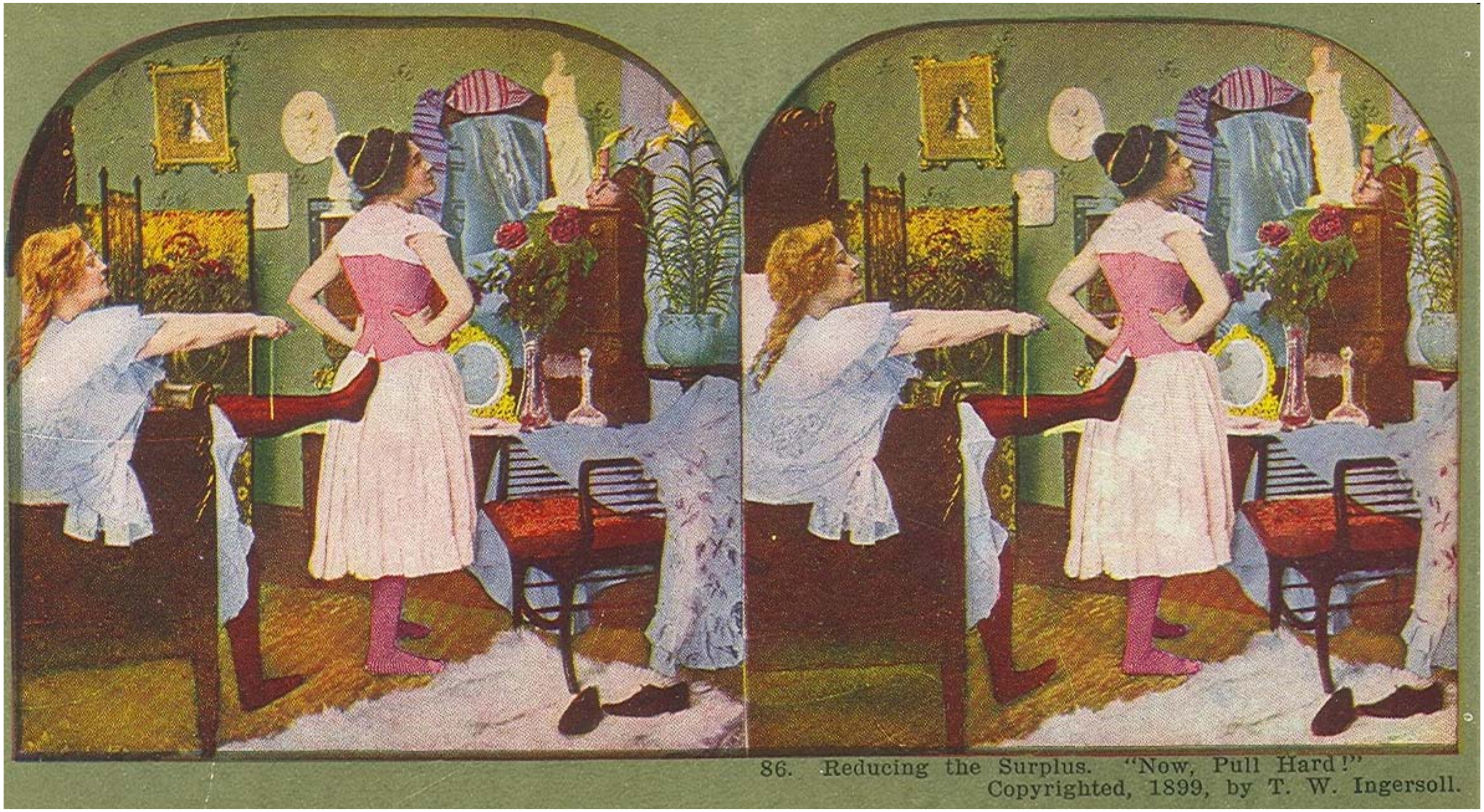
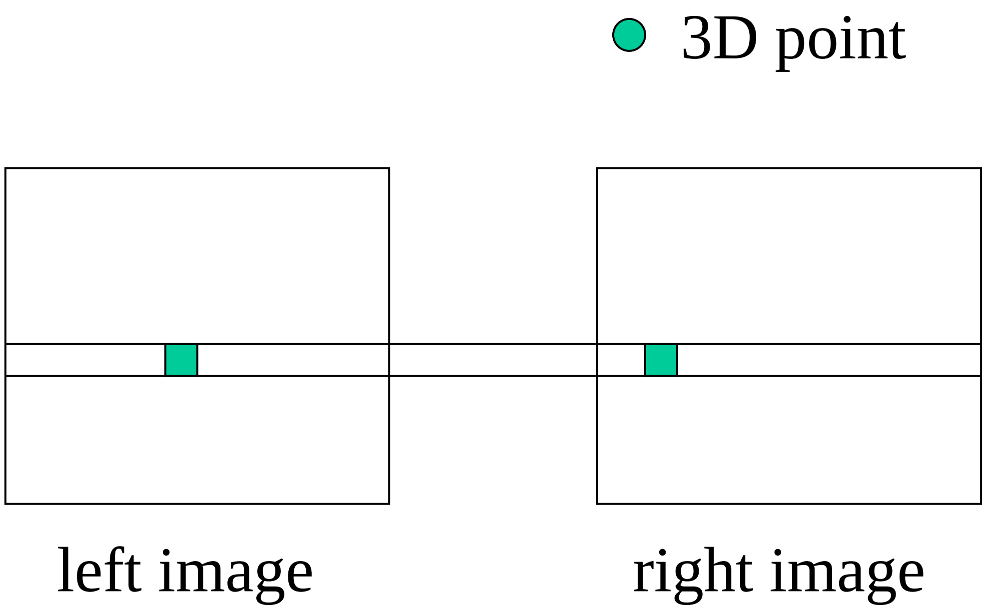
Lecture 08

Stereo and 3D Vision



How do we get 3D from Stereo Images?

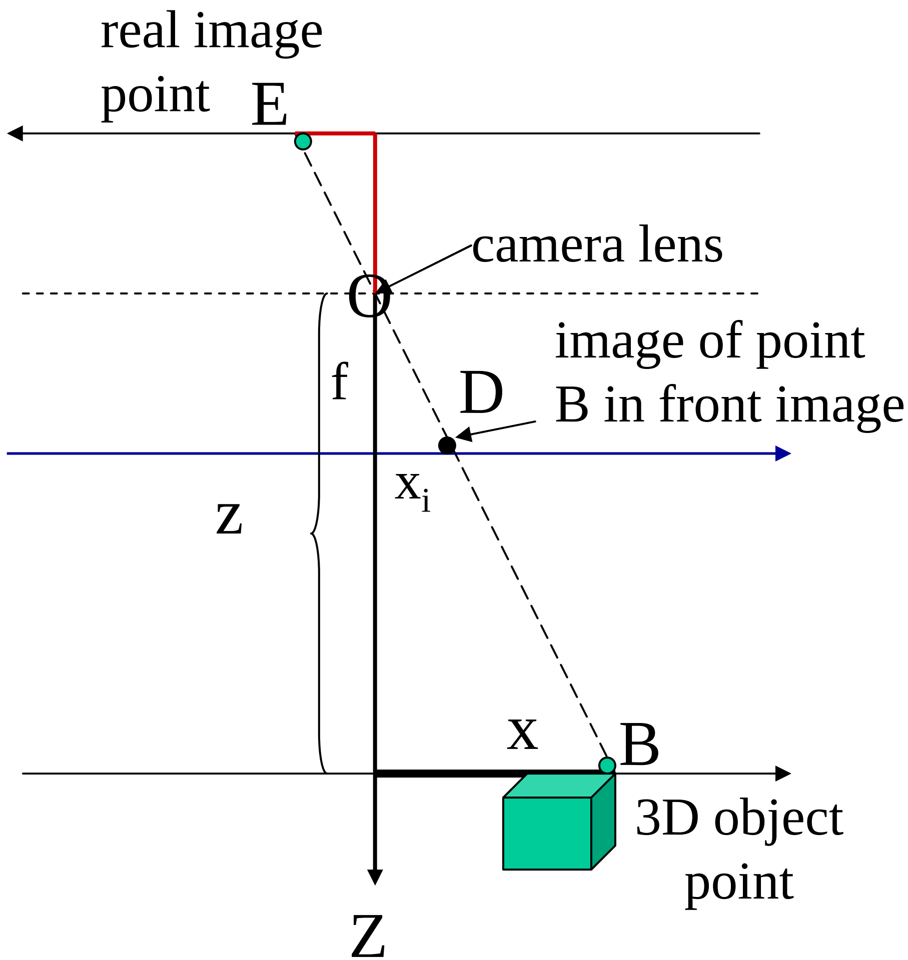
Perception of depth arises from “disparity” of a given 3D point in your right and left retinal images



disparity: the difference in image location of the *same 3D point* when projected under perspective to two different cameras

d = xleft - xright

Recall (from Lecture 2): Perspective Projection



This is the axis of the real image plane.

