = b(xL+ xR)/2(xL-xR)
  Y = by/(xL-xR)

Disparity

disparity:

$d = xL - xR$

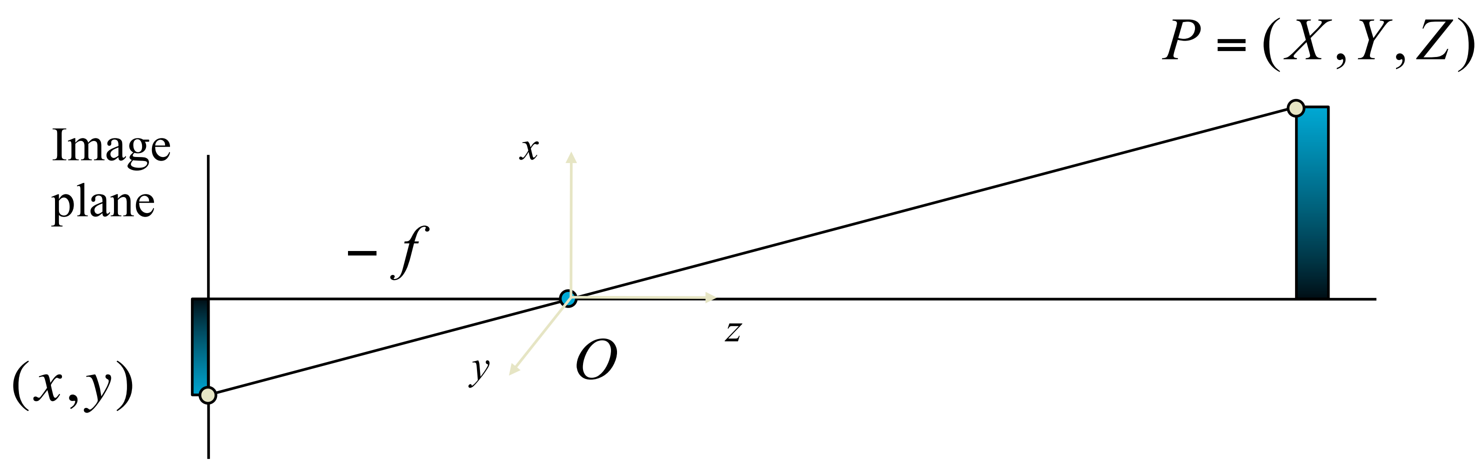
X,Y, Z position is given by

$X = (b[xR + xL]/2)/d$
$Y = by/d$
$Z = bf/d$

Disparity

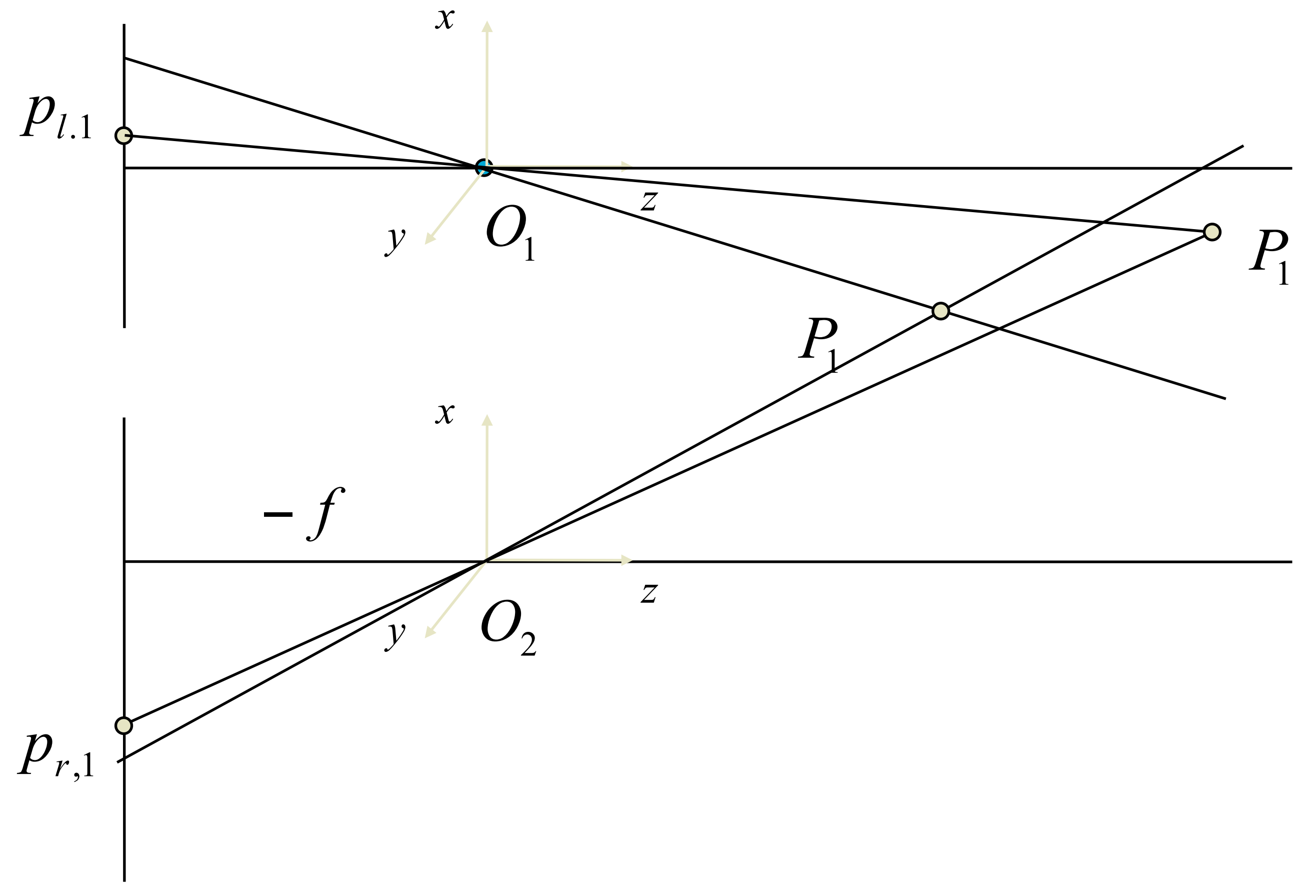
- Distance inversely proportional to disparity
- d = 0 iif P inf.
- Disparity proportional to b
  - b>> better accuracy
  - b>> smaler coverage

