CSCI 4830 / 5722
Computer Vision

University of Colorado **Boulder**

CSCI 4830 / 5722
Computer Vision

Dr. Ioana Fleming
Spring 2019
Lecture 19

University of Colorado **Boulder**

Reminders

Submissions:

- Homework 3: Sat 3/2 at 11 pm

Readings:

- Szeliski:
    - chapter 11 (Stereo correspondence)
- P&F:
    - chapter 7 (Stereopsis)

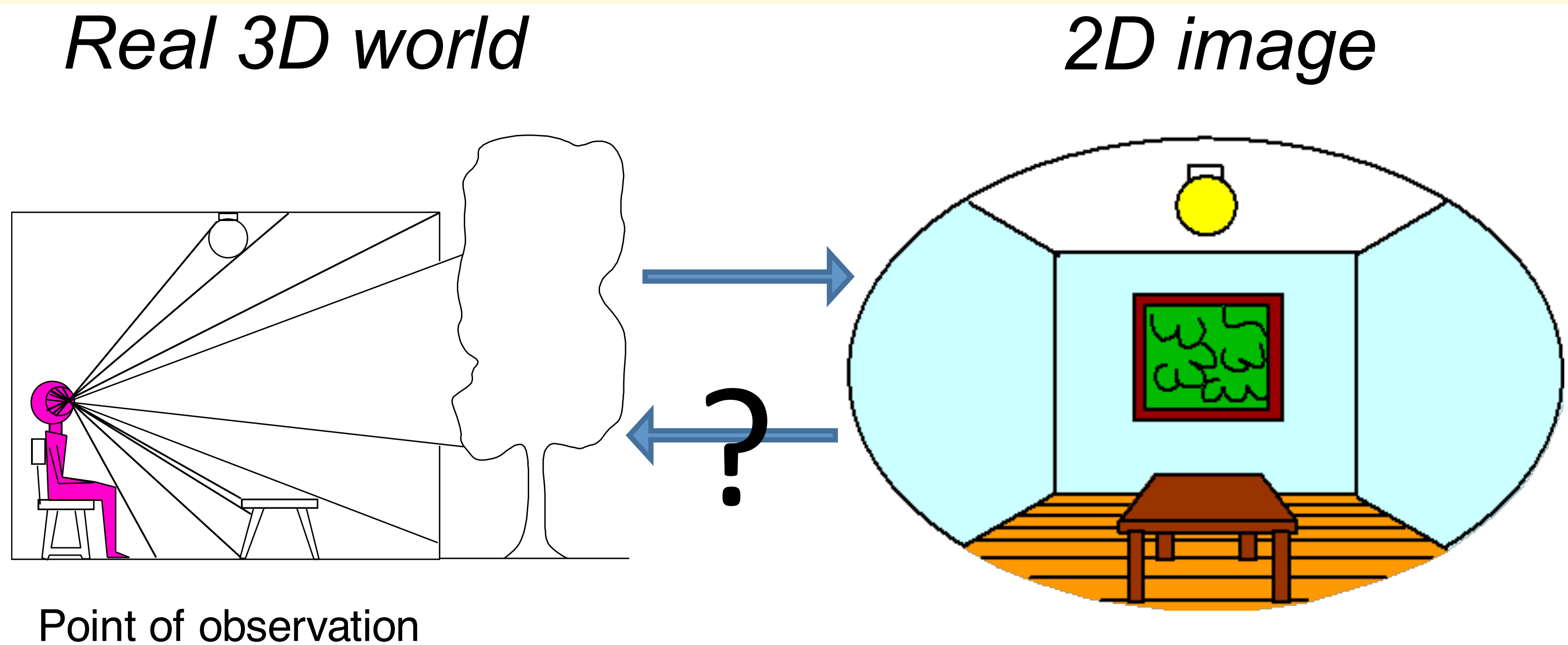
Today

- Stereo vision
- Epipolar geometry

…with a lot of slides stolen from Steve Seitz, Fei Fei Li, Alexei Efros

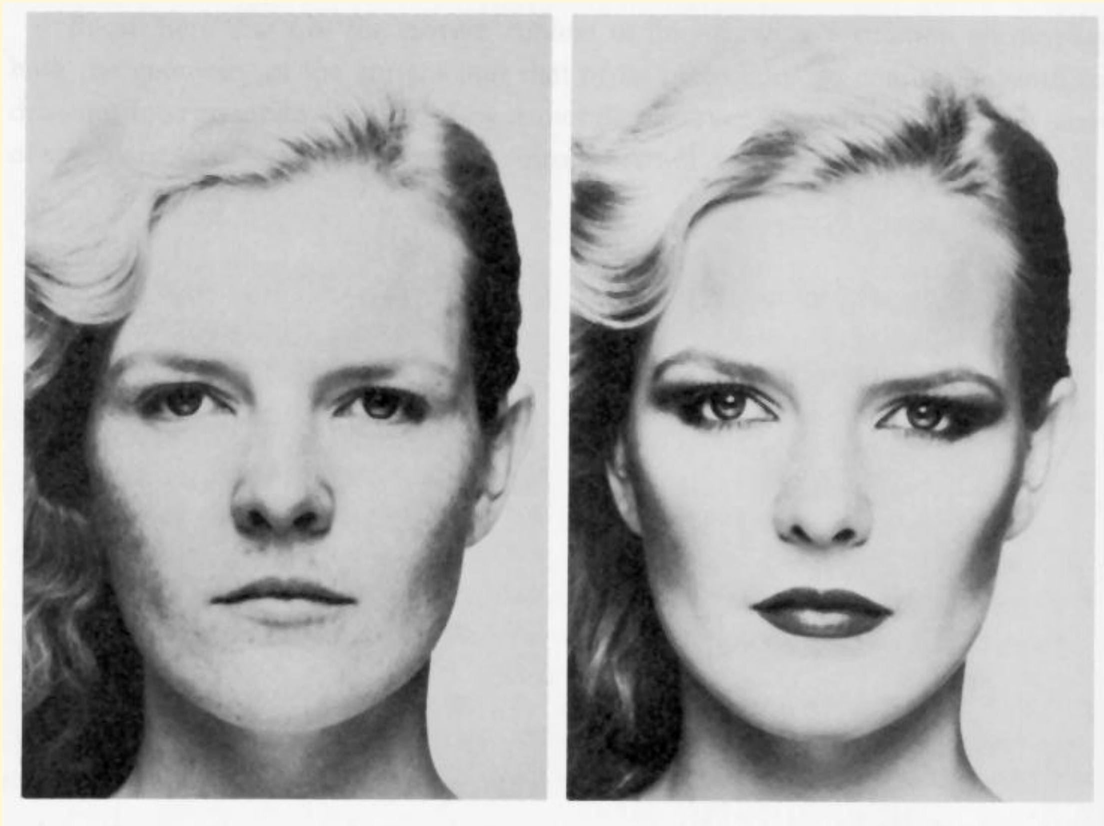
Recovering 3D from Images

- How can we automatically compute 3D geometry from images?
  - What cues in the image provide 3D information?



