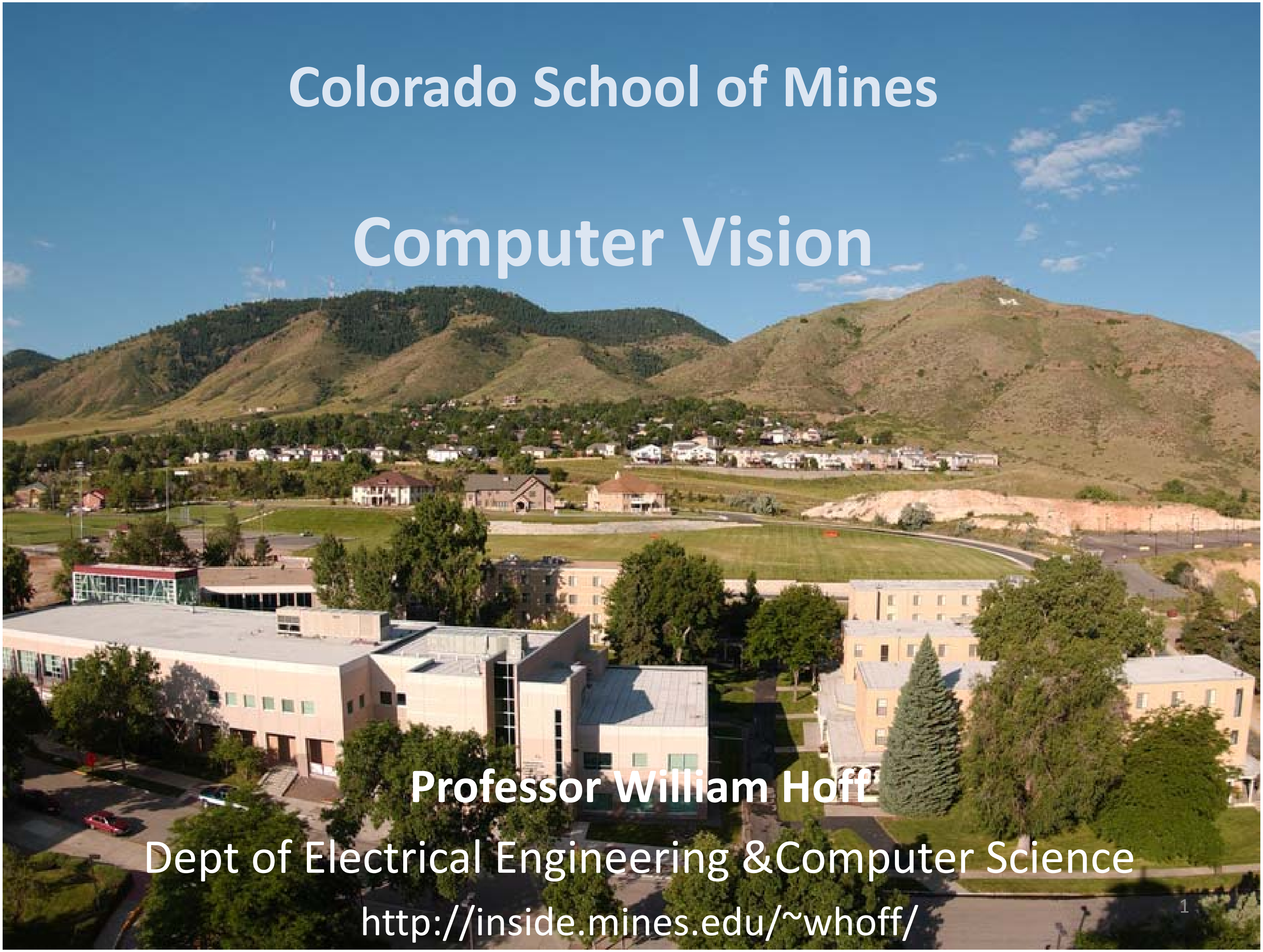
Colorado School of Mines

Computer Vision



Professor William Hoff

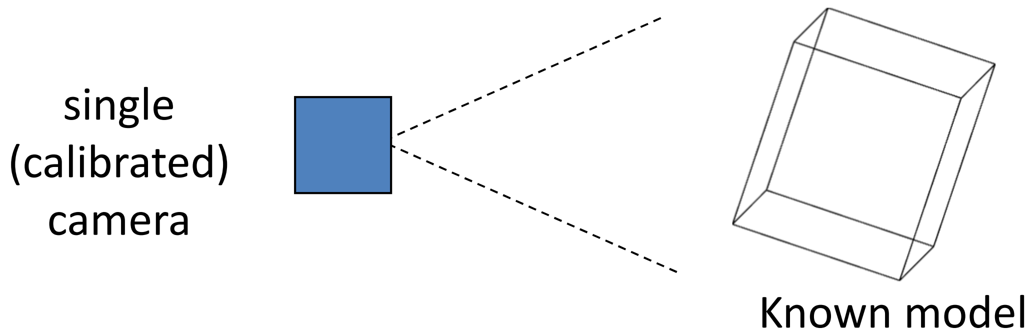
Dept of Electrical Engineering &Computer Science

http://inside.mines.edu/~whoff/

Stereo Vision

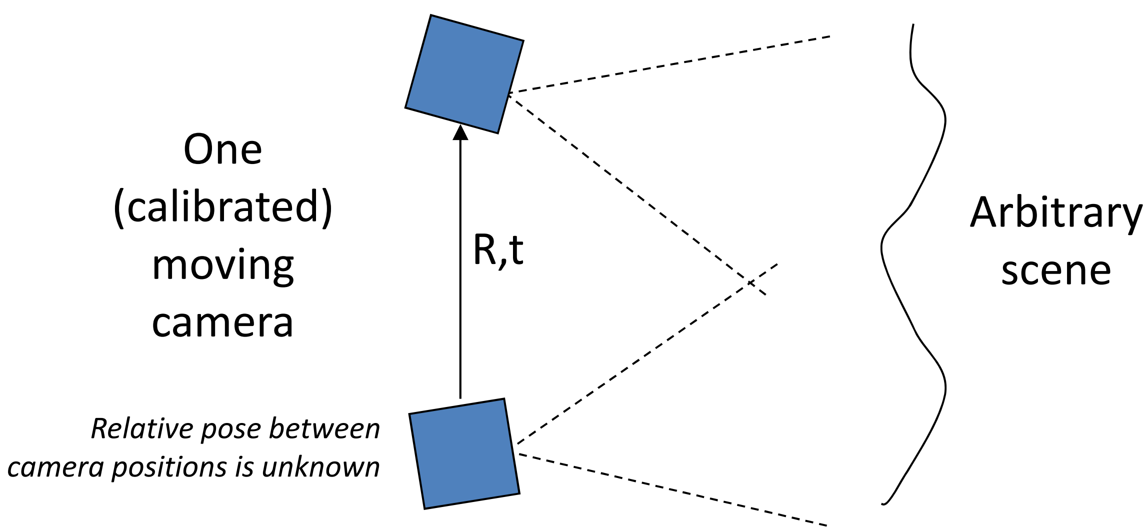
Inferring 3D from 2D

- Model based pose estimation



-> Can determine the pose of the model

- Structure-from-motion

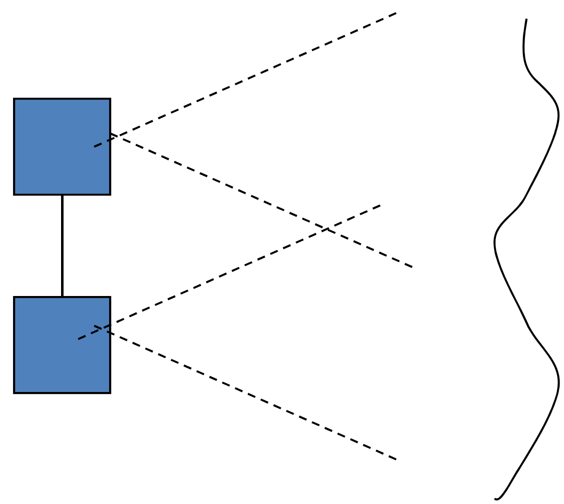


-> Can determine the positions of points, as well as the motion of the camera (R,t), up to a scale factor

Inferring 3D from 2D

- Stereo vision

two (calibrated) cameras



Arbitrary scene

-> Can determine the positions of points in the scene

Relative pose between cameras is also known

  - No camera motion needed
  - No unknown scale factor

Stereo Vision

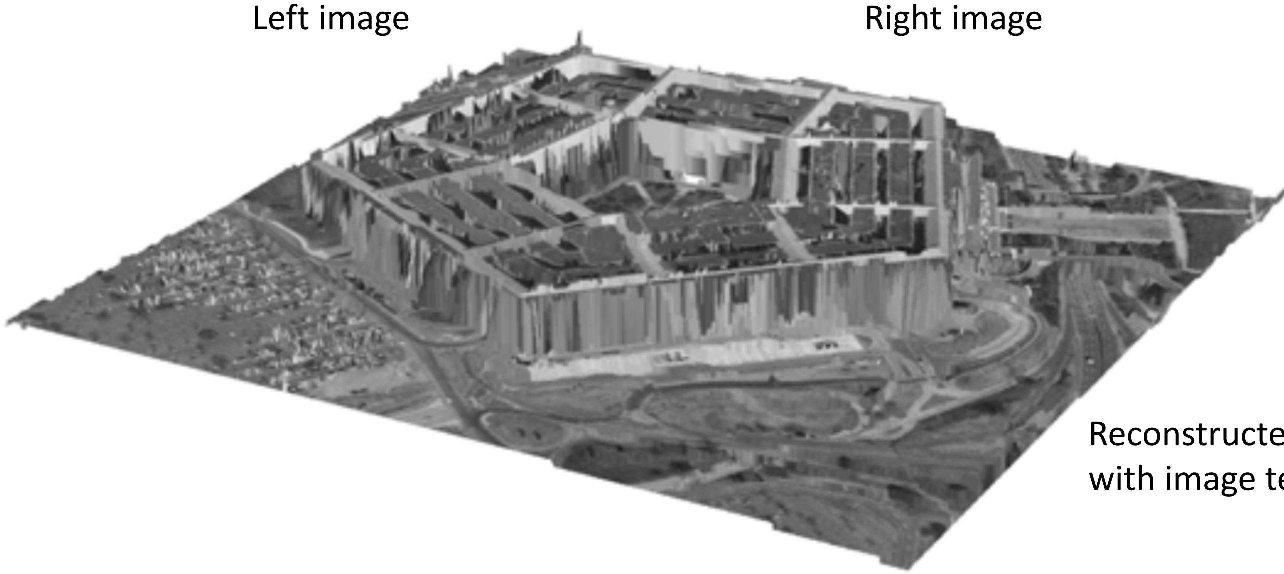
- Get depth (3-D) information from two 2-D images
  - Used by humans and animals, now computers
- Computational stereo vision
  - Studied extensively in the last 30 years
  - Still an active area of research
  - Some commercial systems available (e.g., http://www.ptgrey.com)
- In this lecture we just consider two frames
  - Other approaches are designed for more than two frames

- Good references
  - Scharstein and Szeliski, 2002. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms." *International Journal of Computer Vision,* 47(1-3), 7-42
  - http://vision.middlebury.edu/stereo - extensive website with evaluations of algorithms, test data, code

Example

Left image

Right image



Reconstructed surface
with image texture

Stereo Displays

- Stereograms were popular in the early 1900’s
- A special viewer was needed to display two different images to the left and right eyes

