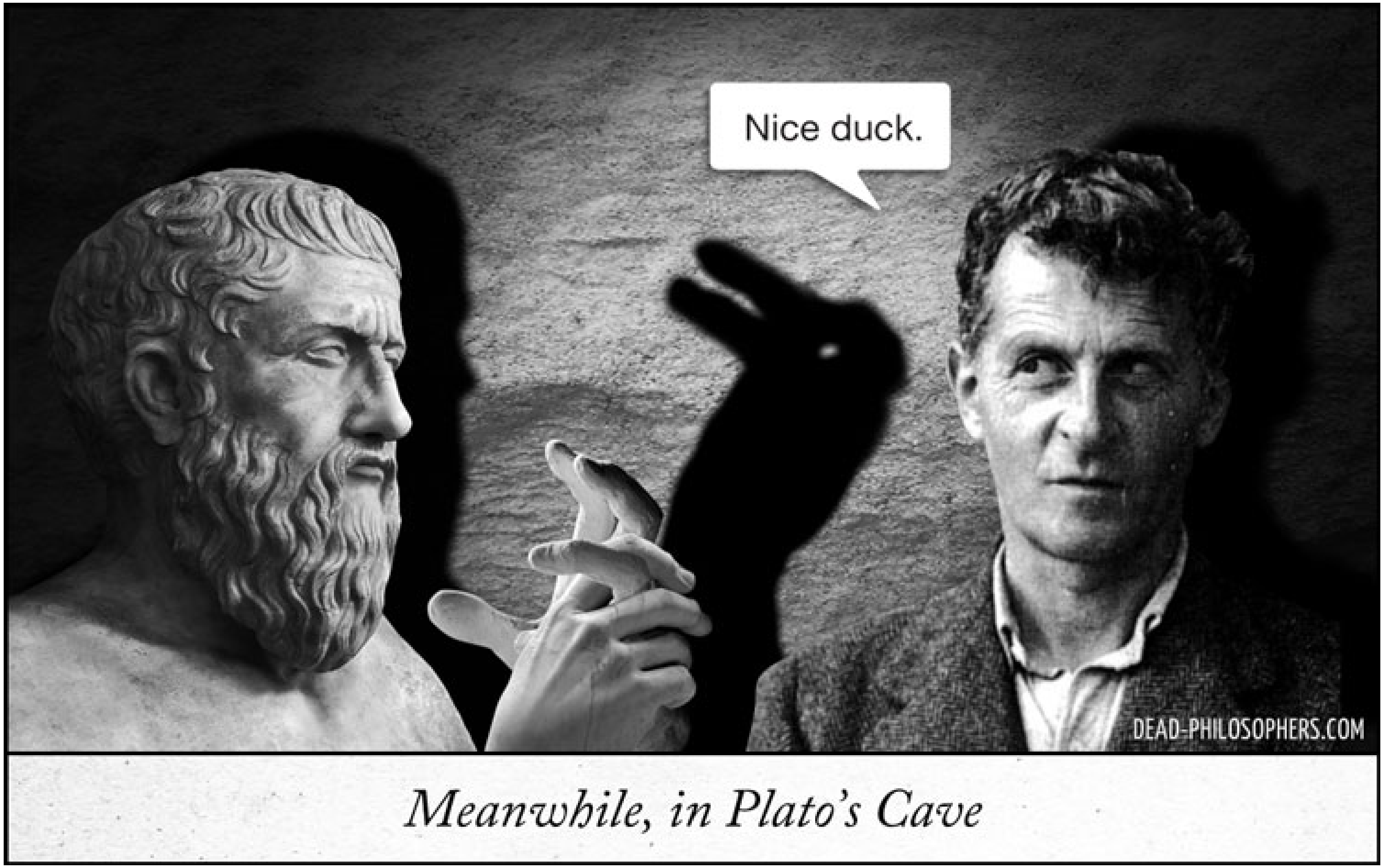
Lecture 15
3D and Stereo

COS 429: Computer Vision

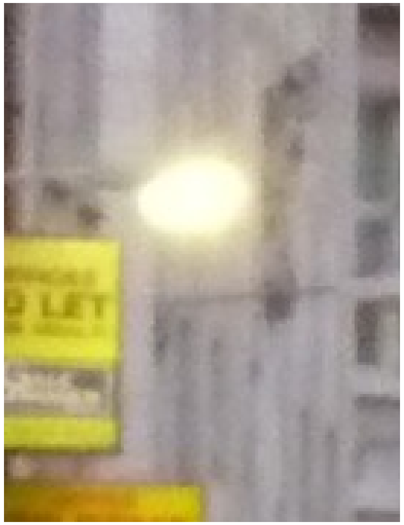
PRINCETON UNIVERSITY

Slides credit:
Many slides adapted from James Hays, Derek Hoeim, Lana Lazebnik, Silvio Saverse, who in turn adapted slides from Steve Seitz, Rick Szeliski, Martial Hebert, Mark Pollefeys, and others



Slide Credit:

Which is closer?

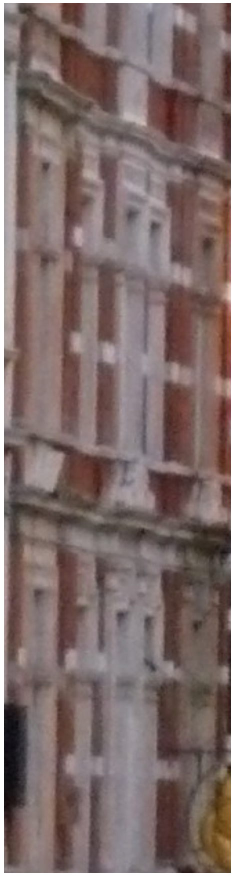


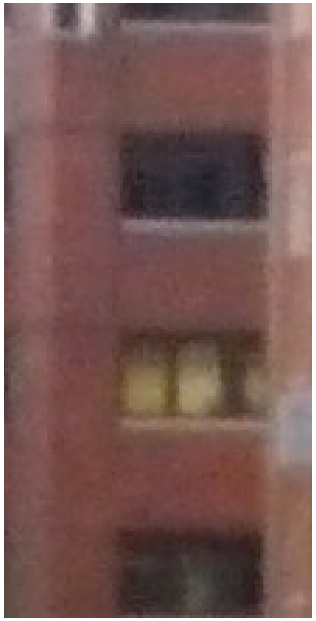
 Slide Credit:

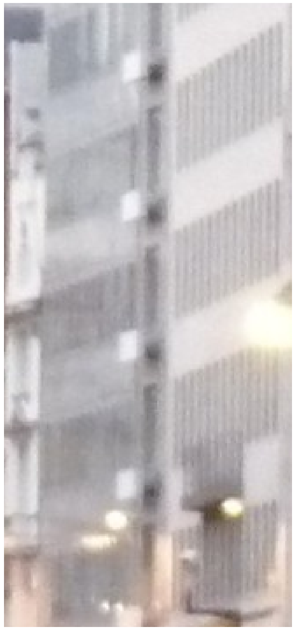
Which is closer?

Slide Credit:

The world is 3D







Slide Credit:

The world is 3D

Slide Credit:

Computer Vision: infer 3D from 2D

Slide Credit:

Why bother with 3D?

The world is 3D
- Compact representation of relationships
- Ability to navigate & manipulate

Some 2D vision problems are easier in 3D
- Occlusion
- Variation with lighting
- Variation with viewpoint
- Segmentation
- Recognition

Computer Vision: infer 3D from 2D

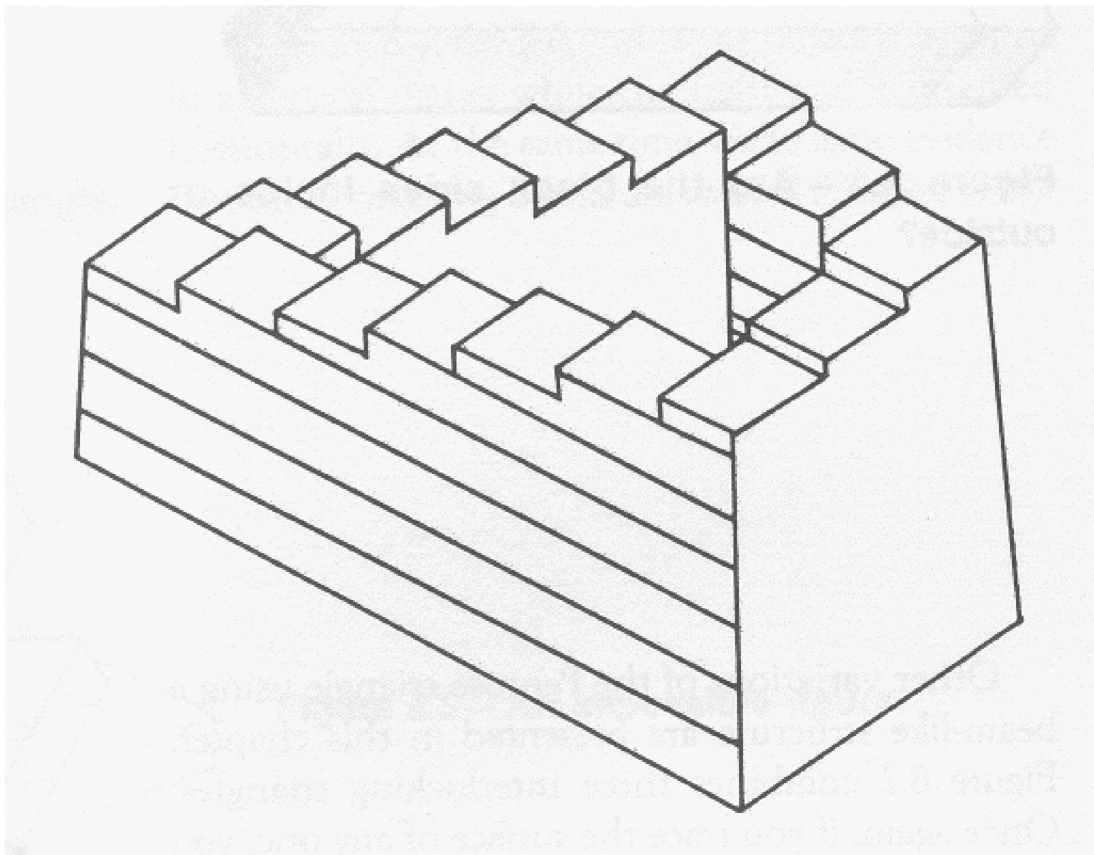
Slide Credit:

Monocular Image Depth Cues

- Occlusion (Interposition)
  - Near surfaces overlapping far ones
- Perspective
  - Parallel lines converging in the distance
- Texture gradient
  - Statistics of texture change (more details nearby)
- Size (relative, familiar, absolute)
  - Smaller objects, especially when known, appear farther
- Relative Position (Elevation)
  - Higher object tend to be farther
- Focus
  - Some depths are less in focus (could be near or far)
- In-Scattering (haze)
  - Far object have lower contrast (look hazy)
- ...