Visual Cues for 3D

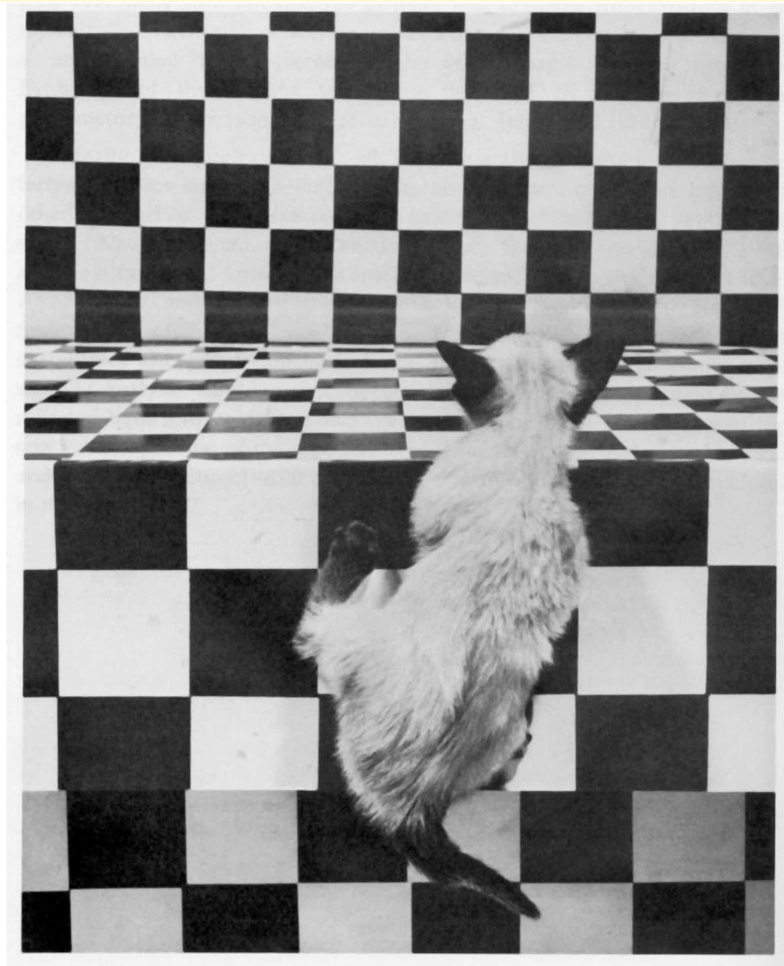
- Shading



Merle Norman Cosmetics, Los Angeles

Visual Cues for 3D

- Texture



The Visual Cliff, by William Vandivert, 1960

Visual Cues for 3D

- Focus

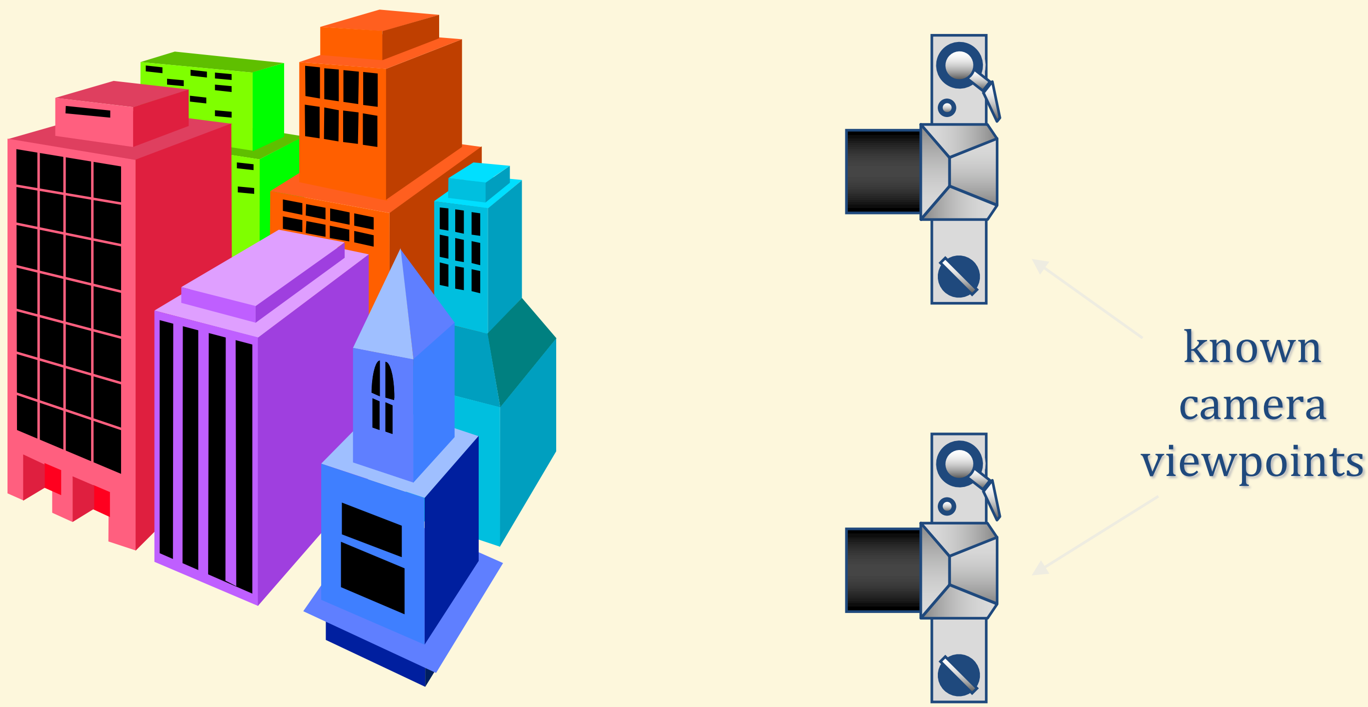
From *The Art of Photography*, Canon

Visual Cues for 3D

- Motion

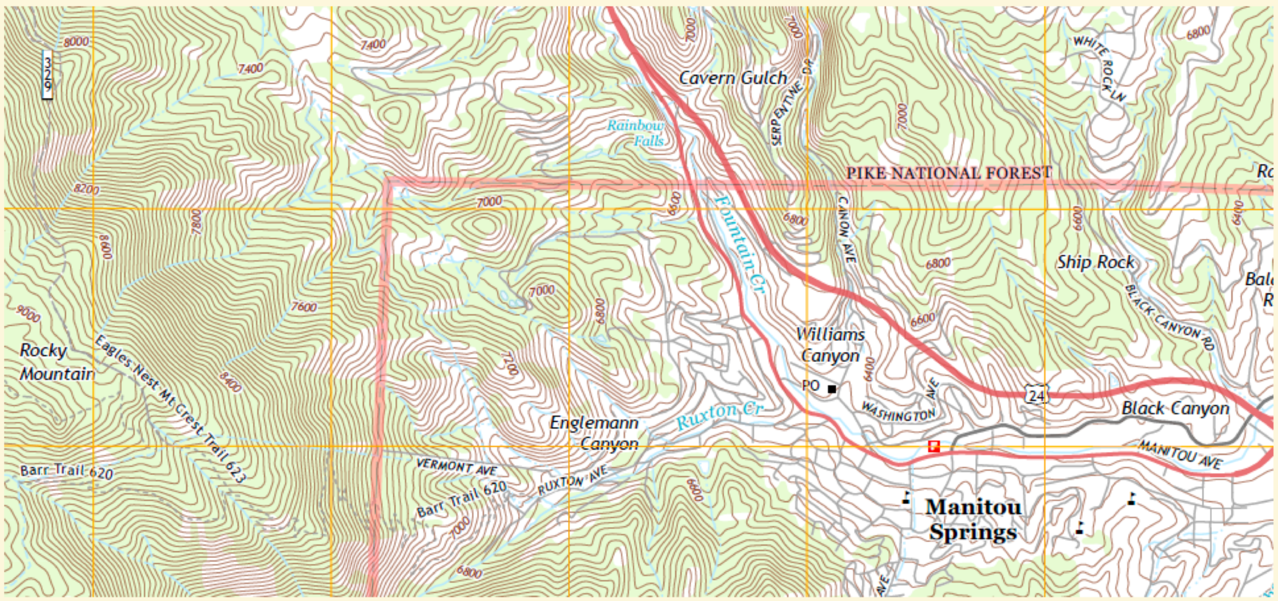
Stereo Reconstruction

- The Stereo Problem
  - Shape from two (or more) images
  - Biological motivation



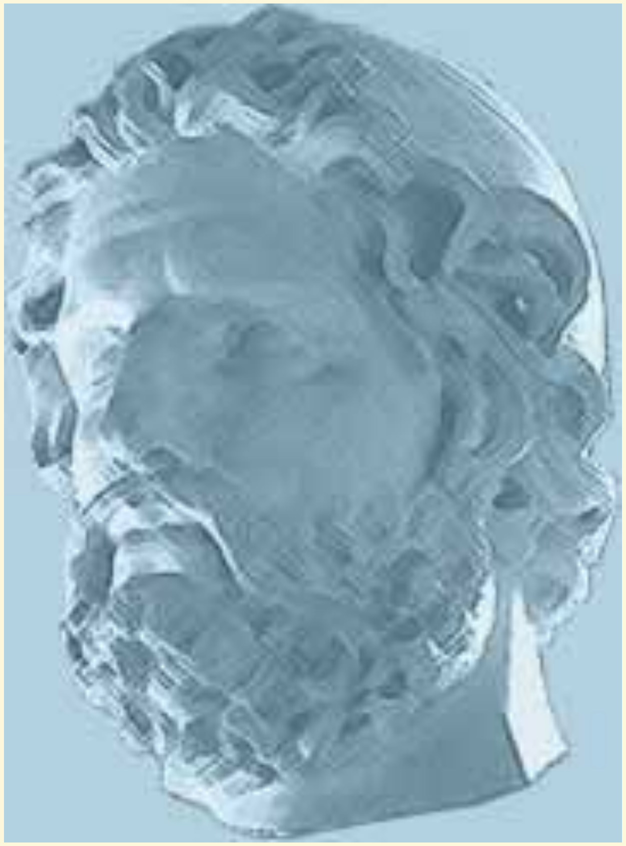
Stereo Reconstruction

- The Stereo Problem
  - Shape from two (or more) images
  - Biological motivation



known camera viewpoints

Why do we have two eyes?



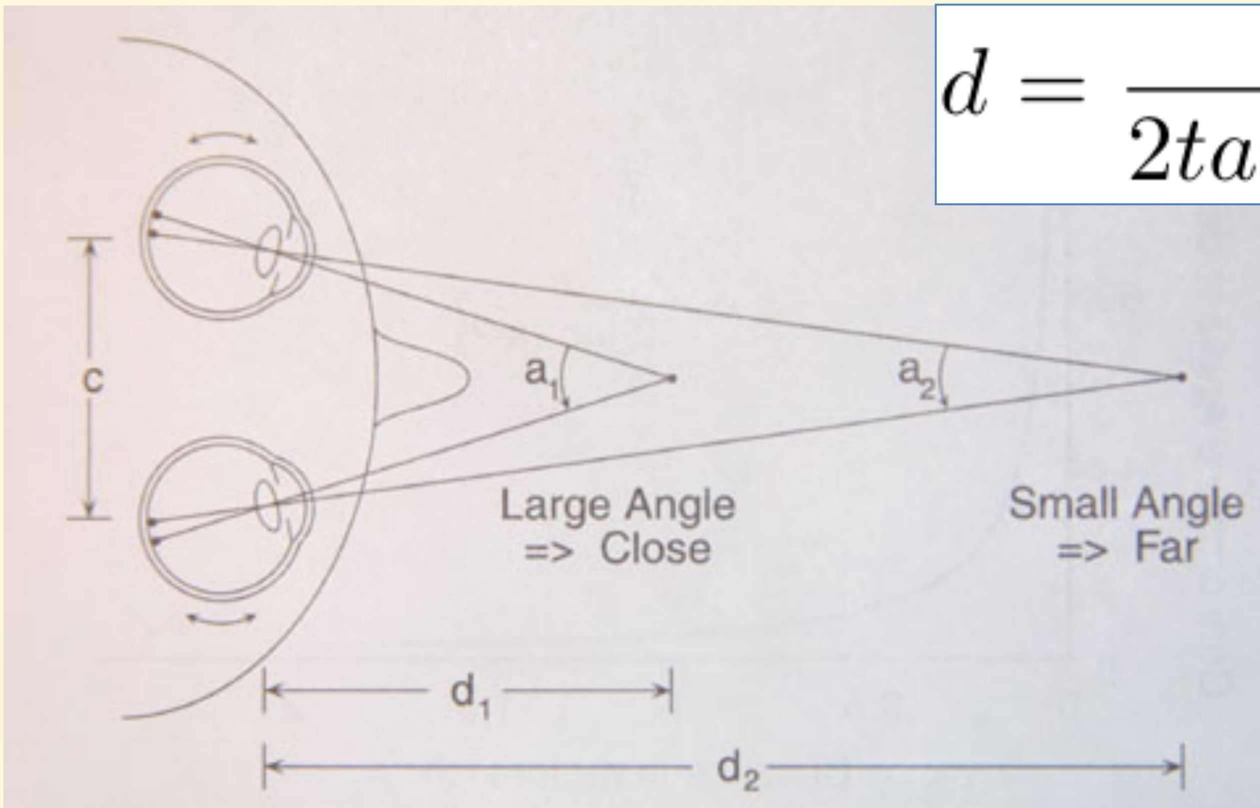
Cyclope **vs.** **Odysseus**

1. Two is better than one

"Just checking."

2. Depth from Convergence

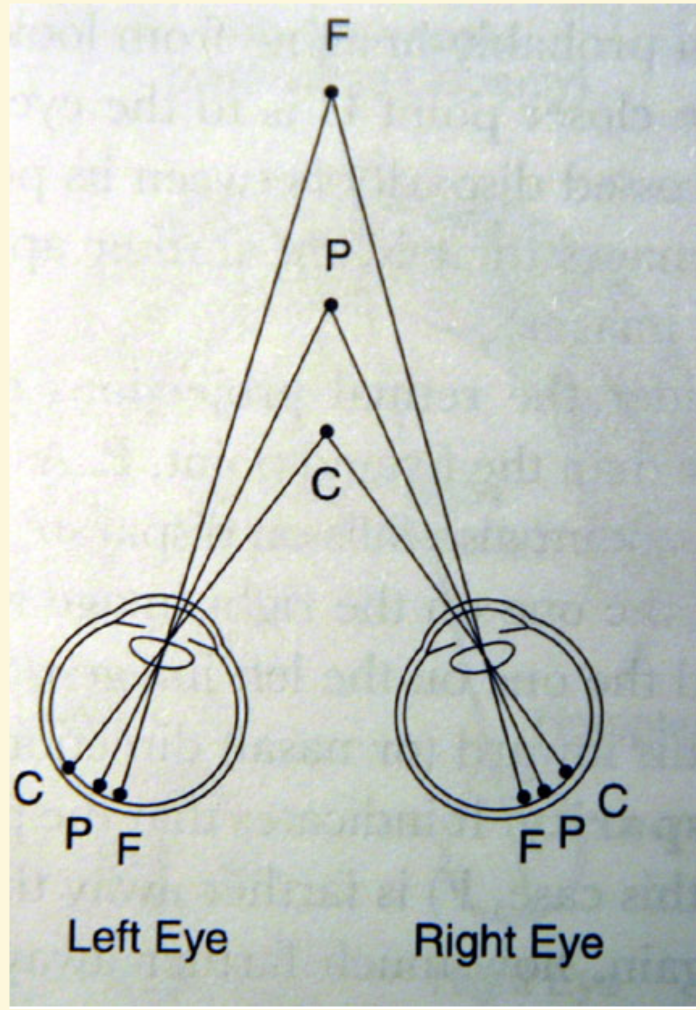
$$d = \frac{c}{2tan(a/2)}$$



Human performance: up to 6-8 feet

3. Depth from Binocular Disparity

- *P: converging point*
- *C: object nearer projects to the outside of the P, disparity = +*
- *F: object farther projects to the inside of the P, disparity = -*

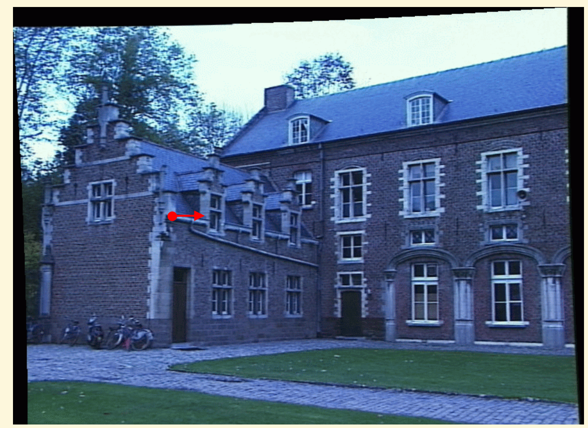


Human vs Computer Stereopsis

- Fusing the pictures recorded by our two eyes and exploiting the difference (or disparity) between them allows us to gain a strong sense of depth.
- For a human, the eyes change their angle according to the distance to the observed object. To a computer this represents significant extra complexity in the geometrical calculations.
- This chapter is concerned with the design and implementation of algorithms that mimic our ability to perform this task, known as **stereopsis**.