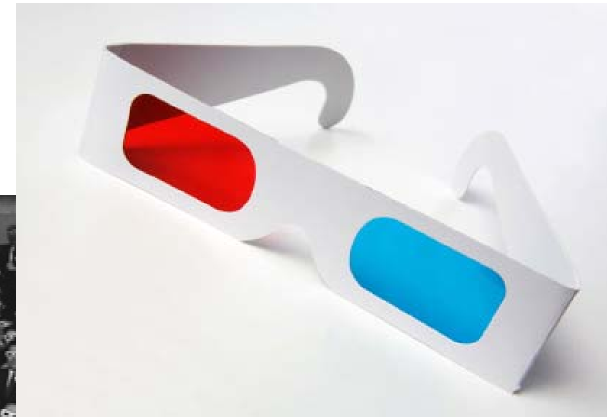
bxp28350 www.fotosearch.com

http://www.columbia.edu/itc/mealac/pritchett/00routesdata/1700_1799/jaipur/jaipurcity/jaipurcity.html

Stereo Displays

- 3D movies were popular in the 1950's
- The left and right images were displayed as red and blue

http://j-walkblog.com/index.php?/weblog/posts/swimmers/

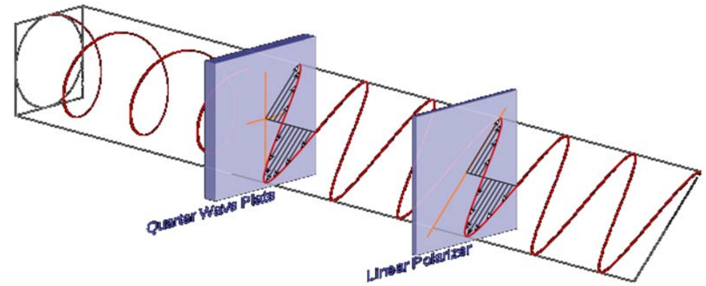


Stereo Displays

- Autostereograms (www.magiceye.com) – popular in the 80's

Stereo Displays

- Current technology for 3D movies and computer displays is to use polarized glasses
- The viewer wears eyeglasses which contain circular polarizers of opposite handedness



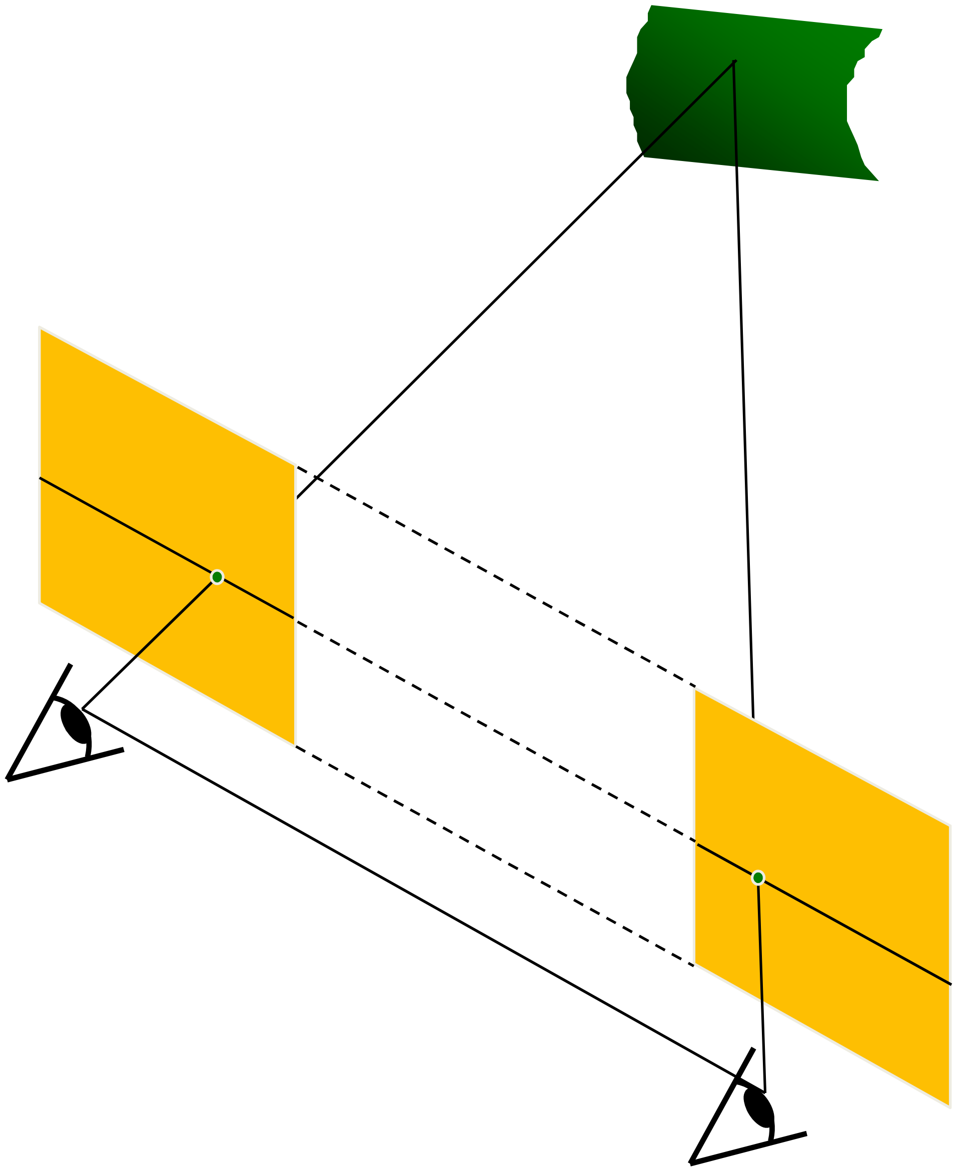
http://www.3dsgamenews.com/2011/01/3ds-to-feature-3d-movies/

Stereo Principle

- If you know
  - intrinsic parameters of each camera
  - the relative pose between the cameras
- If you measure
  - An image point in the left camera
  - The corresponding point in the right camera

- Each image point corresponds to a ray emanating from that camera
- You can intersect the rays (triangulate) to find the absolute point position

Simplest Case: Parallel images



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the images

The y-coordinates of corresponding points are the same

Stereo Geometry – Simple Case

- Assume image planes are coplanar
- There is only a translation in the *X* direction between the two coordinate frames
- *b* is the baseline distance between the cameras

$$x_L = f\frac{X_L}{Z_L},\ x_R = f\frac{X_R}{Z_R}$$

$$Z_L = Z_R = Z$$

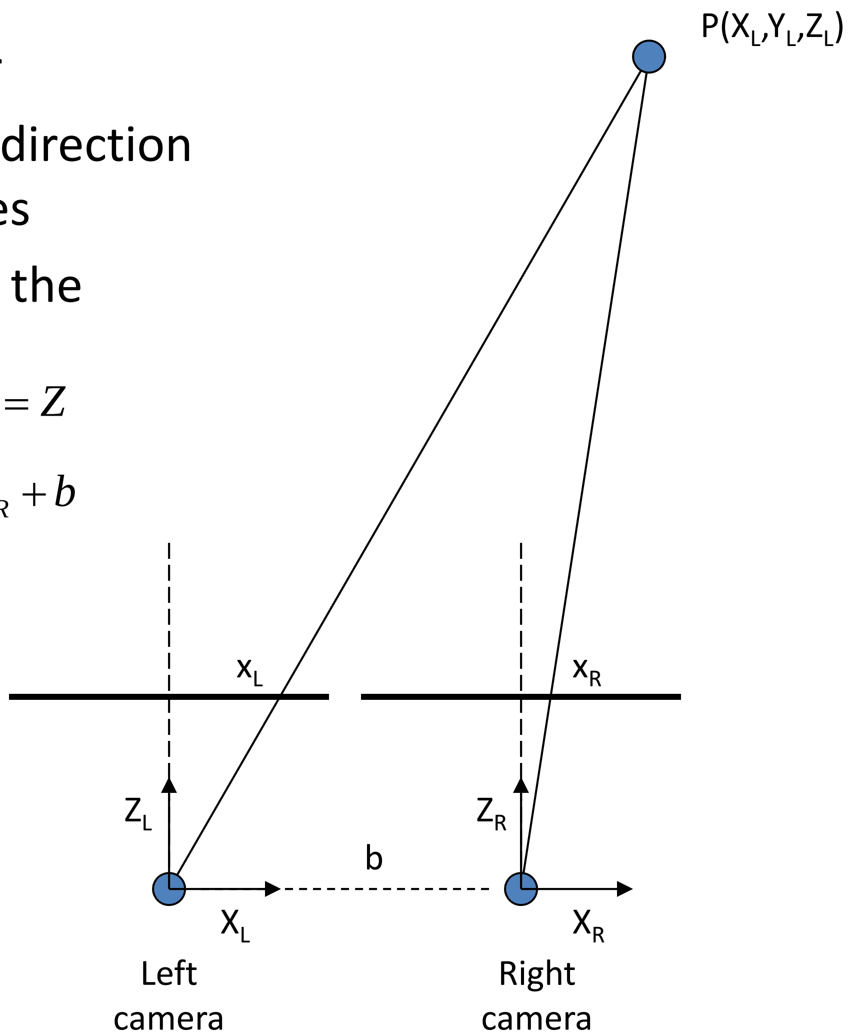
$$X_L = X_R + b$$

$$\rightarrow \quad x_L = f\frac{X_R + b}{Z}$$

$$d = x_L - x_R = f\frac{(X_R + b) - X_R}{Z} = f\frac{b}{Z}$$

$$\boxed{\rightarrow \quad Z = f\frac{b}{d}}$$

Disparity $d = x_L - x_R$

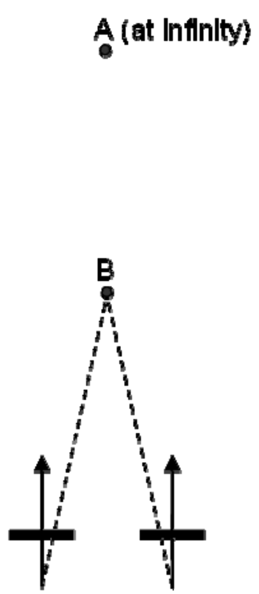


Example

- A stereo vision system estimates the disparity of a point as *d*=10 pixels
  - What is the depth (*Z*) of the point, if *f* = 500 pixels and *b* = 10 cm?

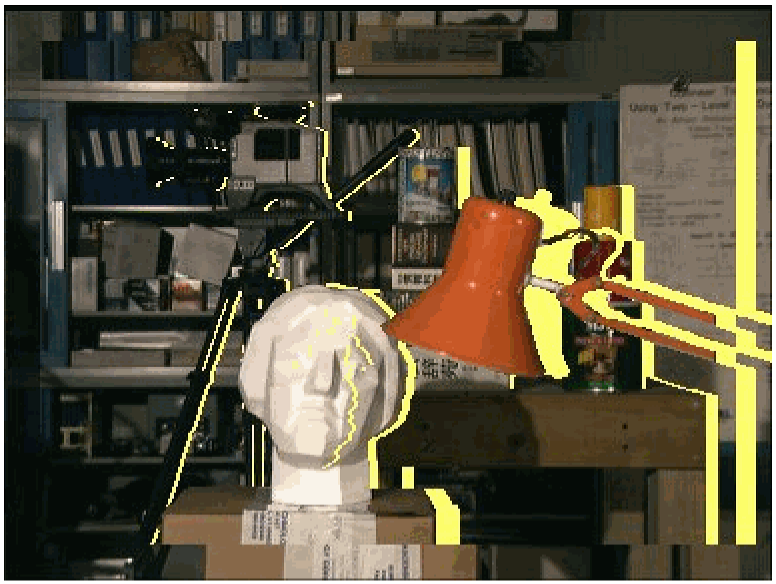
$$Z = f\frac{b}{d} = (500\,\text{pix})\frac{10\,\text{cm}}{10\,\text{pix}} = 500\,\text{cm}$$

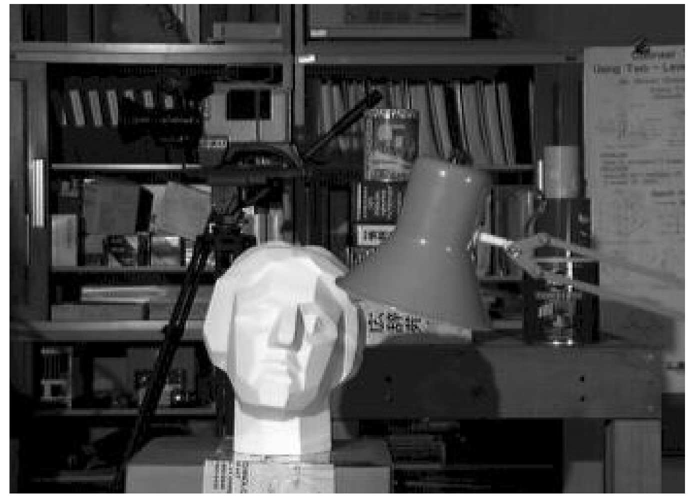
- What is the disparity of points A and B? Point A is located at infinity. Point B is located midway between the cameras at a range of 1 m.

