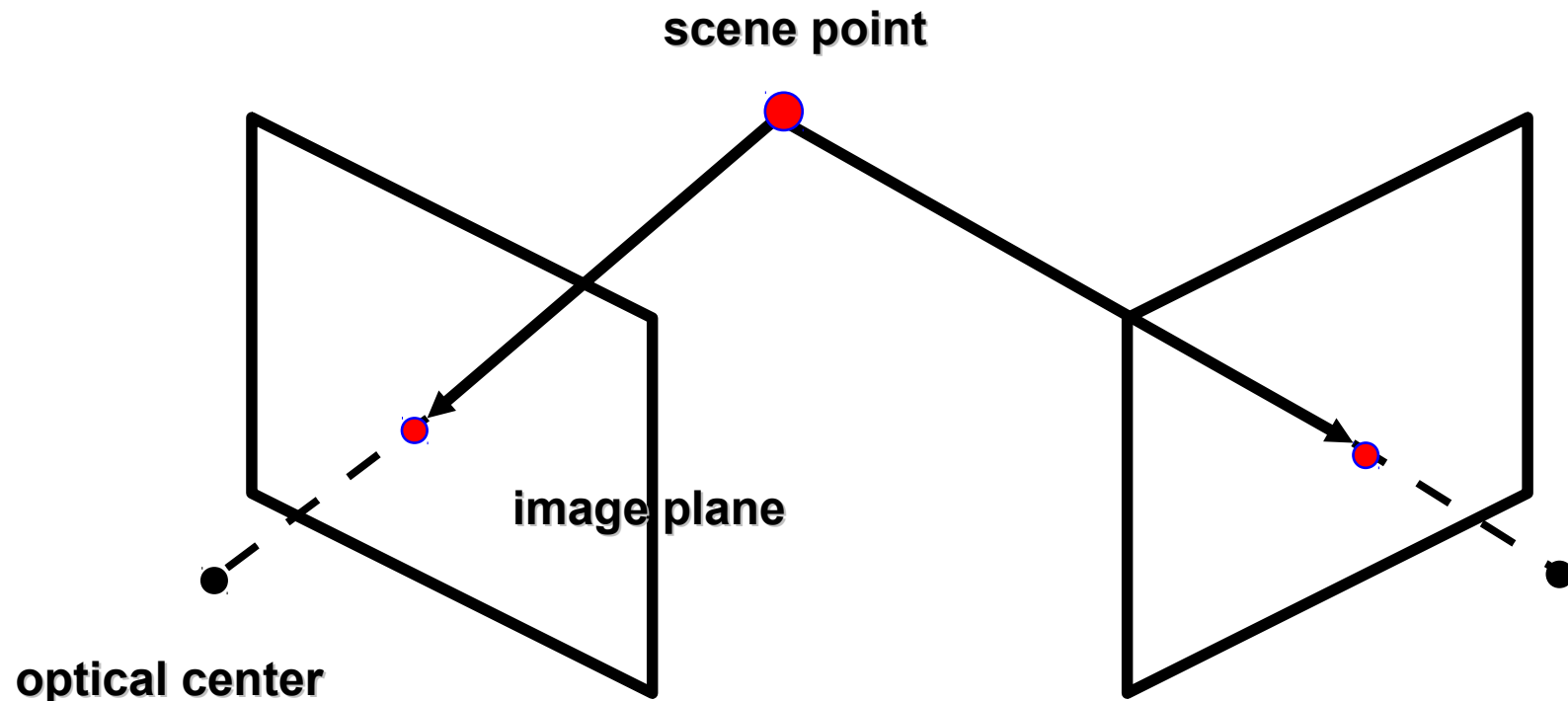


Lecture 14: Basic Multi-View Geometry

Stereo

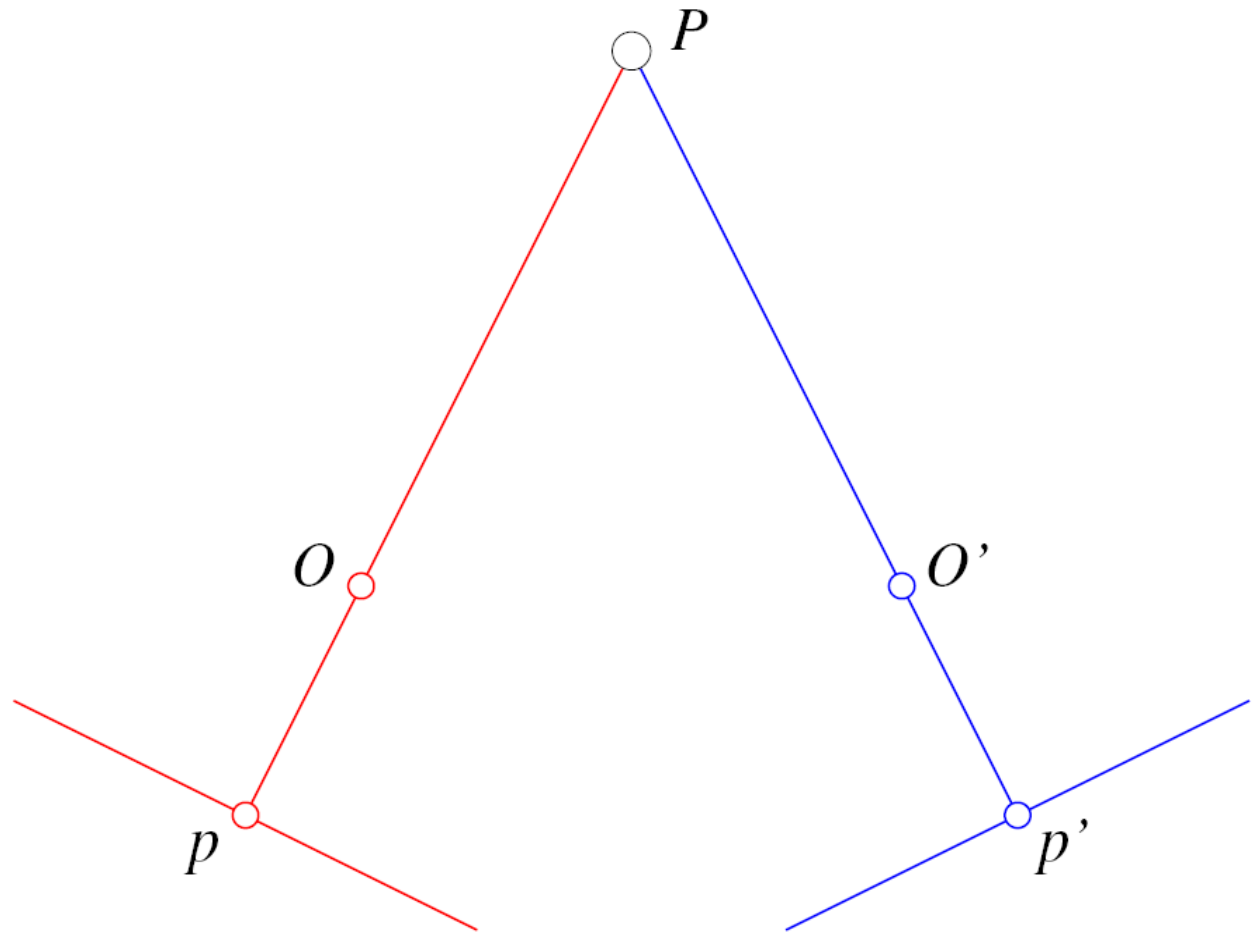
- If I needed to find out how far point is away from me, I could use triangulation and two views



(Graphic from Khurram Shaffique)

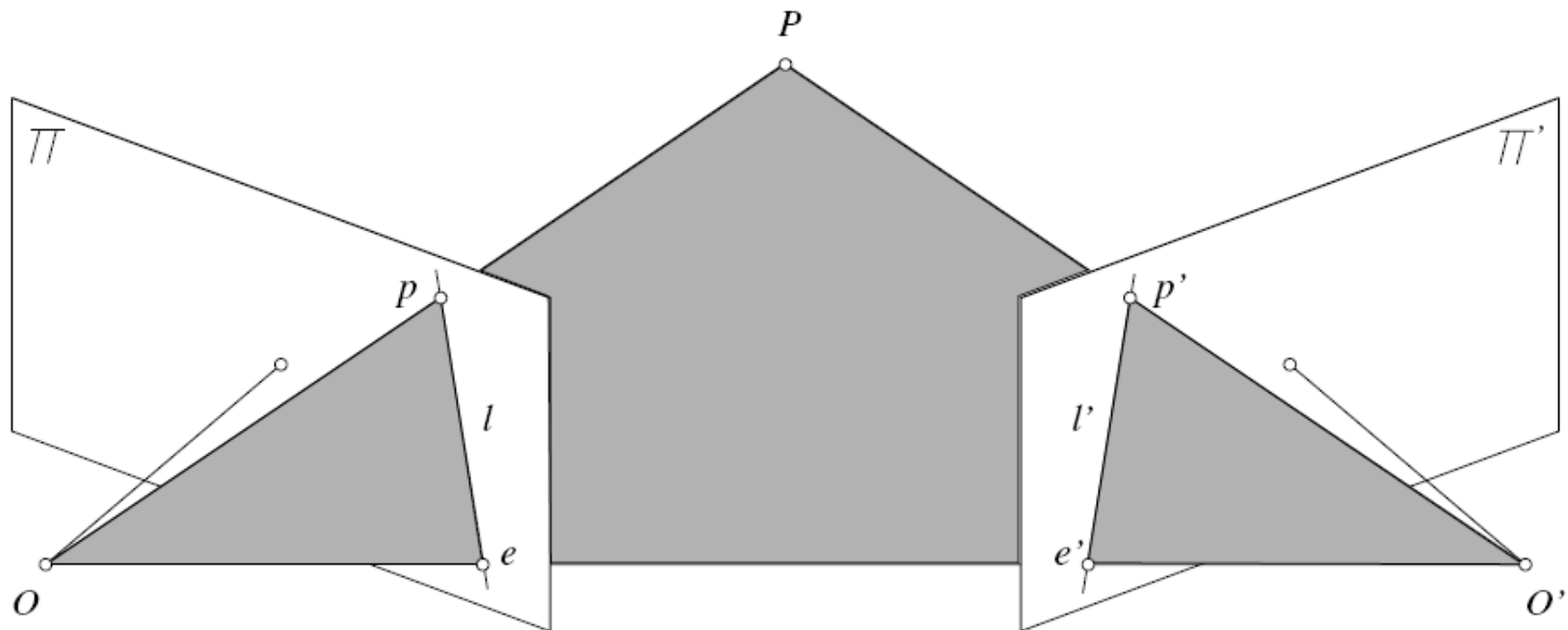
To see this

- I have three unknowns – (x, y, z)
- I have 4 equations
 - P projects to $(u$ and $v)$
 - P projects to $(u$ and $v)$