O is the center of projection.

This is the axis of the front image plane, which we use.

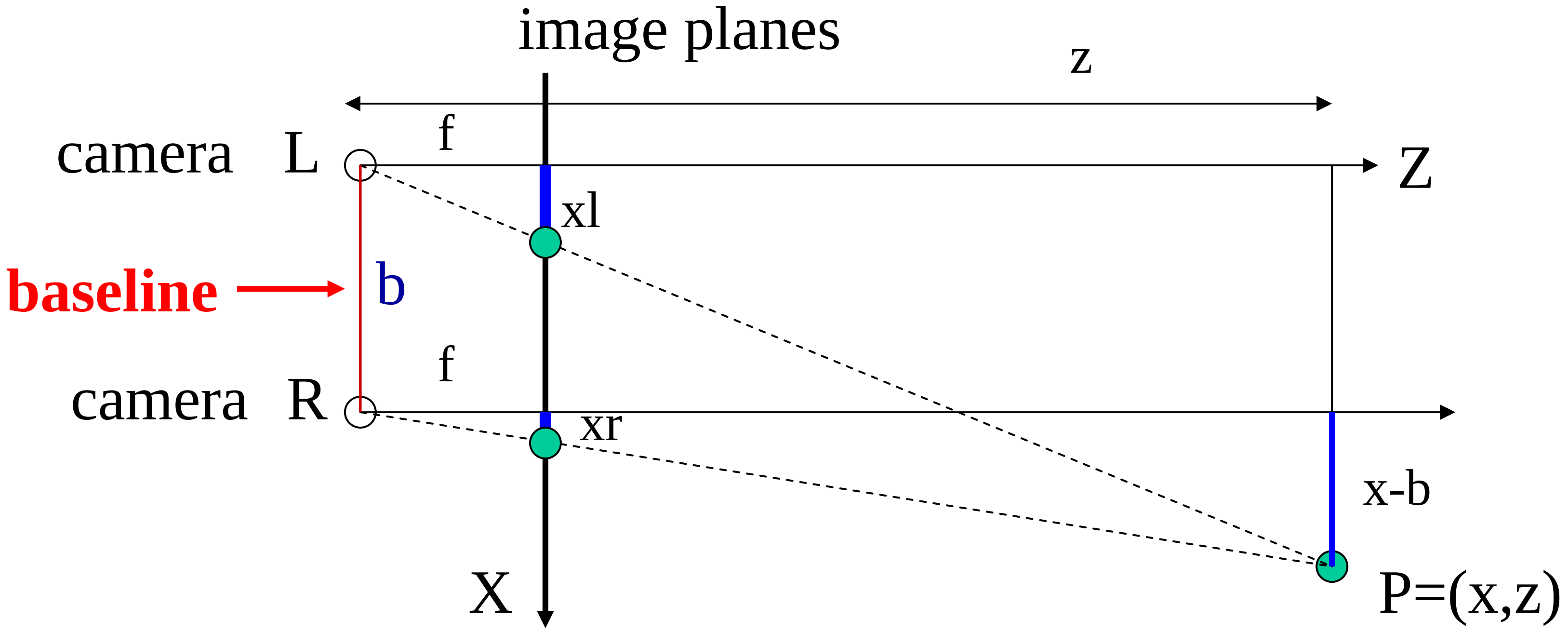
$$\frac{x_i}{f} = \frac{x}{z}$$

(from similar triangles)

(Note: For convenience, we orient Z axis as above and use f instead of –f as in lecture 2)

Projection for Stereo Images

Simple Model: Optic axes of 2 cameras are parallel



$$\frac{z}{f} = \frac{x}{xl}$$

$$\frac{z}{f} = \frac{x-b}{xr}$$

$$\frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

Y-axis is perpendicular to the page.

(from similar triangles)

3D from Stereo Images: Triangulation

For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr), the location of the 3D point can be derived from previous slide's equations:

Depth z = f*b / (xl - xr) = f*b/d

x = xl*z/f or b + xr*z/f

y = yl*z/f or yr*z/f

This method of determining depth from disparity d is called **triangulation.**

Note that **depth is inversely proportional to disparity**

Depth $z = f*b / (xl - xr) = f*b/d$

$x = xl*z/f$ or $b + xr*z/f$

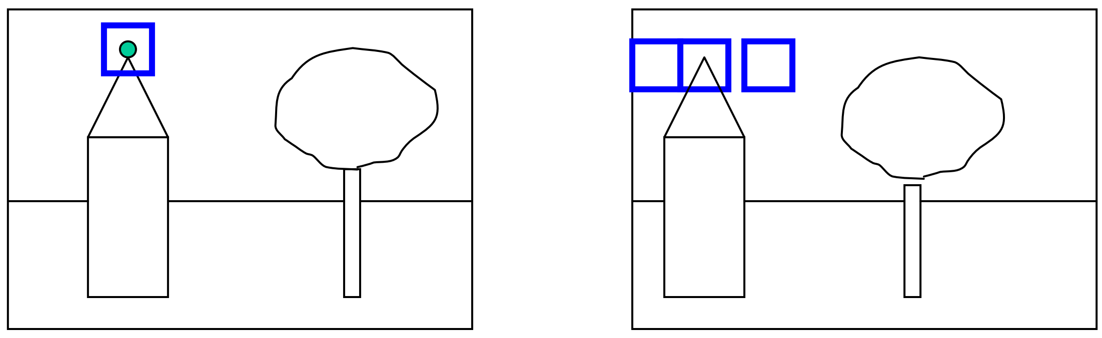
$y = yl*z/f$ or $yr*z/f$

Two main problems:

1. Need to know focal length f, baseline b
   - use prior knowledge or camera calibration
2. Need to find corresponding point (xr,yr) for each (xl,yl) $\Rightarrow$ Correspondence problem

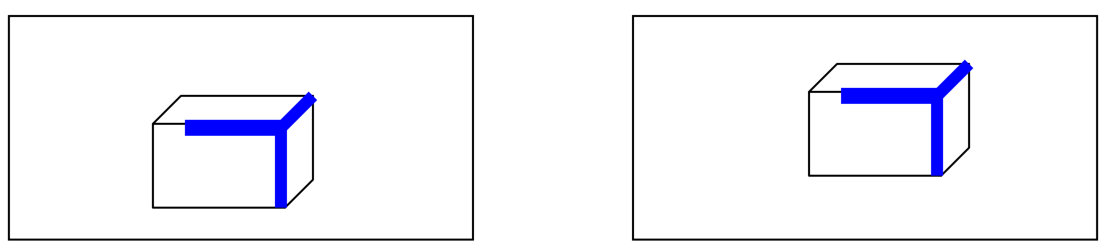
Solving the stereo correspondence problem