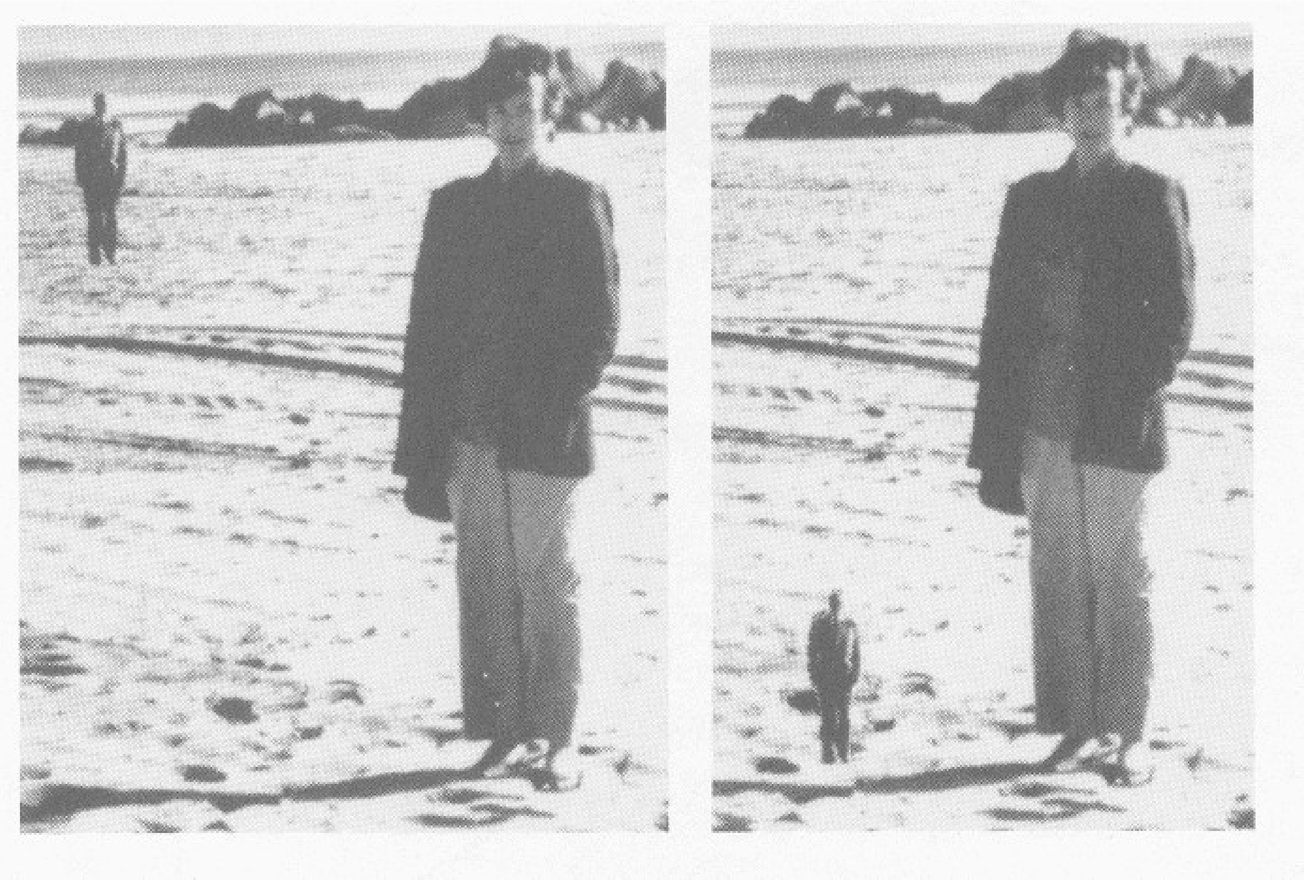
Slide Credit:

3D Perception: Illusions



Slide Credit: Block & Yuker

3D Perception: Illusions

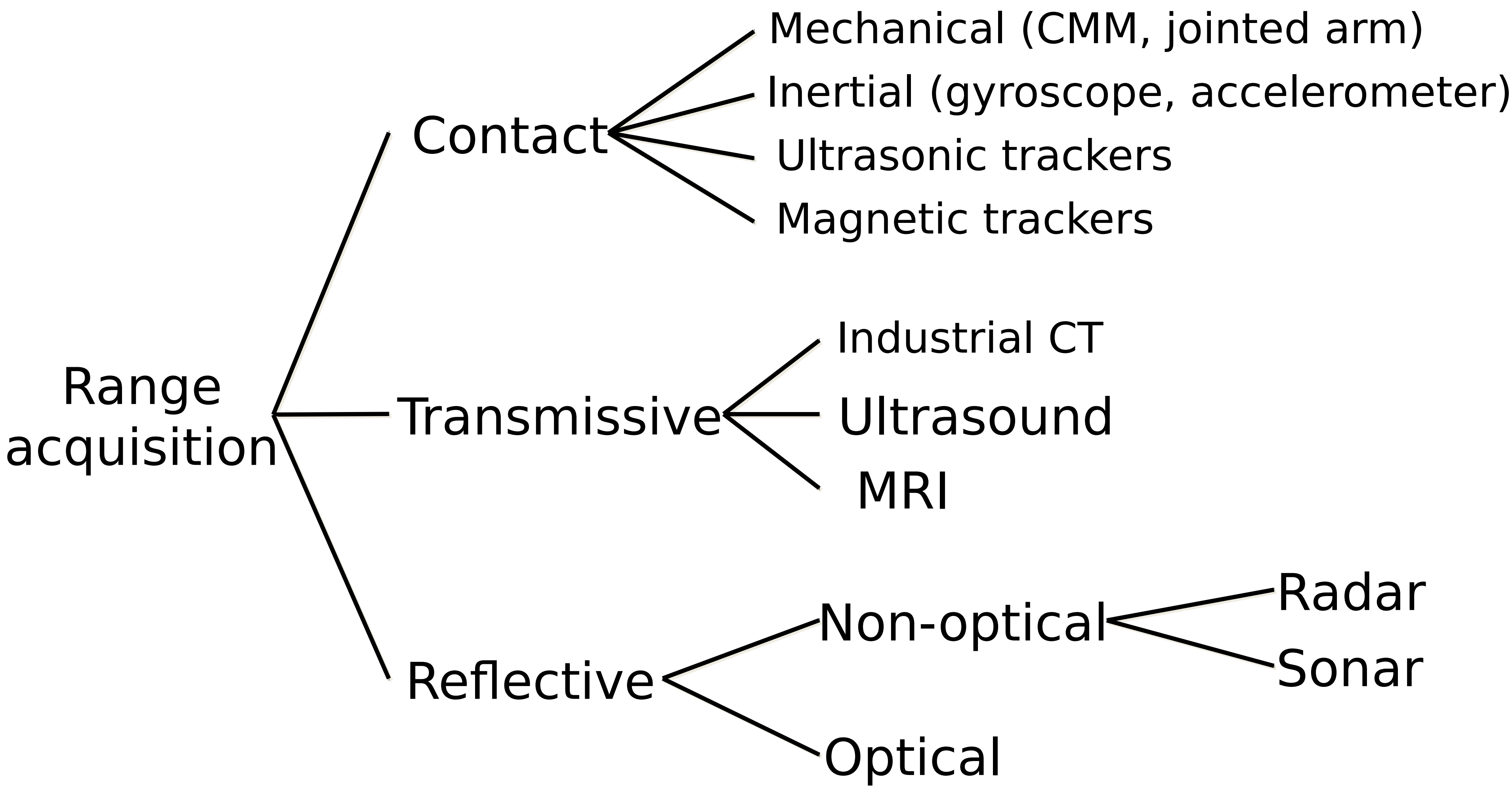


Slide Credit: Block & Yuker

3D Perception: Illusions

Slide Credit: Block & Yuker

Range Acquisition Taxonomy



Touch Probes

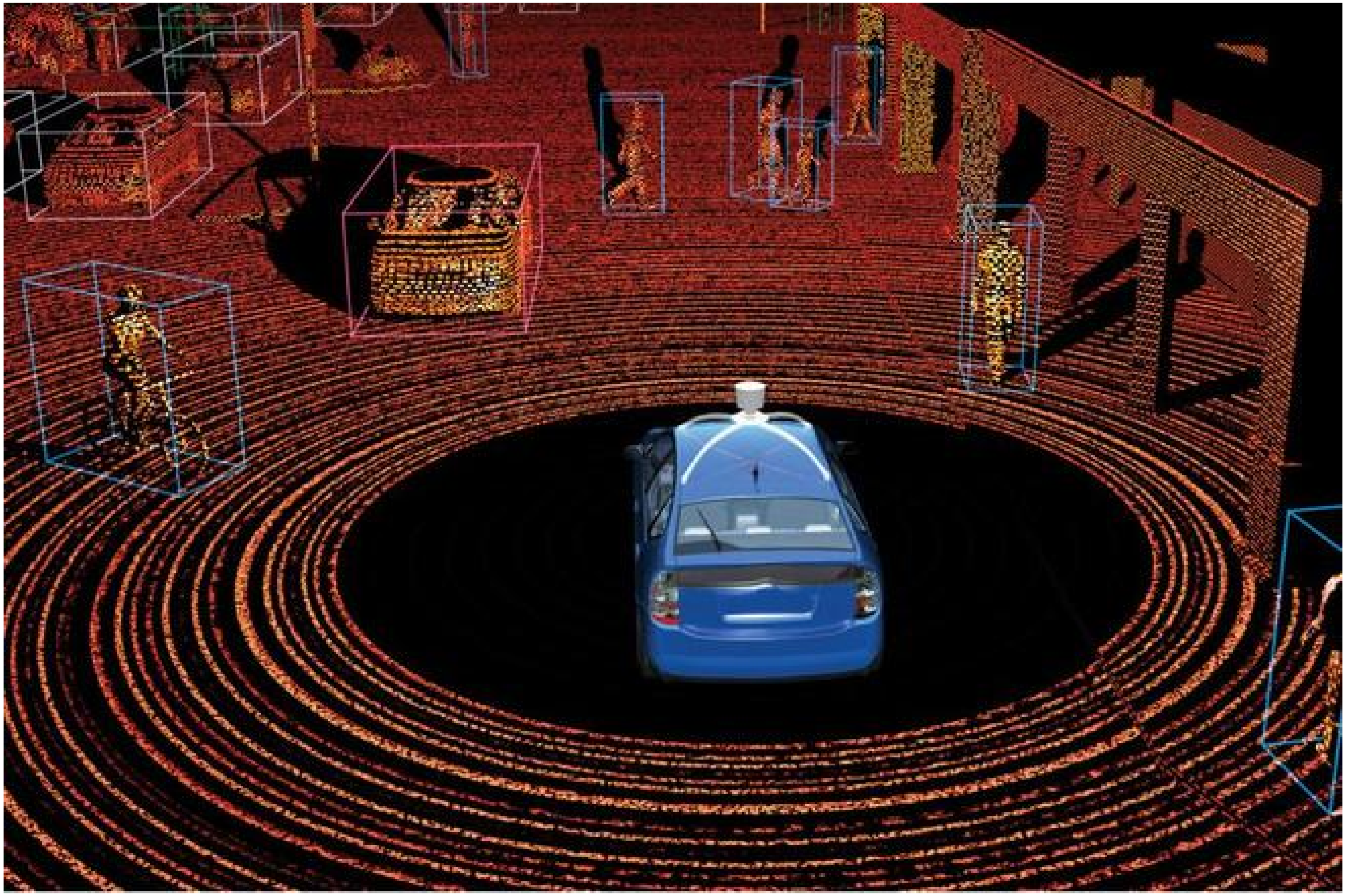
- Jointed arms with angular encoders
- Return position, orientation of tip

Faro Arm – Faro Technologies, Inc.

Range Acquisition Taxonomy

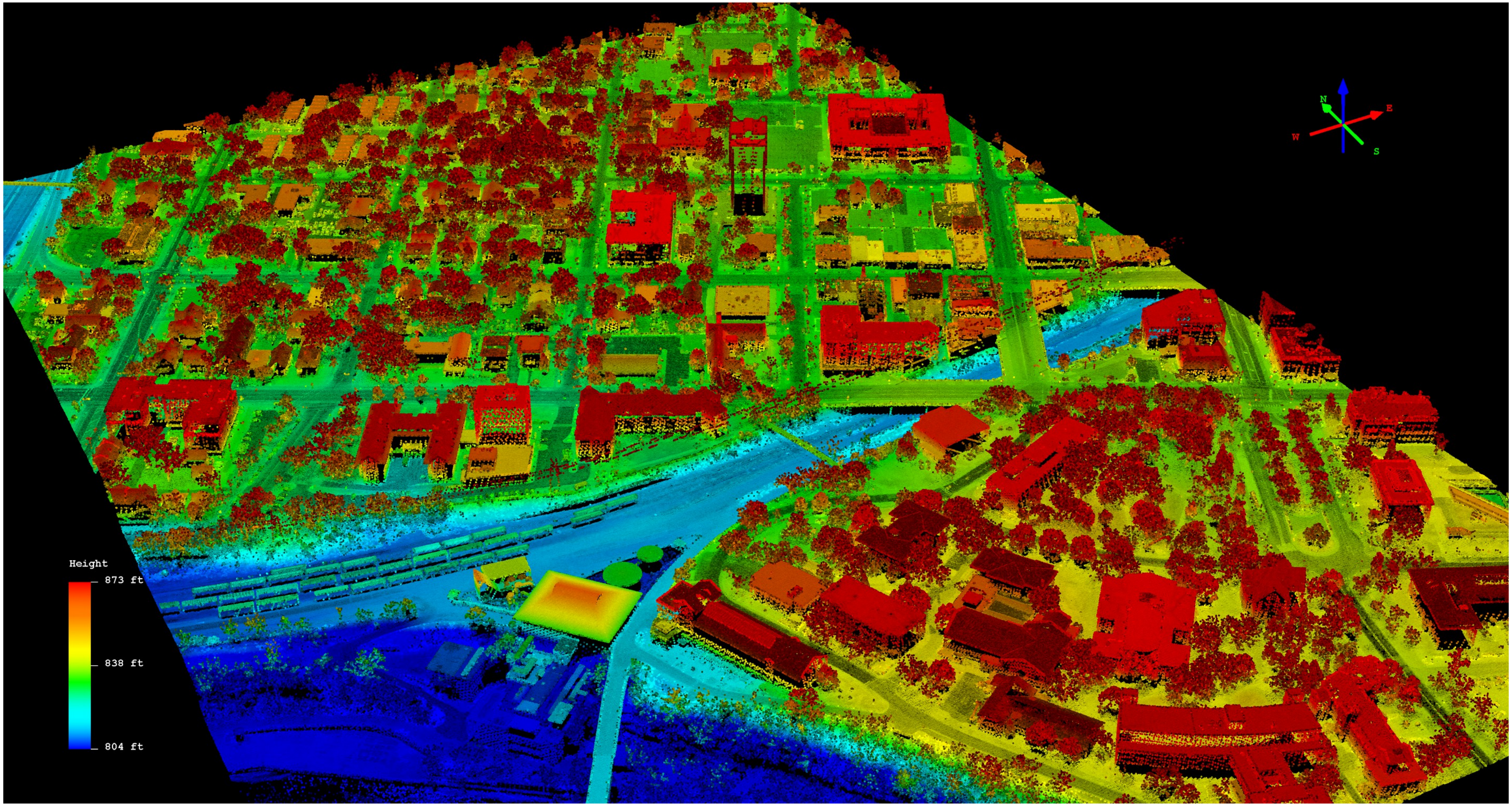
- Optical methods
  - Passive
    - Shape from X:
      - stereo
      - motion
      - shading
      - texture
      - focus
      - haze
  - Active
    - Active variants of passive methods
      - Stereo w. projected texture
      - Active depth from defocus
      - Photometric stereo
    - Time of flight
    - Triangulation