Pinhole camera

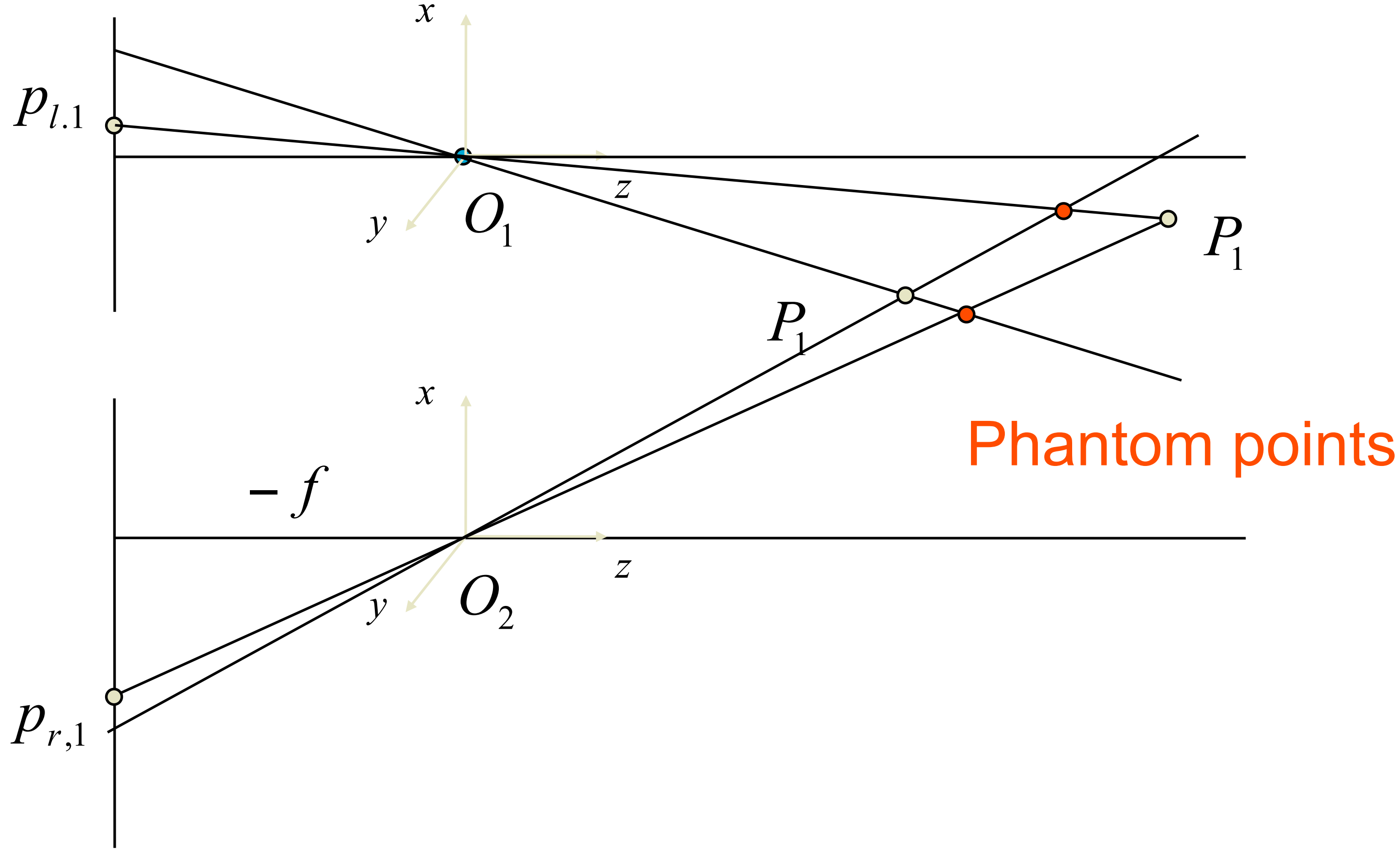


$$(x, y) = (f\frac{X}{Z}, f\frac{Y}{Z})$$

False matching case



False matching case



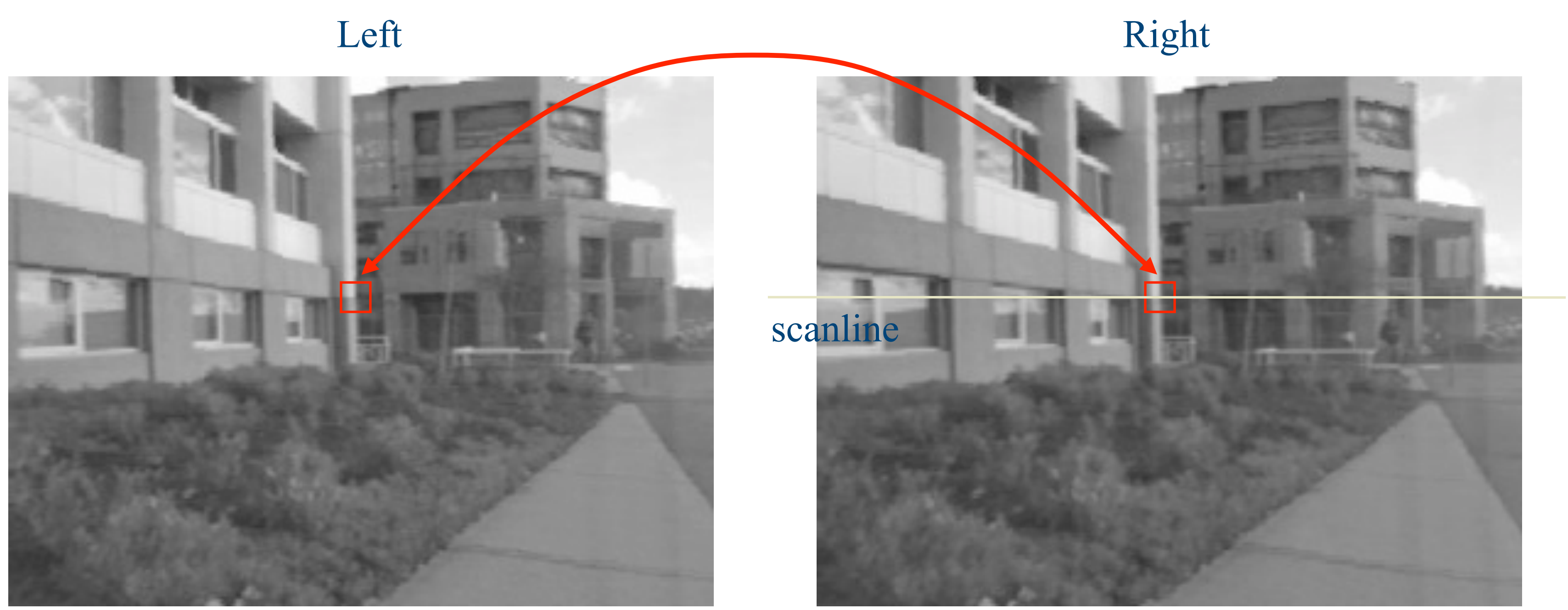
Point matching

- Correlation

Rectified images

Point matching

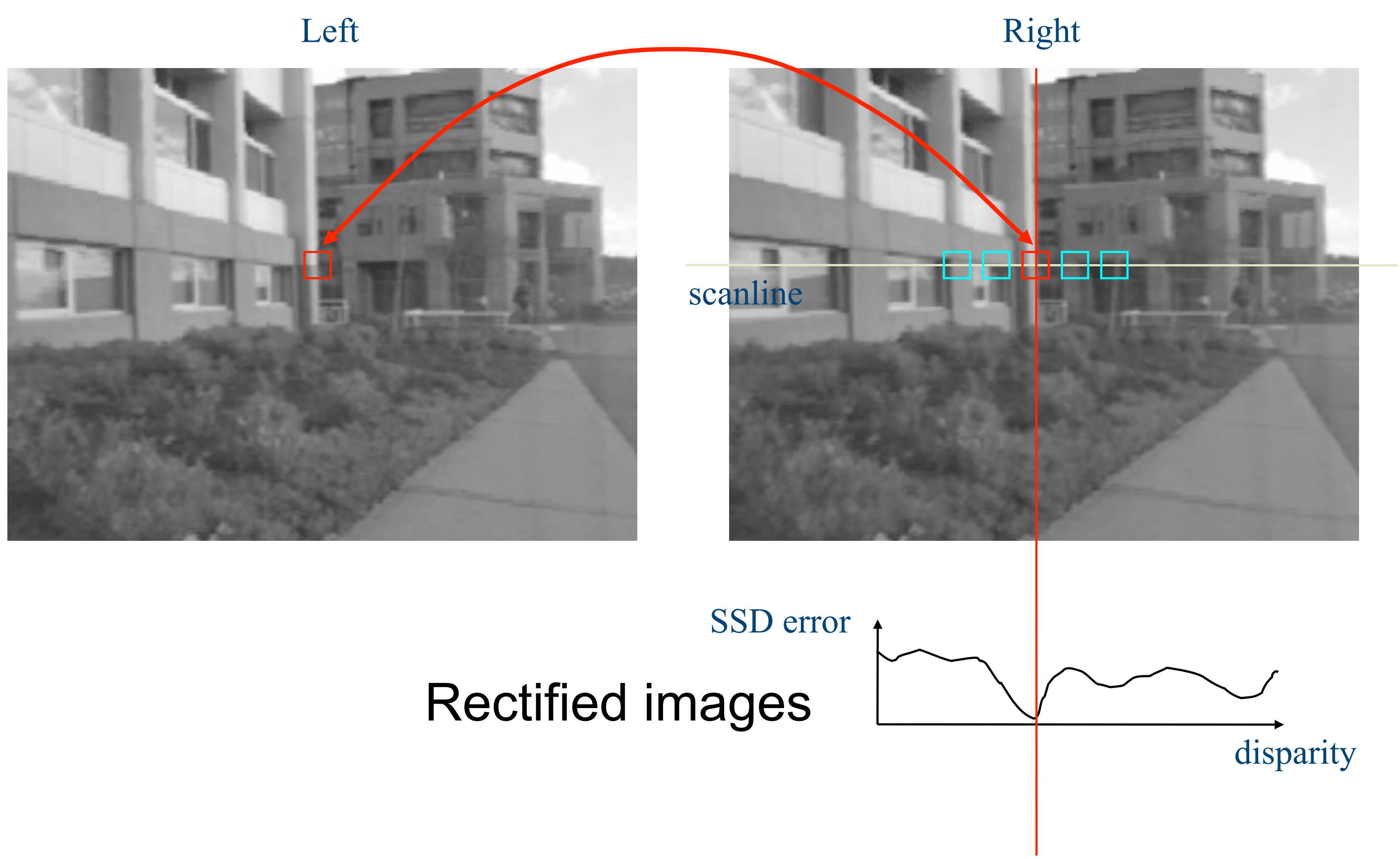
- Correlation



Rectified images

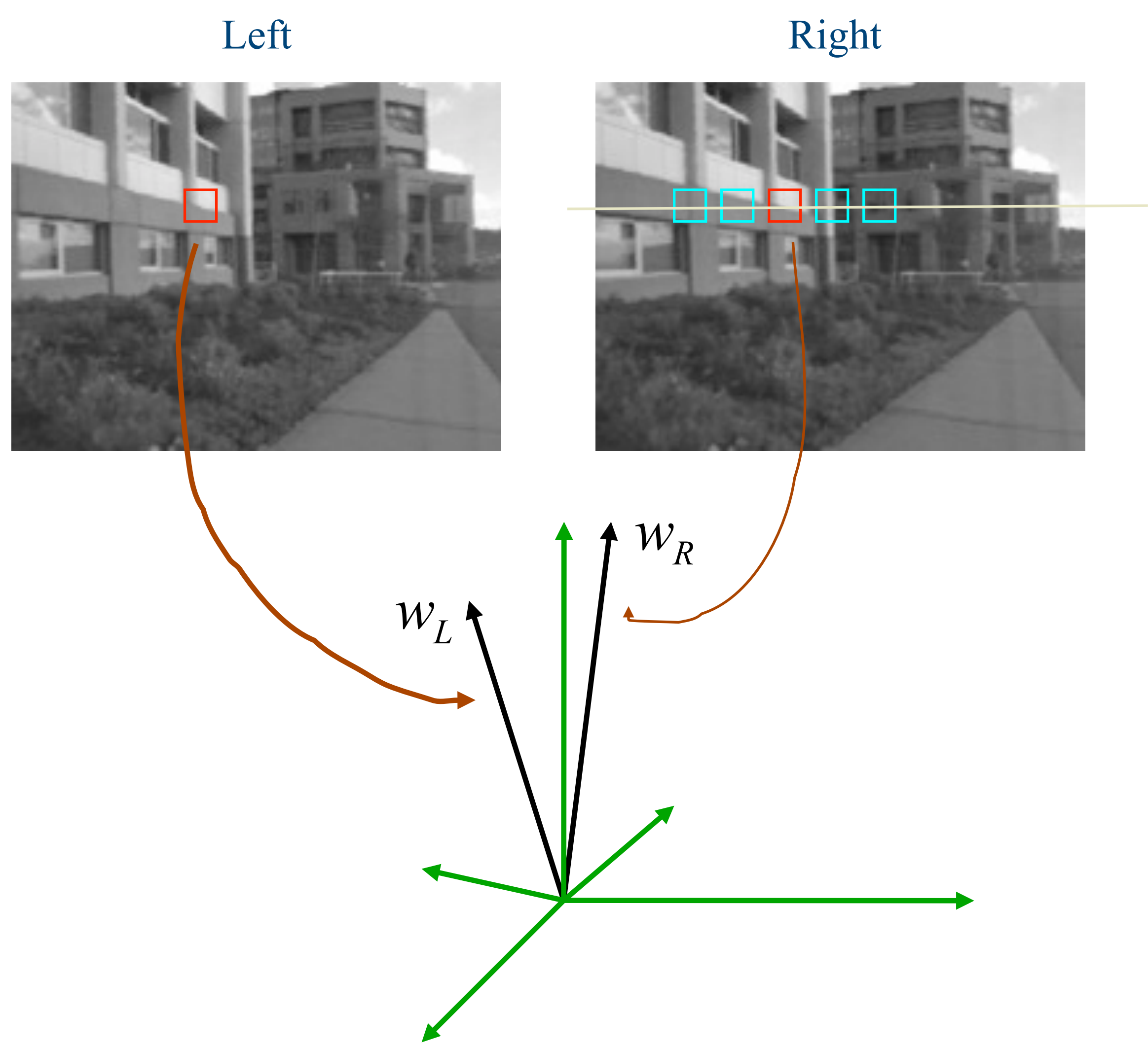
Point matching

- Correlation

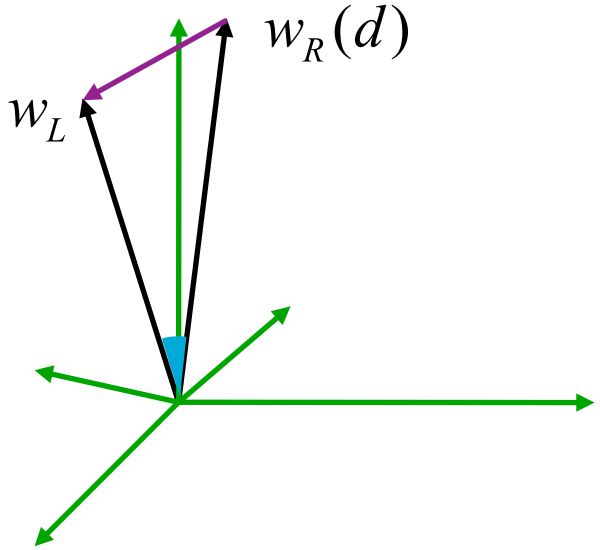


Point matching

- neighborhood signature comparison



Point matching



(Normalized) Sum of Squared Differences

$$C_{\mathrm{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \left\| w_L - w_R(d) \right\|^2$$

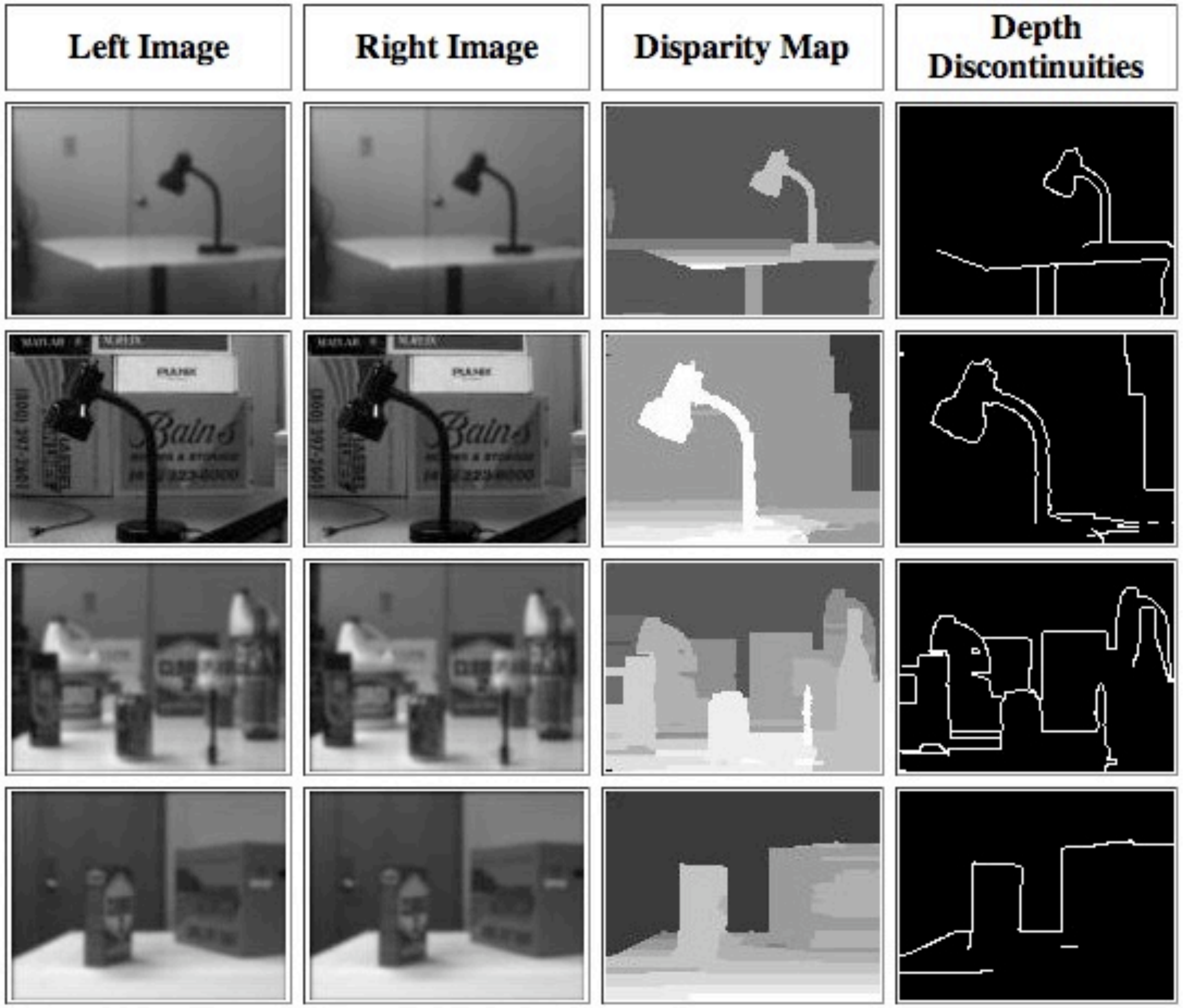