1. Cross correlation or SSD using small windows.



dense

2. Symbolic feature matching, usually using segments/corners.



sparse

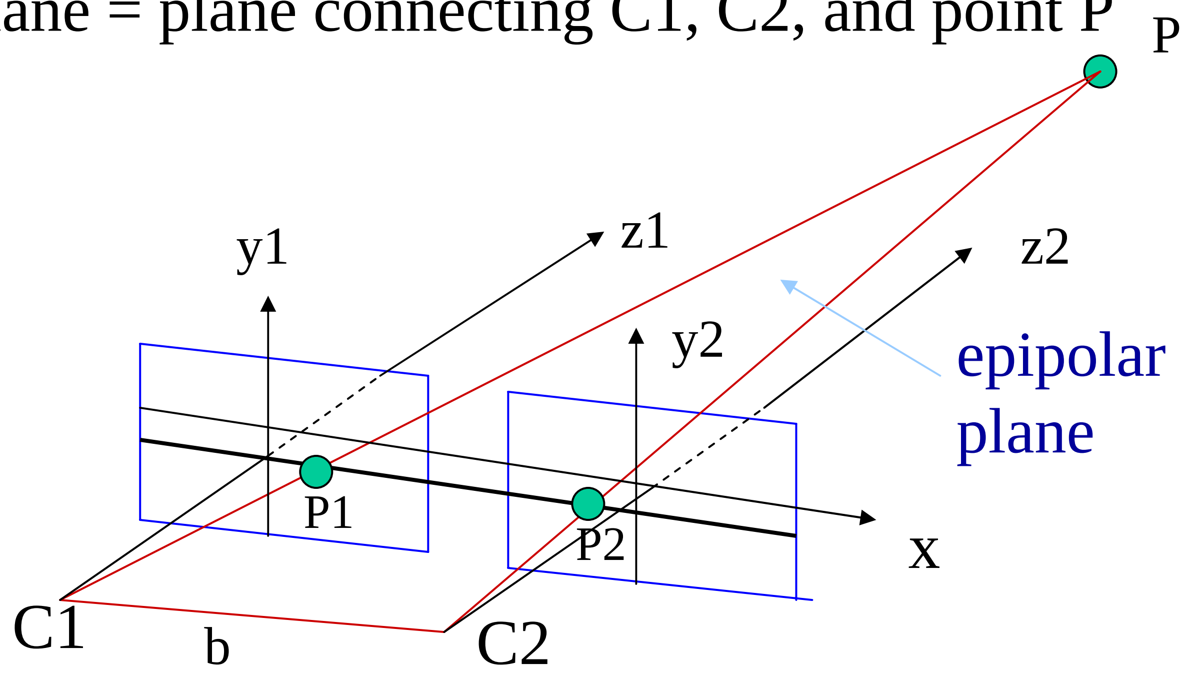
3. Use the newer interest operators, e.g., SIFT.

sparse

Given a point in the left image, do you need to search the entire right image for the corresponding point?

Epipolar Constraint for Correspondence

Epipolar plane = plane connecting C1, C2, and point P



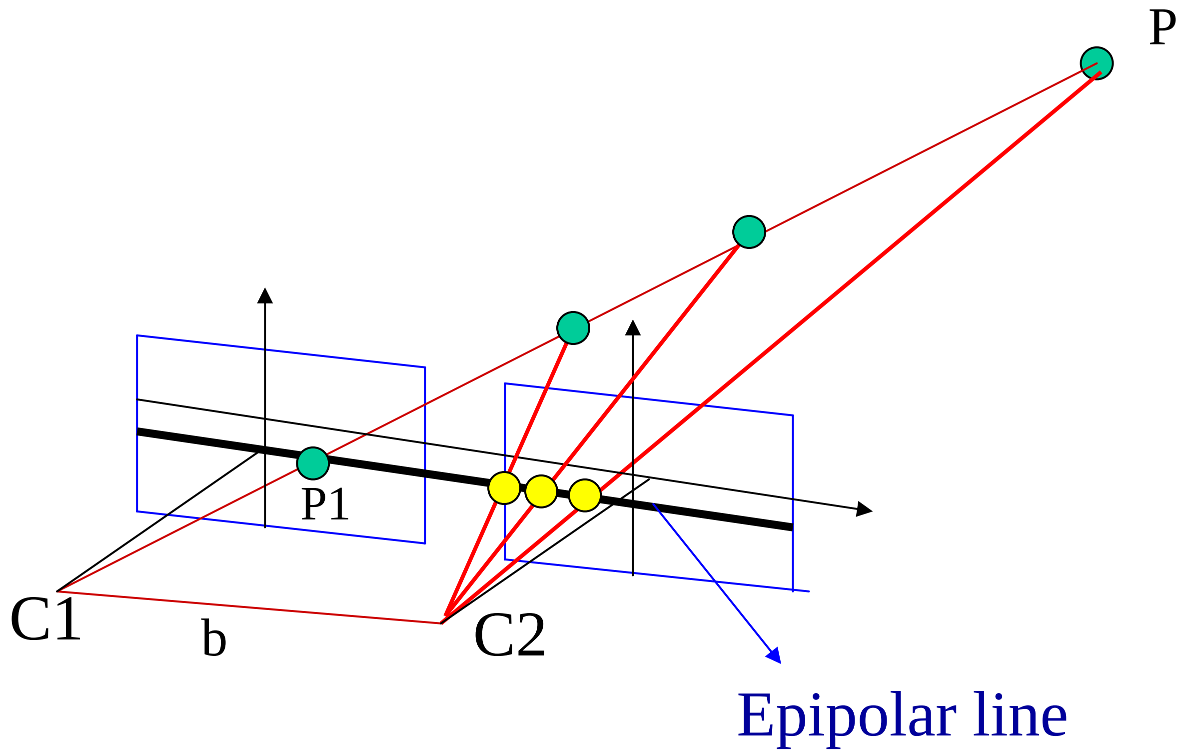
.

Epipolar plane cuts through image planes forming an epipolar line in each plane

Match for P1 (or P2) in the other image must lie on epipolar line

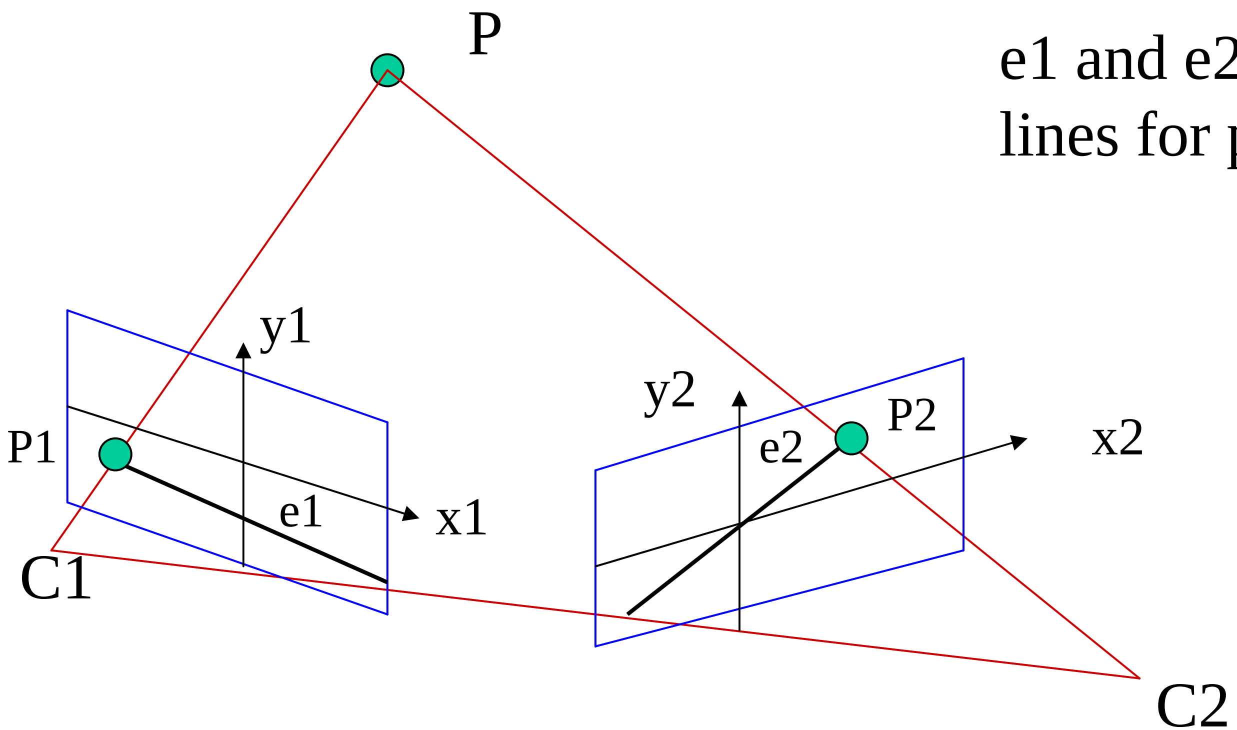
Epipolar Constraint for Correspondence

Match for P1 in the other image must lie on epipolar line
So need search only along this line