Lidar



Slide Credit:

Lidar



Slide Credit:

Optical Range Acquisition

- Advantages:
  - Non-contact
  - Safe
  - Usually fast
- Disadvantages:
  - Sensitive to transparency
  - Confused by specularity and interreflection
  - Texture (helps some methods, hurts others)

Passive Optical Range Acquisition

- Advantages:
  - Very Dense (high resolution)
  - Does not interfere with environment
  - Inexpensive
- Disadvantages:
  - Heavy Processing (CPU time)
  - Only works on textured regions
  - Depth accuracy depends on baseline

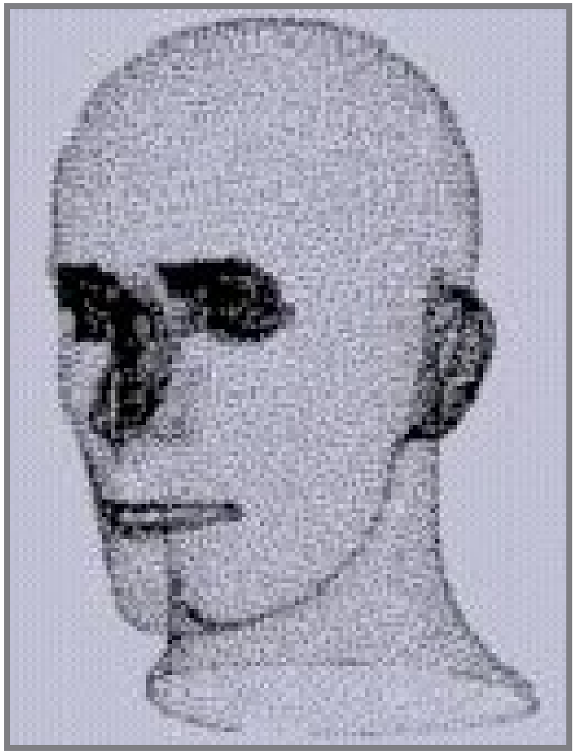
3D Data Types

How do we represent the 3D world?

- Point Data
- Volumetric Data
- Surface Data

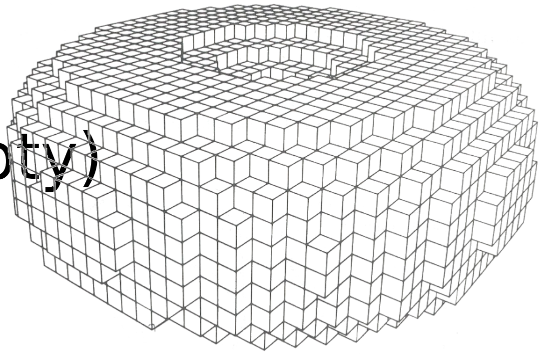
3D Data Types: Point Data

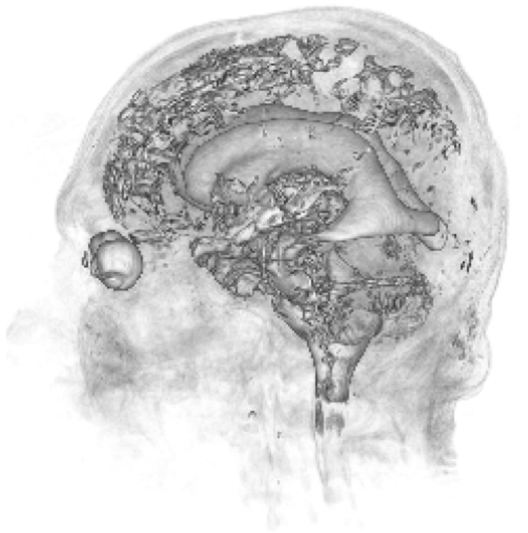
- "Point clouds"
- Advantage: simplest data type
- Disadvantage: no information adjacency / connectivity



3D Data Types: Volumetric Data

- Regularly-spaced grid in (x,y,z): “voxels”
- For each grid cell, store
  - Occupancy (binary: occupied / empty)
  - Density
  - Other properties
- Popular in medical imaging
  - CAT scans
  - MRI

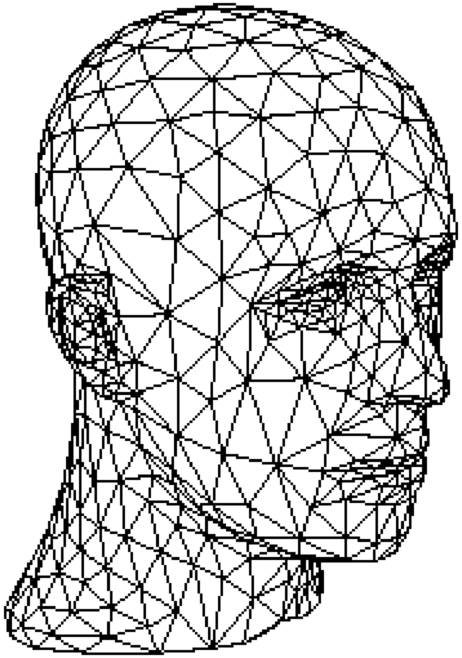




3D Data Types: Volumetric Data

- Advantages:
  - Can represent inside of object
  - Uniform sampling: simpler algorithms
- Disadvantages:
  - Lots of data
  - Wastes space if only storing a surface
  - Most “vision” sensors / algorithms return point or surface data

3D Data Types: Surface Data



- Polyhedral
  - Piecewise planar
  - Polygons connected together
  - Most popular: “triangle meshes”
- Smooth
  - Higher-order (quadratic, cubic, etc.) curves
  - Bézier patches, splines, NURBS, subdivision surfaces, etc.
  - See COS 426 for details…

3D Data Types: Surface Data

- Advantages:
  - Usually corresponds to what we see
  - Usually returned by vision sensors / algorithms
- Disadvantages:
  - How to find “surface” for translucent objects?
  - Parameterization often non-uniform
  - Non-topology-preserving algorithms difficult

2½-D Data

- Image: stores an intensity / color along each of a set of regularly-spaced rays in space
- Range image: stores a depth along each of a set of regularly-spaced rays in space
- Not a complete 3D description: does not store objects occluded (from some viewpoint)
- View-dependent scene description

2½-D Data

- This is what most sensors / algorithms really return
- Advantages
  - Uniform parameterization
  - Adjacency / connectivity information
- Disadvantages
  - Does not represent entire object
  - View dependent

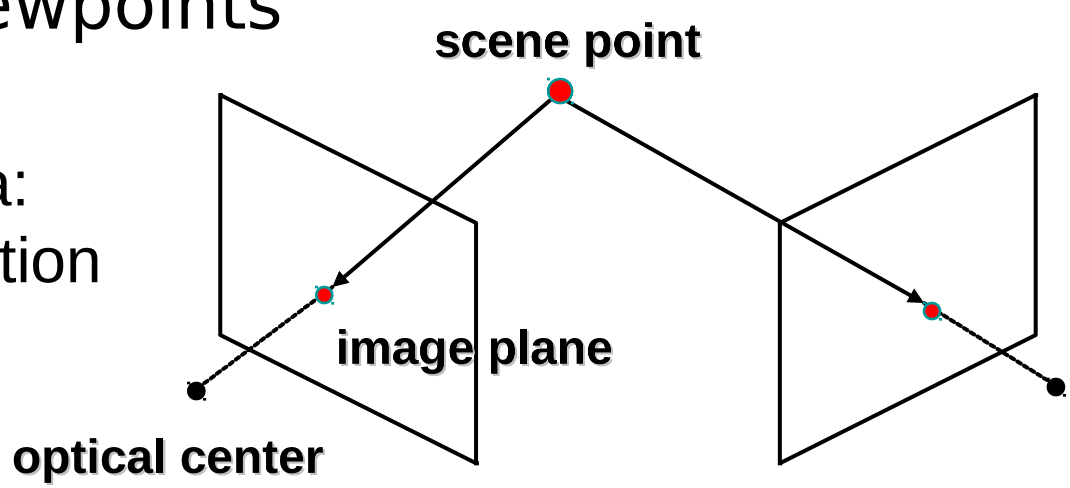
2½-D Data

- RGBD
- Range images
- Range surfaces
- Depth images
- Depth maps
- Height fields
- 2½-D images
- Surface profiles
- *xyz* maps
- ...

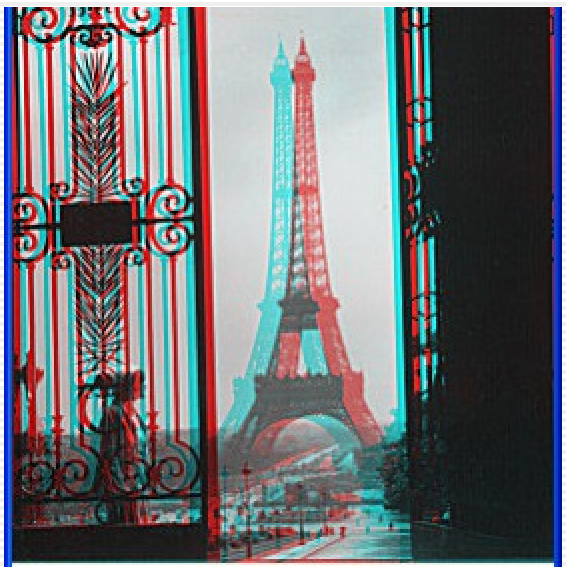
Stereo Image Matching

- Passive Optical Depth Methods (aka. “Shape from X”): Shading, Texture, Focus, Motion…
- **Stereo**:
  - shape from “motion” between two views
  - infer 3d shape of scene from two (multiple) images from different viewpoints

