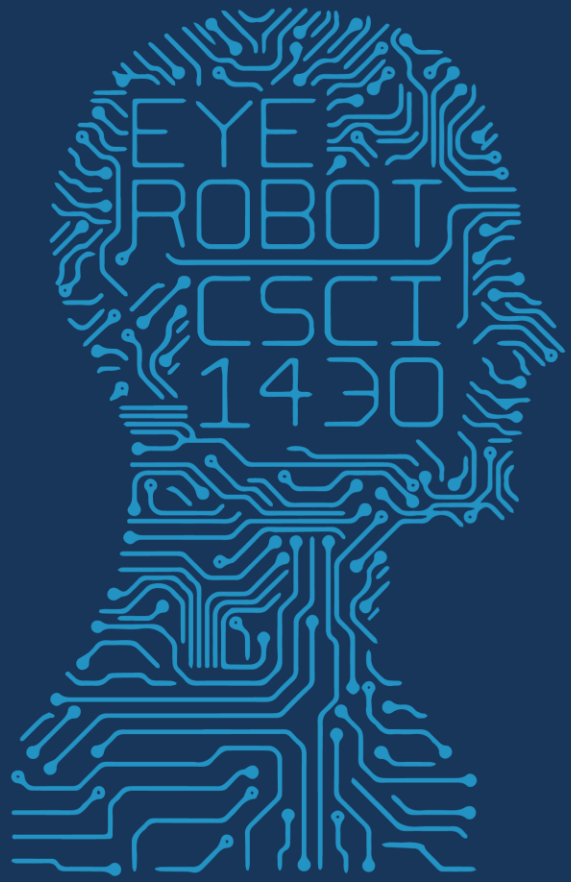
1950

Future Vision



15 April 2019

Computer Vision

Final project bits and pieces

The project is expected to take four weeks of time for up to four people.

At 12 hours per week per person that comes out to: ~192 hours of work for a four person team.

Capstone: “Do more.” An amount commensurate with the course’s extra credit, so another 2-4 hours per week per person.

Chaplin, Modern Times, 1936

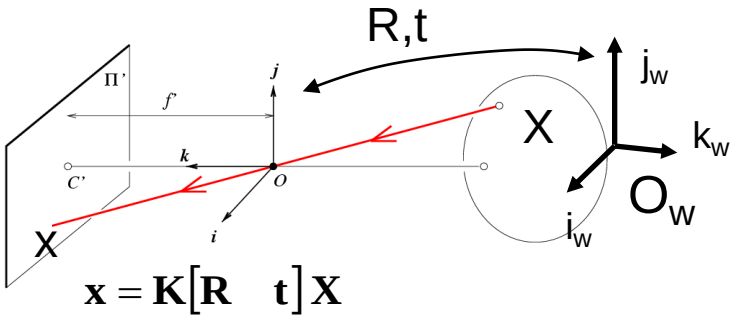
[A Bucket of Water and a Glass Matte: Special Effects in *Modern Times;* bonus feature on The Criterion Collection set]

Computer vision as world measurement

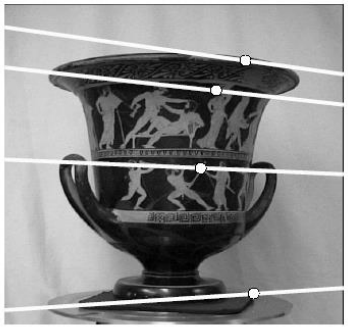
Two cameras, simultaneous views

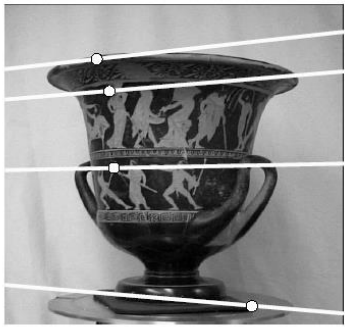
Single moving camera and static scene

Multiple view geometry



Camera calibration





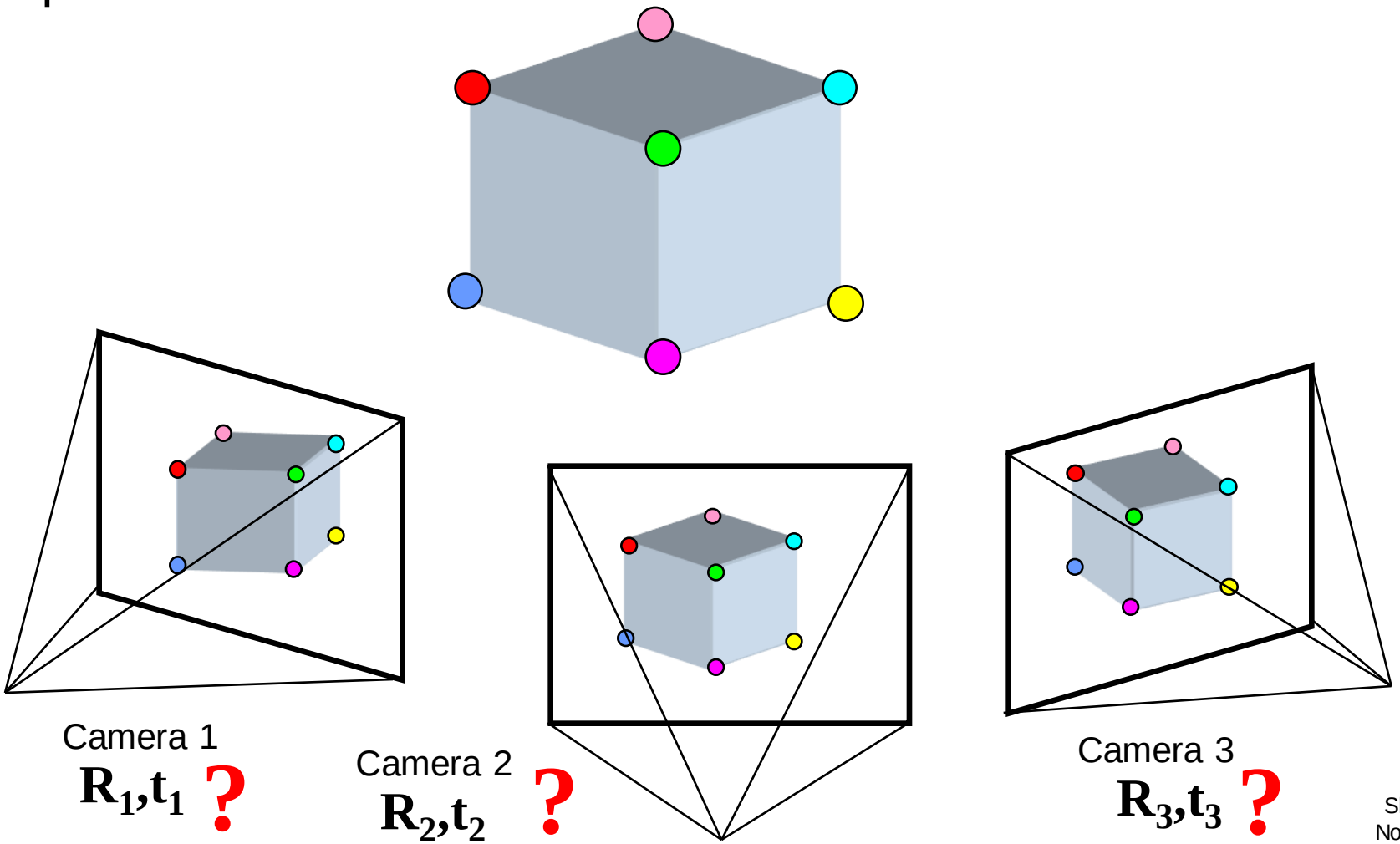
Hartley and Zisserman

Epipolar geometry

Dense depth map estimation

Multi-view geometry problems

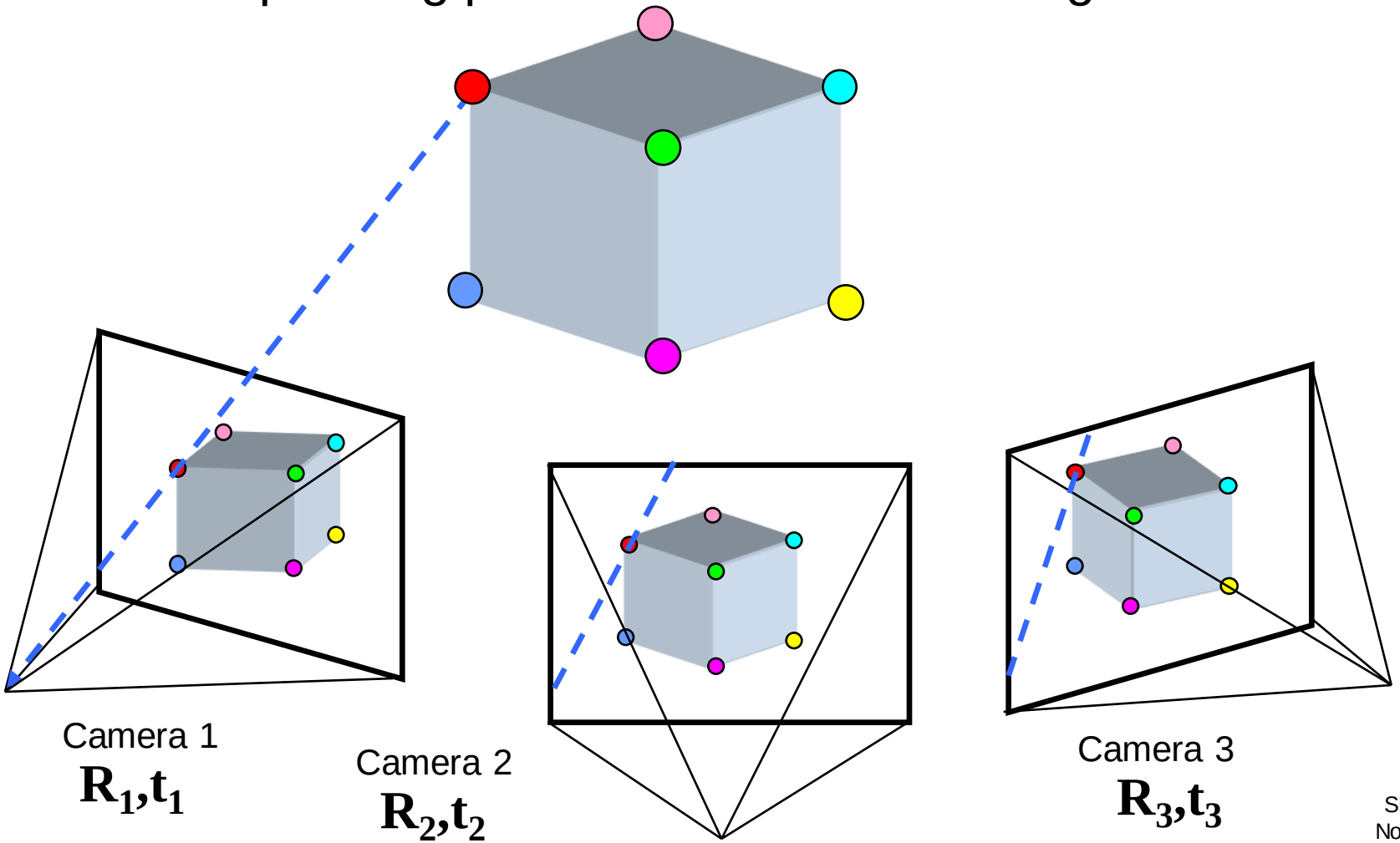
- **Camera 'Motion':** Given a set of corresponding 2D/3D points in two or more images, compute the camera parameters.



Slide credit: Noah Snavely

Multi-view geometry problems

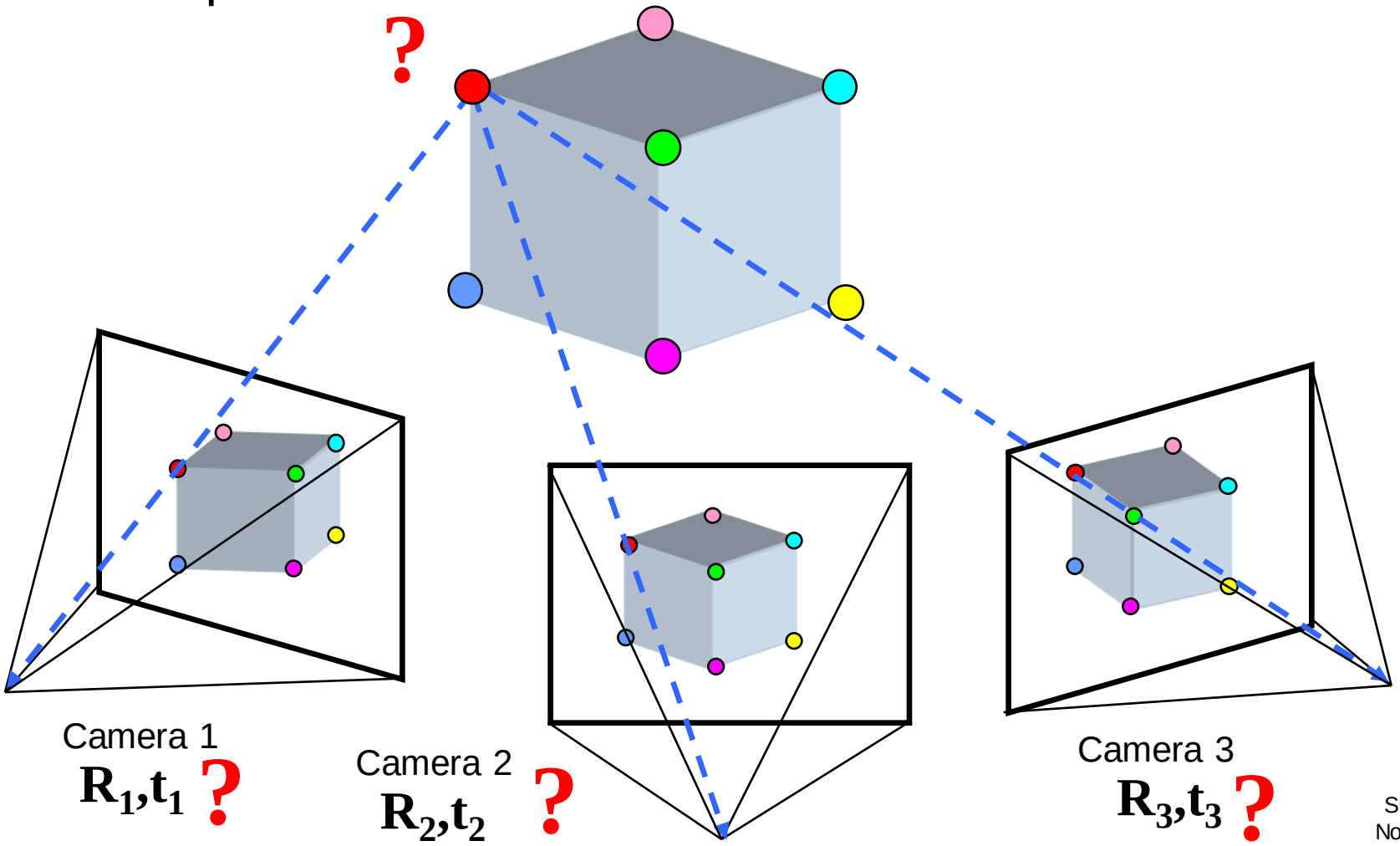
- **Stereo correspondence:** Given known camera parameters and a point in one of the images, where could its corresponding points be in the other images?