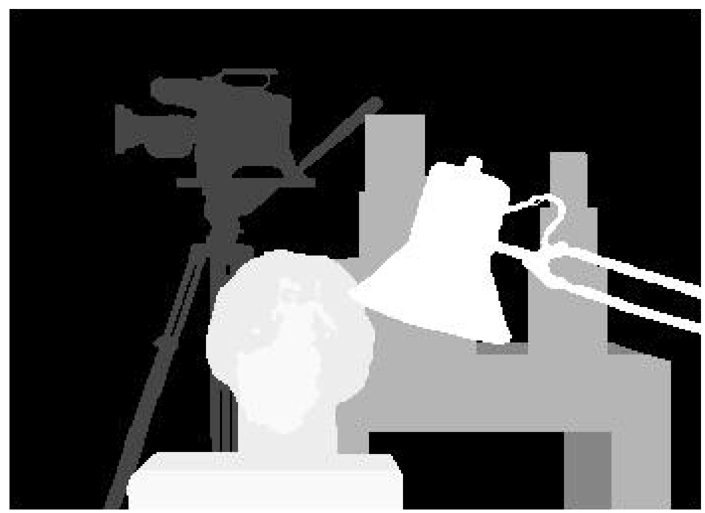


Goal: a complete disparity map

- Disparity is the difference in position of corresponding points between the left and right images



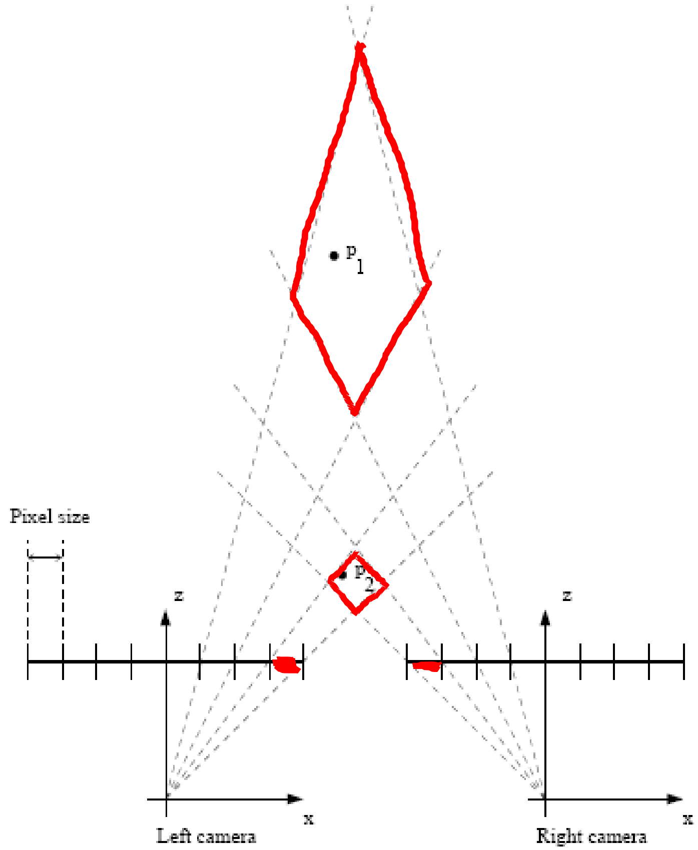




http://vision.middlebury.edu/stereo

Reconstruction Error

- Given the uncertainty in pixel projection of the point, what is the error in depth?
- Obviously the error in depth (ΔZ) will depend on:
  - Z, b, f
  - $\Delta x_L, \Delta x_R$
- Let's estimate the variance of the error



From http://www.danet.dk/sensor_fusion

Reconstruction Error

- Disparity is $d = x_L - x_R$
- The error in disparity Δd, from the error of locating the feature in each image, Δx_L and Δx_R is $\sigma_d^2 = \sigma_L^2 + \sigma_R^2$ (see lecture on "Uncertainty")

- The depth is $Z=fb/d$
  - We take the total derivative of each side
  - If the only uncertainty is in the disparity *d*, then

$$\Delta Z = f \frac{b}{d^2}(-\Delta d)$$

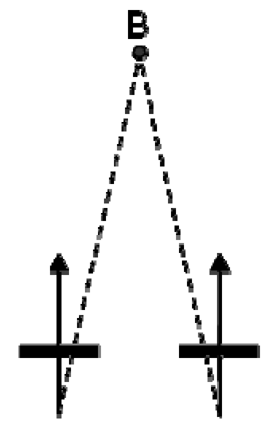
- The variance of the error is $\sigma_Z^2 = E[(\Delta Z - \mu_Z)^2]$

$$E[(\Delta Z - \mu_Z)^2] = E[\Delta Z^2] = \left(f \frac{b}{d^2}\right)^2 E[(-\Delta d)^2]$$

$$\sigma_Z^2 = \left(f \frac{b}{d^2}\right)^2 \sigma_d^2 \quad \longrightarrow \quad \sigma_Z = f \frac{b}{d^2} \sigma_d = Z \frac{\sigma_d}{d}$$

Example

- What is the uncertainty (standard deviation) of the depth of point B, if the standard deviation of locating a feature in each image = 1 pixel?



f = 500 pixels, *b* = 10 cm

- How to handle uncertainty in disparity and the other parameters (baseline, focal length)?

$$Z = f\frac{b}{d}$$

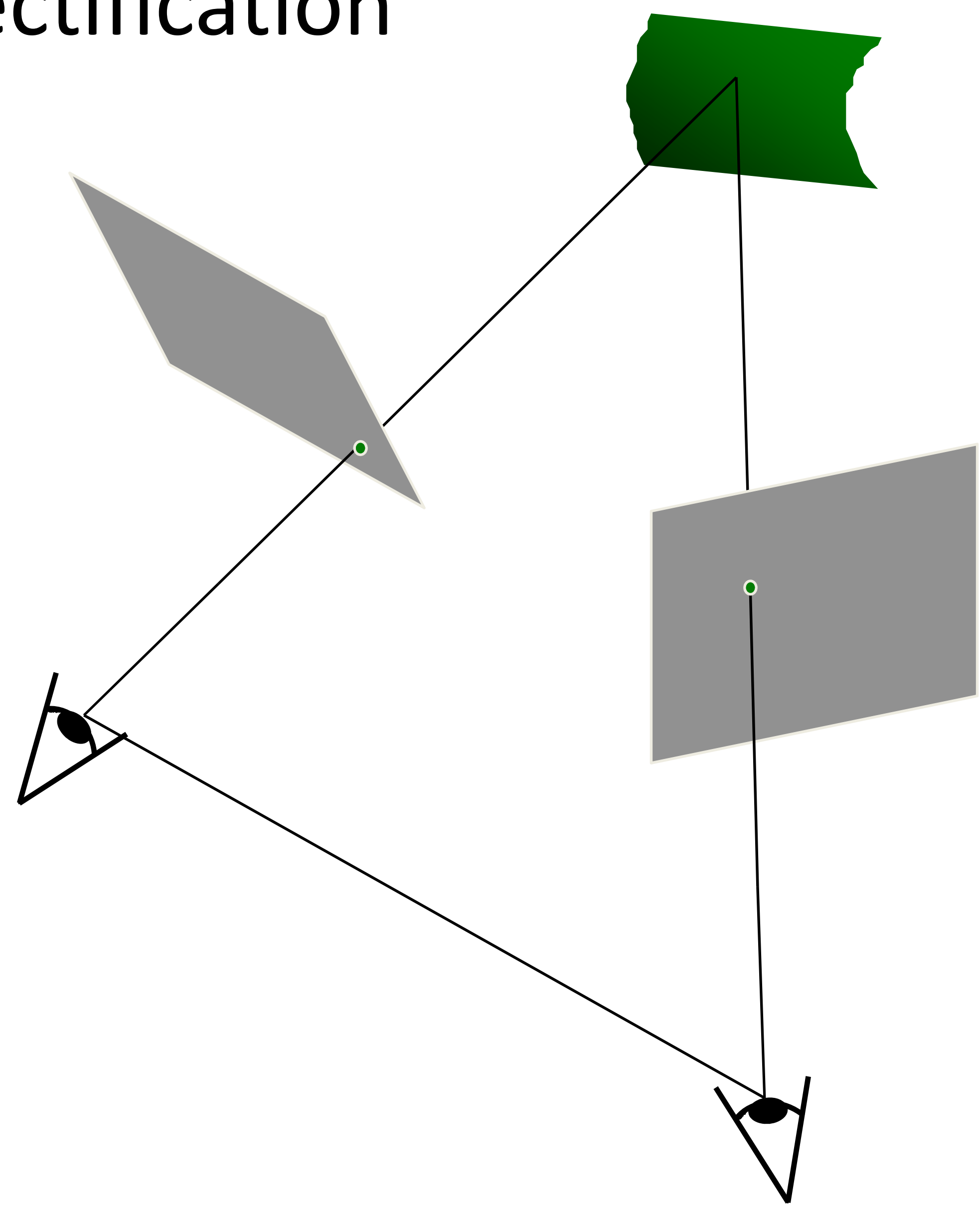
$$\Delta Z = \Delta f\frac{b}{d} + f\frac{b}{d^2}(-\Delta d) + \Delta b\frac{f}{d} = (\frac{fb}{d})\left(\frac{\Delta f}{f} - \frac{\Delta d}{d} + \frac{\Delta b}{b}\right)$$

