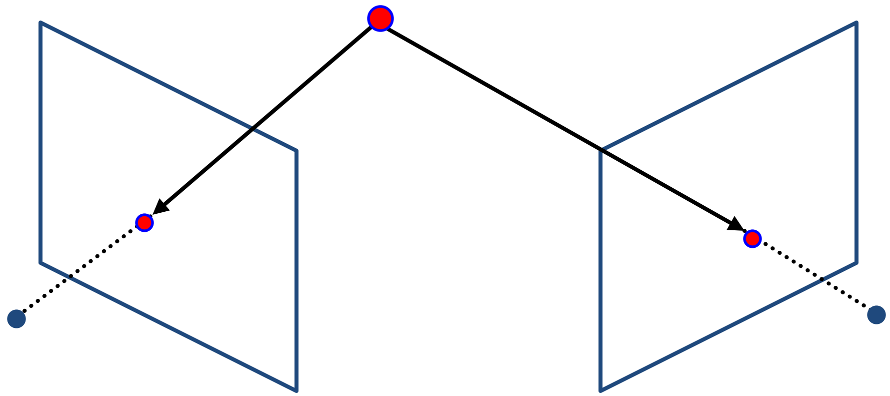
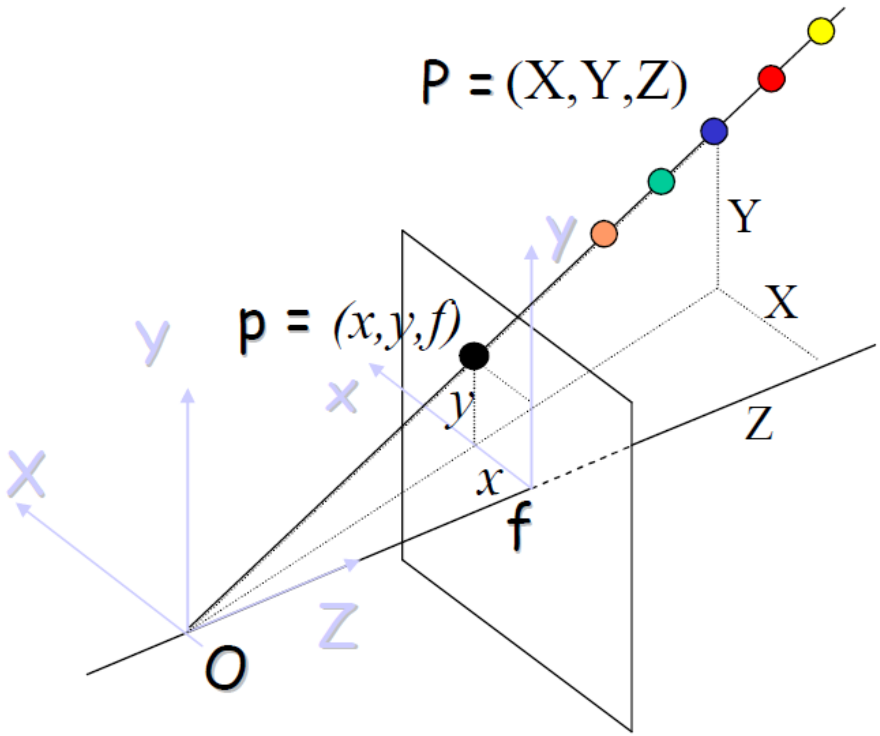
Stereo and Structure from Motion

CS143, Brown
James Hays



Many slides by
Kristen Grauman, Robert
Collins, Derek Hoiem,
Alyosha Efros, and
Svetlana Lazebnik

Why Stereo Vision?



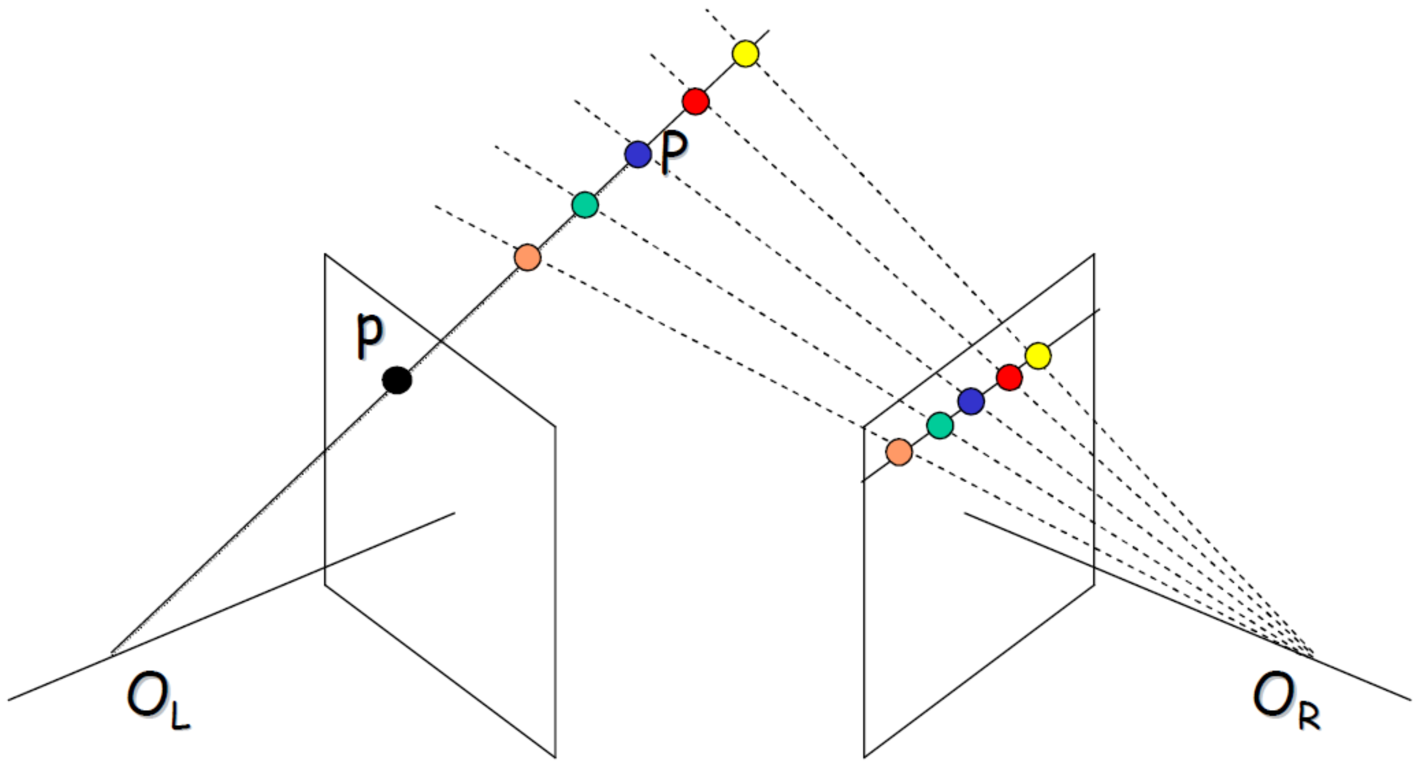
$$x = f\frac{X}{Z} = f\frac{kX}{kZ}$$

$$y = f\frac{Y}{Z} = f\frac{kY}{kZ}$$

Fundamental Ambiguity:

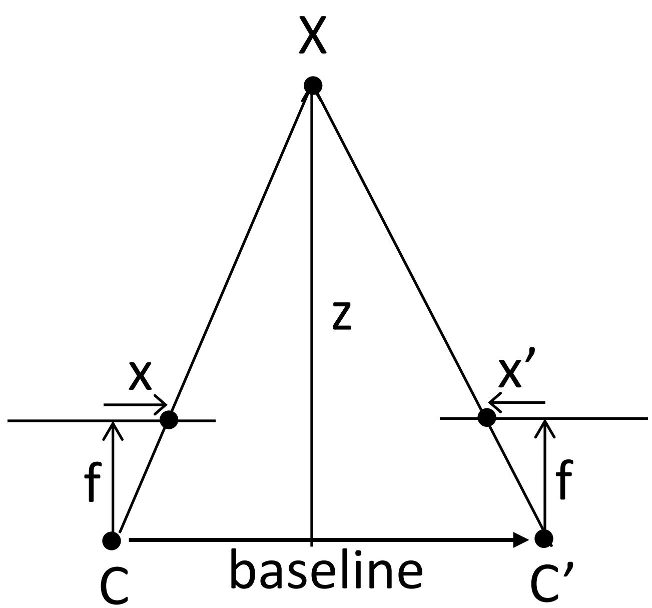
Any point on the ray OP has image p

Why Stereo Vision?



A second camera can resolve the ambiguity, enabling measurement of depth via triangulation.

Depth from disparity



$$(X - X') / f \;\; = \text{baseline} / z$$

$$X - X' \; = (\text{baseline}*f) / z$$

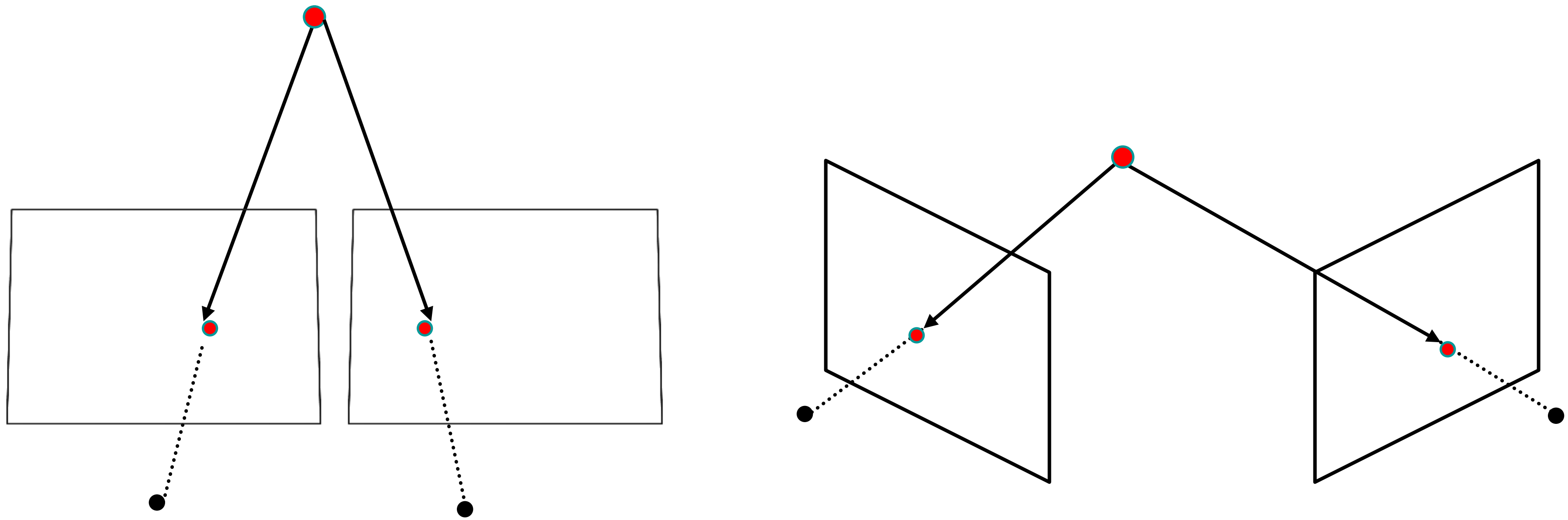
$$z = (\text{baseline}*f) / (X - X')$$

Outline

- Human stereopsis
- Stereograms
- Epipolar geometry and the epipolar constraint
  - Case example with parallel optical axes
  - General case with calibrated cameras

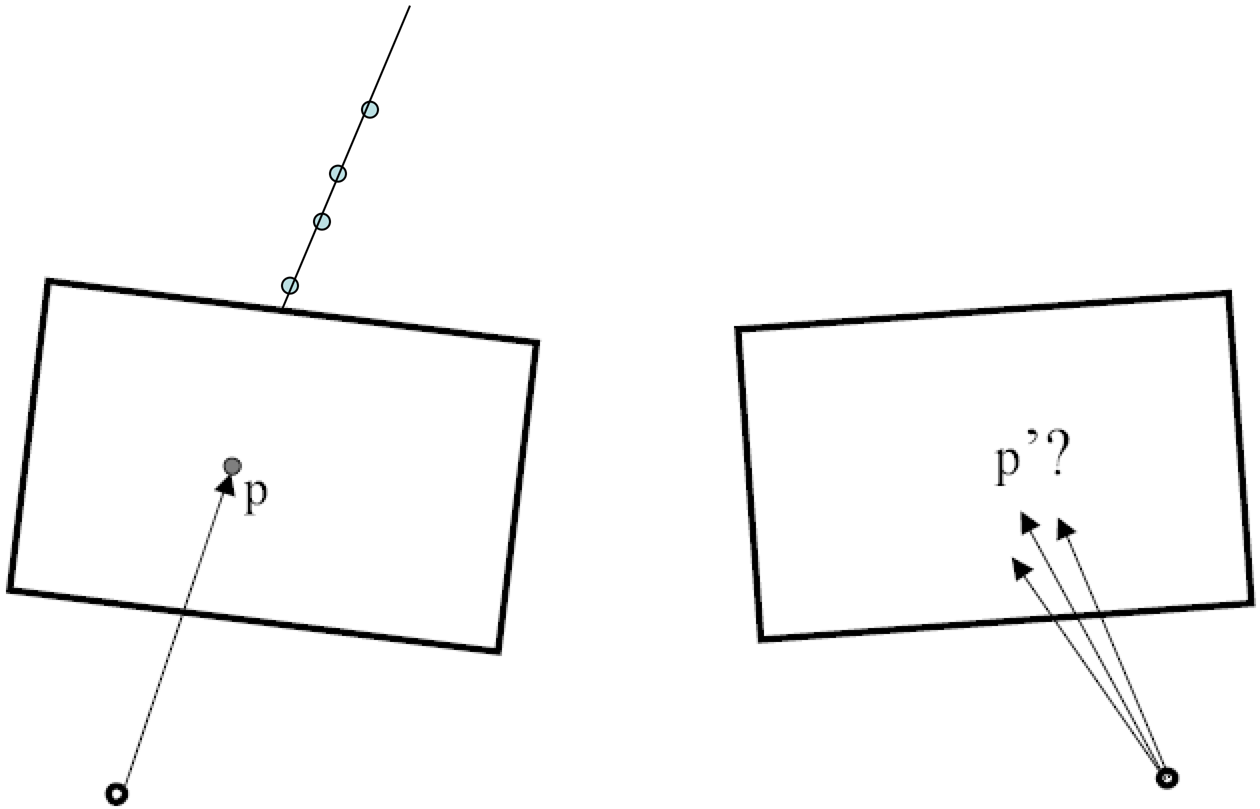
General case, with calibrated cameras

- The two cameras need not have parallel optical axes.



