Multiple View Geometry - Reconstruction from two views

Class Objectives

- Understand the fundamentals of two view geometry
- Learn out how to compute 3D points from two views
- Understand the properties and constraints behind Epipolar geometry
- Know how to compute the Essential and the Fundamental matrices

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Some slides adapted from Quim Salvi, Noah Snavely, Andrew Zisserman, Alexei Efros

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Shape from *X*

Techniques based on:

– Modifying the intrinsic camera parameters

i.e. Depth from Focus/Defocus and Depth from Zooming

– Considering an add al source

i.e. Shape from Structur an

– Considering additional surface in

i.e. Shape from Shading, Shape fro
Geometric Constraints

– Multiple views

i.e. Shape from Stereo and Shape

Shape from Focus/Defocus

Shape from *X*

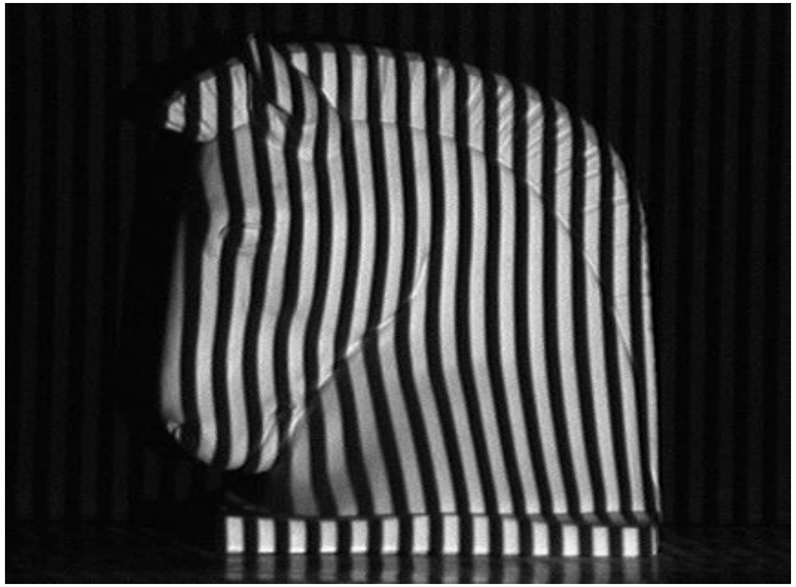
Techniques based on:

- Modifying the intrinsic camera parameters
  i.e. Depth from Focus/Defocus and Depth from Zooming

- Considering an additional source of light onto the scene
  i.e. Shape from Structured Light and Shape from Photometric Stereo

- Considering additio surface i
  i.e. Shape from Shadi f
  Geometric Constraints

- Multiple views
  i.e. Shape from Stereo and Shape



Shape from Structured Light

Shape from *X*

Techniques based on:

- Modifying the intrinsic camera pa
  i.e. Depth from Focus/Defocus and
- Considering an additional source
  i.e. Shape from Structured Light an
- Considering additional surface information
  i.e. Shape from Shading, Shape from Texture and Shape from Geometric Constraints
- Multiple views
  i.e. Shape from Stereo and Shape from Motion

Shape from Shading

Shape from *X*

Techniques based on:

- Modifying the intrinsic camera p

  i.e. Depth from Focus/Defocus ar

- Considering an additional sourc

  i.e. Shape from Structured Light a

- Considering additional surface

  i.e. Shape from Shading, Shape f
  Geometric Constrai

- Multiple views

  i.e. Shape from Stereo and Shape from Motion

Shape from Stereo

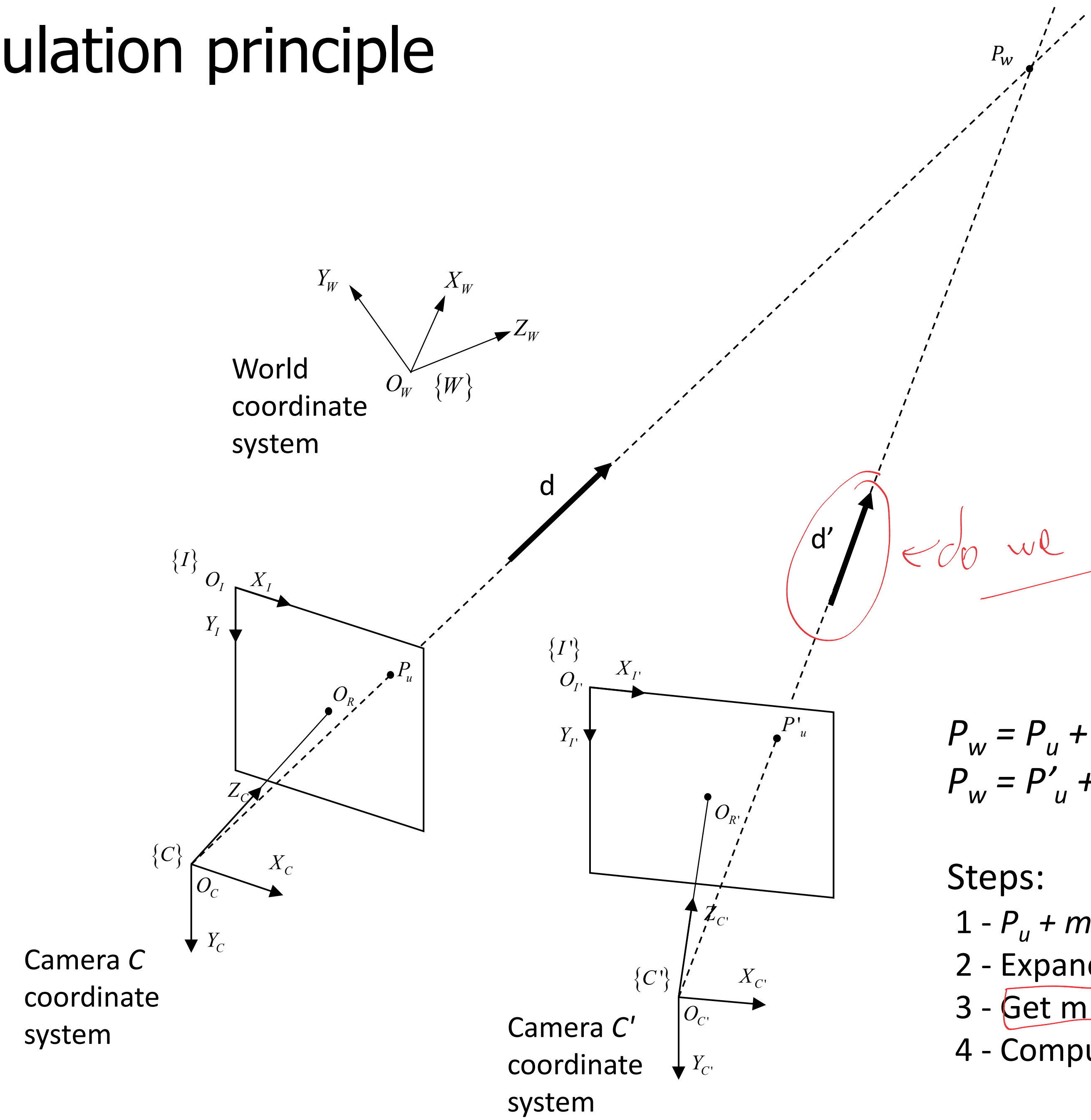
Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Triangulation principle



← do we know this?

$P_w = P_u + m \cdot d$ → magnitude multiplied by d-vector, right?

$P_w = P'_u + m' \cdot d'$

Steps:

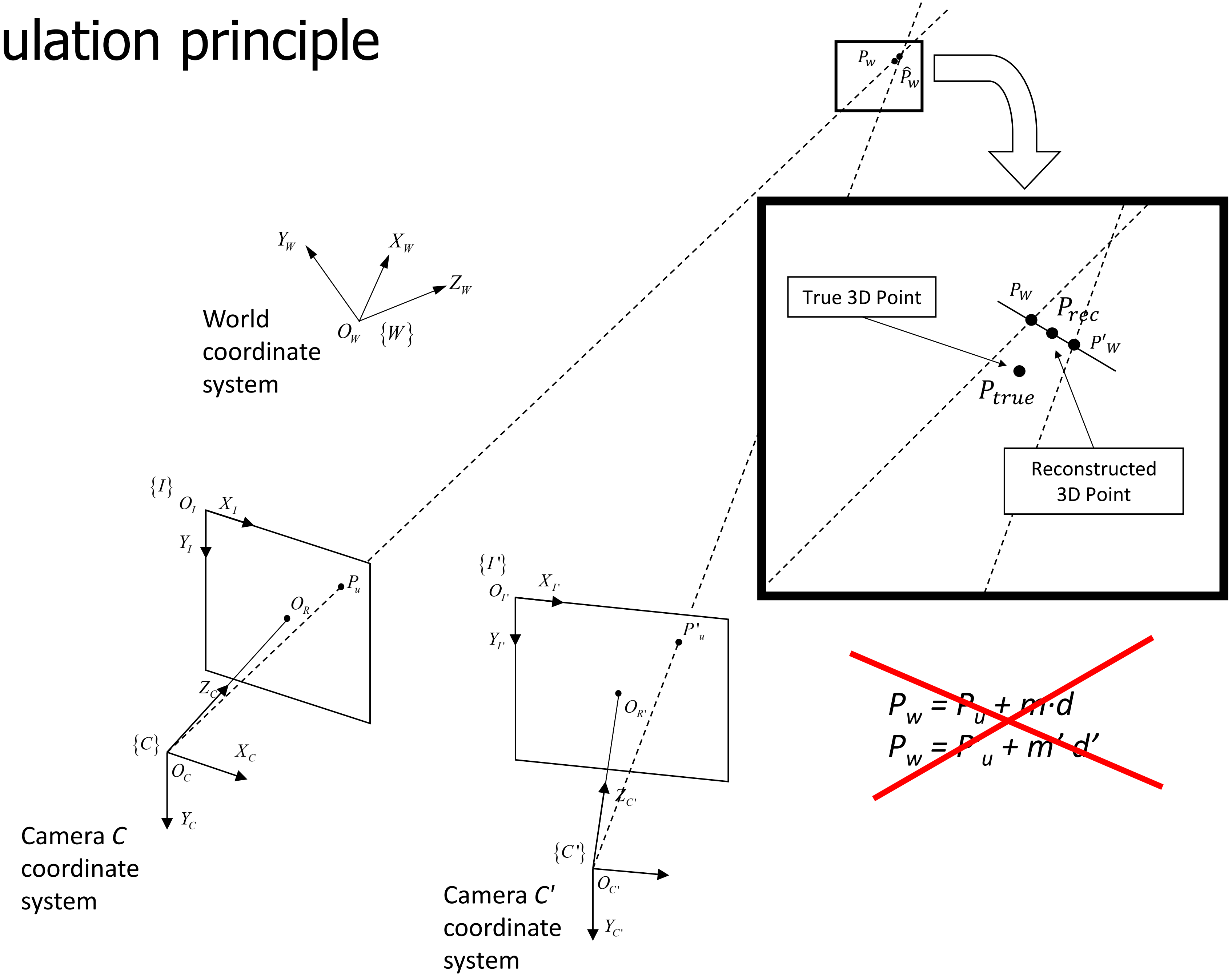
1 - $P_u + m{\cdot}d = P'_u + m'{\cdot}d'$

2 - Expand to x,y,z

3 - $\boxed{\text{Get m and m'}}$ ← How? two variables, one equation.

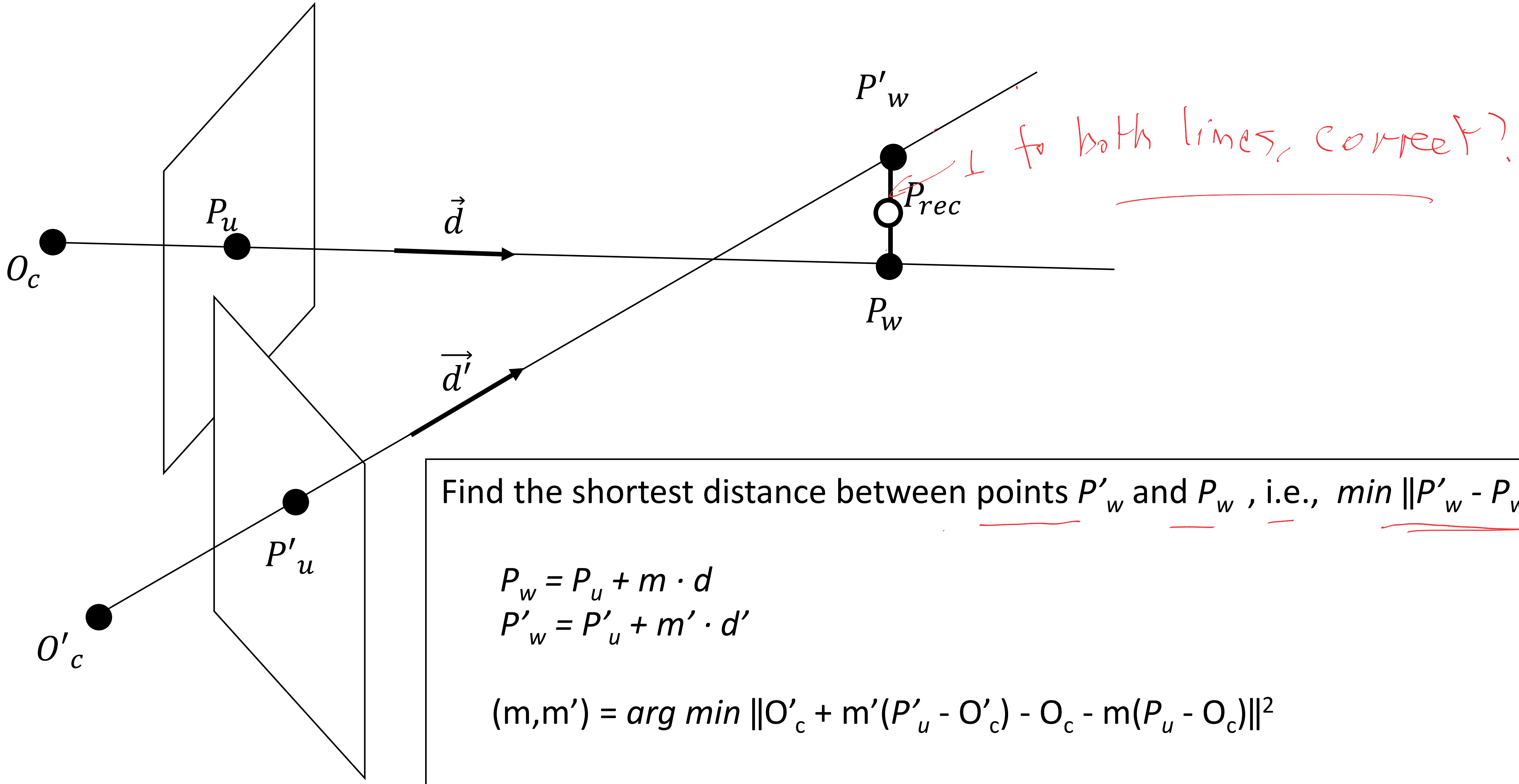
4 - Compute P_w

Triangulation principle



$$\cancel{P_w = P_u + m \cdot d}$$
$$\cancel{P_w = P'_u + m' \cdot d'}$$

Triangulation principle



Find the shortest distance between points P'_w and P_w , i.e., $min\ \|P'_w - P_w\|^2$

$$P_w = P_u + m \cdot d$$
$$P'_w = P'_u + m' \cdot d'$$

$$(m,m') = \arg\min \|O'_c + m'(P'_u - O'_c) - O_c - m(P_u - O_c)\|^2$$

$$P_{rec} = (P_w + P'_w) / 2$$

Triangulation principle

In practice we can use Least-Squares:

$$\begin{bmatrix}\lambda_1 u_1\\ \lambda_1 v_1\\ \lambda_1\end{bmatrix} = \begin{bmatrix}A_{11} & A_{12} & A_{13} & A_{14}\\ A_{21} & A_{22} & A_{23} & A_{24}\\ A_{31} & A_{32} & A_{33} & A_{34}\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix} \qquad \begin{bmatrix}\lambda_2 u_2\\ \lambda_2 v_2\\ \lambda_2\end{bmatrix} = \begin{bmatrix}B_{11} & B_{12} & B_{13} & B_{14}\\ B_{21} & B_{22} & B_{23} & B_{24}\\ B_{31} & B_{32} & B_{33} & B_{34}\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

$$\begin{bmatrix}A_{31}u_1 - A_{11} & A_{32}u_1 - A_{12} & A_{33}u_1 - A_{13}\\ A_{31}v_1 - A_{21} & A_{32}v_1 - A_{22} & A_{33}v_1 - A_{23}\\ B_{31}u_2 - B_{11} & B_{32}u_2 - B_{12} & B_{33}u_2 - B_{13}\\ B_{31}v_2 - B_{21} & B_{32}v_2 - B_{22} & B_{33}v_2 - B_{23}\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\end{bmatrix} = \begin{bmatrix}A_{14} - A_{34}u_1\\ A_{24} - A_{34}v_1\\ B_{14} - B_{34}u_2\\ B_{24} - B_{34}v_2\end{bmatrix}$$

$$Q \cdot X = C$$

$$X = Q^{+} \cdot C \qquad (X = Q\backslash \mathrm{C} \text{ in matlab})$$

more constraints

Q: What can we do if we have additional views of the same point ?