Fusing the pictures recorded by our two eyes and exploiting the difference (or disparity) between them allows us to gain a strong sense of depth

image $I(x,y)$ | Disparity map $D(x,y)$ | image $I'(x',y')$



$$(x',y')=(x+D(x,y),\ y)$$

So if we could find **corresponding points** in two images, could we **estimate** **<u>relative</u>** **depth?**

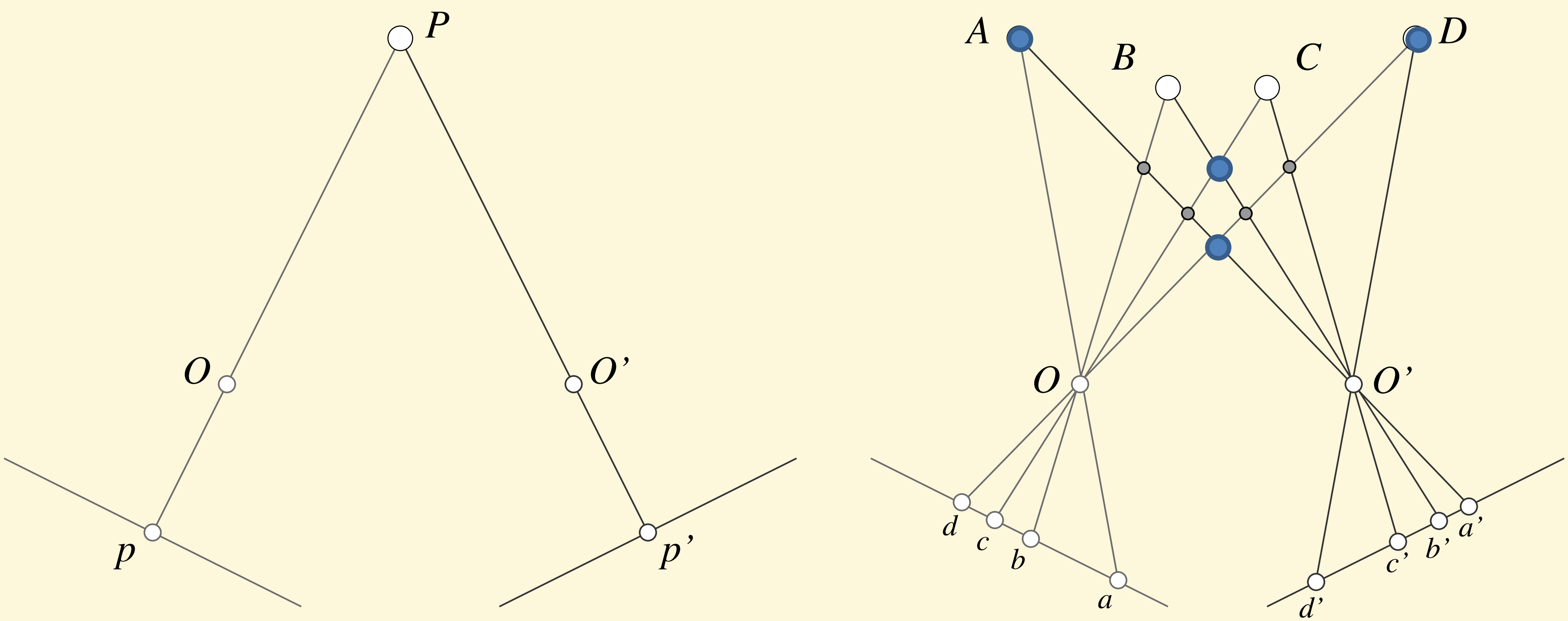
Stereopsis - applications

- visual robot navigation
- cartography
- aerial reconnaissance
- image segmentation for object recognition
- constructing three-dimensional scene models for computer graphics applications

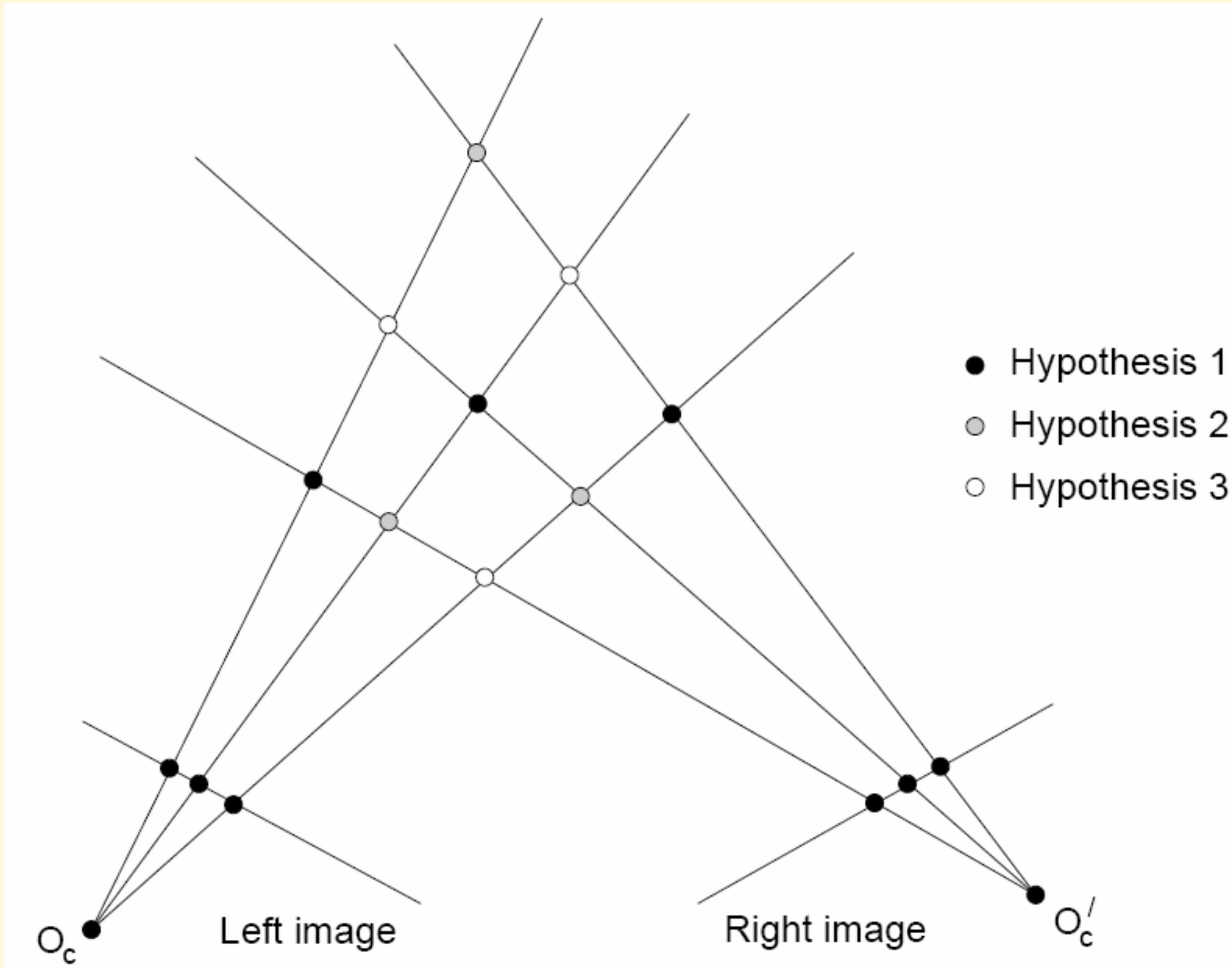
Stereopsis – two parts

1. The **correspondence problem** = fusion of features observed by two(or more) eyes
   - establishing correct correspondences is hard
2. The **reconstruction problem** = computing the three-dimensional preimage.
   - pretty straight-forward, for just one feature
   - harder for multiple features

Stereopsis – binocular fusion problem



Stereopsis – solving the correspondence problem



Multiple matching hypotheses, but which one is correct?

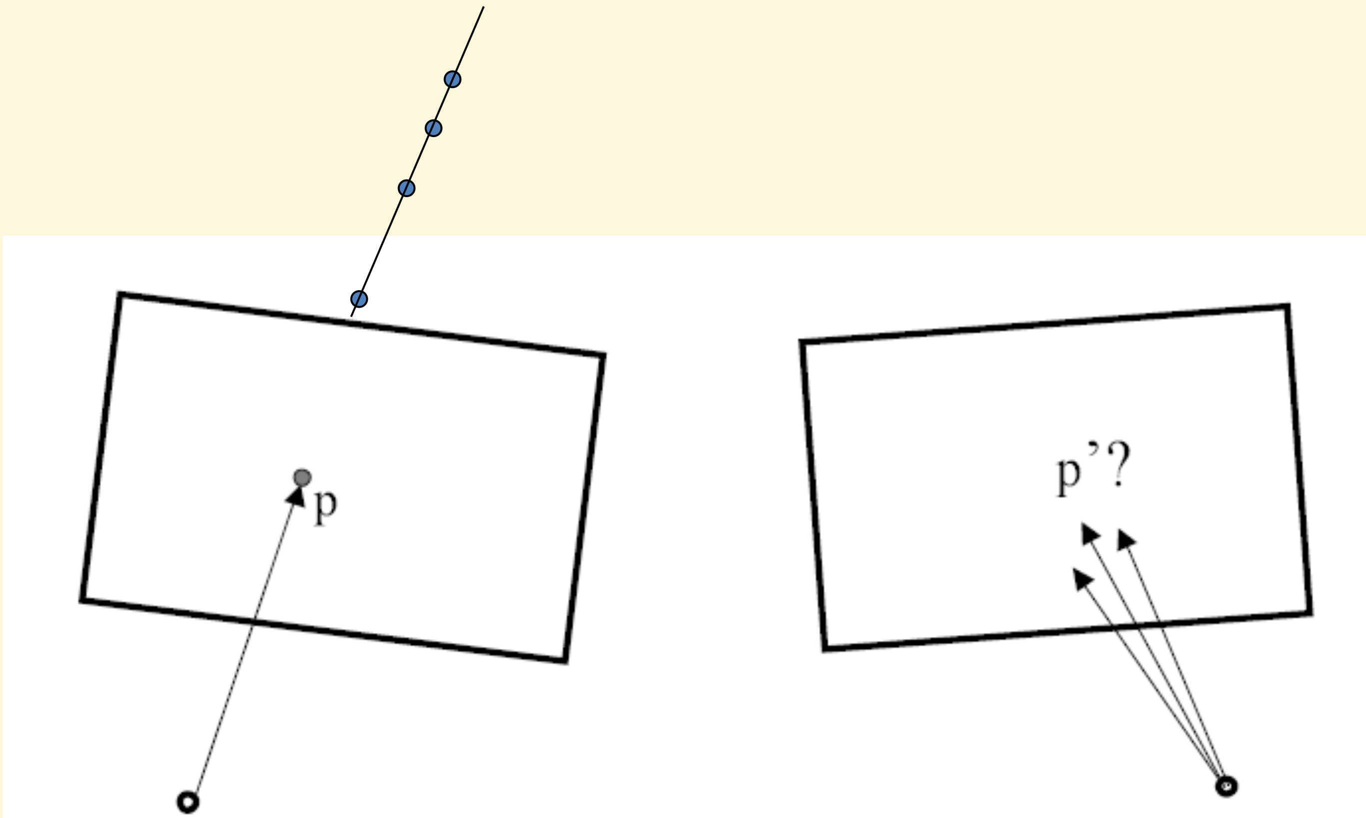
Why is the correspondence problem difficult?