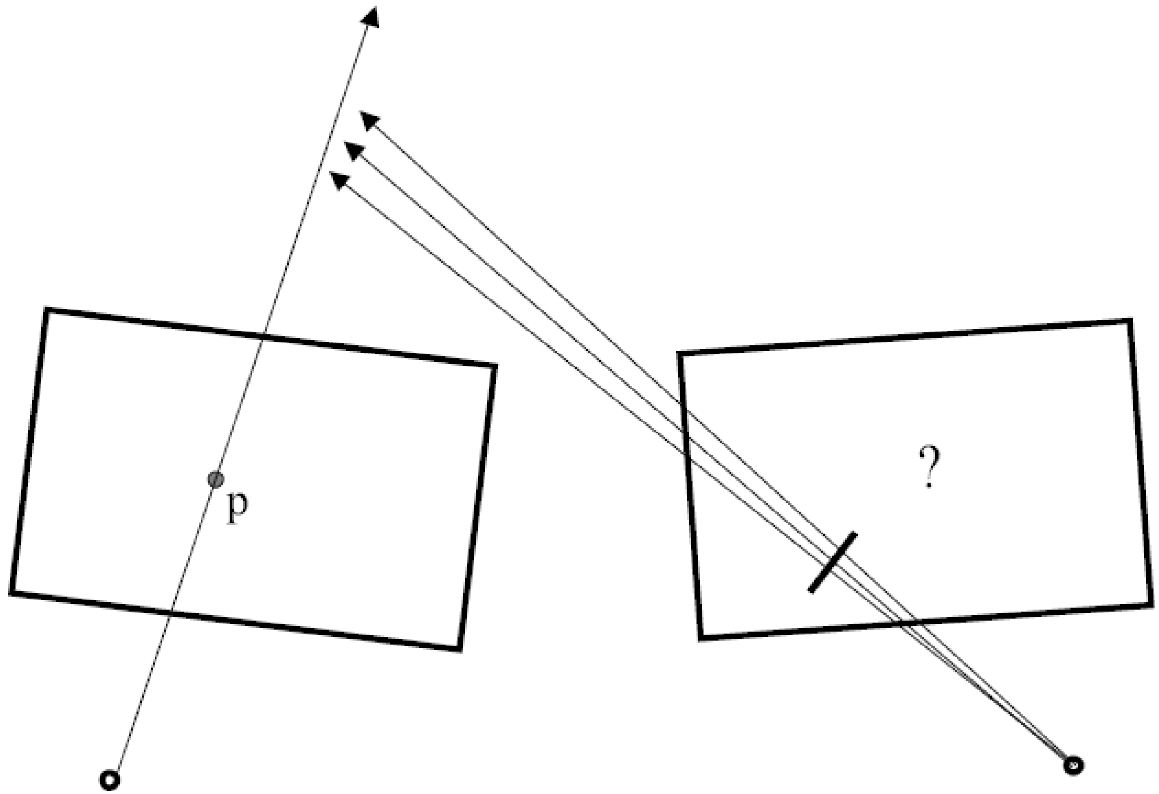
Vs.

Stereo correspondence constraints

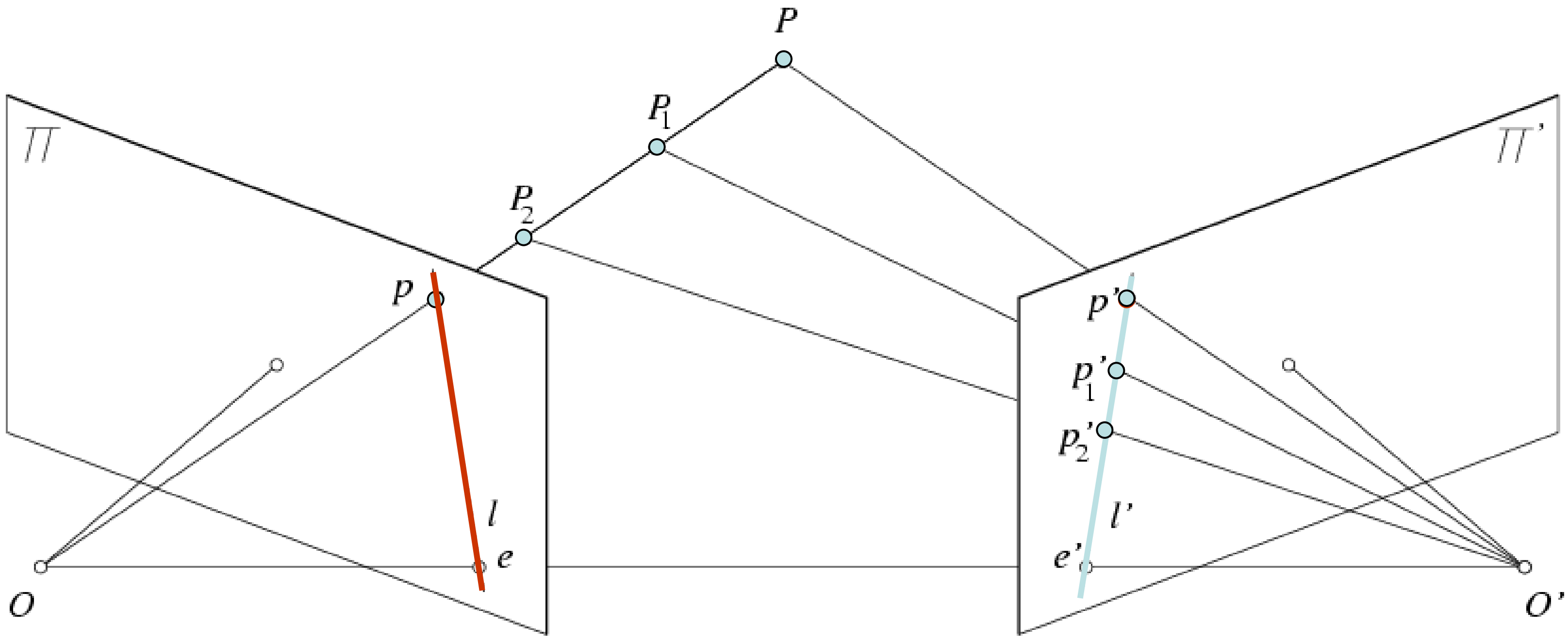


- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



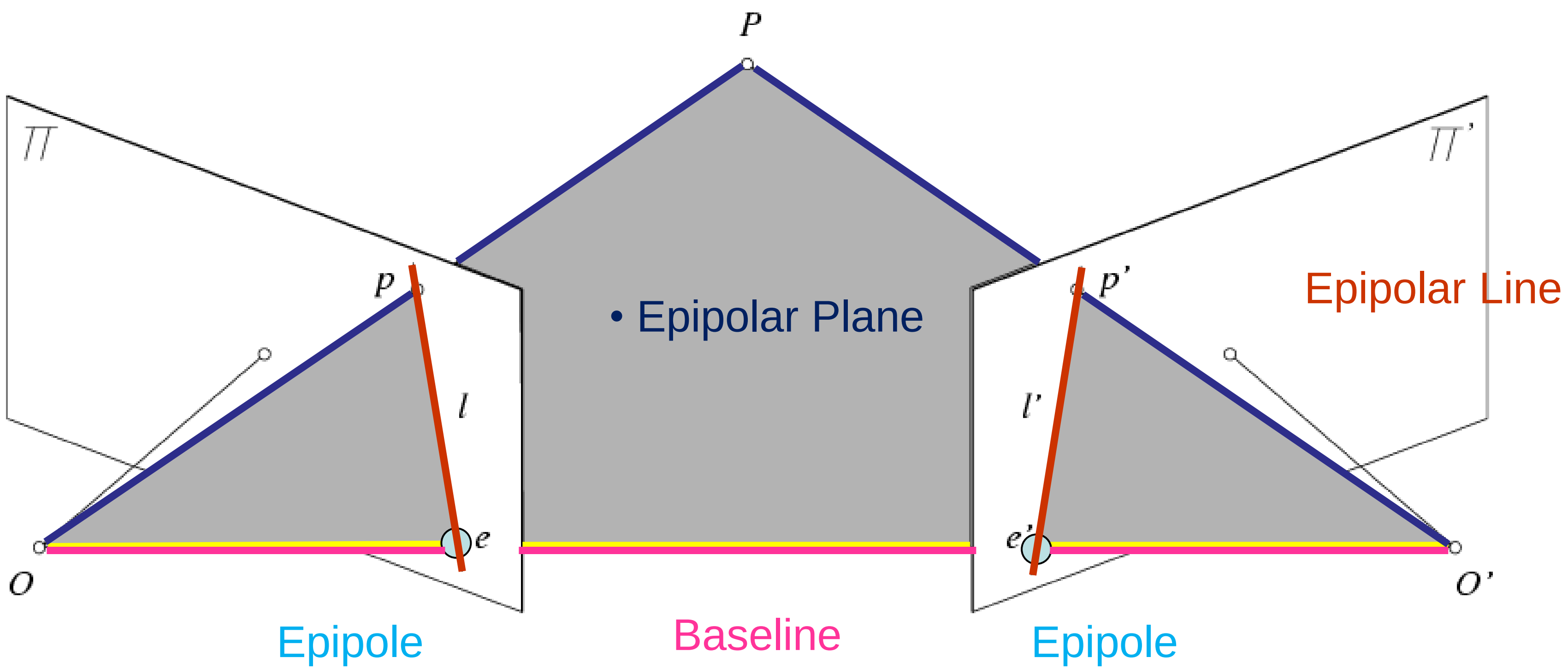
Epipolar constraint



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



http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

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

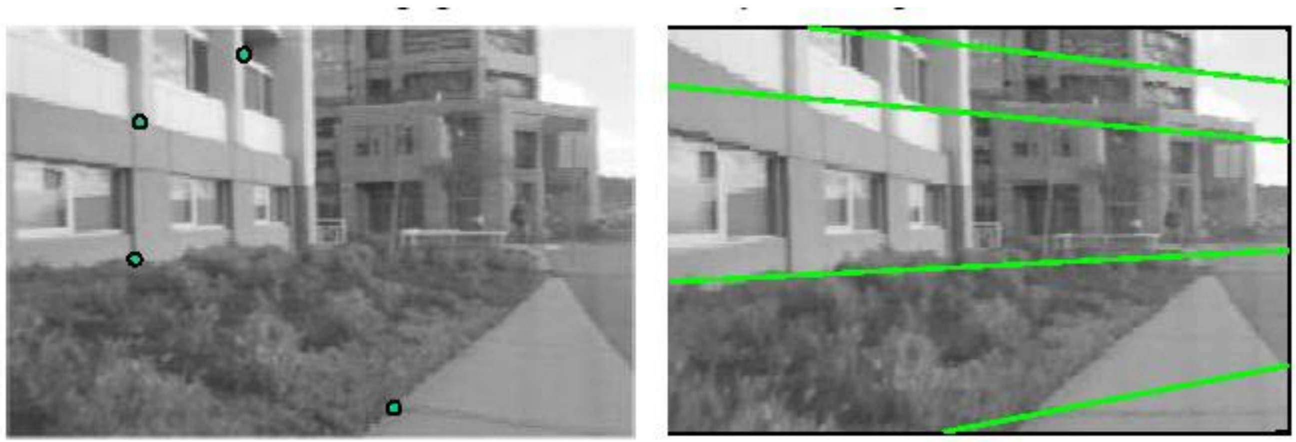
Why is the epipolar constraint useful?

Epipolar constraint

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

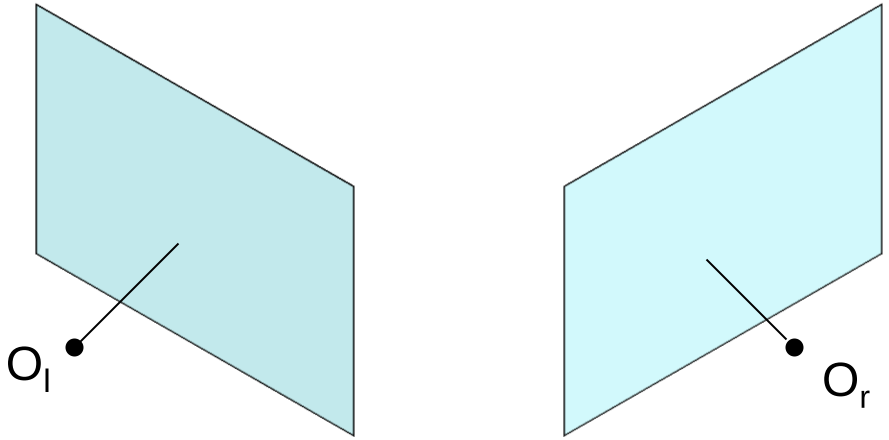
Image from Andrew Zisserman

Example

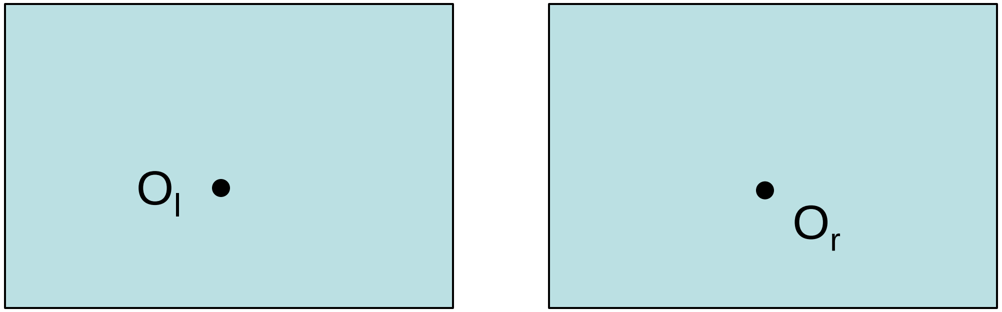


What do the epipolar lines look like?