Main idea:
Triangulation



Slide Credit: James Hays



© Copyright 2001 Johnson-Shaw Stereoscopic Museum

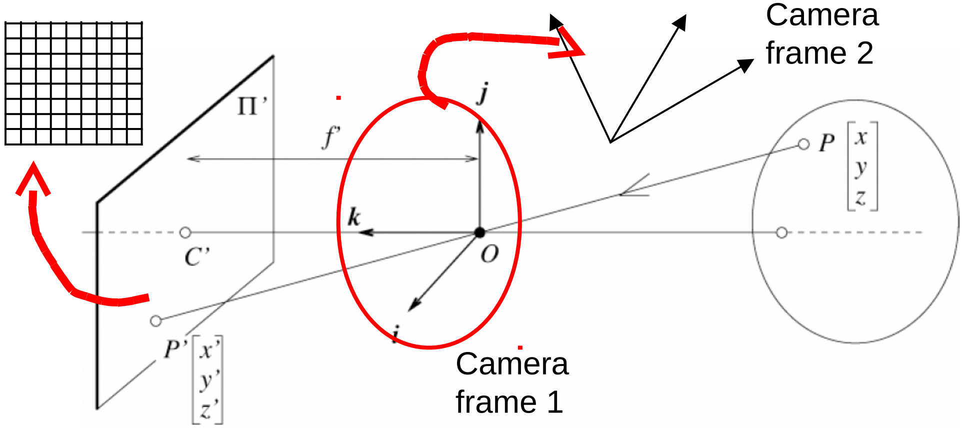
http://www.johnsonshawmuseum.org

Slide Credit: James Hays

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

Slide Credit: James Hays

Camera parameters



Extrinsic parameters:
Camera frame 1 ←→ Camera frame 2

Intrinsic parameters:
Image coordinates relative to camera ←→ Pixel coordinates

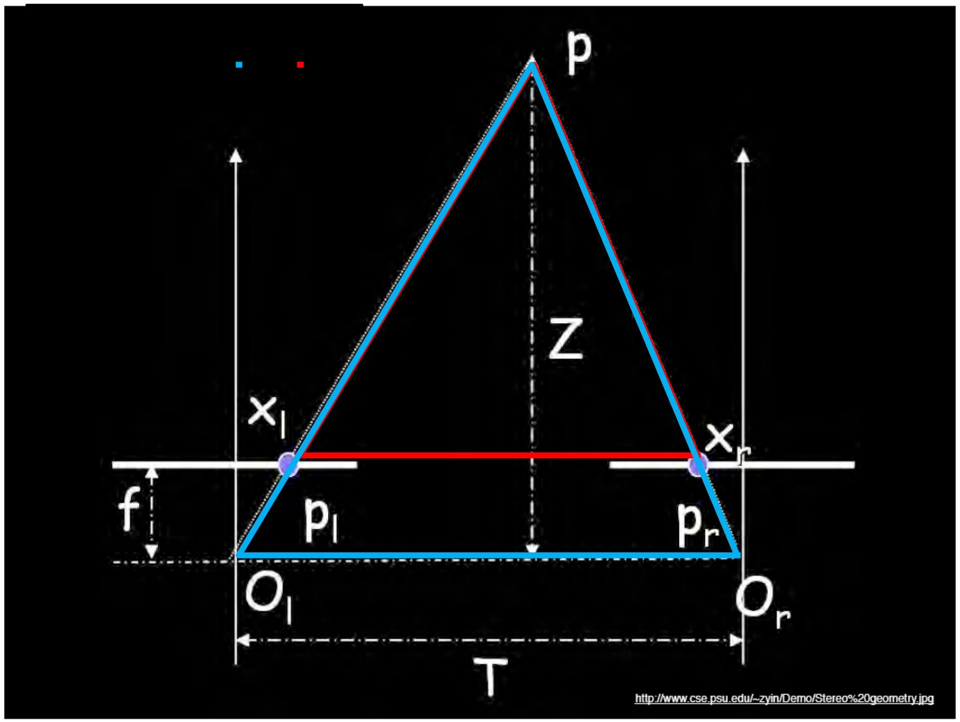
- *Extrinsic* params: rotation matrix and translation vector
- *Intrinsic* params: focal length, pixel sizes (mm), image center point, radial distortion parameters

We'll assume for now that these parameters are given and fixed.

Slide Credit: James Hays

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). **What is**



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

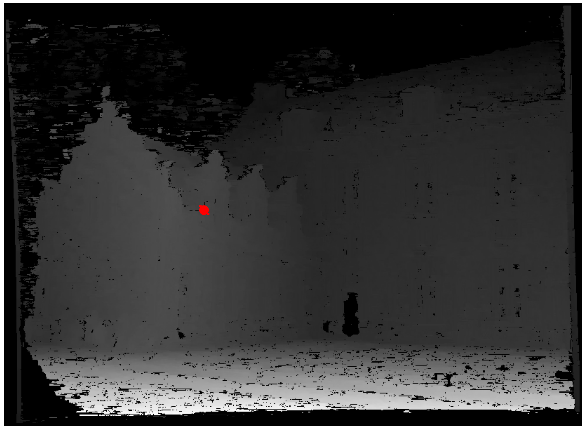
 Slide Credit: James Hays

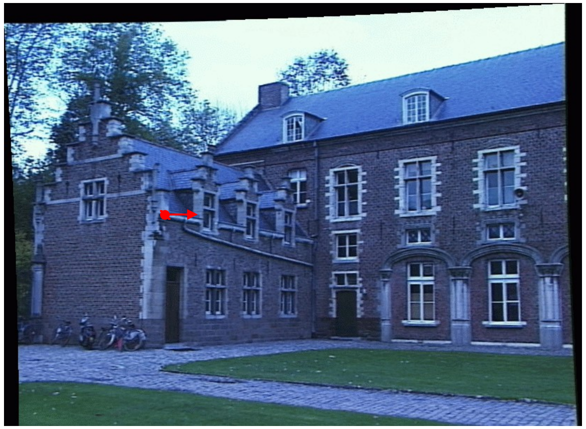
Depth from disparity

image I(x,y)

Disparity map D(x,y)

image I´(x´,y´)



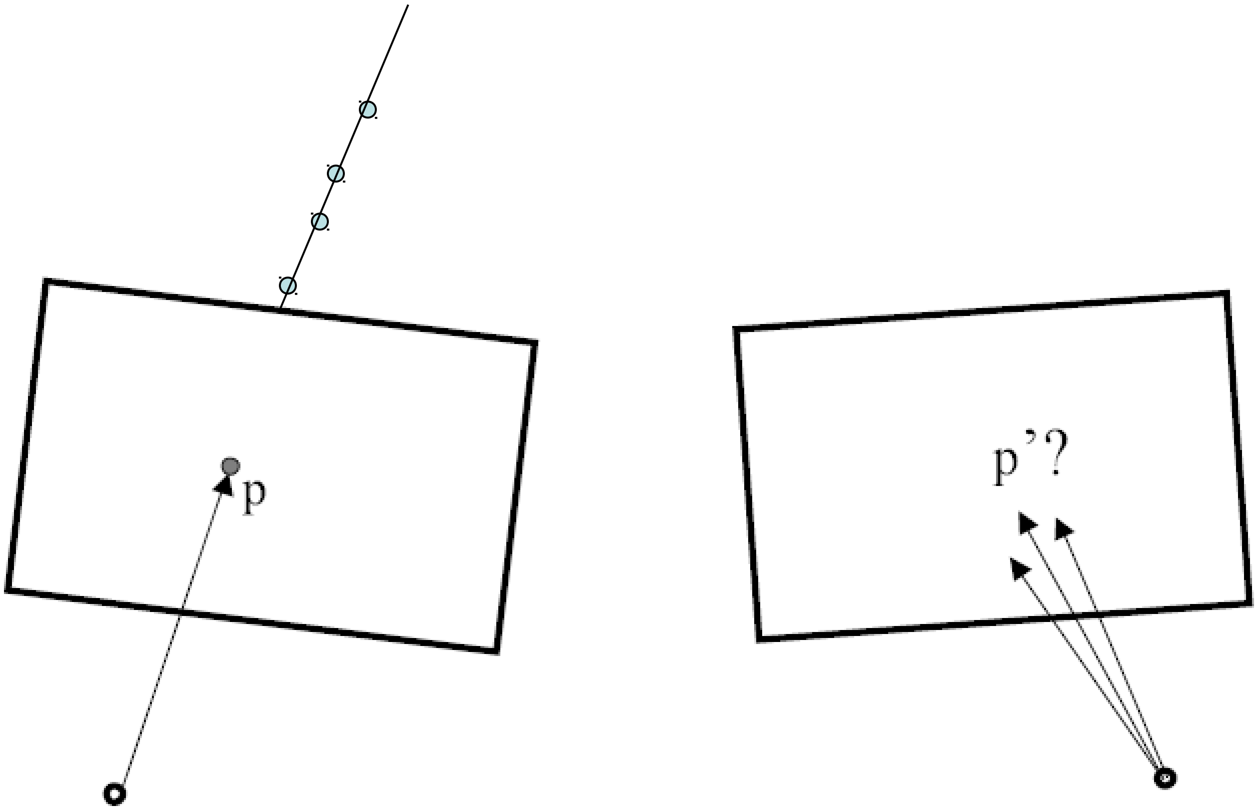


$$(x', y') = (x + D(x,y), y)$$

So if we could find the **corresponding points** in two images, we could **estimate relative depth**…