What if the optical axes of the 2 cameras are not parallel to each other?

Epipolar constraint still holds…

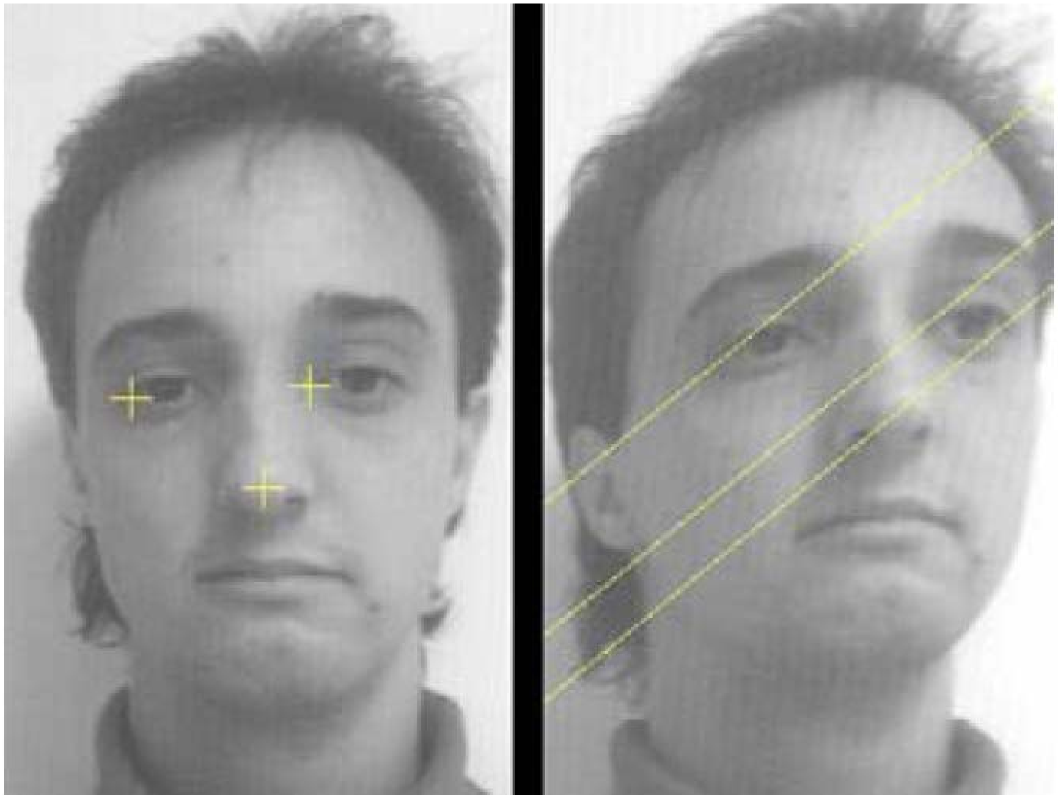


e1 and e2 are the epipolar lines for point P

But the epipolar lines may no longer be horizontal

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

Example

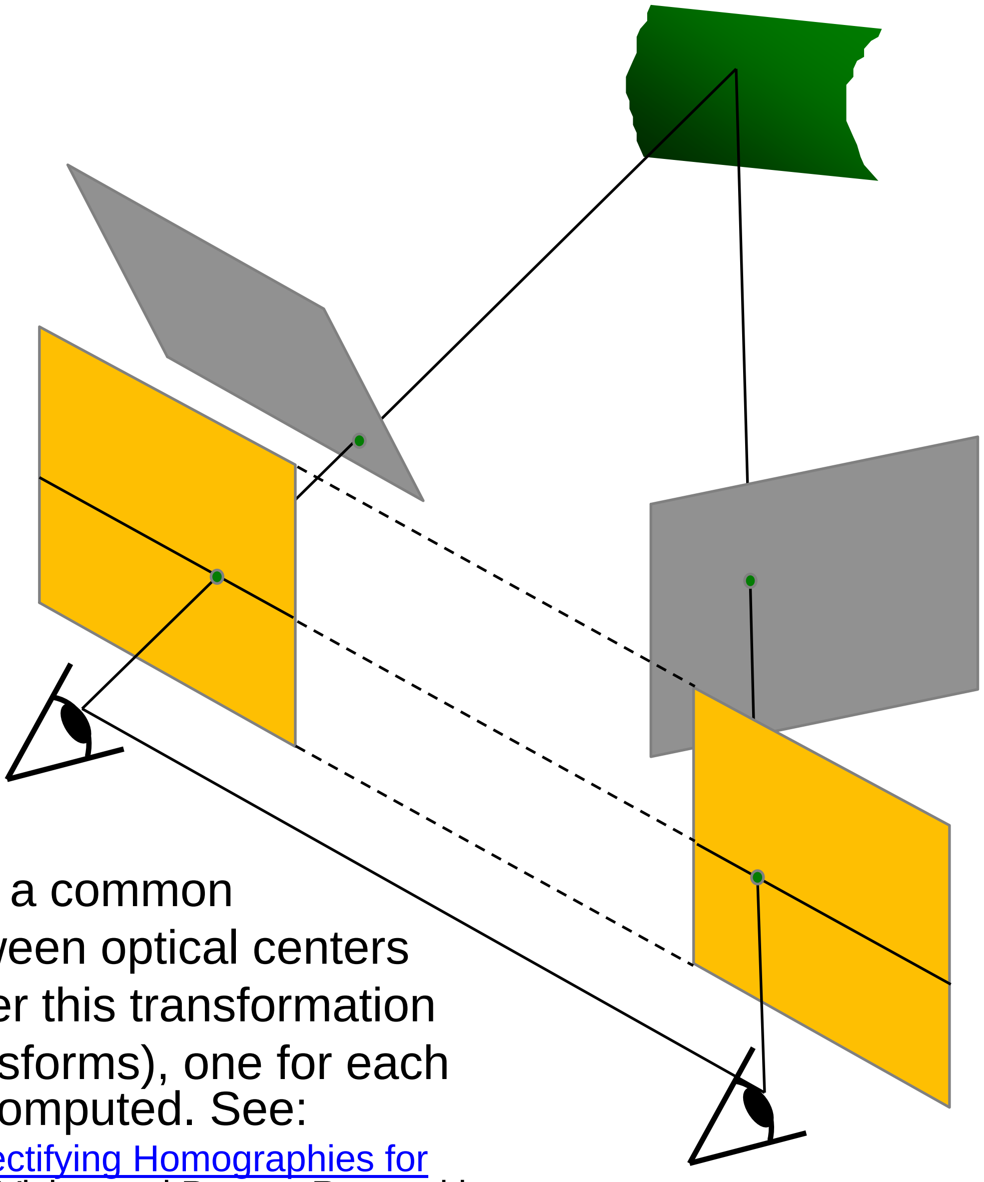


(from a slide by Pascal Fua)