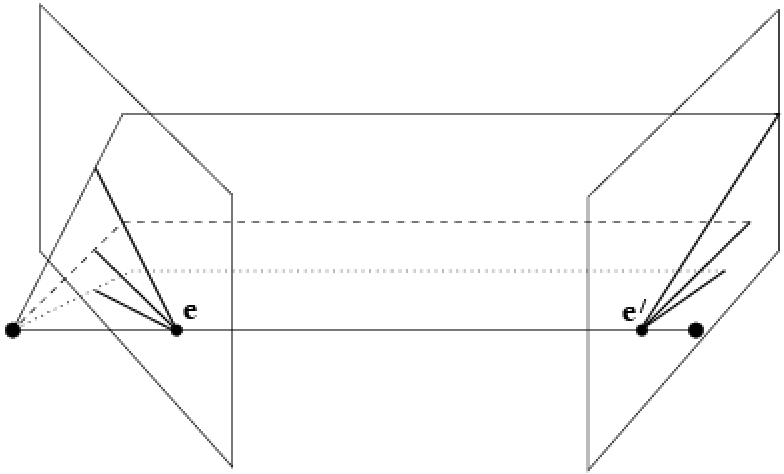
1.



2.



Example: converging cameras



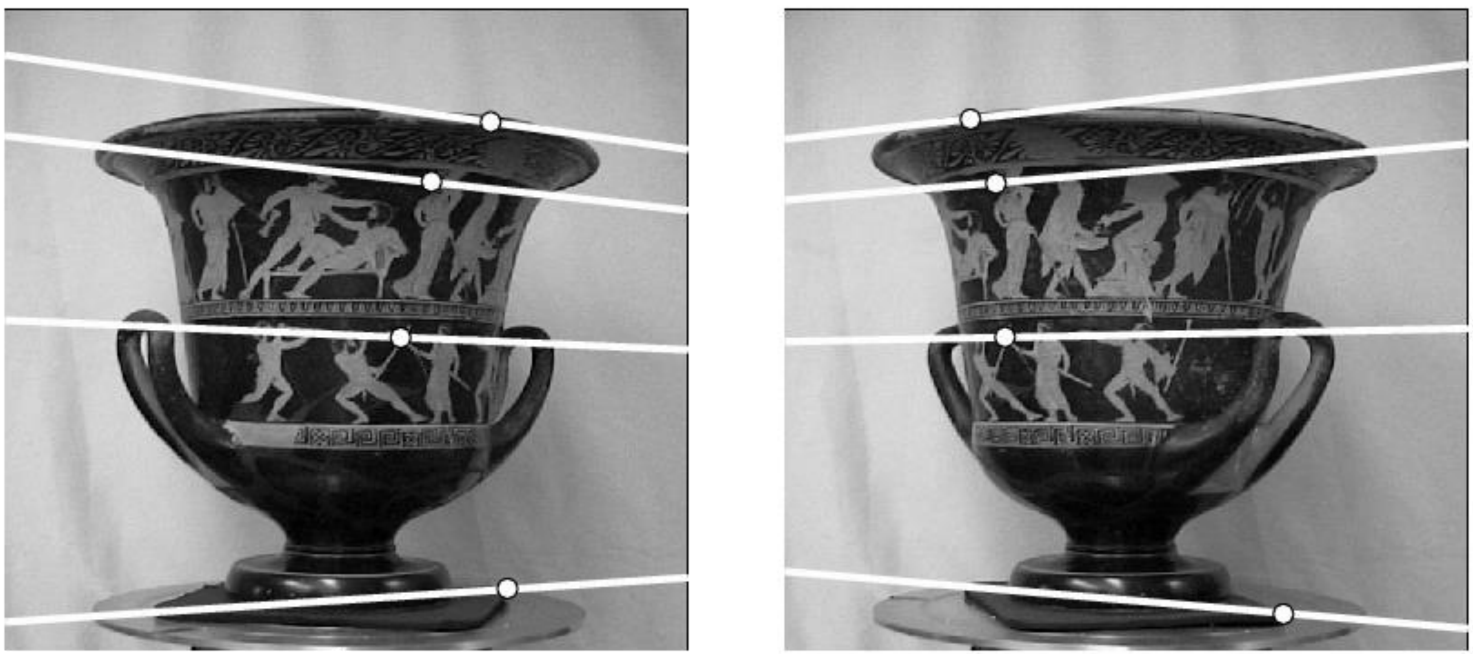
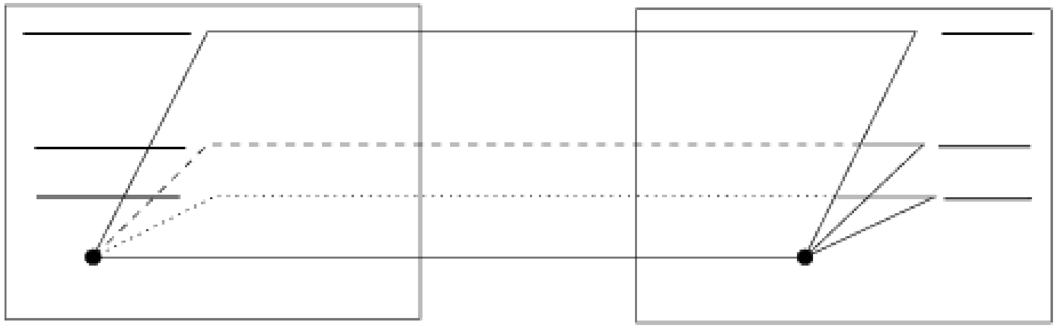


Figure from Hartley & Zisserman

Example: parallel cameras



Where are the epipoles?

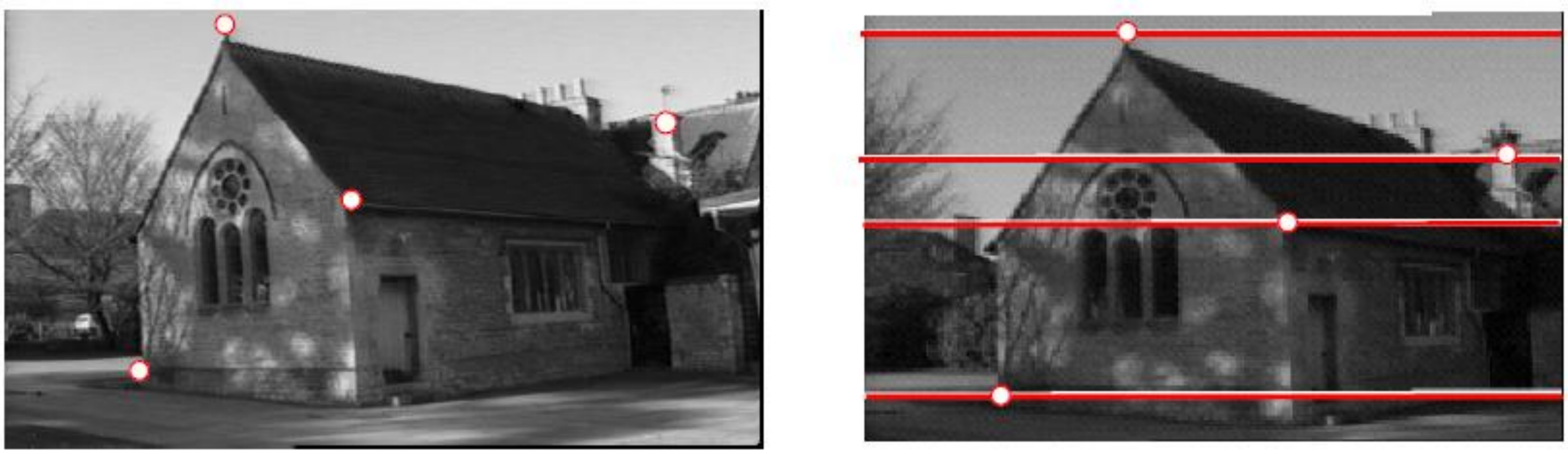
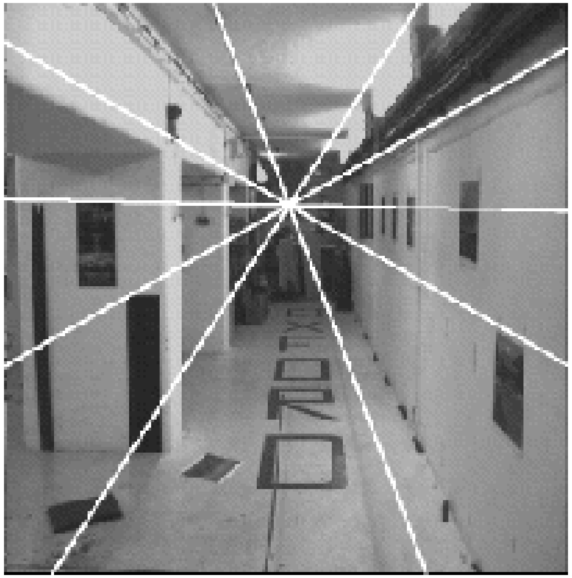


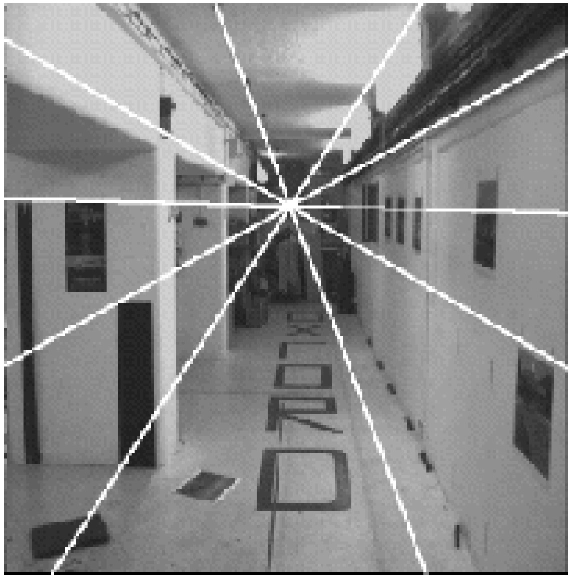
Figure from Hartley & Zisserman

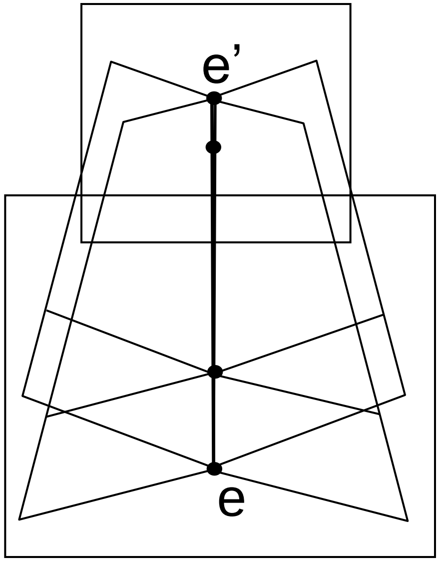
Example: Forward motion

What would the epipolar lines look like if the camera moves directly forward?

Example: Forward motion



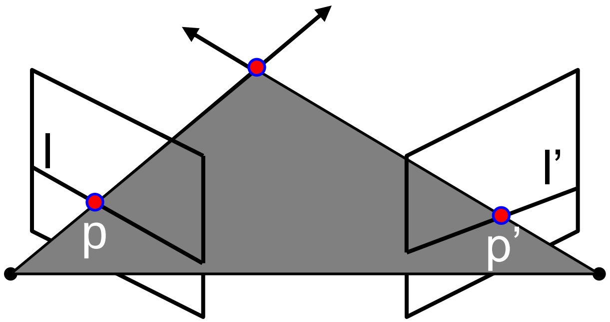




Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Fundamental matrix

Let *p* be a point in left image, *p'* in right image



Epipolar relation

- *p* maps to epipolar line *l'*
- *p'* maps to epipolar line *l*

Epipolar mapping described by a 3x3 matrix *F*

$$l' = Fp$$

$$l = p'F$$

It follows that

$$p'Fp = 0$$

Fundamental matrix

This matrix F is called

- the "Essential Matrix"
  - when image intrinsic parameters are known
- the "Fundamental Matrix"
  - more generally (uncalibrated case)