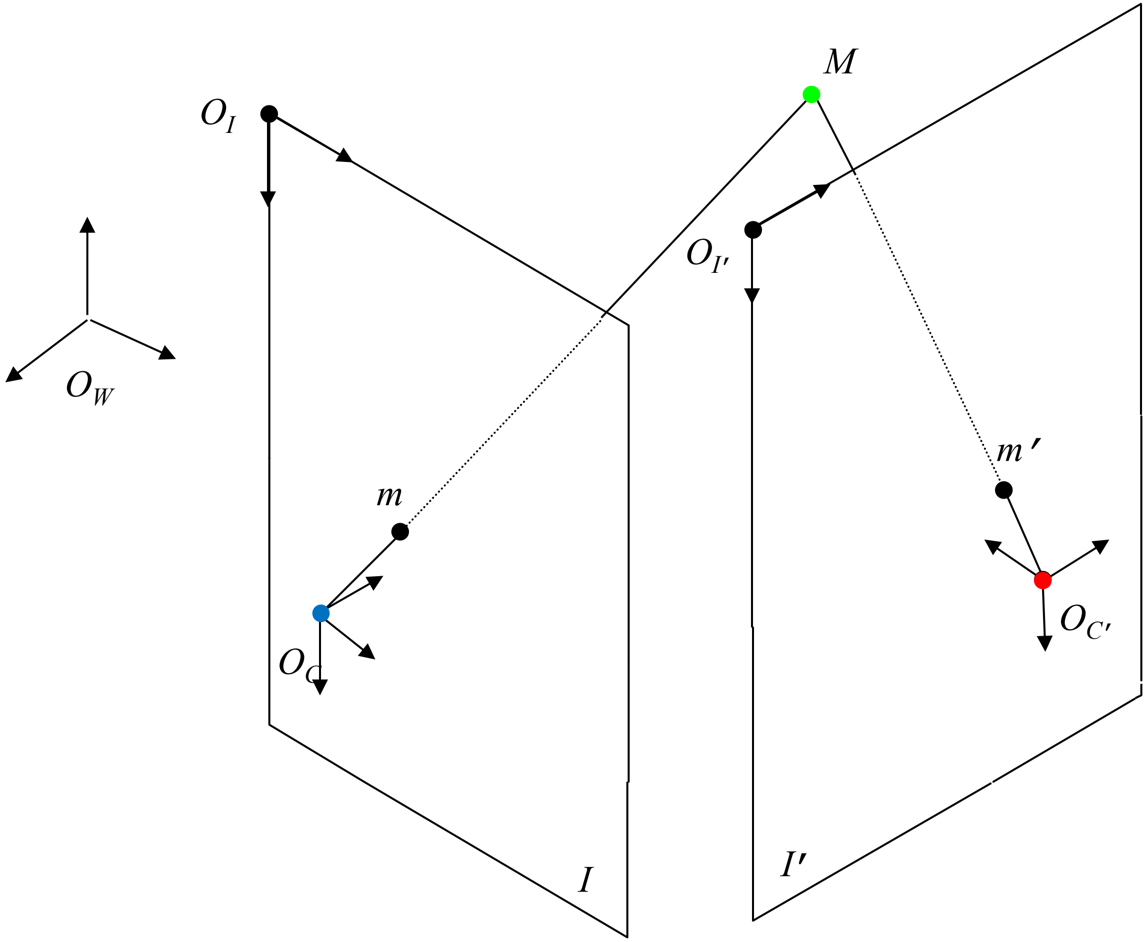
Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

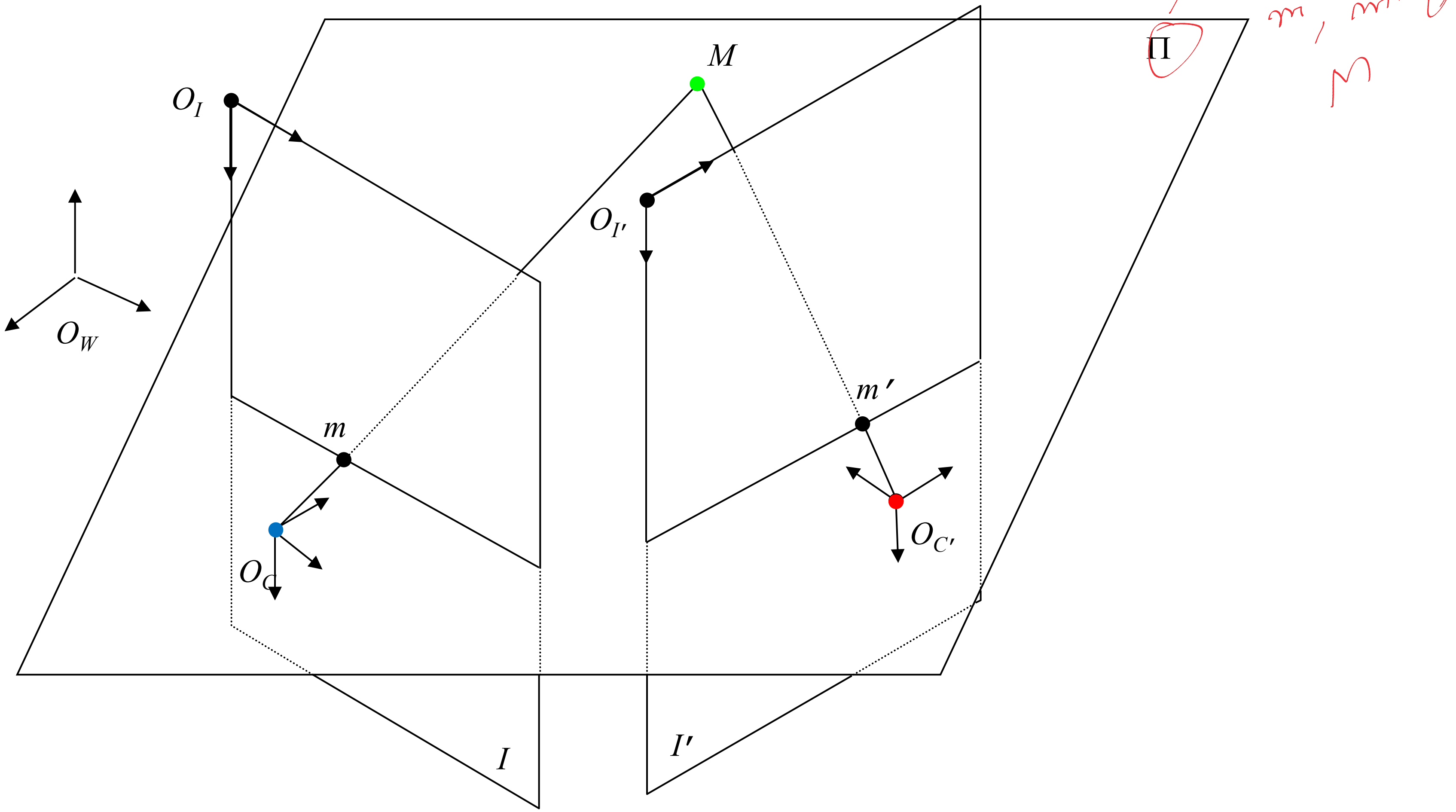
Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

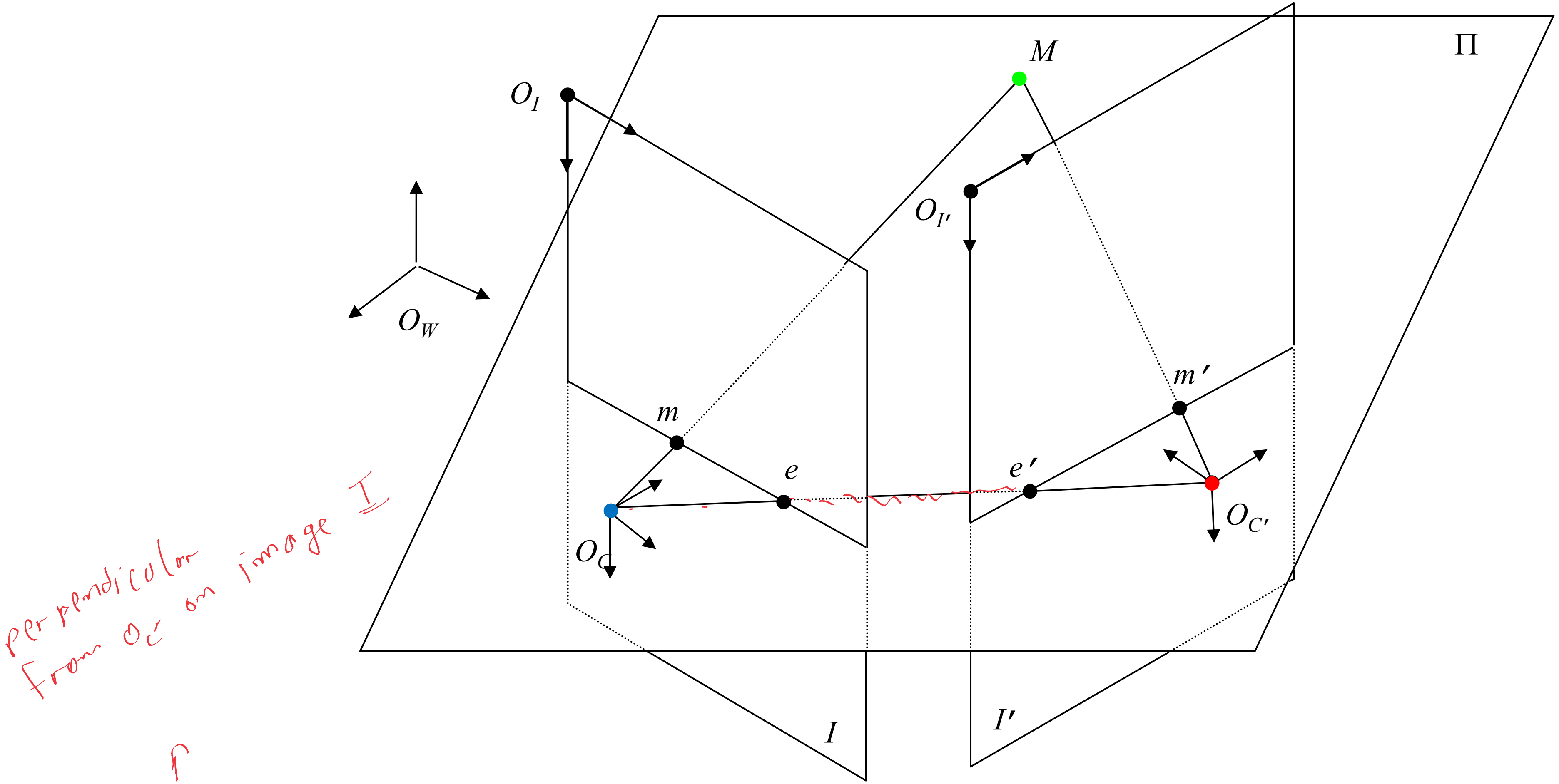
Epipoles and epipolar lines



Epipoles and epipolar lines

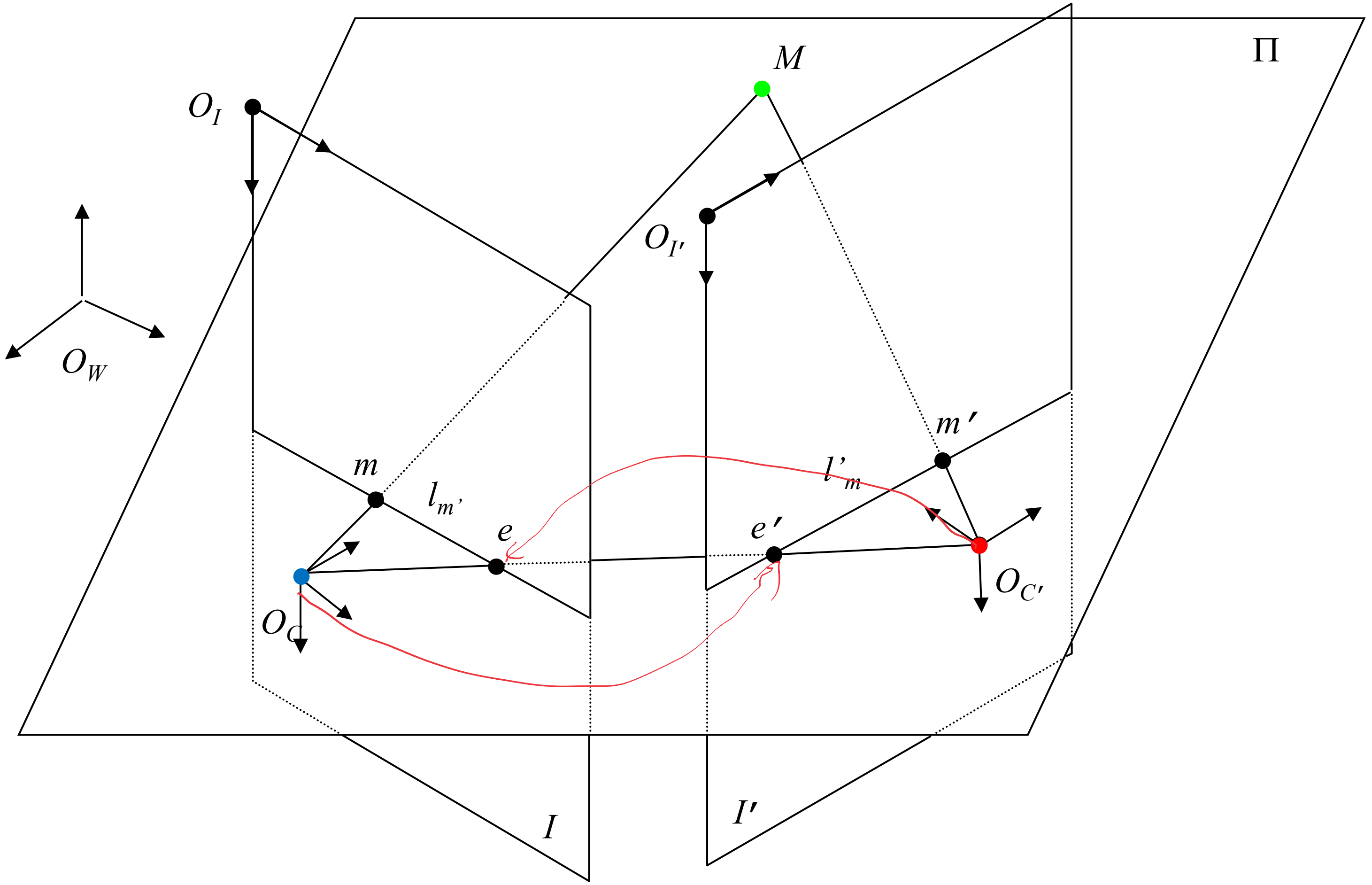


Epipoles and epipolar lines



- e is the projection of $O_{C'}$ in image I,
- e' is the projection of O_C in image I',

Epipoles and epipolar lines

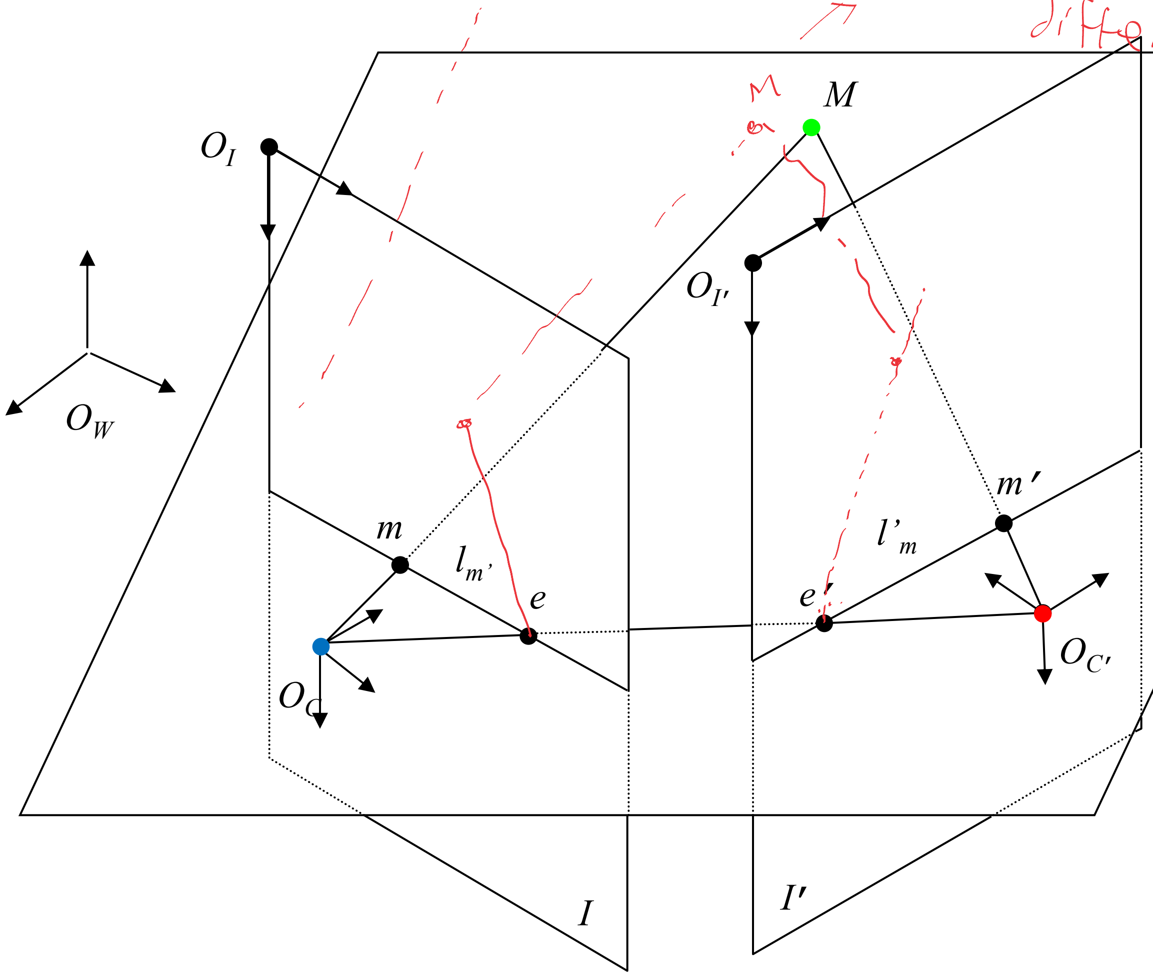


- e is the projection of $O_{C'}$ in image I
- e' is the projection of O_C in image I'
- m and e define an epipolar line in image I'
- m' and e' define an epipolar line in image I

What do the epipolar lines have in common?