Yellow epipolar lines for the three points shown on the left image

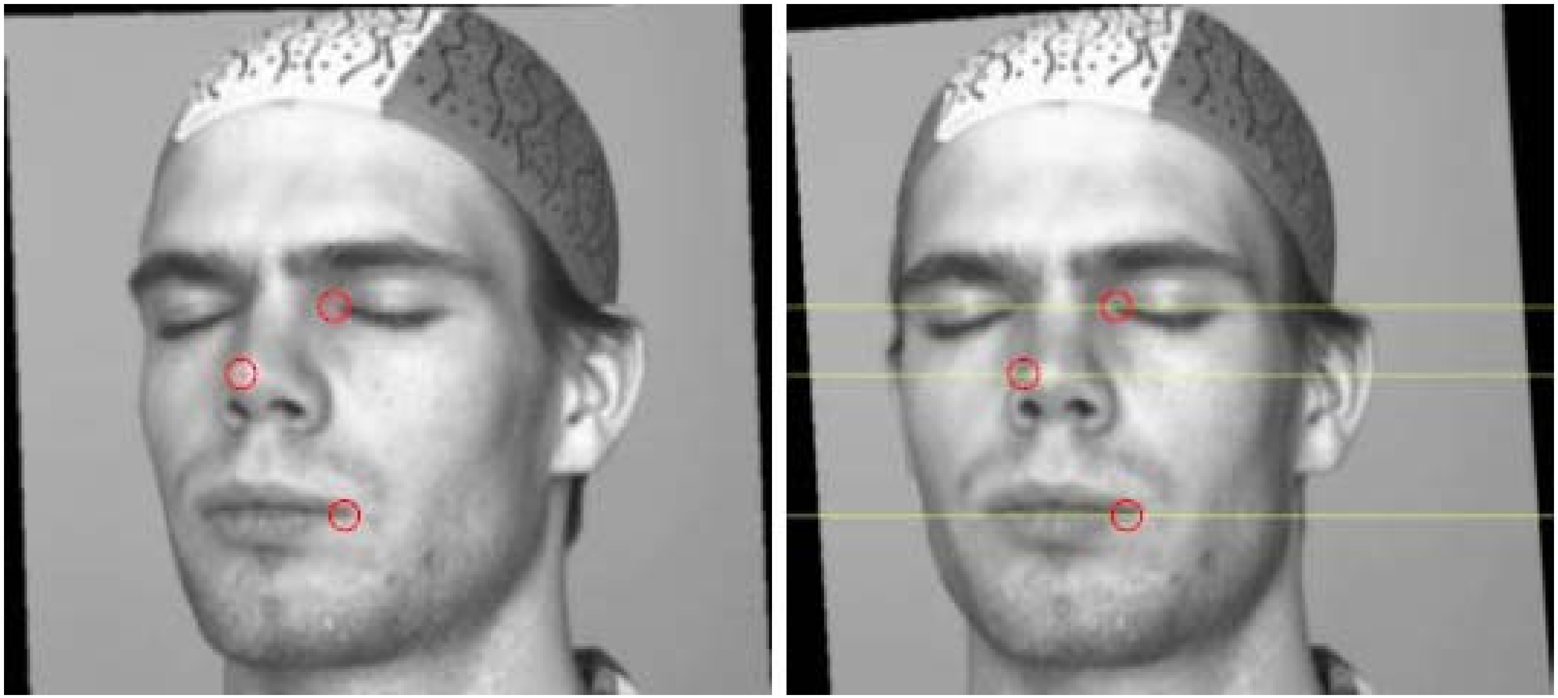
Given a point P1 in left image on epipolar line e1, can find epipolar line e2 provided we know relative orientations of cameras ⇒ Requires camera calibration (see lecture 2)

Alternate approach: Stereo image rectification



- Reproject image planes onto a common plane parallel to the line between optical centers
- Epipolar line is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection, is computed. See:
  - C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.

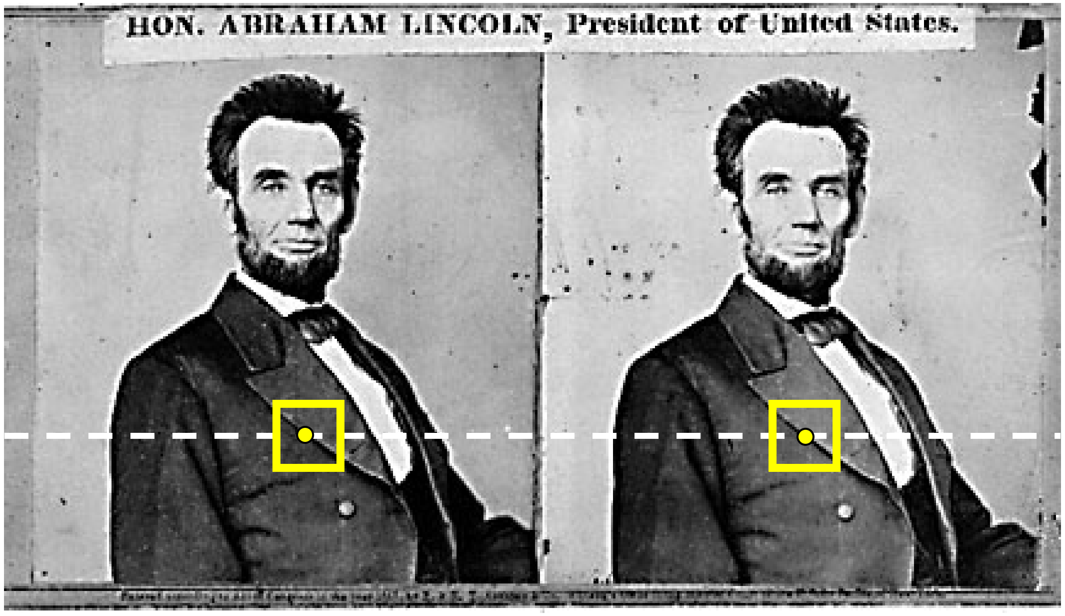
Example



After rectification, need only search for matches along horizontal scan line

(adapted from slide by Pascal Fua)

Your basic stereo algorithm



For each epipolar line

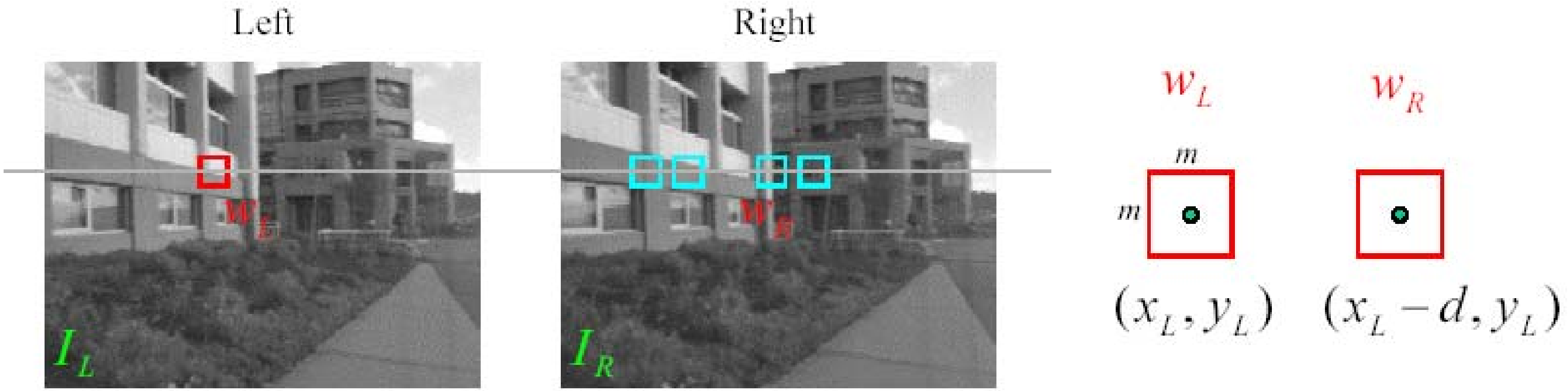
For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

A good survey and evaluation: http://vision.middlebury.edu/stereo/

Matching using Sum of Squared Differences (SSD)



w_L and w_R are corresponding m by m windows of pixels.