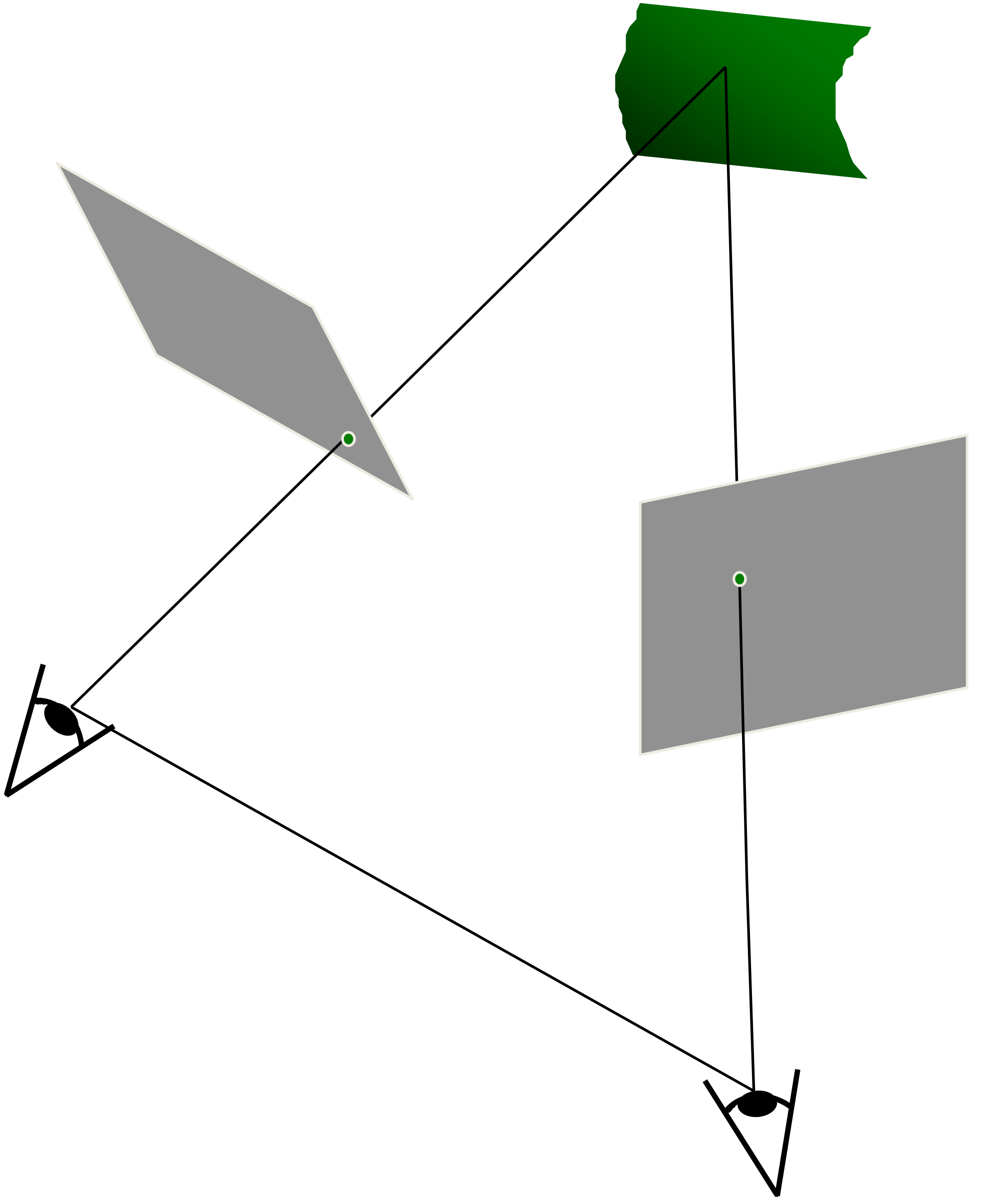
Can solve for F from point correspondences

- Each (p, p') pair gives one linear equation in entries of F

$$p' F p = 0$$

- F has 9 entries, but really only 7 or 8 degrees of freedom.
- With 8 points it is simple to solve for F, but it is also possible with 7. See Marc Pollefey's notes for a nice tutorial

Stereo image rectification



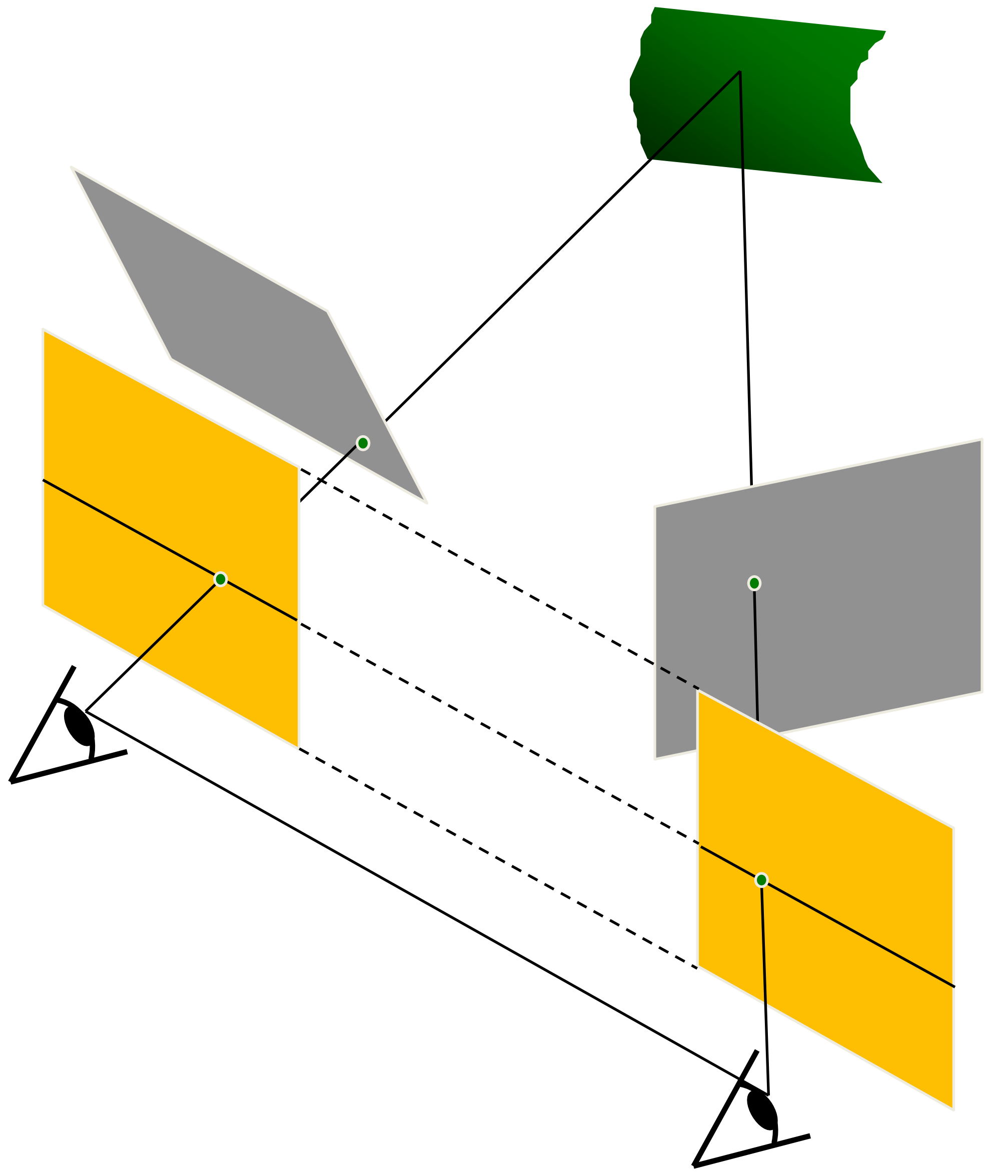
Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between camera centers

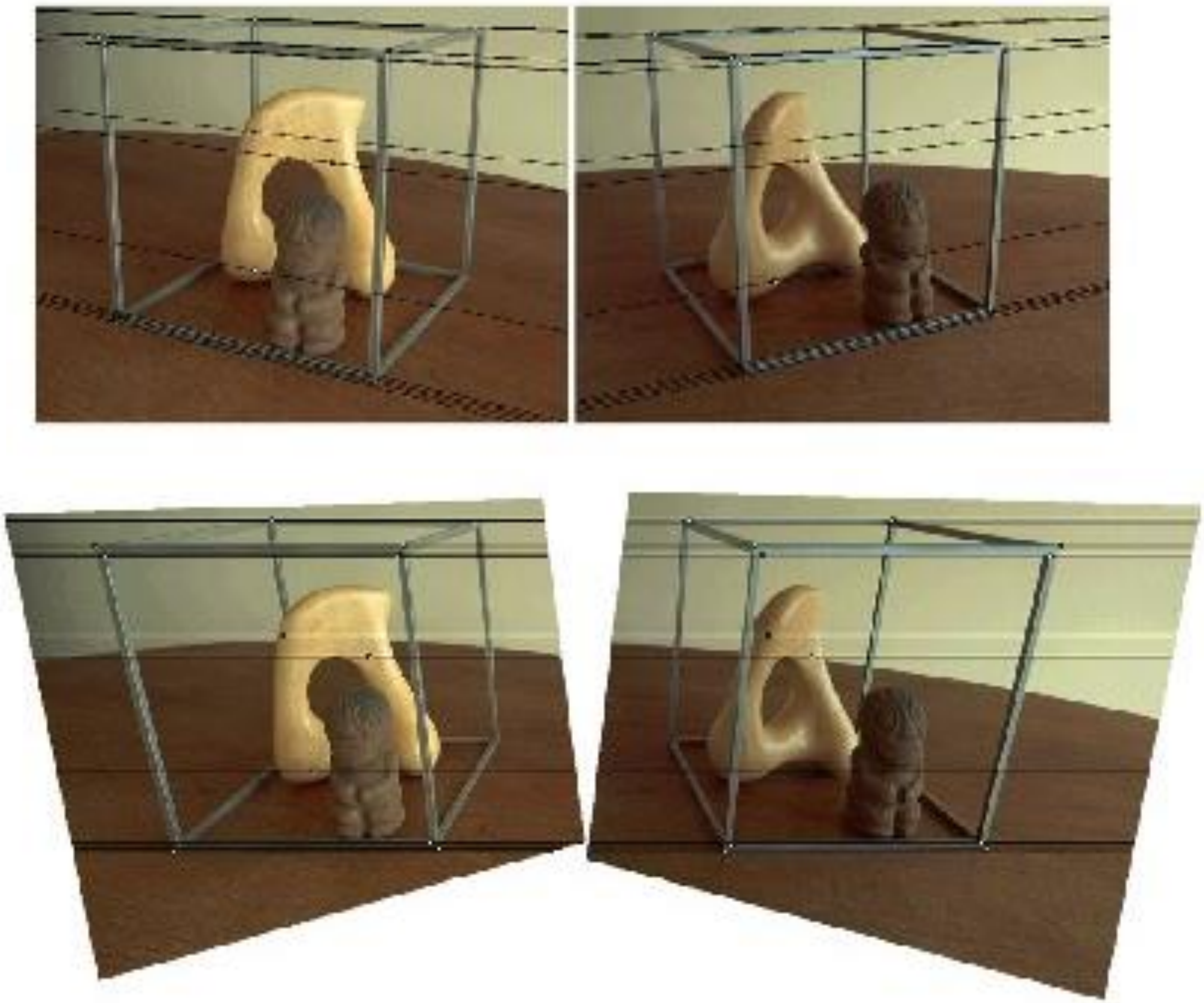
- Pixel motion is horizontal after this transformation

- Two homographies (3x3 transform), one for each input image reprojection

➢ C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.



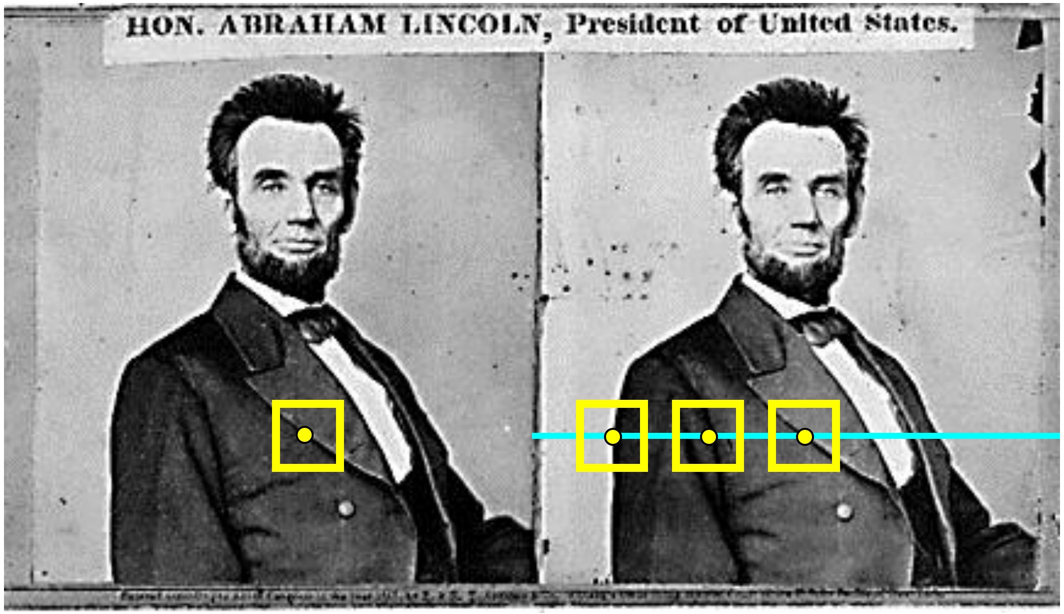
Rectification example



The correspondence problem

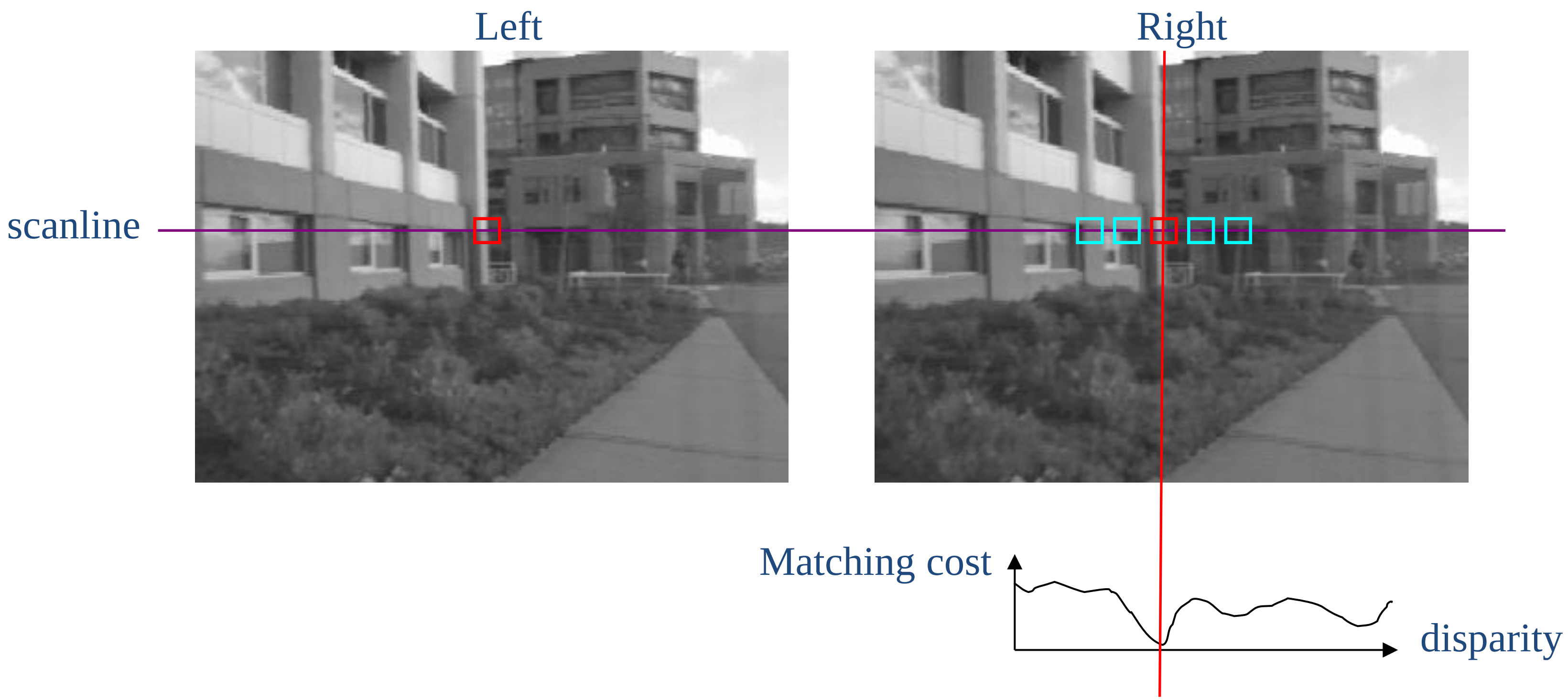
- Epipolar geometry constrains our search, but we still have a difficult correspondence problem.