$$(\sigma_Z/Z)^2 = \left(\sigma_f/f\right)^2 + (\sigma_d/d)^2 + (\sigma_b/b)^2$$

Assuming errors are independent of each other, and have zero mean

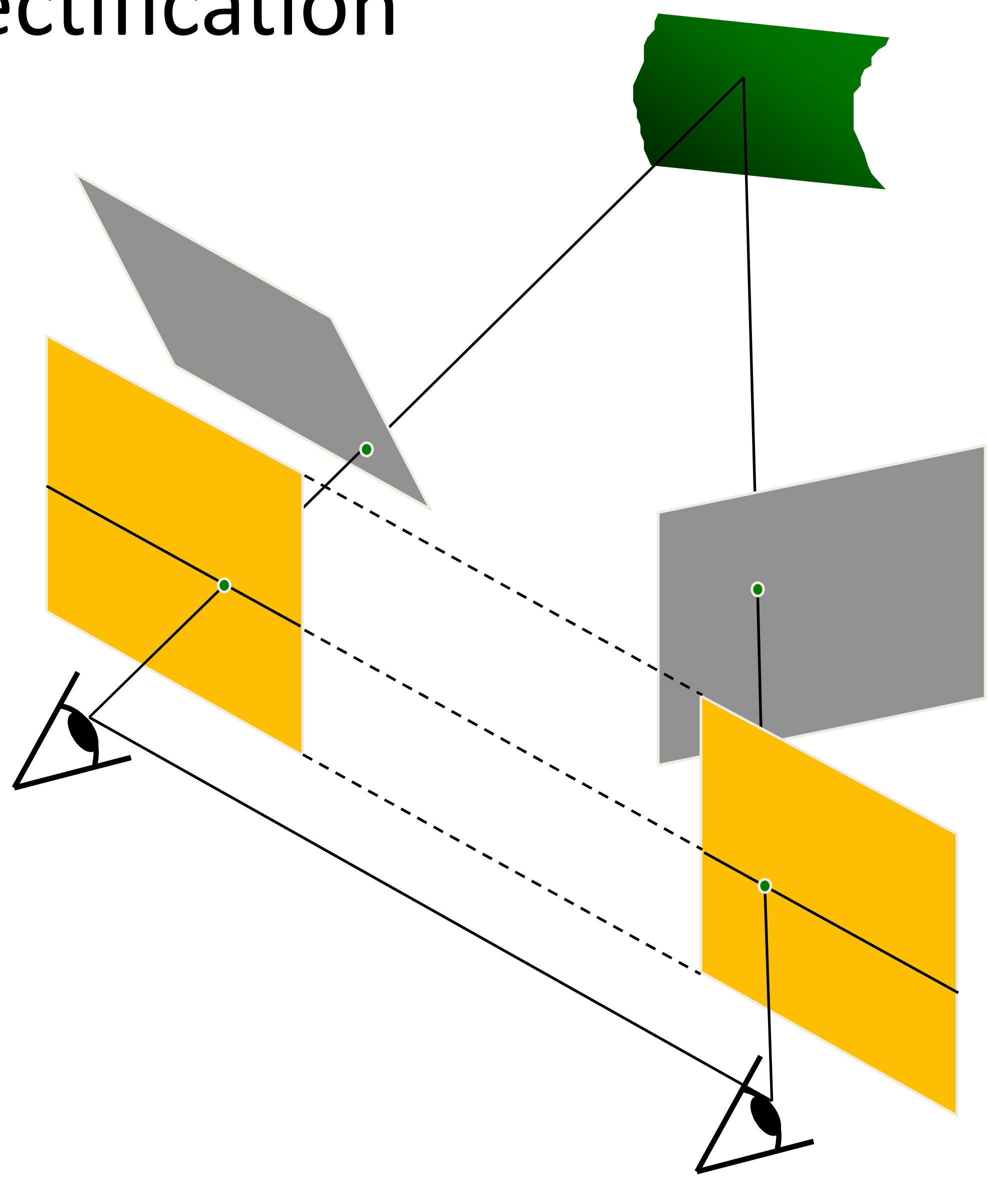
Stereo image rectification

- Sometimes the image planes are not co-planar



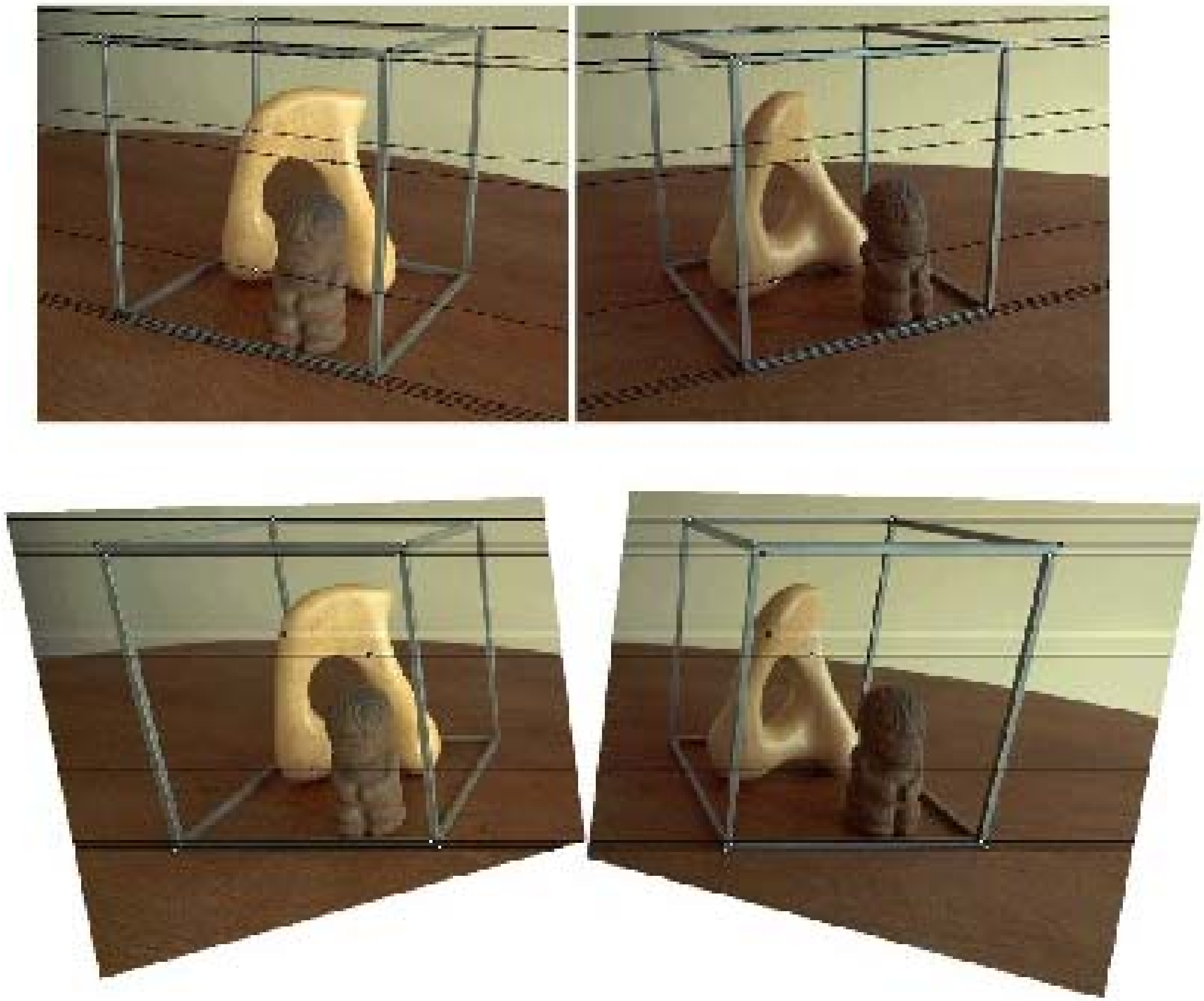
Stereo image rectification

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



Stereo Process

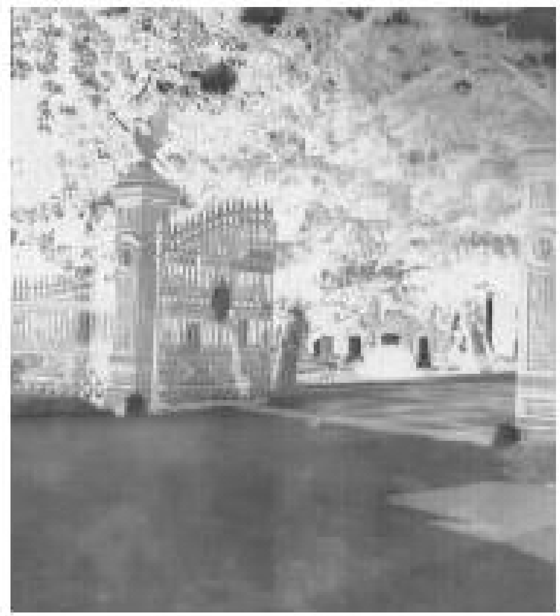
- Extract features from the left and right images
- Match the left and right image features, to get their disparity in position (the “correspondence problem”)
- Use stereo disparity to compute depth (the reconstruction problem)

http://vision.middlebury.edu/stereo/data/scenes2003/

- The correspondence problem is the most difficult

Characteristics of Human Stereo Vision

- Matching features must appear similar in the left and right images

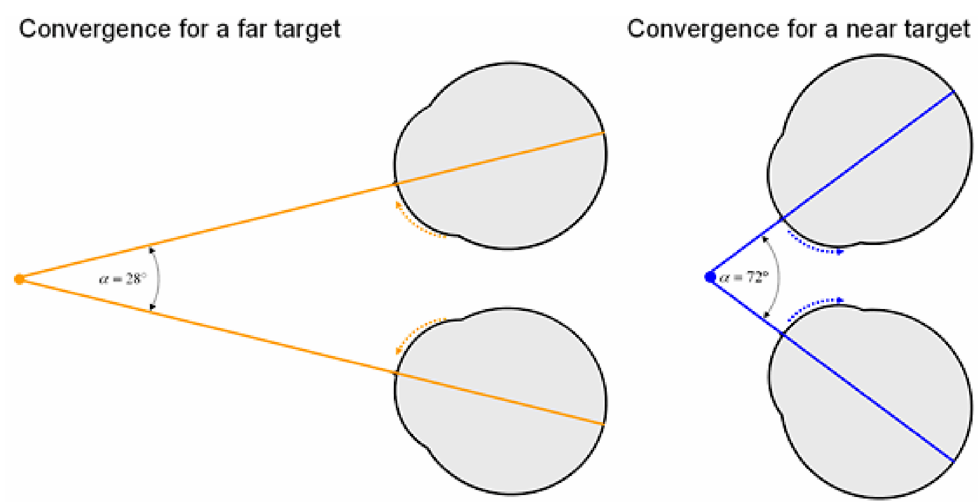



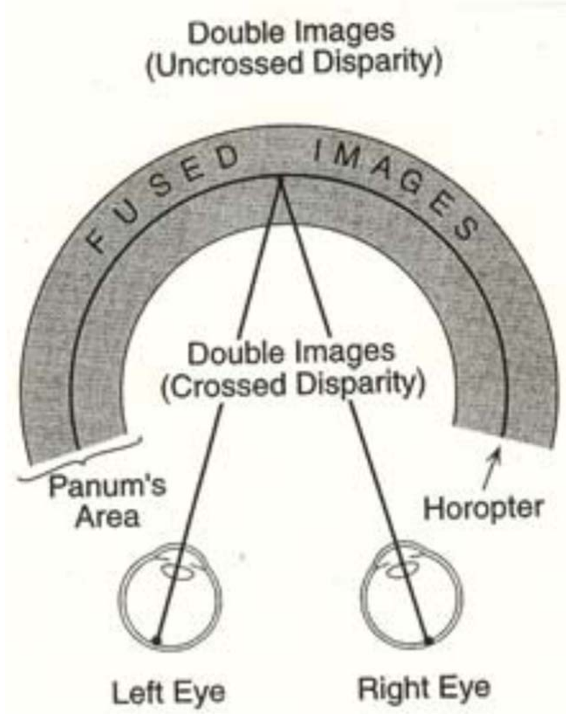
For example, we can't fuse a left stereo image with a negative of the right image…

http://cs.wellesley.edu/~cs332/

Characteristics of Human Stereo Vision

- Can only "fuse" objects within a limited range of depth around the fixation distance
- Vergence eye movements are needed to fuse objects over larger range of depths





http://cs.wellesley.edu/~cs332/

Panum’s Fusional Area

- Panum's fusional area is the range of depths for which binocular fusion can occur (without changing vergence angles)
- It’s actually quite small … we are able to perceive a wide range of depths because we are changing vergence angles