Today

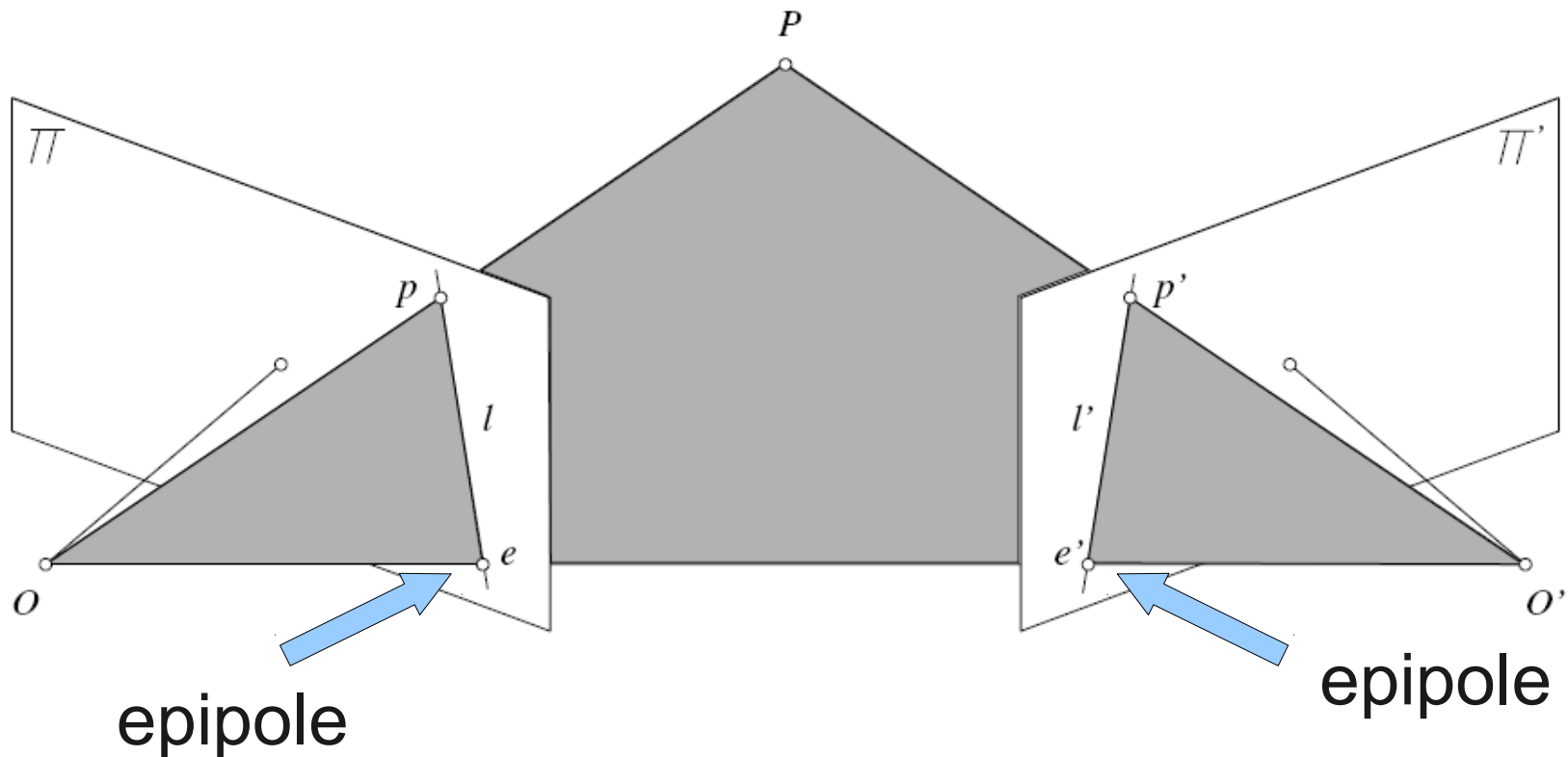
- For the rest of the lecture we will talk about the geometry of multiple views
- To begin we will talk about epipolar geometry



(Image from Forsyth and Ponce)

Epipoles

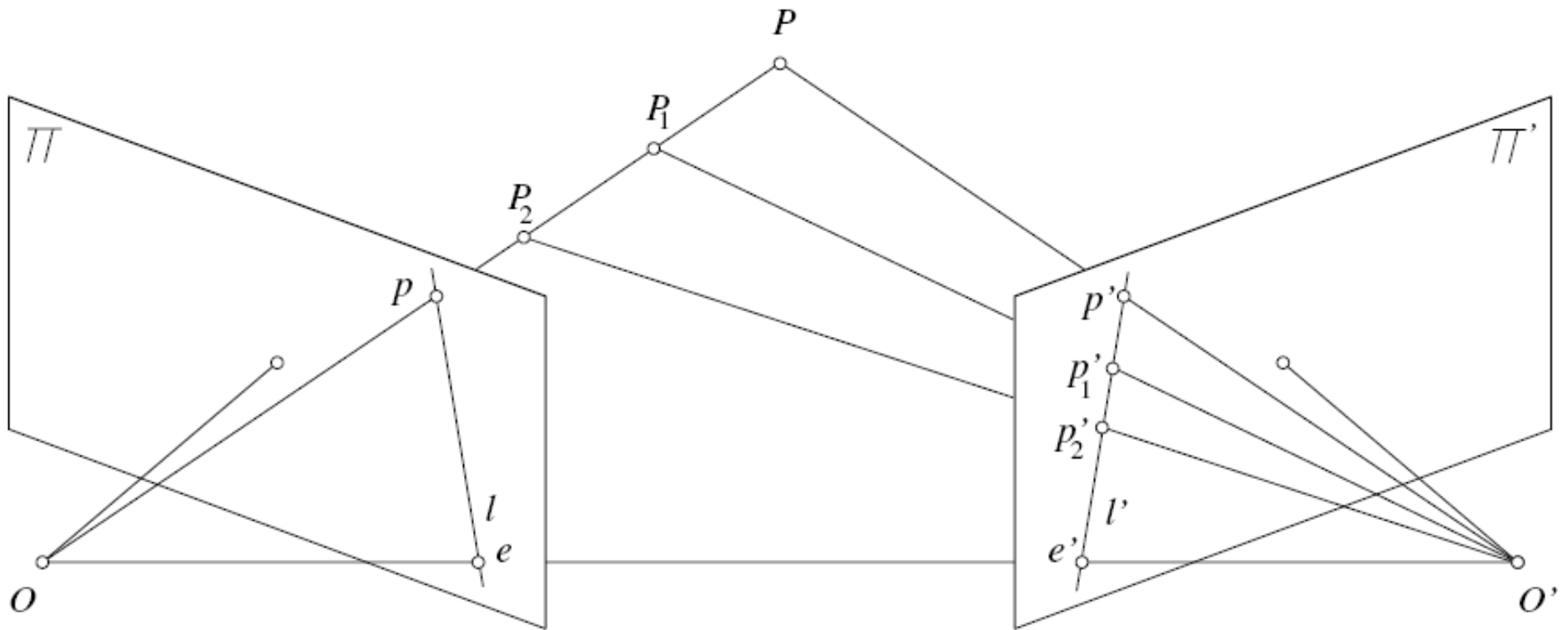
- The projection of the optical centers of each camera



(Image from Forsyth and Ponce)

Epipoles and epipolar lines

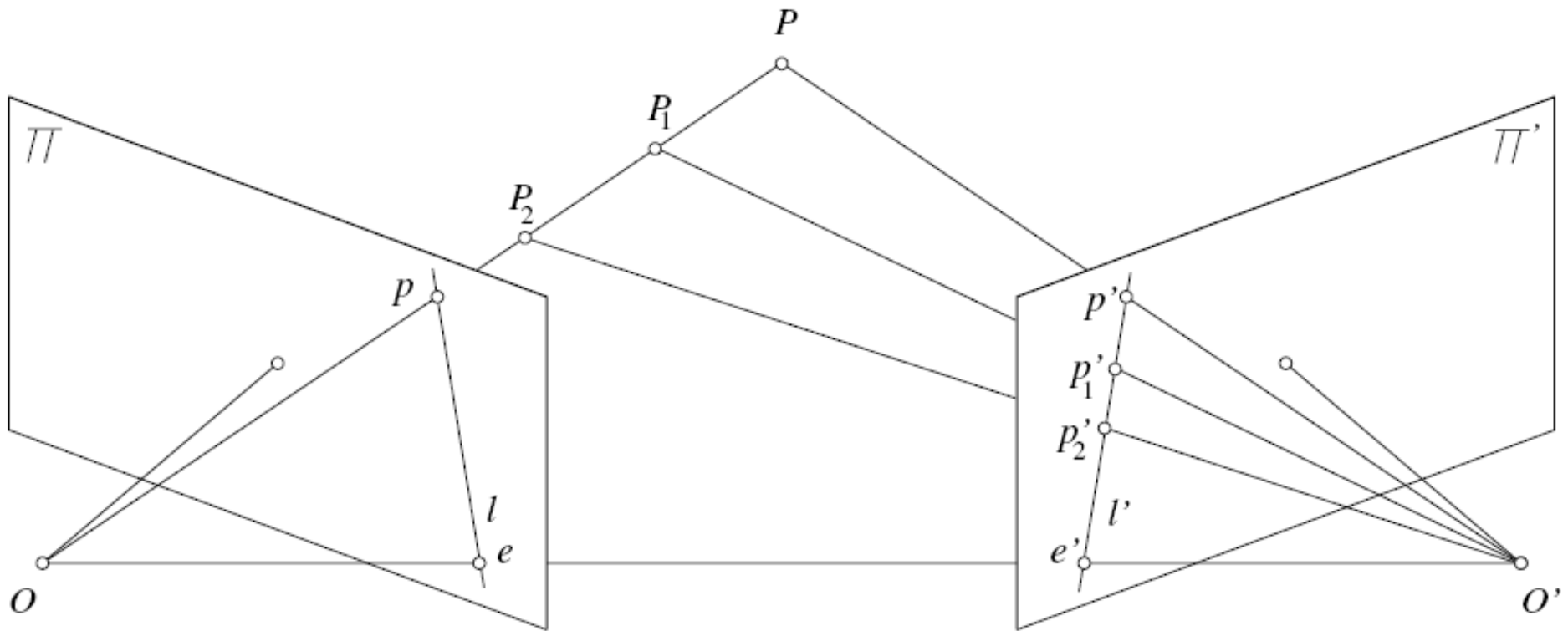
- For a point P , we may not know where it projects to (p')
- We do know that it lies on an *epipolar* line