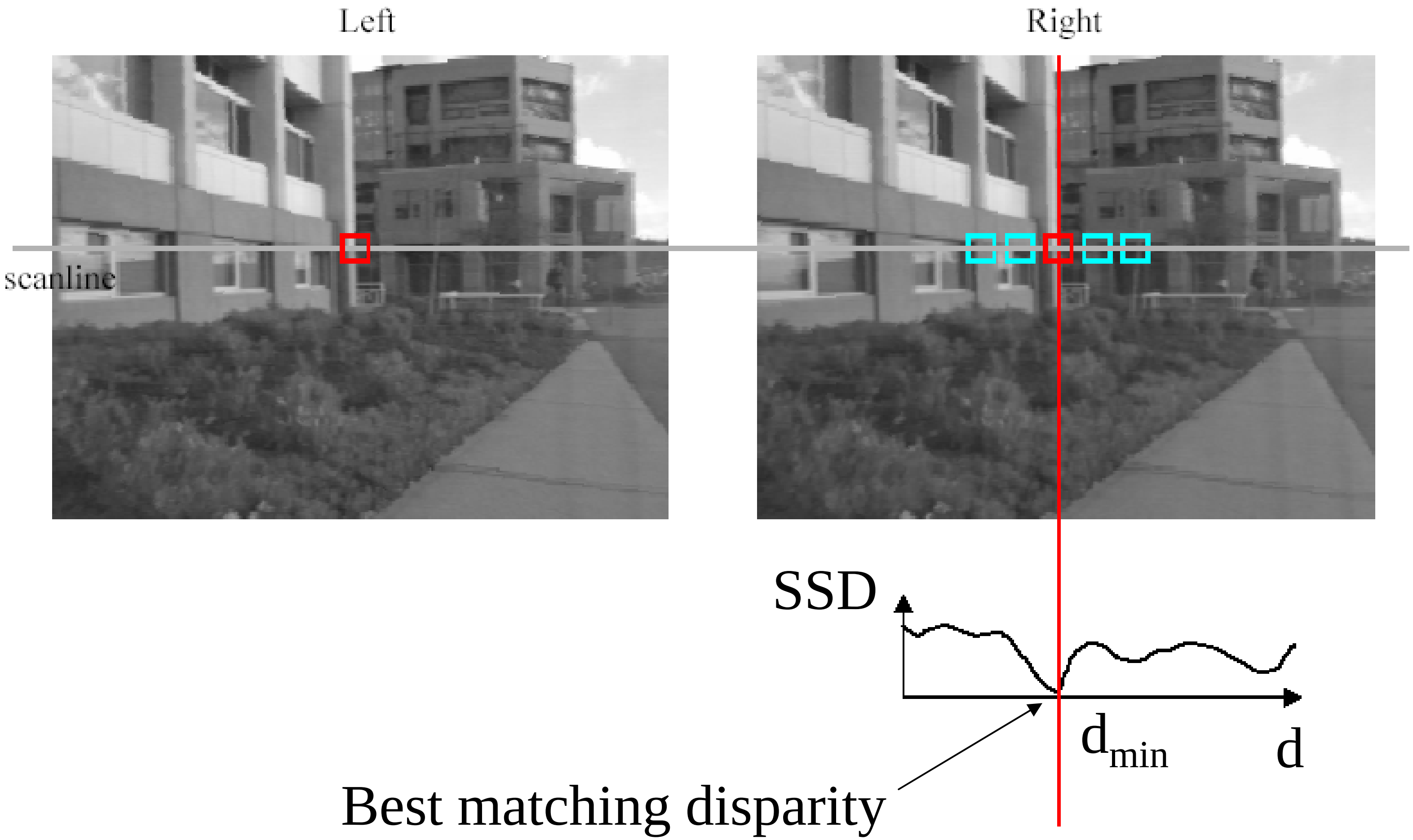
We define the window function :

$$W_m(x, y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

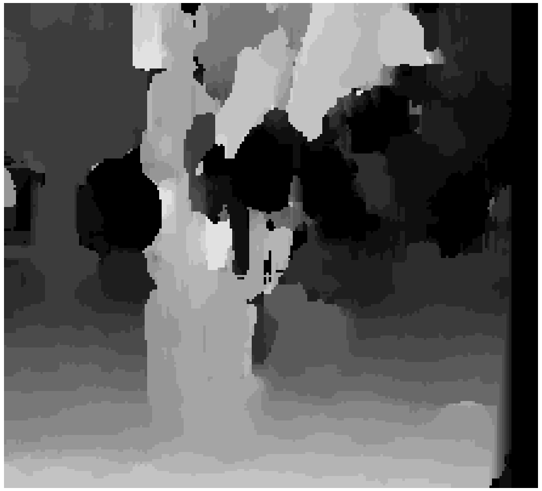
Stereo matching based on SSD



Problems with window size

Input stereo pair

W = 3



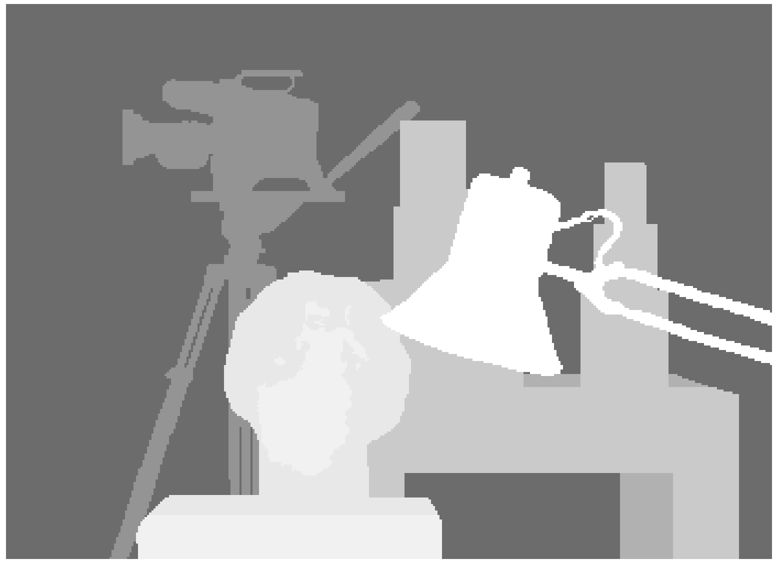
W = 20

Effect of window size W

- Smaller window
  - \+ Good precision, more detail
  - – Sensitive to noise
- Larger window
  - \+ Robust to noise
  - – Reduced precision, less detail