Slide credit: Noah Snavely

Multi-view geometry problems

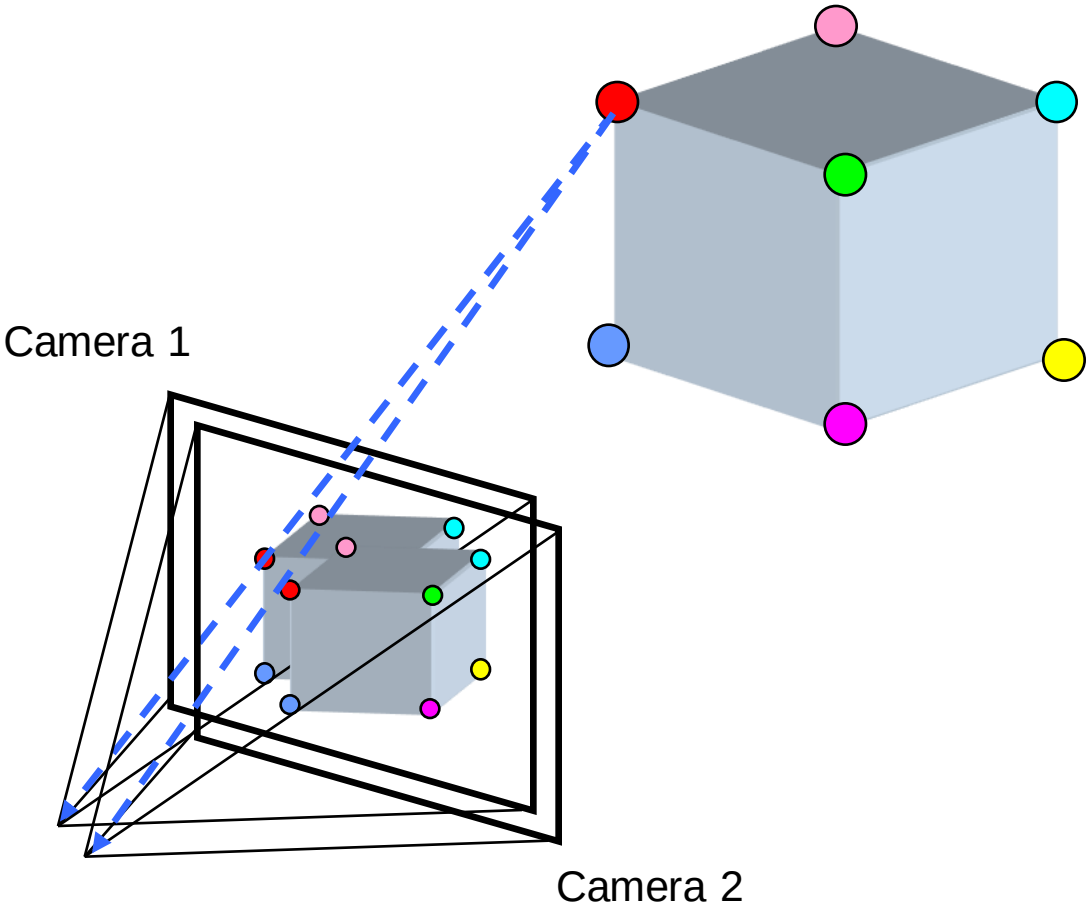
- **Structure from Motion:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



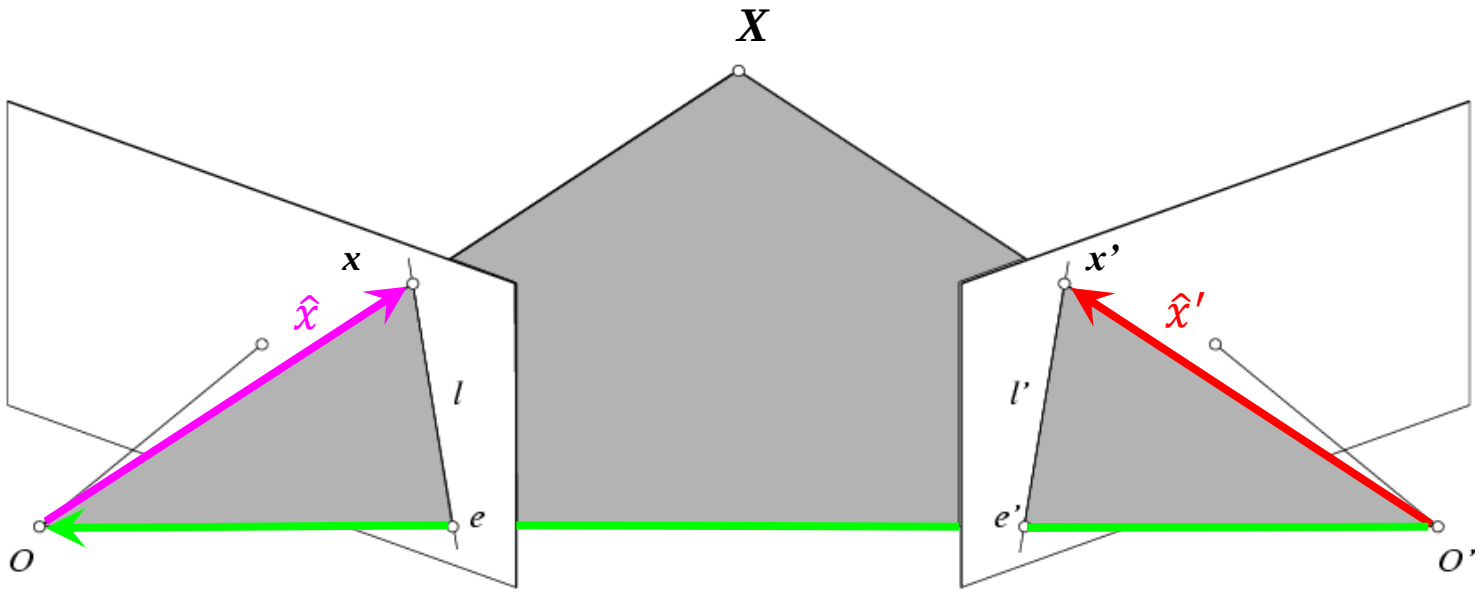
Slide credit: Noah Snavely

Multi-view geometry problems

- **Optical flow:** Given two images, find the location of a world point in a second close-by image with no camera info.



Essential matrix



$$\hat{x}\cdot[t\times(R\hat{x}')]=0 \quad \Rightarrow \quad \hat{x}^T E\,\hat{x}'=0 \quad \text{with} \quad E=[t]_\times R$$

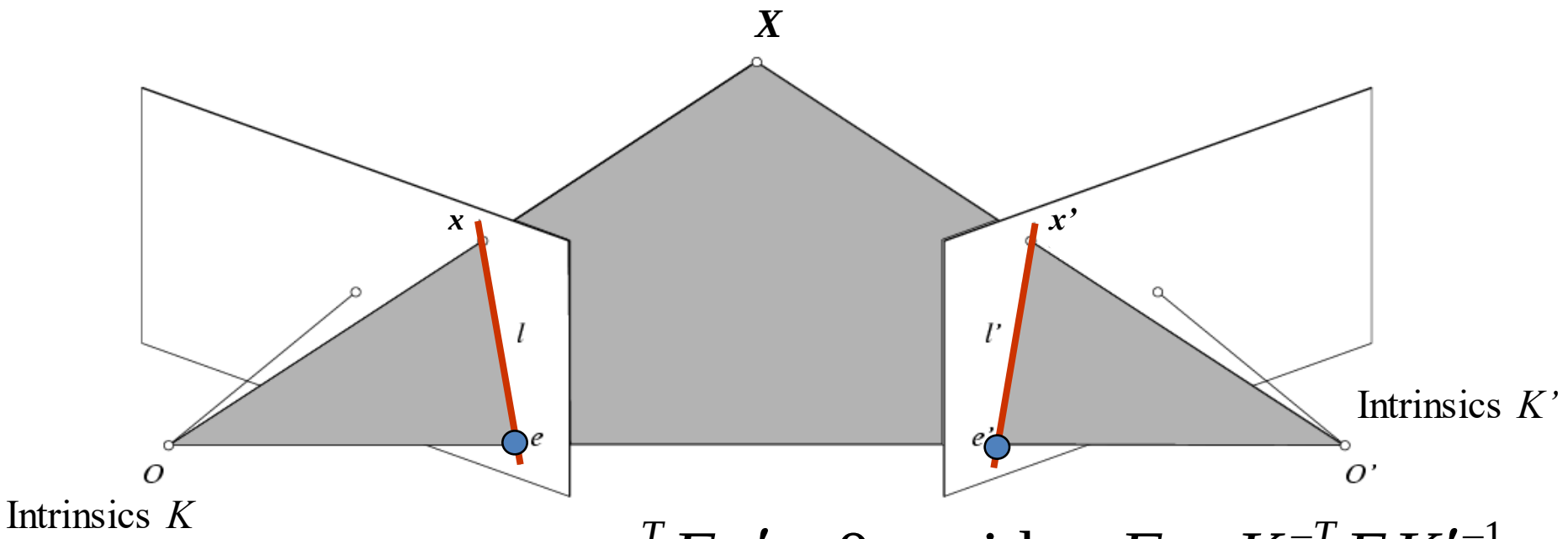
E is a 3x3 matrix which relates corresponding pairs of normalized homogeneous image points across pairs of images – for *K* calibrated cameras.

Essential Matrix
(Longuet-Higgins, 1981)

Estimates relative position/orientation.

Note: $[t]_\times$ is matrix representation of cross product

Fundamental matrix for uncalibrated cases



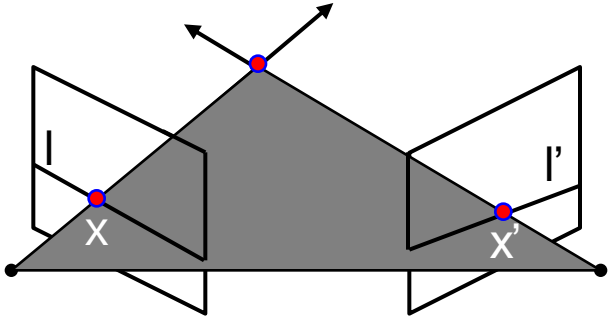
$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F\,x' = 0$ is the epipolar line l associated with x'
- $F^T x = 0$ is the epipolar line l' associated with x

- F is singular (rank two): det(F)=0
- $F\,e' = 0$ and $F^T e = 0$ (nullspaces of F = e'; nullspace of F^T = e')
- F has seven degrees of freedom: 9 entries but defined up to scale, det(F)=0

Fundamental matrix

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



Epipolar relation

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

Epipolar mapping described by a 3x3 matrix *F:*

$$l' = Fx$$
$$l = F^T x'$$

It follows that: $x' F x = 0$

Fundamental matrix

This matrix F is called

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

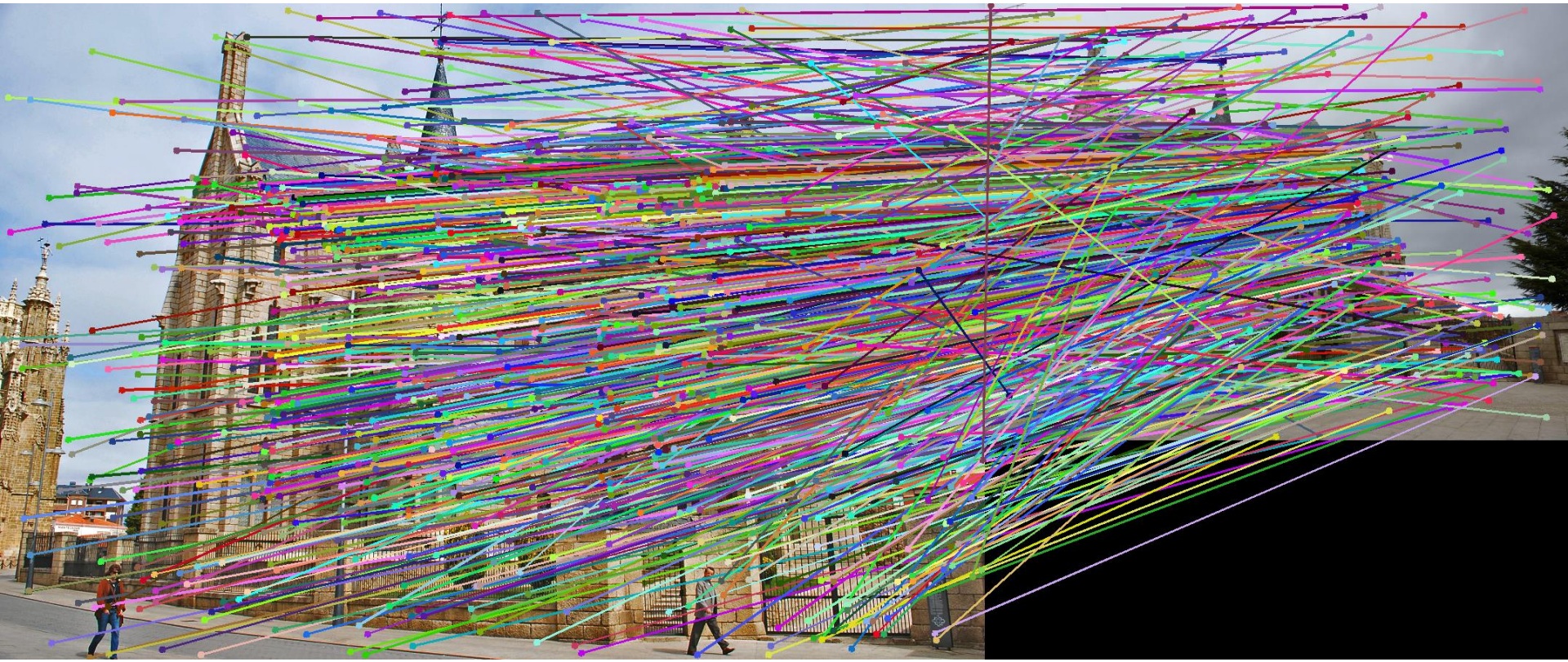
Can solve for F from point correspondences

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

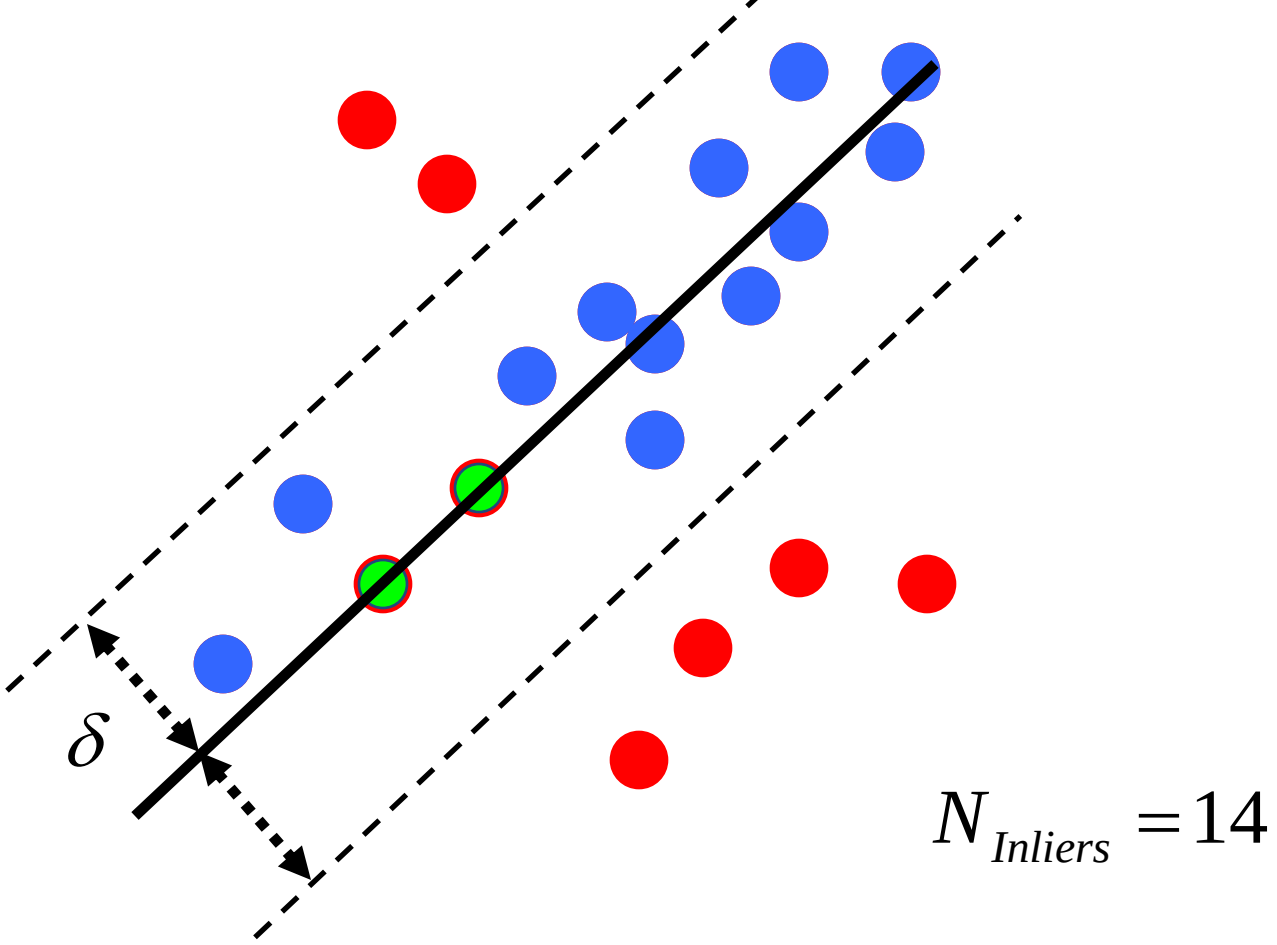
$$x'Fx = 0$$

- F has 9 entries, but really only 7 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

VLFeat's 800 most confident matches among 10,000+ local features.