- Some points in each image will have no corresponding point in the other image, because:

- The cameras may have different fields of view
- Due to occlusion
- A stereo system must be able to determine the image parts that should not be matched.

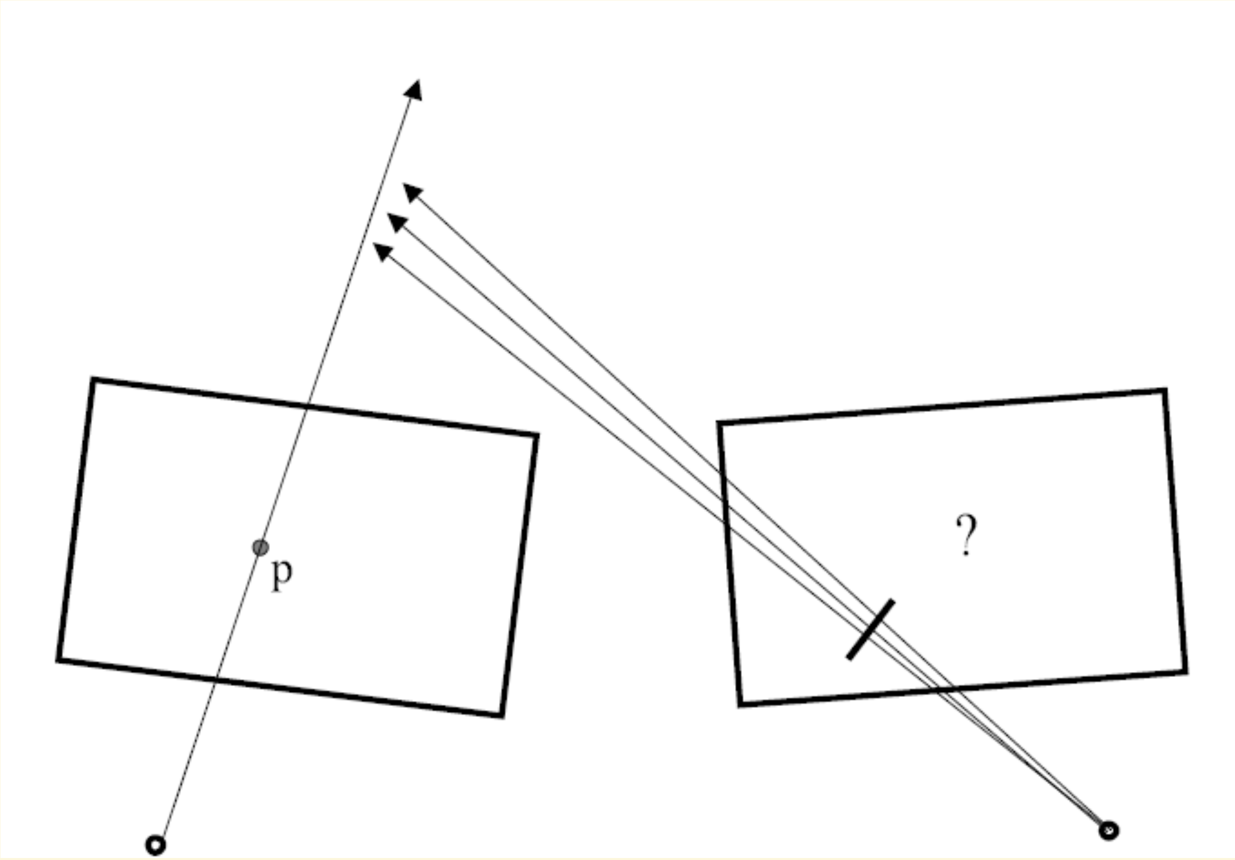
- In the above picture, the part with green and red are the parts that show the different viewpoint of the cameras
- The task is to find points, that can be seen for both cameras
- Occlusion is both visible at the right edge of the box

Stereo correspondence constraints

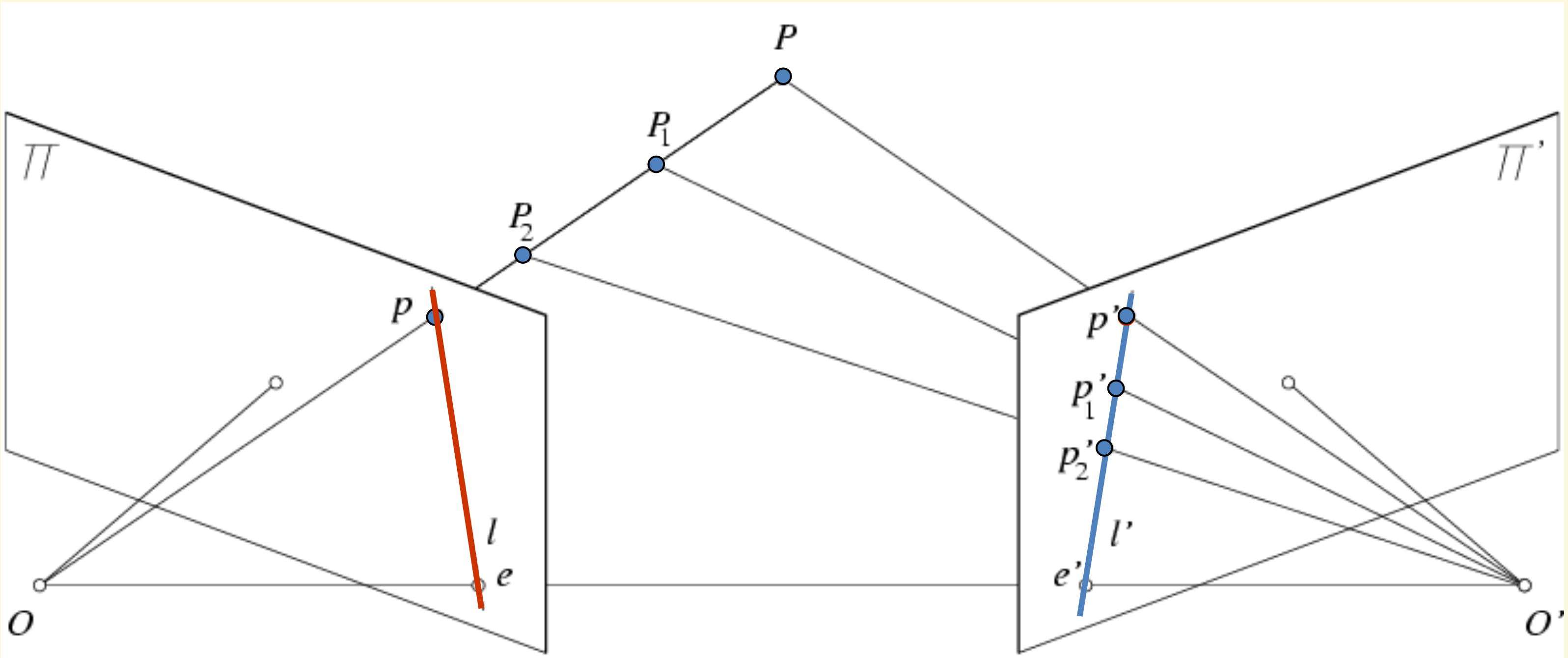


- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



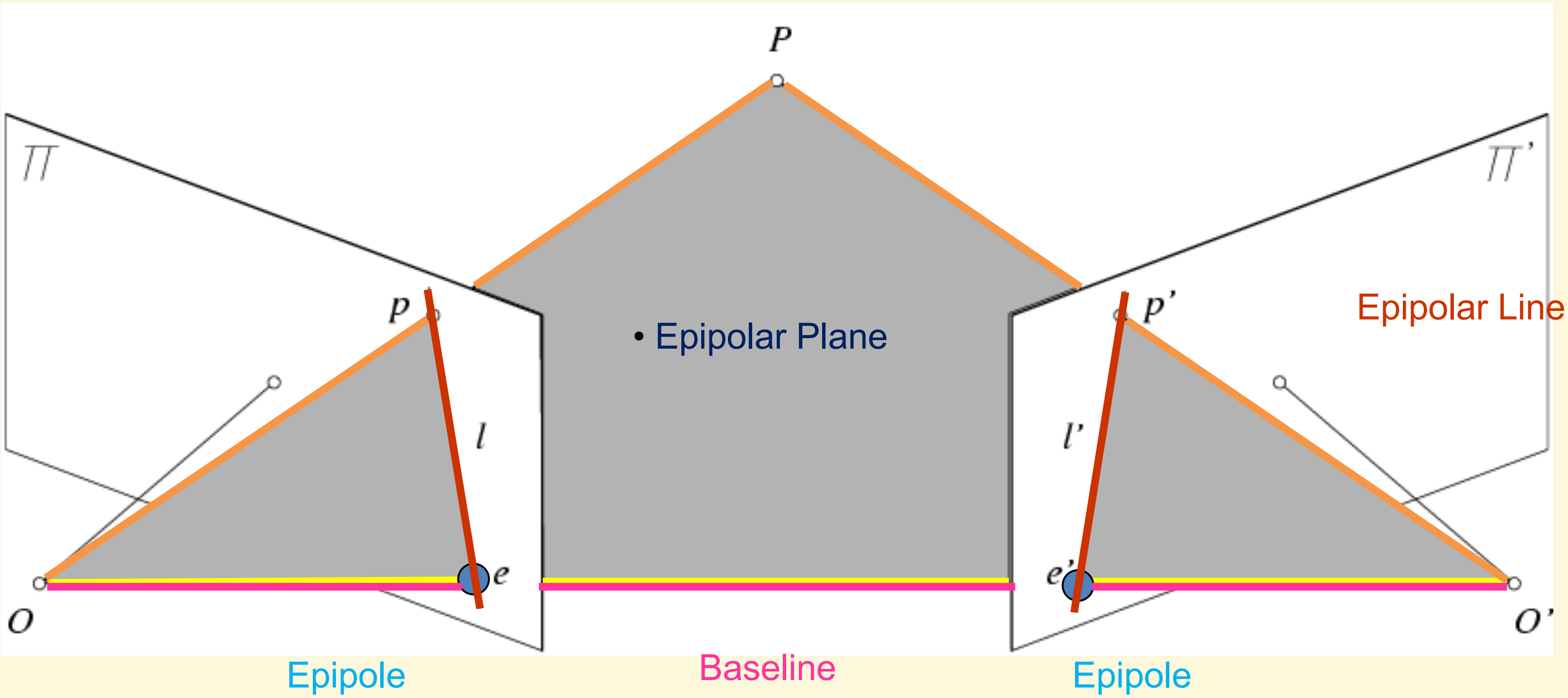
Epipolar constraint



Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view.

- It must be on the line carved out by a plane connecting the world point and optical centers.