Example depth from stereo results

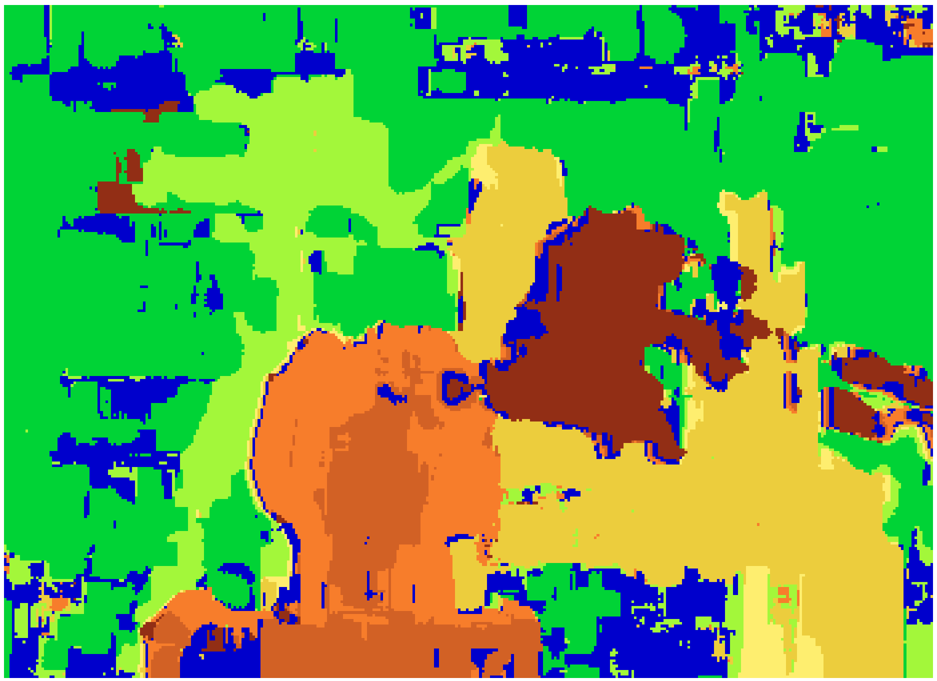
- Data from University of Tsukuba



Scene

Ground truth

Results with window-based stereo matching



Window-based matching
(best window size)

Ground truth

Better methods exist...

State of the art method:

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

Ground truth

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

Stereo reconstruction pipeline

Steps

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

What will cause errors?

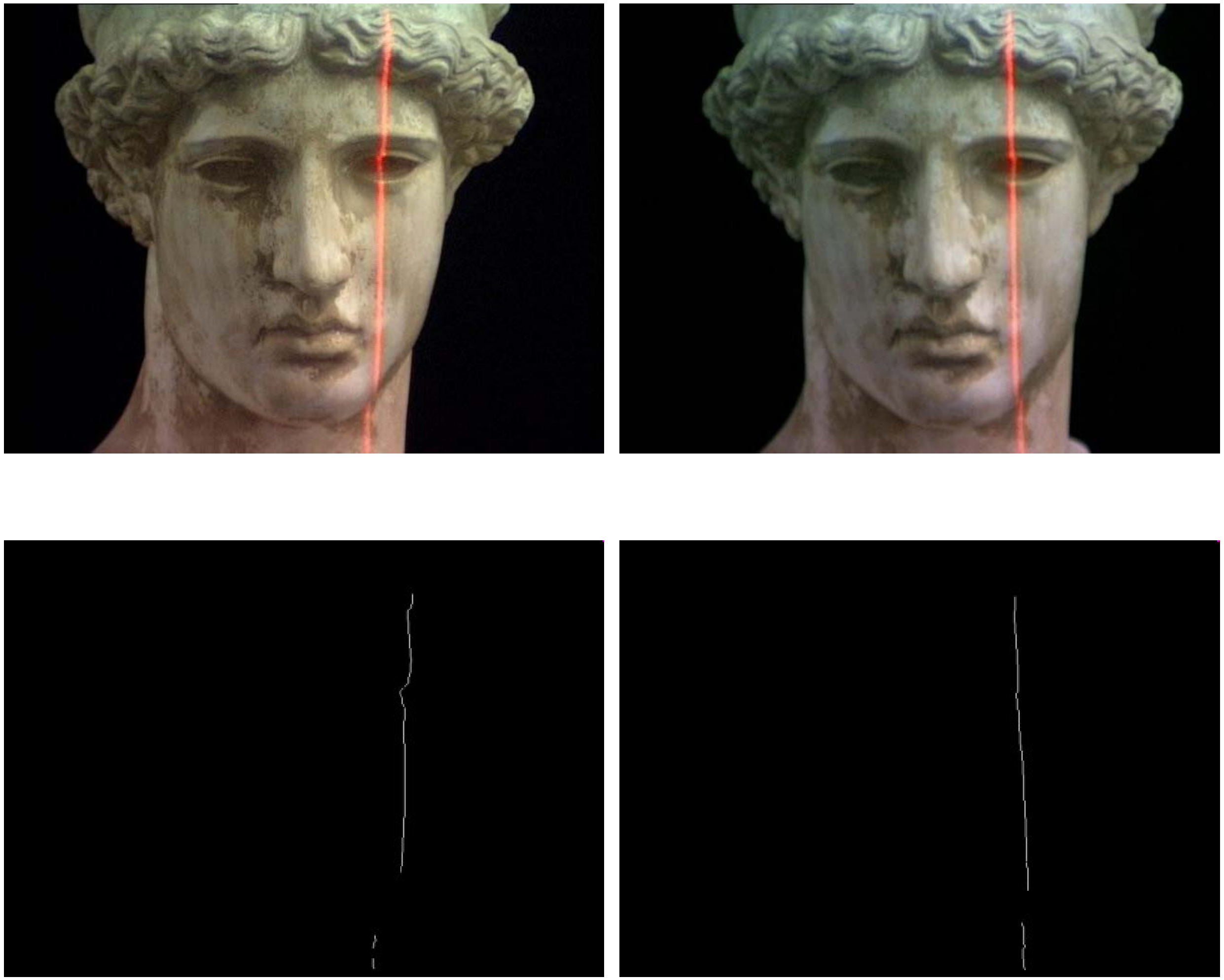
- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- Low-contrast image regions

What if 3D object has little or no texture?

Matching points might be difficult or impossible

Can we still recover depth information?

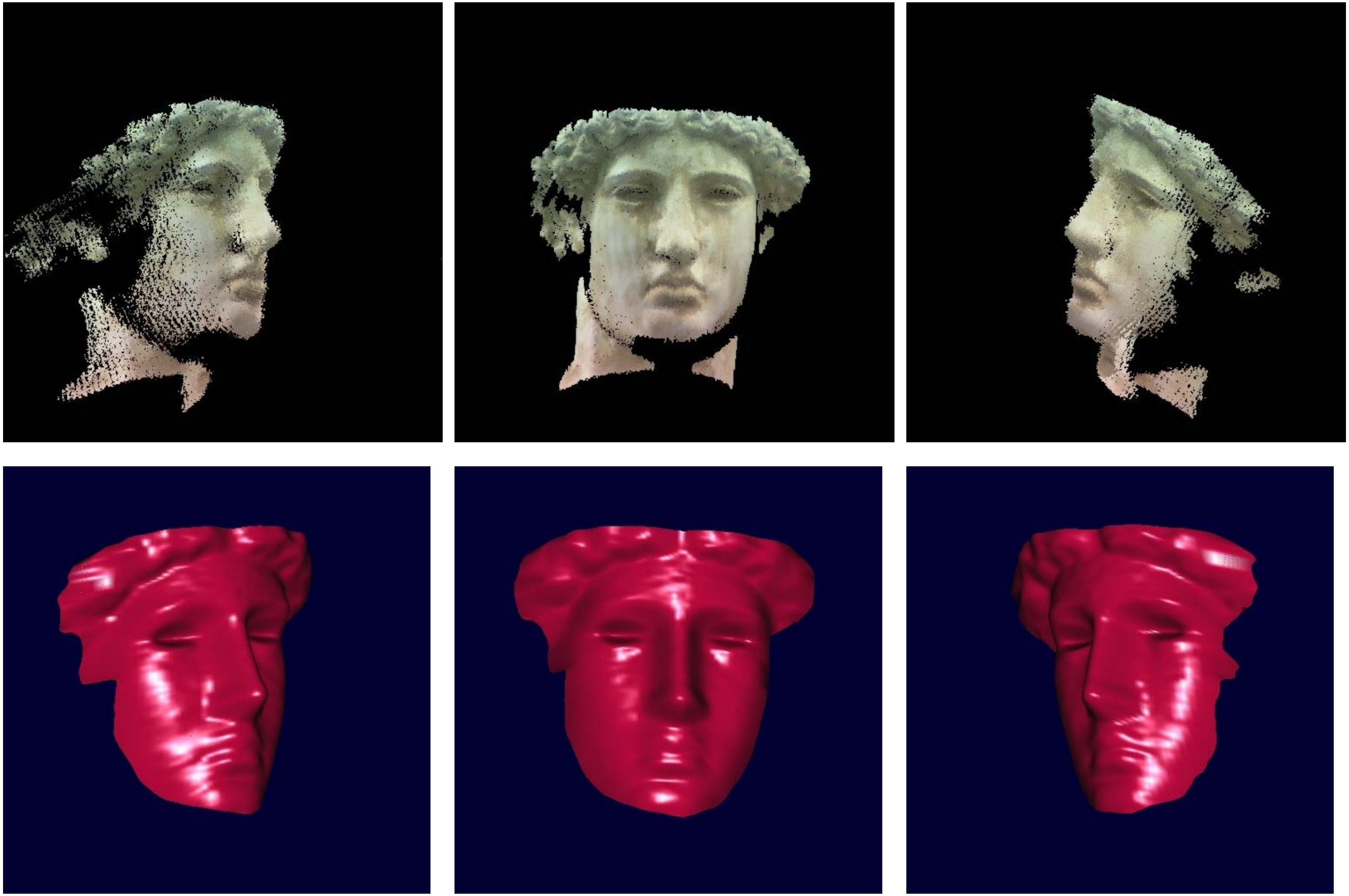
Idea: Use structured light!