Normalized Correlation

$$C_{\mathrm{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \times w_R(d) = \cos\theta$$

$$d^* = \arg\min_d \left\| w_L - w_R(d) \right\|^2 = \arg\max_d w_L \times w_R(d)$$

Point matching



http://www.ces.clemson.edu/~stb/research/stereo_p2p/

Disparity map

Disparity map

W = 3

W = 20

General case

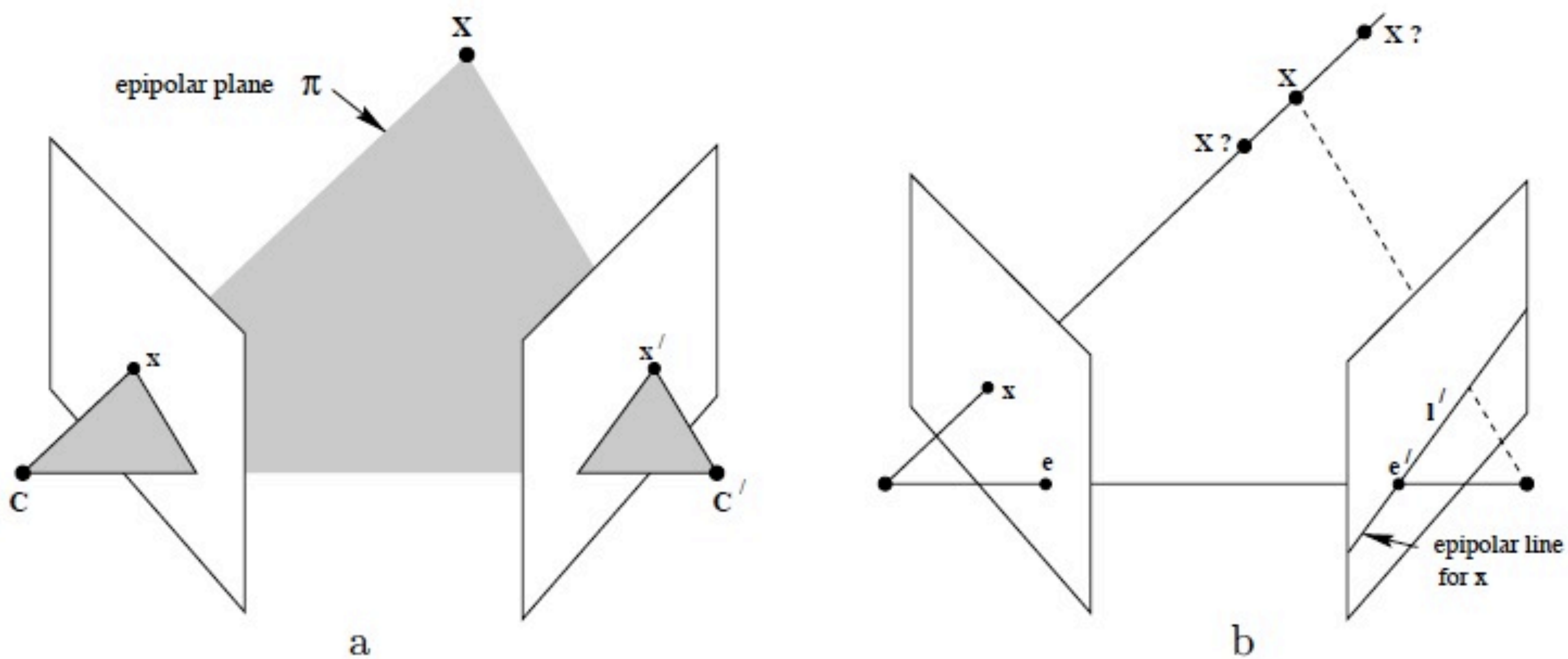
- Difficult :
  - optical axis not //
  - base line not the same
  - angle chosen to optimise scene coverage
  - calibration needed
    - 2 solids oriented by R + T)

Point matching using special points

- Harris, surf, fast ,...
- add some robustness to deformation, allows wider angle between point of view
- use of structured light
- ...