Epipolar geometry



Epipolar geometry: terms

- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with image plane
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane

- All epipolar lines intersect at the epipole
- An epipolar plane intersects the left and right image planes in epipolar lines

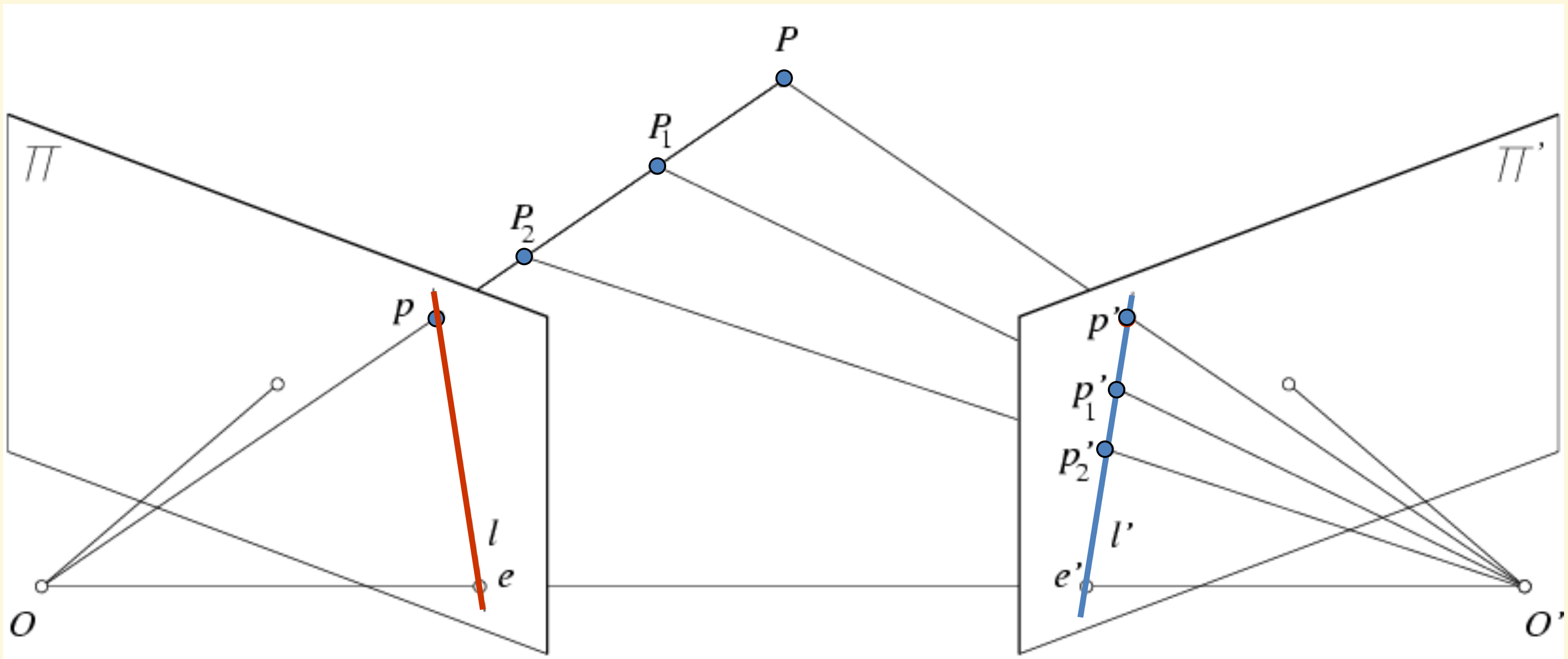
Why is the epipolar constraint useful?

Epipolar constraint

This is useful because it reduces the correspondence problem to a 1D search along an epipolar line.

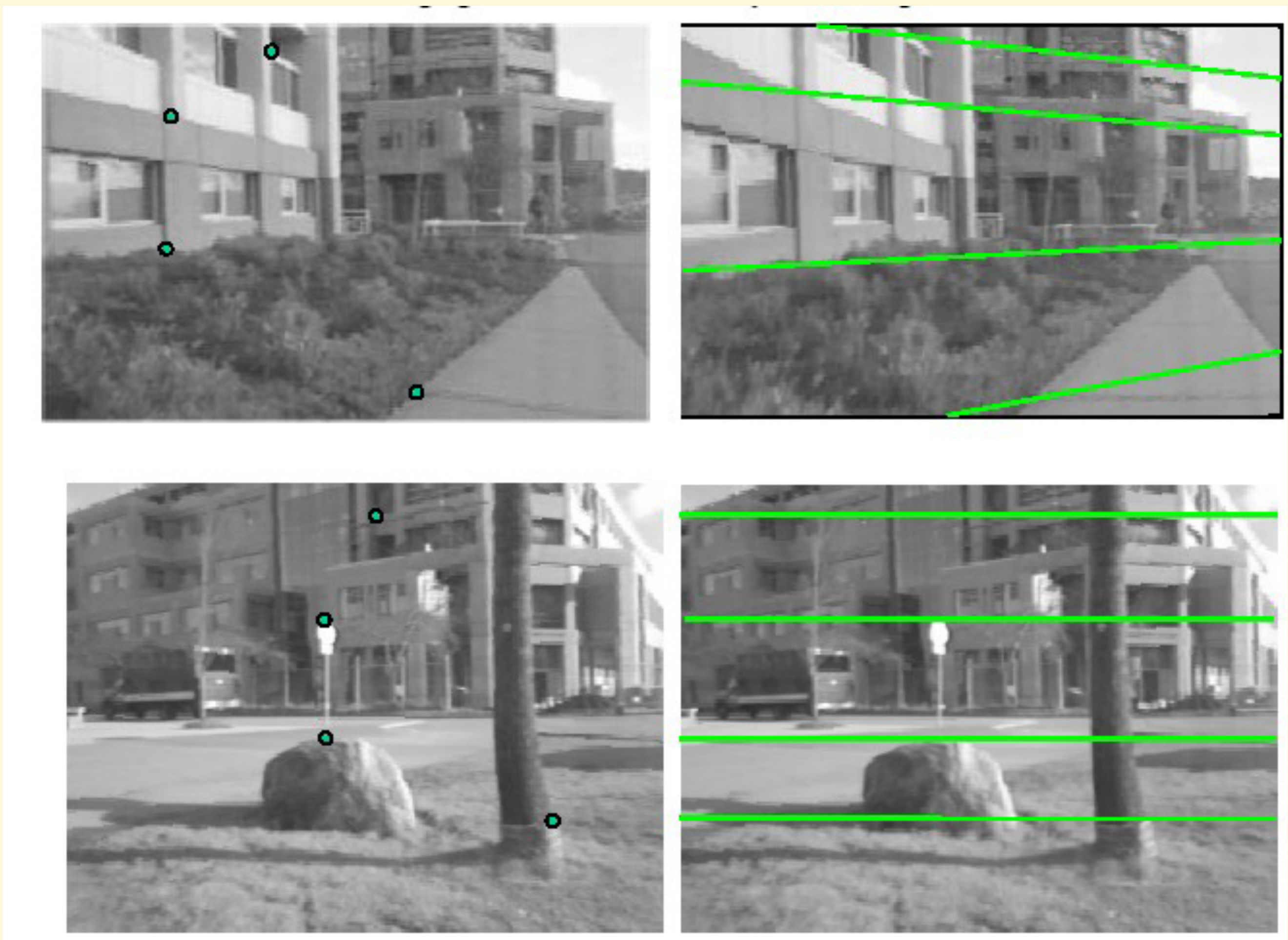
Image from Andrew Zisserman

Epipolar constraint



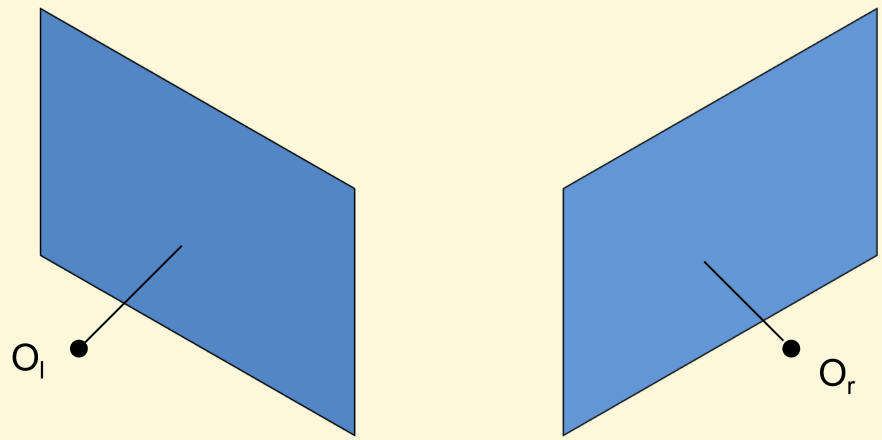
Each camera may be internally calibrated, but the rigid transformation separating the two camera coordinate systems is unknown. In this case, the epipolar geometry constrains the set of possible motions.

What do the epipolar lines look like?

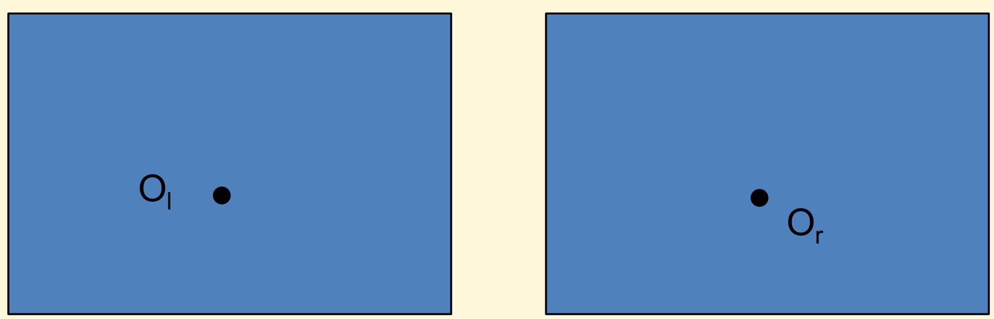


What do the epipolar lines look like?

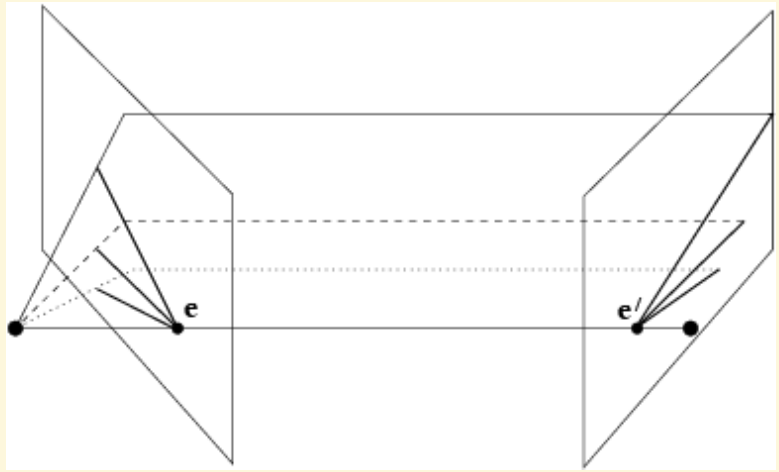
1.