(Image from Forsyth and Ponce)

When the cameras are calibrated

- The vectors \overrightarrow{Op} , $\overrightarrow{OO'}$, and $\overrightarrow{O'p'}$ are coplanar
- Can be expressed as $\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$



(Image from Forsyth and Ponce)

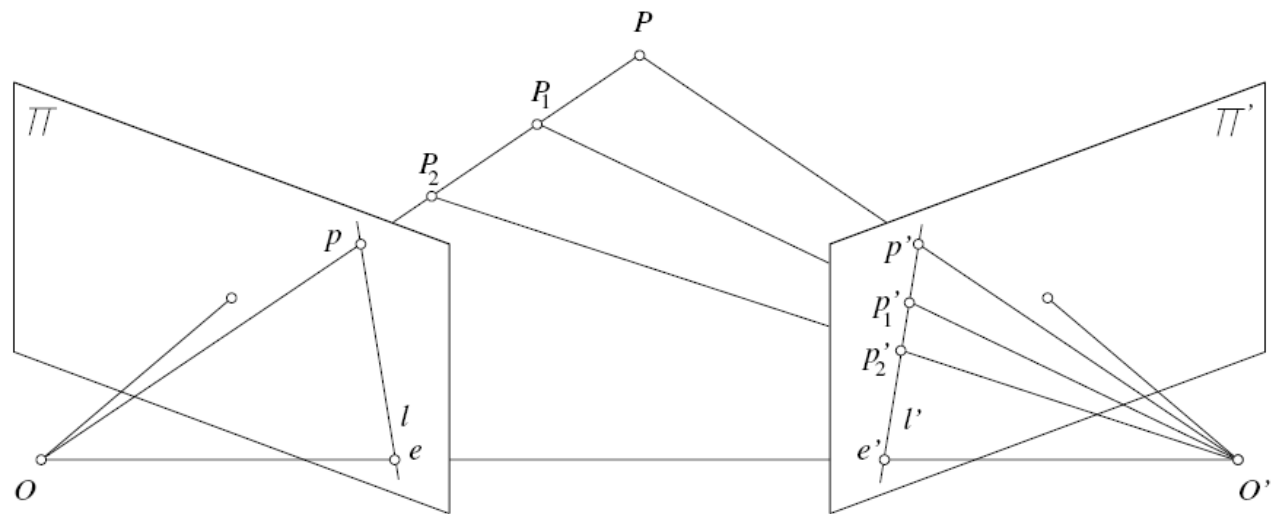
Relating the two views

- First, remember that we have have points in two coordinate systems
- Can express a rigid transformation (translation and rotation) between the two system

R is a Rotation Matrix

$${}^O p' = {}^O O' \mathcal{R} p' + t$$

T is a Translation vector



(Image from Forsyth and Ponce)

Now, rewrite constraint in terms of the coordinates of the left camera

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$

$$p \cdot \left[t \times \begin{matrix} O' \\ O \end{matrix} \mathcal{R} p' \right]$$

For simplicity:

$$p \cdot [t \times \mathcal{R} p']$$

The essential matrix

- Starting with:

$$\mathbf{p} \cdot [t \times \mathcal{R}\mathbf{p}']$$

- A cross product can be rewritten as a matrix multiplication, leading to the constraint

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

\mathcal{E} is called the *essential* matrix

Properties of the essential matrix

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

- 3x3 matrix
- Rank 2
- Only relates the extrinsic parameters of the two views

What if the cameras aren't calibrated

- The relationship still holds, but we have to calibrate the cameras first.
- Those calibration matrices, combined with the essential matrix are known as the *fundamental matrix*

$$\mathbf{p}^T \mathcal{F} \mathbf{p}' = 0$$

- Encodes information from the intrinsic and extrinsic parameters
- Also Rank 2

Finding the fundamental matrix

- Basic algorithm: 8-point algorithm
- Find 8 corresponding points in the images
- Once you have the corresponding p and p' points,

$$p^T \mathcal{F} p' = 0$$

- Is linear in \mathcal{F}

Finding the fundamental matrix

- Minimize over the coefficients of F

$$\sum_{i=1}^n (\mathbf{p}_i^T \mathcal{F} \mathbf{p}'_i)^2$$

- Improvements proposed by Hartley (1995)
 - Normalize corresponding points
 - After you compute the first estimate of F, enforce Rank-2 constraint using SVD (see text)
- Significantly improves results

Adding more views

- Text also describes the geometry of 3 or more

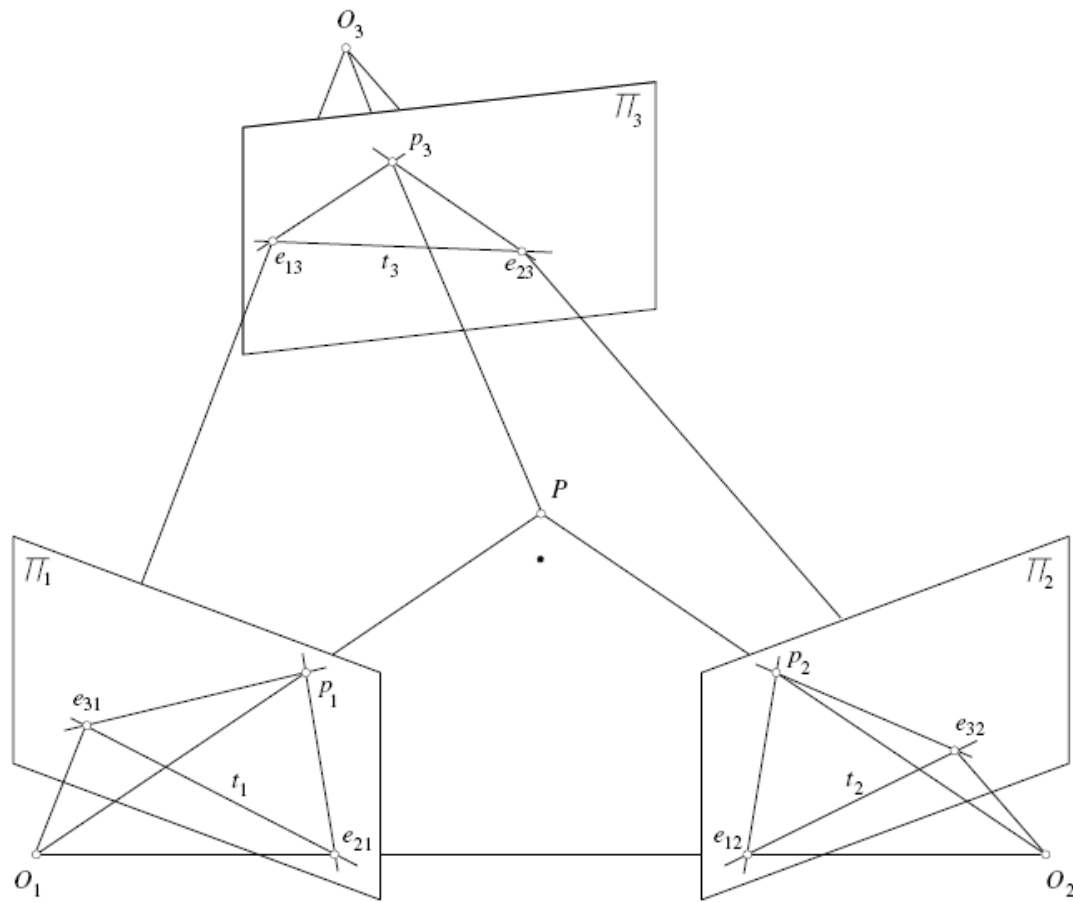


Figure 11.5. Trinocular epipolar geometry.

(From Forsyth and Ponce)

How do we use the fundamental matrix?