General case

- both camera are linked by a solid transform
- in general the line joining c to x is the epipolar line in c'
- epipolar plane : c,c' and x



http://www.robots.ox.ac.uk/~vgg/hzbook/hzbook1/HZepipolar.pdf

Ambiguity

- mismatch

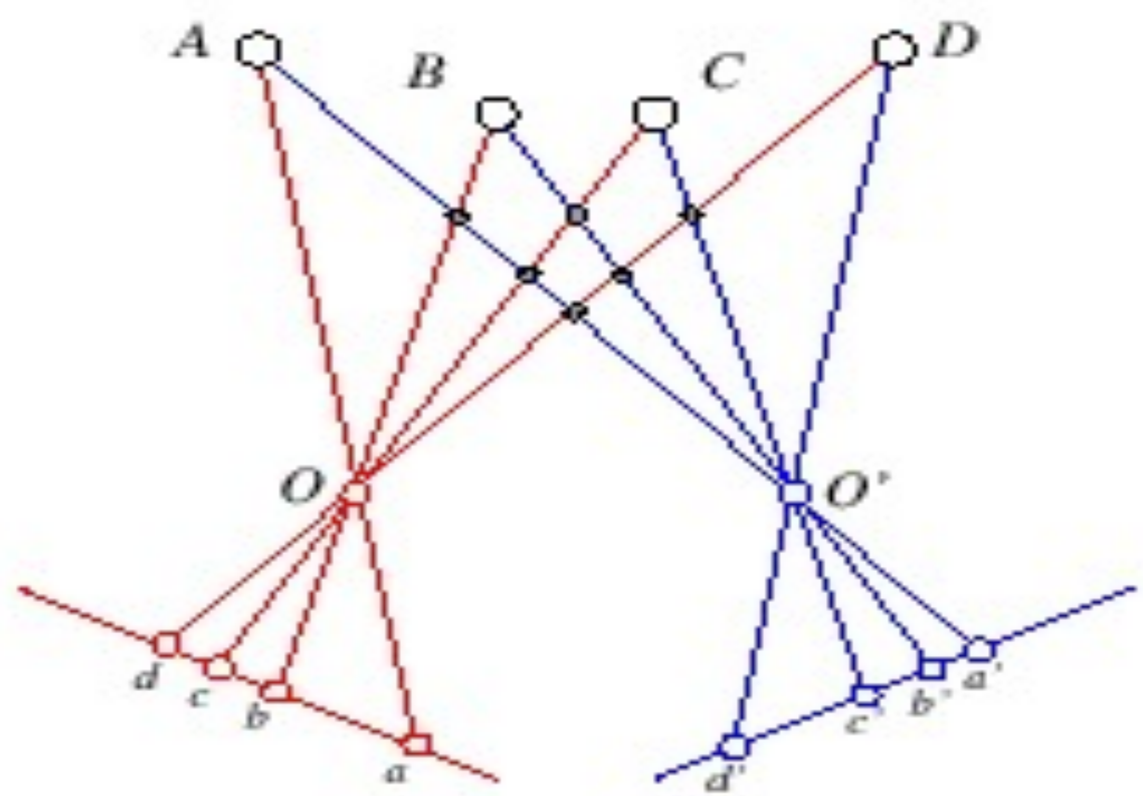
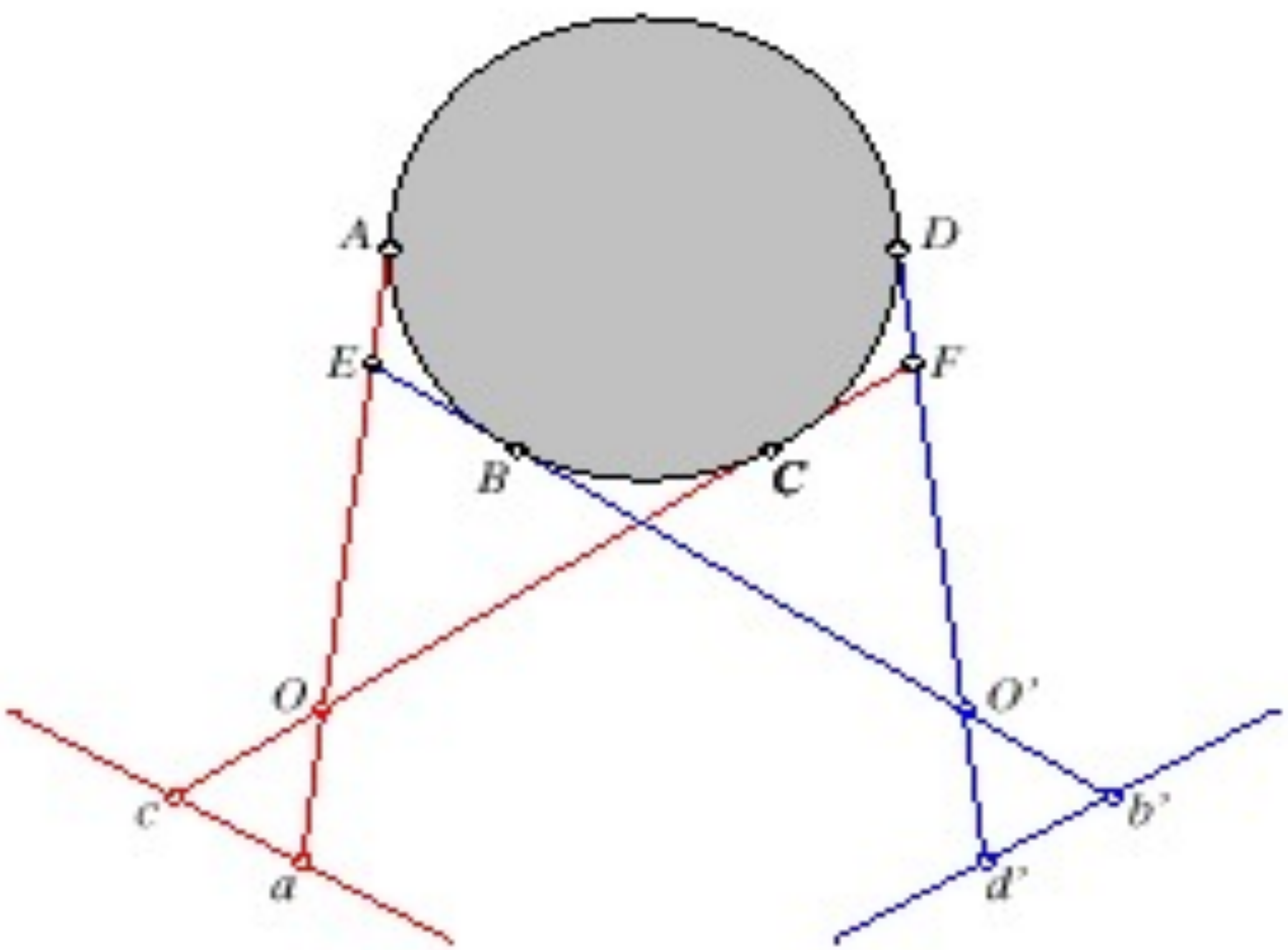


Figure from Forsyth & Ponce

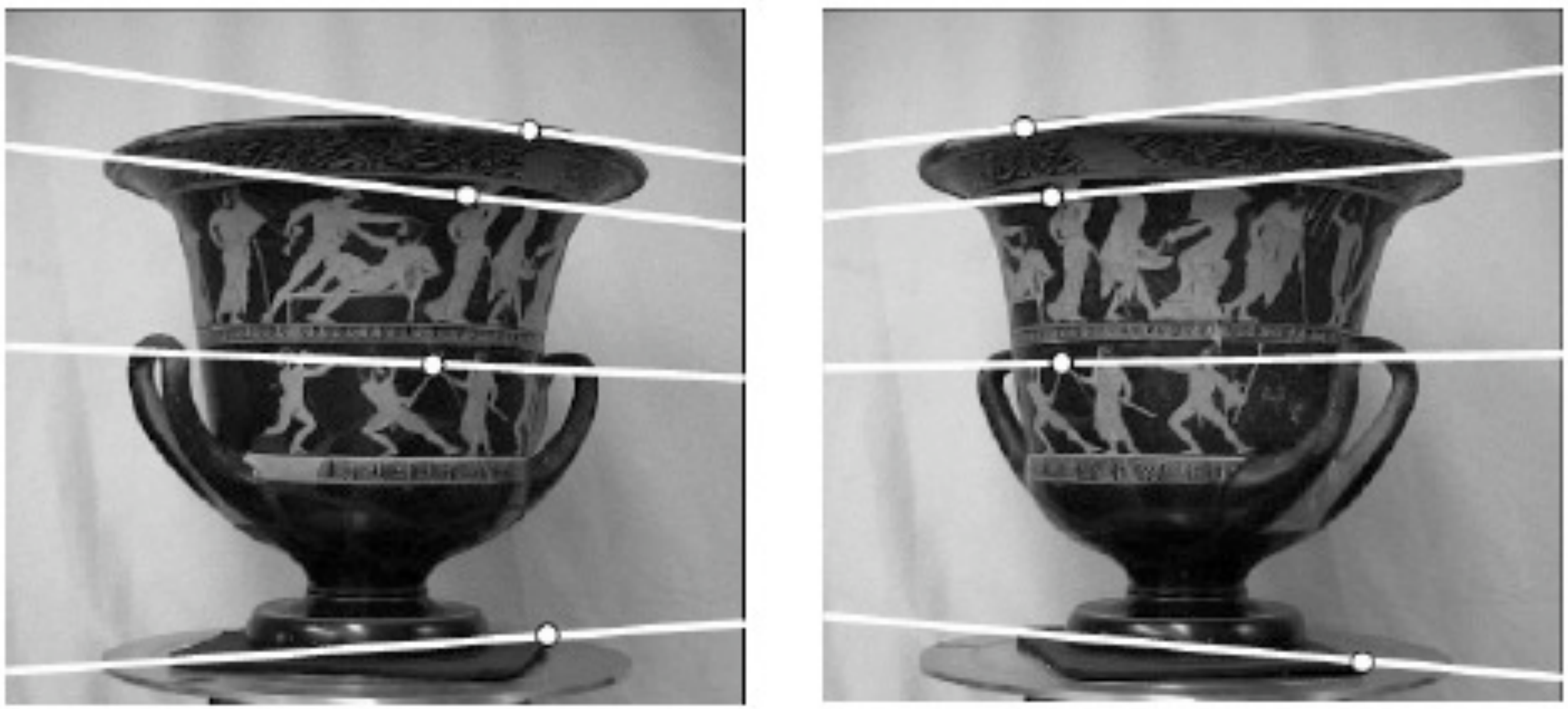
Ambiguity

- no fix characteristic points (smooth surface)