Epipoles and epipolar lines

Ask professor to consider different M points to explain.



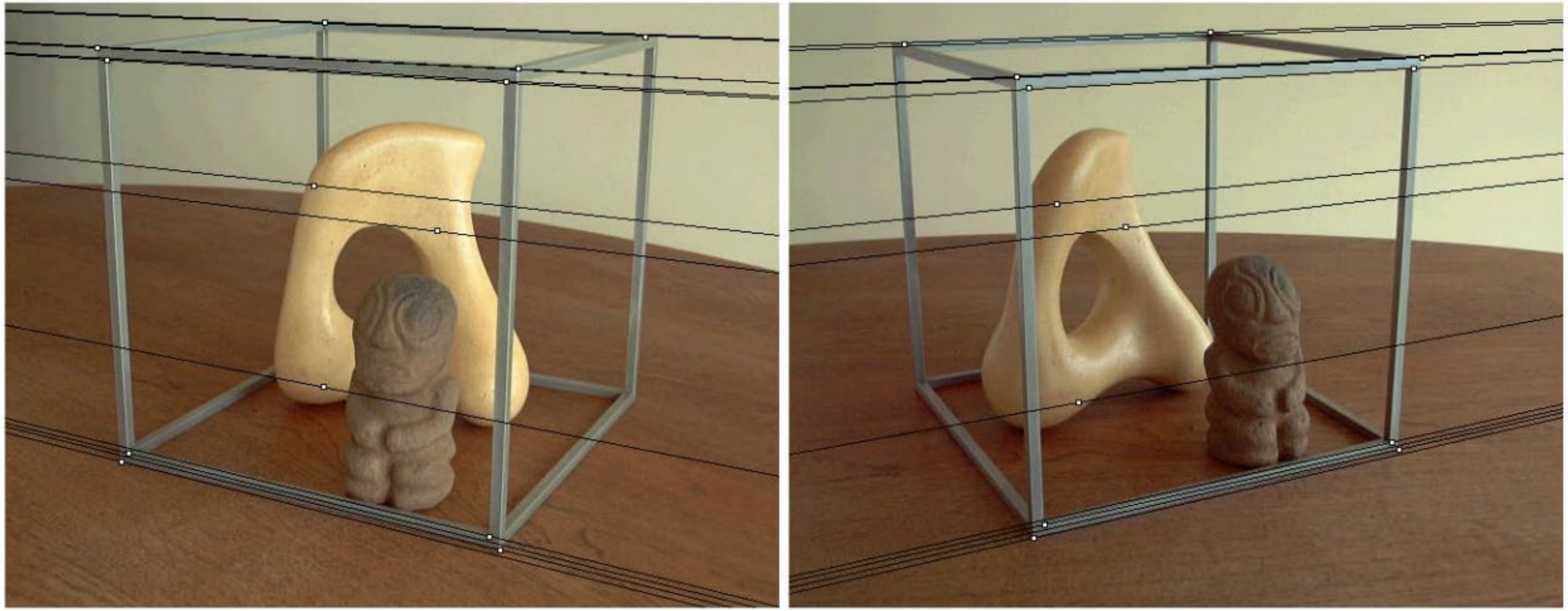
→ epipoles lie in the image plane

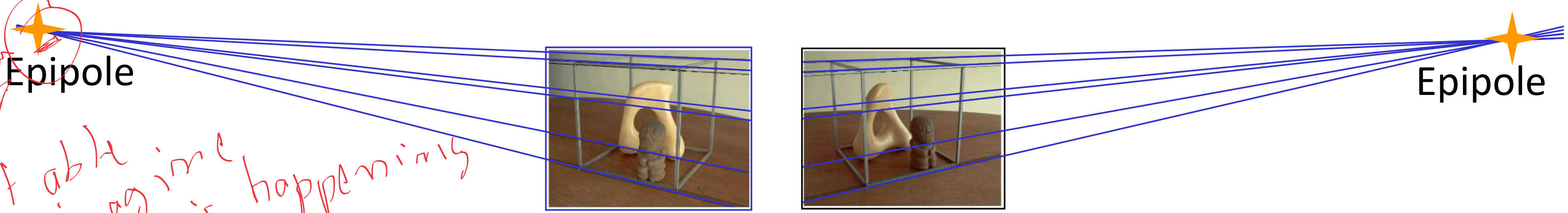
epipoles

- e is the projection of $O_{C'}$ in image I
- e' is the projection of O_C in image I'
- m and e define an epipolar line in image I'
- m' and e' define an epipolar line in image I

Different M points generate different epipolar lines, but they *all go through the epipoles*

Example of epipoles and epipolar lines

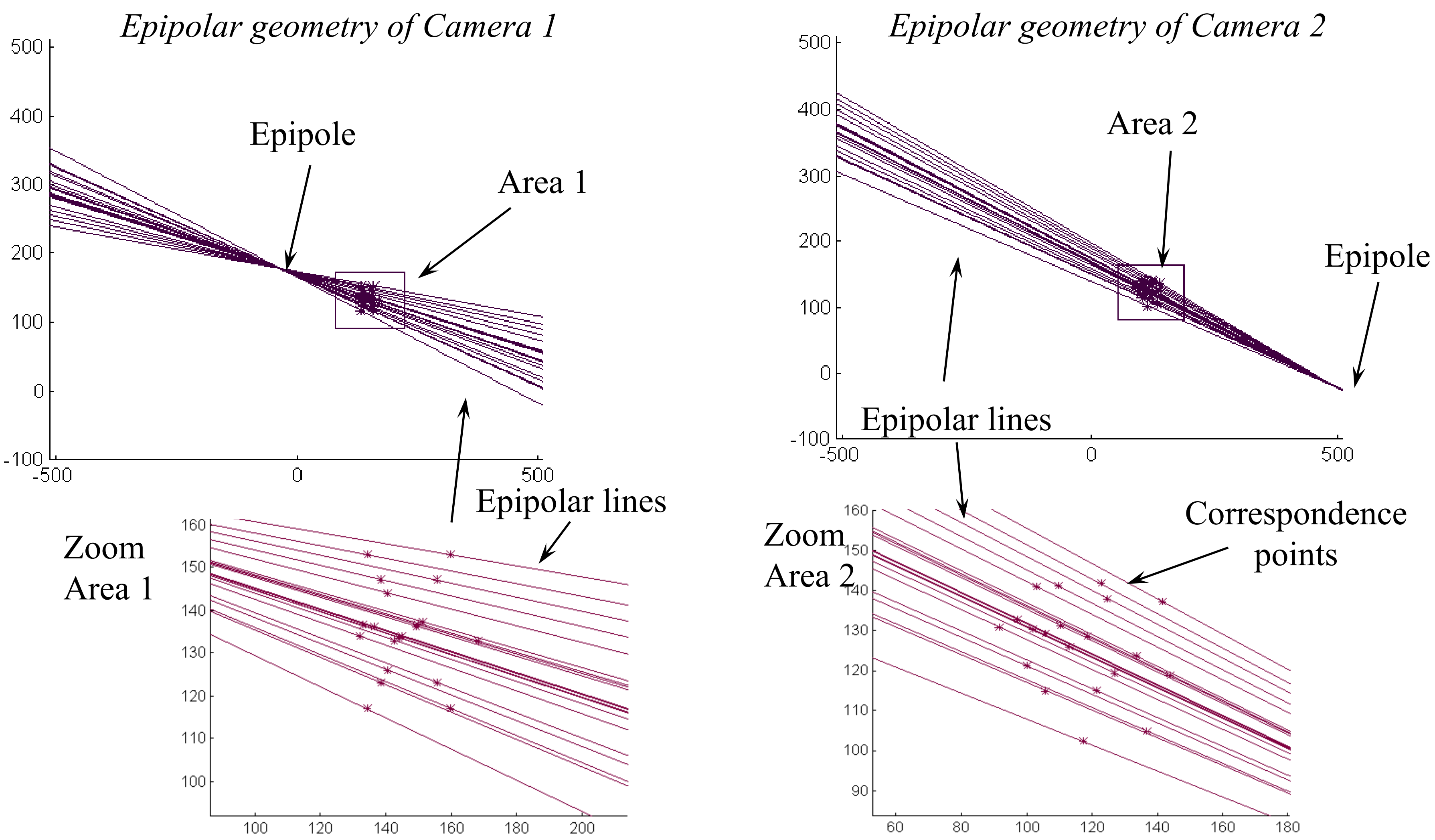




should be in the image plane, right?

not able to imagine what is happening here?

Example of epipoles and epipolar lines



Contents

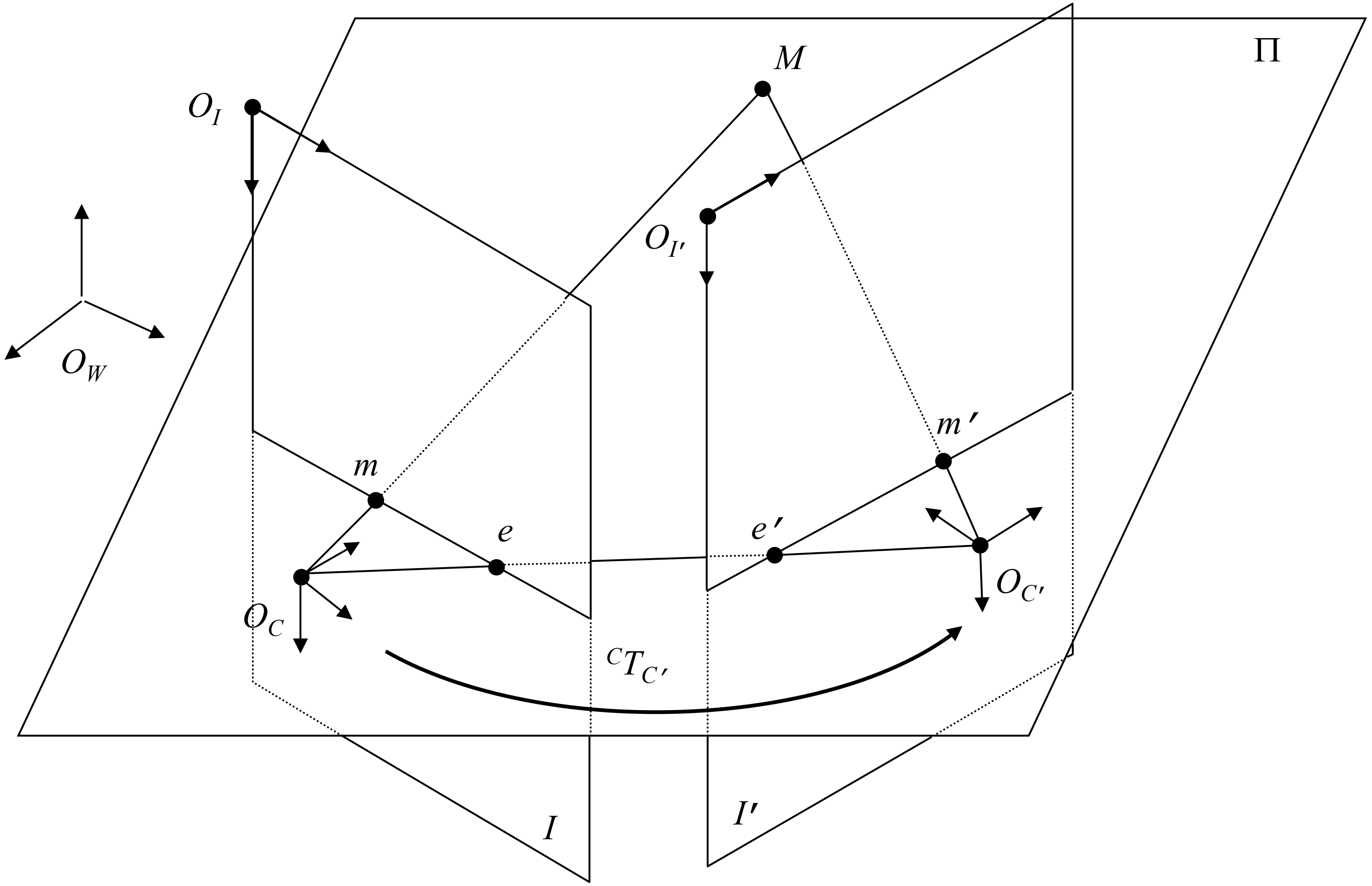
- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Contents

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

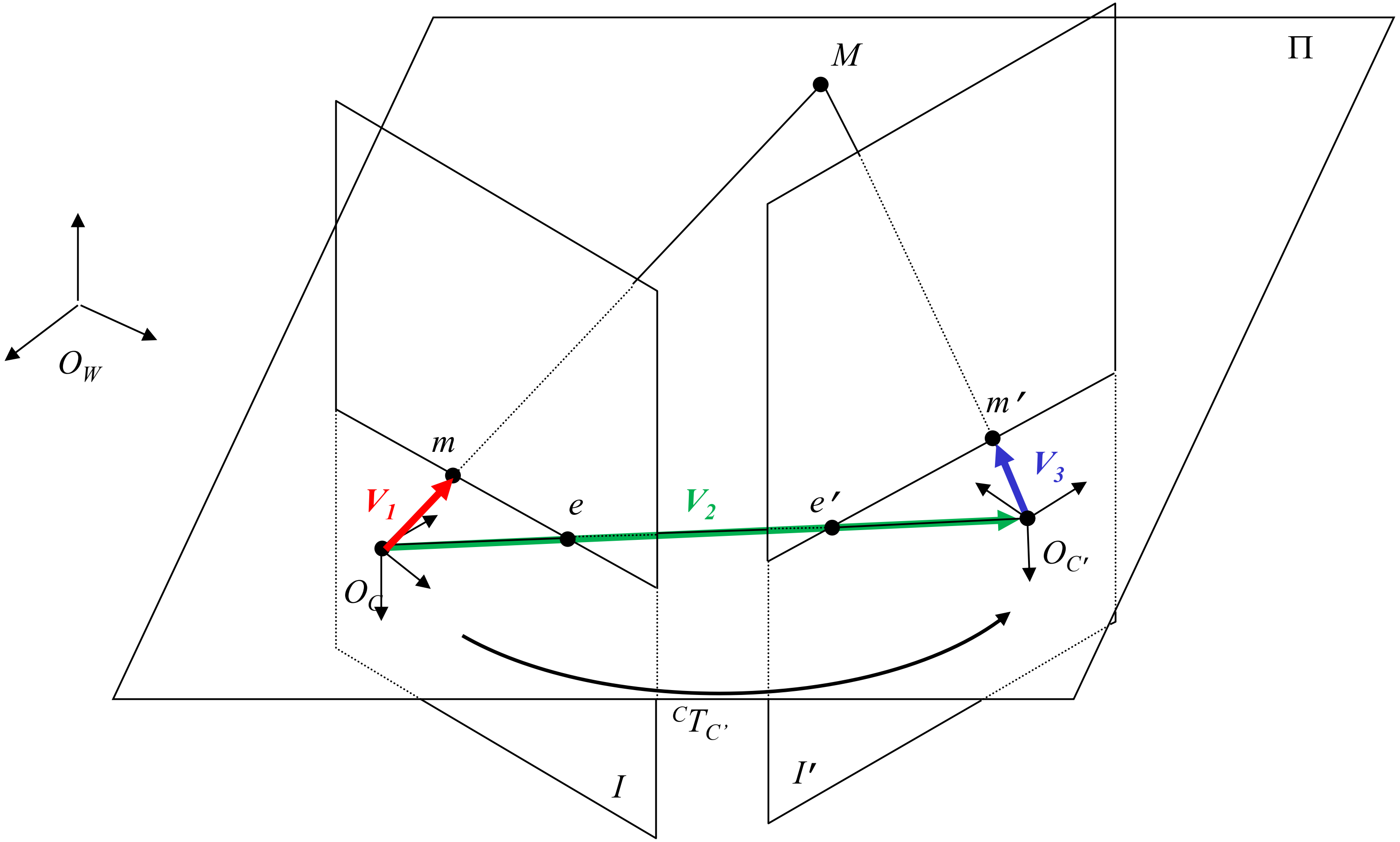
Epipolar Geometry – Modelling



The Epipolar Geometry concerns the problem of computing the plane Π.