RANSAC

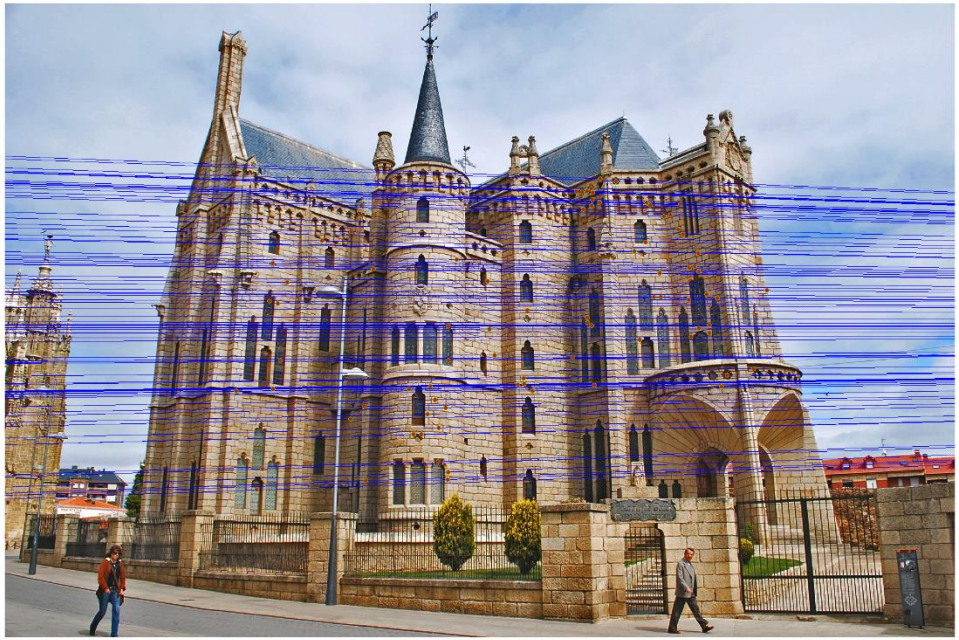


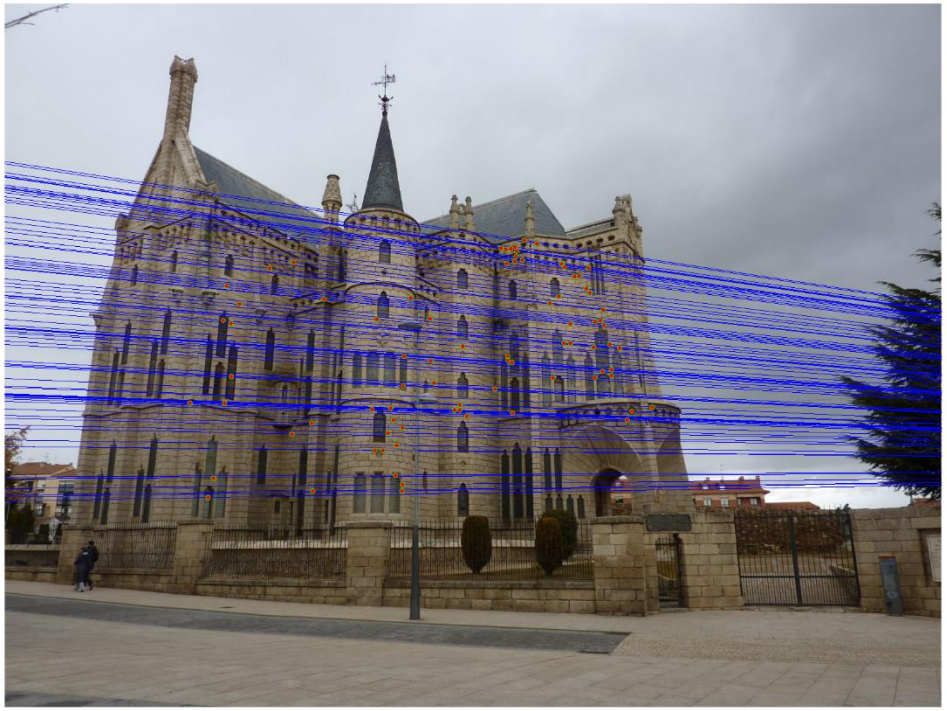
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (*s*=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

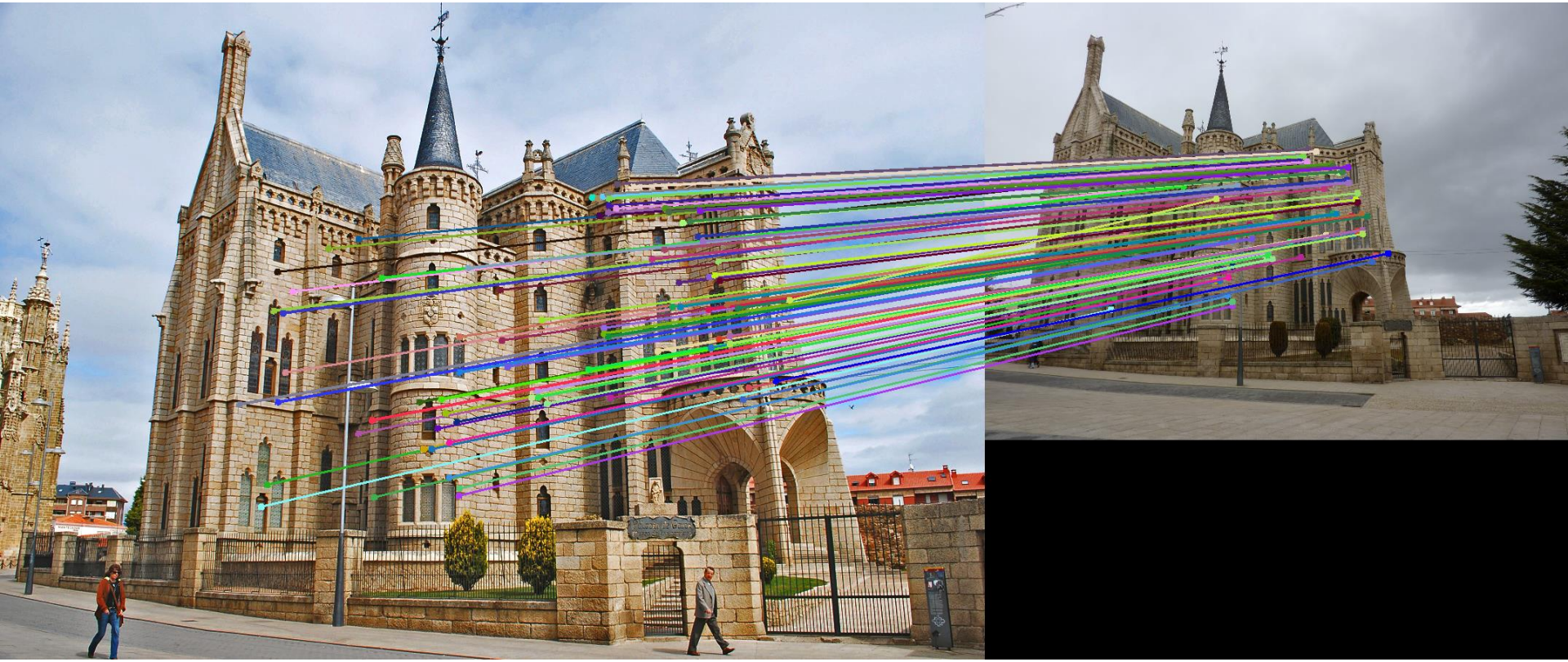
Repeat 1-3 until the best model is found with high confidence

Epipolar lines

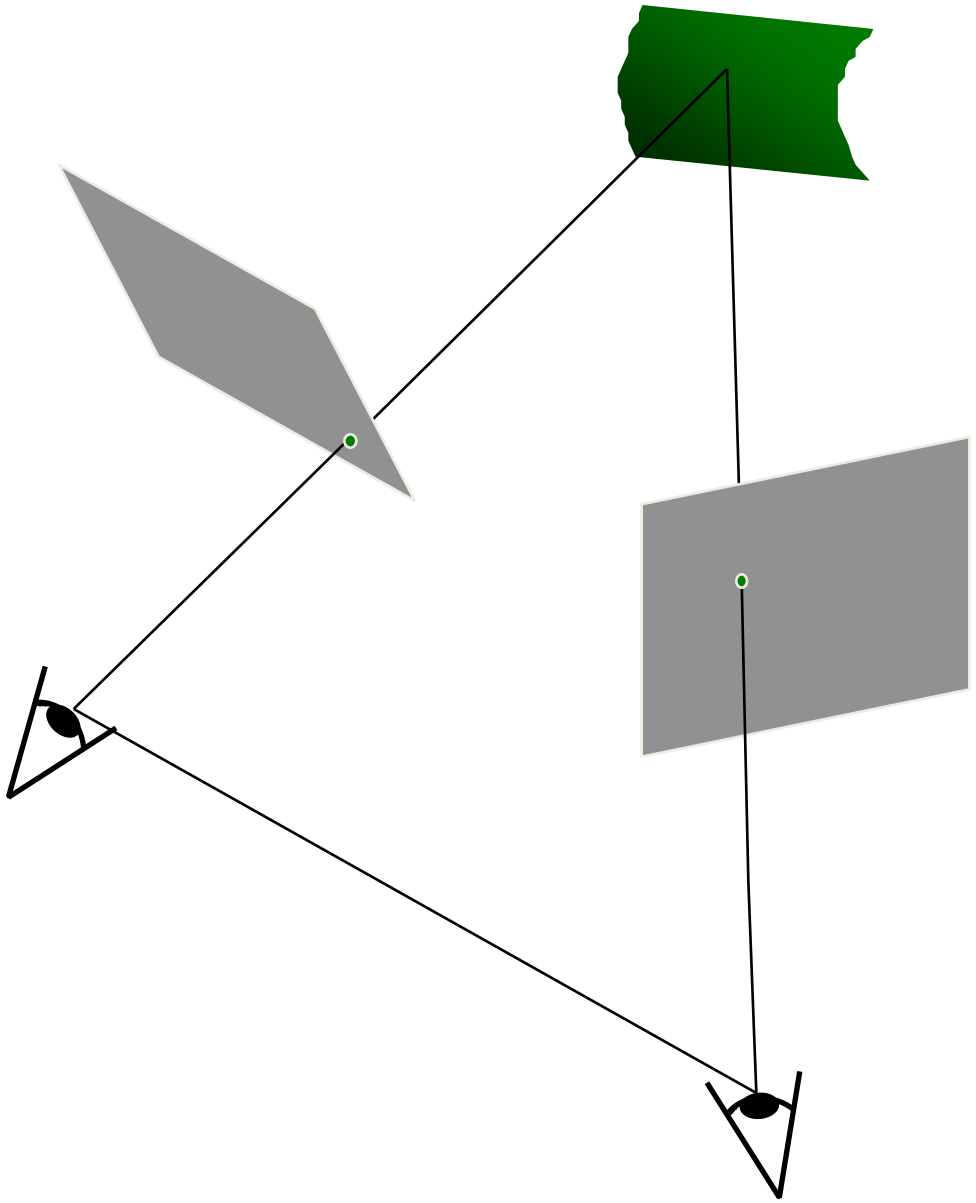




Keep only the matches at are “inliers” with respect to the “best” fundamental matrix