- It can tell us where to look for points in the other image
- The quantity $(p^T \mathcal{F})$ is a vector
- So, $p^T \mathcal{F} p' = 0$ defines a line
- This tells us where to look for the point that corresponds to \mathbf{p}
- **(Demo)**

Triangulating points

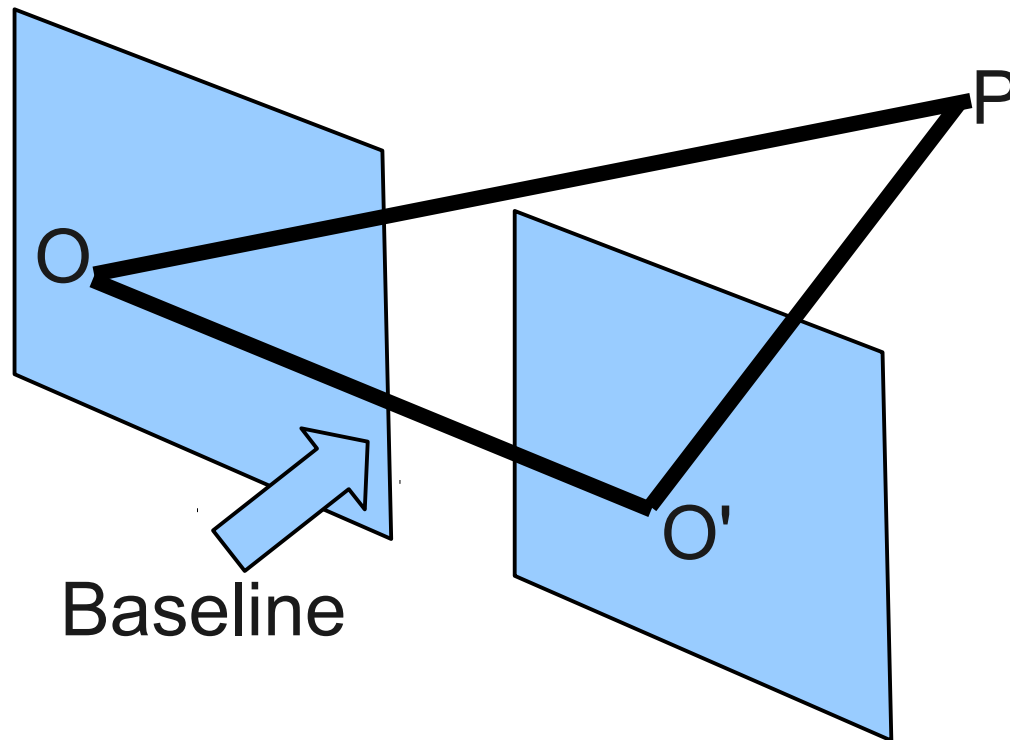
- If we know the projection matrices M and the locations of corresponding points,
- We can generate a set of linear constraints

$$\begin{cases} \mathbf{p} \times \mathcal{M}\mathbf{P} = 0 \\ \mathbf{p}' \times \mathcal{M}'\mathbf{P} = 0 \end{cases} \iff \begin{pmatrix} [\mathbf{p}_{\times}] \mathcal{M} \\ [\mathbf{p}'_{\times}] \mathcal{M}' \end{pmatrix} \mathbf{P} = 0.$$

- And solve those using a linear-least squares technique

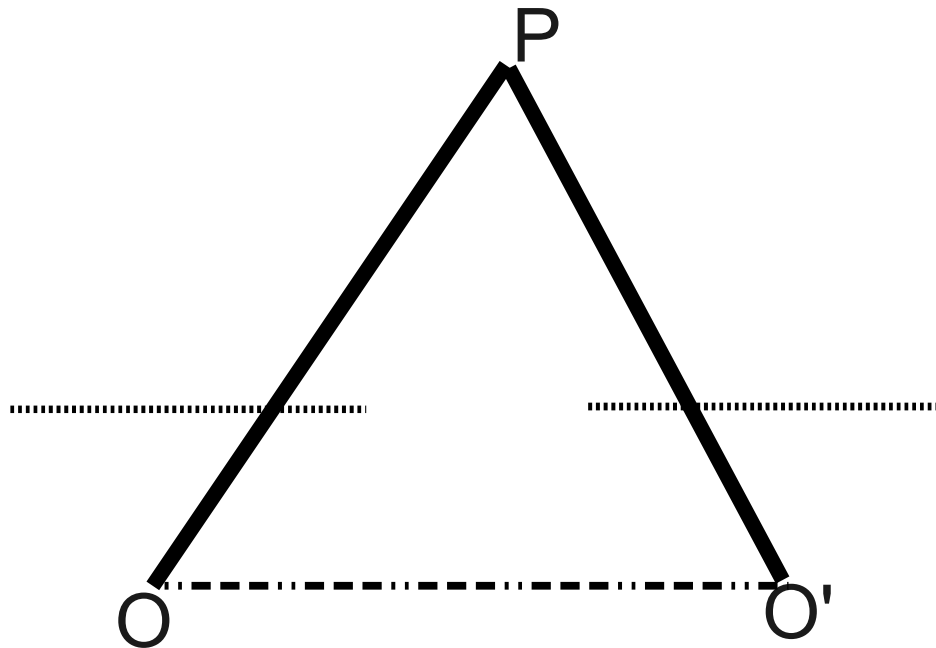
But it can be easier

- Image planes are parallel
- Also parallel to line connecting the two optical centers