Basic stereo matching algorithm



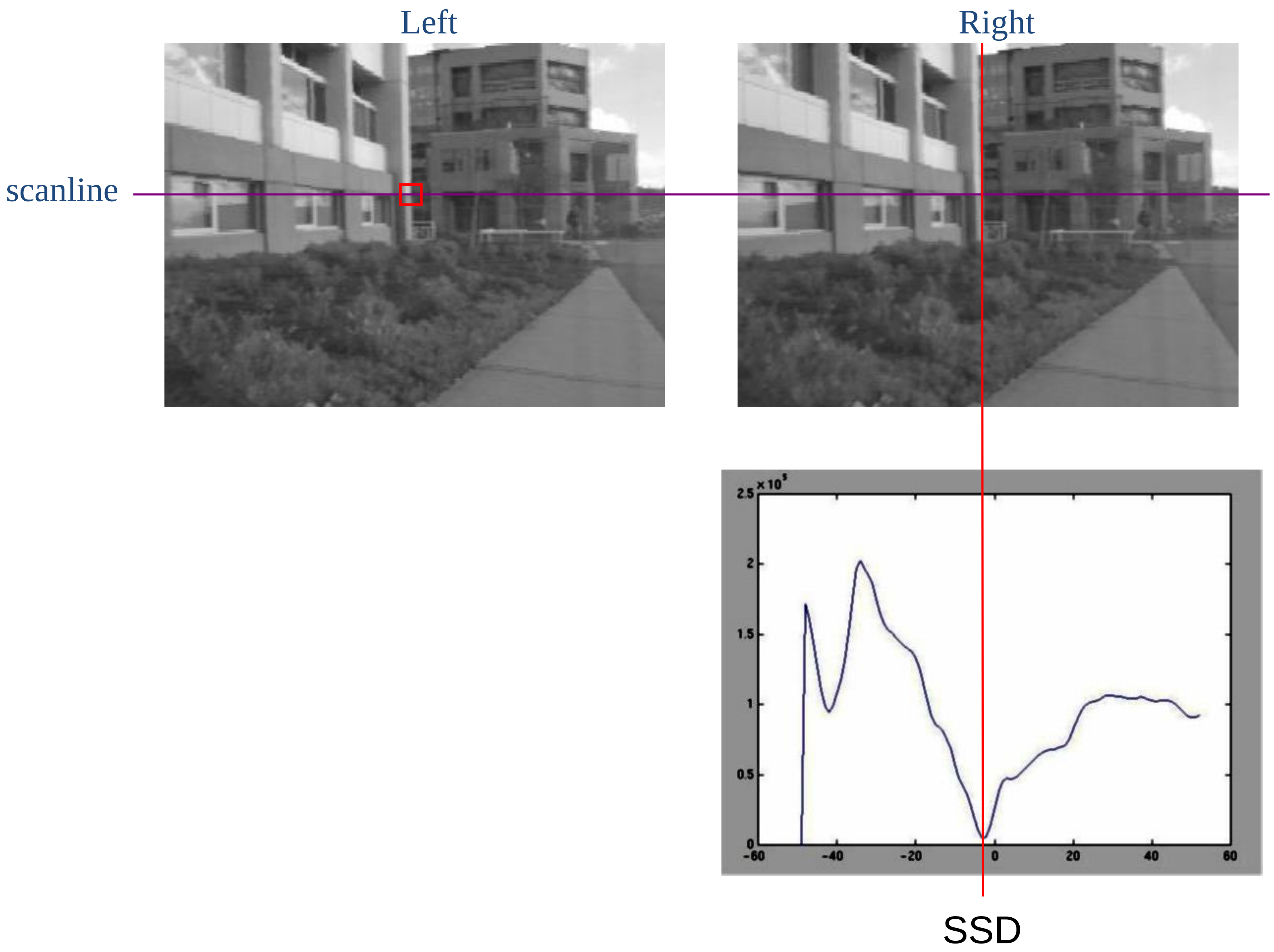
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Examine all pixels on the scanline and pick the best match x'
  - Compute disparity x-x' and set depth(x) = fB/(x-x')

Correspondence search



- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search

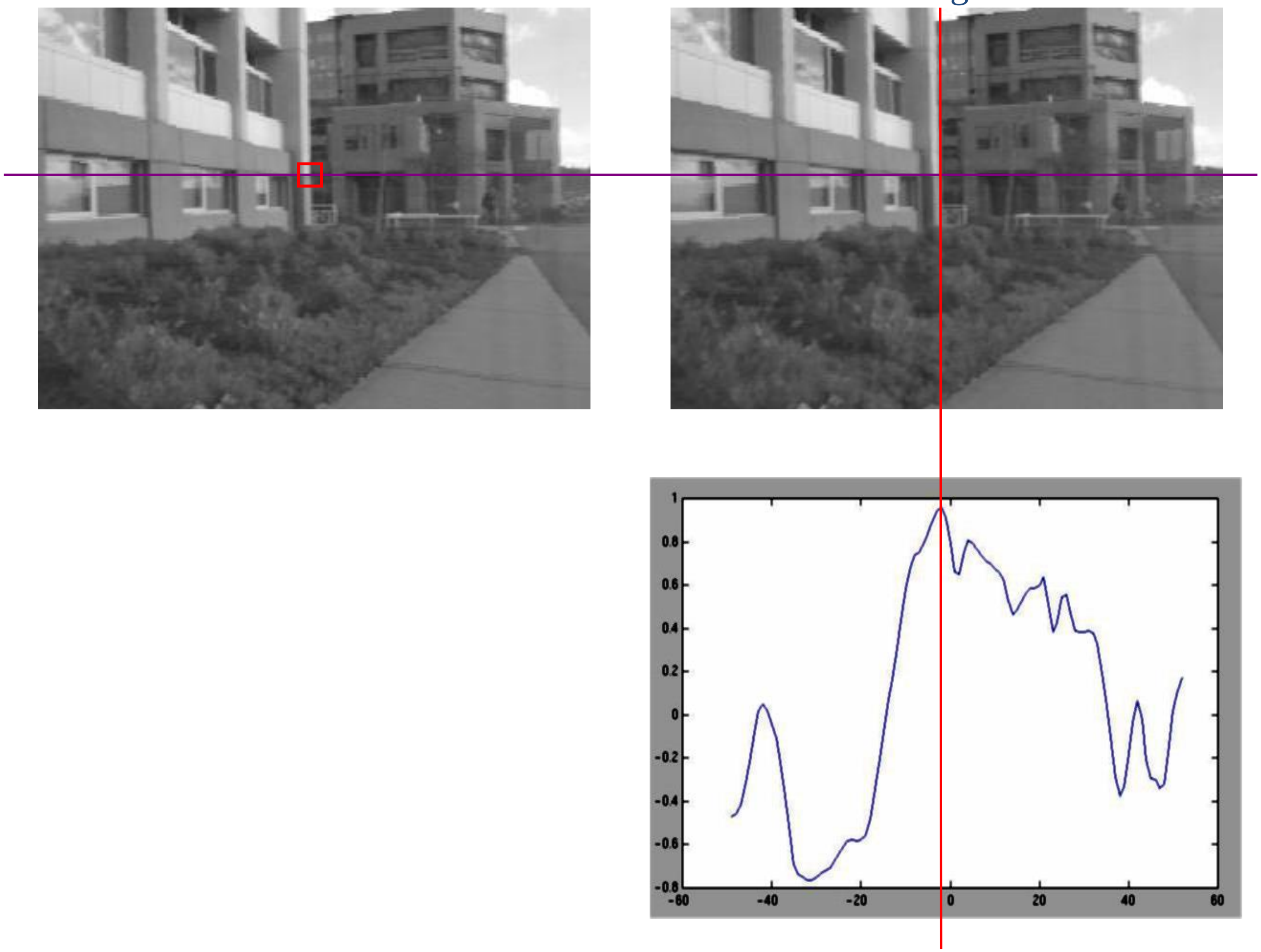


Correspondence search

Left

Right

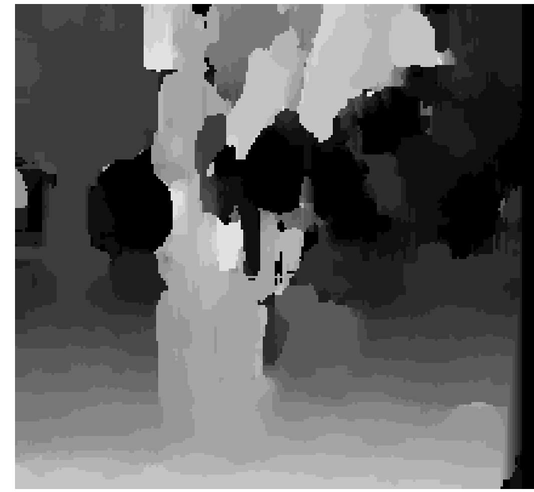
scanline



Norm. corr

Effect of window size

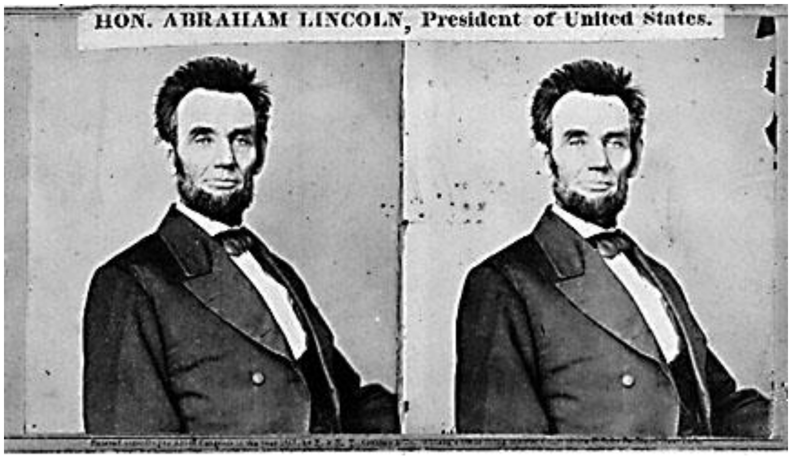
W = 3



W = 20

- Smaller window
  - \+ More detail
  - – More noise
- Larger window
  - \+ Smoother disparity maps
  - – Less detail