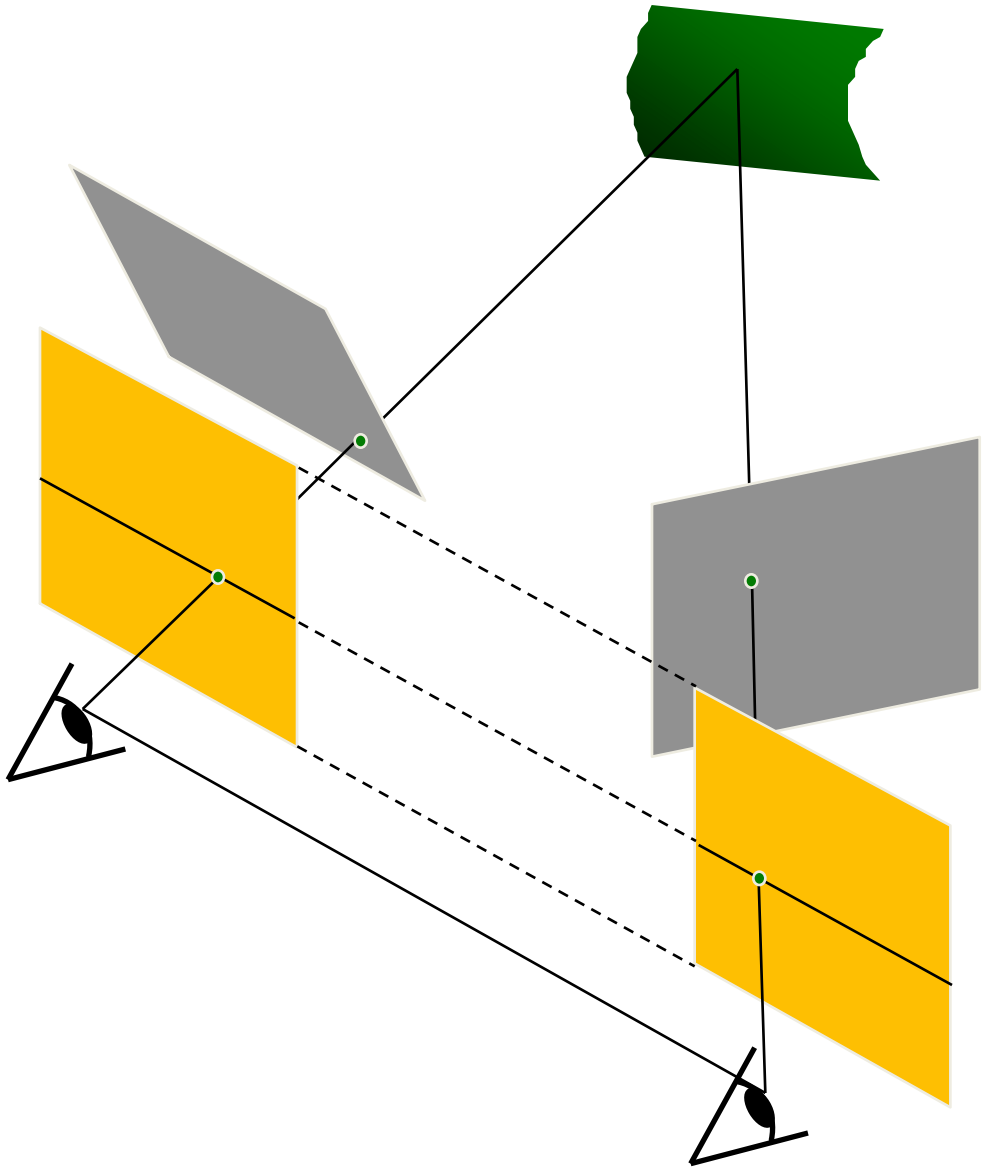


Stereo image rectification



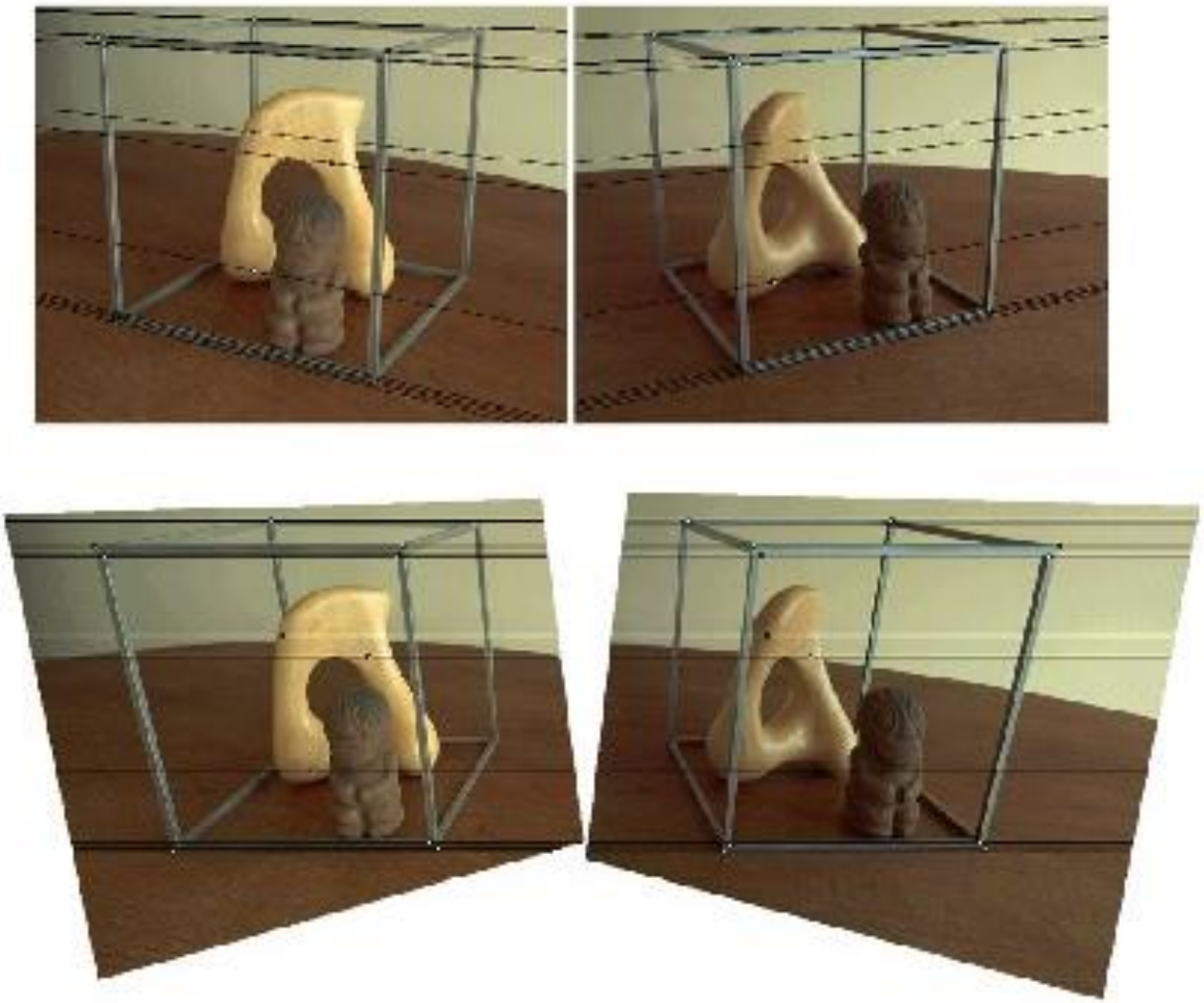
Stereo image rectification

- 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



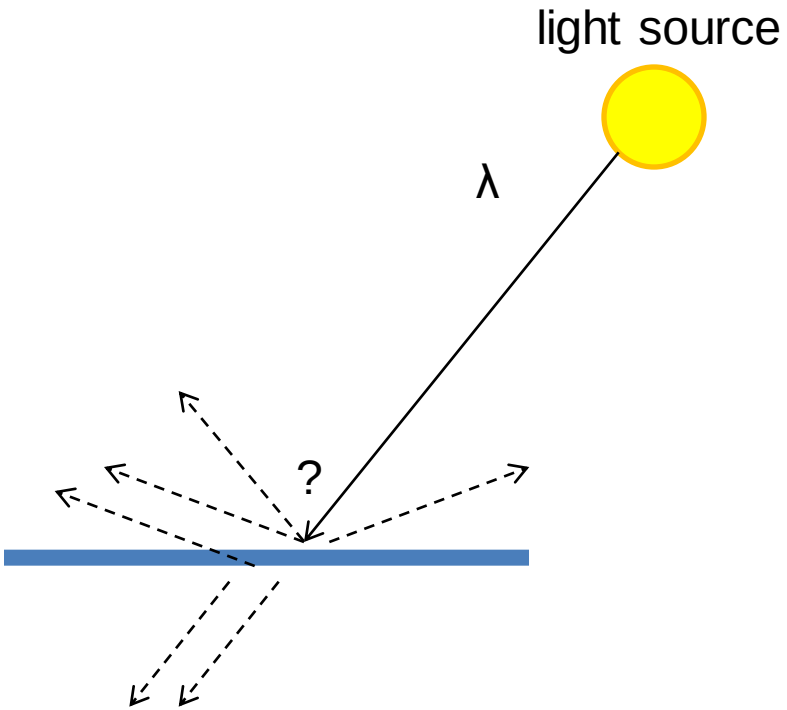
➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectification example



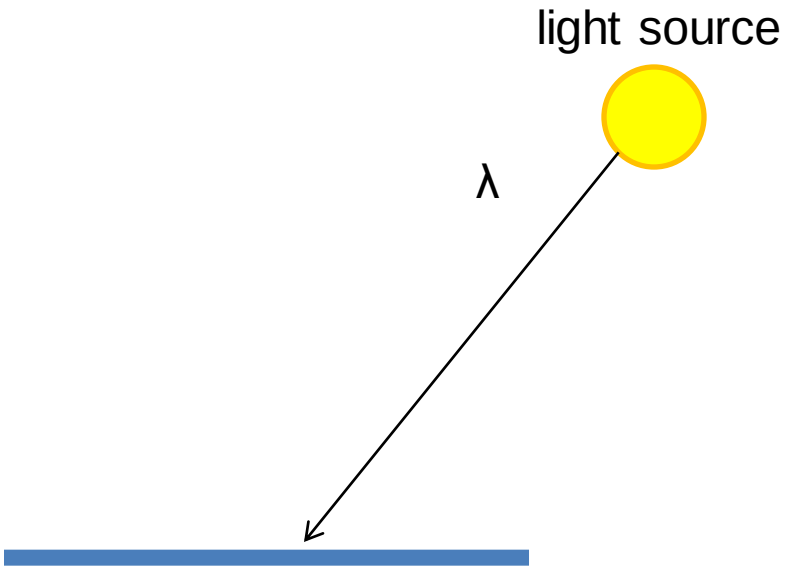
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



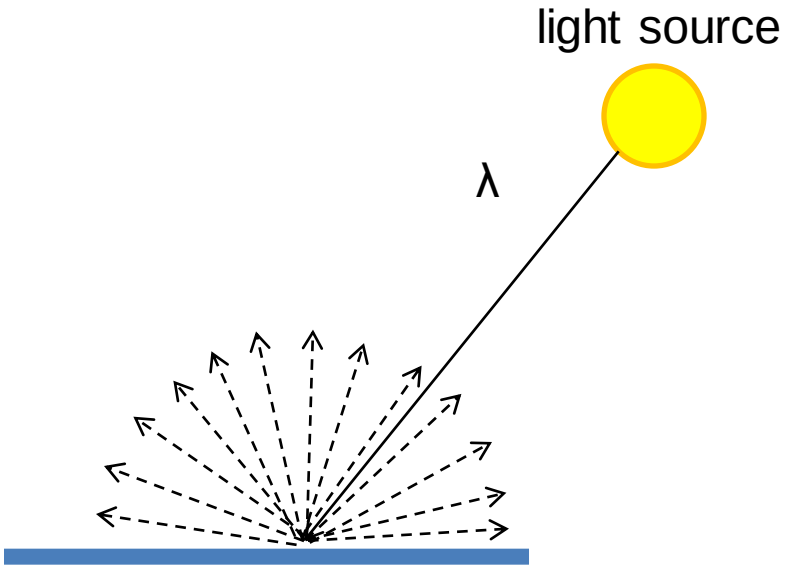
A photon's life choices

- **Absorption**
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



A photon's life choices

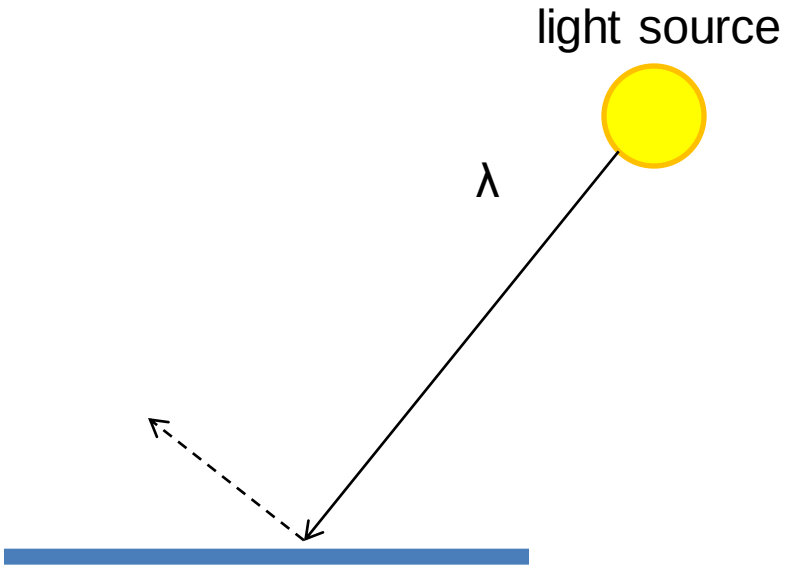
- Absorption
- **Diffuse Reflection**
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Perfect diffuse
= Lambertian
= Equal in all directions

A photon's life choices

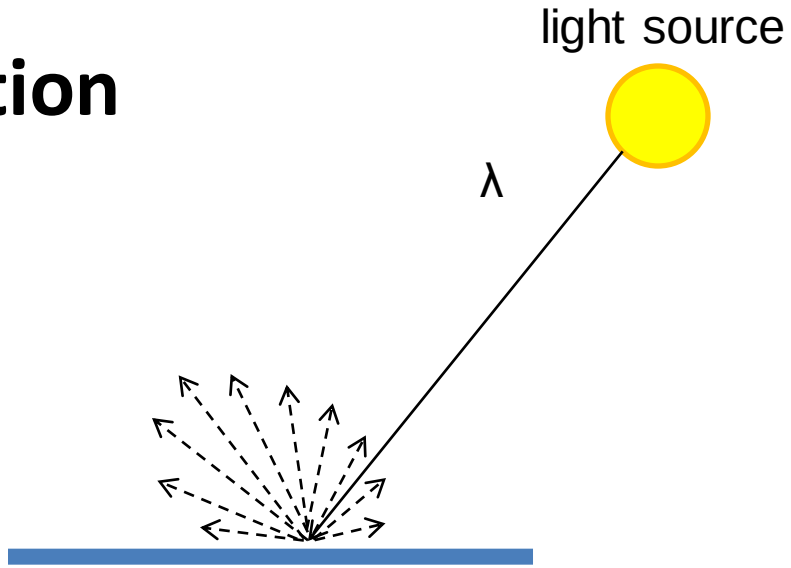
- Absorption
- Diffusion
- **Specular Reflection**
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Perfect specular
= mirror reflection
= only one direction

A photon's life choices

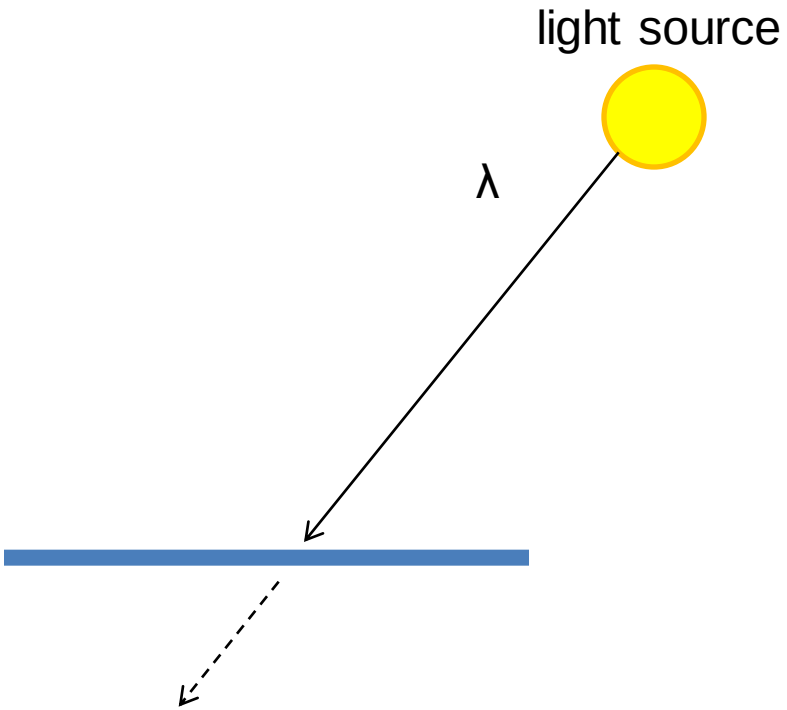
- Absorption
- Diffusion
- **Specular (Glossy) Reflection**
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Glossy reflection
= 'specular lobe'
= varying across directions

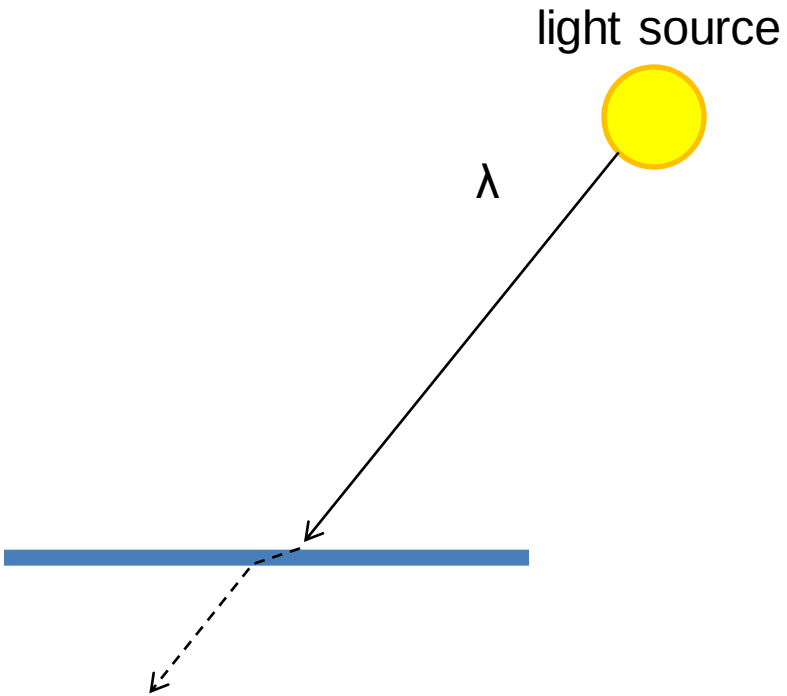
A photon's life choices

- Absorption
- Diffusion
- Reflection
- **Transparency**
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