Epipolar lines

- convergent axis

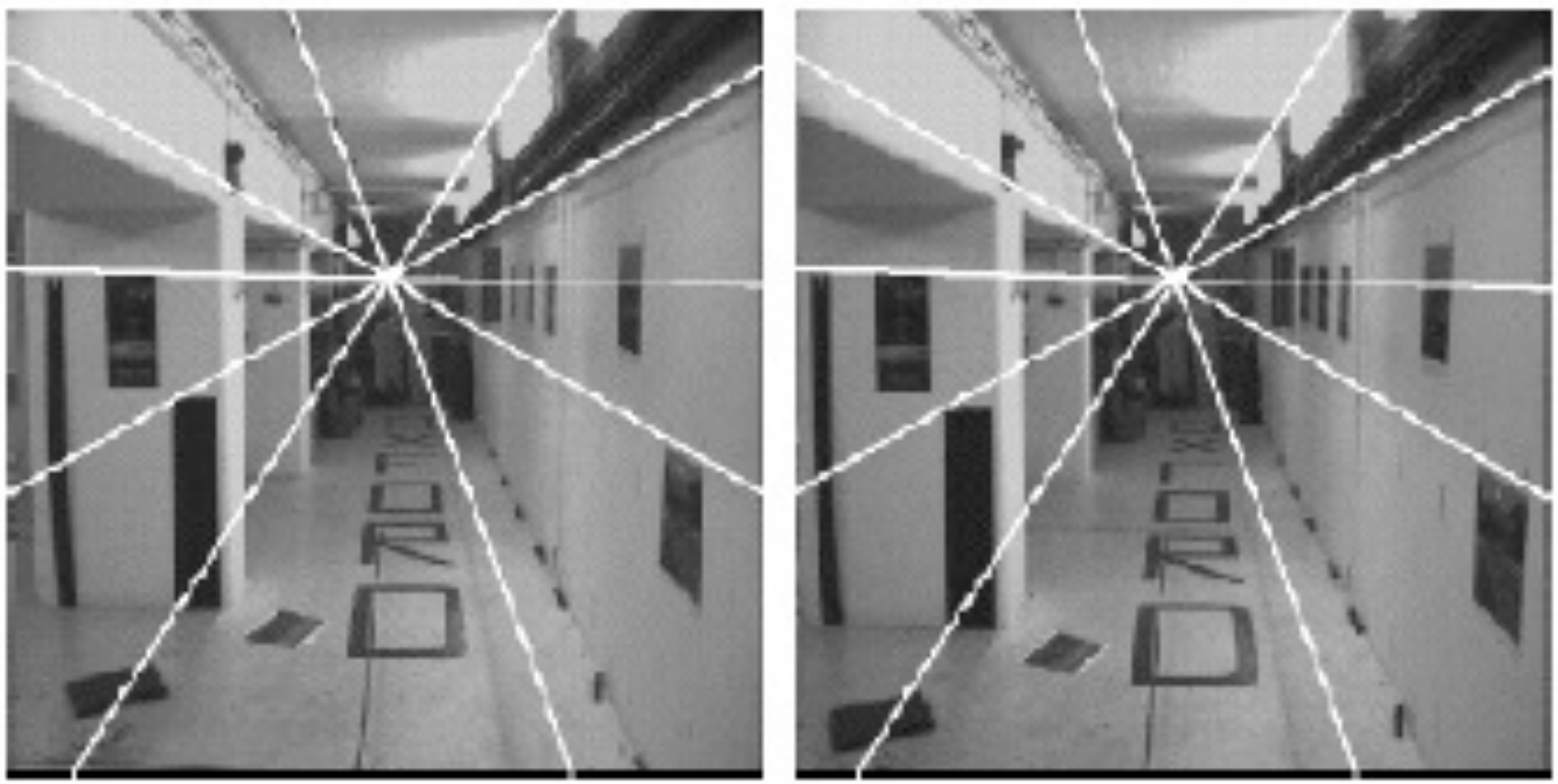


Epipolar lines

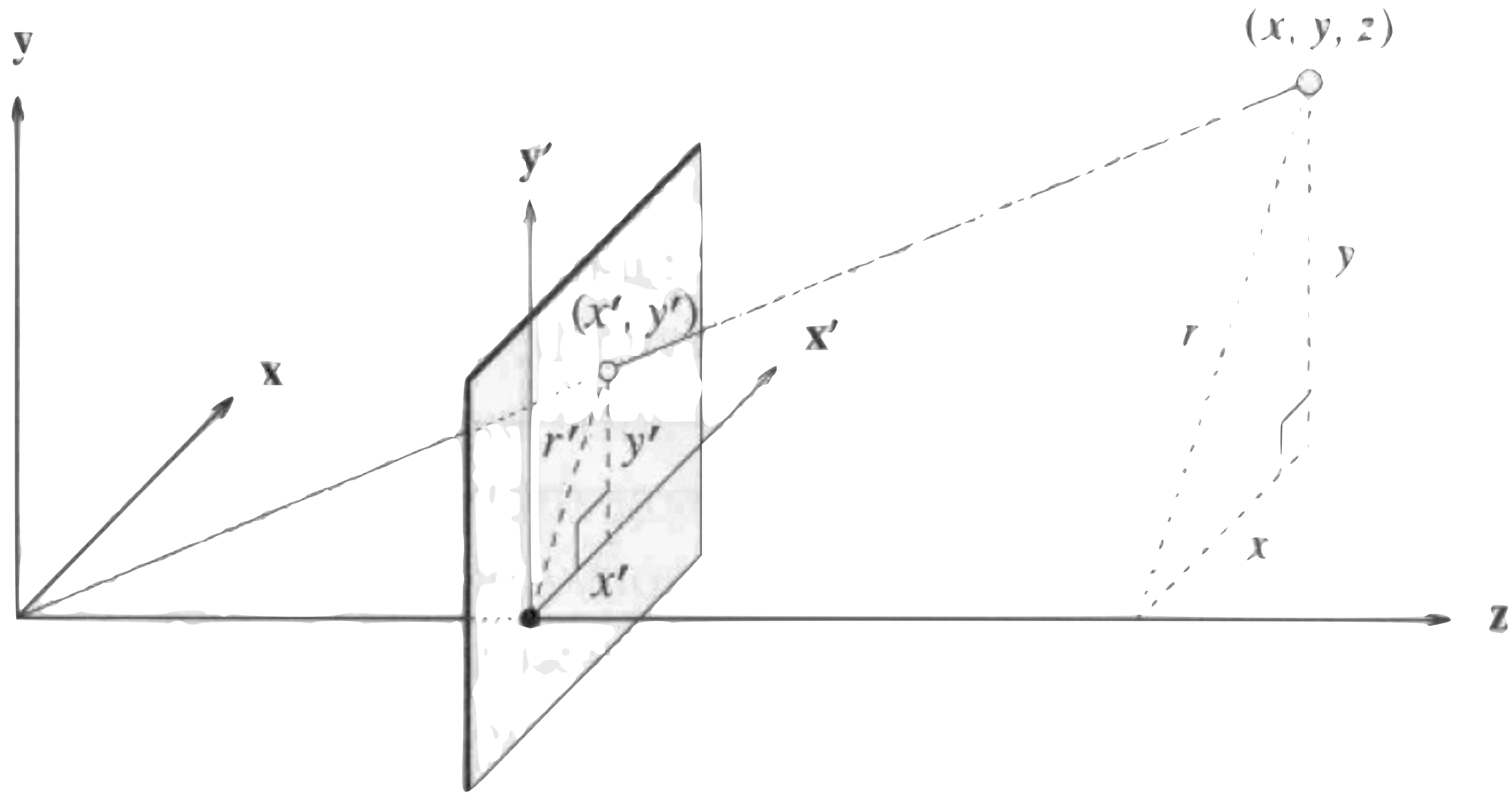
- Translation // (XY)