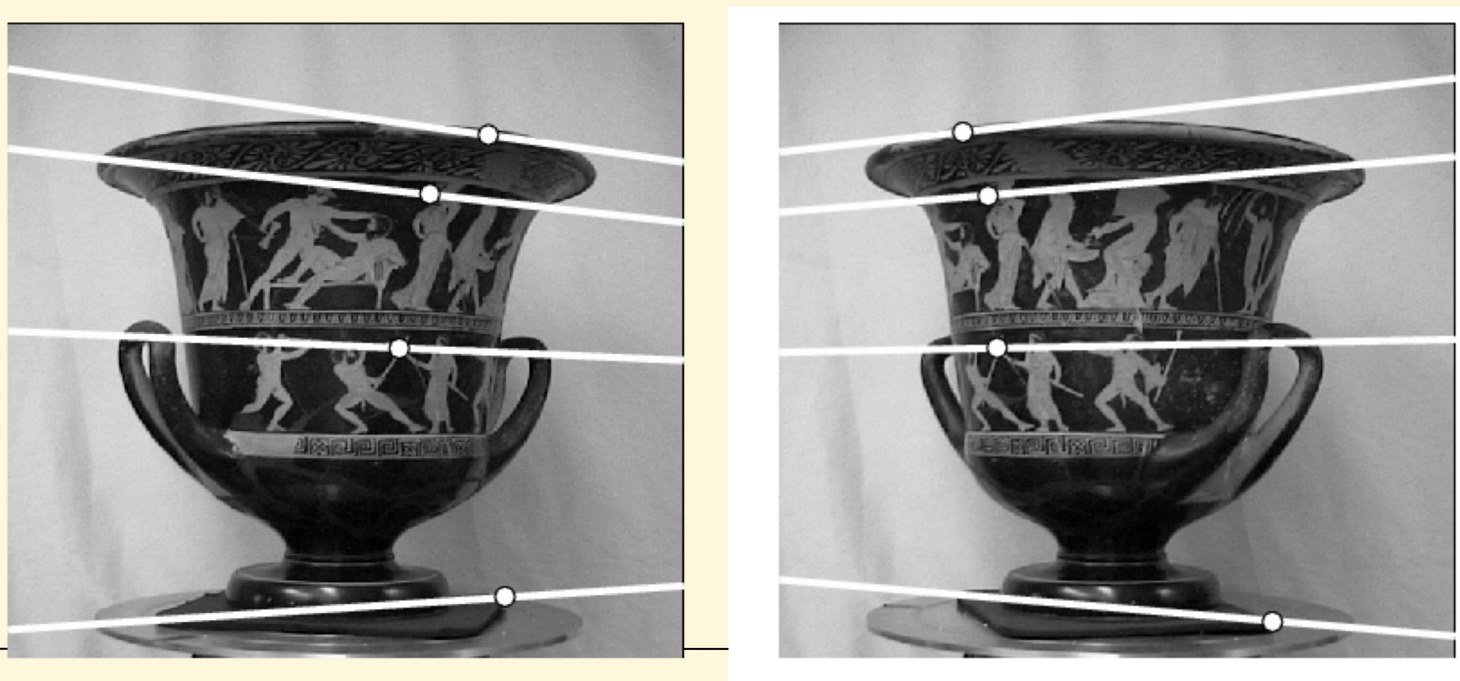


2.



Example: converging cameras

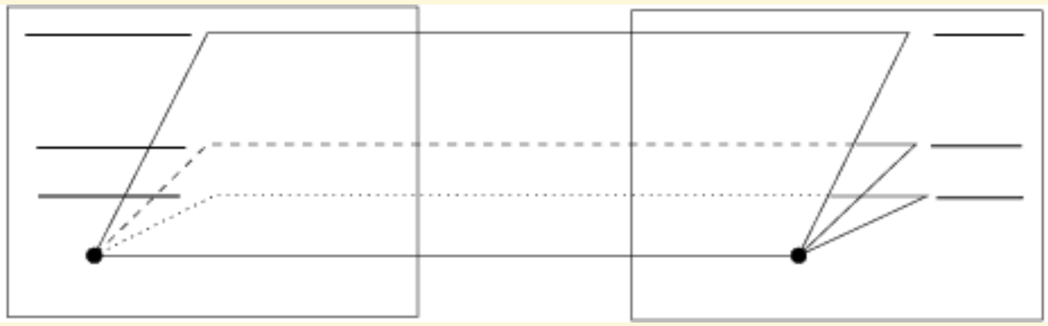




University of Colorado **Boulder**

Figure from Hartley & Zisserman

Example: parallel cameras



Where are the epipoles?

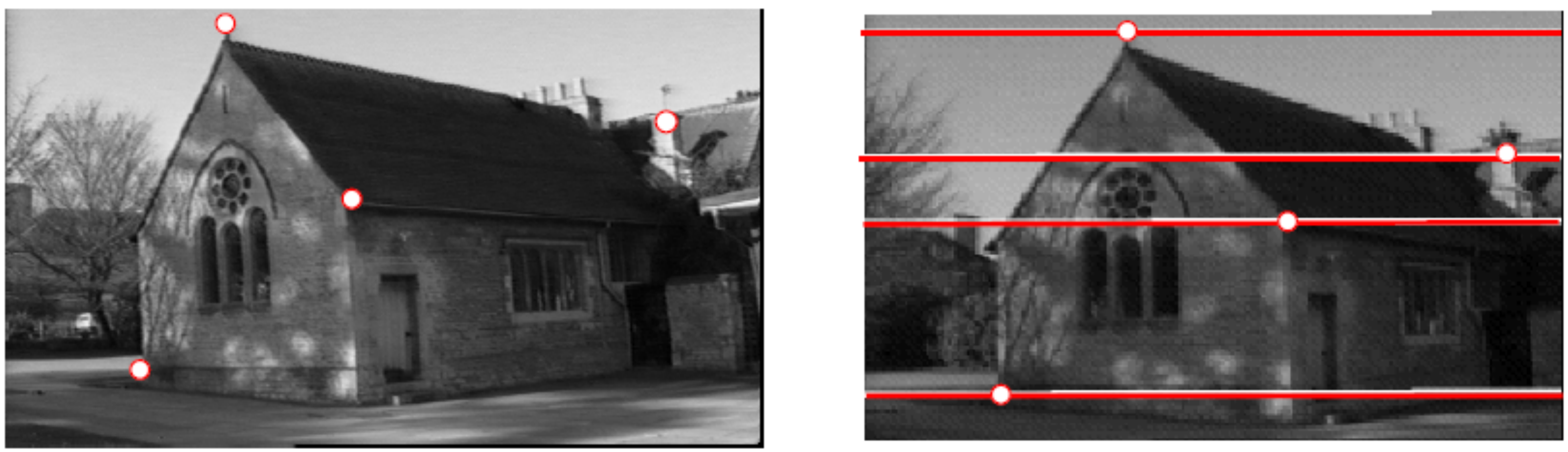
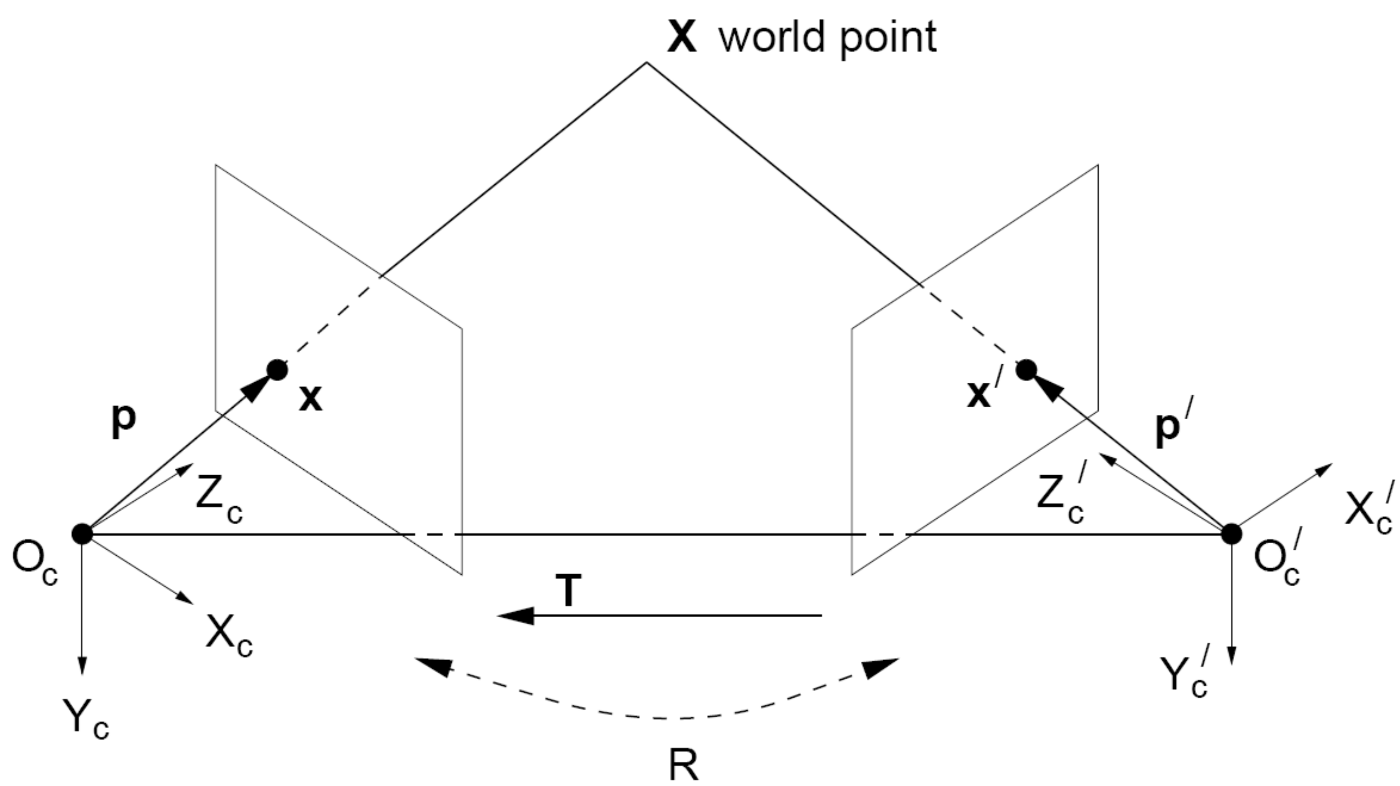


Figure from Hartley & Zisserman

- So far, we have the explanation in terms of geometry.
- Now, how to express the epipolar constraints algebraically?

Stereo geometry, with calibrated cameras

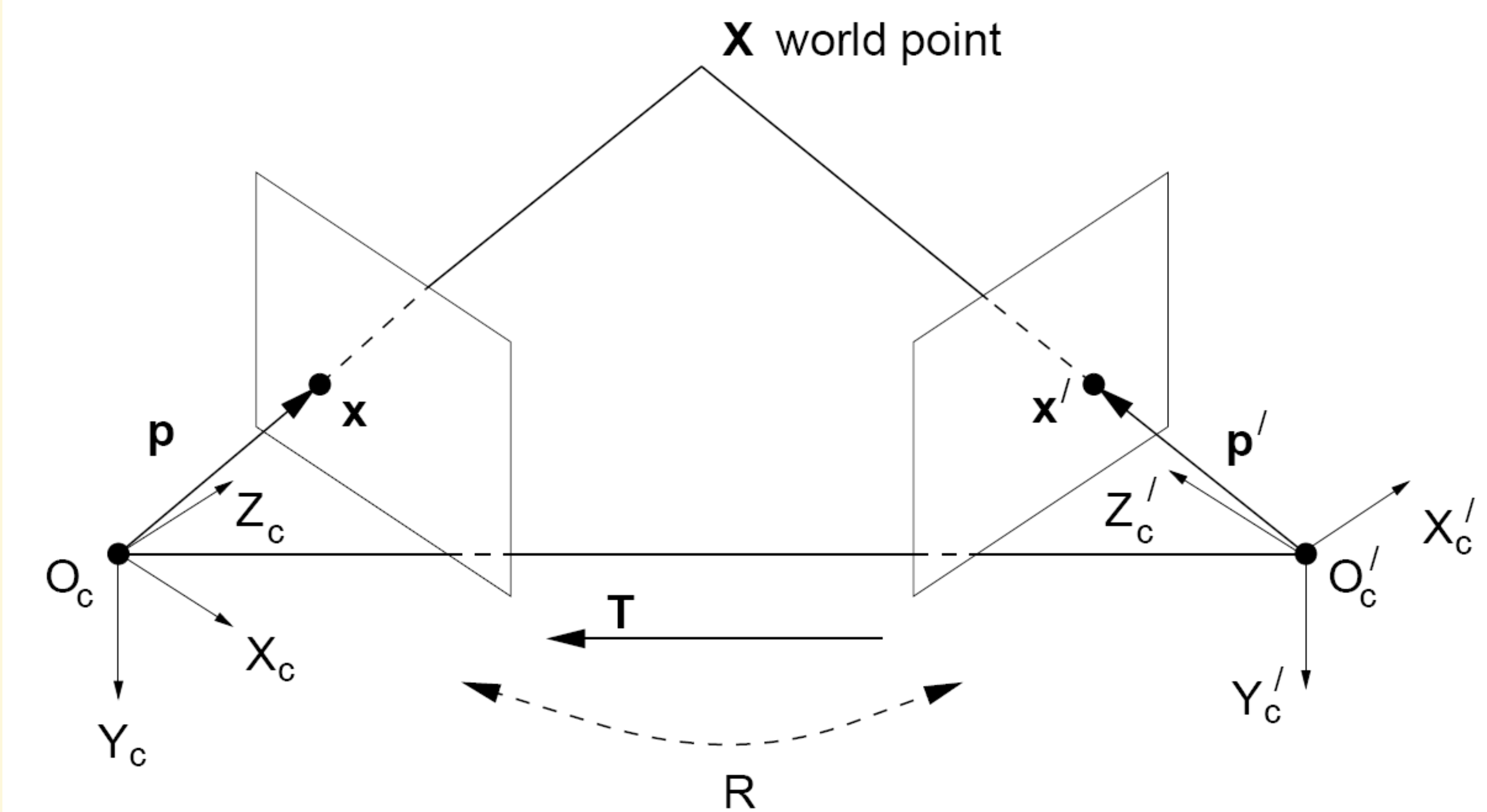


If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix **R**; translation: 3 vector **T**.