- a plane is defined by the cross product between two vectors
- M is unknown, m and m' are observed → pixel locations in image plane, right?

Epipolar Geometry – Modelling



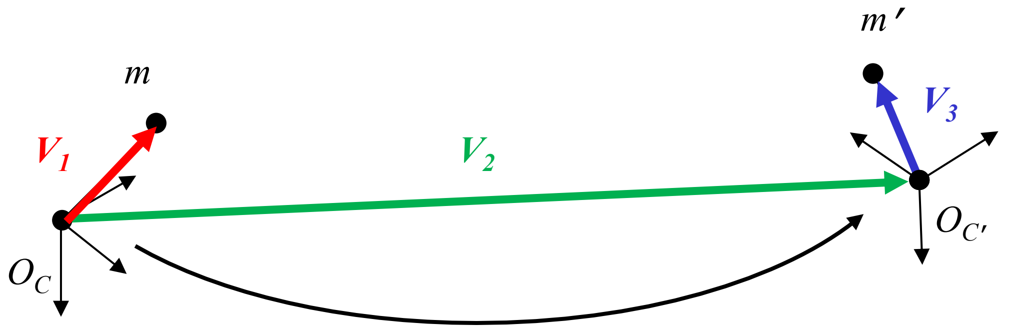
Vectors V_1, V_2 and V_3 are co-planar

We can express this as $V_1 \cdot (V_2 \times V_3) = 0$

Epipolar Geometry – Modelling

Let's assume camera *C* is at the origin, and that both cameras have intrinsics equal to the identity

$$K = I_3 \qquad P = [I_3 \;\; 0]$$
$$K' = I_3 \qquad P' = [R \;\; t]$$



${}^{C}T_{C'} = (R,t)$

Then $V_1 = {}^{\{C\}}\begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix} - O_c = m \qquad V_2 = O_{c'} - O_c = t$

→ expressed in {C}

$V_3 = {}^{\{C'\}}\begin{bmatrix} u_{m'} \\ v_{m'} \\ 1 \end{bmatrix} - O_{c'} = (R \cdot m' + t) - O_c = R \cdot m'$ → where did 't' go?

Epipolar Geometry – Modelling

- We can express co-planarity as $V_1 \cdot (V_2 \times V_3) = 0$, thus

$$m \cdot (t \times R\, m') = 0$$

- or alternatively

$$m^T [t]_\times R\, m' = 0$$

- where $[t]_\times$ is the 3x3 skew symmetric matrix for t that implements the cross product operator

$$[t]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

What is the rank of $[t]_\times$?

Essential matrix E

- We can express co-planarity as $V_1 \cdot (V_2 \times V_3) = 0$

$$m \cdot (t \times R\, m') = 0$$

or

$$m^T \underbrace{[t]_\times R}\, m' = 0$$

$$\boxed{m^T E\, m' = 0}$$

- The 3x3 matrix E is called the **Essential Matrix**

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

UdG

Fundamental matrix F

image coordinate / pixel coordinate system

couldn't understand this point

- Now we consider the intrinsics of the two cameras different from the identity, so that we can use points in pixels instead of metric units.

m was already in image plane, so it must be in pixel coordinate system already, no?

$$\tilde{m} = Km \qquad m = K^{-1}\tilde{m}$$
$$\tilde{m}' = K'm' \qquad m' = K'^{-1}\tilde{m}'$$

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

m & m' were in camera coordinate system

- We can express

$$m^T E\, m' = 0$$

$$(K^{-1}\tilde{m})^T E\, K'^{-1}\tilde{m}' = 0$$

$$\tilde{m}^T \underbrace{K^{-T} E\, K'^{-1}}\, \tilde{m}' = 0$$

$$\boxed{\tilde{m}^T F\, \tilde{m}' = 0}$$

Remember :
$(AB)^T = B^T A^T$
$(A^{-1})^T = (A^T)^{-1} = A^{-T}$

- The 3x3 matrix F is called the **Fundamental Matrix**

Fundamental matrix F properties

not able to understand this statement

The Essential Matrix is the calibrated case of the Fundamental Matrix.
If the intrinsic matrices K and K' are known, then finding F is equivalent to finding E.

how !, K & K' are not same

$$F = K^{-T}[t]_\times R K'^{-1} \qquad\qquad E = [t]_\times R$$

If we reverse the roles of the cameras we can find F' what's the use of F'?

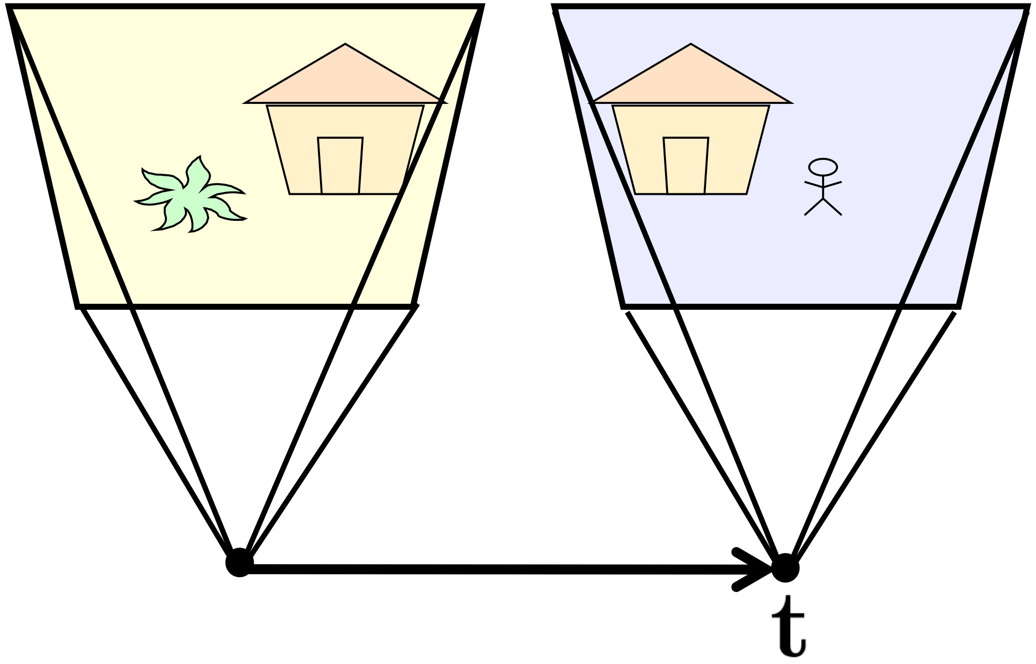
$$\tilde{m}^T F \tilde{m}' = 0 \qquad\qquad F = K^{-T}\, [t]_\times R\, K'^{-1}$$

$$\tilde{m}'^T F' \tilde{m} = 0 \qquad\qquad F' = K'^{-T}\, R^T\, [t]_\times\, K^{-1}$$

K'

Both F and E have rank 2. Why? Answer?

Rectified case



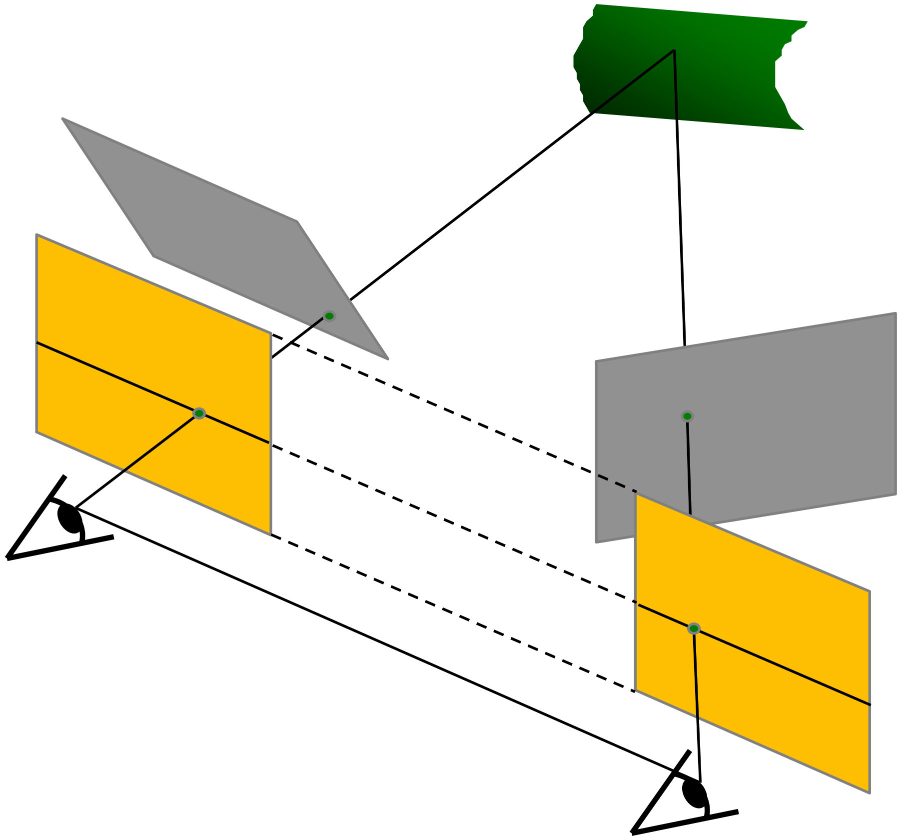
→ what were we trying to understand here?

$$\mathbf{R} = \mathbf{I}_{3\times 3}$$

$$\mathbf{t} = [\ 1 \quad 0 \quad 0\ ]^T$$

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

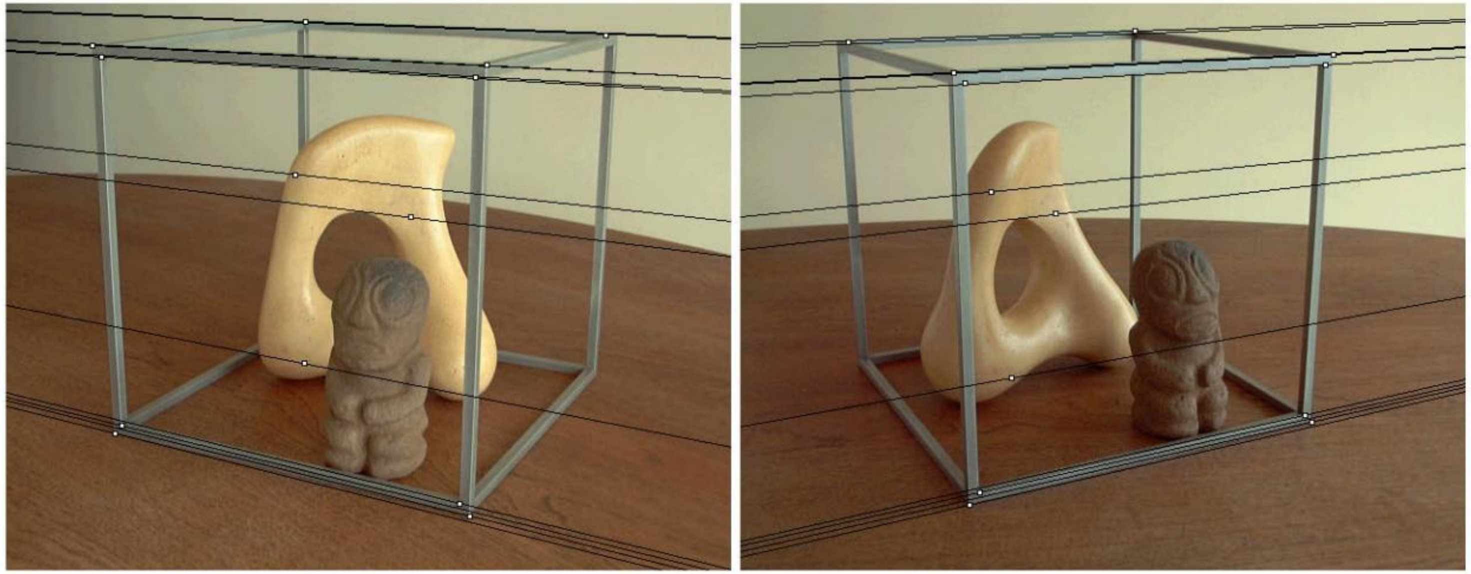
Stereo image rectification



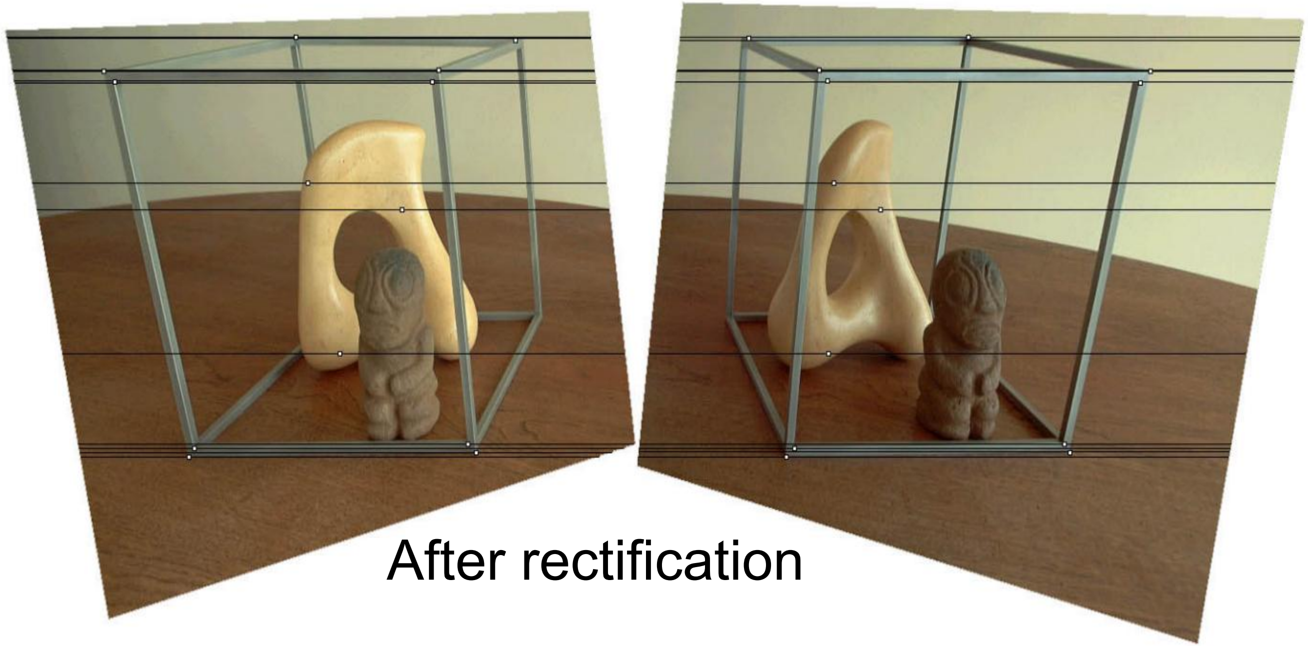
Ask professor to explain these statements

- Goal is to rewarp the images to make epipolar lines become the same horizontal lines on the two images
- Pixel motion is horizontal after this transformation
- Can be done with two homographies, one for each input image

(see C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). CVPR 1999)