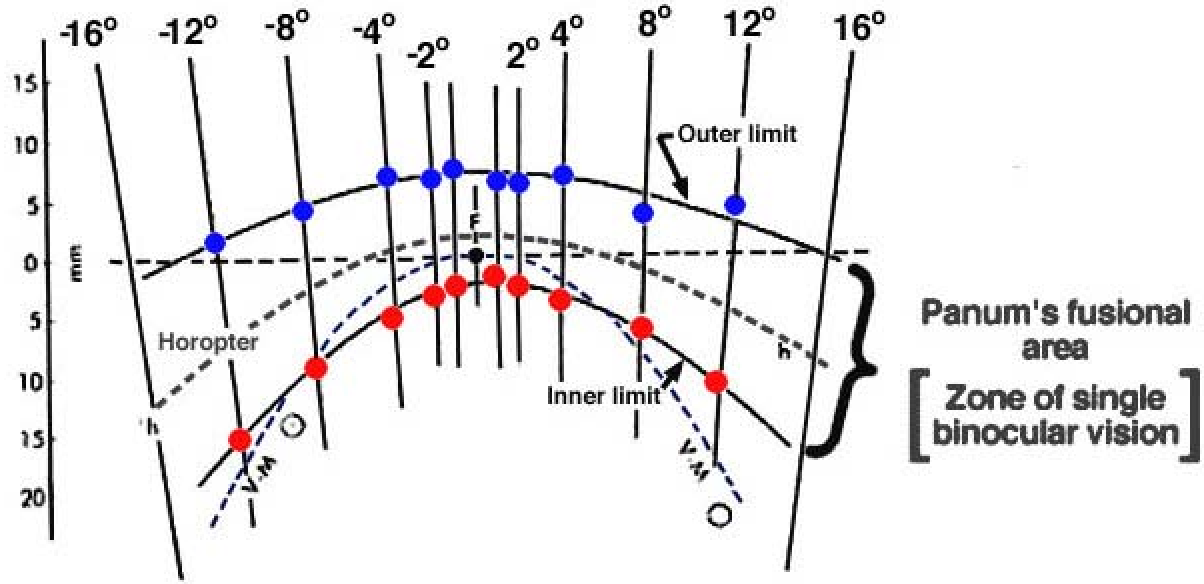
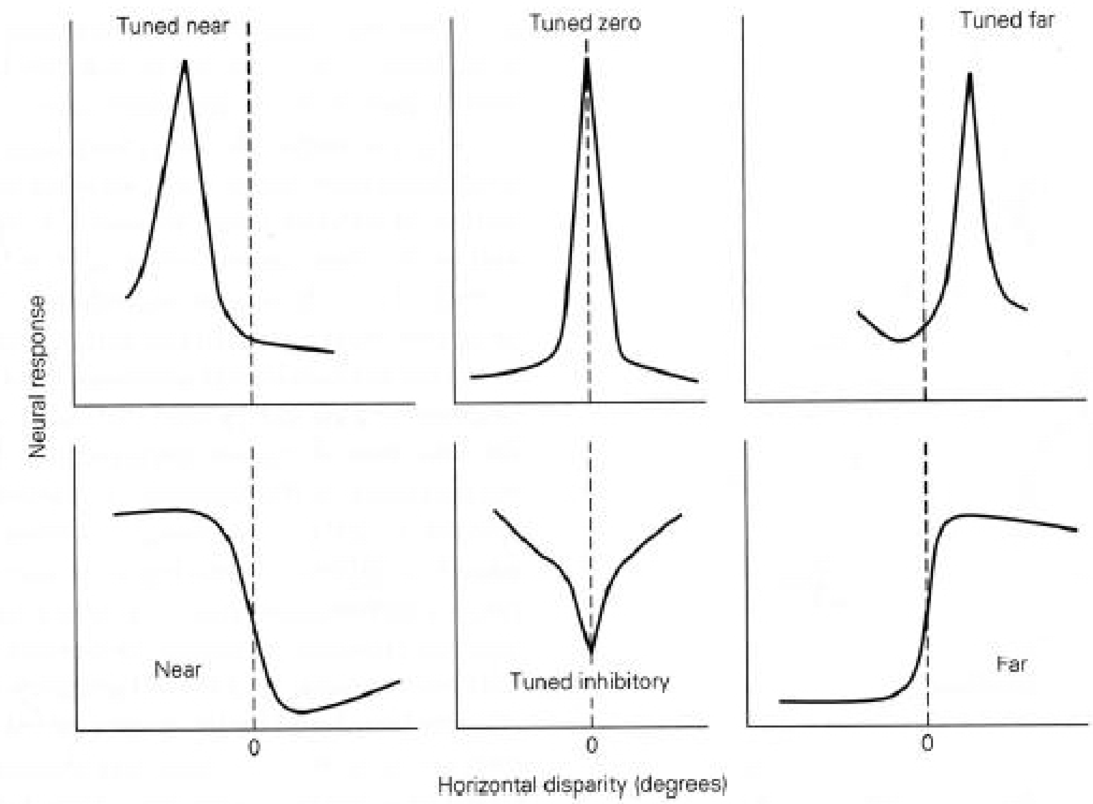


Figure 7. Haplopic method of determining the horopter involves locating the region of single binocular vision at a distance of 40cm. Panum's fusional area lies between the outer and inner limits of the region of single binocular vision.

http://webvision.med.utah.edu/imageswv/KallDepth7.jpg

Characteristics of Human Stereo Vision

- Cells in visual cortex are selective for stereo disparity
- Neurons that are selective for a larger disparity range have larger receptive fields

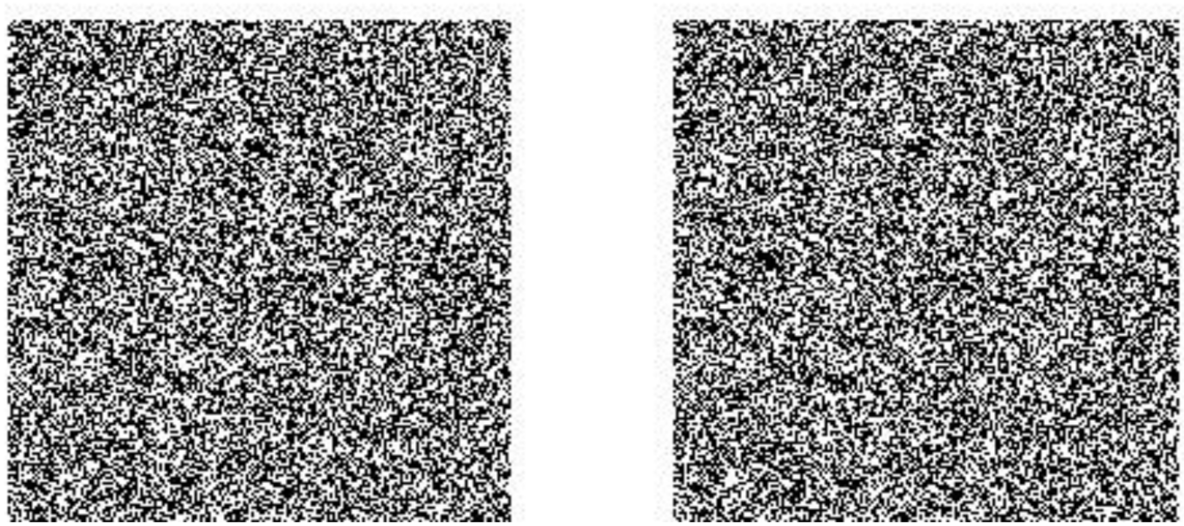


- zero disparity: at fixation distance
- near: in front of point of fixation
- far: behind point of fixation

http://cs.wellesley.edu/~cs332/

Characteristics of Human Stereo Vision

- Can fuse random-dot stereograms



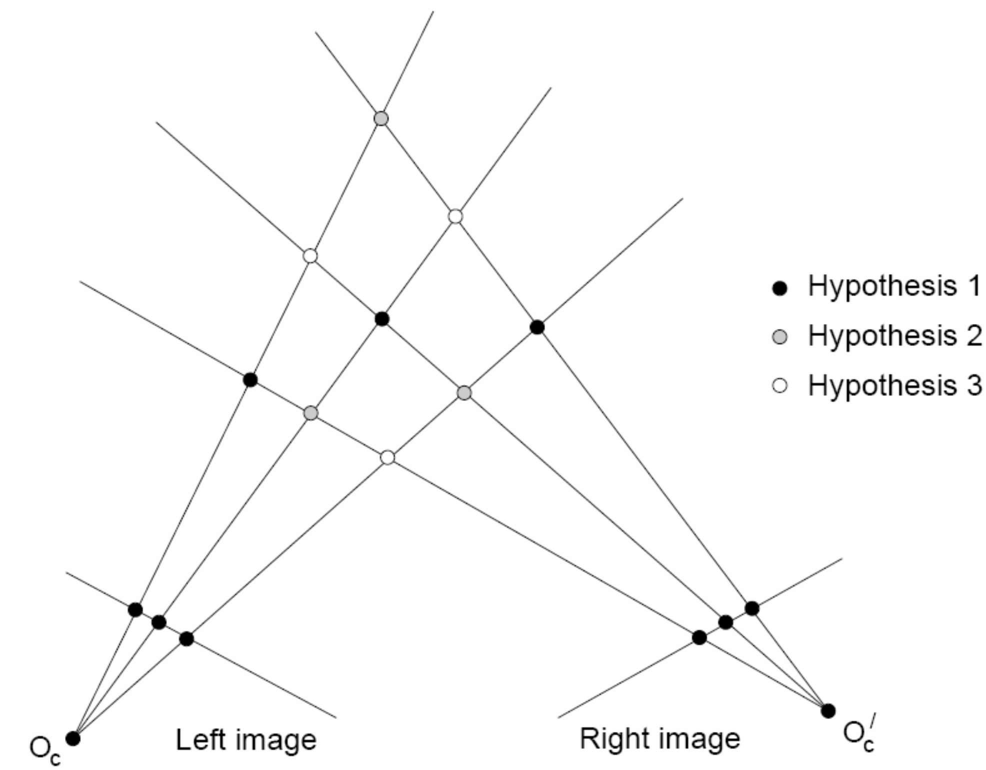
Bela Julesz, 1971

- Shows
  - Stereo system can function independently
  - We can match “simple” features
  - Highlights the ambiguity of the matching process