Stereo geometry, with calibrated cameras



If the stereo rig is calibrated, we know :
how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

$$\mathbf{X'}_c = \mathbf{R}\mathbf{X}_c + \mathbf{T}$$

An aside: cross product

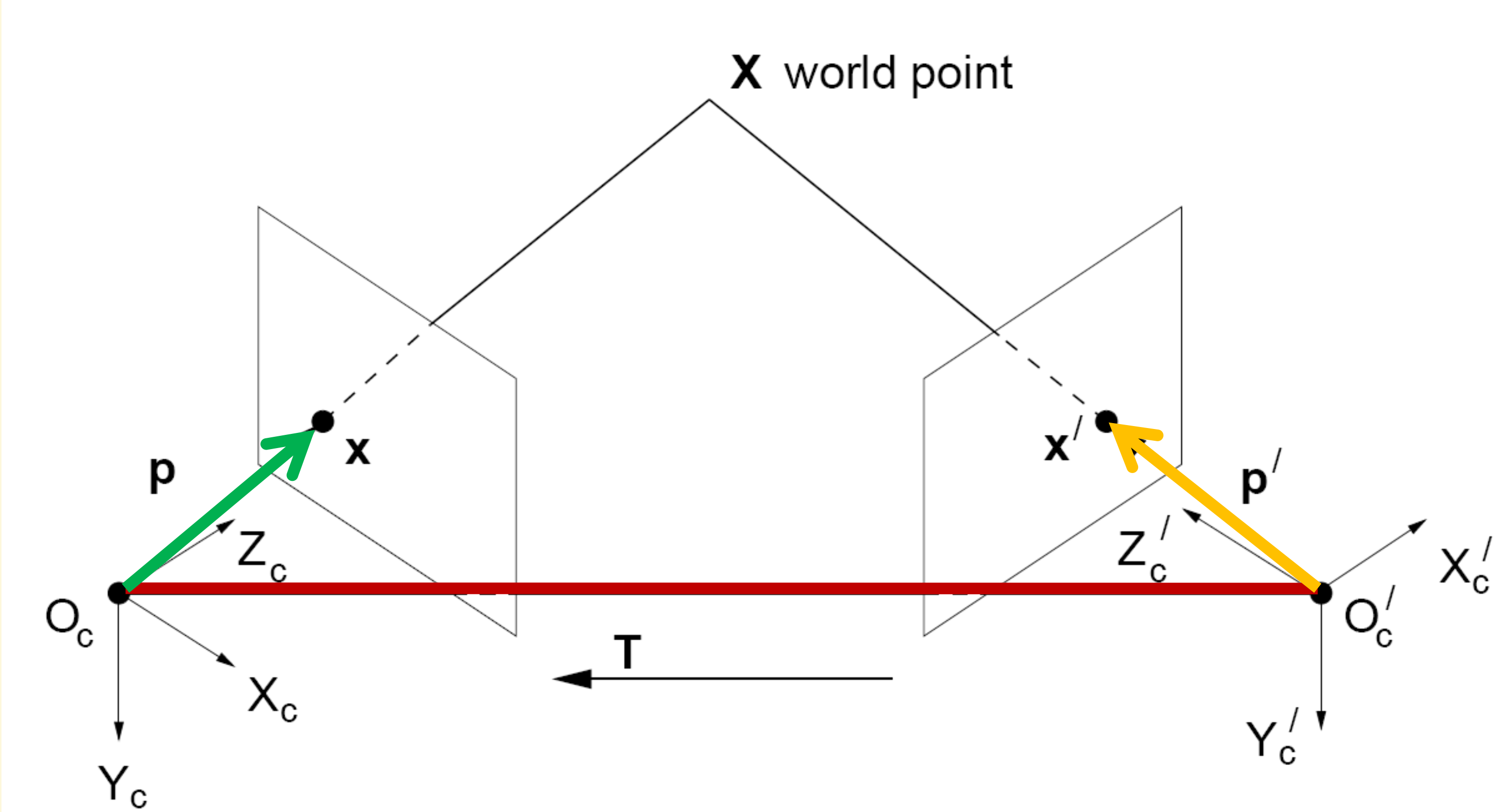
$$\vec{a} \times \vec{b} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

So here, c is perpendicular to both a and b, which means the dot products will be = 0.

From geometry to algebra



$$\boxed{\mathbf{X}'} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$
$$= 0$$

Cross product as matrix multiplication

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

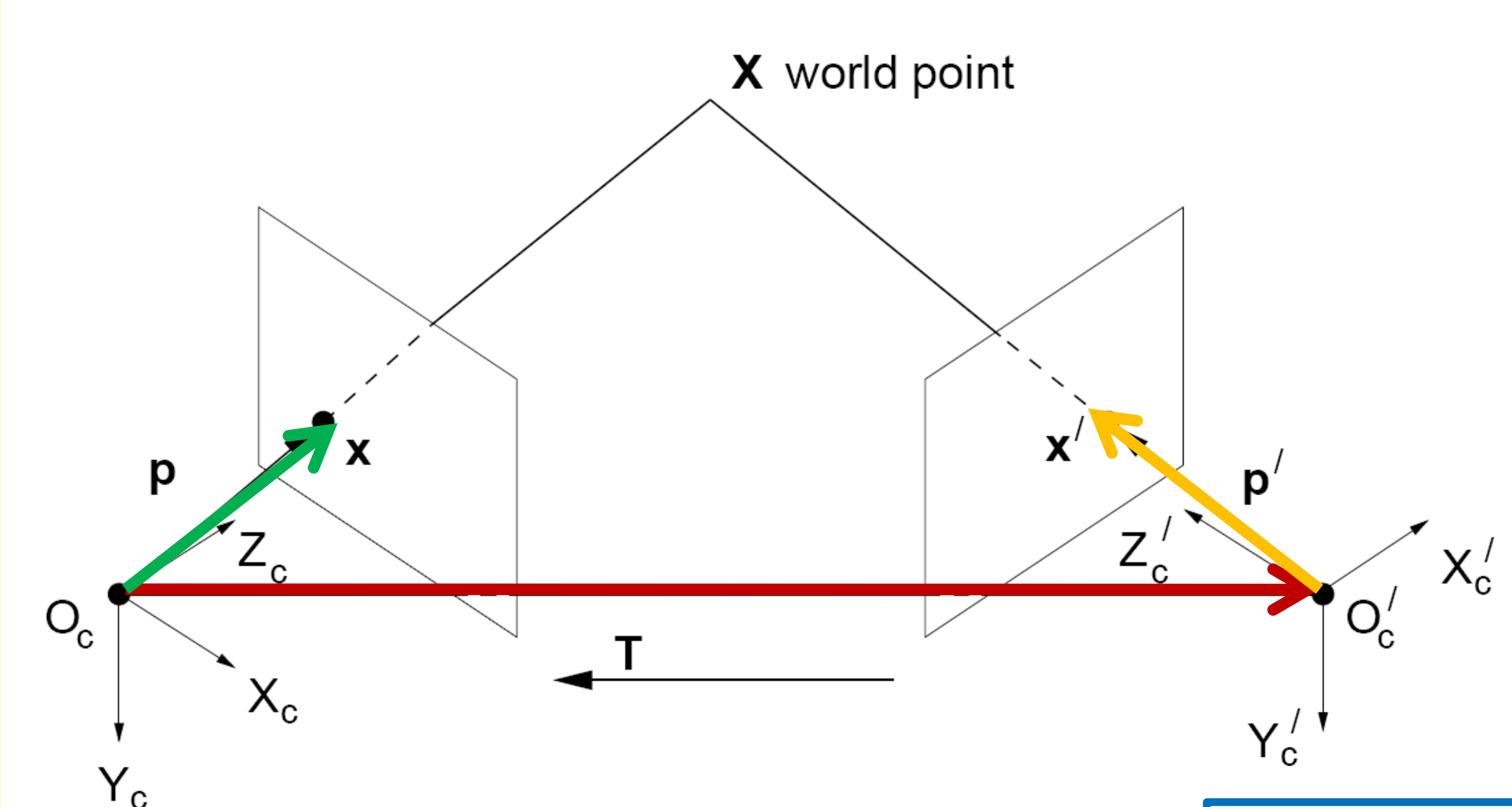
Can be expressed as a matrix multiplication.

$$[a_x] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

$$\boxed{\vec{a} \times \vec{b} = [a_x]\vec{b}}$$

a_x is a skew symmetric matrix.

From geometry to algebra



$$\mathbf{X'} = \mathbf{RX} + \mathbf{T}$$

$$\underbrace{\mathbf{T}\times\mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T}\times\mathbf{RX} + \mathbf{T}\times\mathbf{T}$$

$$= \mathbf{T}\times\mathbf{RX}$$

$$\mathbf{X}'\cdot\left(\mathbf{T}\times\mathbf{X}'\right) = \mathbf{X}'\cdot\left(\mathbf{T}\times\mathbf{RX}\right) = 0$$