Slide Credit: James Hays

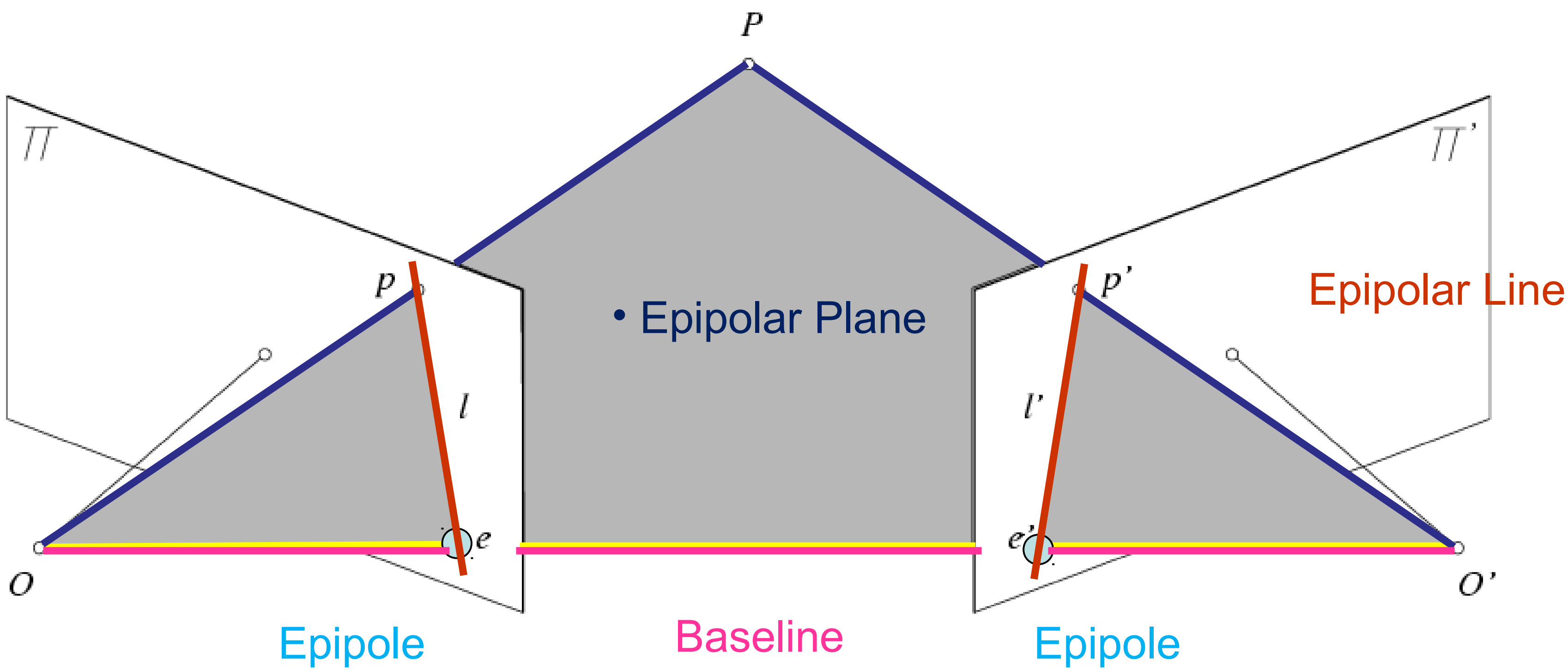
Stereo correspondence constraints



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

Slide Credit: James Hays

Epipolar geometry



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

Slide Credit: James Hays

Example

 Slide Credit: James Hays

Example: converging cameras

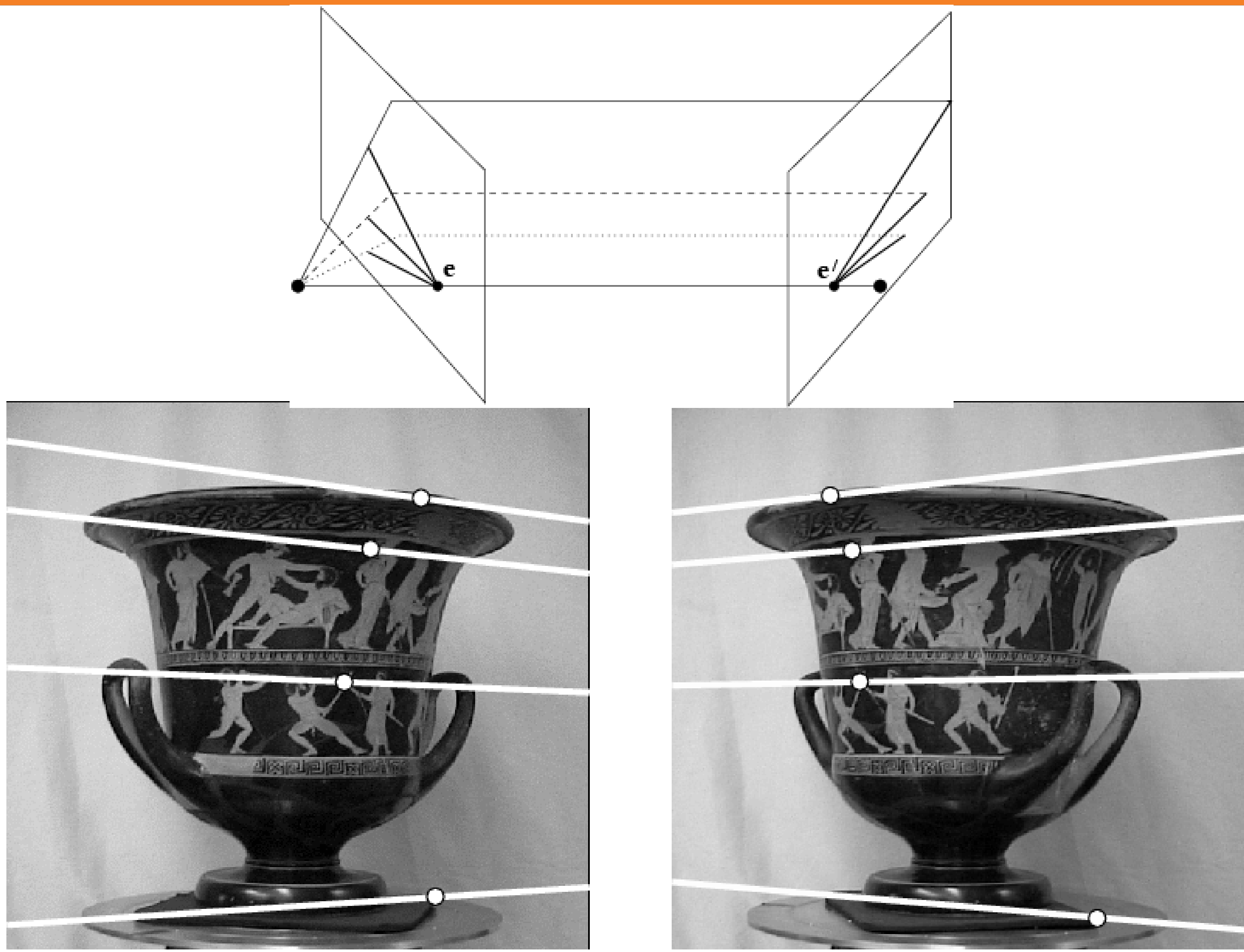
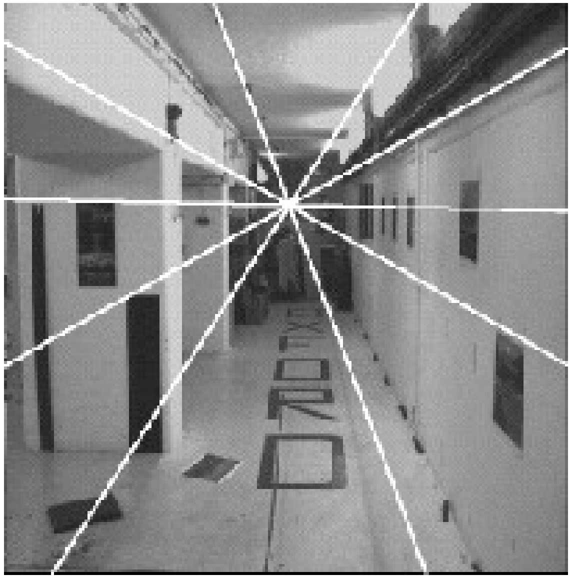
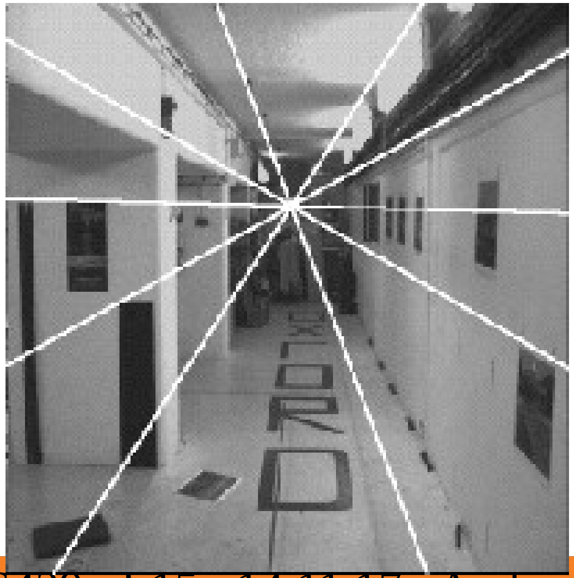


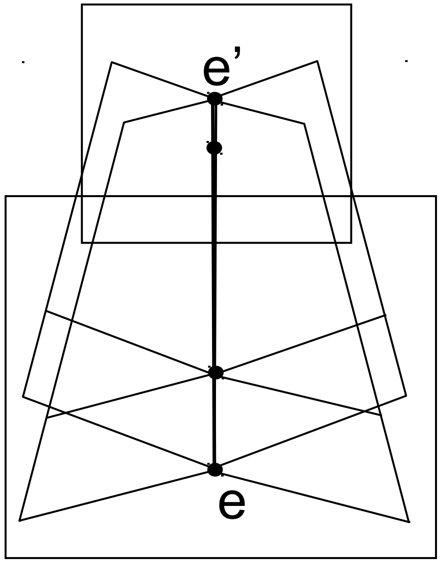
Figure from Hartley & Zisserman

Slide Credit: James Hays

Example: Forward motion





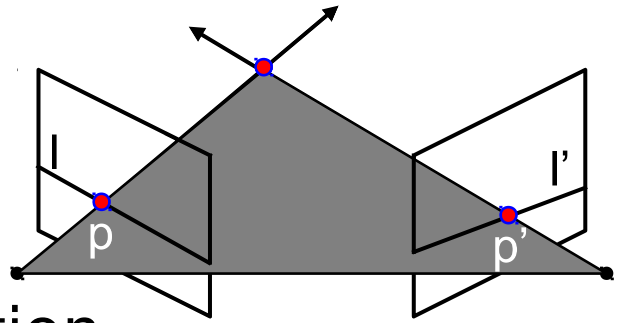


Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Slide Credit: James Hays

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

$$p'Fp = 0$$

Slide Credit: James Hays

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

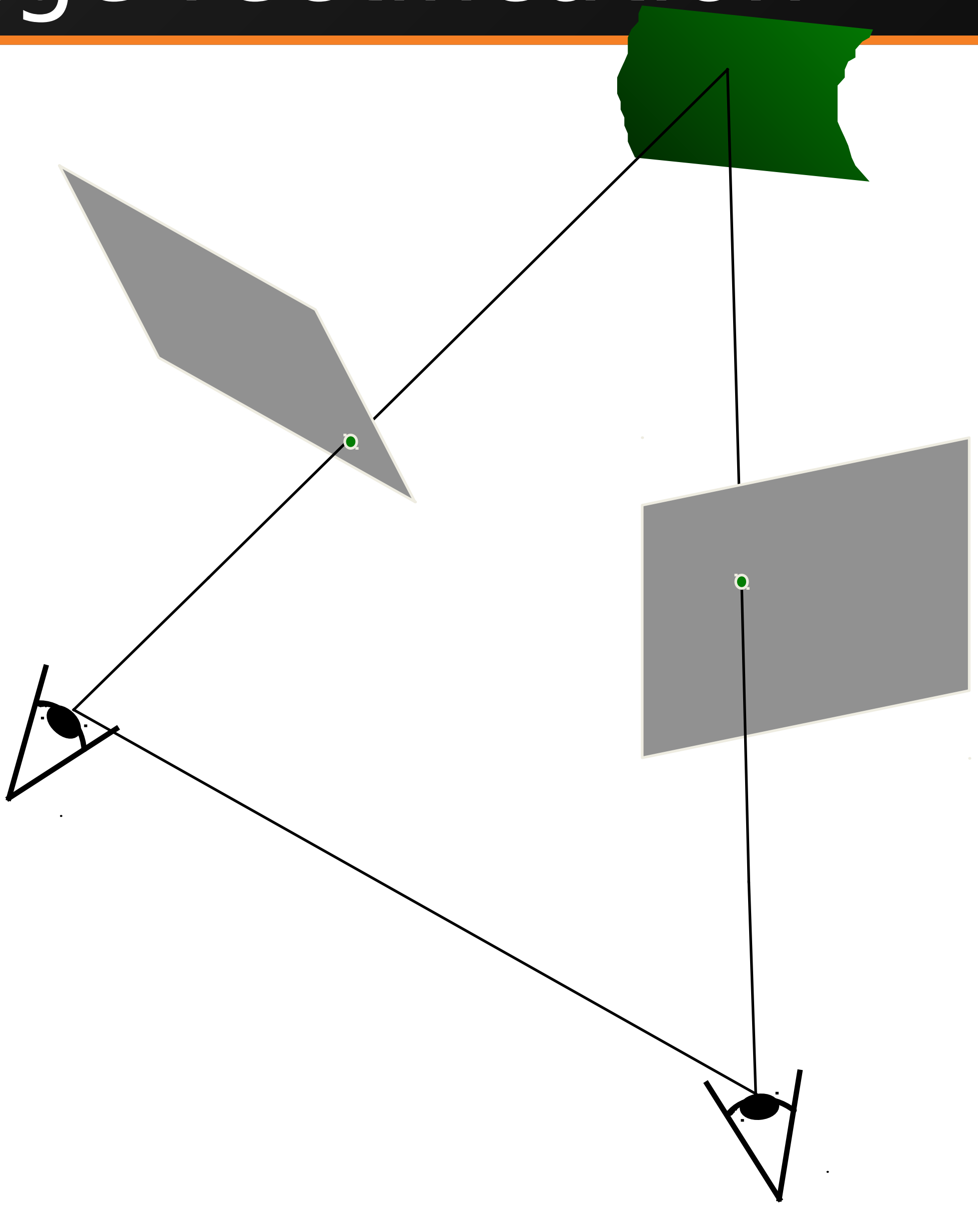
- Each (p, p') pair gives one linear equation in entries of F

$$p'Fp = 0$$

- 8 points give enough to solve for F (8-point algorithm)
- see Marc Pollefey's notes for a nice tutorial