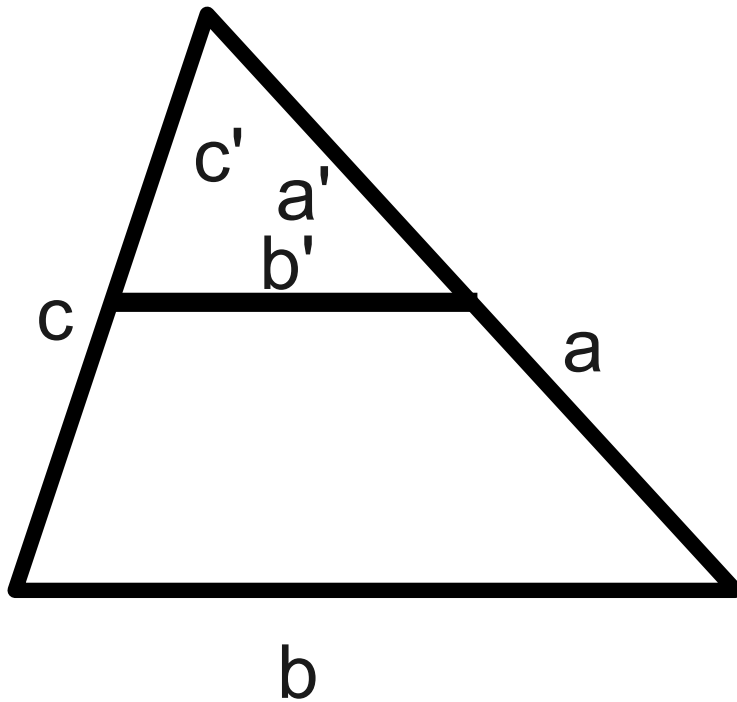
Overhead view

- Now triangulation is much easier

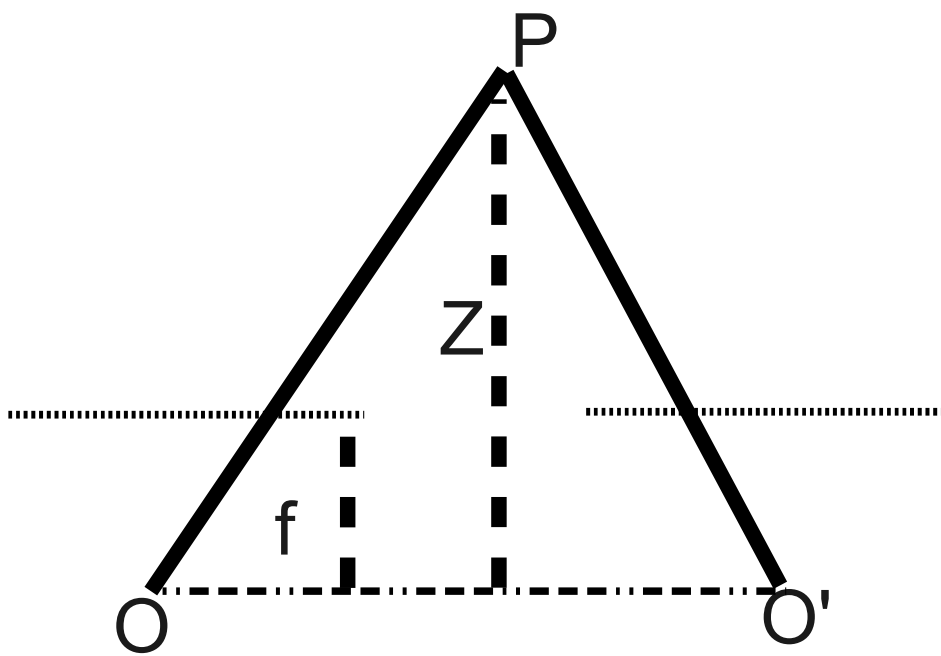


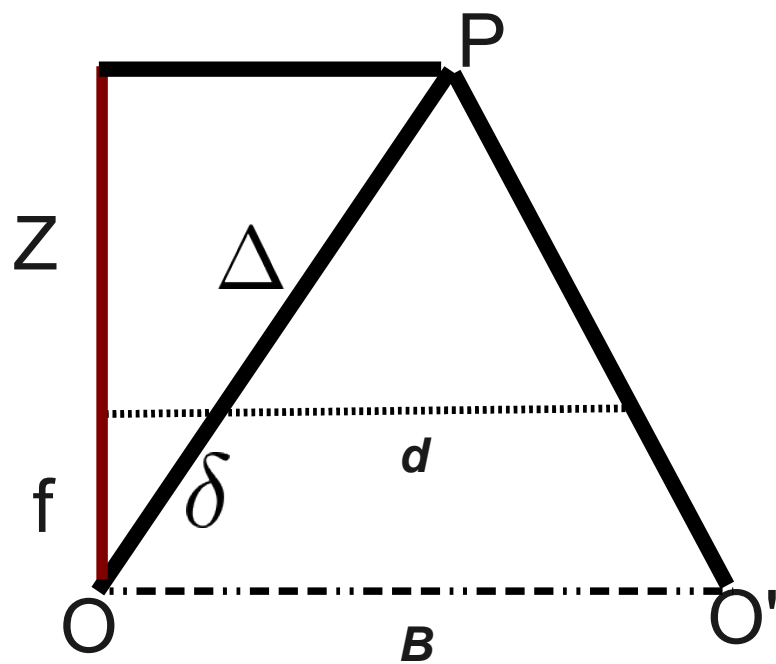
First, Some Geometry

- Similar triangles

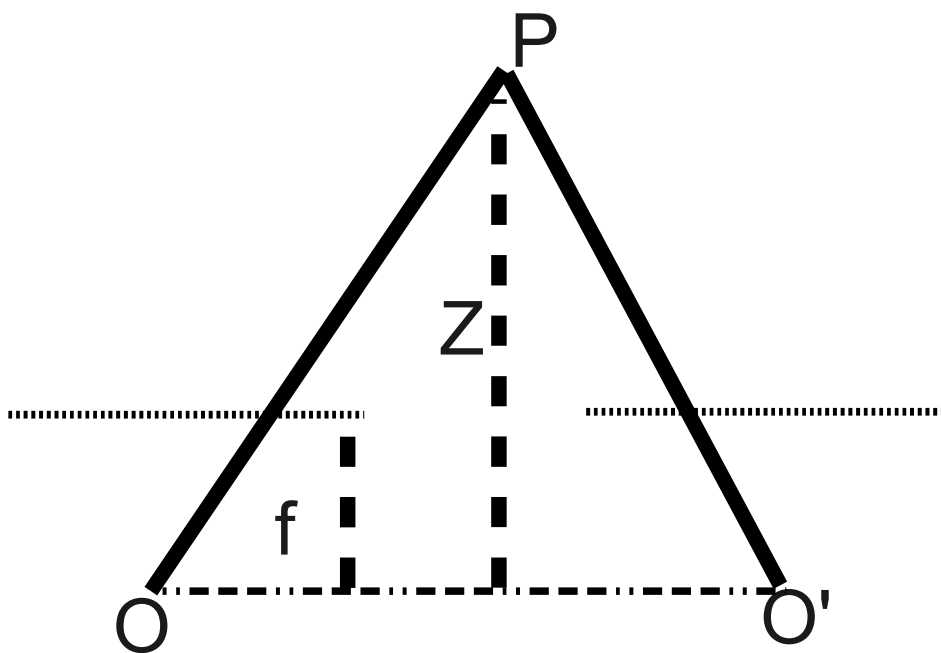


$$\frac{a'}{a} = \frac{b'}{b} = \frac{c'}{c}$$

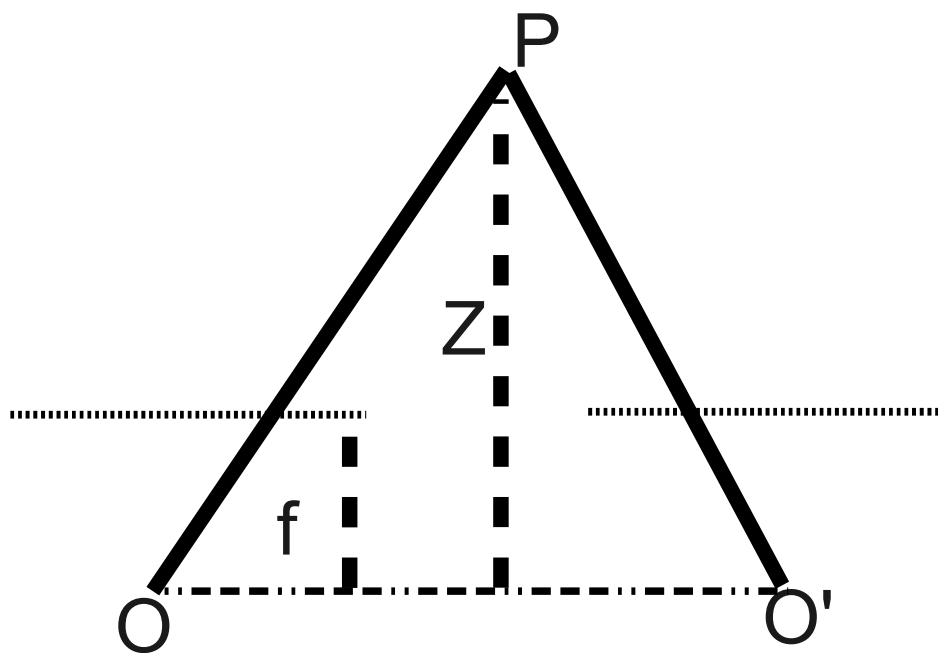




$$\frac{Z}{f} = \frac{\Delta}{\delta} = \frac{B}{d} \quad Z = \frac{fB}{d}$$



$$Z = \frac{f(O' - O)}{(p'_x - p_x)} \quad Z = \frac{fB}{d}$$



$$Z = f \frac{O' - O}{d}$$

$$d = p_x - p'_x$$



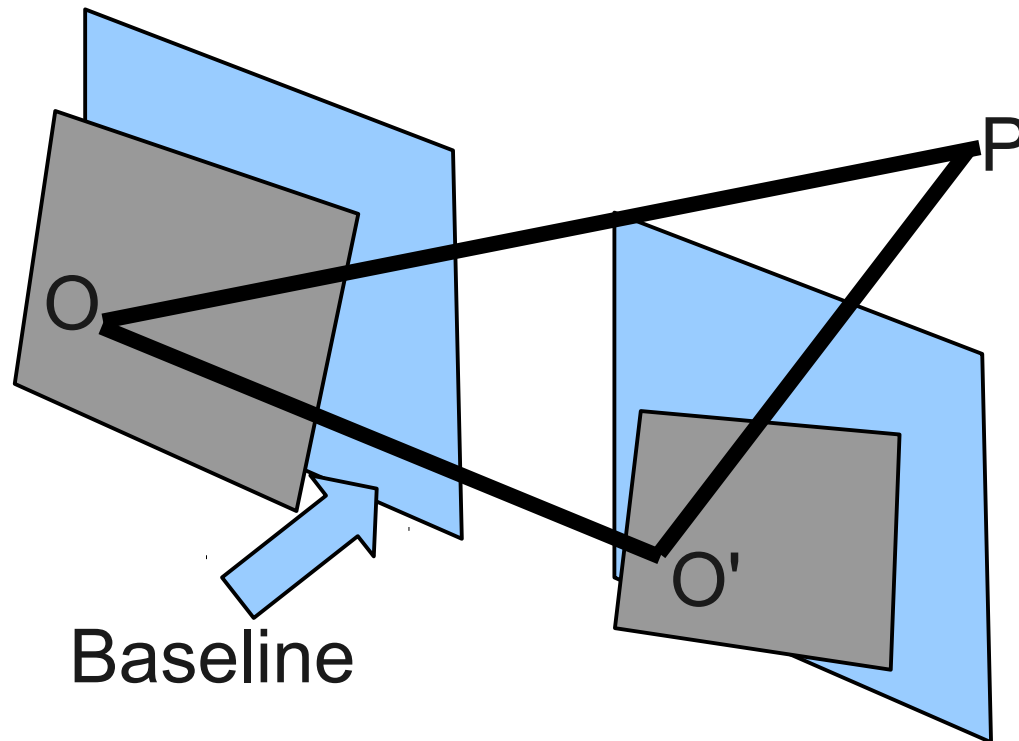
Disparity

Image Rectification

- All we need to calculate the depth is the distance between corresponding points
- Would be easier to implement if the epipolar lines corresponded with scanlines
- Image rectification transforms the images so they have this property

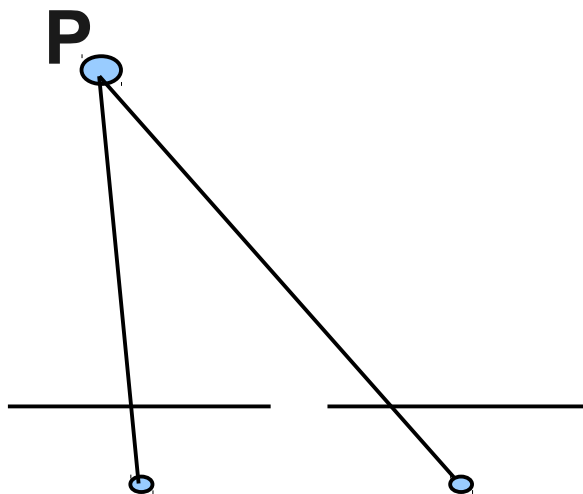
Image Rectification

- For each image, compute mapping between image from the camera and the rectified image



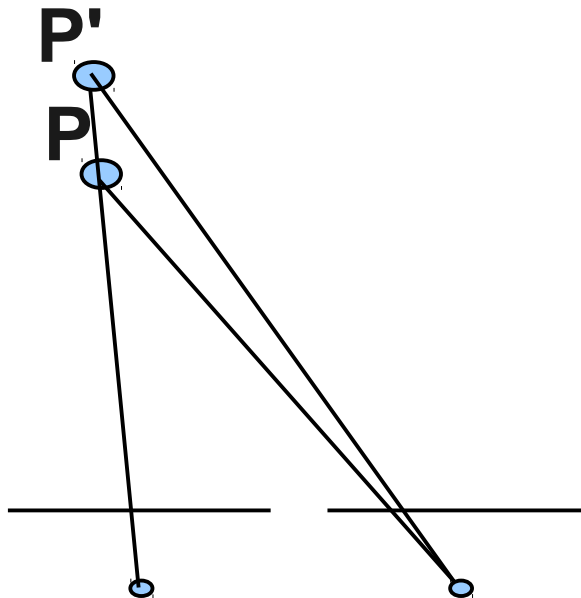
Intuitions on Rectification

- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth



Intuitions on Rectification

- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth



Intuitions on Rectification

- But if I just rotate the camera, then I don't need to know the depth
- Notice that P' and P still project to the same point in the rotated camera