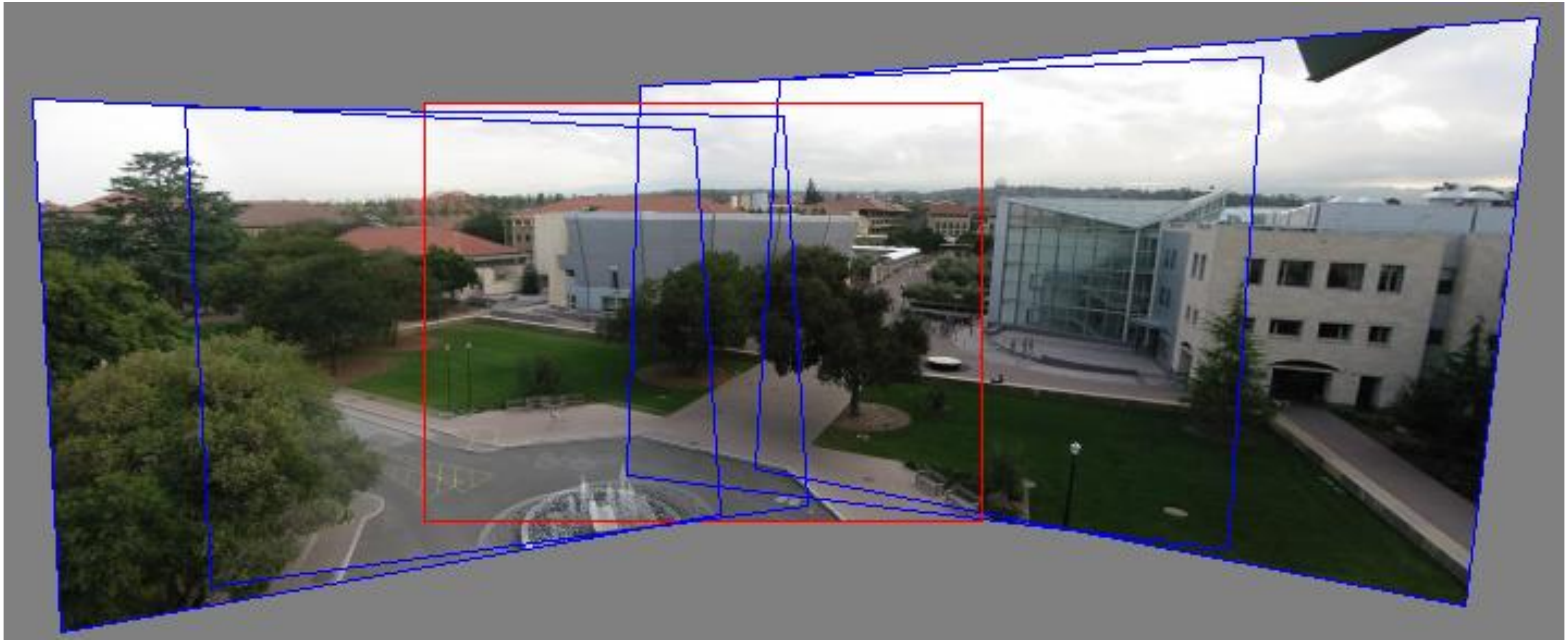


Original stereo pair



After rectification

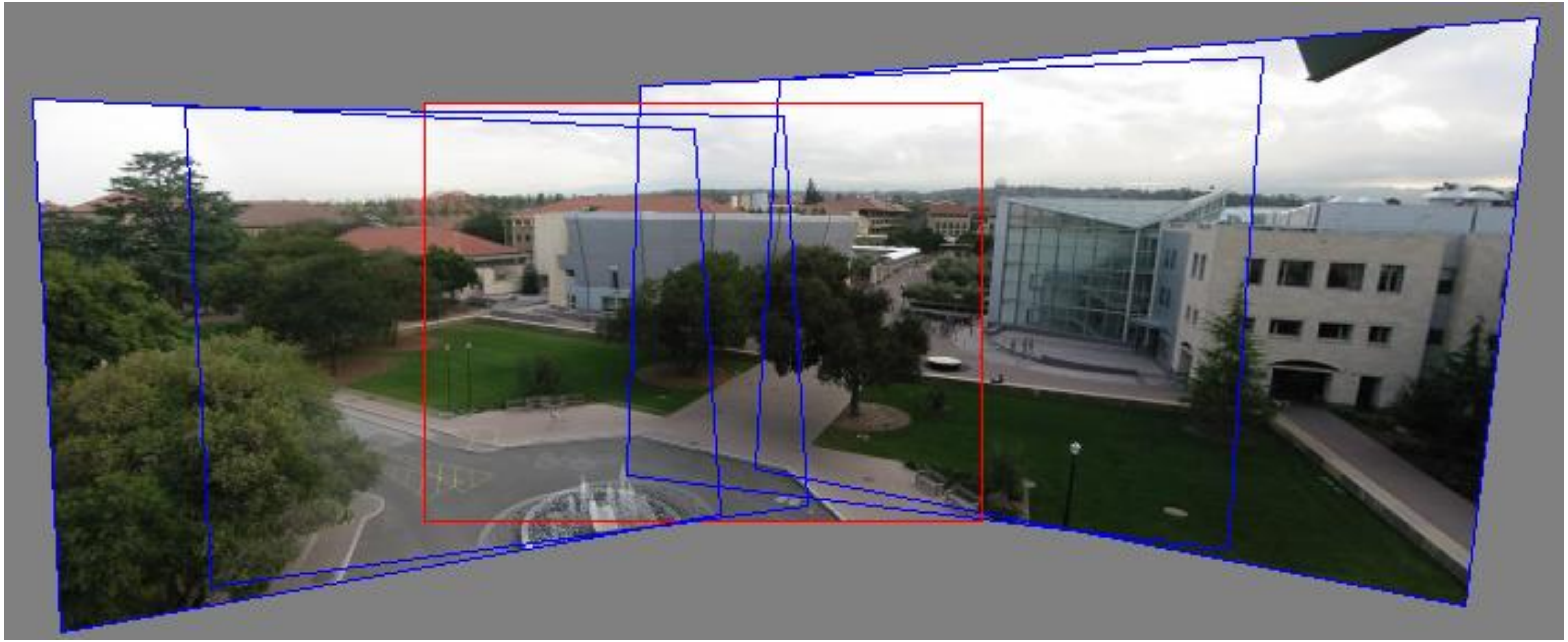
Epipolar Geometry – Modelling



What happens in this case, when there is only camera rotation?

↳ cannot find the fundamental matrix

Epipolar Geometry – Modelling



$$F = K^{-t}[t]_{\times}RK'^{-1}$$

$\neq 0$

$$[t]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Computing the F matrix

The epipolar geometry is described by the fundamental matrix F as:

$$m^T F m' = 0 \qquad [u_i \quad v_i \quad 1]F\begin{bmatrix} u_i{}' \\ v_i{}' \\ 1 \end{bmatrix} = 0$$

Each pair of 2D points provides one equation :

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

$$U_n f = 0$$

$$\begin{bmatrix} u_1u_1{}' & v_1u_1{}' & u_1{}' & u_1v_1{}' & v_1v_1{}' & v_1{}' & u_1 & v_1 & 1 \\ u_2u_2{}' & v_2u_2{}' & u_2{}' & u_2v_2{}' & v_2v_2{}' & v_2{}' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n{}' & v_nu_n{}' & u_n{}' & u_nv_n{}' & v_nv_n{}' & v_n{}' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$$f = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^t$$

Computing the F matrix

Solution 1 - Least Squares

I need to understand this statement, ask professor to explain this

$\boldsymbol{F}$ is defined up to a scale factor, so we can fix one of the component to 1. Let's fix $f_{33} = 1$.

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + 1 = 0$$

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}$$

Solution: $$f' = -U_n'^{+} 1_n$$

Ask professor to explain this

Computing the F matrix

Solution 2 - Solving homogenous system

what do we mean by null space of U_n?

$U_n f = 0$ is a homogenous set of equations. The solution is null-space of U_n. We want to find $\min_f \|U_n f\|$, subjected to $\|f\| = 1$

where did this come from?

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Solution: Use the Singular Value Decomposition to approximate the null-space of U_n.
This will be the last column of V, that corresponds to the smallest singular value.

$$\mathbf{U\Sigma V^T} = \text{SVD}\,(U_n)$$

$$f = V(:,9)$$

Computing the F matrix

F *should have rank 2. The previous solutions do not enforce it.*

To enforce F to be rank 2, we can find F_{r2} that minimizes $||F - F_{r2}||$ subject to the rank constraint.

This is achieved by SVD again.

Let $\mathbf{U\Sigma V}^{\mathrm{T}} = \mathrm{SVD}(F)$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \quad \text{let} \quad \Sigma_{r2} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

then $F_{r2} = \mathbf{U}\Sigma_{r2}\mathbf{V}^T$ is the closest rank-2 approximation to F.

8-point algorithm

```
# x1, x2 are 2×N arrays (each column a point); npts = number of points
# Example: x1.shape = (2, npts), x2.shape = (2, npts)

    npts = x1.shape[1]

    # Build the constraint matrix
    U_n = np.column_stack([
        x2[0, :] * x1[0, :],   x2[0, :] * x1[1, :],  x2[0, :],
        x2[1, :] * x1[0, :],   x2[1, :] * x1[1, :],  x2[1, :],
        x1[0, :],  x1[1, :],   np.ones(npts)
    ])


    # Singular Value Decomposition
    U, D, Vt = np.linalg.svd(U_n)

    # Extract fundamental matrix from the column of V
    # corresponding to the smallest singular value
    F = Vt[-1, :].reshape(3, 3).T

    # Enforce rank-2 constraint
    U, D, Vt = np.linalg.svd(F)
    D[2] = 0  # Force last singular value to zero
    F_rank2 = U @ np.diag(D) @ Vt
```

Computing the F matrix

$$
\begin{array}{c}
\begin{bmatrix}
u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\
u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1
\end{bmatrix} \\
\begin{array}{ccccccccc}
\sim 10000 & \sim 10000 & \sim 100 & \sim 10000 & \sim 10000 & \sim 100 & \sim 100 & \sim 100 & 1
\end{array}
\end{array}
\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0
$$

Orders of magnitude difference between column of data matrix
→ least-squares yields poor results

- Transform input by $\hat{\mathbf{x}}_\mathbf{i} = \mathbf{T}\mathbf{x}_\mathbf{i}$,
- Estimate $\hat{\mathbf{F}}$ from $\hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{x}}_\mathbf{i}'$
- Invert transformation by $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}$

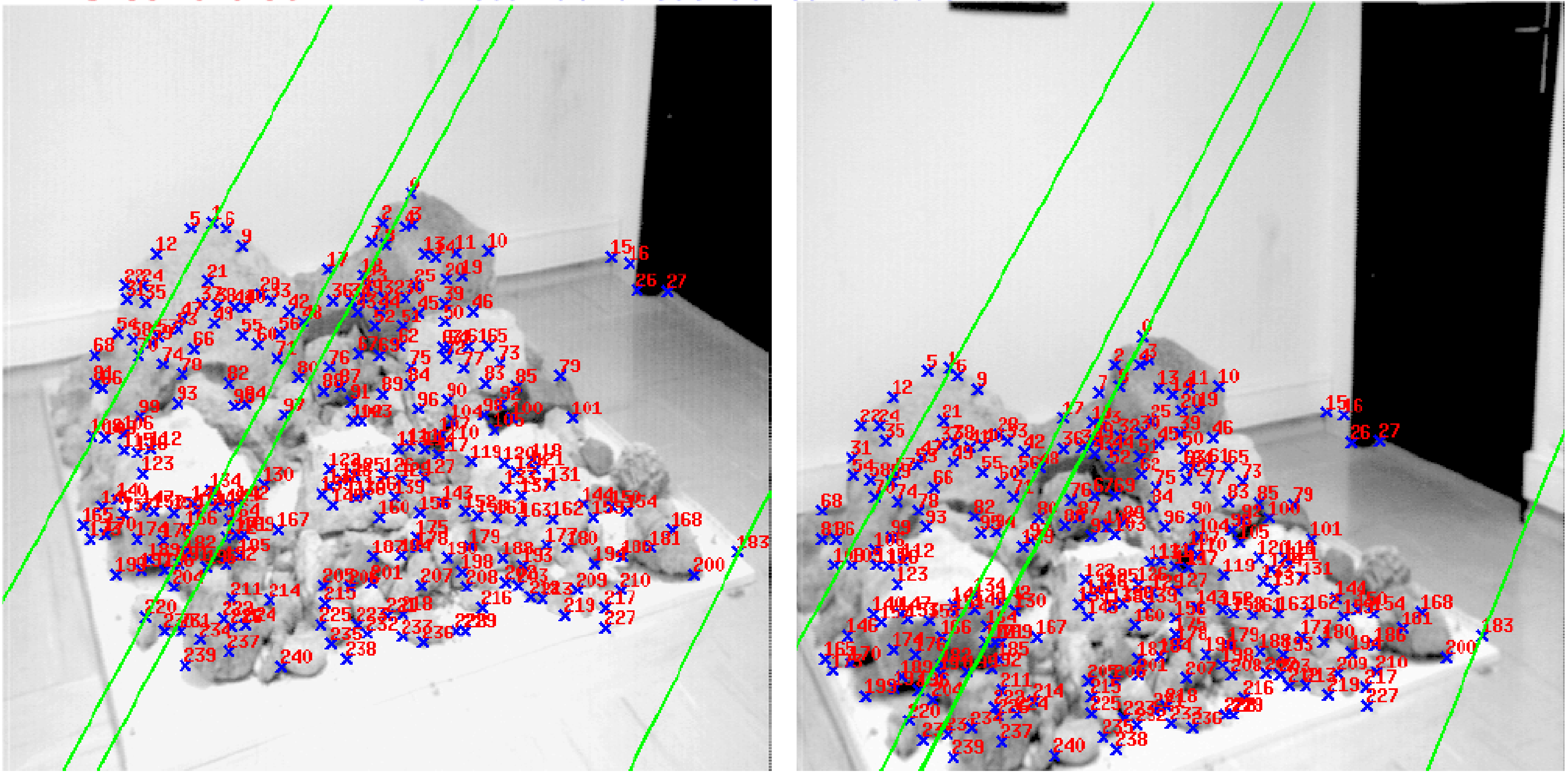
$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

$$\boxed{\hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{T}'^{-\mathrm{T}}}\,\mathbf{F}\,\boxed{\mathbf{T}^{-1}\hat{\mathbf{x}}} = 0$$

$$\underbrace{\mathbf{T}'^{-\mathrm{T}}\mathbf{F}\mathbf{T}^{-1}}_{\hat{\mathbf{F}}}$$

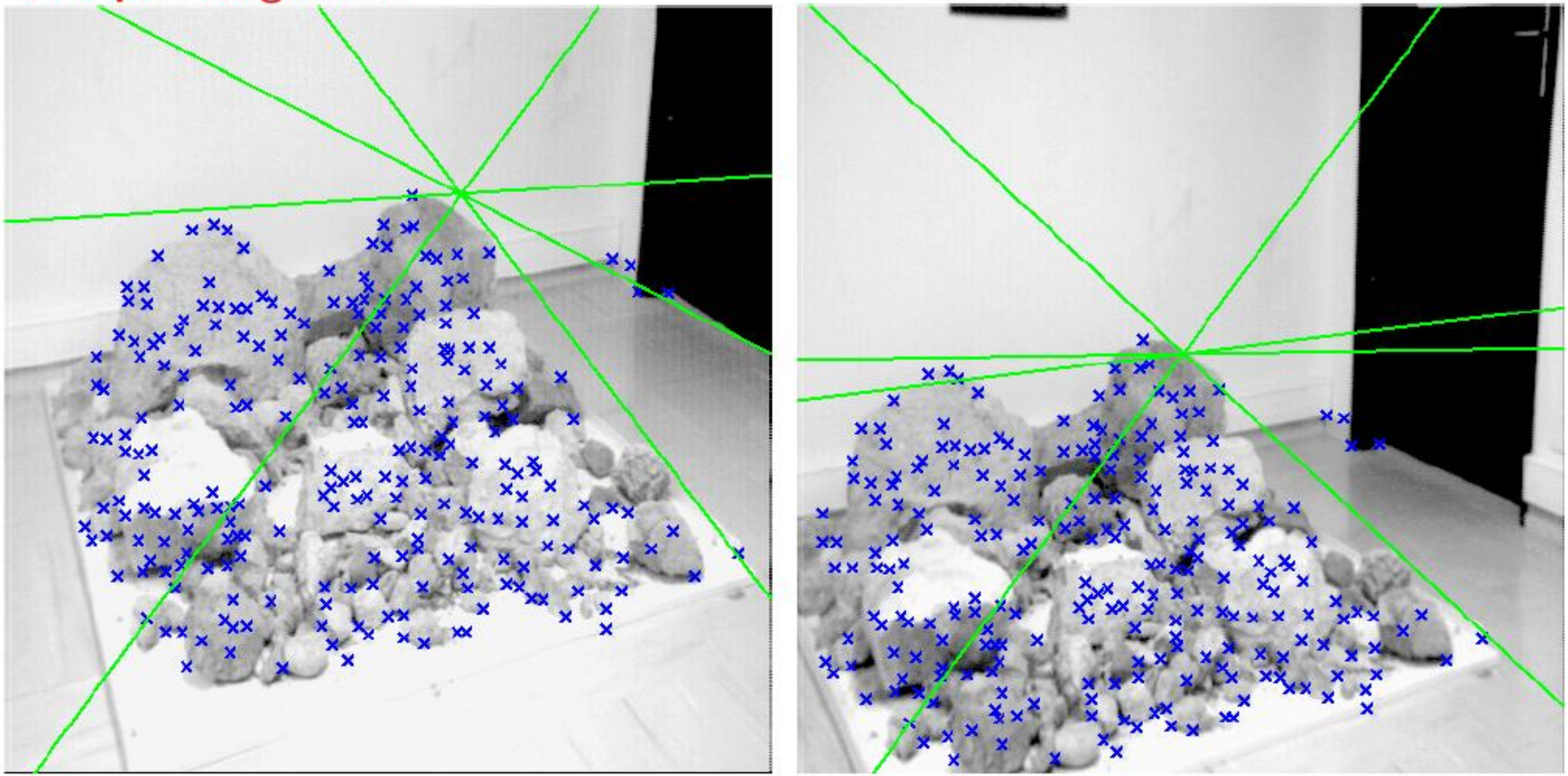
Results (ground truth)