Failures of correspondence search



Textureless surfaces

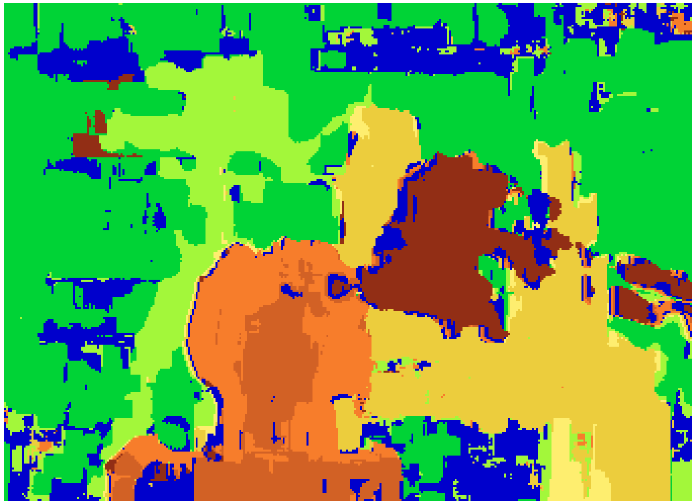
Occlusions, repetition

Non-Lambertian surfaces, specularities

Results with window search

Data

Window-based matching



Ground truth

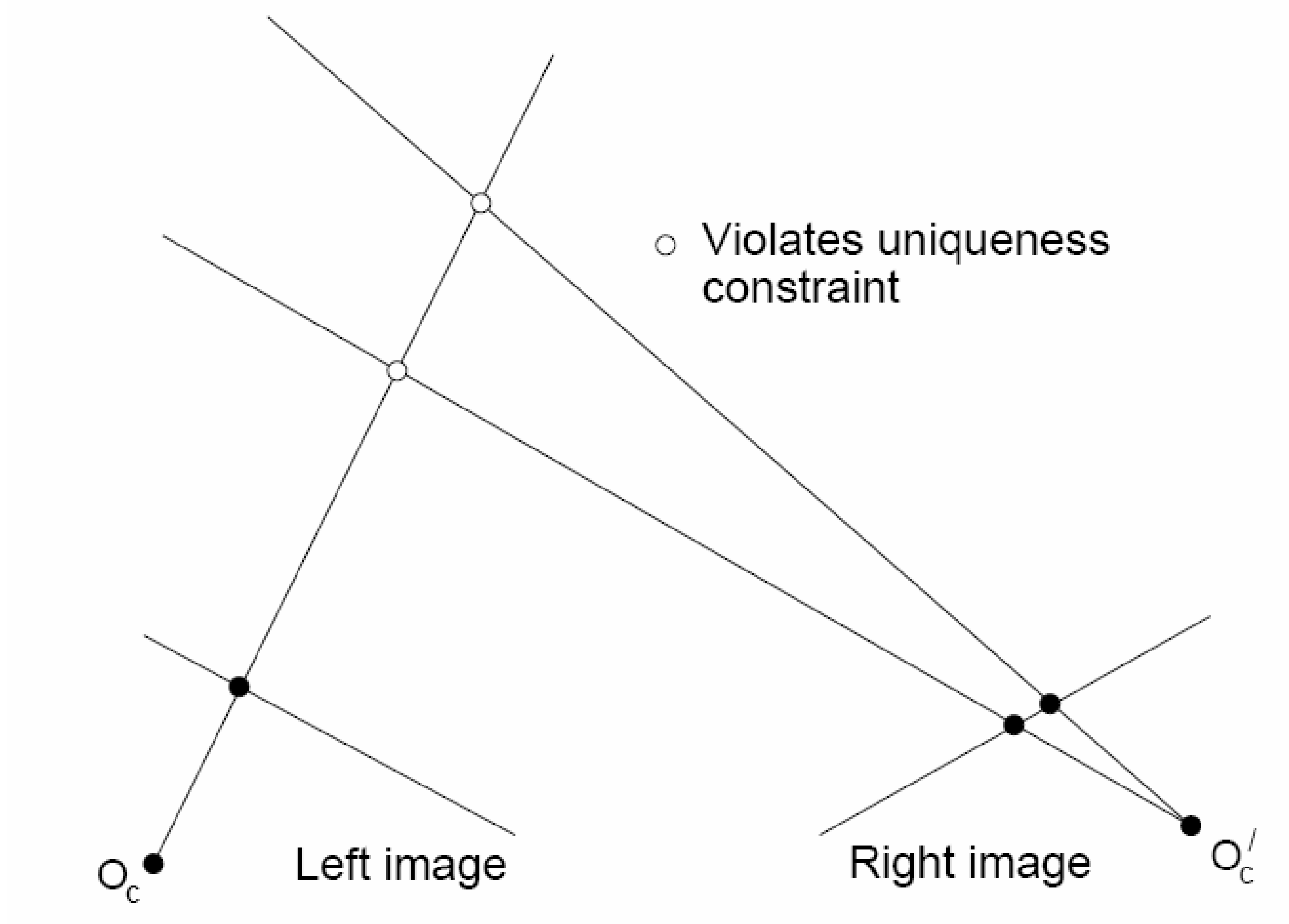
How can we improve window-based matching?

- So far, matches are independent for each point

- What constraints or priors can we add?

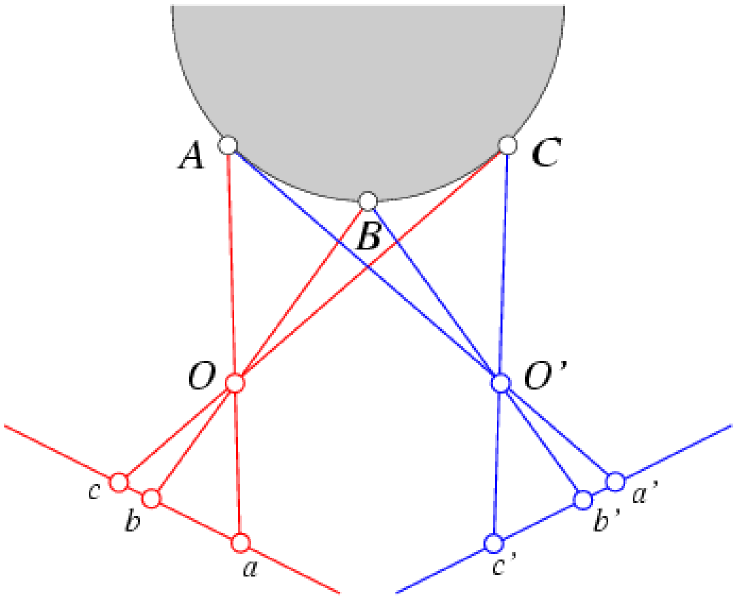
Stereo constraints/priors

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image



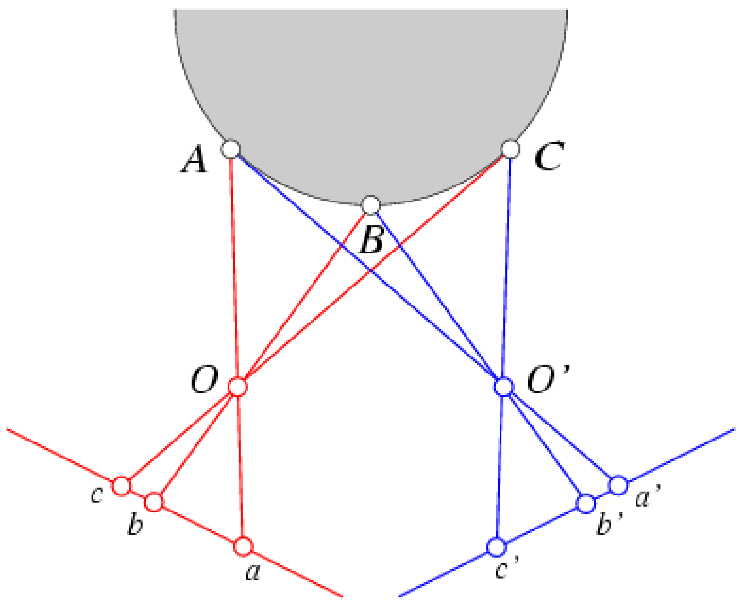
Stereo constraints/priors

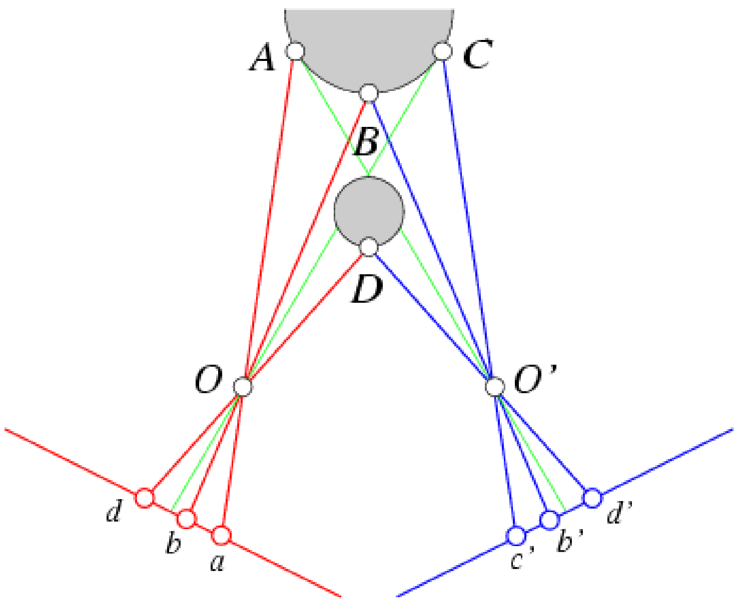
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



Stereo constraints/priors

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views





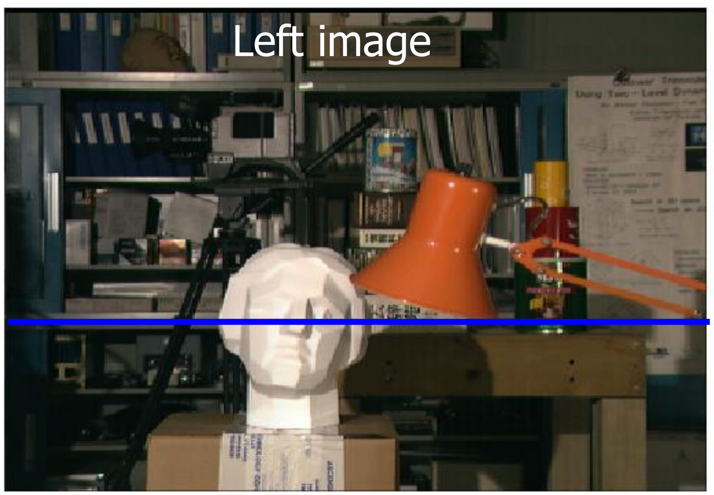
Ordering constraint doesn't hold

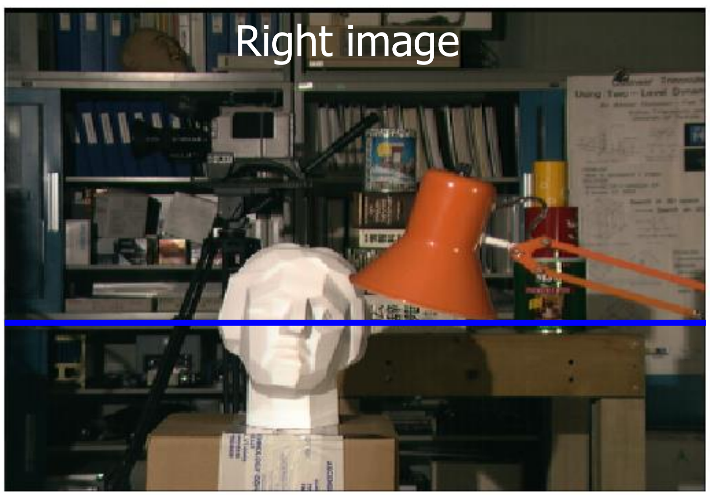
Priors and constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views
- Smoothness
  - We expect disparity values to change slowly (for the most part)

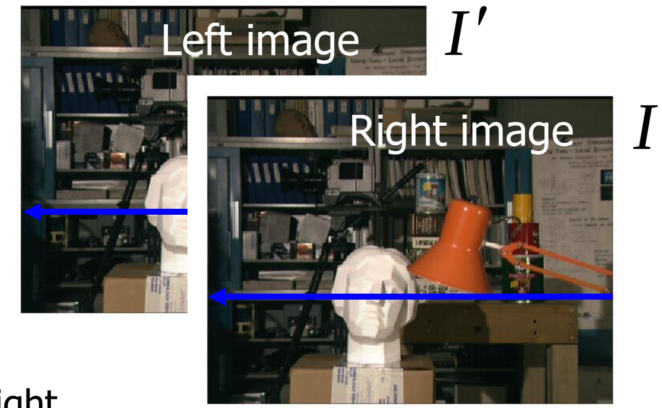
Scanline stereo

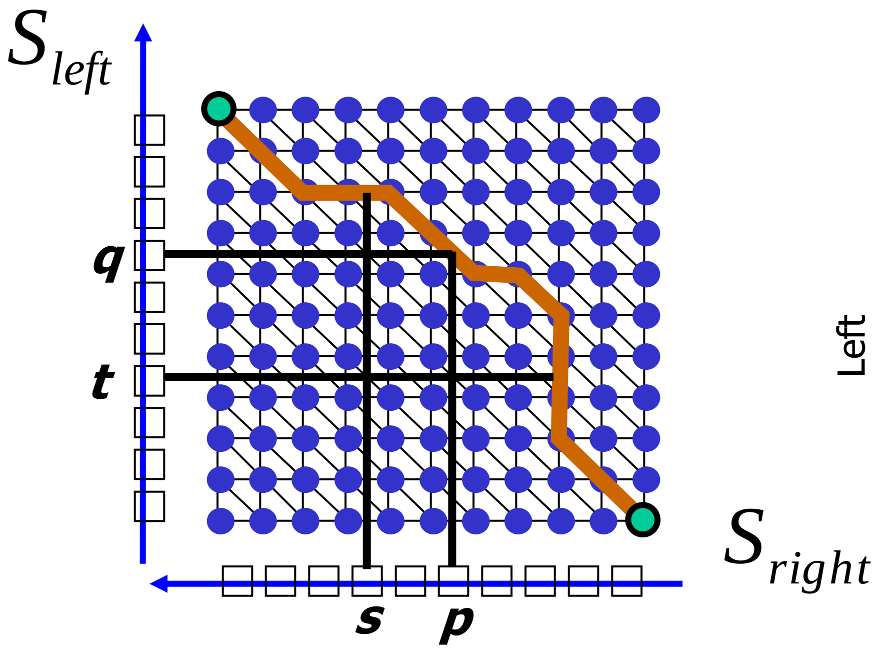
- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

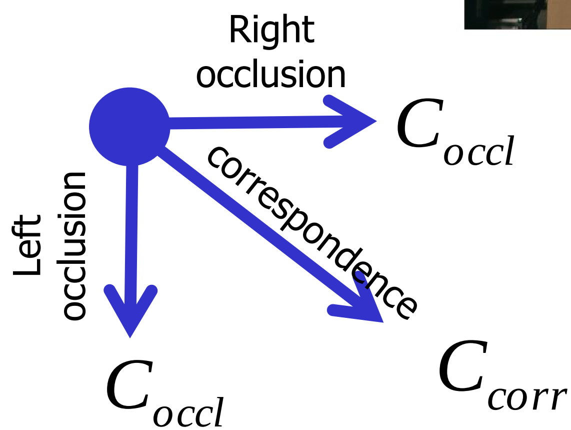




“Shortest paths” for scan-line stereo







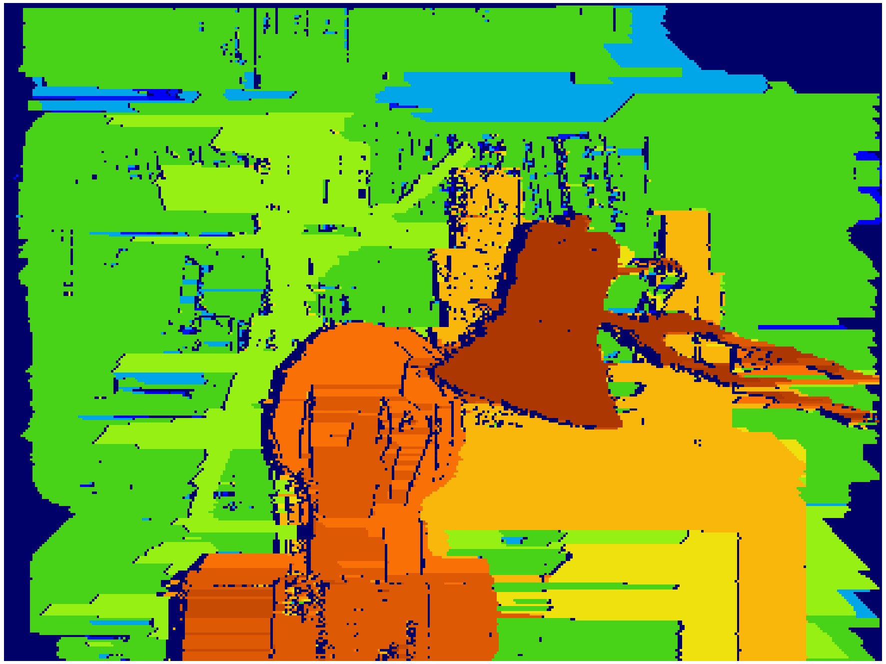
Can be implemented with dynamic programming