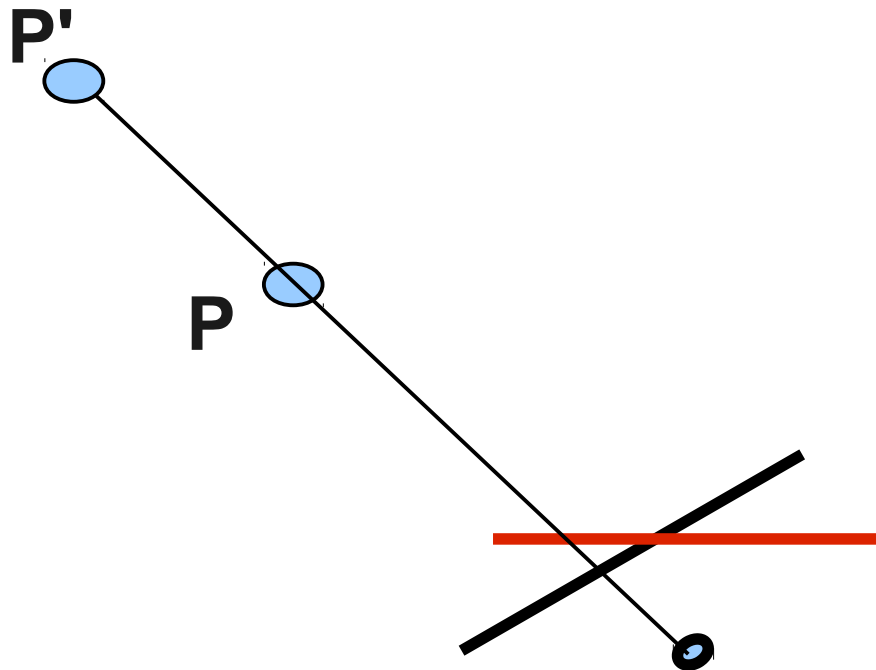
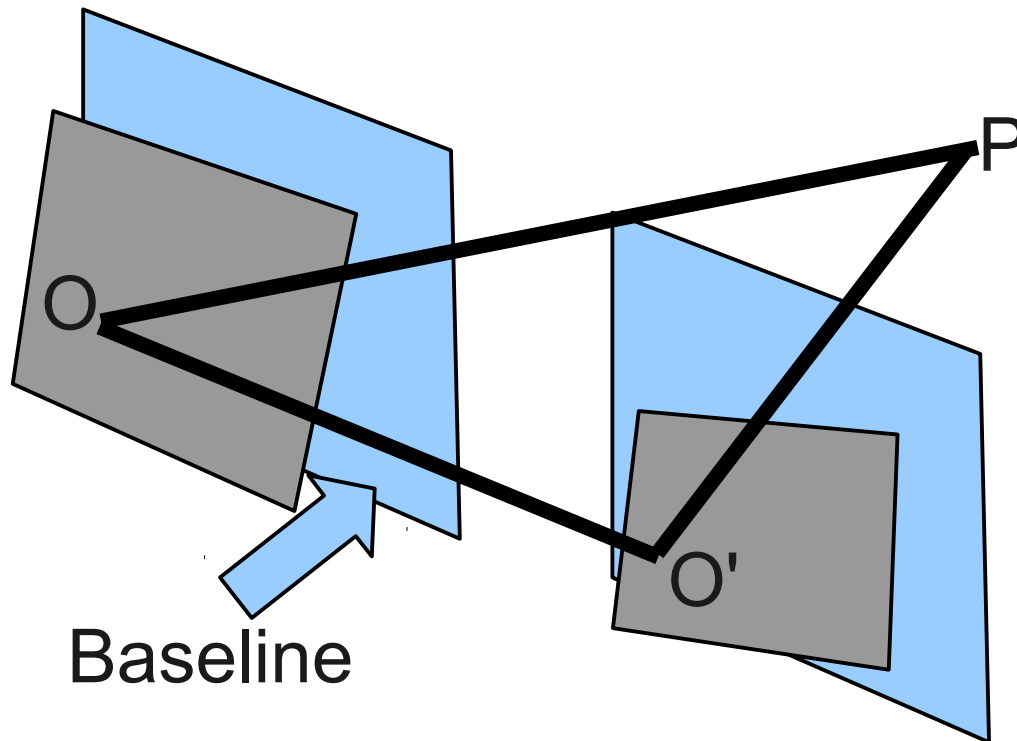
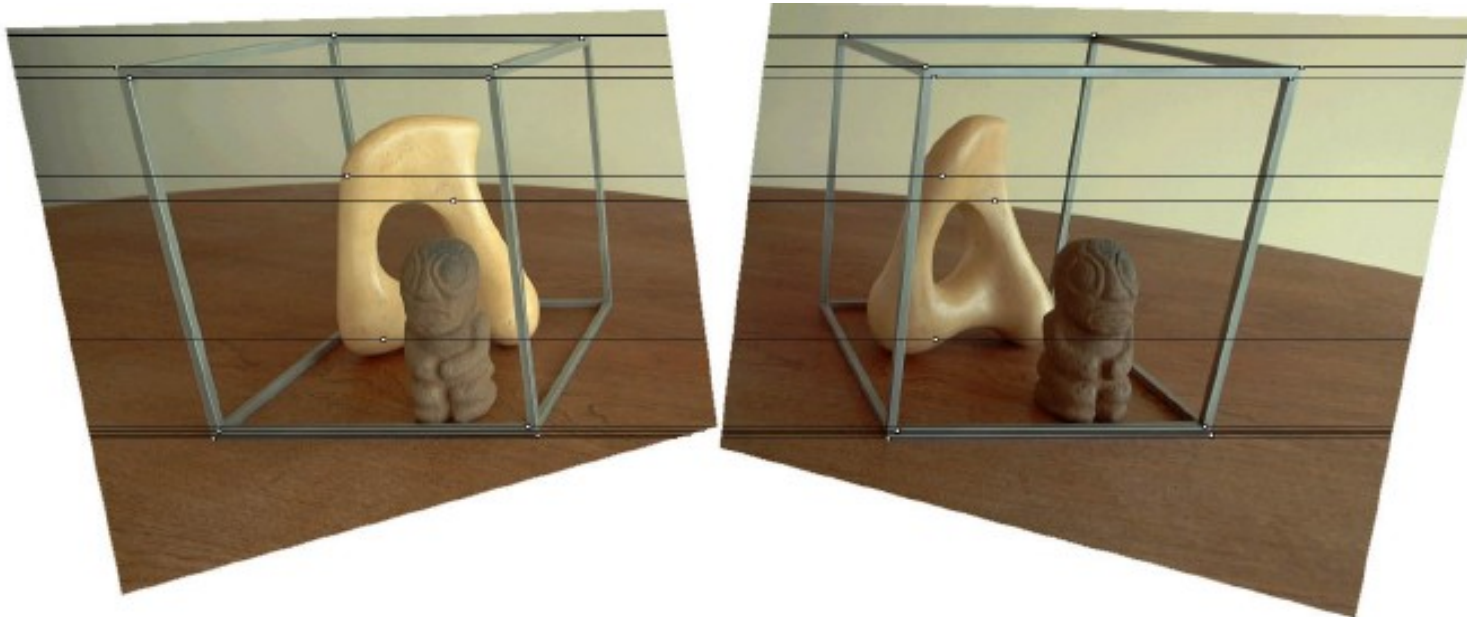
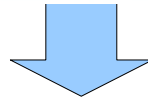
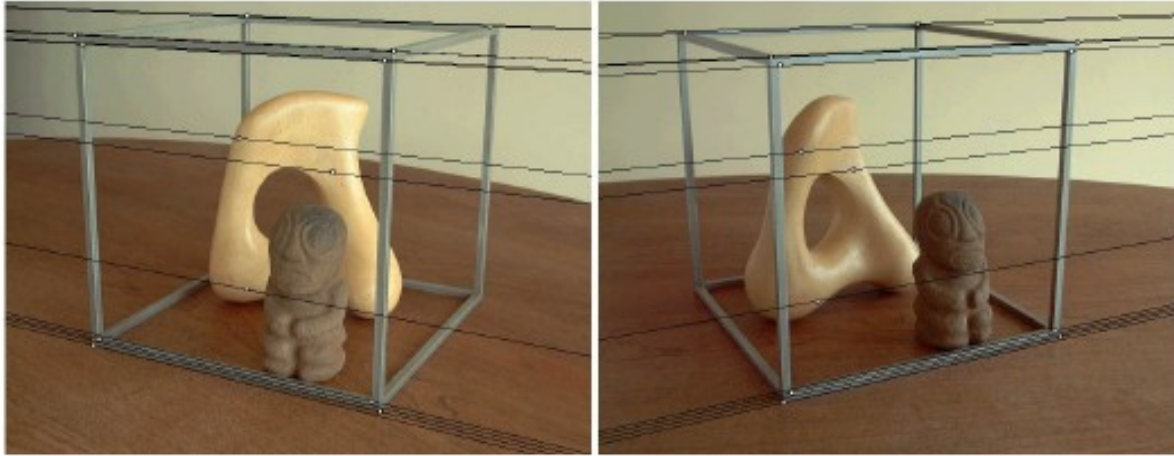