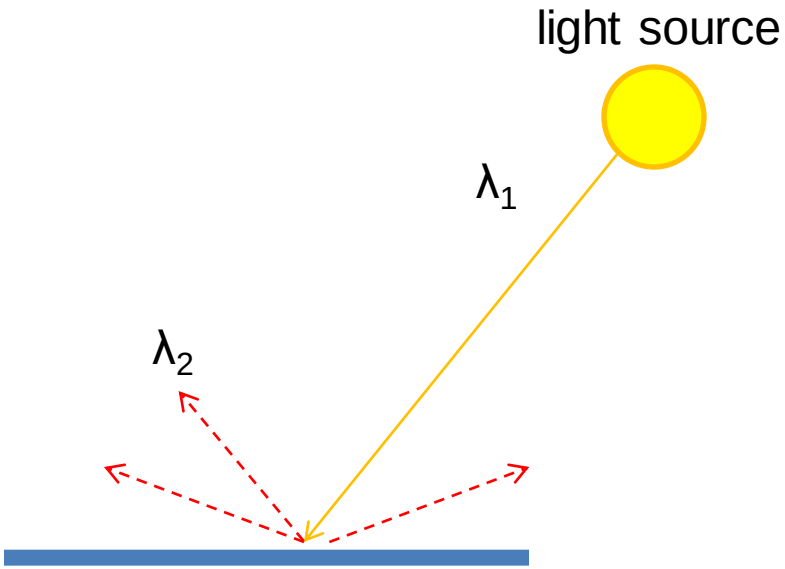
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- **Refraction**
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



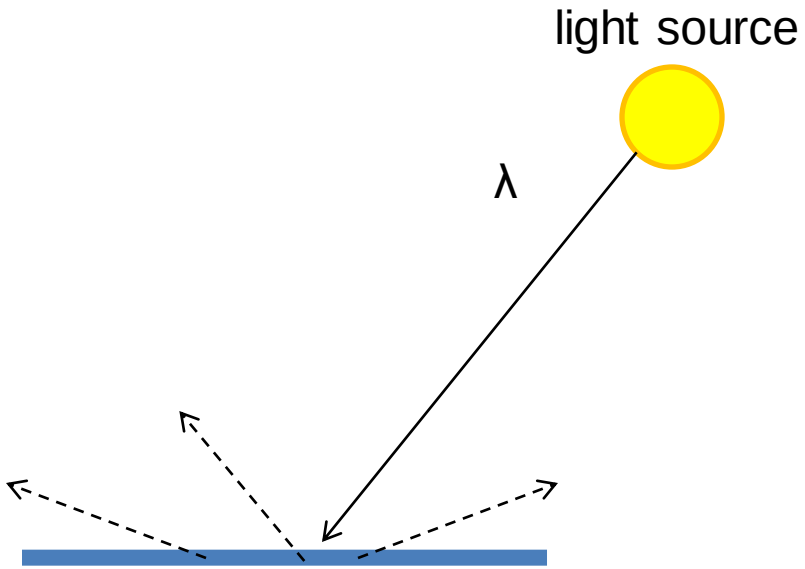
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- **Fluorescence**
- Subsurface scattering
- Phosphorescence
- Interreflection



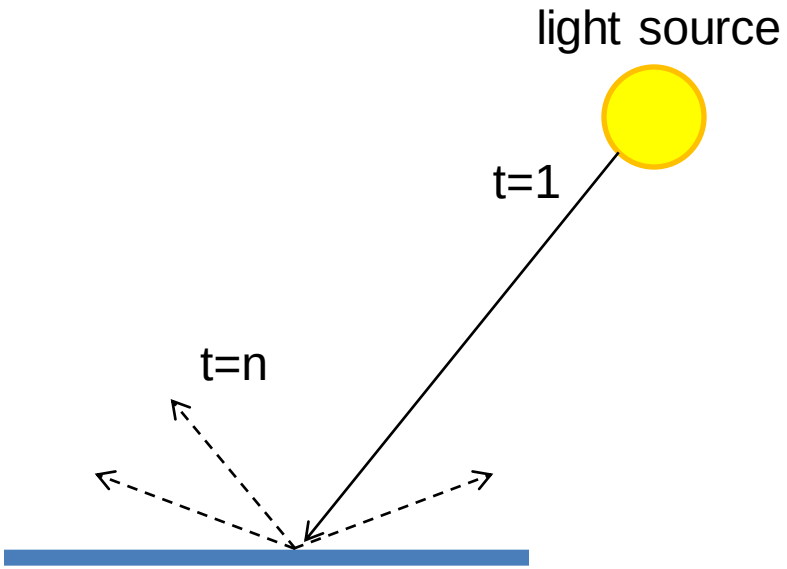
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- **Subsurface scattering**
- Phosphorescence
- Interreflection



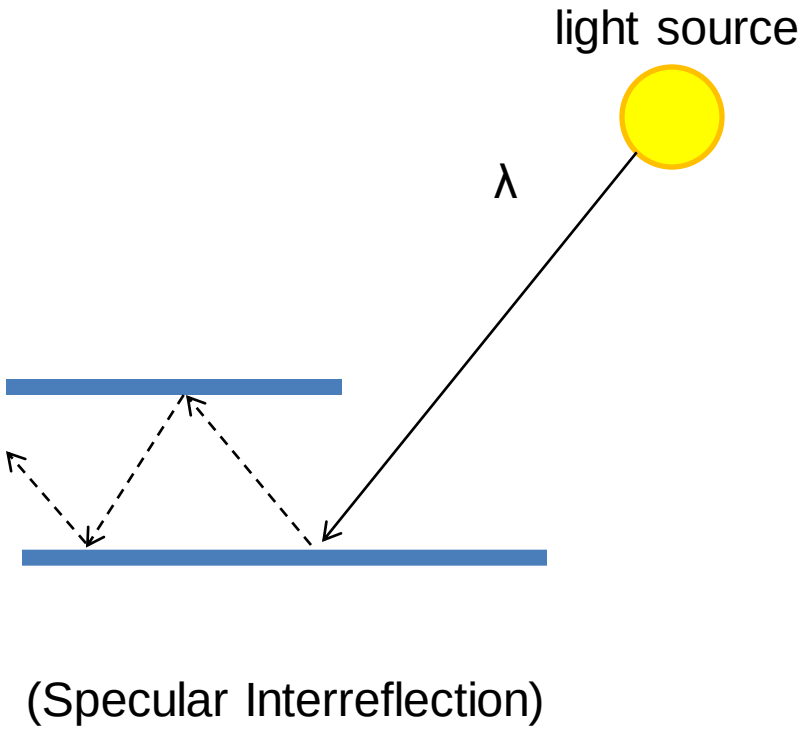
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- **Phosphorescence**
- Interreflection



A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- **Interreflection**

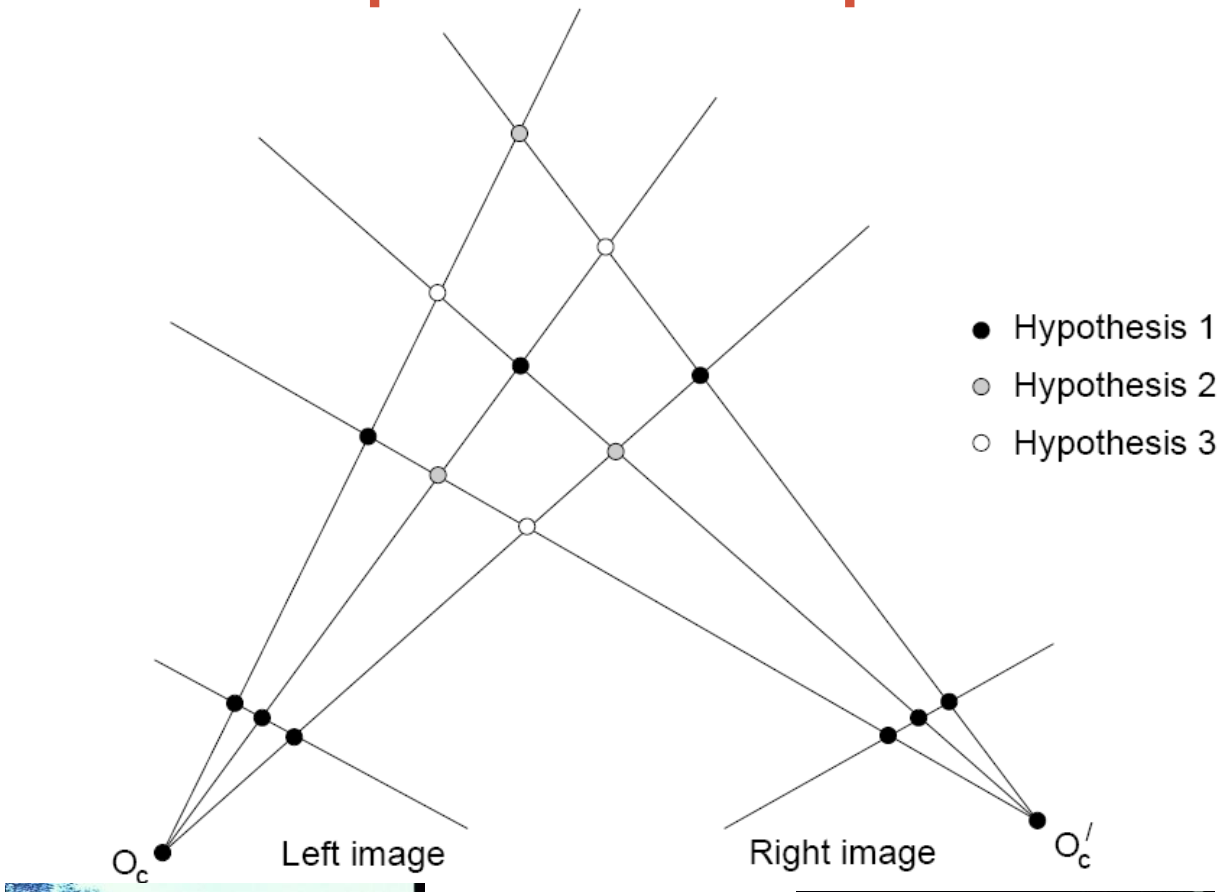


Lambertian Reflectance

In computer vision, surfaces are often assumed to be ideal diffuse reflectors with no dependence on viewing direction.

This is obviously nonsense, but a useful model!

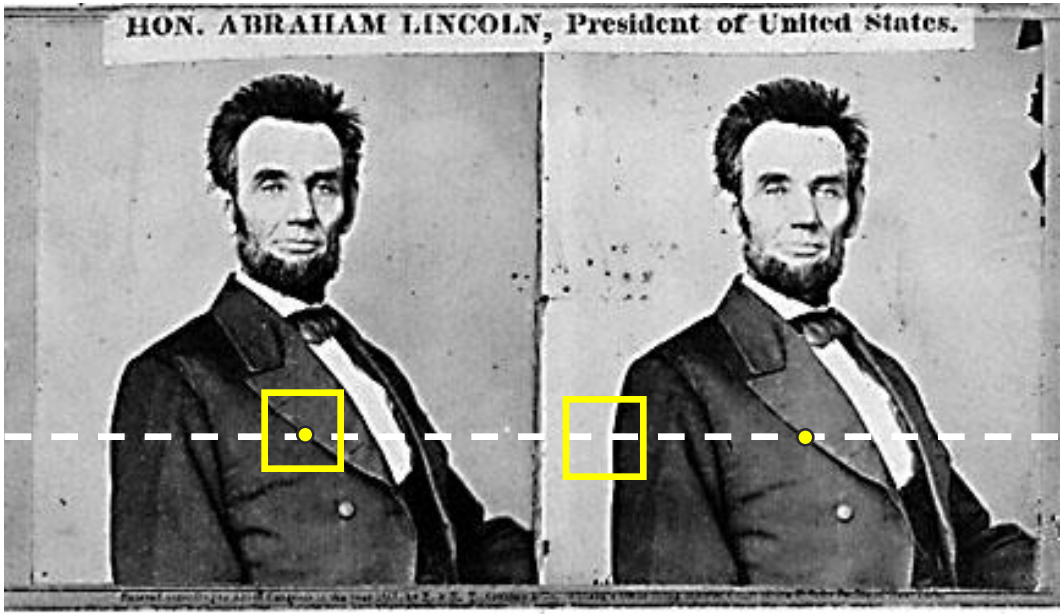
Correspondence problem



Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Figure from Gee & Cipolla 1999

Dense correspondence search



For each epipolar line:

For each pixel / window in the left image:

- Compare with every pixel / window on same epipolar line in right image
- Pick position with minimum match cost (e.g., SSD, normalized correlation)

Adapted from Li Zhang

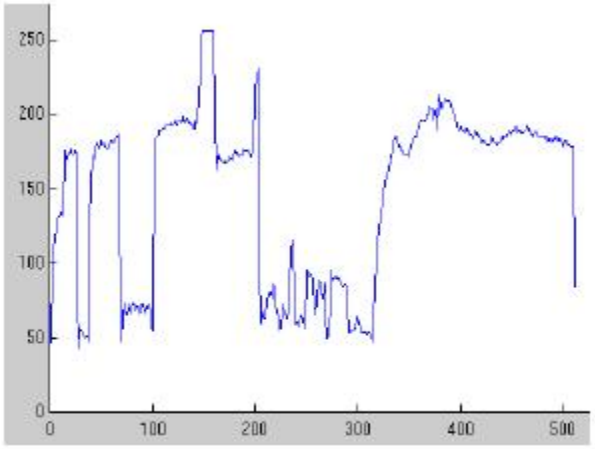
Think-Pair-Share

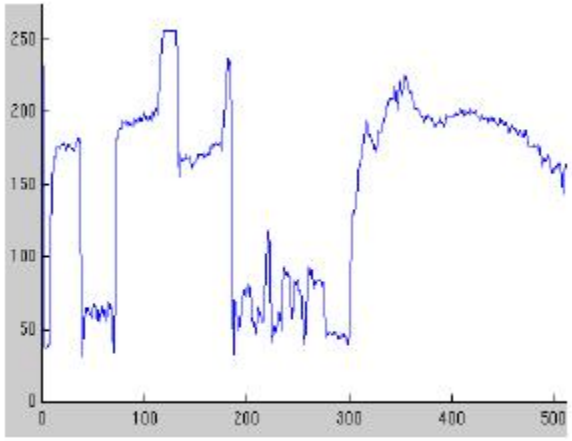
How can we solve this problem?

For which ‘real-world’ phenomena will this work?
For which will it not?