Ohta & Kanade ’85, Cox et al. ‘96

Slide credit: Y. Boykov

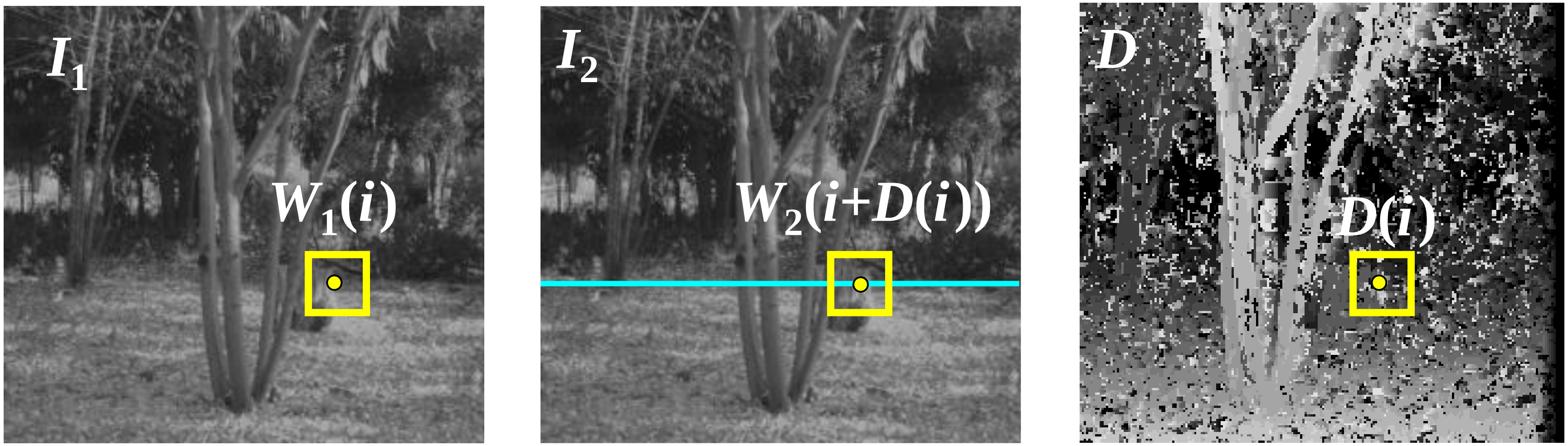
Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts



- Can’t use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

Stereo matching as energy minimization (random field interpretation)



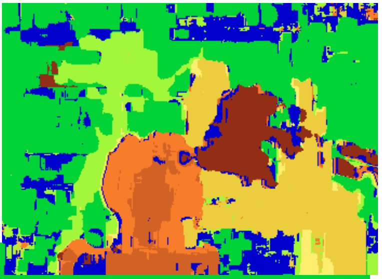
$$E(D) = \underbrace{\sum_i \big(W_1(i) - W_2(i + D(i))\big)^2}_{\textit{data term}} + \lambda \underbrace{\sum_{\text{neighbors } i,j} \rho\big(D(i) - D(j)\big)}_{\textit{smoothness term}}$$

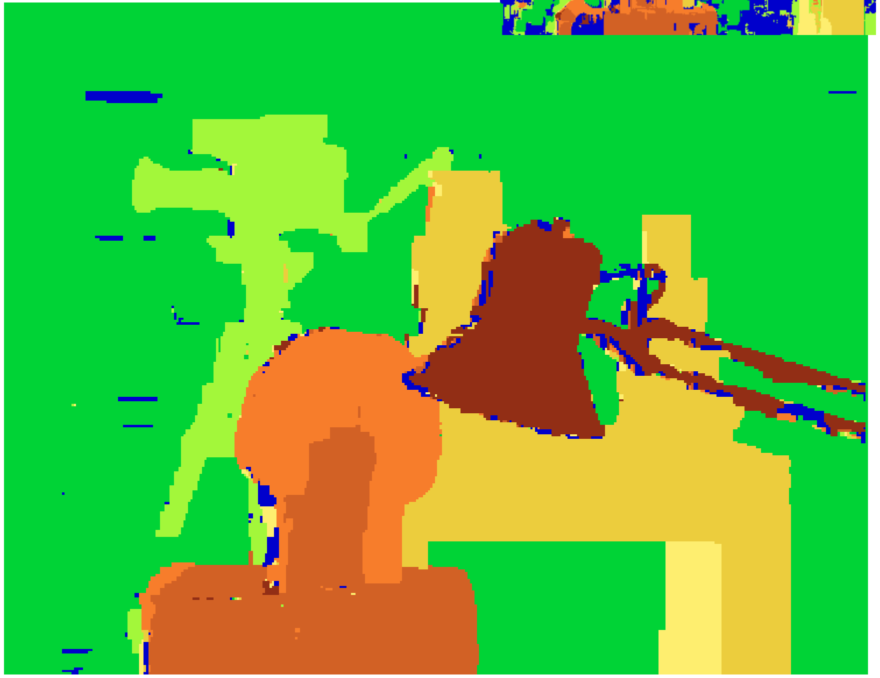
- Energy functions of this form can be minimized using *graph cuts*

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

Many of these constraints can be encoded in an energy function and solved using graph cuts

Before





Graph cuts

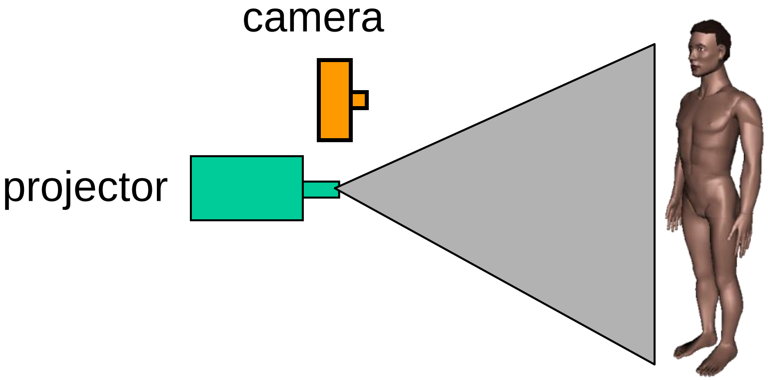
Ground truth

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

For the latest and greatest: http://www.middlebury.edu/stereo/

Active stereo with structured light

- Project “structured” light patterns onto the object
  - Simplifies the correspondence problem
  - Allows us to use only one camera

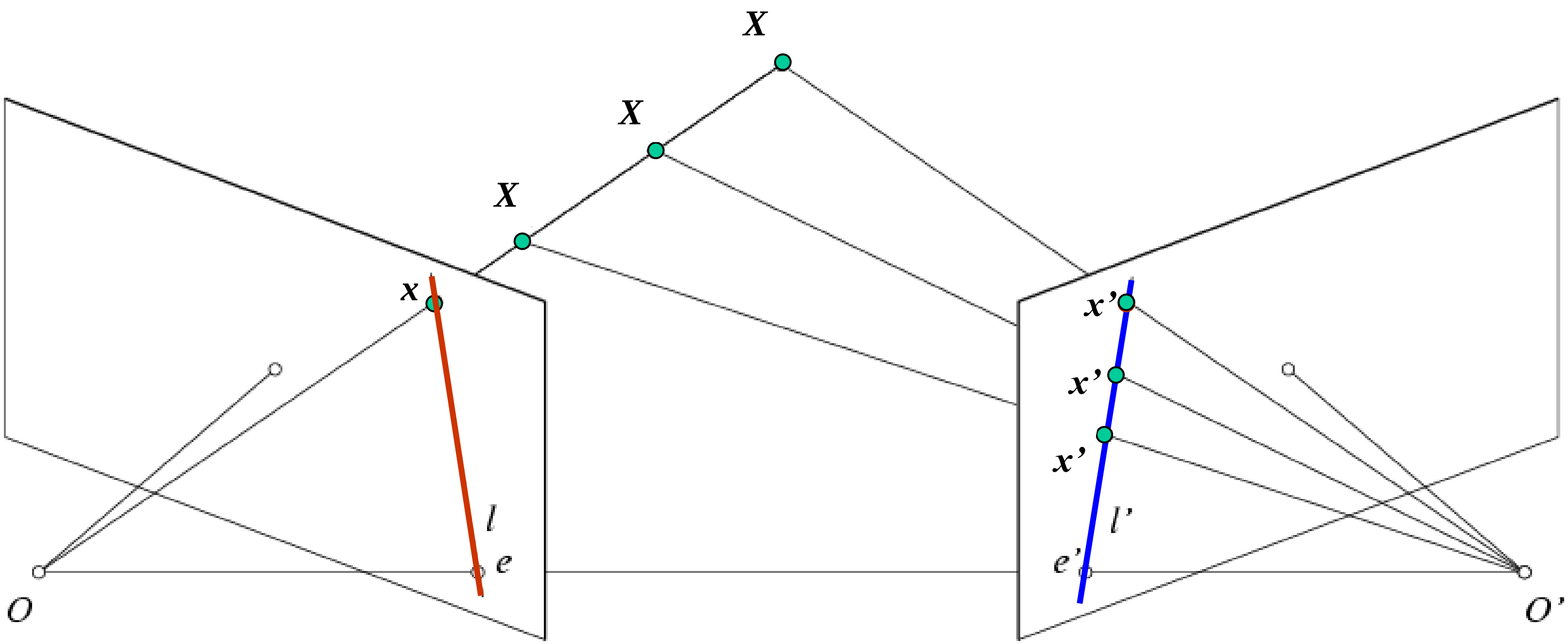


L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming.](#) *3DPVT* 2002

Kinect: Structured infrared light

http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/

Summary: Key idea: Epipolar constraint



Potential matches for *x* have to lie on the corresponding line *l'*.

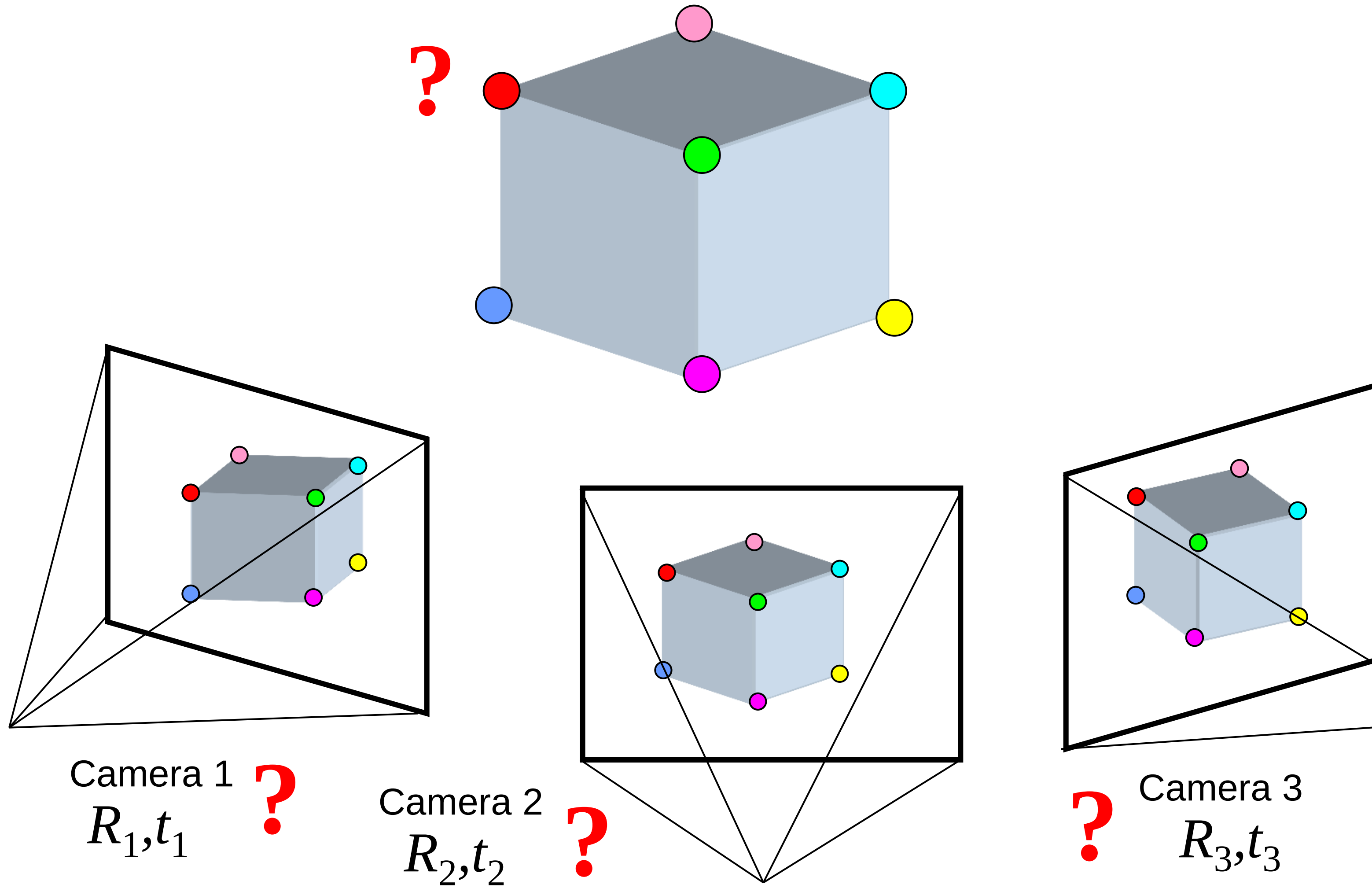
Potential matches for *x'* have to lie on the corresponding line *l*.