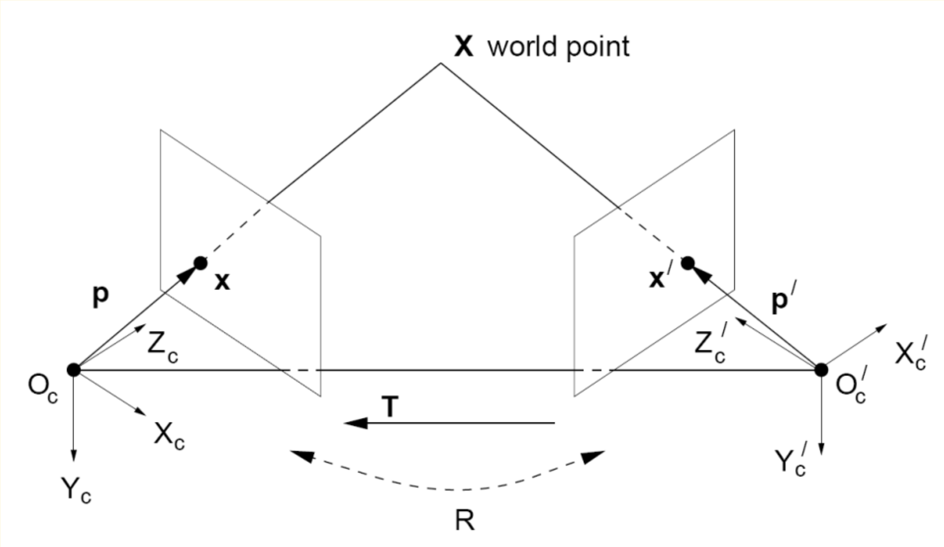
Essential matrix

$$\boxed{\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0}$$

$$\mathbf{X}' \cdot ([\mathrm{T}_x]\mathbf{R}\mathbf{X}) = 0$$

Let $\mathbf{E} = [\mathrm{T}_x]\mathbf{R}$

$$\mathbf{X}'^T \mathbf{E}\mathbf{X} = 0$$

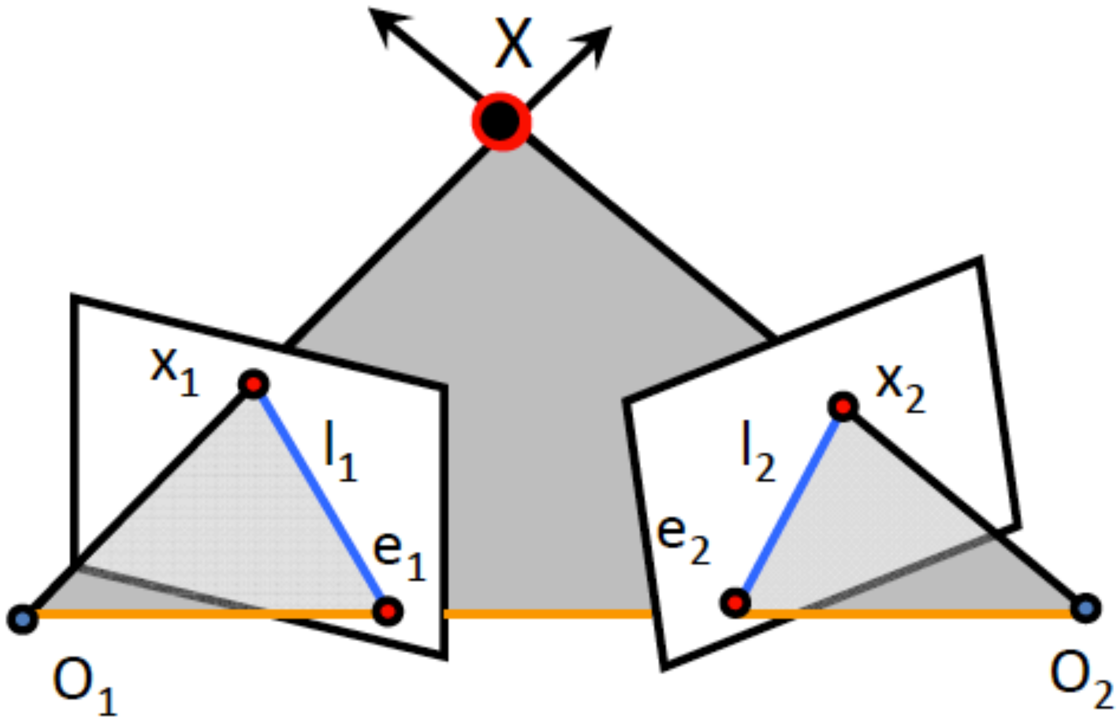


E is called the **essential matrix**, and it relates corresponding image points between both cameras, given the rotation and translation.

If we observe a point in one image, its position in other image is constrained to lie on line defined by above.

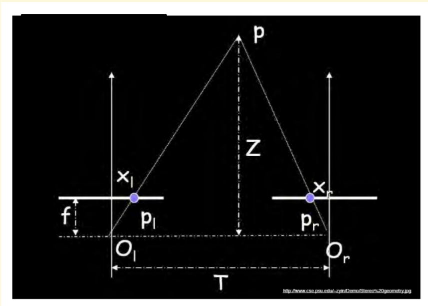
Note: these points are in **camera coordinate systems**.

Essential matrix



- $E\,x_2$ is the epipolar line associated with x_2 ($l_1 = E\,x_2$)
- $E^T x_1$ is the epipolar line associated with x_1 ($l_2 = E^T x_1$)
- E is singular (rank two)
- $E\,e_2 = 0$ and $E^T e_1 = 0$

Essential matrix example: parallel cameras



$$\mathbf{R} =$$

$$\mathbf{T} =$$

$$\mathbf{E} = [\mathrm{T}_{\mathbf{x}}]\mathbf{R} =$$

$$\mathbf{p} = [x, y, f]$$

$$\mathbf{p'} = [x', y', f]$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

image I(x,y)

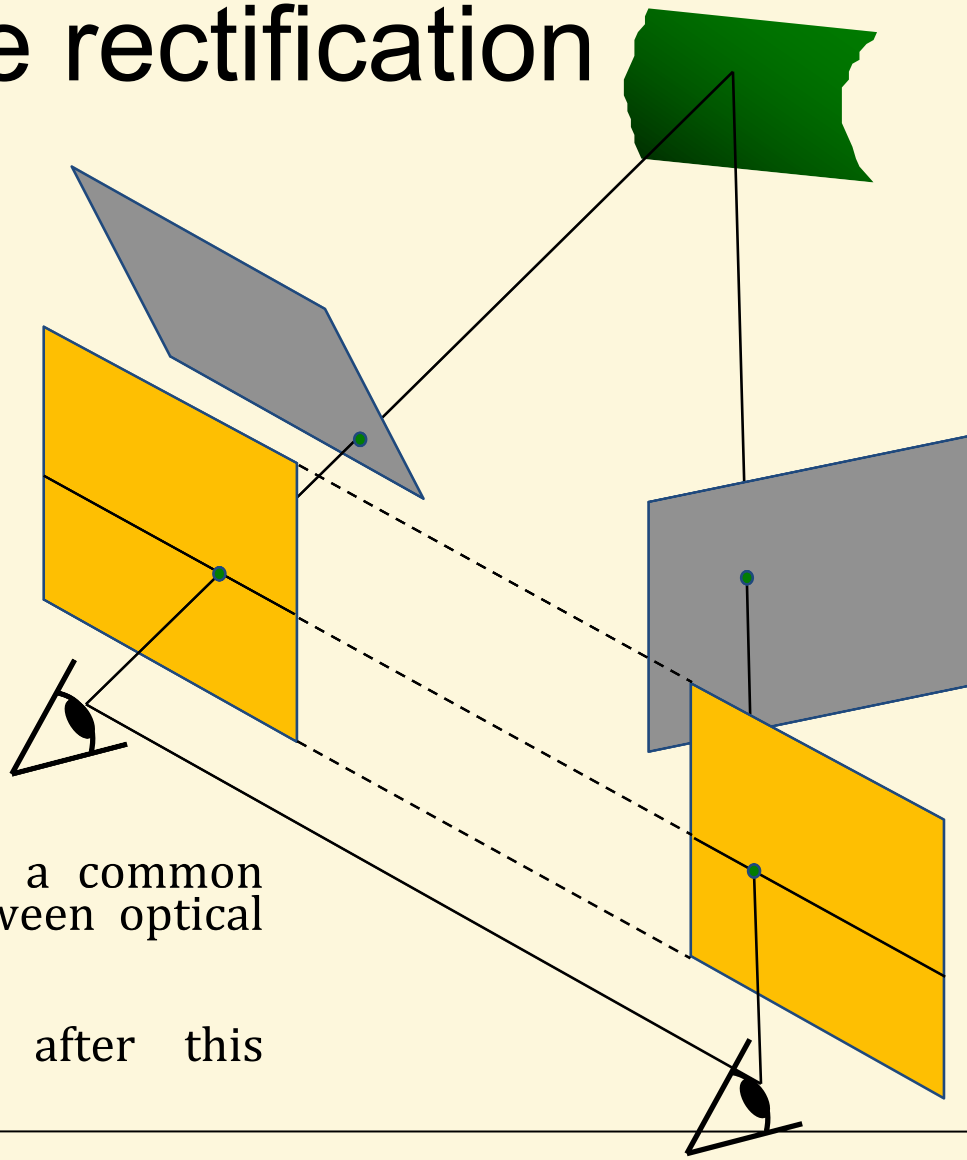
Disparity map D(x,y)

image I′(x′,y′)

$$(x',y')=(x+D(x,y),y)$$

What about when cameras' optical axes are not parallel?

Stereo image rectification



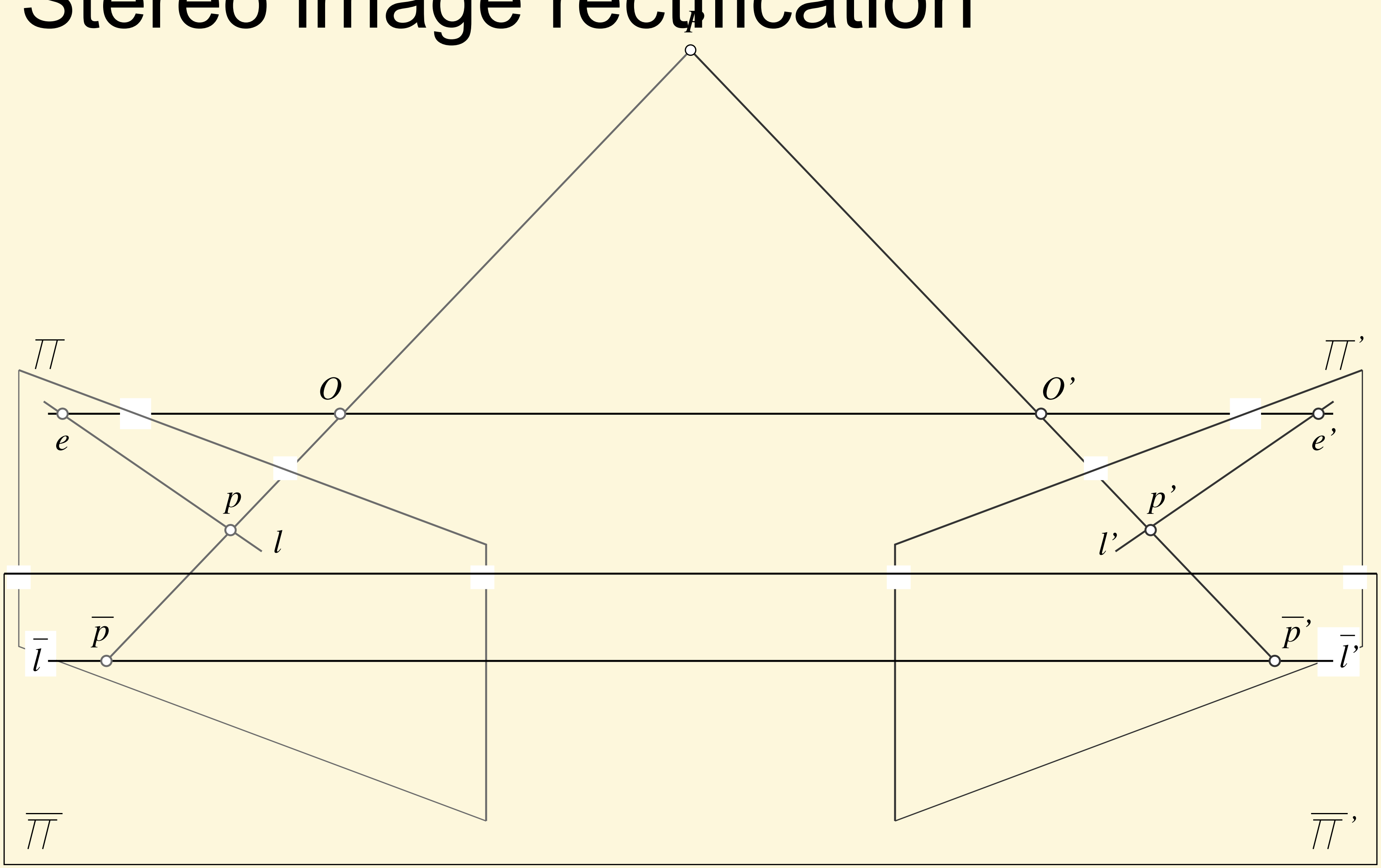
In practice, it is convenient if image scanlines (rows) are the epipolar lines.

Re-project image planes onto a common plane parallel to the line between optical centers.

Pixel motion is horizontal after this transformation

Slide credit: Li Zhang

Stereo image rectification



Stereo image rectification: example

Source: Alyosha Efros