Disparity between laser points on the same scanline in the images determines the 3-D coordinates of the laser point on object

http://www.cs.wright.edu/~agoshtas/stereorange.html

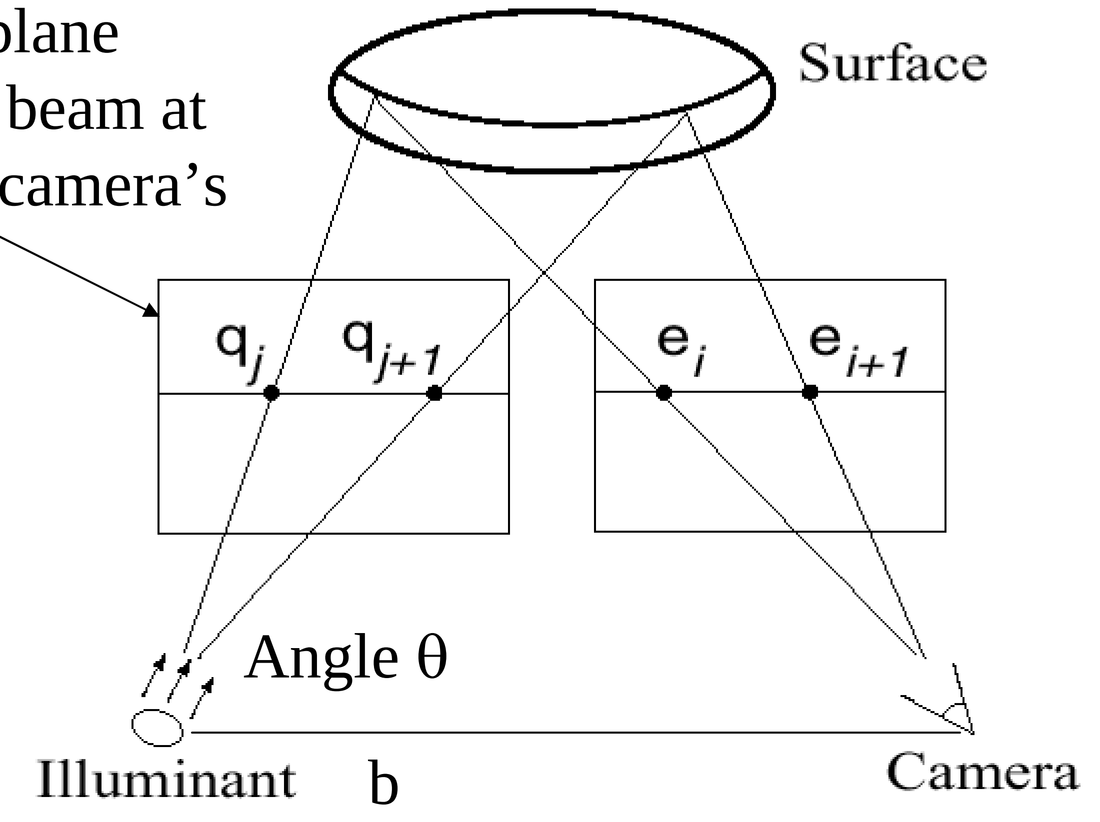
Recovered 3D Model



http://www.cs.wright.edu/~agoshtas/stereorange.html

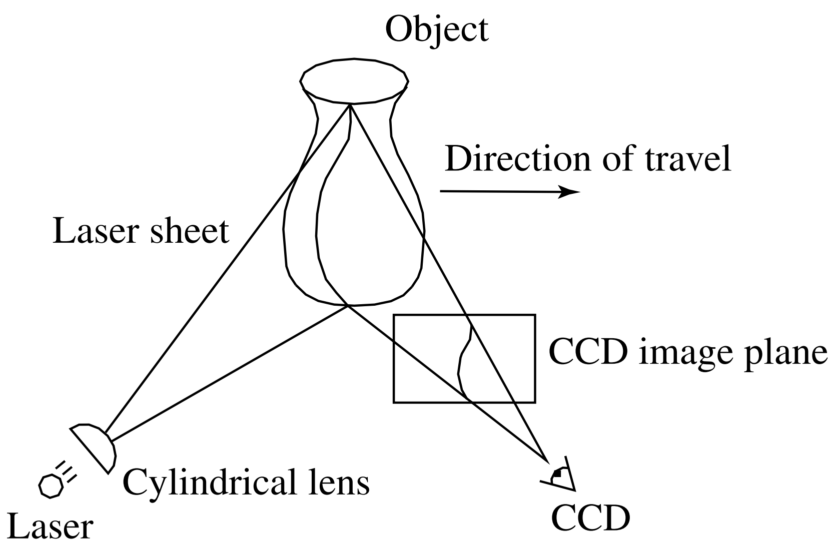
Actually, we can make do with just 1 camera

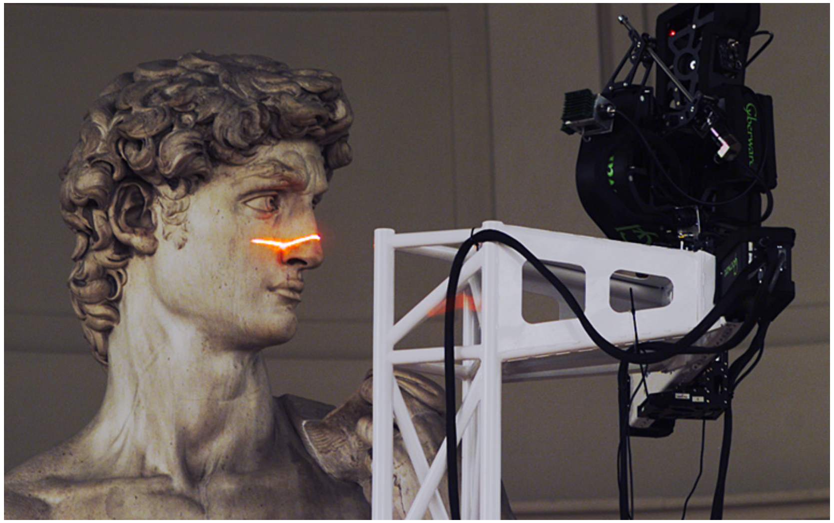
Virtual "image" plane intersecting laser beam at same distance as camera's image place



From calibration of both camera and light projector, we can compute 3D coordinates laser points on the surface

The Digital Michelangelo Project





http://graphics.stanford.edu/projects/mich/

Optical triangulation

- Project a single stripe of laser light
- Scan it across the surface of the object

Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Laser scanned models

The Digital Michelangelo Project, Levoy et al.