Summary

- Epipolar geometry
  - Epipoles are intersection of baseline with image planes
  - Matching point in second image is on a line passing through its epipole
  - Fundamental matrix maps from a point in one image to a line (its epipolar line) in the other
  - Can solve for F given corresponding points (e.g., interest points)
- Stereo depth estimation
  - Estimate disparity by finding corresponding points along scanlines
  - Depth is inverse to disparity

Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Slide credit: Noah Snavely

Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\,\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

How do we know the scale of image content?


Photography
Your Eyes Deserve It
TOTAL
RECLAIMS
TOTAL
RECLAIMS
TOTAL
RECLAIMS

Structure from motion ambiguity

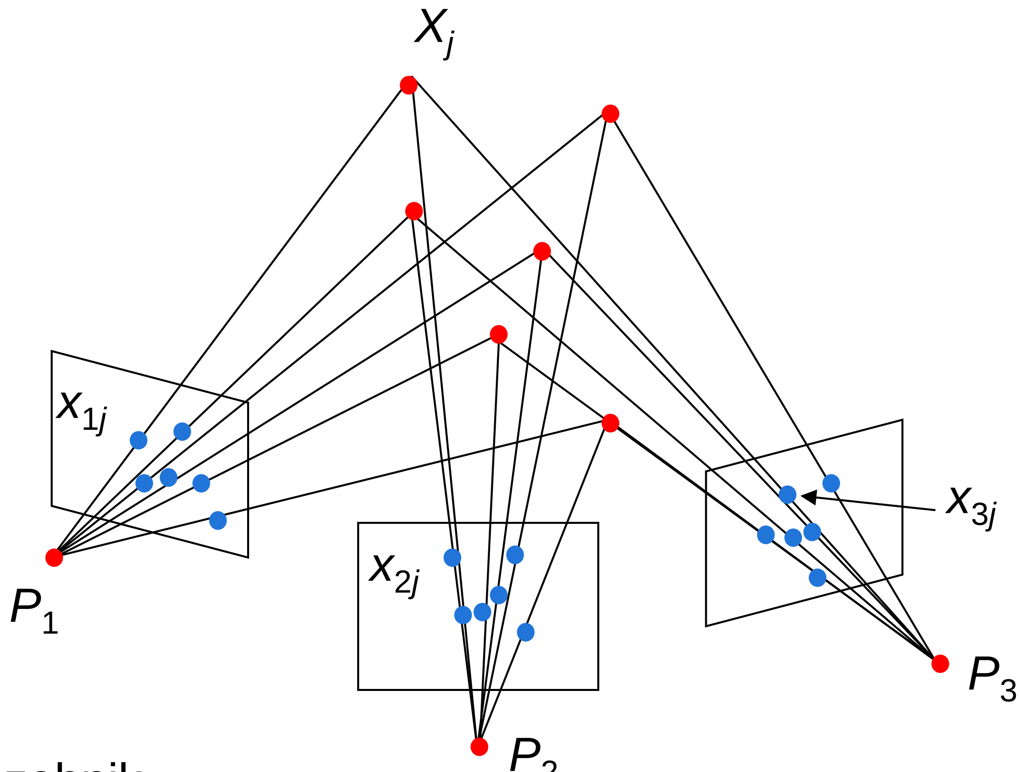
- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same

- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, then the images do not change

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

Projective structure from motion

- Given: *m* images of *n* fixed 3D points
  - $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j\,,\; i = 1,\dots,m, \quad j = 1,\dots,n$
- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* corresponding points $\mathbf{x}_{ij}$

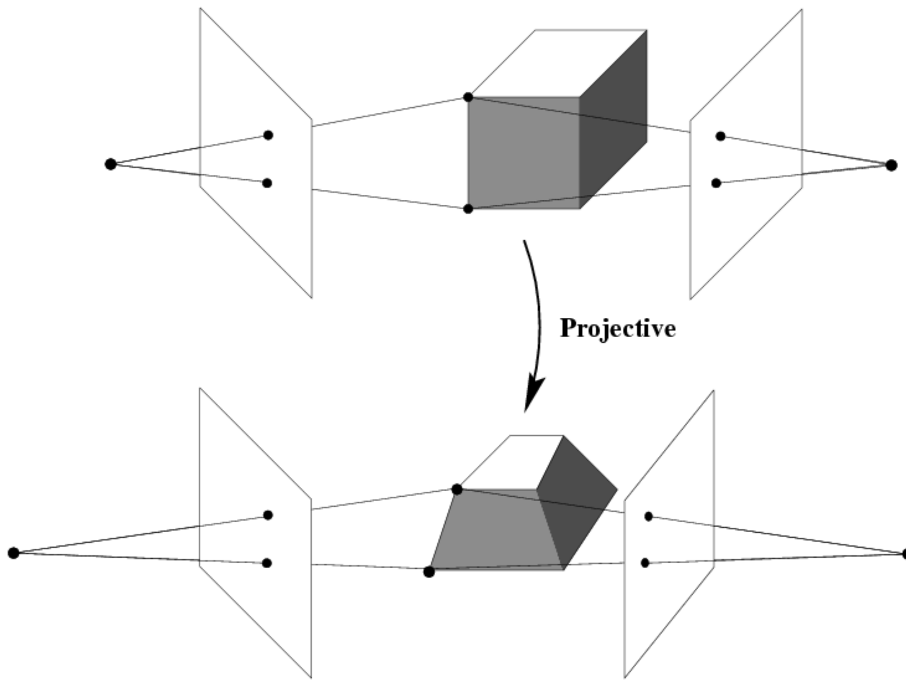


Slides from Lana Lazebnik

Projective structure from motion

- Given: *m* images of *n* fixed 3D points
  - $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* corresponding points $\mathbf{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation **Q**:
  - $\mathbf{X} \rightarrow \mathbf{QX}, \mathbf{P} \rightarrow \mathbf{PQ}^{-1}$
- We can solve for structure and motion when
  - $2mn >= 11m + 3n - 15$
- For two cameras, at least 7 points are needed

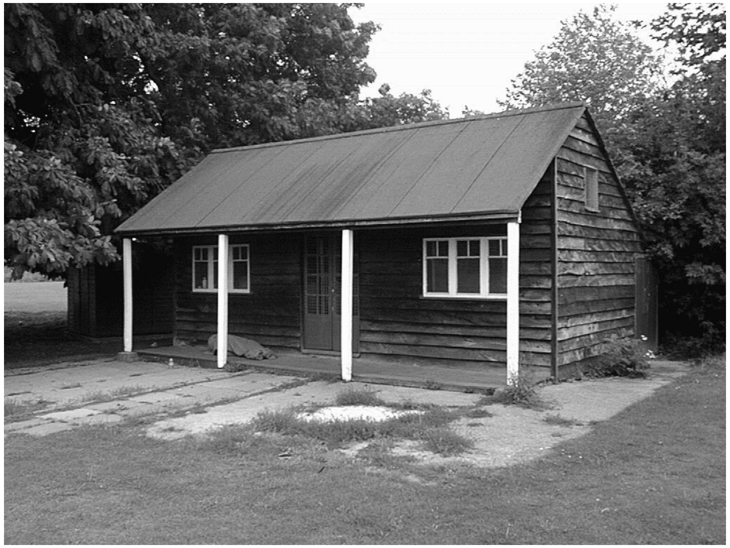
Projective ambiguity

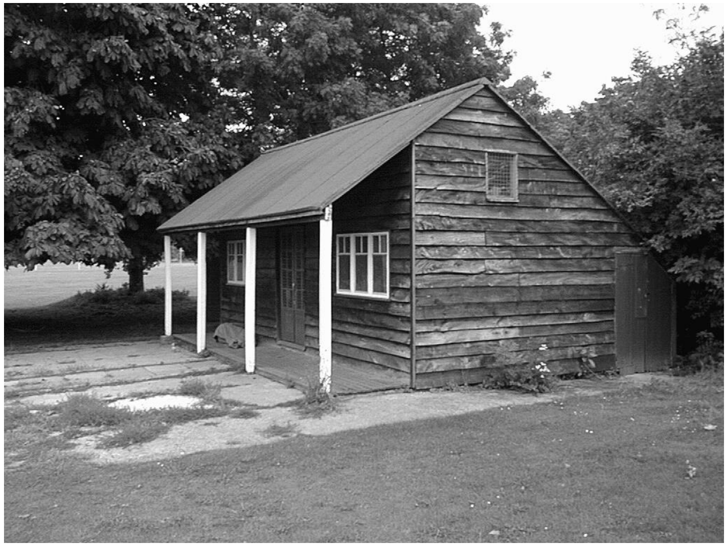


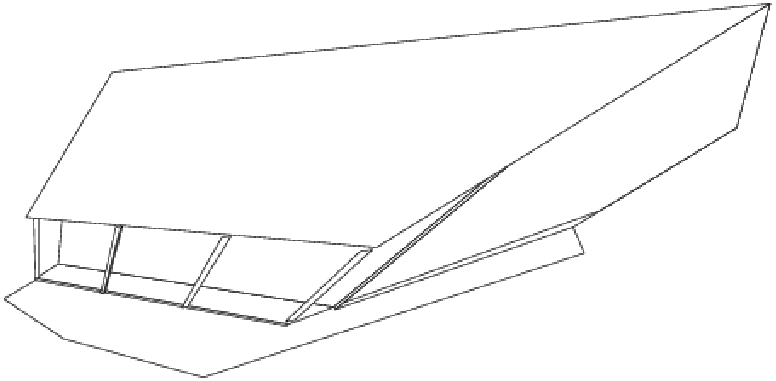
$$\mathbf{Q_p} = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^{\mathsf{T}} & v \end{bmatrix}$$

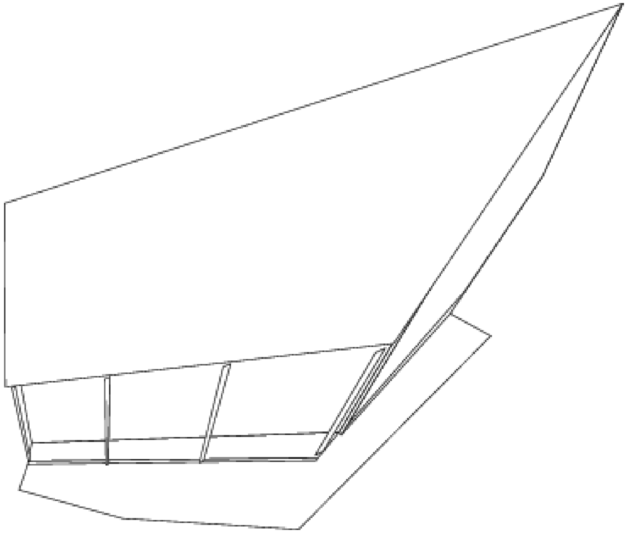
$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{P}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{P}}\,\mathbf{X}\right)$$

Projective ambiguity









Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

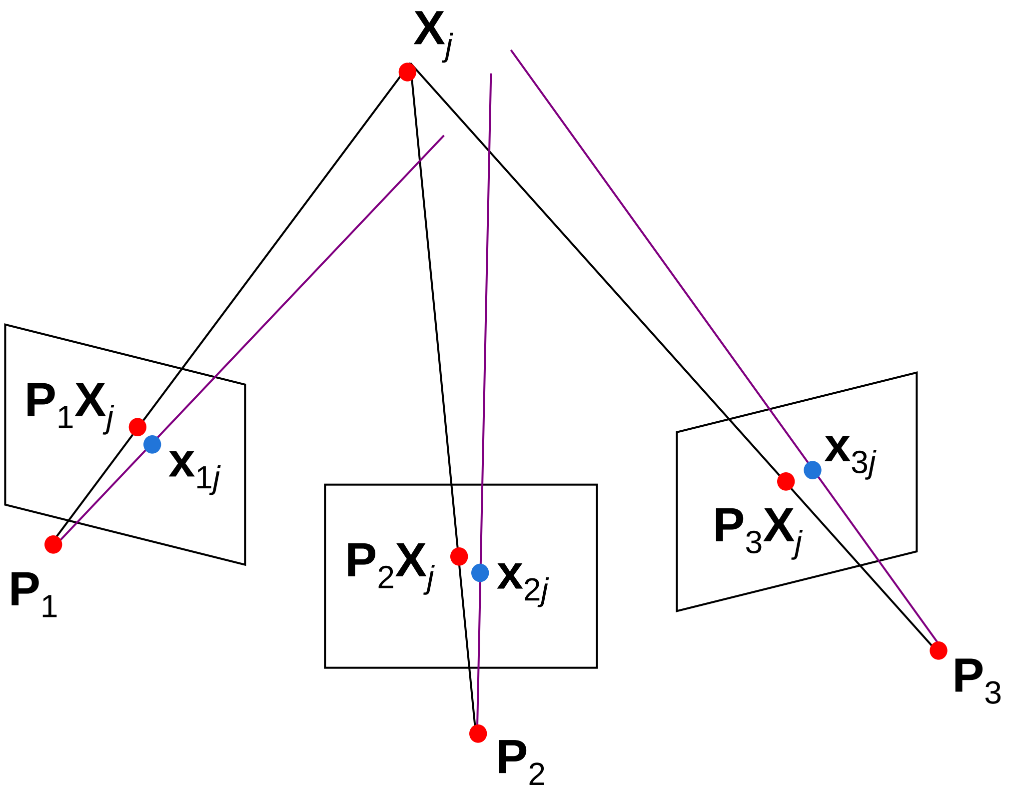


Photo synth

Noah Snavely, Steven M. Seitz, Richard Szeliski, "[Photo tourism: Exploring photo collections in 3D]," SIGGRAPH 2006

http://photosynth.net/