Correspondence problem

Intensity profiles

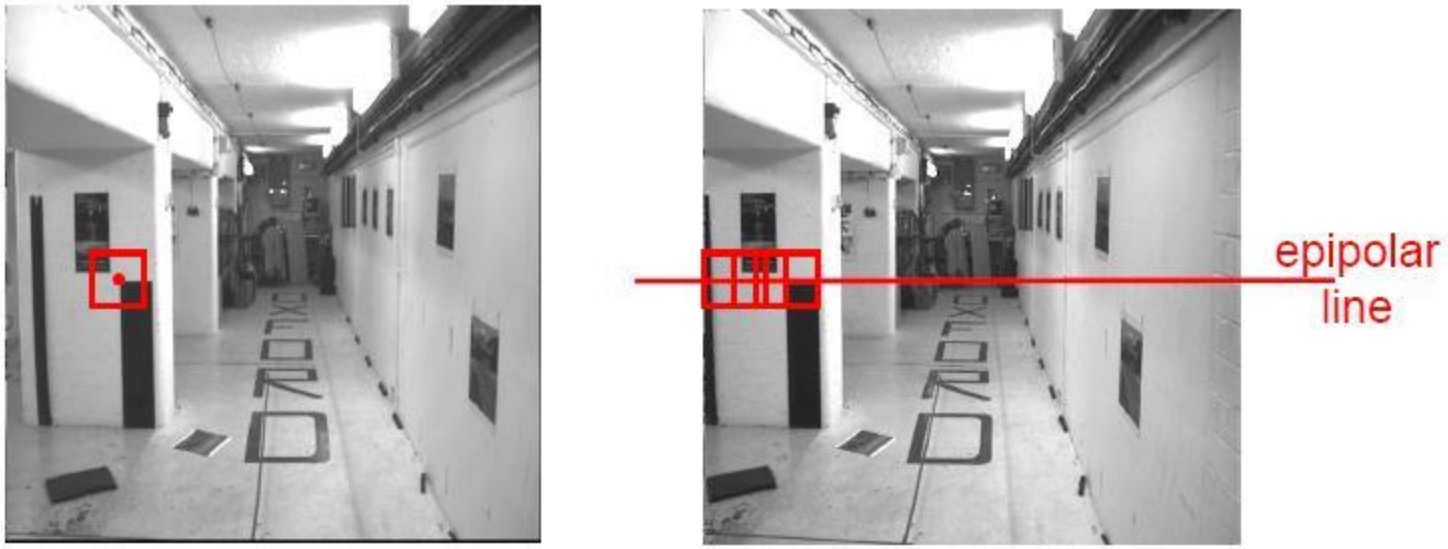




- Clear correspondence between intensities, but also noise and ambiguity

Source: Andrew Zisserman

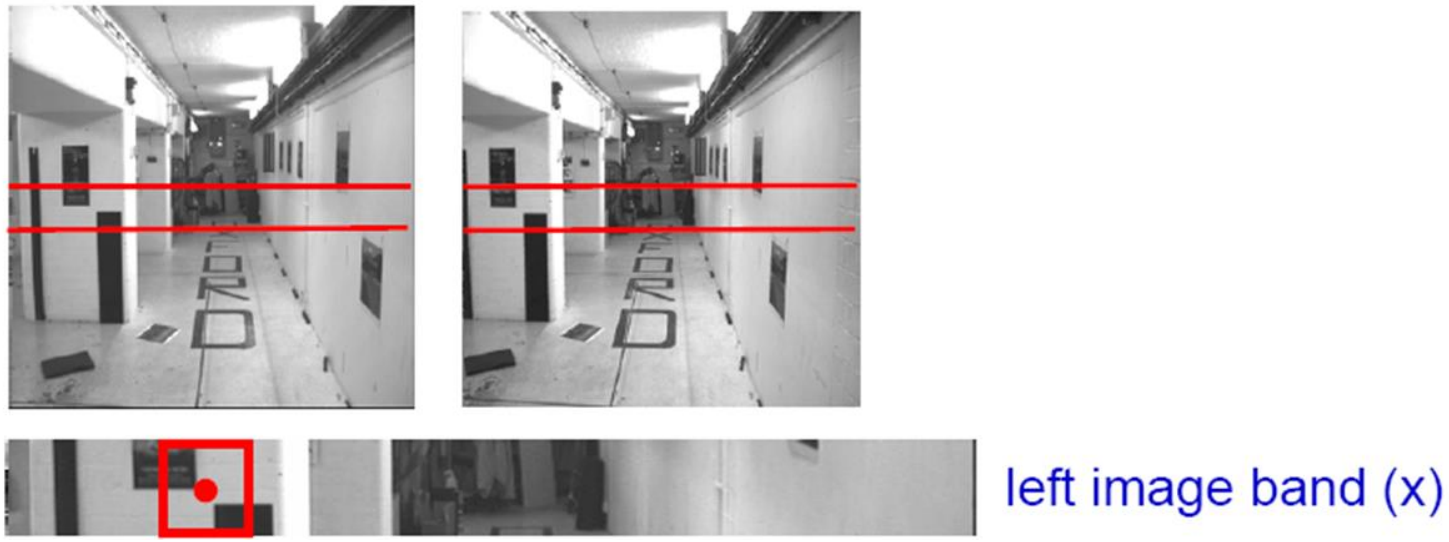
Correspondence problem



Neighborhoods of corresponding points are similar in intensity patterns.

Source: Andrew Zisserman

Correlation-based window matching



Correlation-based window matching

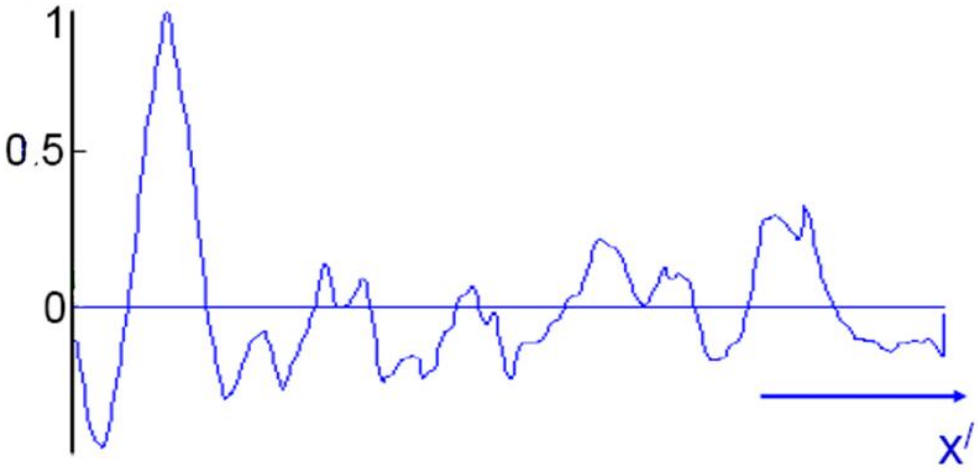
left image band (x)

right image band (x')

Correlation-based window matching

left image band (x)

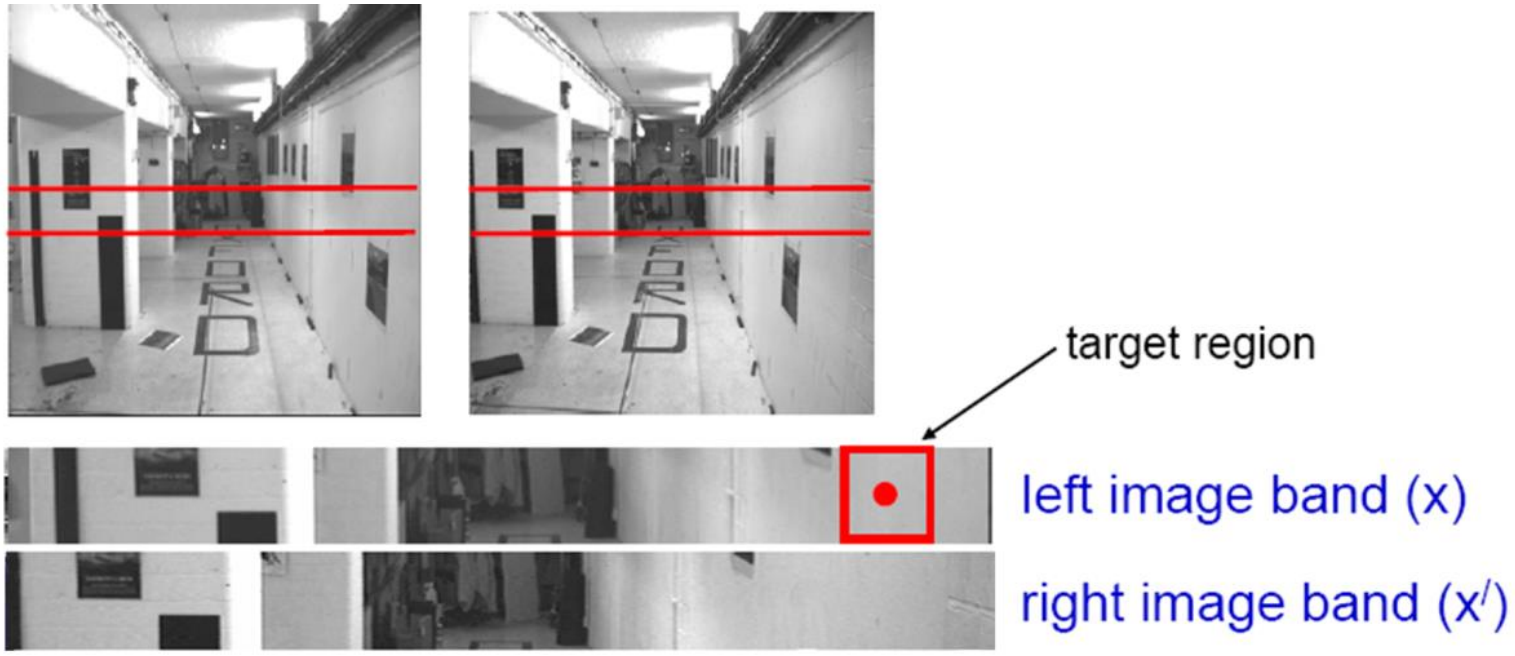
right image band (x')



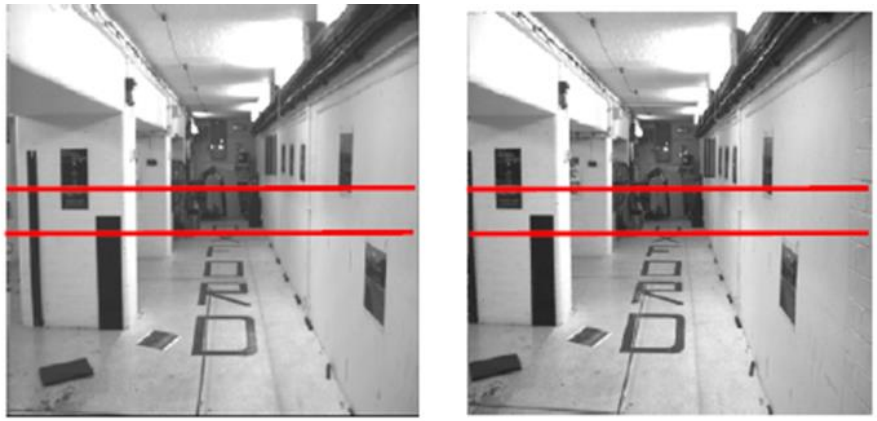
cross correlation

disparity = $x' - x$

Correlation-based window matching



Correlation-based window matching

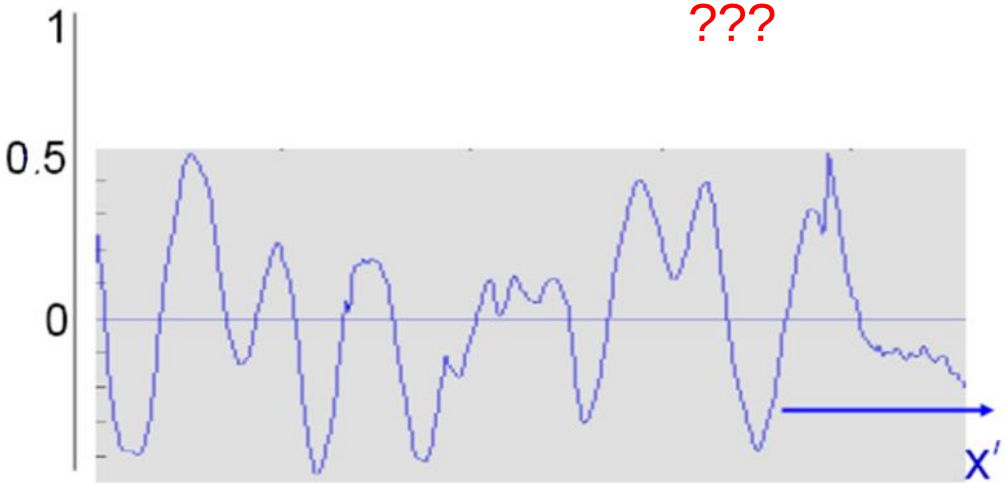


target region

left image band (x)

right image band (x')

???

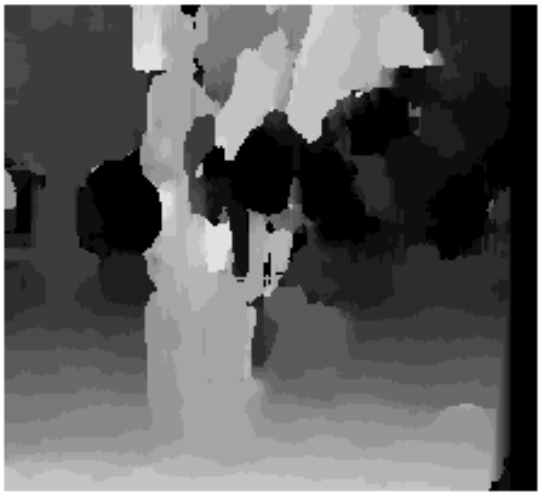


cross correlation

Textureless regions are non-distinct; high ambiguity for matches.

Effect of window size

W = 3



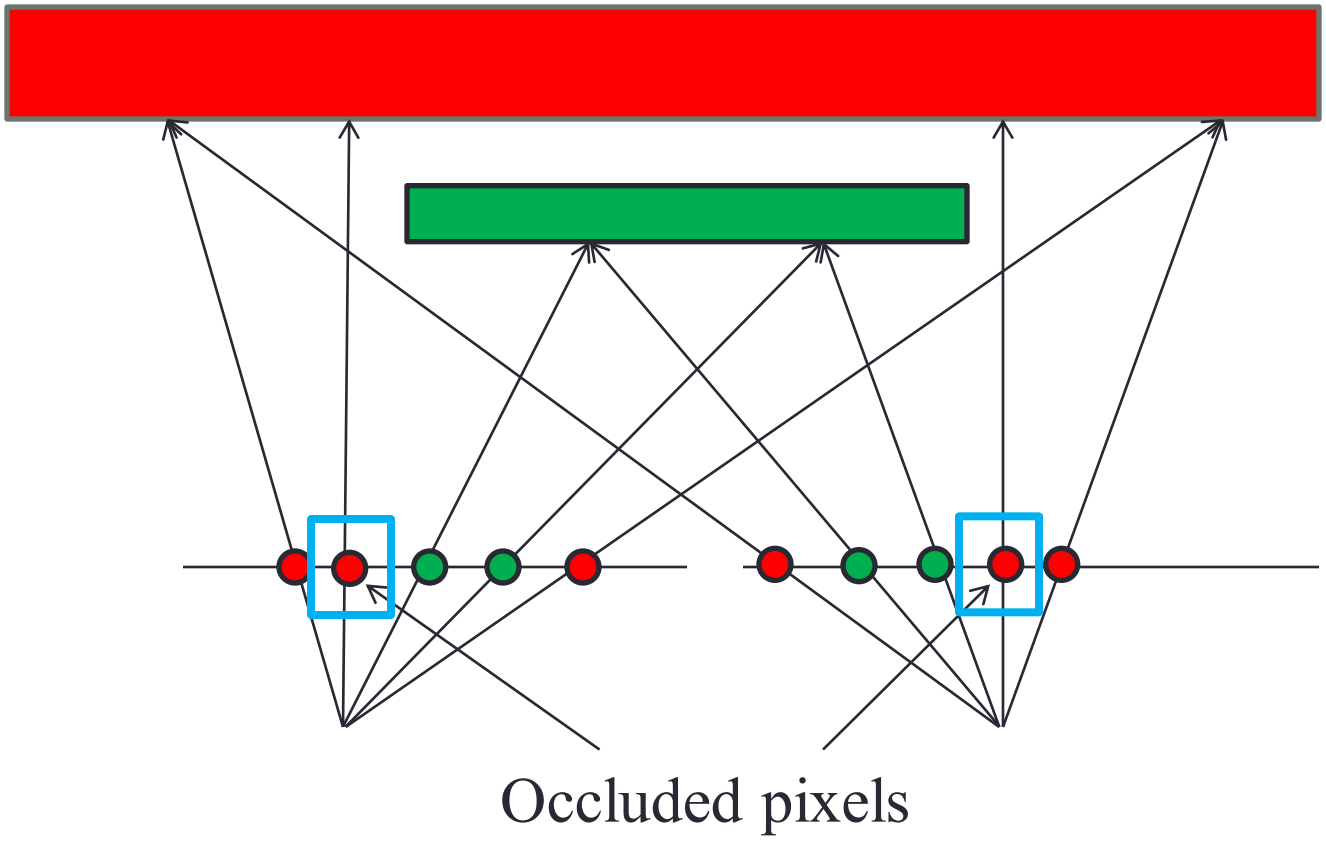
W = 20

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Figures from Li Zhang

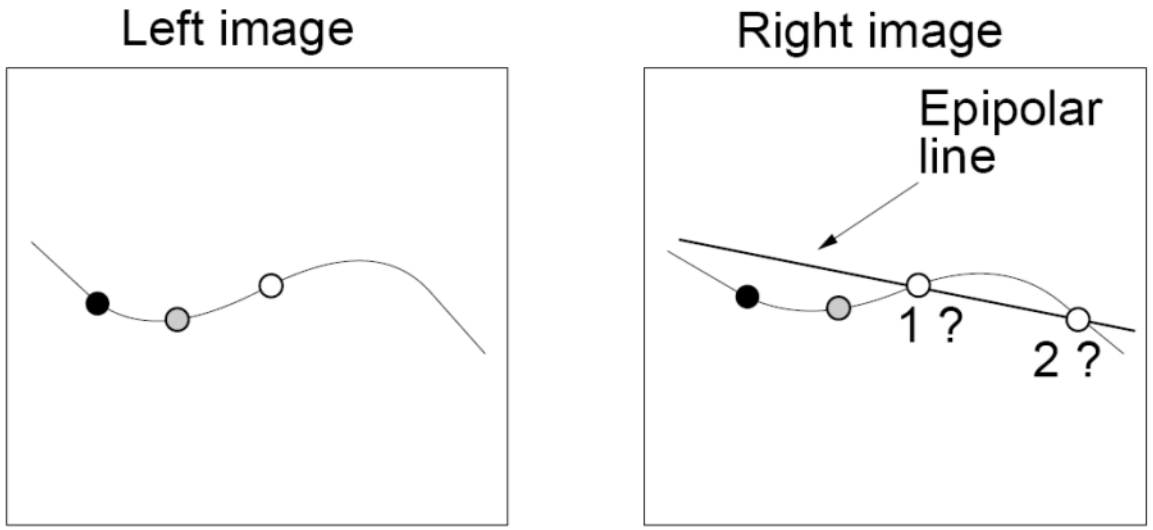
Problem: Occlusion

- Uniqueness says “up to match” per pixel
- When is there no match?