 Slide Credit: James Hays

Stereo image rectification

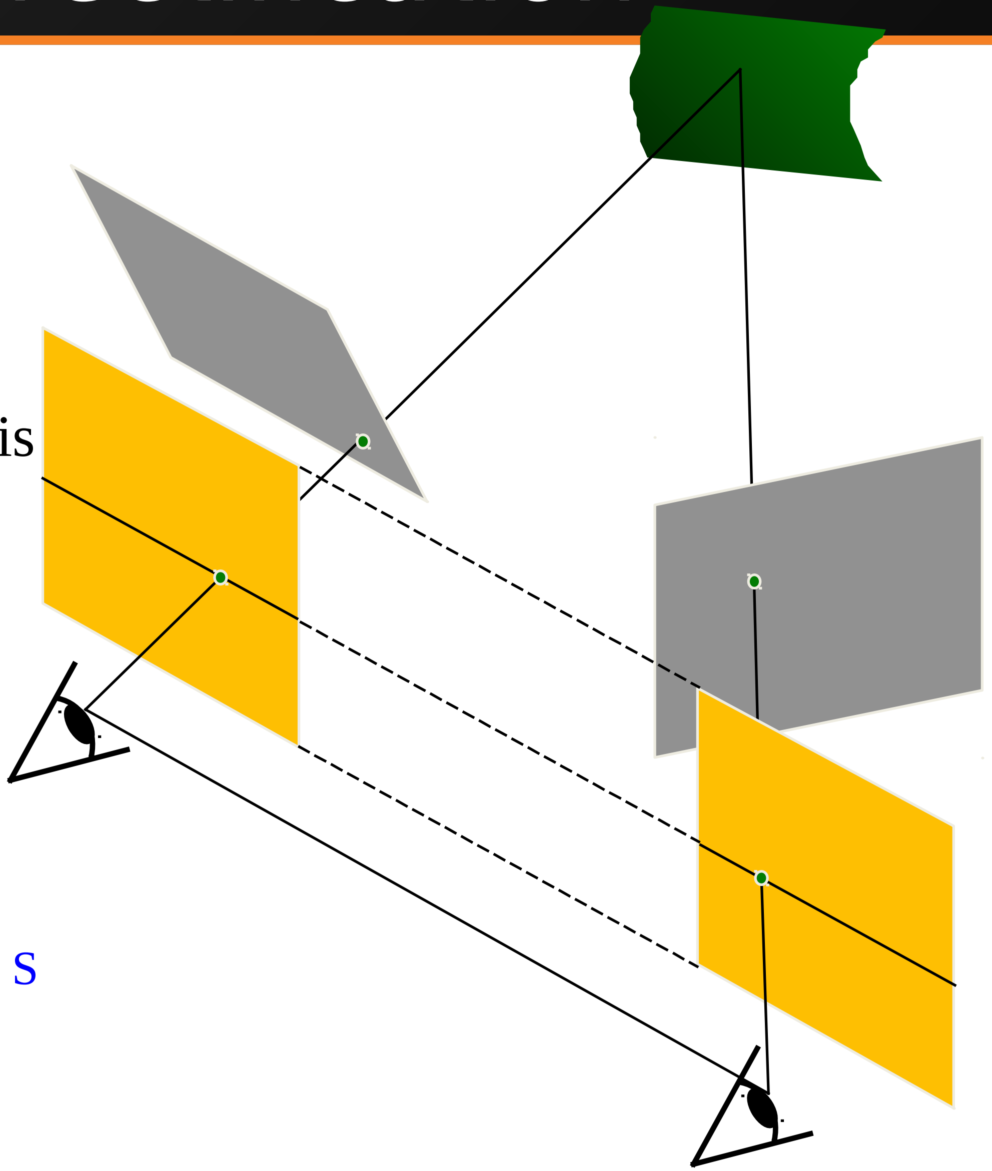


Slide Credit: James Hays

Stereo image rectification

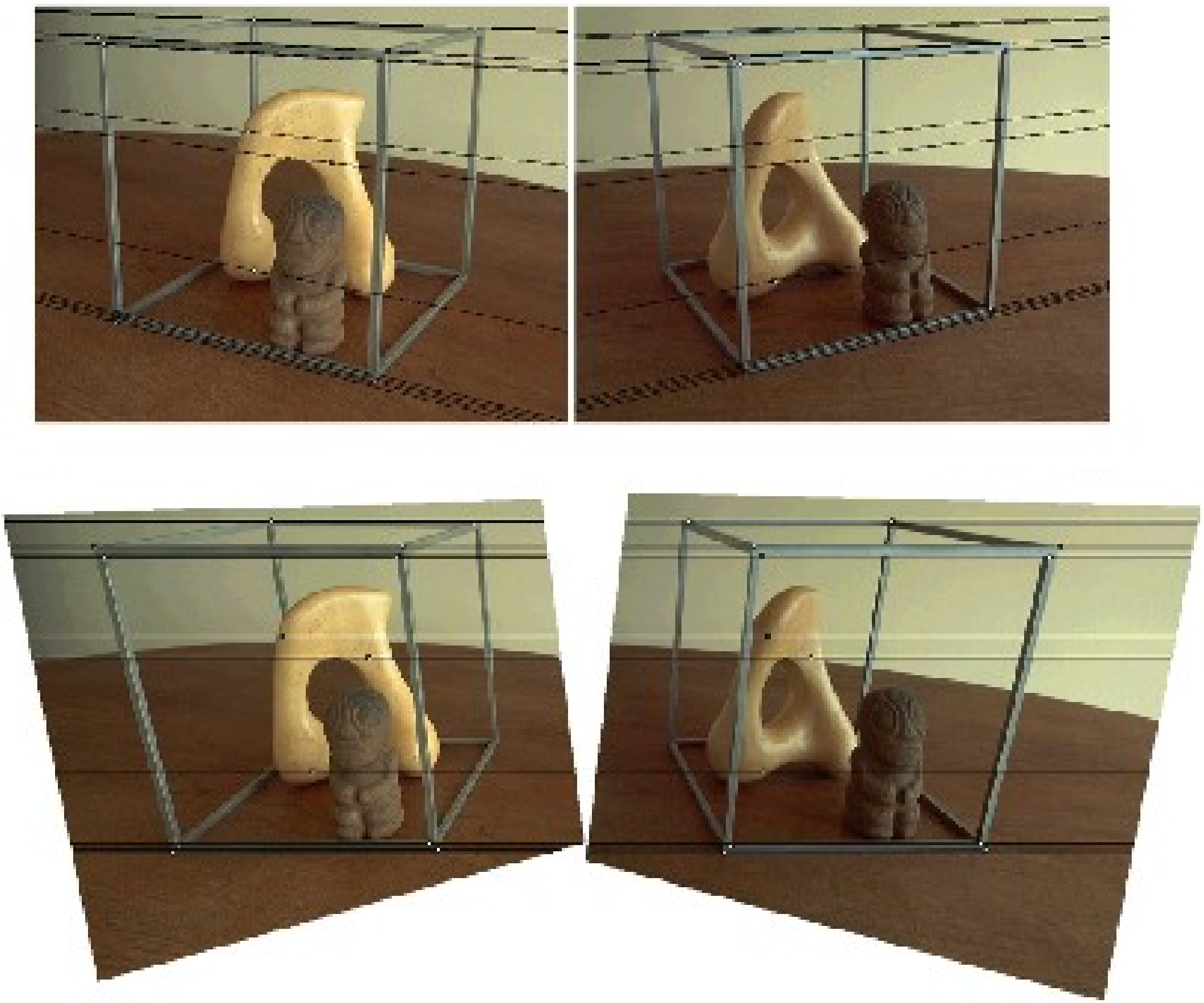
- Reproject image planes onto a common plane parallel to the line between camera centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

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



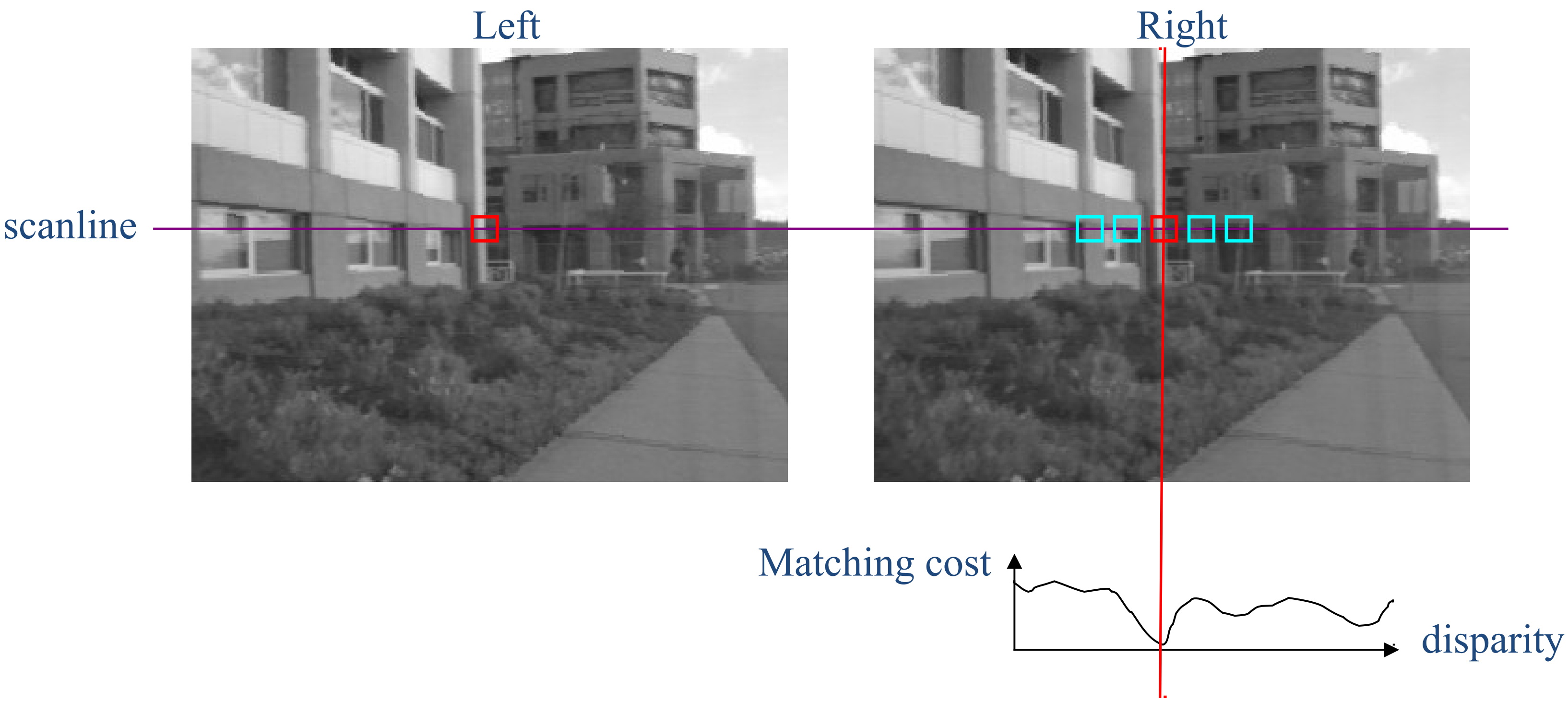
Slide Credit: James Hays

Rectification



Slide Credit: James Hays

Correspondence search

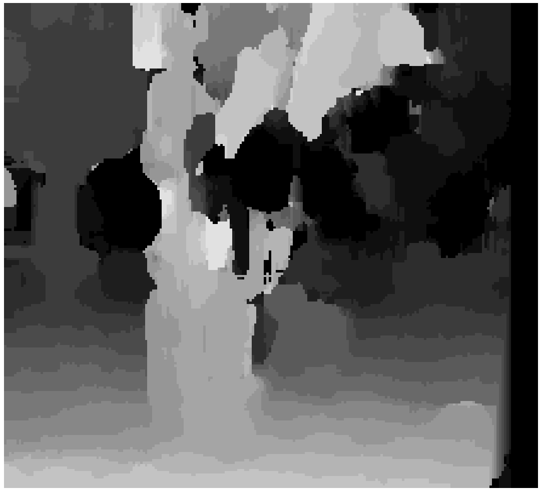


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

 Slide Credit: James Hays

Effect of window size

W = 3



W = 20

- Smaller window
  - \+ More detail
  - – More noise

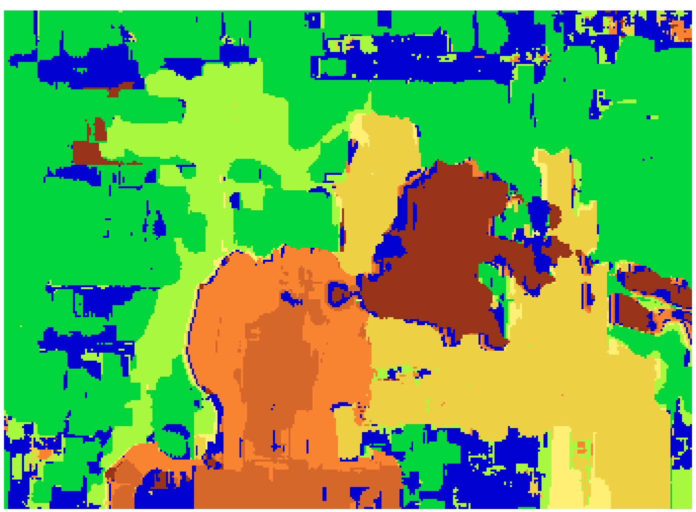
- Larger window
  - \+ Smoother disparity maps
  - – Less detail