http://cs.wellesley.edu/~cs332/

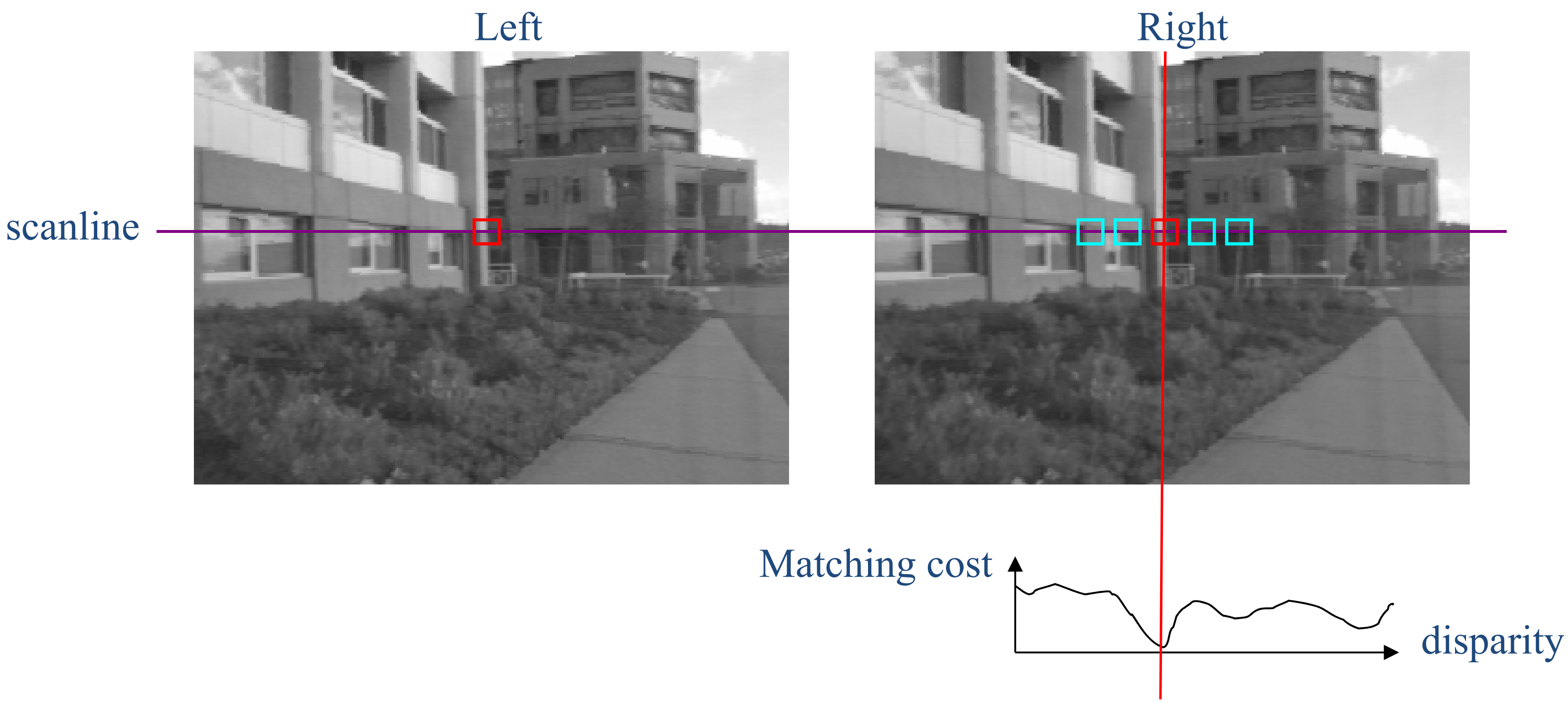
Correspondence Problem – Most difficult part of stereo vision

- For every point in the left image, there are many possible matches in the right image
- Locally, many points look similar -> matches are ambiguous



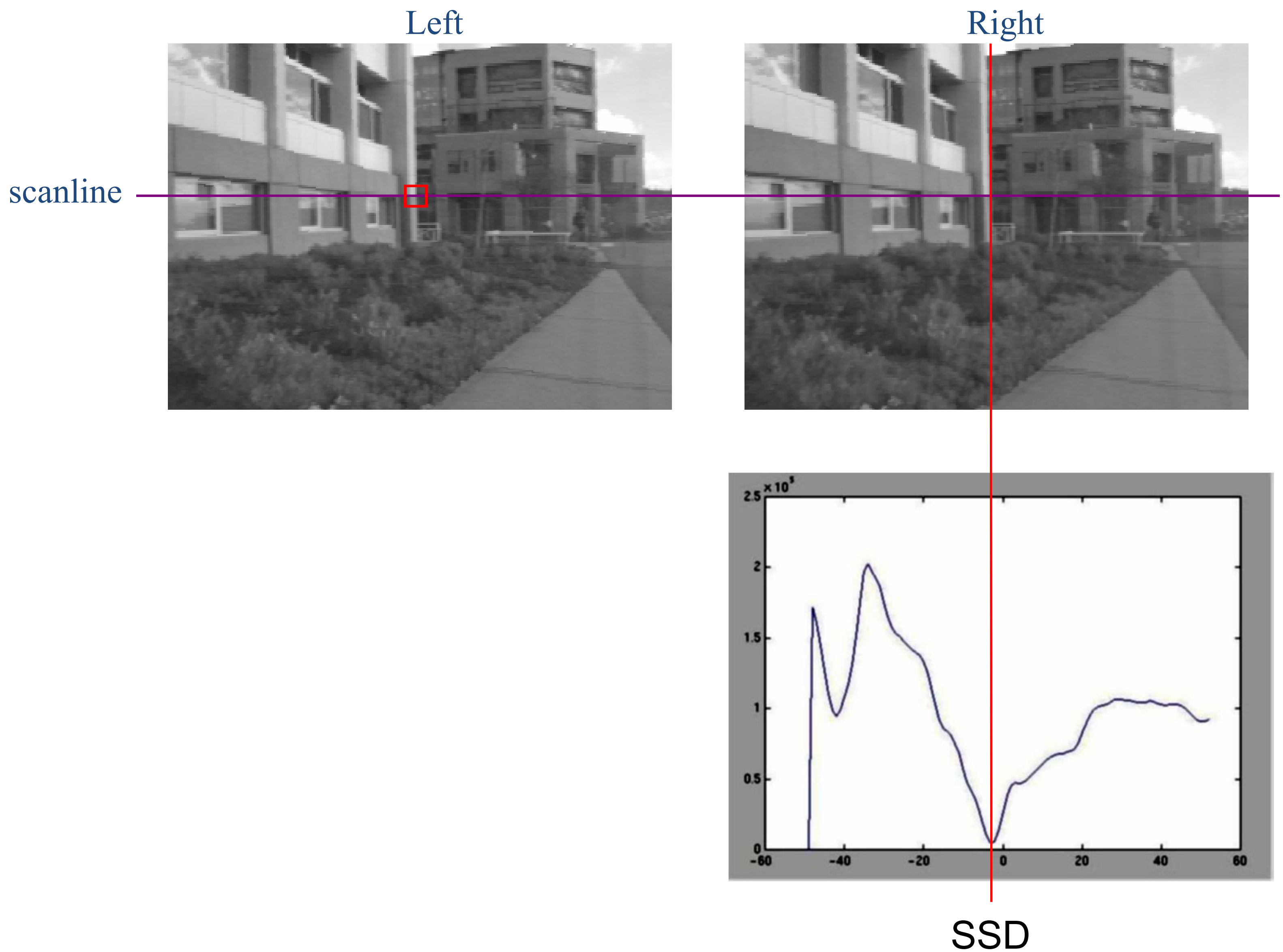
- Multiple matching hypotheses satisfy the epipolar constraint, but which one is correct?

Simple Correspondence Search: Block Matching

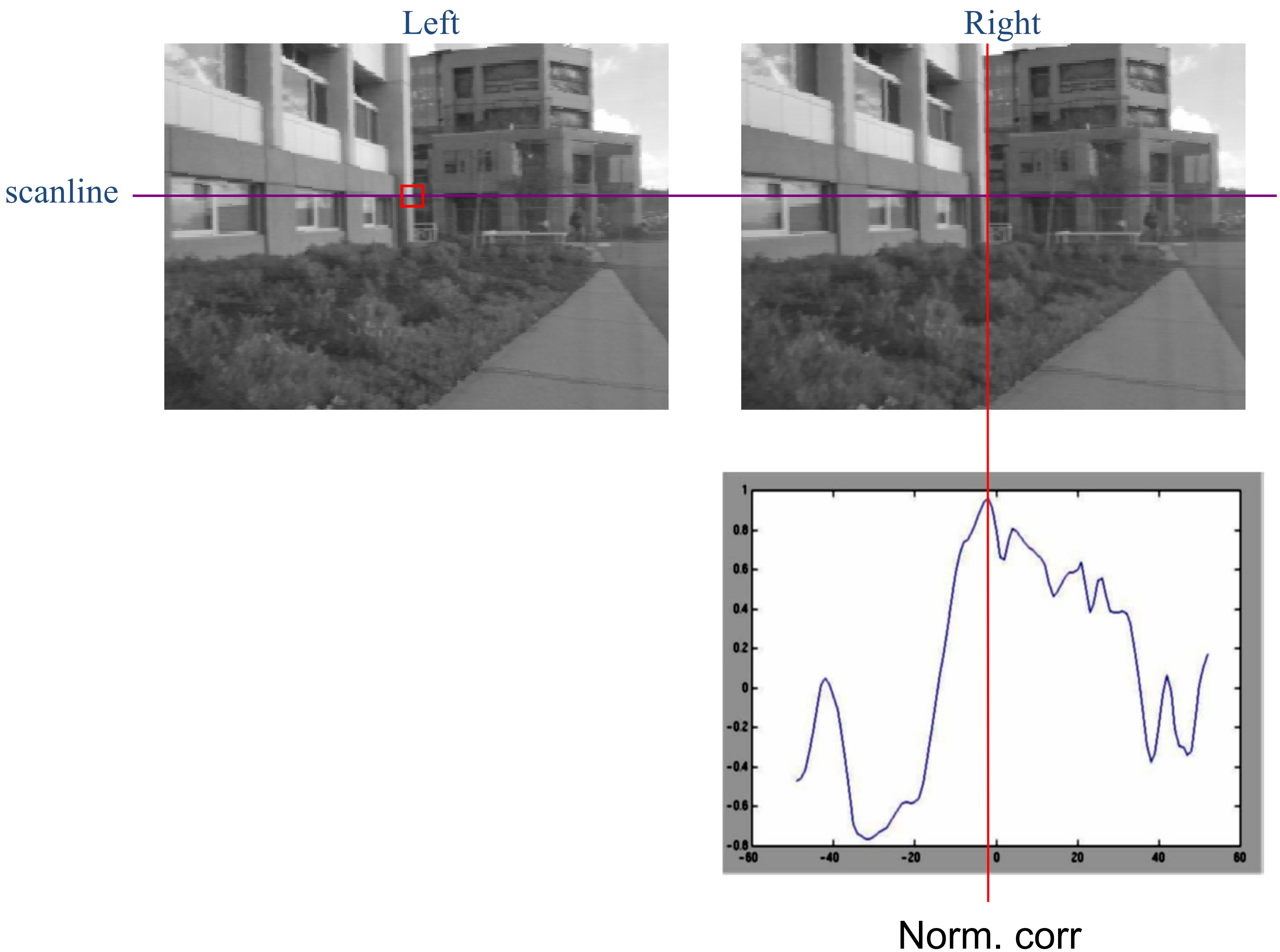


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD, SAD, or normalized correlation

Simple Correspondence Search: Block Matching

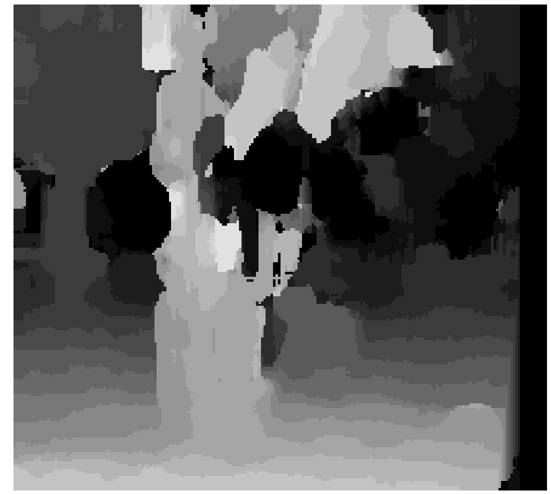


Simple Correspondence Search: Block Matching



Effect of window size

W = 3



W = 20

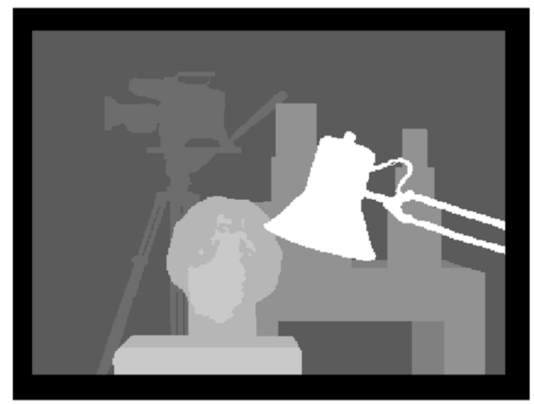
- Smaller window
  - \+ More detail
  - • More noise