Disparity gradient constraint

- Assume piecewise continuous surface, so want disparity estimates to be locally smooth



Given matches ● and ◎, point ○ in the left image must match point 1 in the right image. Point 2 would exceed the disparity gradient limit.

Figure from Gee & Cipolla 1999

Ordering constraint

- Points on **same surface** (opaque object) will be in same order in both views

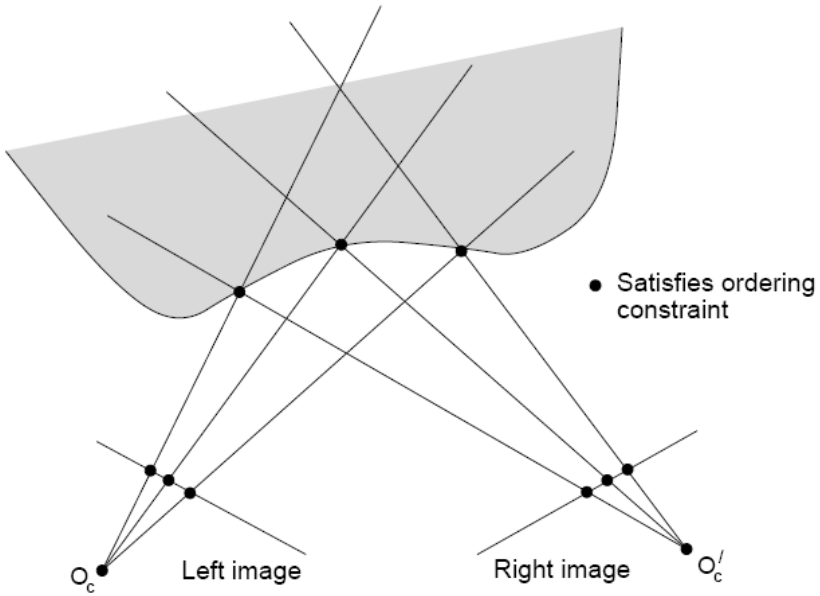
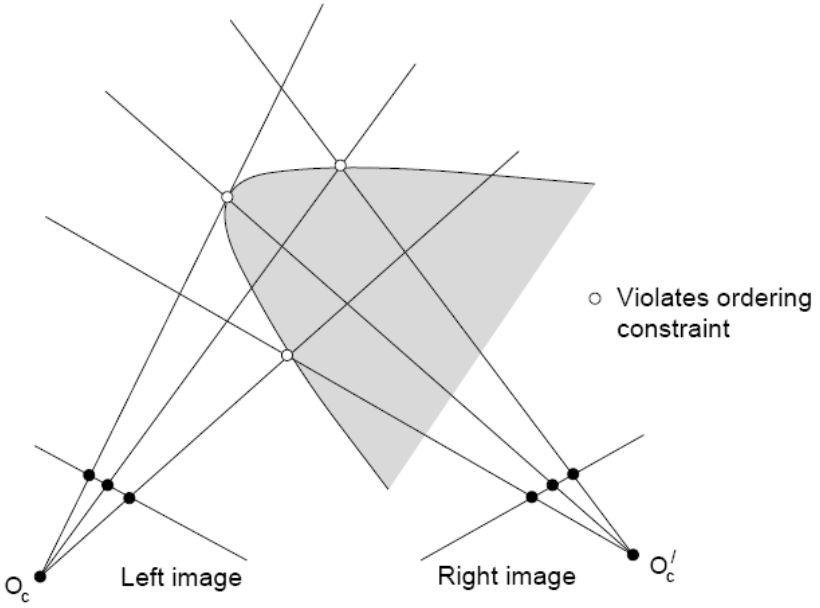
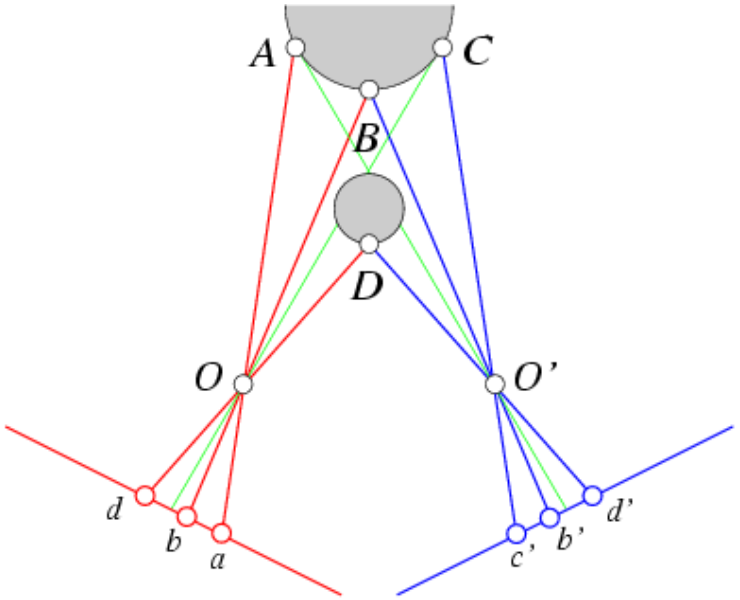


Figure from Gee & Cipolla 1999

Ordering constraint

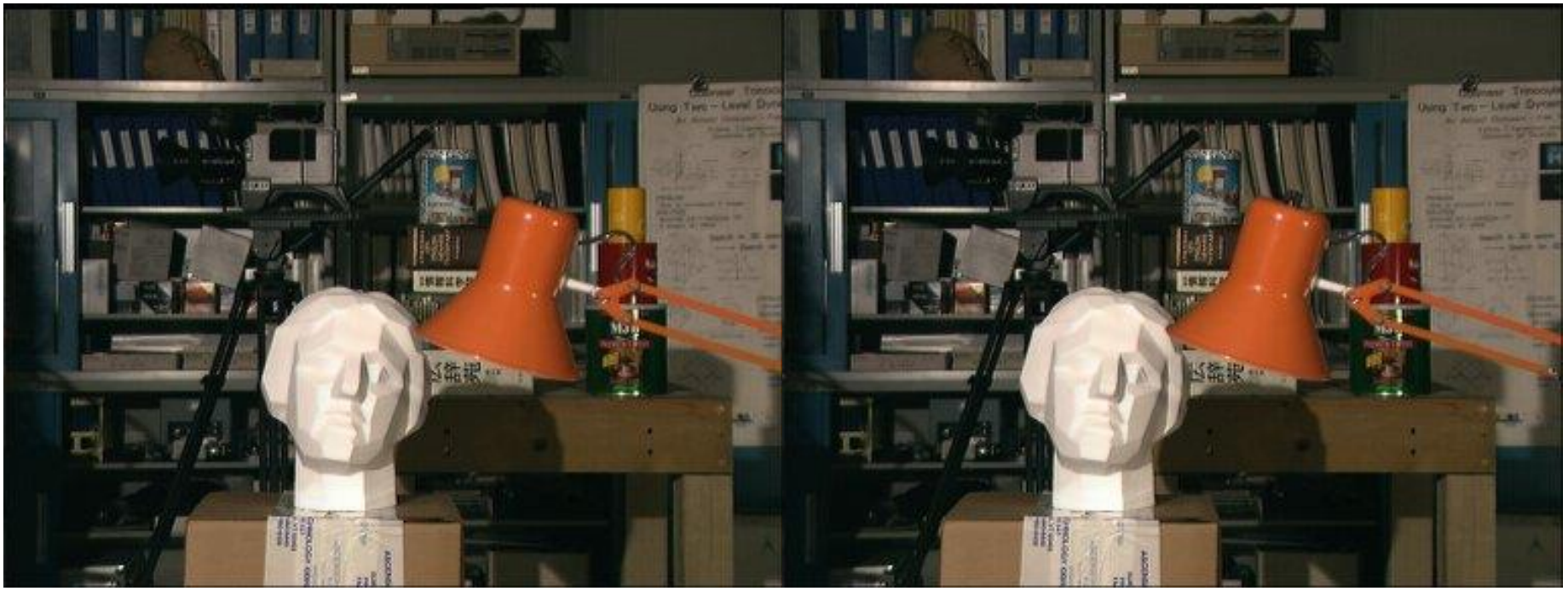
- Won't always hold, e.g. consider transparent object, or an occluding surface



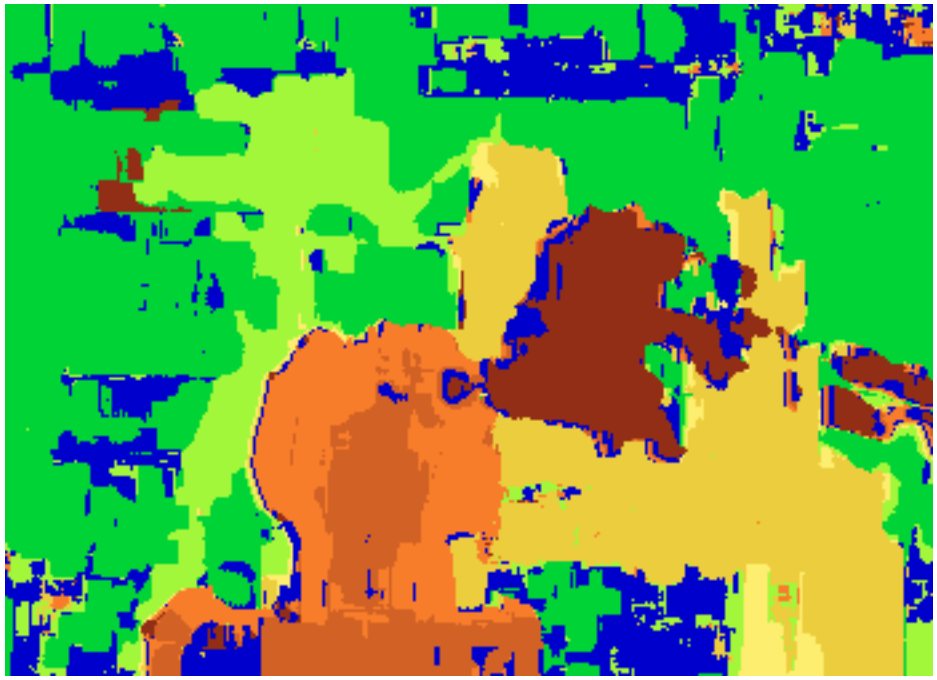


Figures from Forsyth & Ponce

Stereo – Tsukuba test scene (now old)



Results with window search



Window-based matching
(best window size)

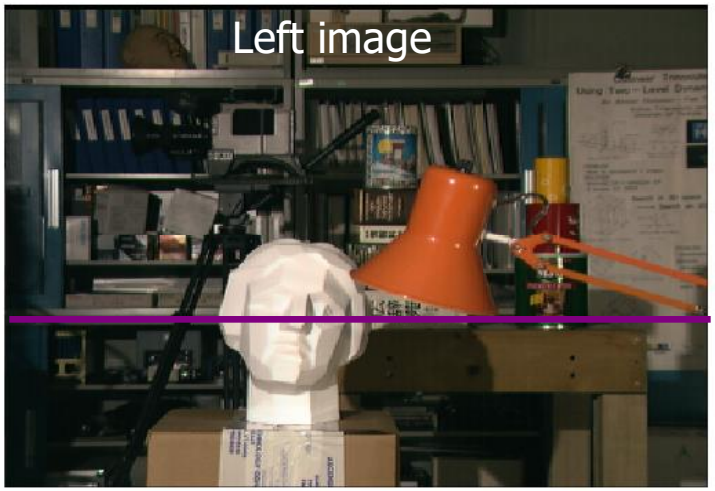
‘Ground truth’

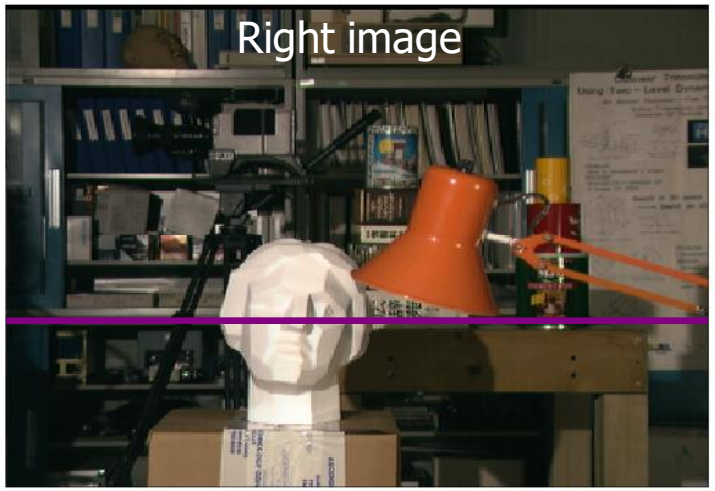
Better solutions

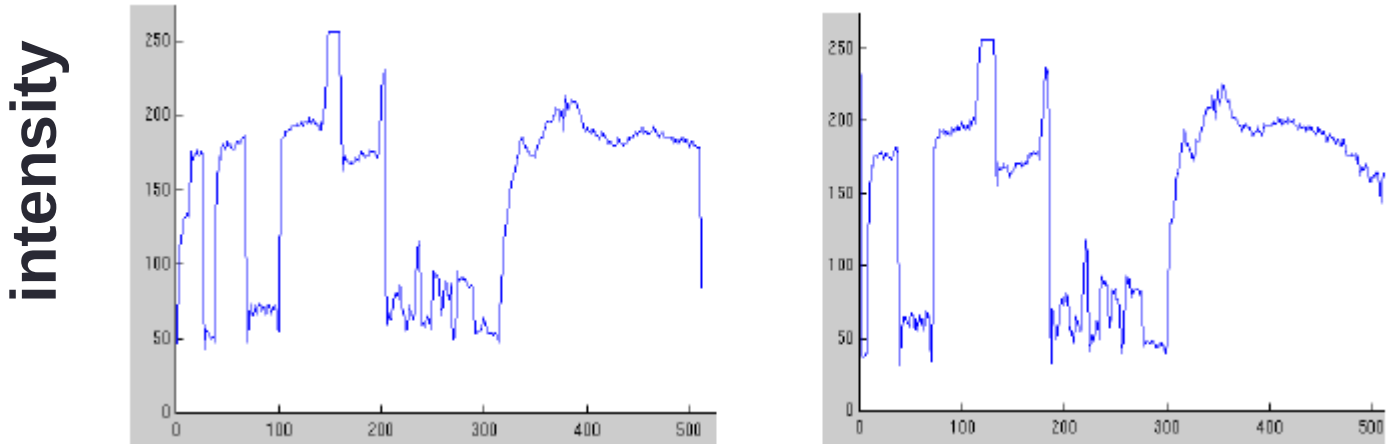
- Beyond individual correspondences to estimate disparities:
- Optimize correspondence assignments jointly
  - Scanline at a time (DP)
  - Full 2D grid (graph cuts)