Epipolar lines

- translation Z



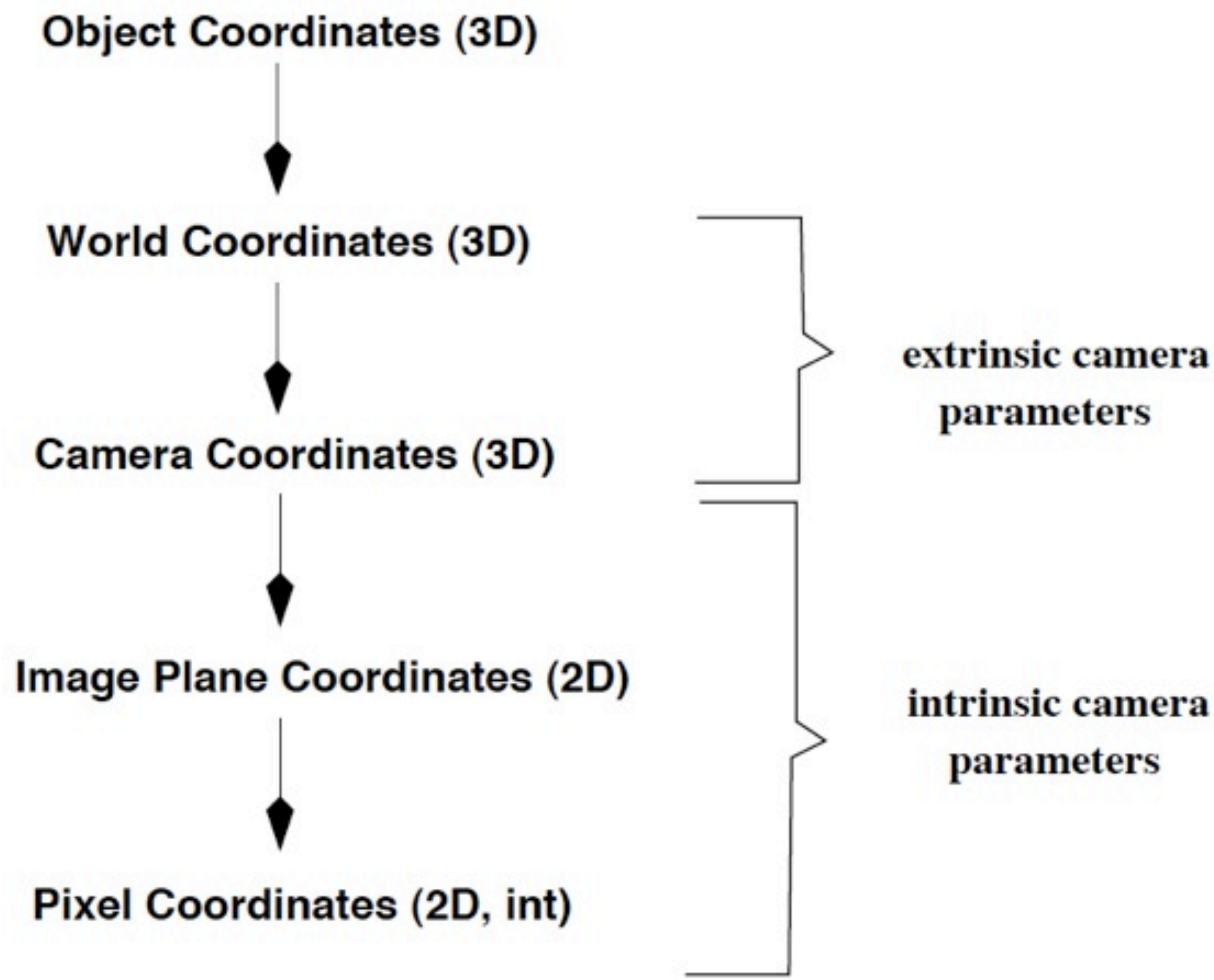
Camera parameters



http://www.cse.unr.edu/~bebis/CS791E/Notes/CameraParameters.pdf

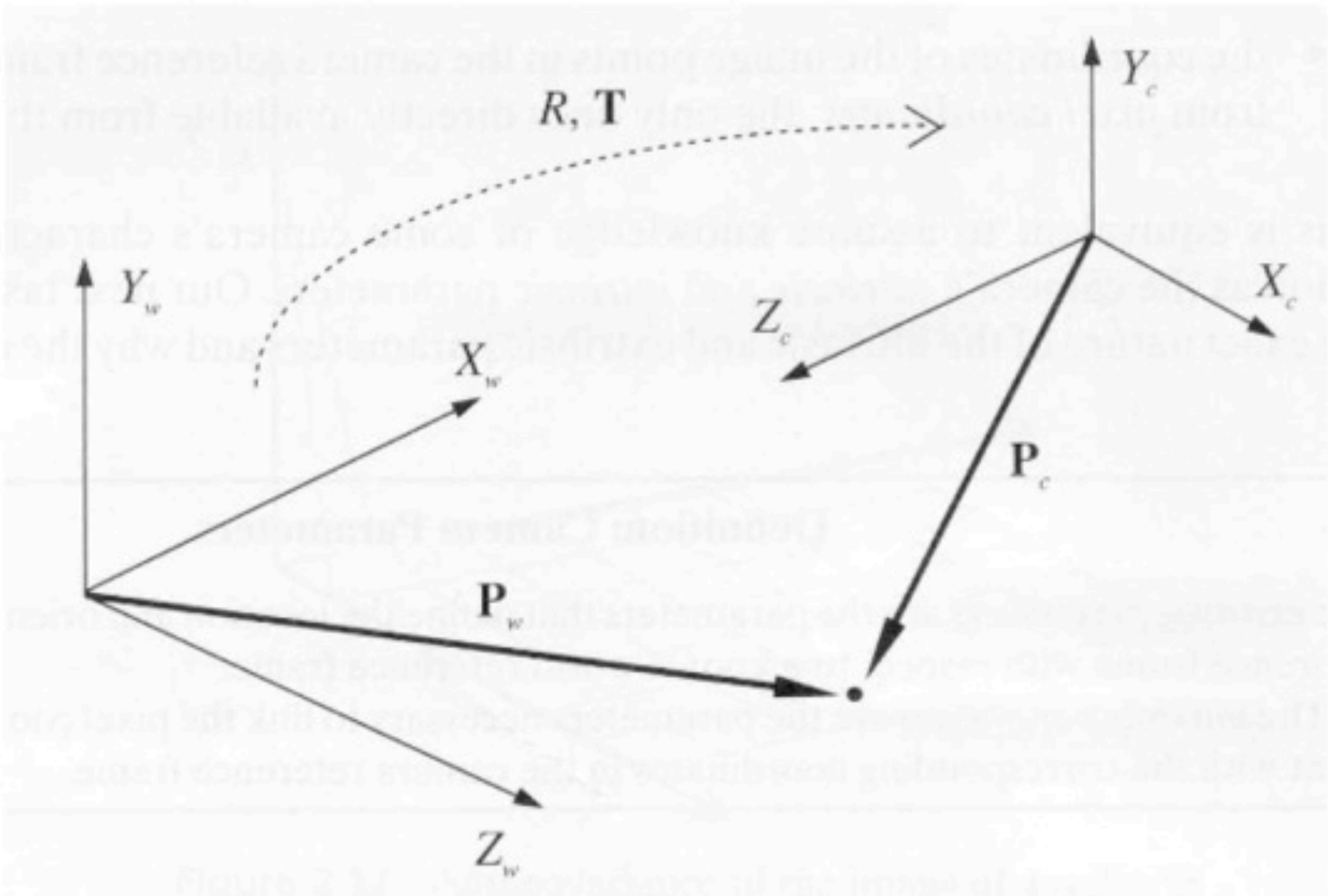
Camera parameters

- extrinsic parameters
  - position and orientation in space
- instrinsic parameters
  - link between pixel projected and space



Extrinsic parameters

- identify camera position w.r.t. global space
  - translation T
  - rotation R



Extrinsic parameters

- Pw coordinate in world space
- Pc coordinate in camera space

$$P_c = R(P_w - T)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w - T_x \\ Y_w - T_y \\ Z_w - T_z \end{bmatrix}$$

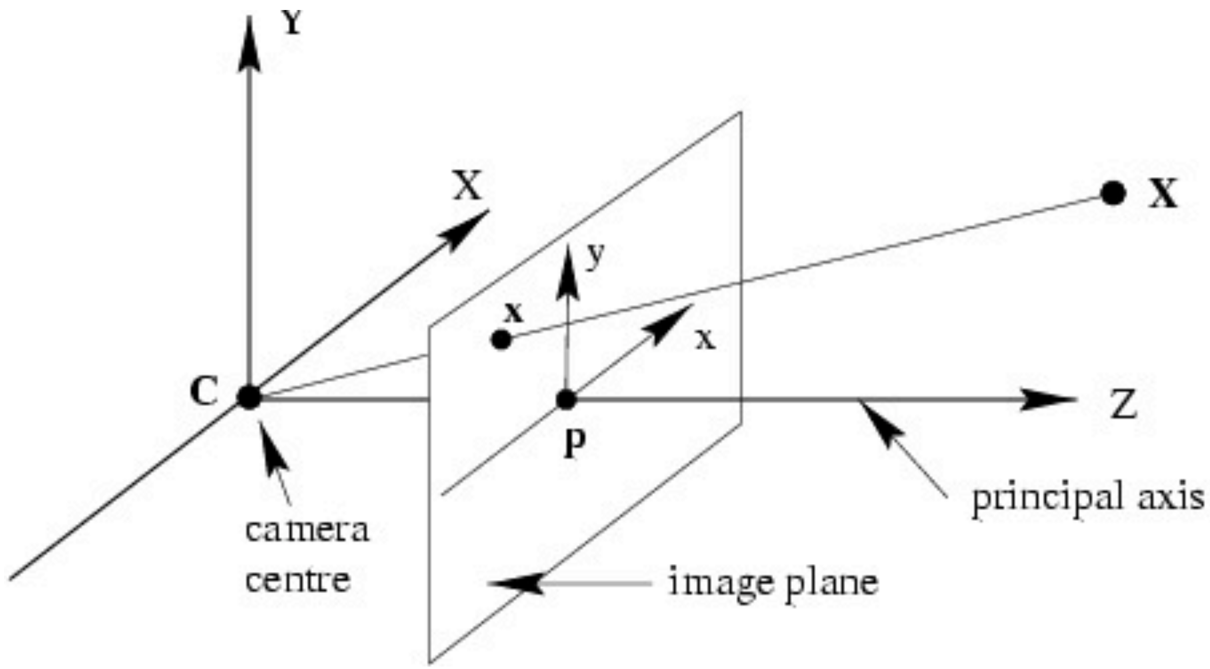
$$X_c = R_1^T(P_w - T)$$
$$Y_c = R_2^T(P_w - T)$$
$$Z_c = R_3^T(P_w - T)$$