- **Ground truth** with standard stereo calibration

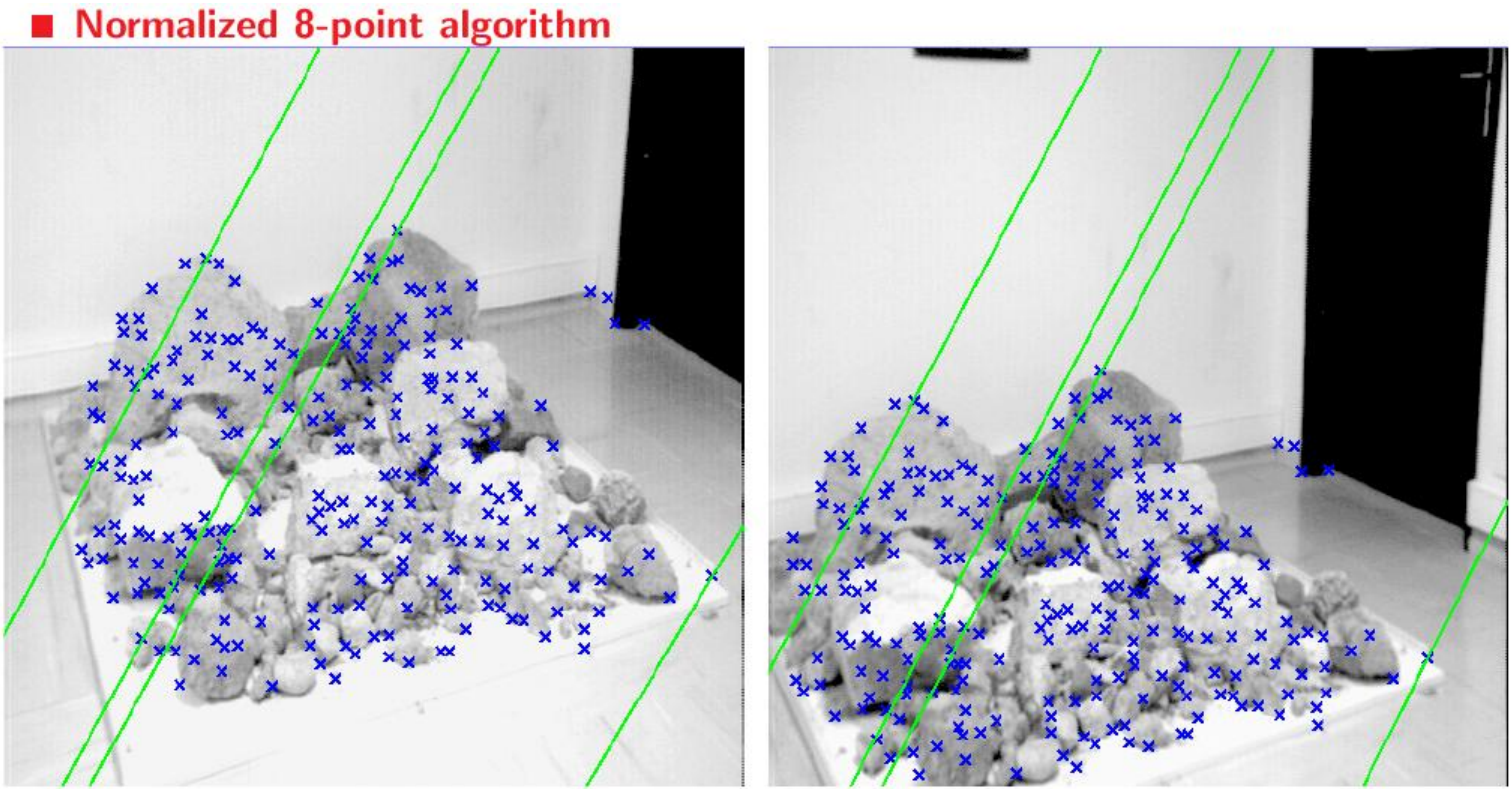


Results (8-point algorithm)

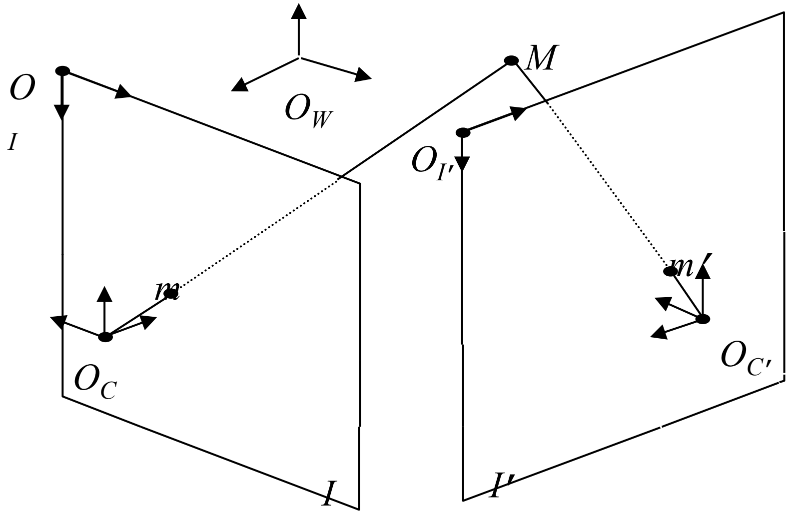
- 8-point algorithm



Results (normalized 8-point algorithm)



Epipolar Geometry – Configurations



Q: What can we compute from a given stereo pair?
A: Depending on the information available, we can compute different things:

Configurations:

- *Calibrated Stereo*: Intrinsics and Extrinsics known → Triangulation!
- *Uncalibrated Stereo*: Intrinsics and Extrinsics unknown → F matrix
- *Calibrated Monocular*: Intrinsics known, Extrinsics unknown → E matrix
- *Uncalibrated Monocular*: Intrinsics and Extrinsics unknown → F matrix

Contents

- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

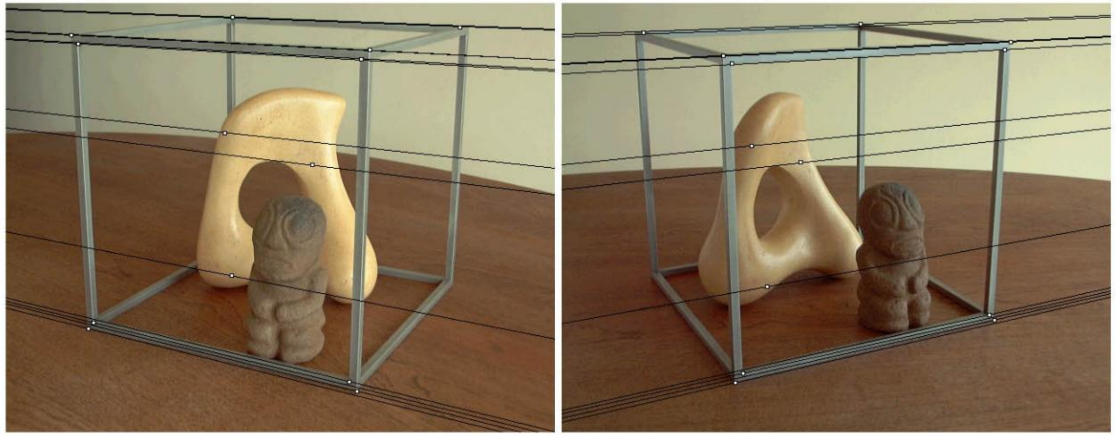
Contents

Contents

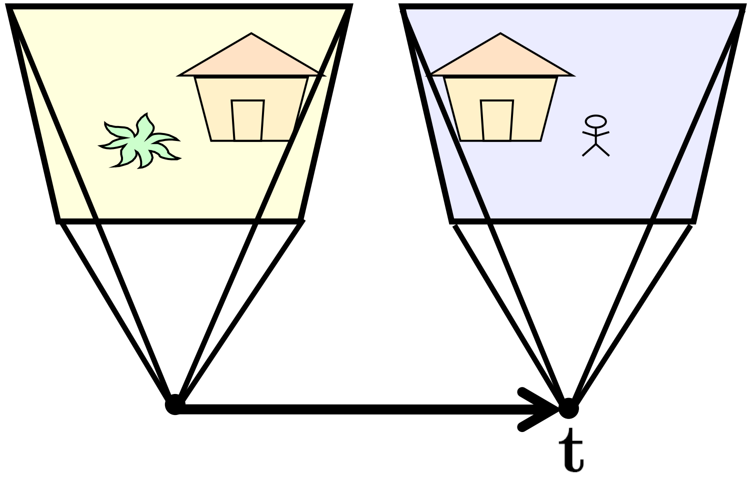
- Shape from *X*
- Triangulation principle
- Epipolar geometry – Epipoles and epipolar lines
- Epipolar geometry – The Fundamental matrix
- Estimating the Fundamental matrix
- Reconstruction from 2 views

Reconstruction from 2 views

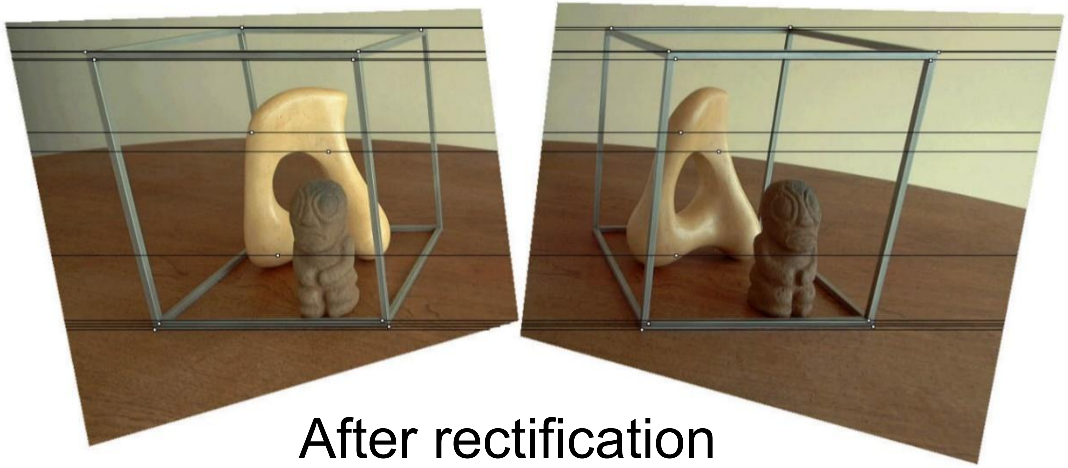
Let go back to the case of stereo rectification



Original stereo pair



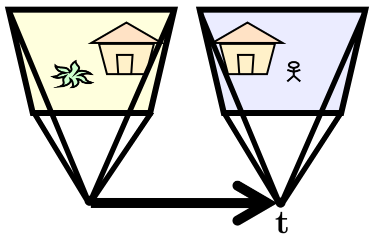
No Rotation, translation along X-axis

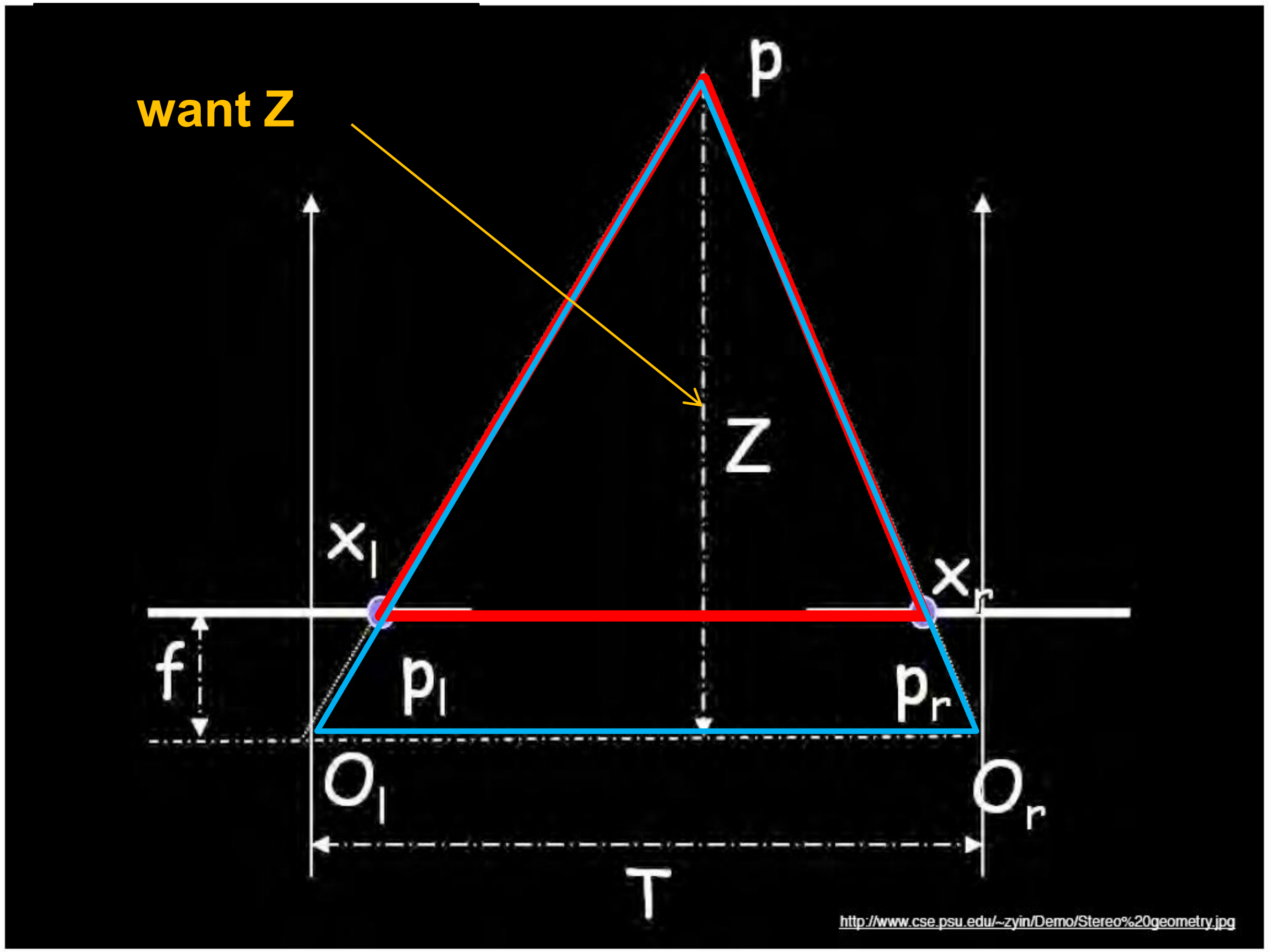


After rectification

Reconstruction from 2 views

Assume **parallel** optical axes, known camera parameters.





Jse similar triangles (p_l, P, p_r) and
O$_l$, P, O_r):

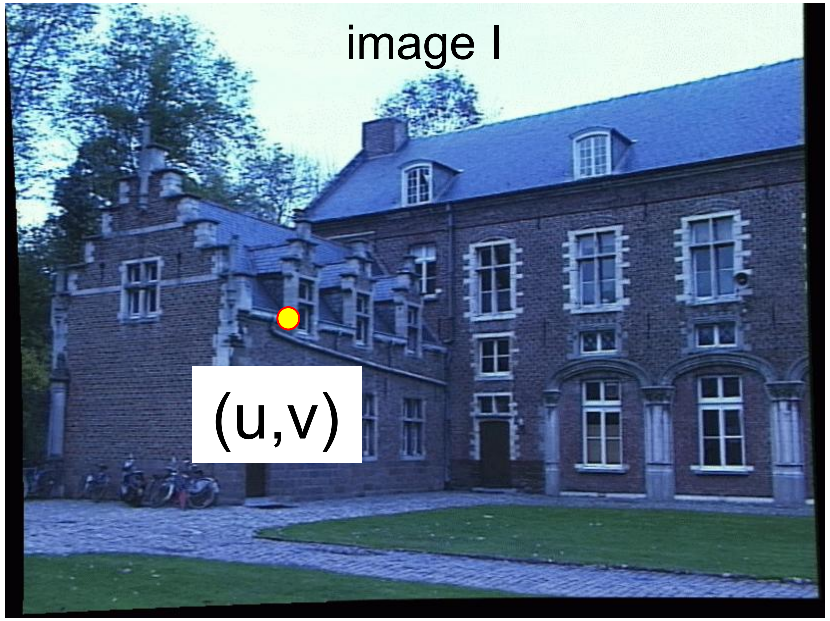
$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