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

Matlab Example

- Download images from the course website:
  - scene1.row3.col1.ppm
  - scene1.row3.col3.ppm
  - truedisp.row3.col3.pgm

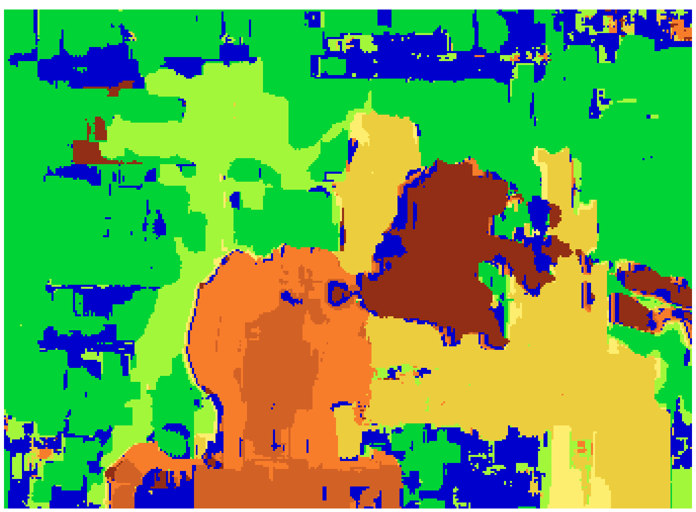
“tsukuba” images from the website
http://vision.middlebury.edu/stereo/



- Run program “stereo_BasicBlockMatch.m”

Results with window search

Window-based matching



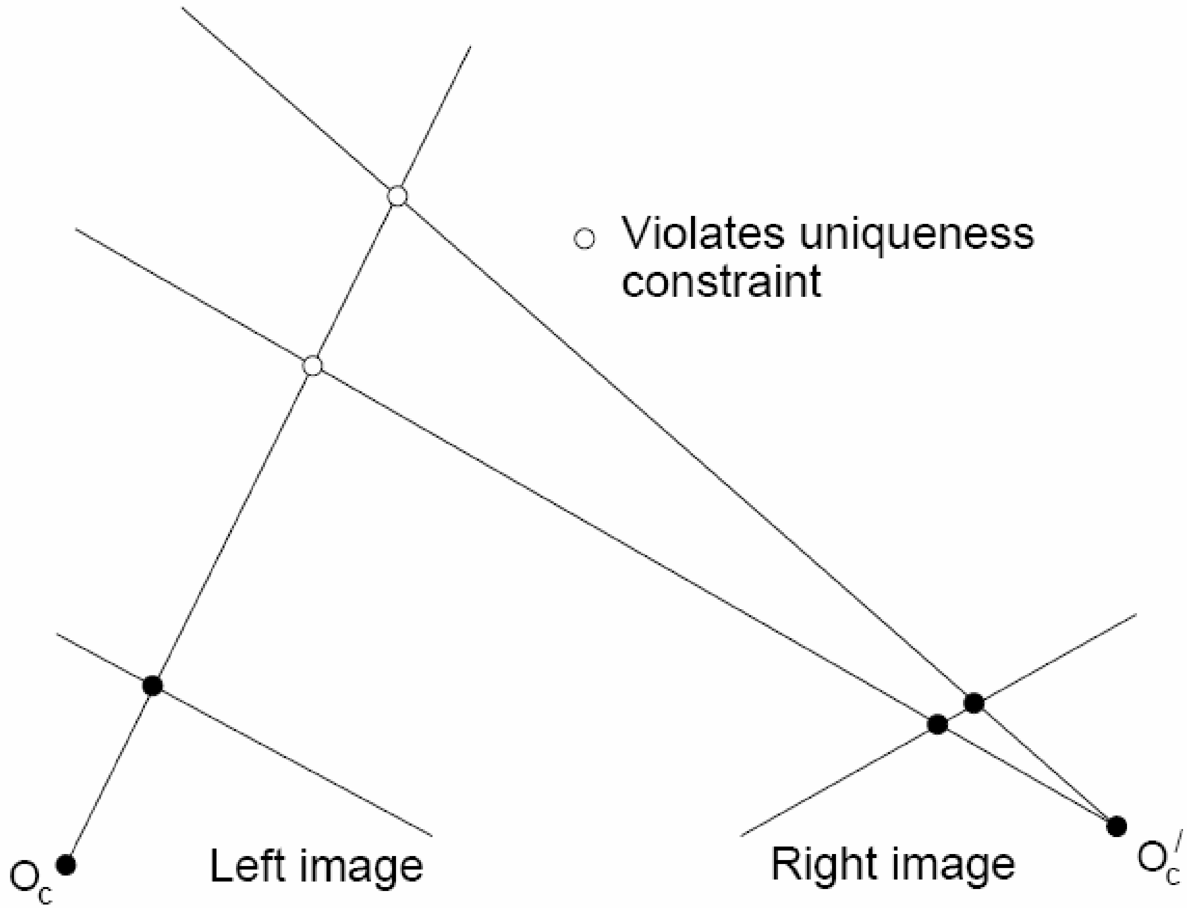
Ground truth

How can we improve window-based matching?

- The similarity constraint is **local** (each reference window is matched independently)
- Need to enforce **non-local** correspondence constraints
  - Uniqueness
  - Ordering
  - Smoothness

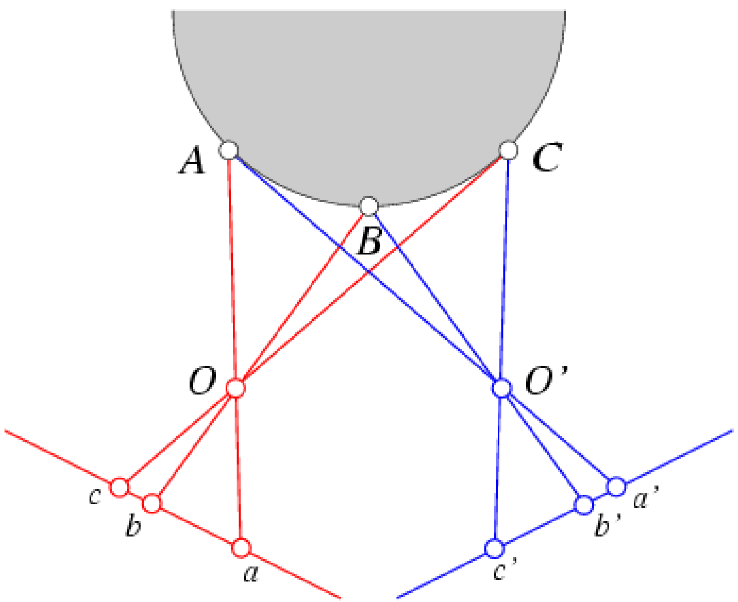
Non-local constraints

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



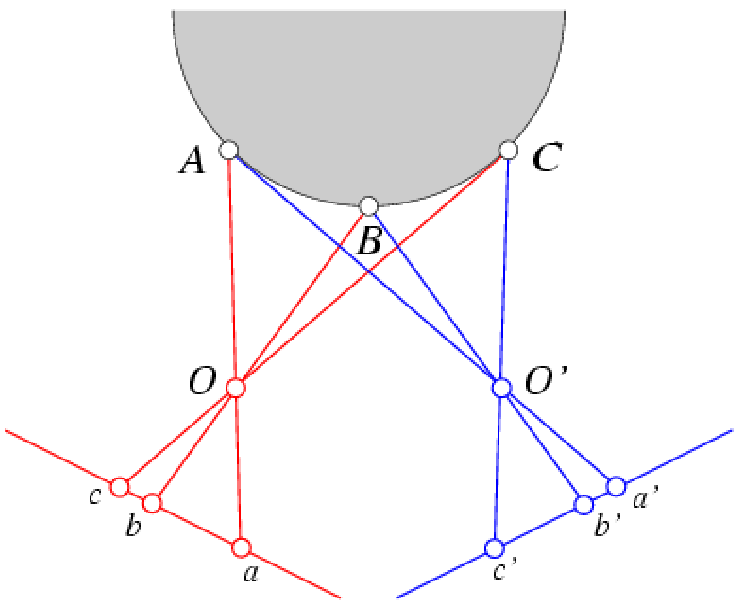
Non-local constraints

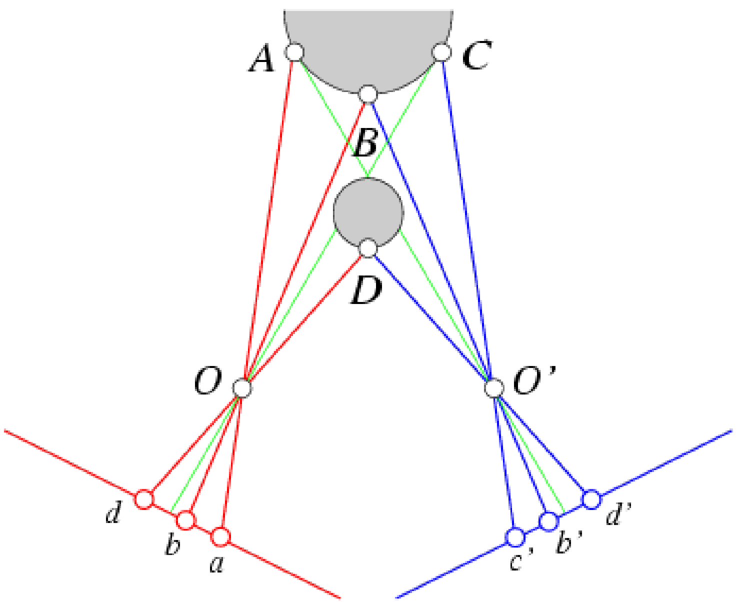
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