Image Rectification

- I can rotate views to be parallel to each other and baseline



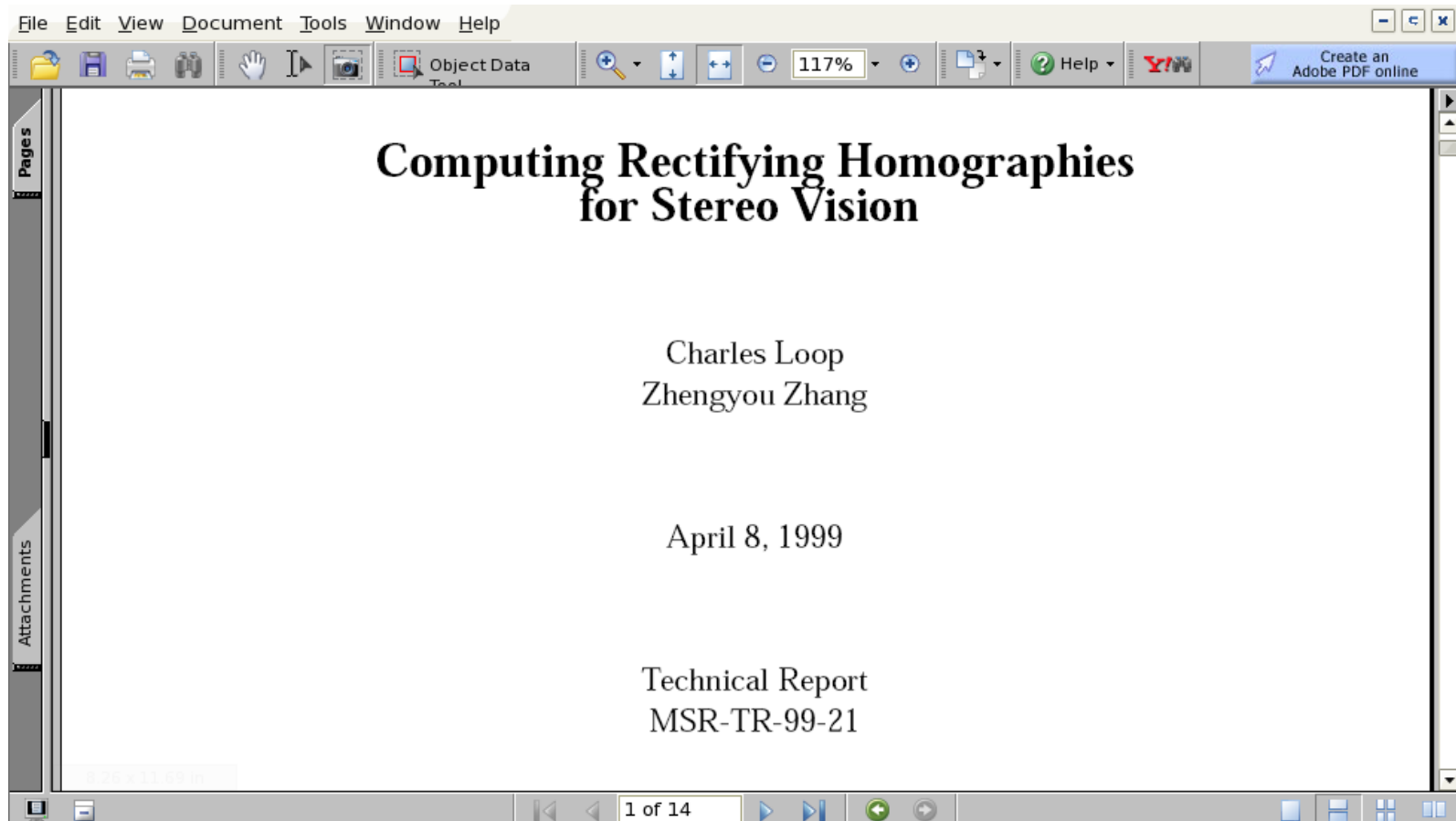
Lines up Epipolar lines with scanlines



(Image from Loop and Zhang)

Rectifying images

- Can be computed using the fundamental matrix



Interesting Connections to Fundamental Matrix

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

- After rectification, it should have this form

Implication

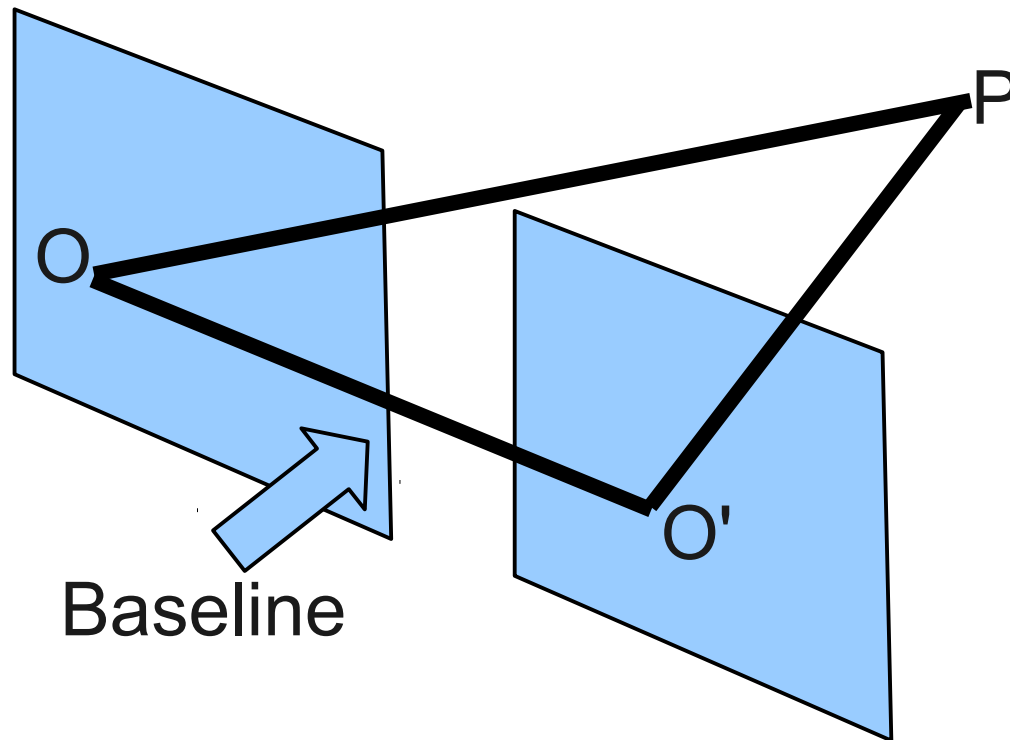
$$[x' \ y' \ 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$y' - y = 0$$

$$y = y'$$

Where are the epipoles?

- In this case, the epipole is at infinity
- In homogeneous coordinates, this is $[1 \ 0 \ 0]$

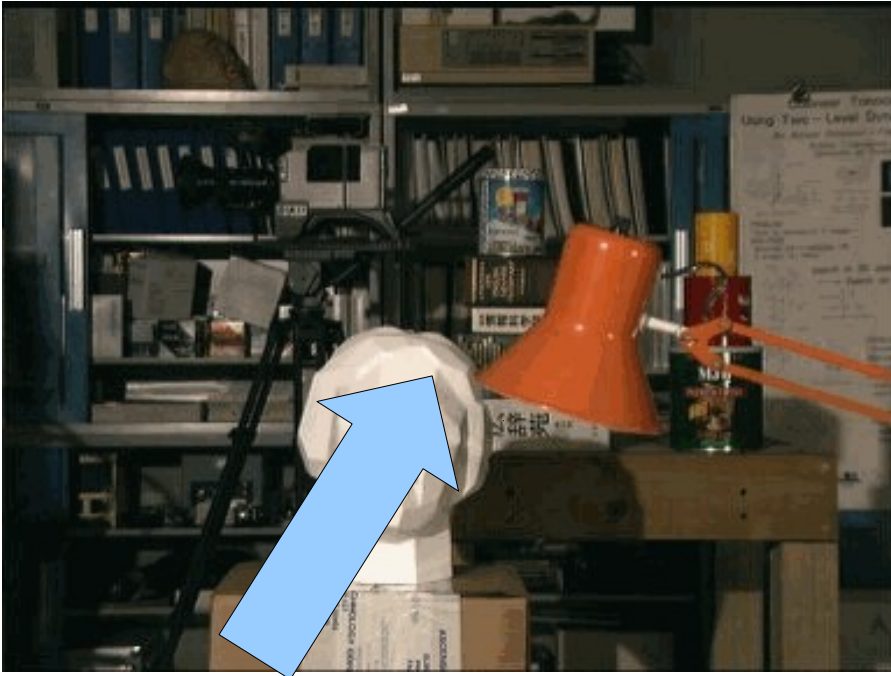


Now to the hard problem

- Once we have rectified images, the hard problem is find the corresponding points
- Then we can triangulate
- Known as the correspondence problem
- When people say “stereo algorithm”, they usually mean “correspondence algorithm”