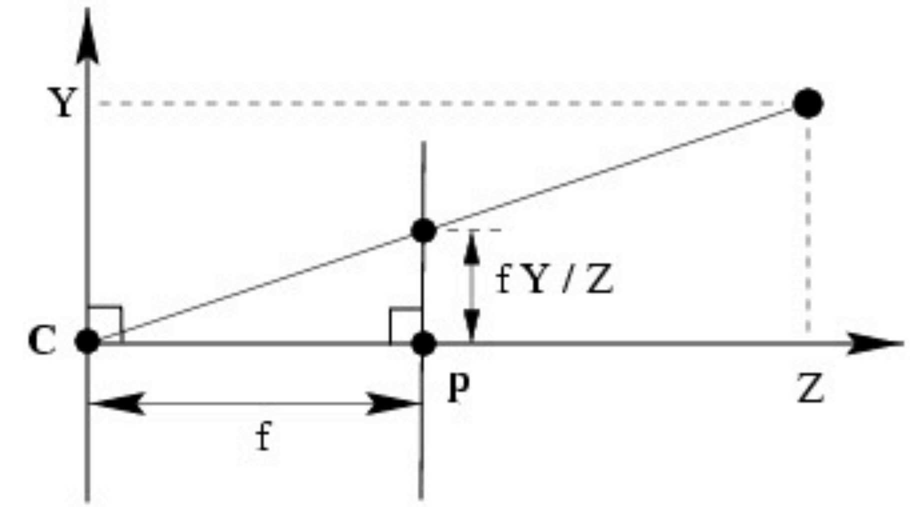
Intrinsic parameters

- Geometrical and optical camera characteristics
    - projection (focal distance f)
    - image plane <> pixel transform
    - optical distortions

Intrinsic parameters

- **projection (focal distance f)**



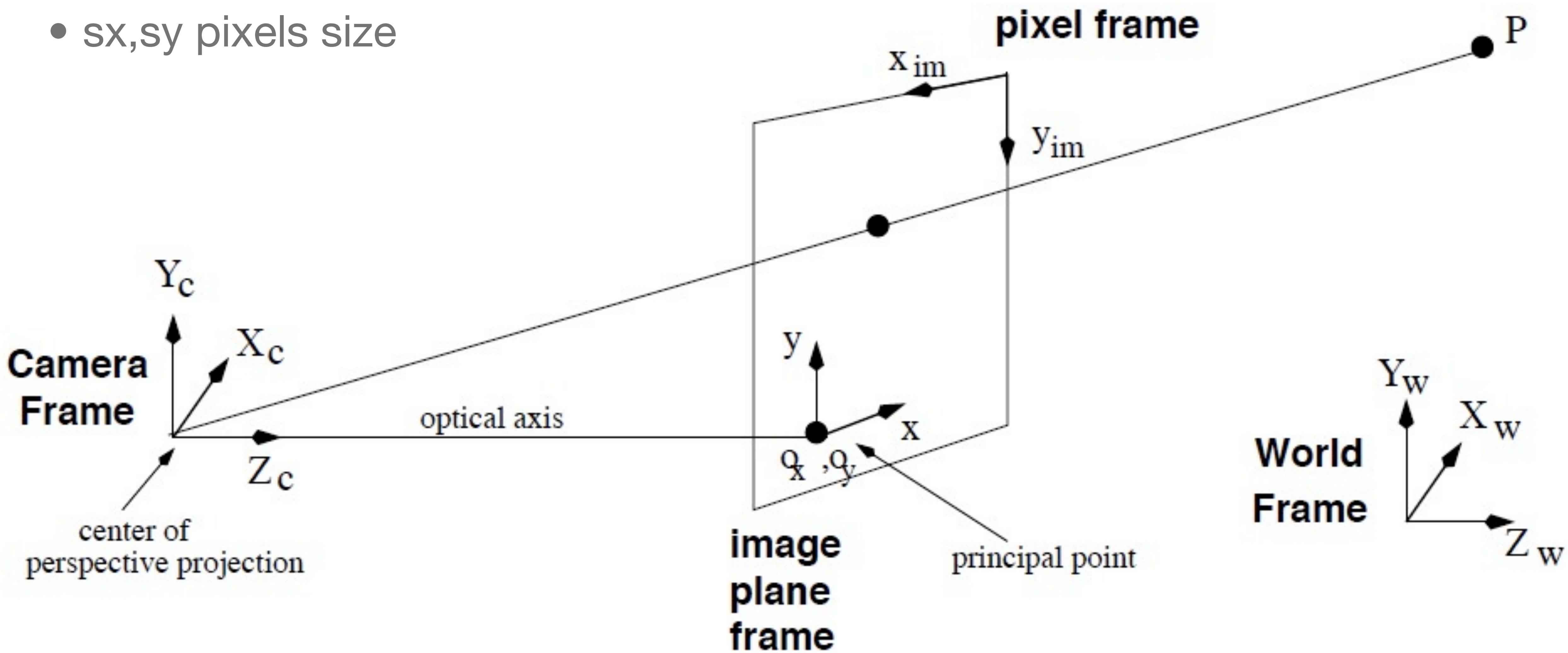


$$x = f\frac{X_c}{Z_c} = f\frac{R_1^T(P_w - T)}{R_3^T(P_w - T)}$$

$$y = f\frac{Y_c}{Z_c} = f\frac{R_2^T(P_w - T)}{R_3^T(P_w - T)}$$

Intrinsic parameters

- **image plane<> pixels transform**
- sx,sy pixels size



$$x = -(x_{im} - o_x)s_x$$

$$y = -(y_{im} - o_y)s_y$$

Intrinsic parameters

- **image plane<> pixels transform**
- sx,sy pixels size

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- image coordinates

$$x_{im} = -f s_x \frac{R_1^T(P_w^T)}{R_3^T(P_W - T)} + o_x$$

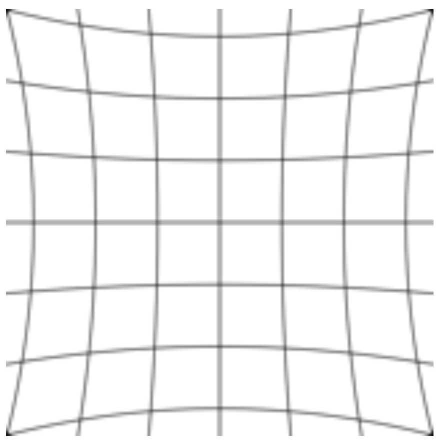
$$y_{im} = -f s_y \frac{R_2^T(P_w^T)}{R_3^T(P_W - T)} + o_y$$

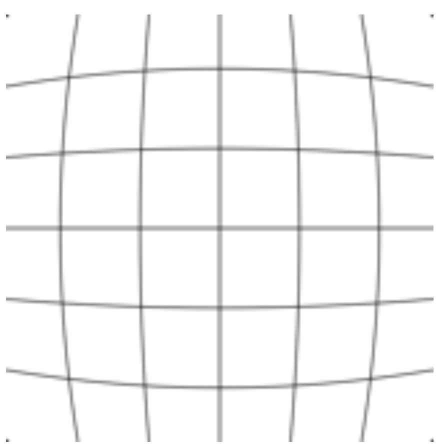
Intrinsic parameters

- **optical distortions**

$$x_u = x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + ...) + (P_1(r^2 + 2(x_d - x_c)^2) + 2P_2(x_d - x_c)(y_d - y_c))(1 + P_3 r^2 + ...)$$

$$y_u = y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + ...) + (P_2(r^2 + 2(y_d - y_c)^2) + 2P_1(x_d - x_c)(y_d - y_c))(1 + P_3 r^2 + ...)$$





(x_u, y_u) = undistorted image point,
(x_d, y_d) = distorted image point,
(x_c, y_c) = centre of distortion (ie. the principal point),
K_n = n^{th} radial distortion coefficient,
P_n = n^{th} tangential distortion coefficient,
$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$, and
... = an infinite series.

Camera parameters

- intrinsic parameters

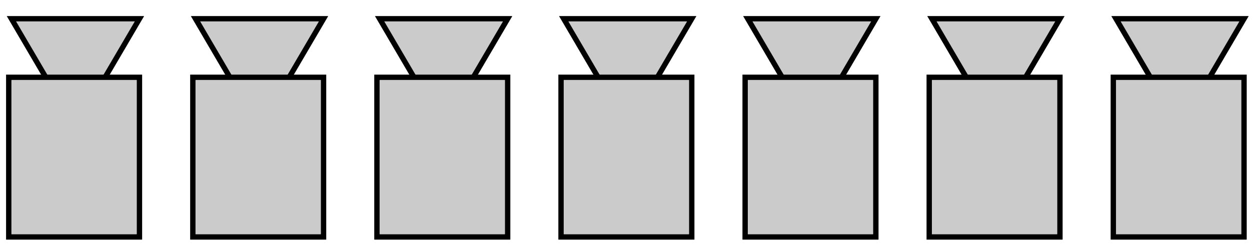
$$M_{in} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