Scanline stereo

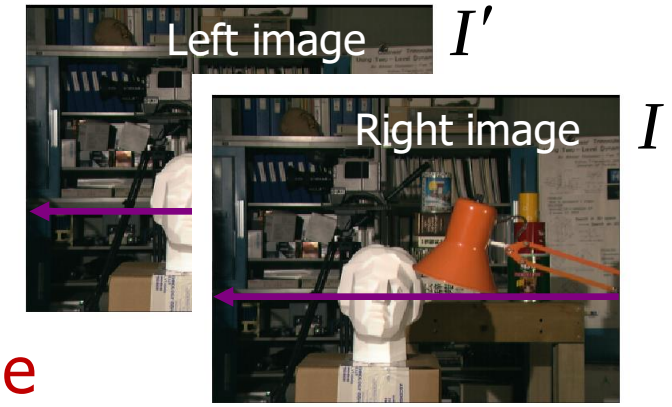
- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

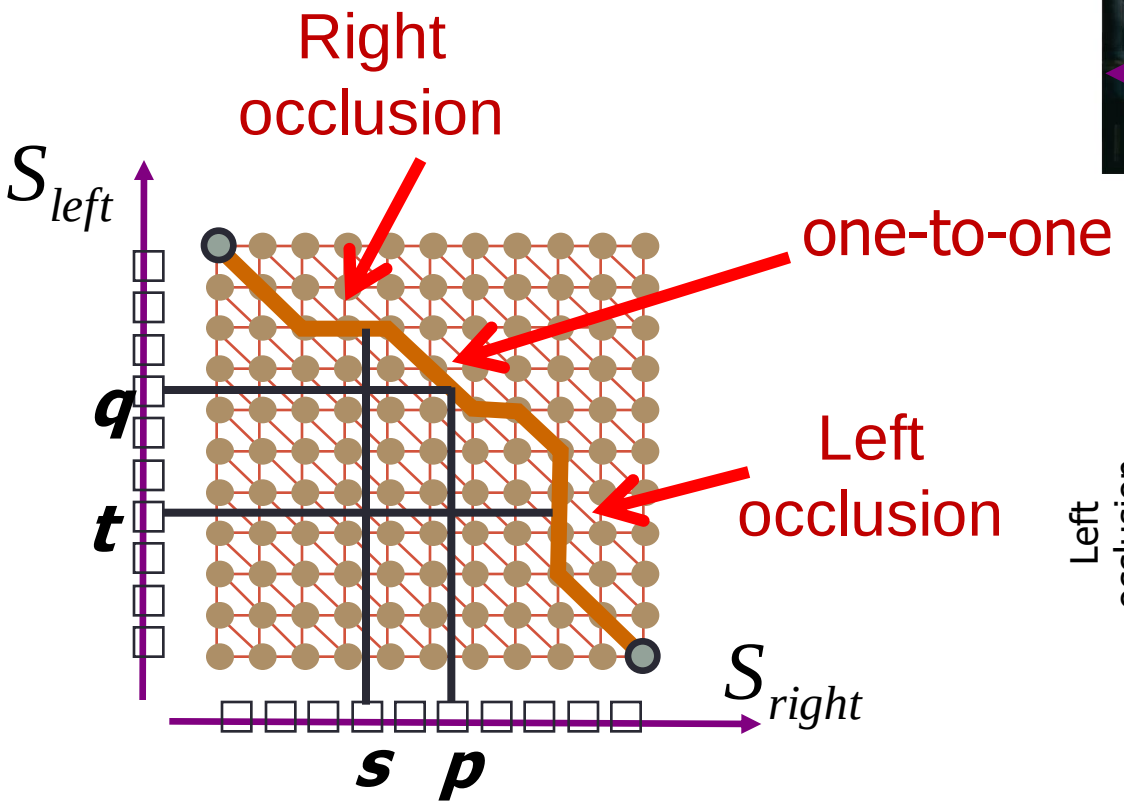


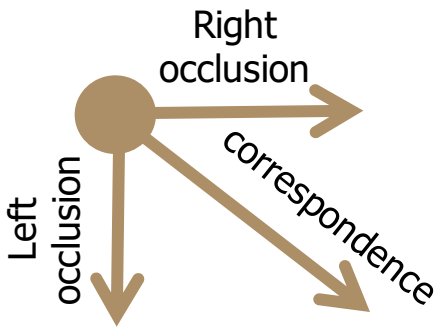




“Shortest paths” for scan-line stereo







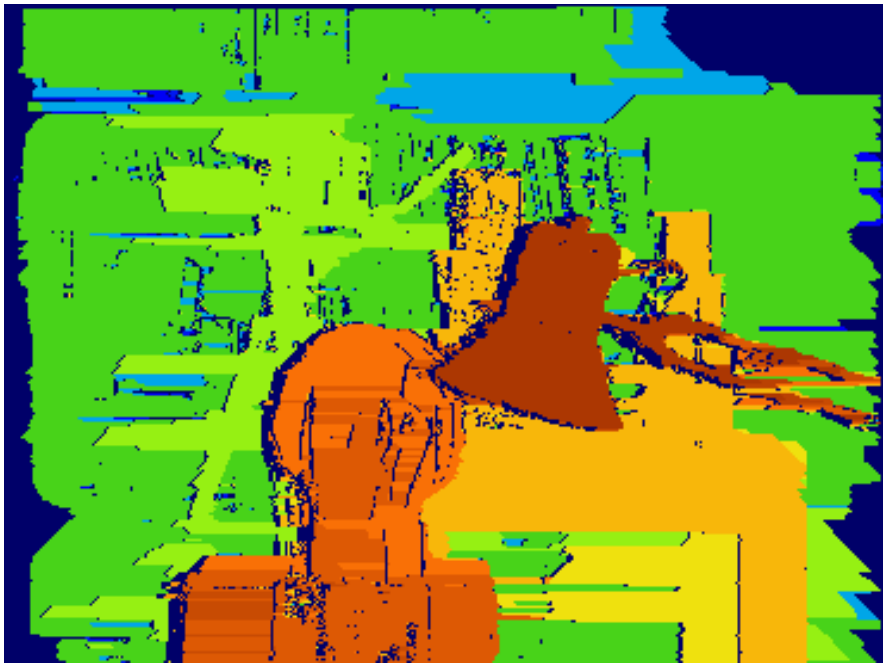
Can be implemented with dynamic programming

Ohta & Kanade ’85, Cox et al. ’96, Intille & Bobick, ‘01

Slide credit: Y. Boykov

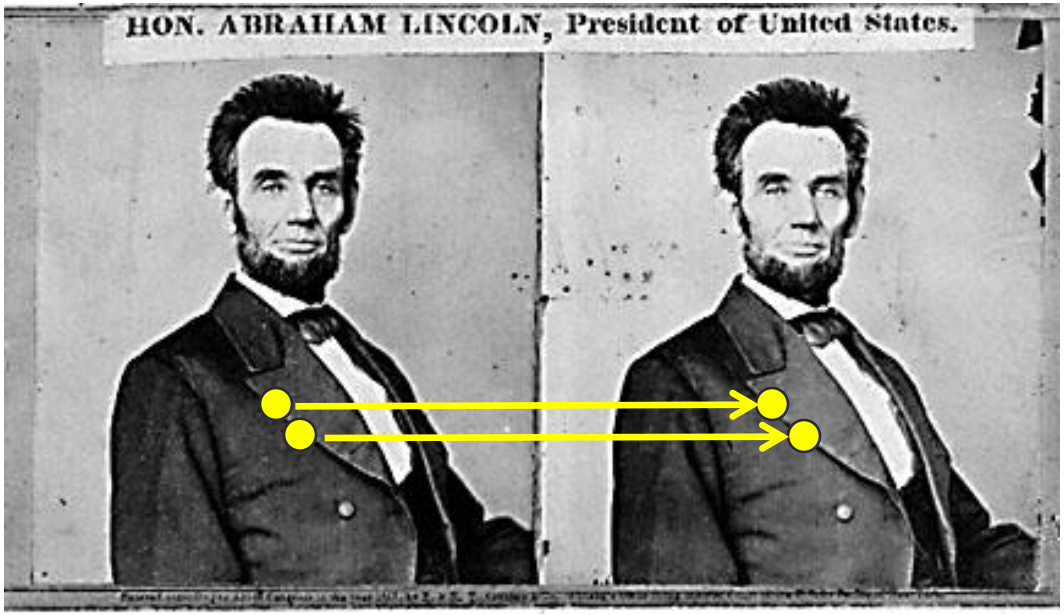
Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts



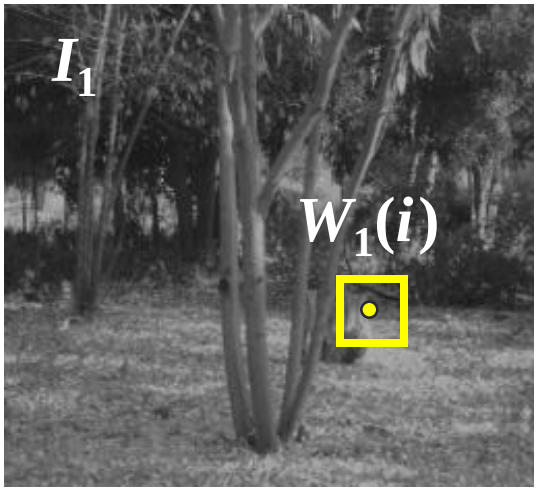
- Can’t use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

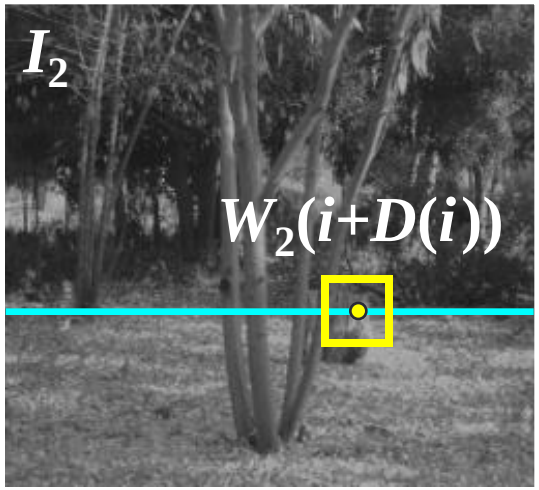
Stereo as energy minimization

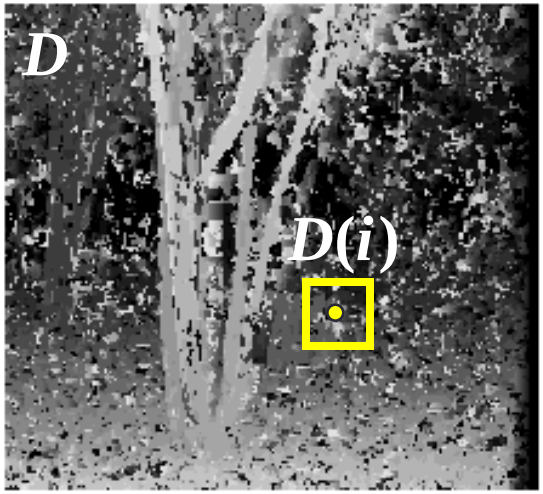


- What defines a good stereo correspondence?
  1. Match quality
     - Want each pixel to find a good match in the other image
  2. Smoothness
     - If two pixels are adjacent, they should (usually) move about the same amount

Stereo matching as energy minimization







$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

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

Source: Steve Seitz

Better results…

Graph cut method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

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

Challenges

- Low-contrast ‘textureless’ image regions
- Occlusions
- Violations of brightness constancy
  - Specular reflections
- Really large baselines
  - Foreshortening and appearance change
- Camera calibration errors

SIFT + Fundamental Matrix + RANSAC + Sparse correspondence

SIFT + Fundamental Matrix + RANSAC + dense correspondence

Despite their scale invariance and robustness to appearance changes, SIFT features are *local* and do not contain any global information about the image or about the location of other features in the image. Thus feature matching based on SIFT features is still prone to errors. However, since we assume that we are dealing with rigid scenes, there are strong geometric constraints on the locations of the matching features and these constraints can be used to clean up the matches. In particular, when a rigid scene is imaged by two pinhole cameras, there exists a 3×3 matrix F, the *Fundamental matrix*, such that corresponding points x_{ij} and x_{ik} (represented in homogeneous coordinates) in two images j and k satisfy[10]:

$$x_{ij}^{\top} F x_{ij} = 0. \tag{3}$$

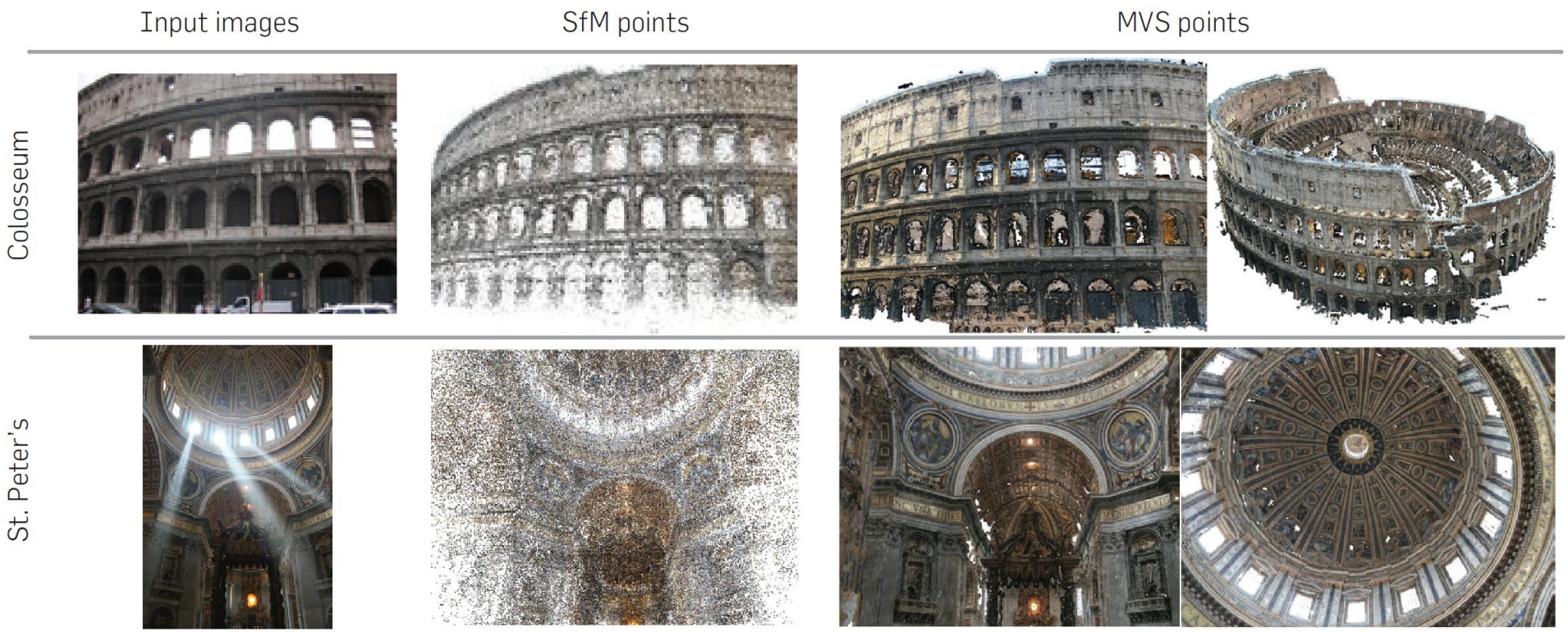
A common way to impose this constraint is to use a greedy randomized algorithm to generate suitably chosen random estimates of F and choose the one that has the largest support among the matches, i.e., the one for which the most matches satisfy (3). This algorithm is called Random Sample Consensus (RANSAC)[6] and is used in many computer vision problems.

Building Rome in a Day

By Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, Richard Szeliski
Communications of the ACM, Vol. 54 No. 10, Pages 105-112

2009

SIFT + Fundamental Matrix + RANSAC + dense correspondence



Building Rome in a Day

By Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, Richard Szeliski
Communications of the ACM, Vol. 54 No. 10, Pages 105-112

SIFT + Fundamental Matrix + RANSAC + dense correspondence