Slide Credit: James Hays

Results with window search

Data

Window-based matching



Ground truth

Slide Credit: James Hays

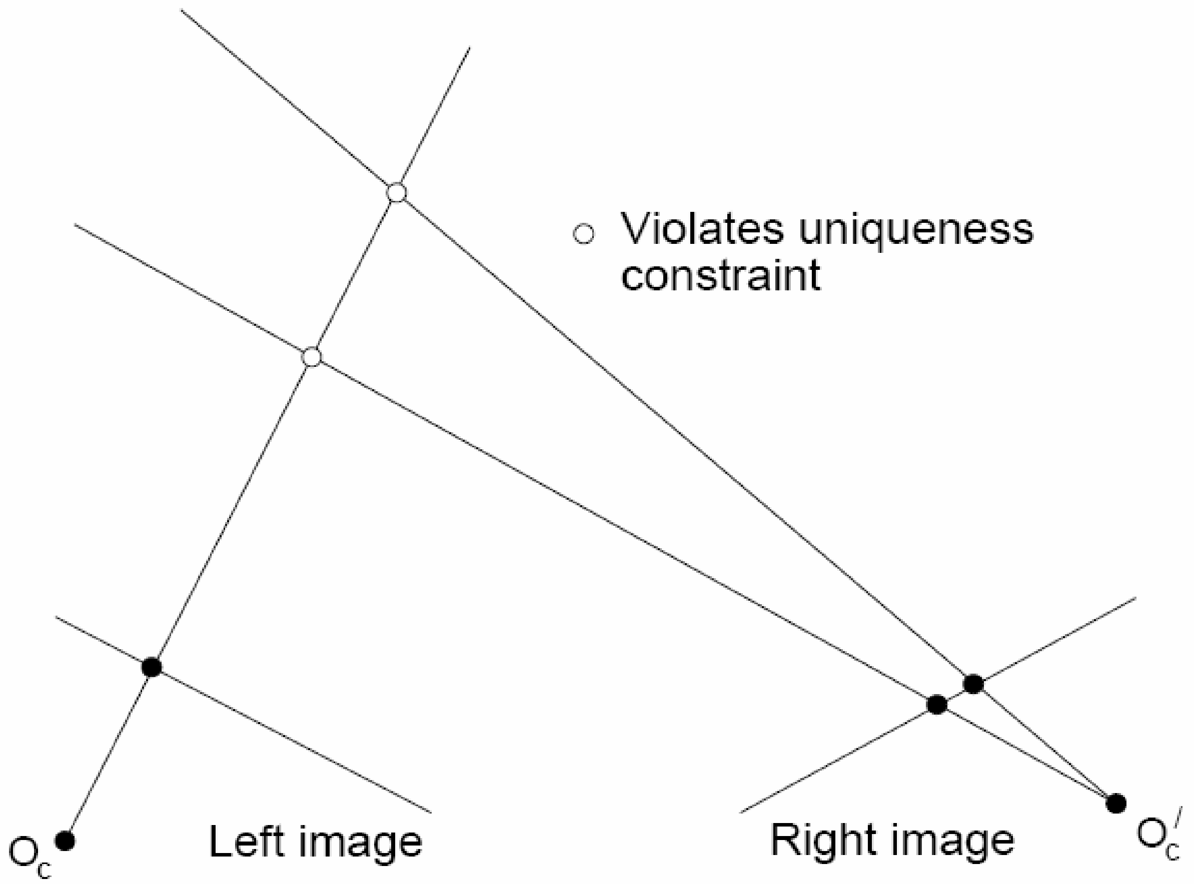
How can we improve window-based matching?

- So far, matches are independent for each point
- What constraints or priors can we add?

 Slide Credit: James Hays

Stereo constraints/priors

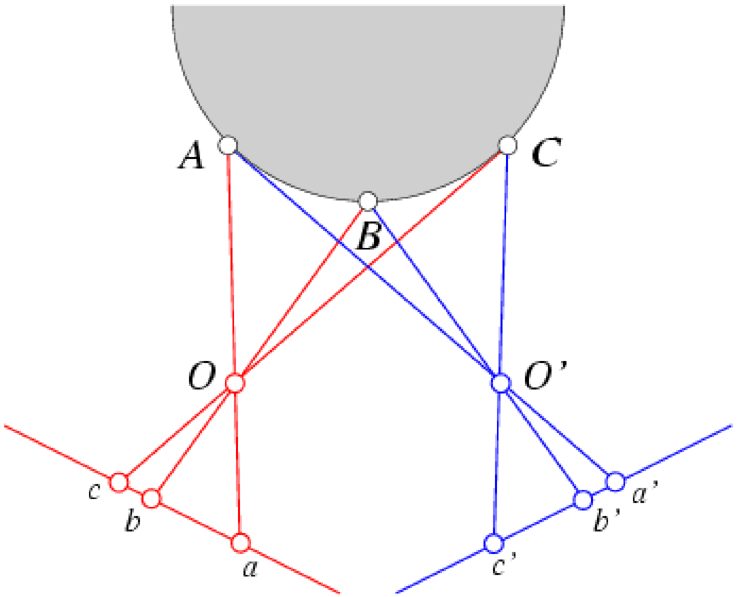
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image



Slide Credit: James Hays

Stereo constraints/priors

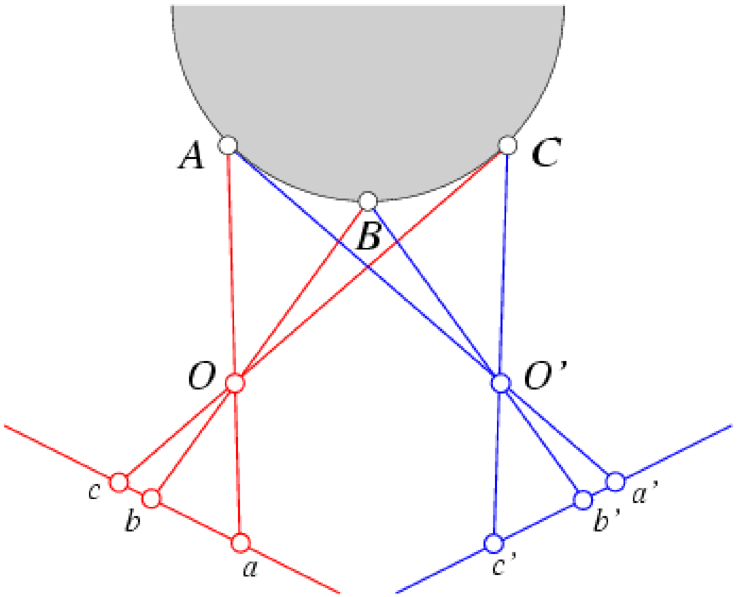
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views

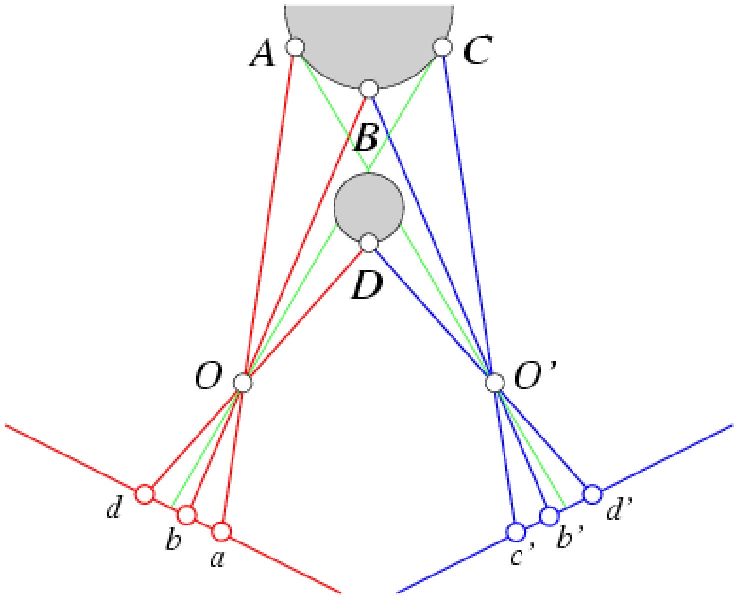


Slide Credit: James Hays

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

Slide Credit: James Hays

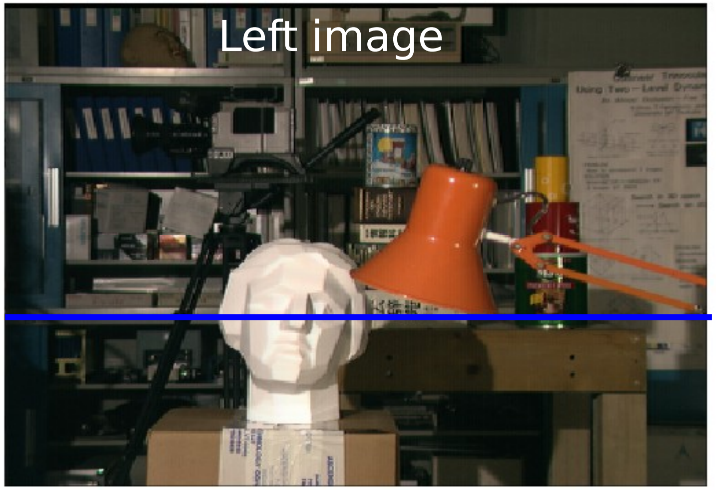
Priors and constraints

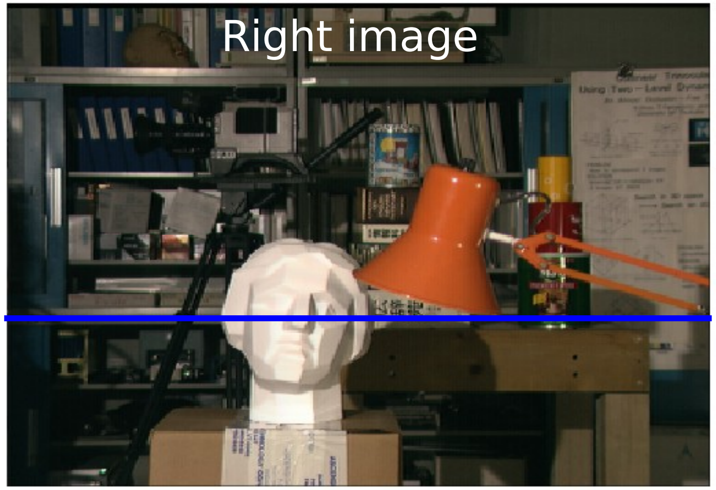
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views
- Smoothness
  - We expect disparity values to change slowly (for the most part – with a small sparse set of discontinuities)

 Slide Credit: James Hays

Scanline stereo

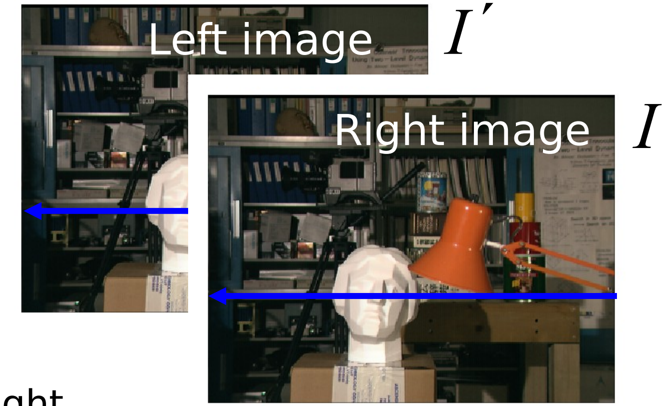
- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