- extrinsic parameters

$$M_{ex} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix}$$

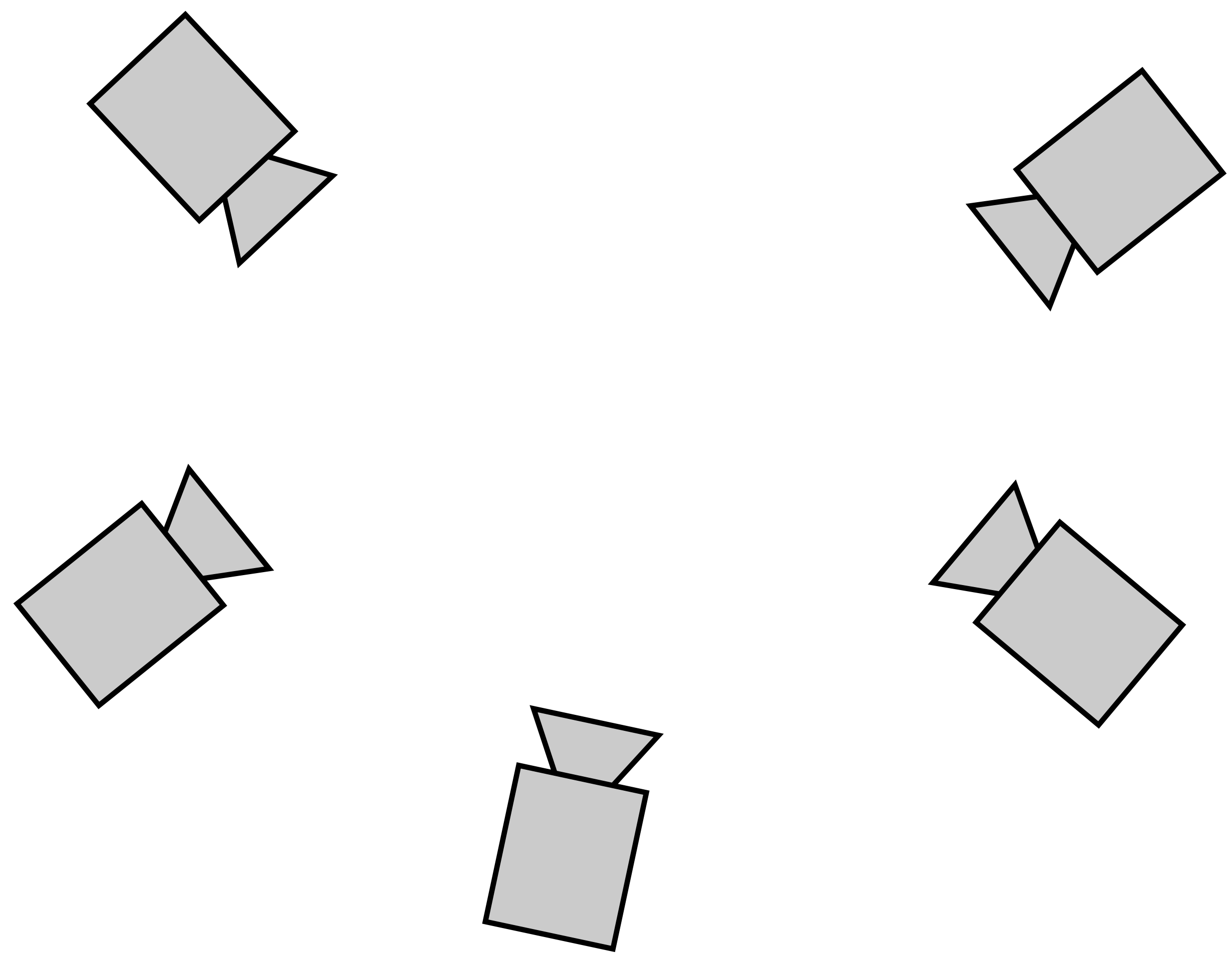
Multi-camera

- camera en translation
  - ligne de base commune
  - distance = cste



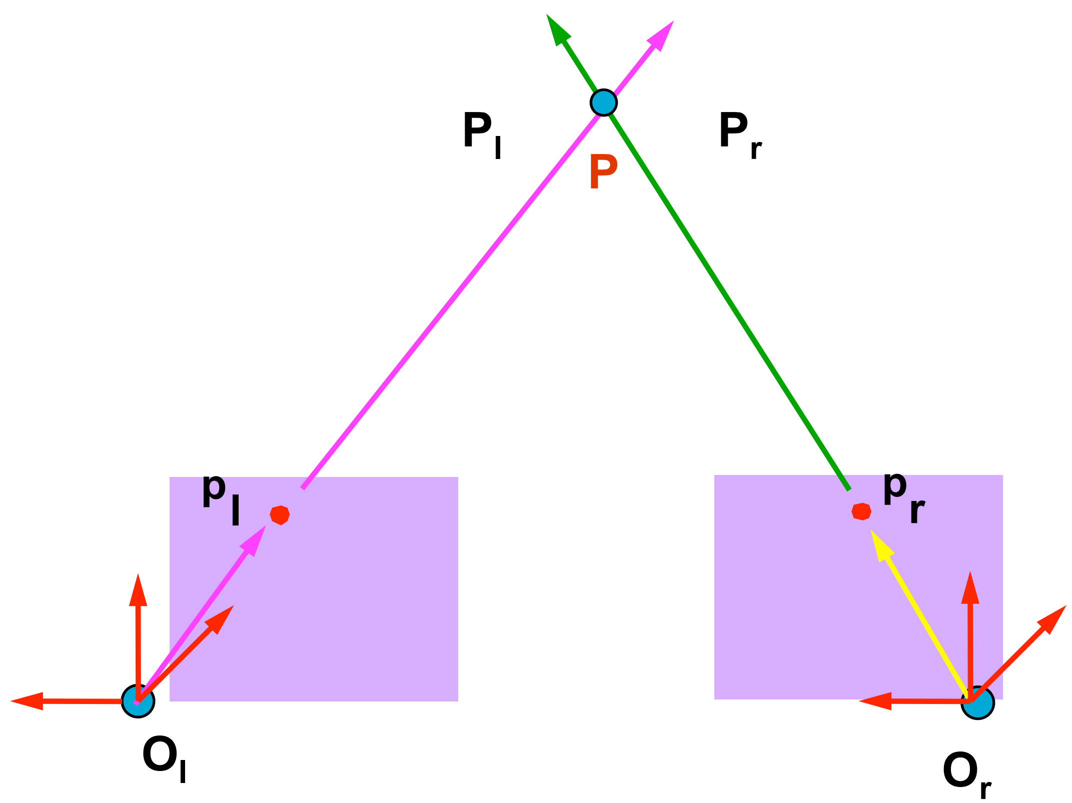
Multi-camera

- different base lines
  - géométries épipolaires différentes
  - permet l’observation des dépouilles



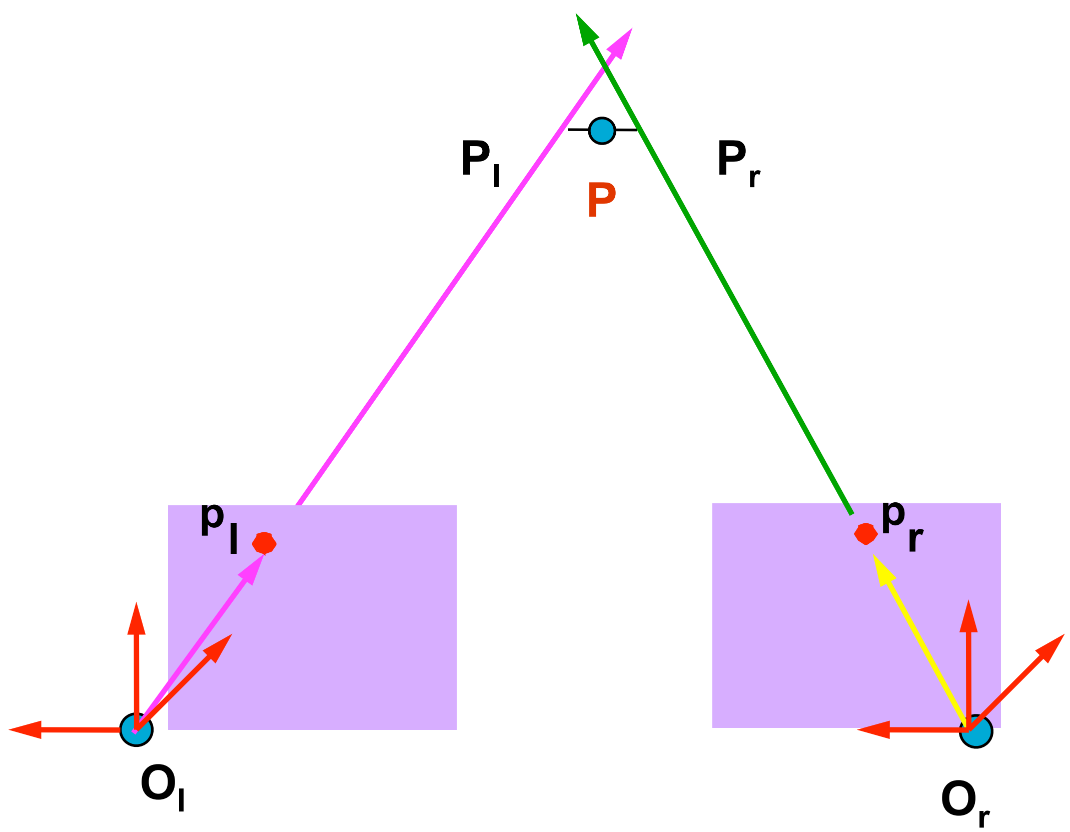
Reconstruction

- ideal case



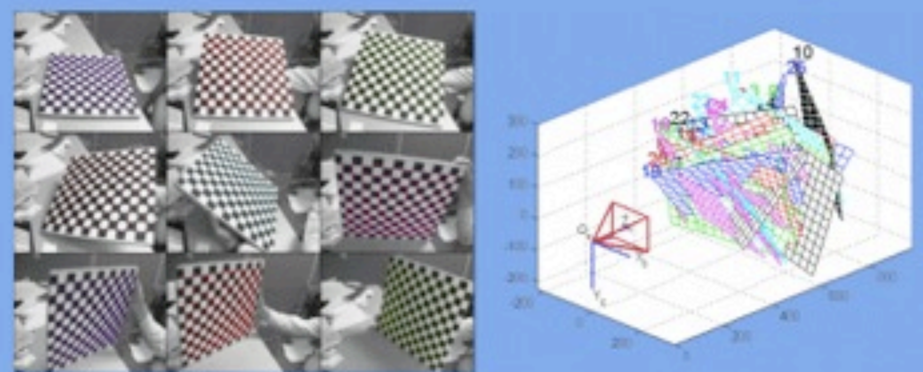
Reconstruction

- real case two lines are not secant



Calibration software example

Camera Calibration Toolbox for Matlab



New Fisheye Calibration Toolbox included! (calib_gui_fisheye.m)
Not yet documented

This is a release of a Camera Calibration Toolbox for Matlab® with a complete documentation. This document may also be used as a tutorial on camera calibration since it includes general information about calibration, references and related links.

The C implementation of this toolbox is included in the Open Source Computer Vision library distributed by Intel and freely available online.

Content:

- System requirements
- Getting started
- Calibration examples
- Description of the calibration parameters
- Description of the functions in the calibration toolbox
- Doing your own calibration
- Undocumented features of the toolbox
- References
- A few links related to camera calibration

http://www.vision.caltech.edu/bouguetj/calib_doc/

Calibration software example