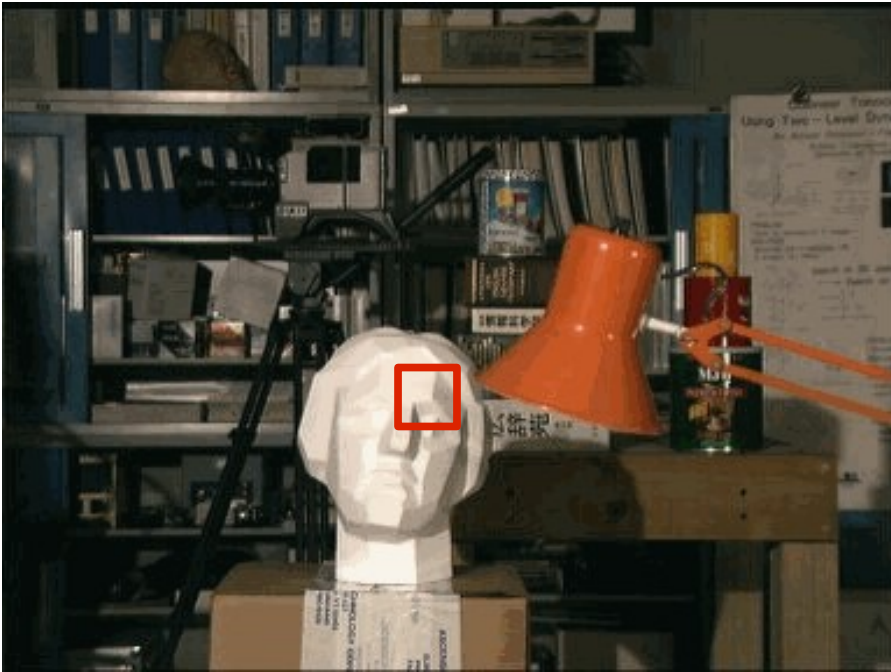


A very simple algorithm

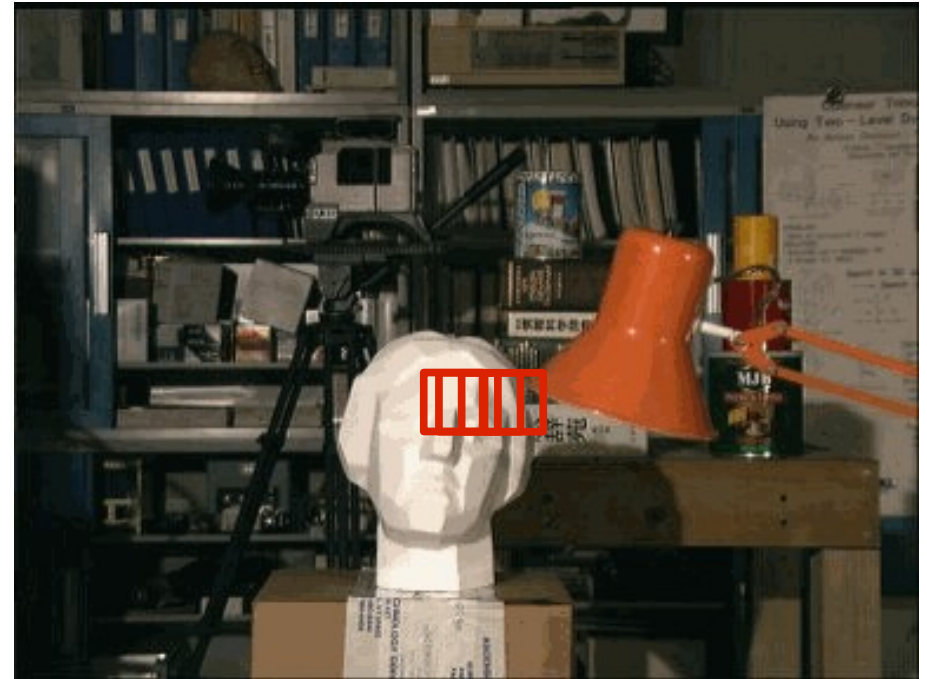


To find the point that matches this point

A very simple algorithm



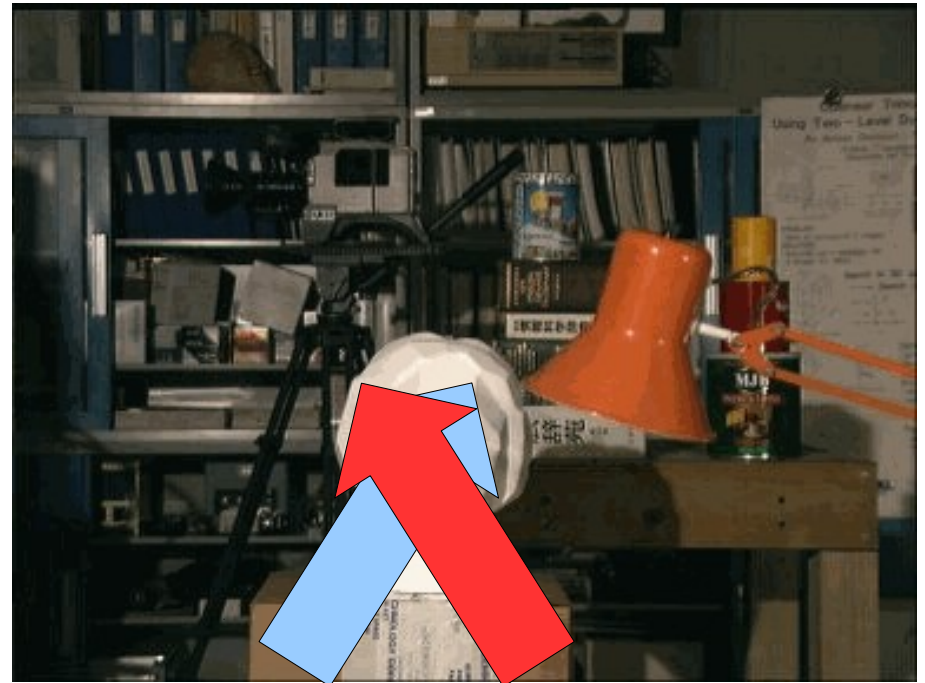
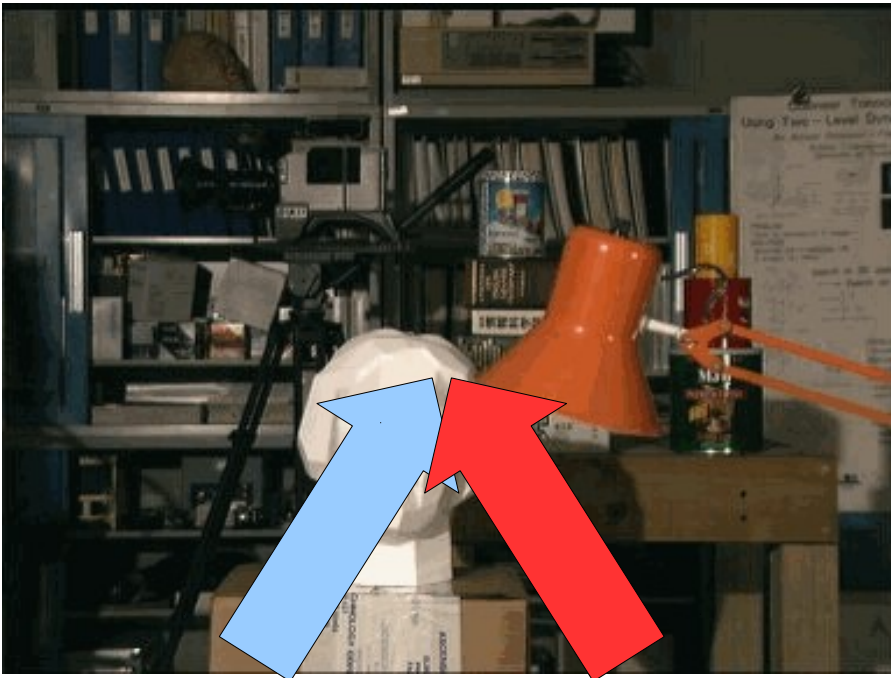
Look at a square window
around that point



and compare it to
windows in the other
image

Could measure sum of
squared differences
(SSD) or correlation

What's wrong with this algorithm?



This set of correspondences don't bother it

Areas of the image without texture will be a problem for this algorithm.

To do better we need a better model of images

- We can make reasonable assumptions about the surfaces in the world
- Usually assume that the surfaces are smooth
- Can pose the problem of finding the corresponding points as an energy (or cost) minimization

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d).$$

- The data term measures how well the local windows match up for different disparities

The smoothness cost

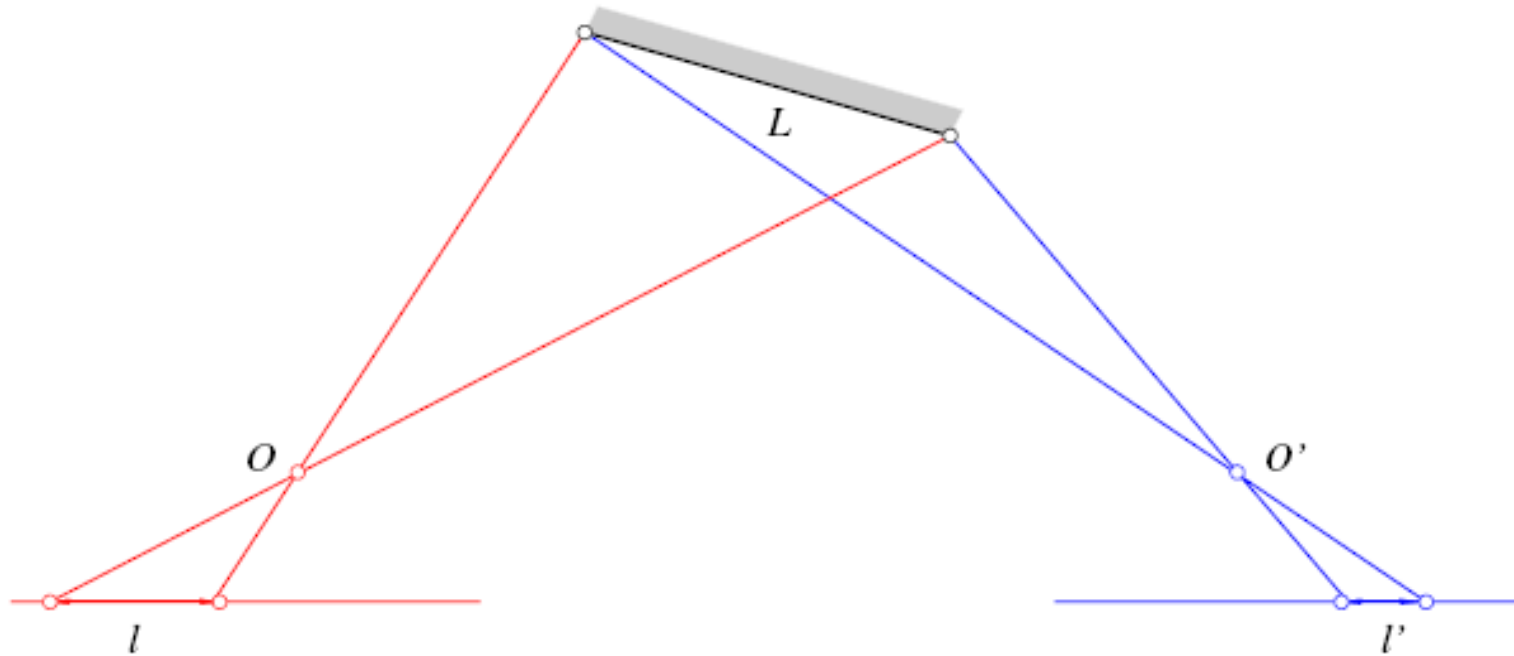
- The smoothness is usually implemented by penalizing differences in the disparity

$$E_{smooth}(d) = \sum_{(x,y)} \rho(d(x,y) - d(x+1,y)) + \rho(d(x,y) - d(x,y+1)),$$

- Different penalty functions lead to different assumptions about the surfaces
 - $\log(1+x^2)$
 - x^2
 - First deriv, Second derivative

Pitfalls in stereo

- Foreshortening
 - Projected size of a patch of the surface can change significantly



(Image from Forsyth and Ponce)

Pitfalls in Stereo

- Occlusion
 - Don't always see points in both images



So, which algorithm works best?

- www.middlebury.edu/stereo
- Results are a combination of the model and the optimization algorithm

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d).$$

- Best results are obtained from algorithms using a model known as a Markov Random Field
 - This will be our next topic