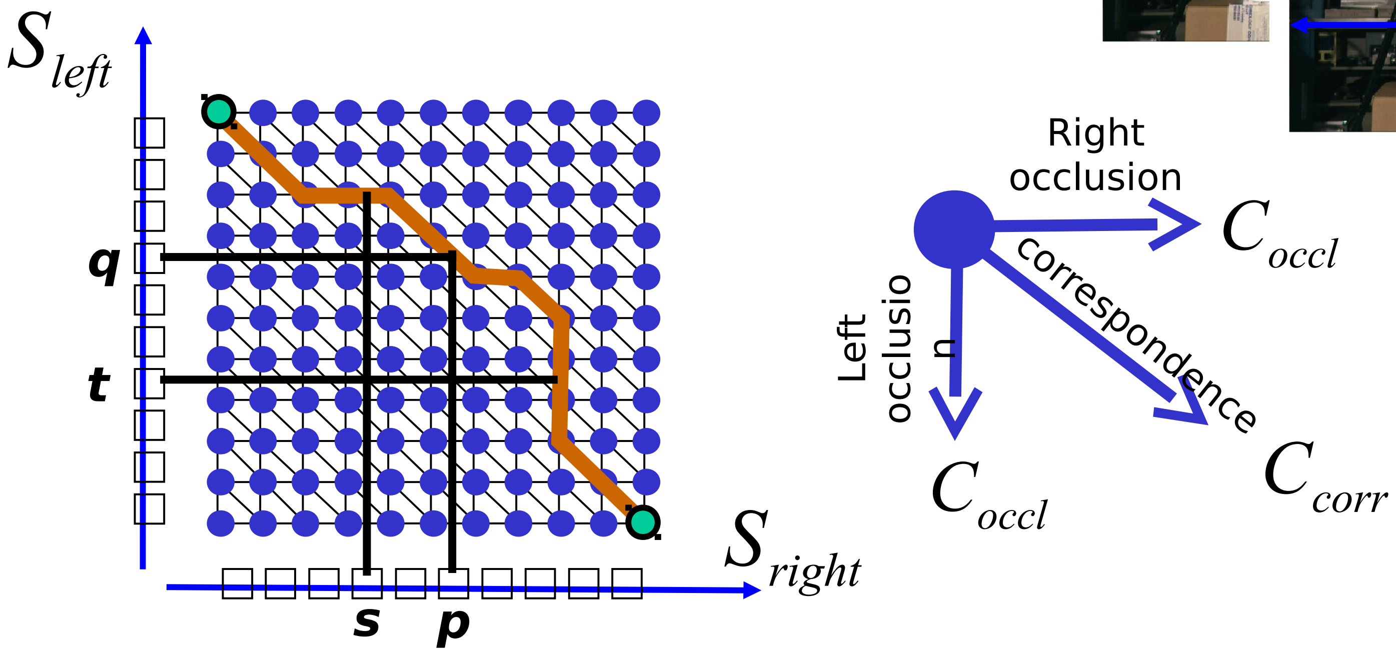




Slide Credit: James Hays

"Shortest paths" for scan-line stereo





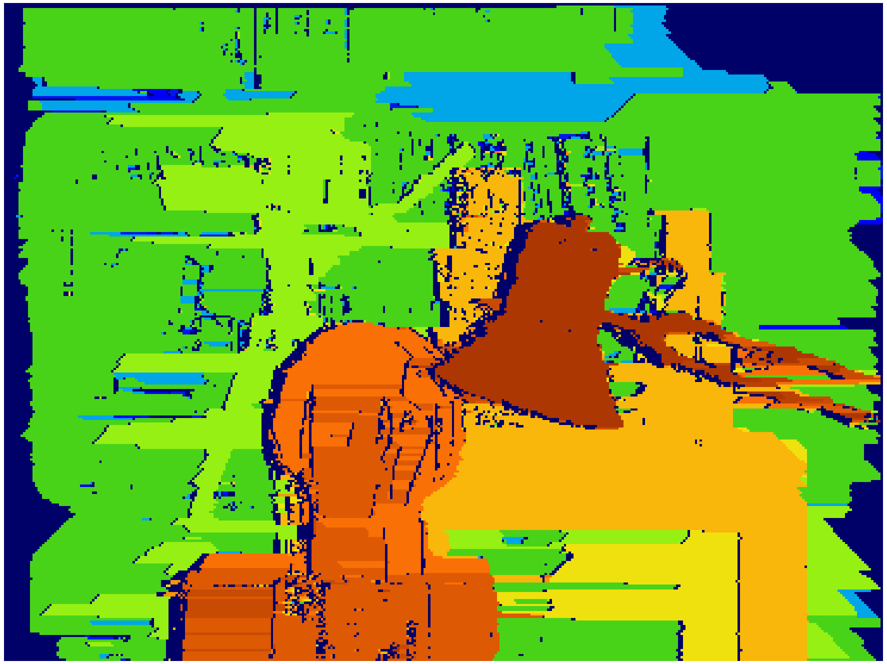
Can be implemented with dynamic programming

Ohta & Kanade '85, Cox et al. '96

Slide Credit: Y. Boykov

Coherent stereo on 2D grid

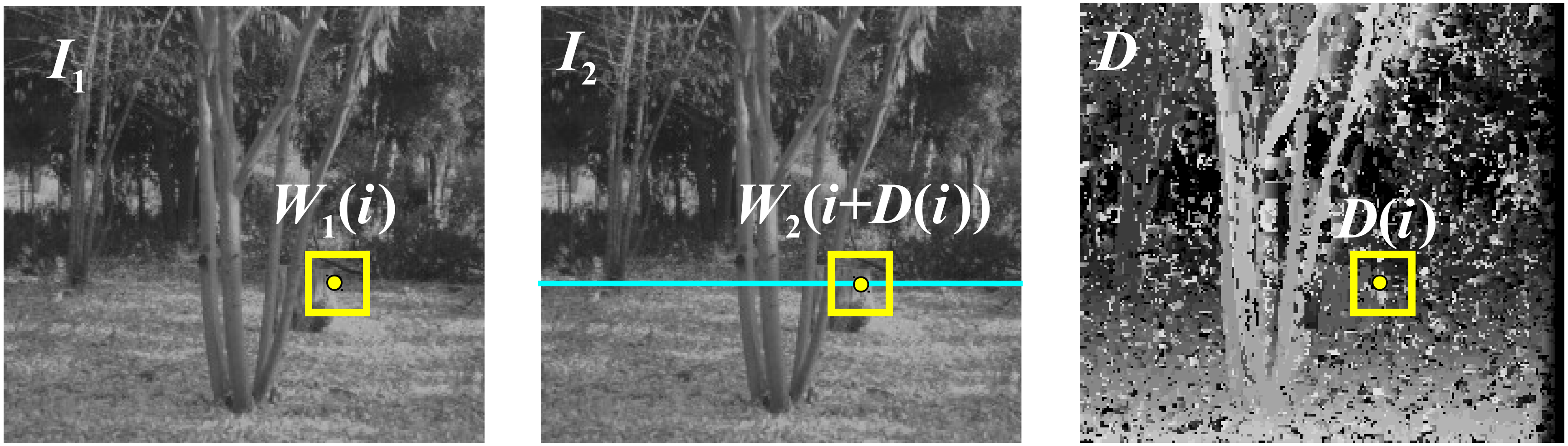
- Scanline stereo generates streaking artifacts



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

Slide Credit: James Hays

Stereo matching as energy minimization



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

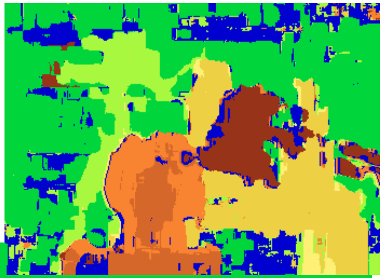
- Energy functions of this form can be minimized using *graph cuts*

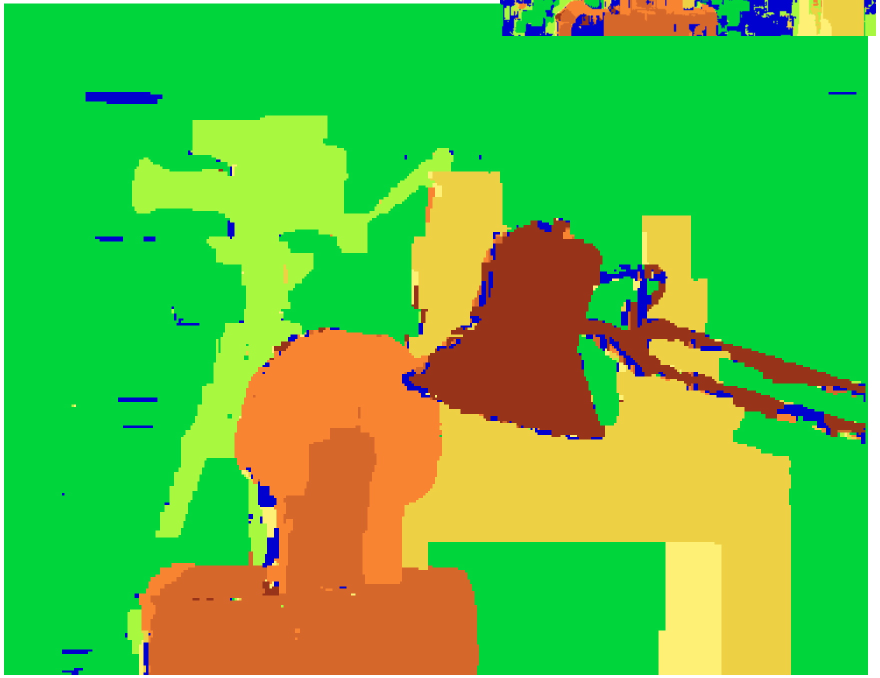
Y. Boykov, O. Veksler, and R. Zabih,
Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

 Slide Credit: James Hays

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/

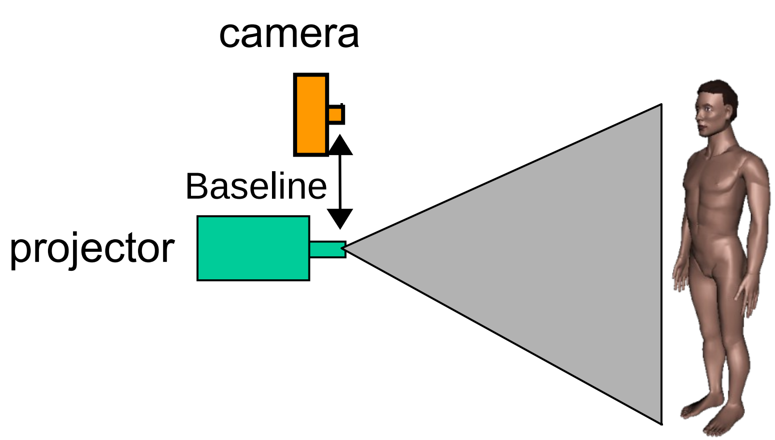
Slide Credit: James Hays

Stereo

- Advantages:
  - Passive
  - Cheap hardware (2 cameras)
  - Easy to accommodate motion
  - Intuitive analogue to human vision
- Disadvantages:
  - Only acquire good data at “features”
  - Sparse, relatively noisy data (correspondence is hard)
  - Bad around silhouettes
  - Confused by non-diffuse surfaces
- Variant: multibaseline stereo to reduce ambiguity

Active stereo with structured light

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



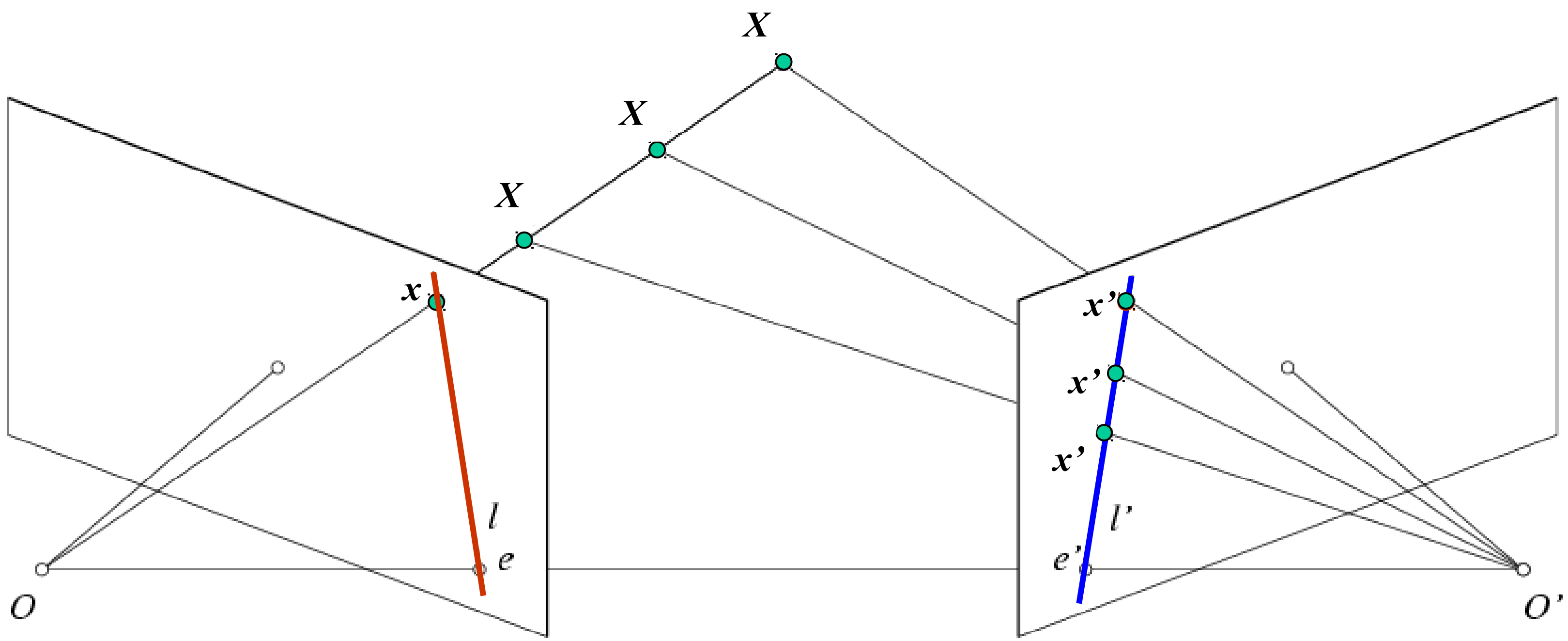
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. 3DPVT 2002

Kinect: Structured infrared light

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

Slide Credit: James Hays

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*.

Slide Credit: James Hays