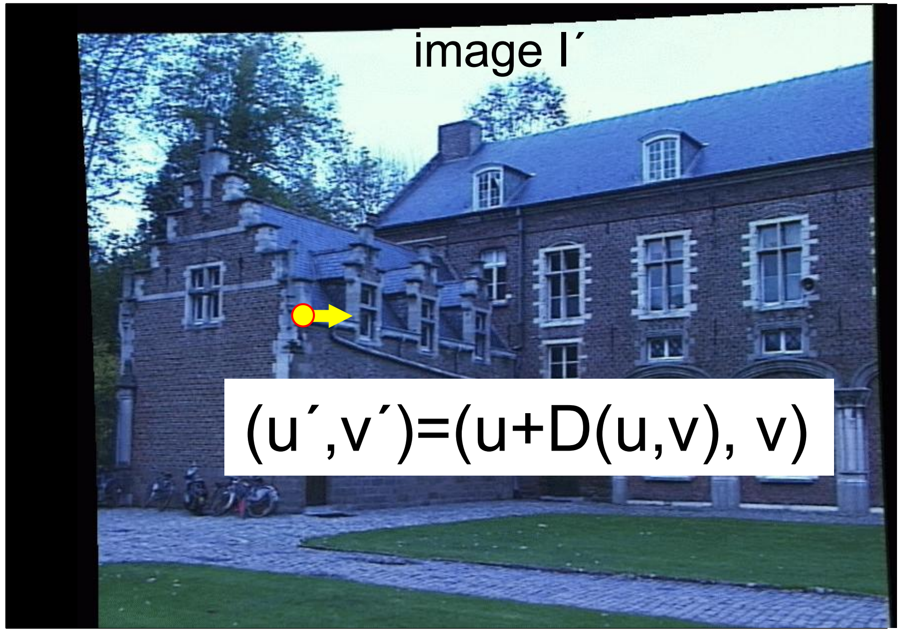
$$Z = f\frac{T}{x_r - x_l}$$

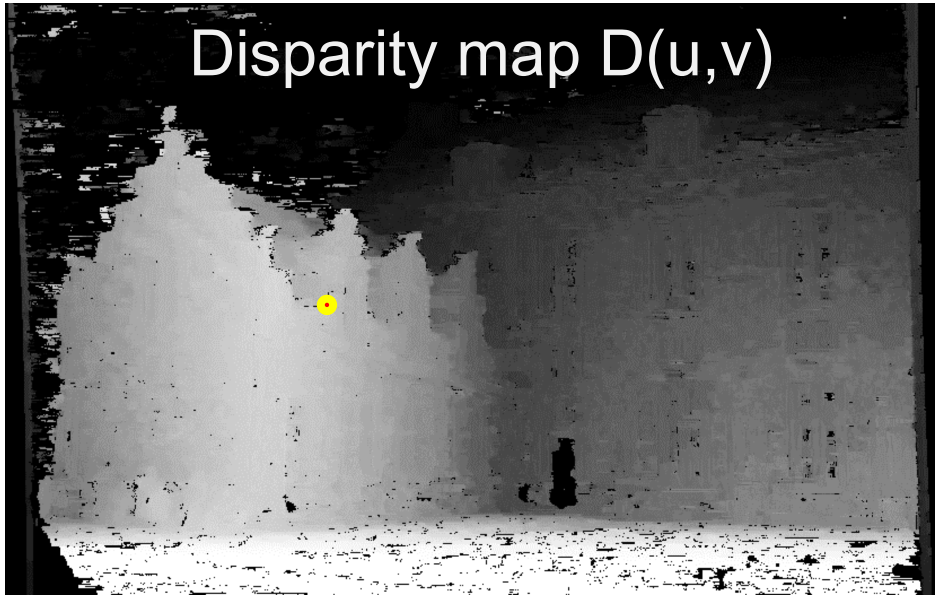
disparity

Grauman

Disparity map



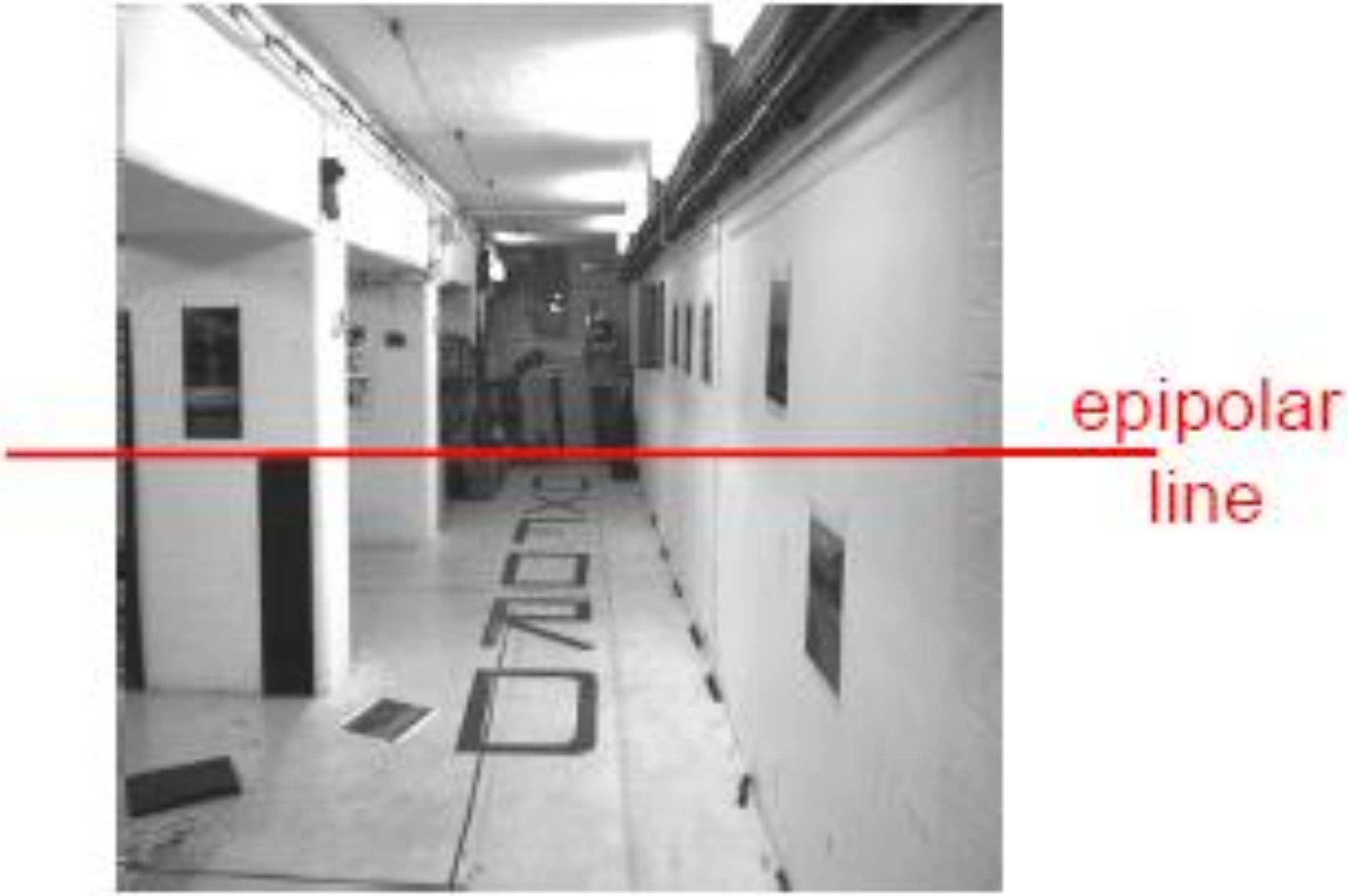




Grauman

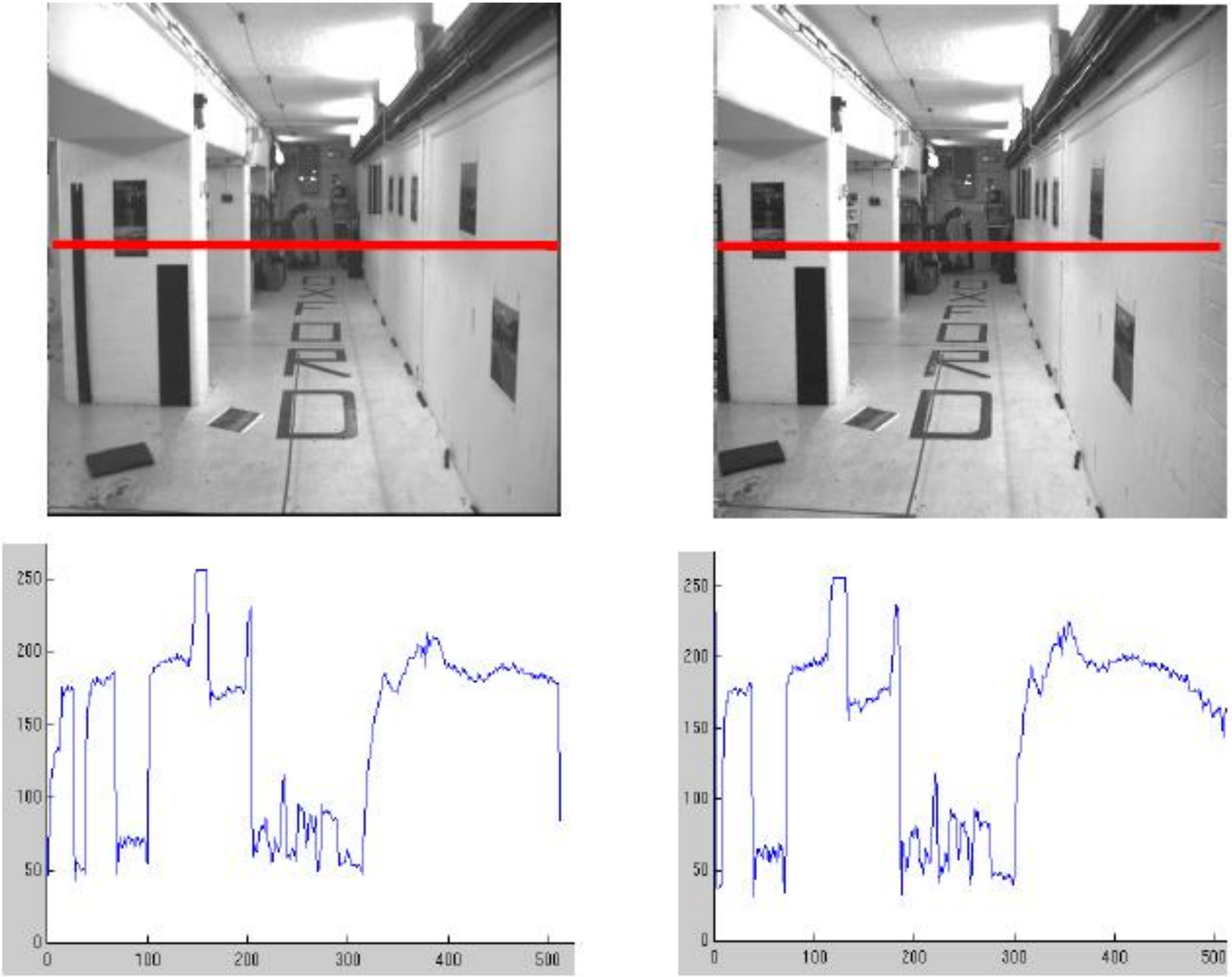
Correspondence problem

Two rectified views – a point in one image corresponds to

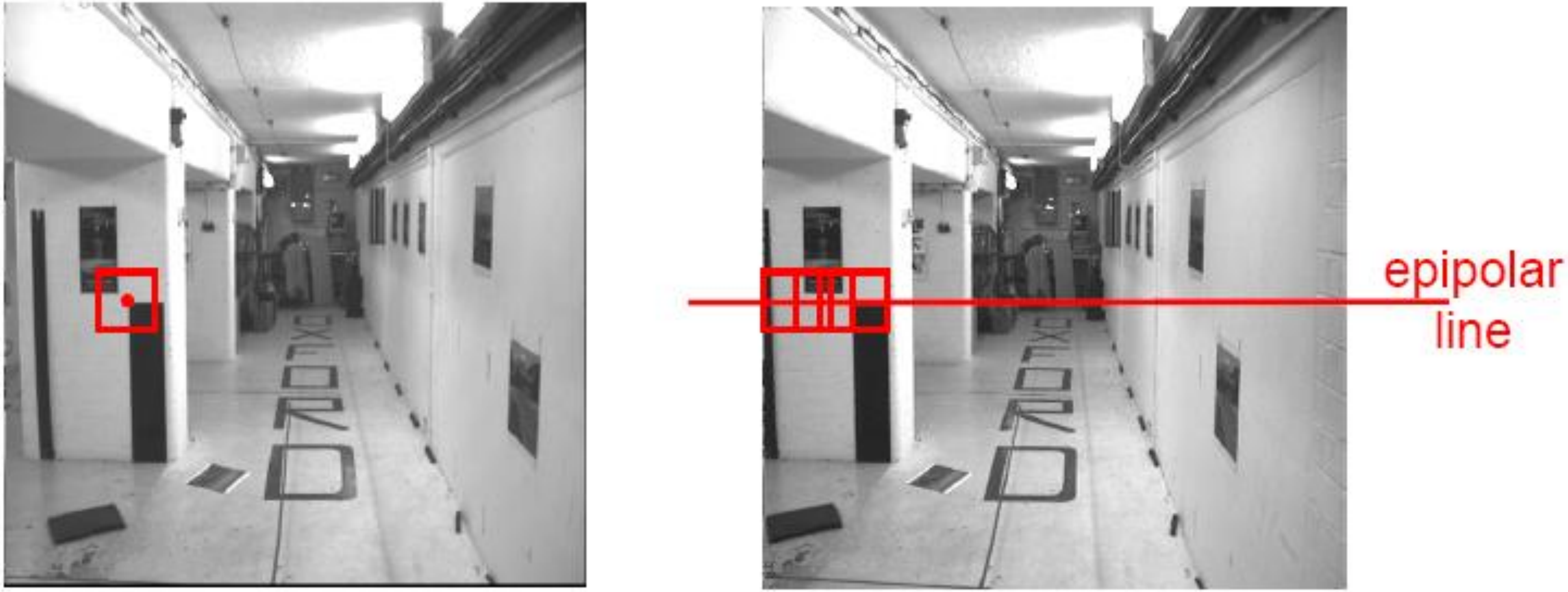


adapted from A. Zisserman

Reconstruction from 2 views – intensity profiles



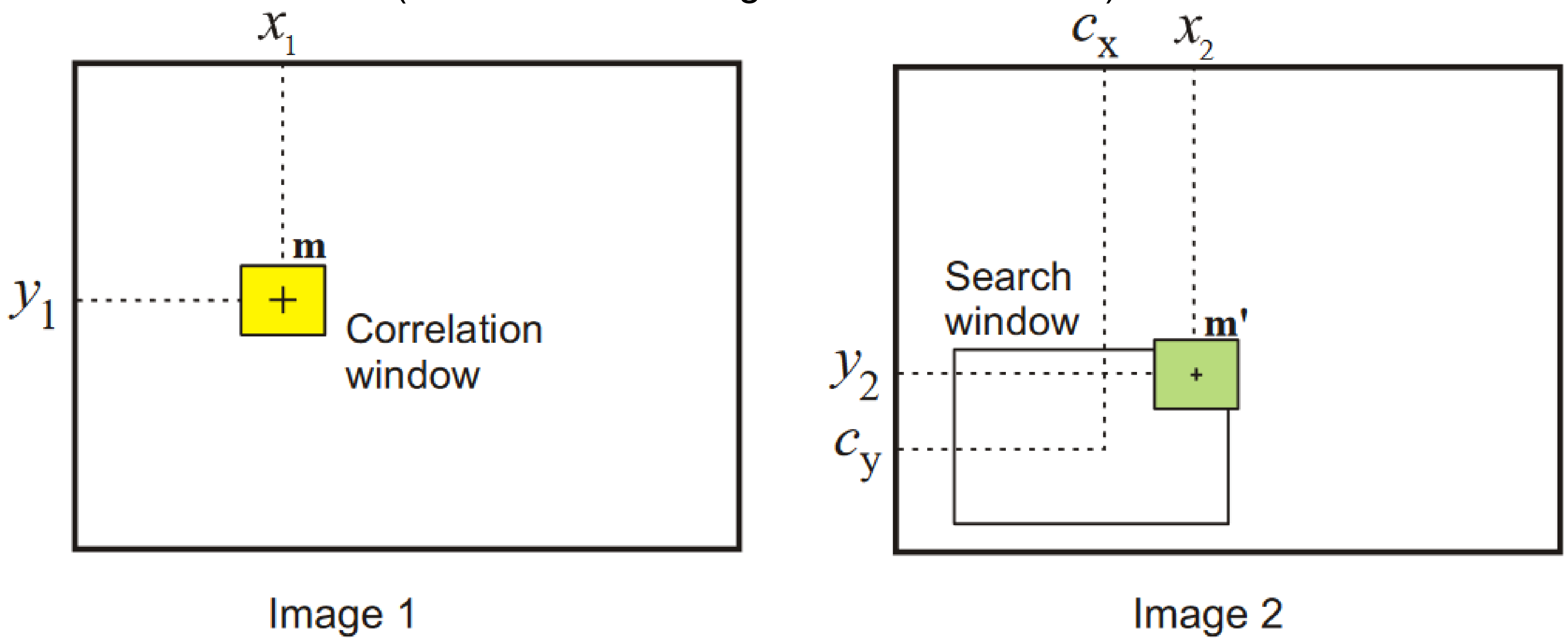
- Clear correspondence between intensities, but also noise and ambiguity



Neighborhood of corresponding points have similar intensity patterns.