Non-local 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





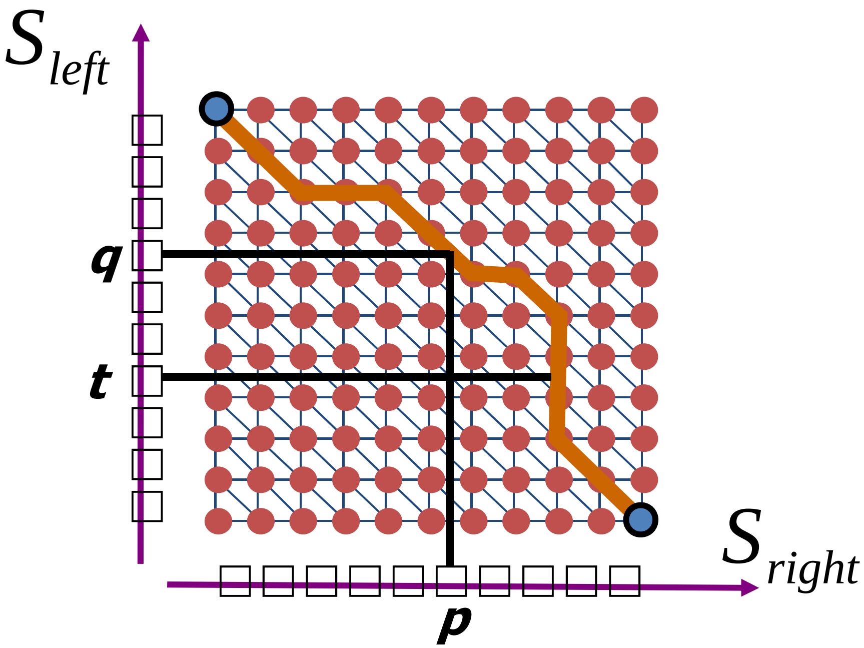
Sometimes constraint doesn't hold

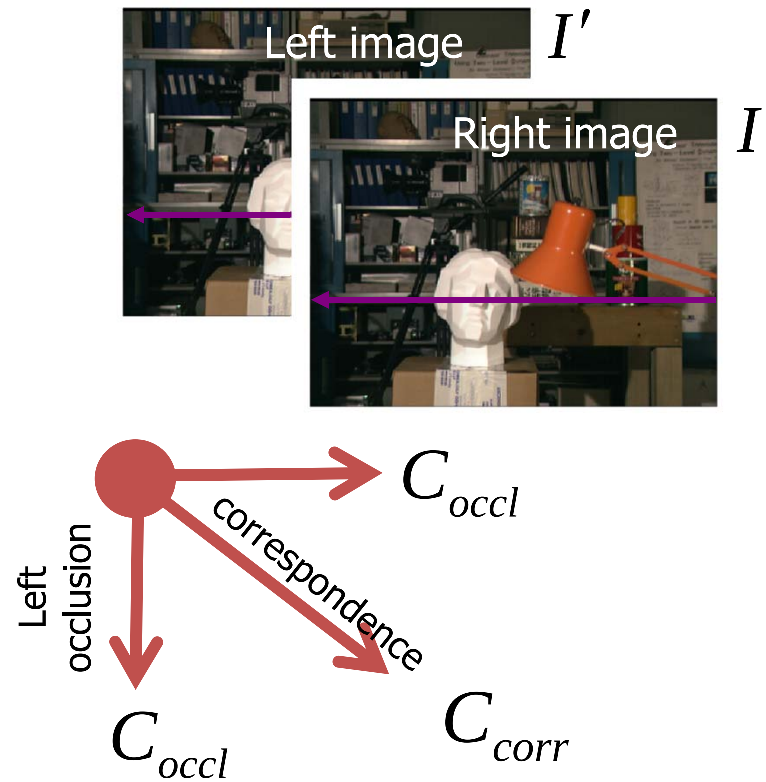
Non-local constraints

- 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)

Approach: Scanline stereo

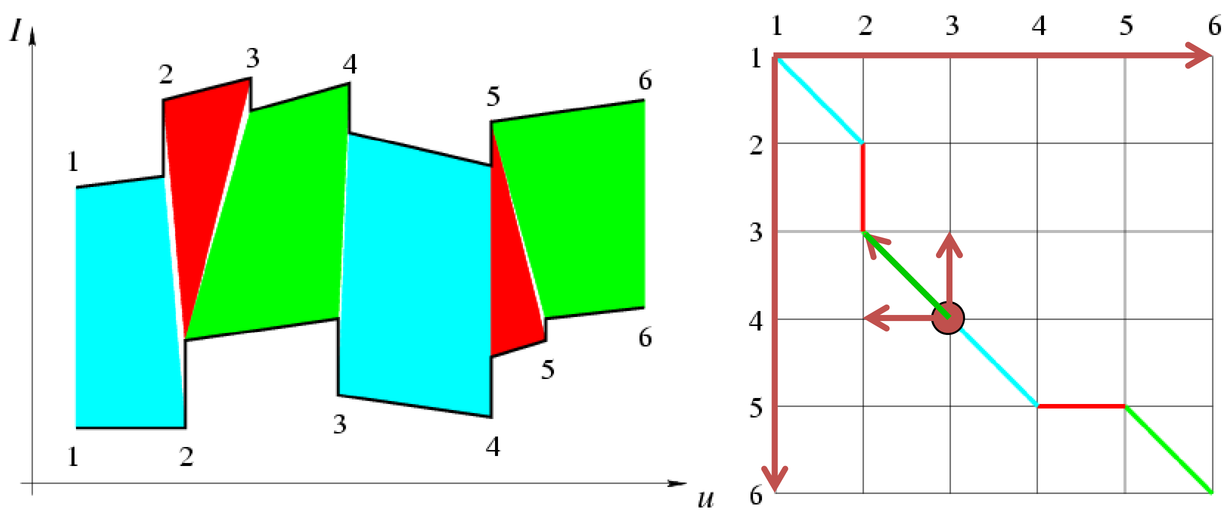
- Try to coherently match pixels on each scanline
- Search for lowest cost path through space of correspondences





Approach: Scanline stereo

- Scanline stereo can be implemented efficiently using "dynamic programming"
  - Dynamic programming is a general method for solving a complex problem by breaking it down into a collection of simpler subproblems
  - Each subproblem is solved just once, and the solution is saved

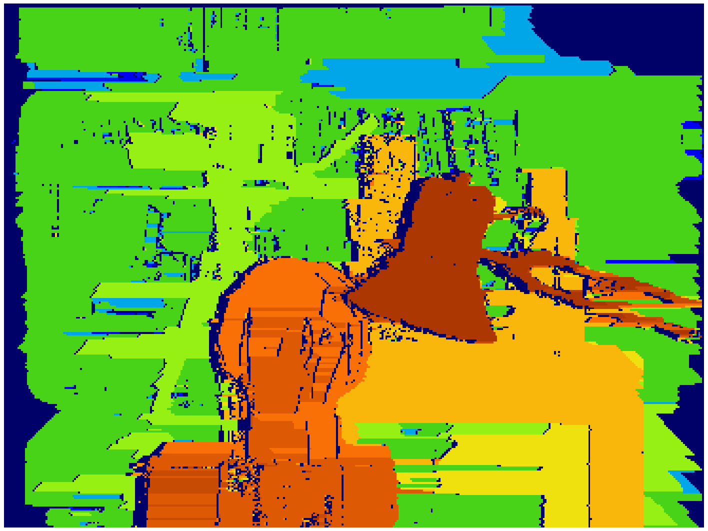


The optimal solution at each node is determined by the solutions above and to the left

www.cs.nccu.edu.tw/~whliao/acv2008/acv07b.ppt

Matlab Example

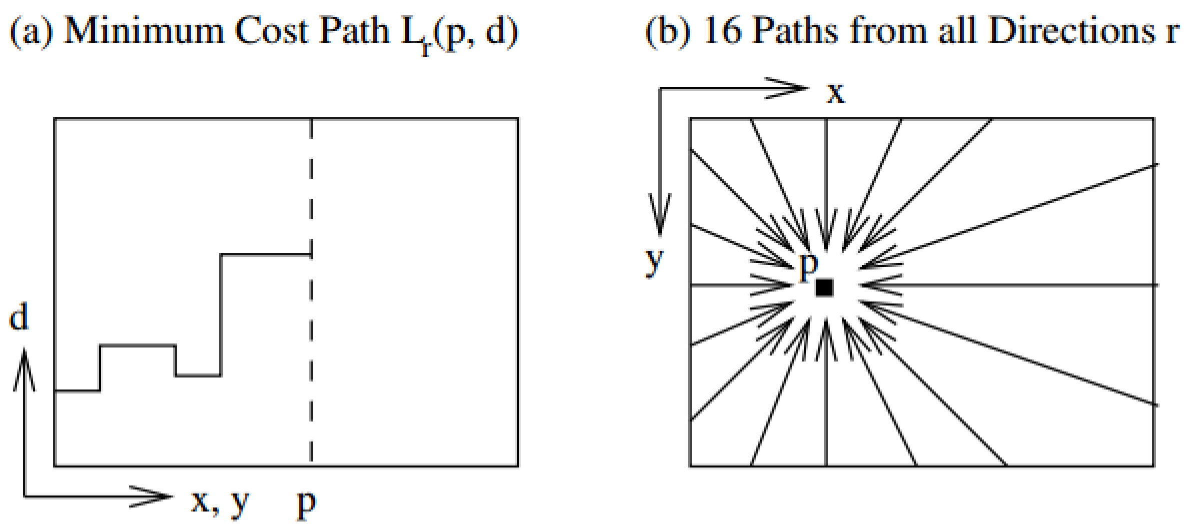
- Run program “stereo_BlockMatchDynamicProg.m”



- Scanline stereo generates horizontal streaking artifacts

Better solution: Semi-Global Matching

- Do scanline optimization from multiple directions and sum the cost



Hirschmuller, Heiko. "Accurate and efficient stereo processing by semi-global matching and mutual information." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, 2005.

This algorithm is implemented in OpenCV and the MATLAB Computer Vision toolbox

Matlab example

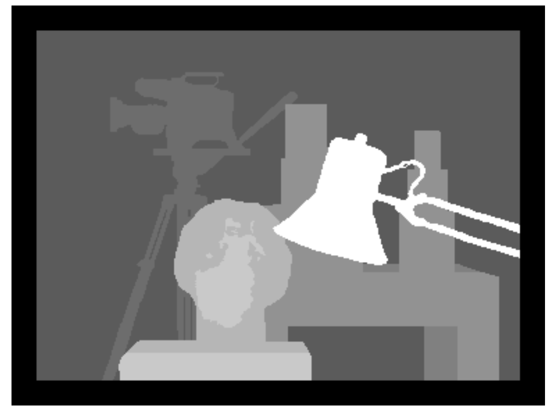
- See function “disparity” in Computer Vision toolbox

```
clear all
close all

% Load the images.
I1 = imread('scene1.row3.col1.ppm');
I2 = imread('scene1.row3.col3.ppm');

% Show stereo anaglyph. Use red-cyan stereo glasses to view image in 3-D.
figure
imshow(stereoAnaglyph(I1,I2));
title('Red-cyan composite view of the stereo images');

% Compute the disparity map.
Method = 'SemiGlobal';      % 'SemiGlobal' (default) | 'BlockMatching'
disparityRange = [0 32];    % [min max], diff must be divisible by 16
BlockSize = 15;             % 15 (default) | odd integer

disparityMap = disparity(rgb2gray(I1),rgb2gray(I2), ...
    'Method', Method, ...
    'BlockSize',BlockSize, ...
    'DisparityRange',disparityRange);

% Display the disparity map. For better visualization, use the disparity
% range as the display range for imshow.
figure
imshow(disparityMap, disparityRange);
title('Disparity Map');
```

Compare to ground truth

- Ground truth disparities are in "truedisp.row3.col3.pgm"
- Note: according to the website, the ground truth disparity image should be divided by 8 to yield to correct values of disparity

True disparities



truedisp.row3.col3.pgm

- Identify "bad" pixels in the result (those with disparity errors greater than say, 2.0)
- Compare your percentage of bad pixels to the best stereo algorithm on http://vision.middlebury.edu/stereo/eval/

```
%%%%%%%%%%%%%%
% Compare to ground truth.
Dtrue = imread('truedisp.row3.col3.pgm');
Dtrue = single(Dtrue);
Dtrue = Dtrue/8;

% Identify reliable disparities in the result.  According to the
% documentation on disparity, unreliable disparities are given the value
% -realmax('single').
Ireliable = (disparityMap ~= -realmax('single'));

% Ignore the border, where the true disparities are not defined.  It looks
% like valid disparities are greater than zero, in the ground truth.
Ivalid = (Dtrue > 0);

% Get the error between the true and measured disparities.
Derror = disparityMap - Dtrue;

Derror = Derror .* Ivalid .* Ireliable;
figure, imshow(Derror, []), title('Disparity errors in valid region'), impixelinfo;


% Measure % of "bad" pixels; ie those with a disparity error above DTHRESH.
nPixels = sum(Ivalid(:));       % Number of valid pixels
DTHRESH = 2;
Ibad = abs(Derror) > DTHRESH;
figure, imshow(Ibad, []), title('Large disparity errors'), impixelinfo;
```