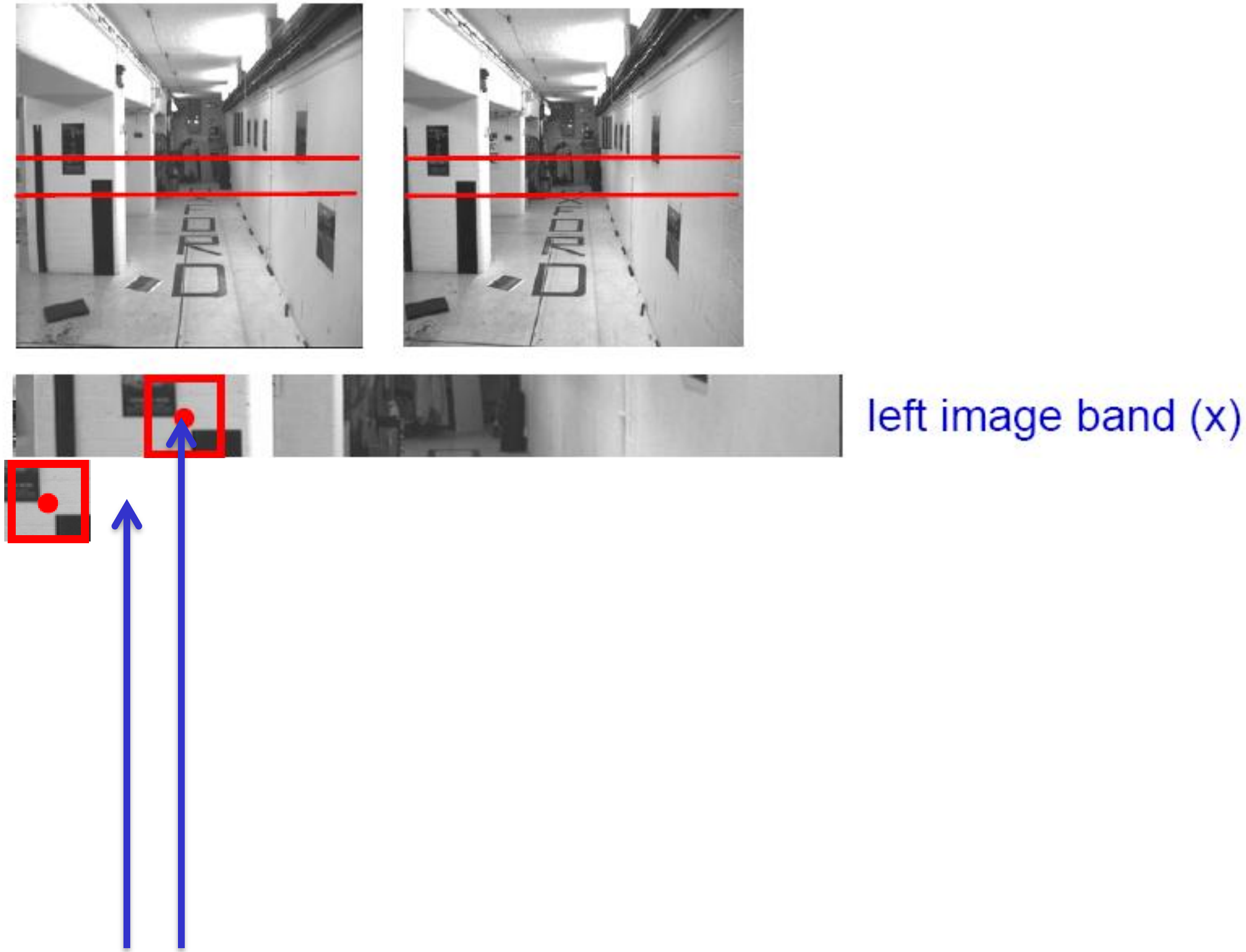
Normalized cross correlation

(remember the Image Primitives lecture)

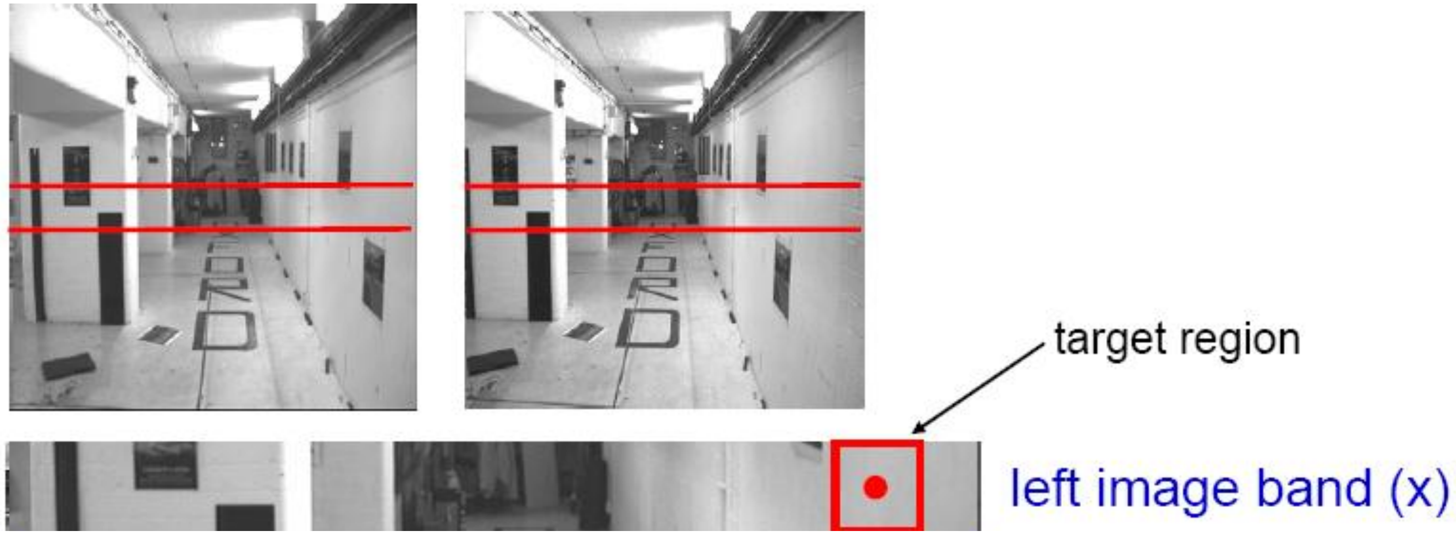


$$corr(\mathbf{m},\mathbf{m}') = \frac{\sum_{i=-n/2}^{n/2}\sum_{j=-n/2}^{n/2}\left[I_1(x_1+i,y_1+j)-\overline{I_1(x_1,y_1)}\right]\cdot\left[I_2(x_2+i,y_2+j)-\overline{I_2(x_2,y_2)}\right]}{(n+1)^2\sqrt{\sigma^2(I_1)\cdot\sigma^2(I_2)}}$$

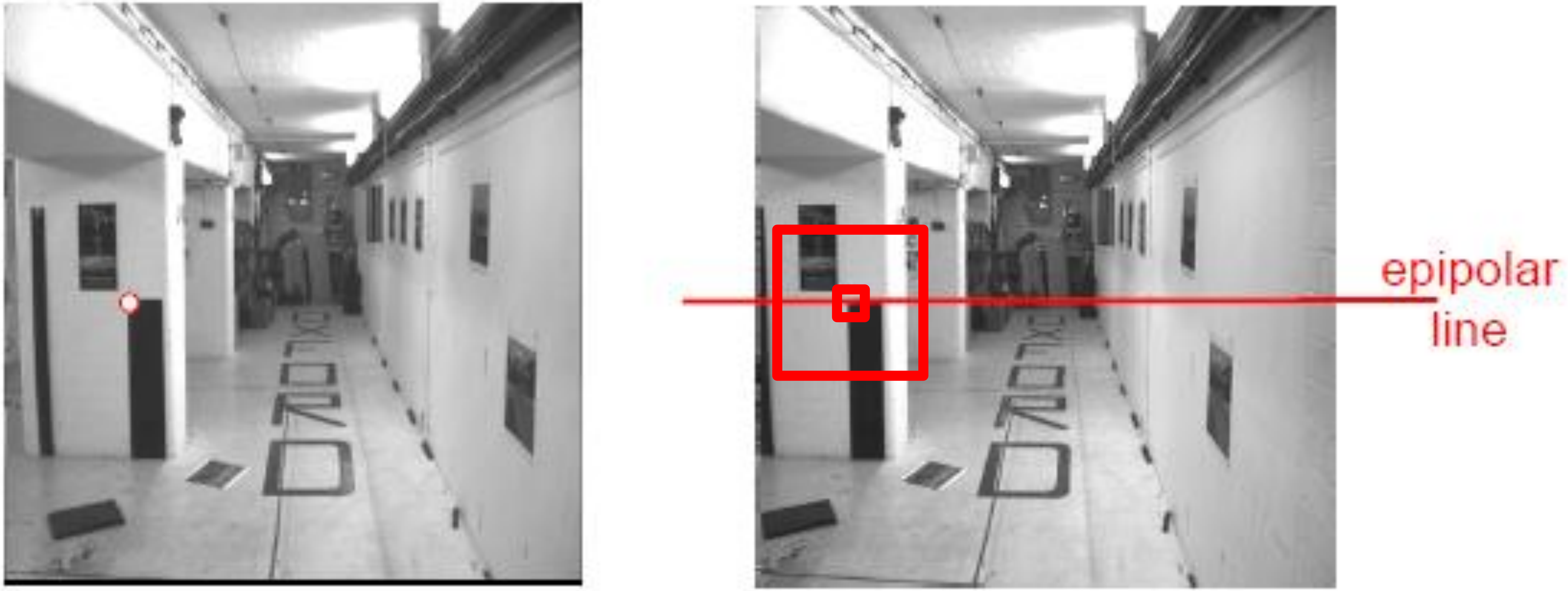
Correlation-based window matching



Textureless regions



So which window size shall we choose?



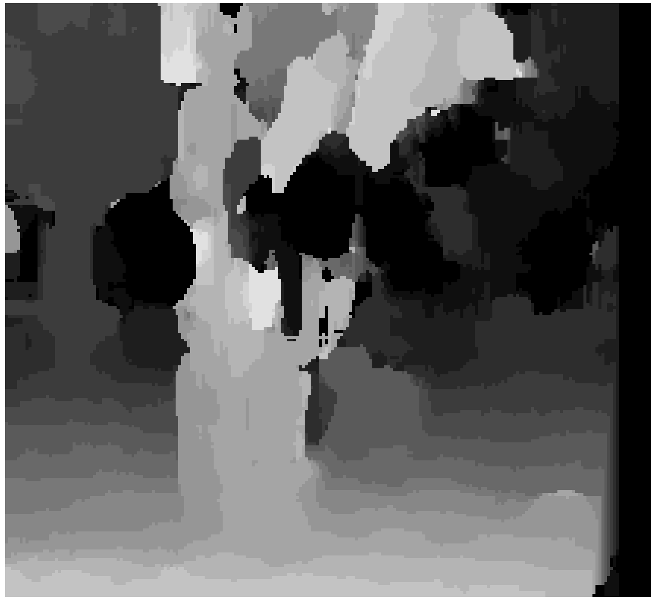
Effect of window size

We need a window :

- large enough to have sufficient intensity variation, but
- small enough to contain pixels of just the same disparity

This is a difficult trade-off

Patch Size = 3x3

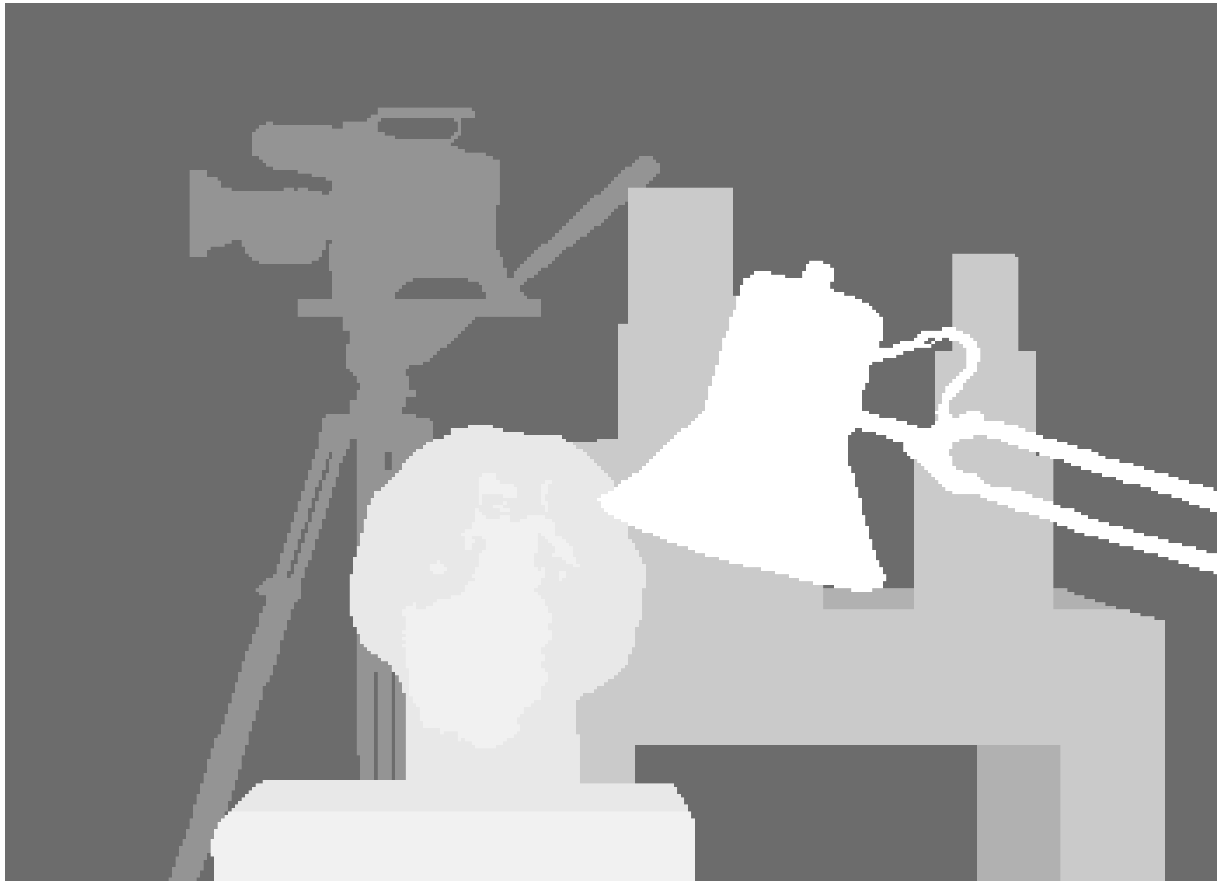


Patch Size = 20x20

Figures from Li Zhang

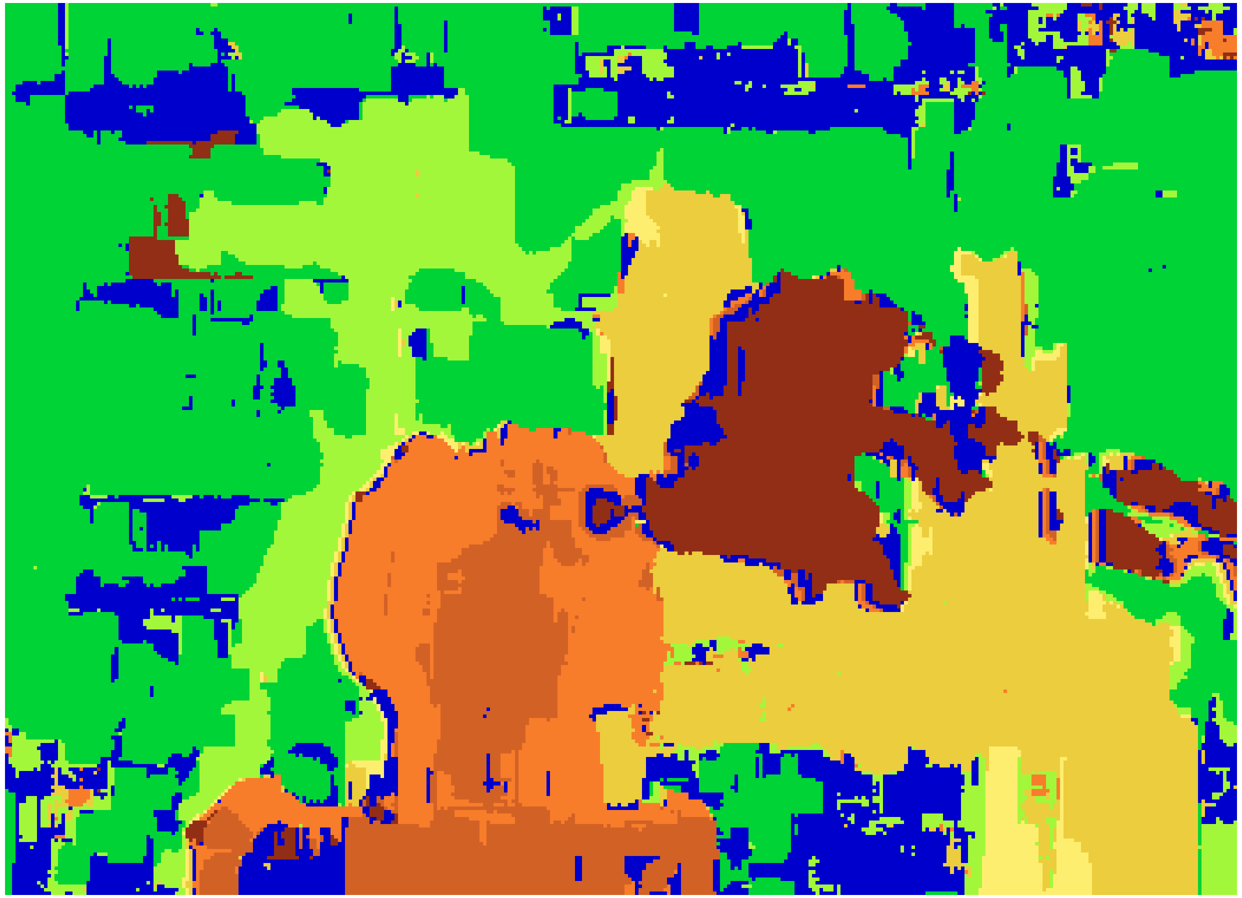
Results with window search

Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Results with window search

Energy Minimization

Boykov *et al.,* Fast Approximate Energy Minimization via Graph Cuts, ICCV, Sept 1999.

Ground truth