Stereopsis

Exam

#1 5 pts "-2.5 any computational error, error with units, etc"

#2 2pts each all or nothing (tend to give credit unless answer is very far off)

i. (answers for 'perspective projection' were uniformly weak, so I was unusually liberal with credit)

iii must mention parallel lines

iv. Partial credit unavoidable here. -.5 per missed

vi. Must mention neighbors, or a neighborhood, etc.

x. must say power per unit area, (Can use meters^2 instead of area, etc.)

I write OK if their answer is less than great, but I'm not taking off credit

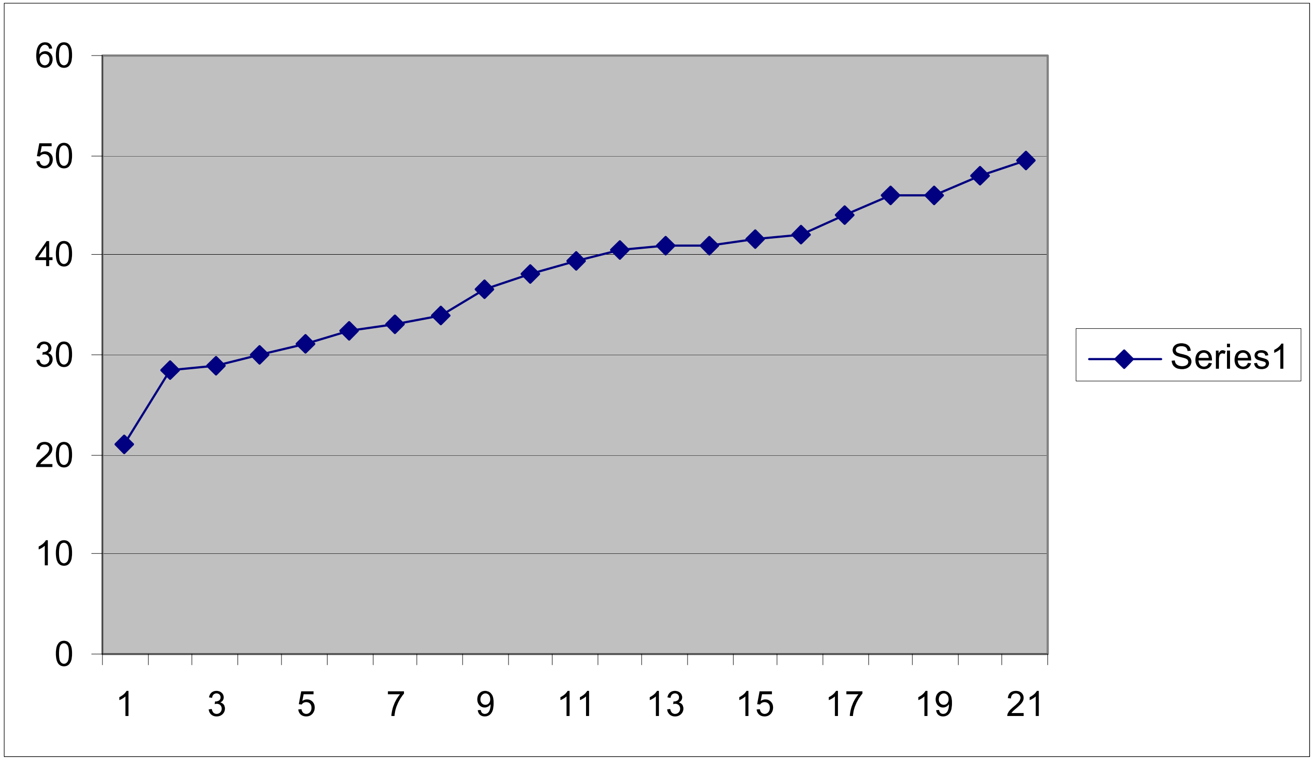
#3 4 pts formula for convolution of filter H with image I. Don't care about if they put limits on the summation.

8 pts the actual output. 1 pt per pixel. (there are 8 pixels fully covered.)

#4 9 pts -0 If they choose mask 1, regardless of reasoning'

-4 if they choose mask 2 or mask 3, and give some reasonably plausible reasoning'

-9 if they choose mask 4, or choose (2 or 3) with very bad reasoning

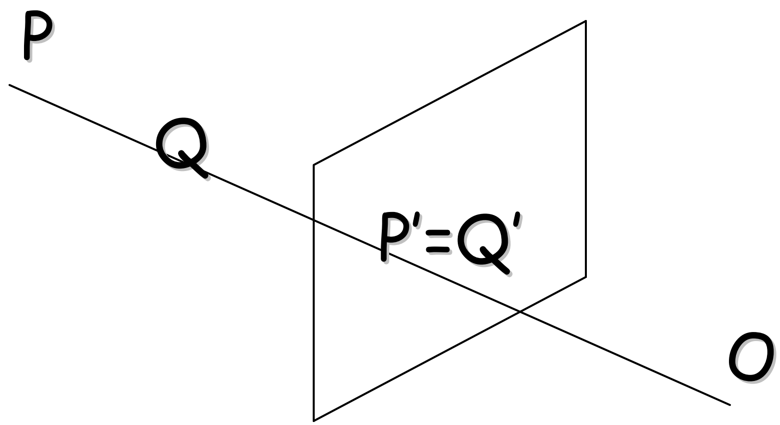

60
50
40
30
20
10
0
1
3
5
7
9
11
13
15
17
19
21
Series1

Mark Twain at Pool Table", no date, UCR Museum of Photography

Woman getting eye exam during immigration procedure at Ellis Island, c. 1905 - 1920 , UCR Museum of Phography

Why Stereo Vision?

- 2D images project 3D points into 2D:

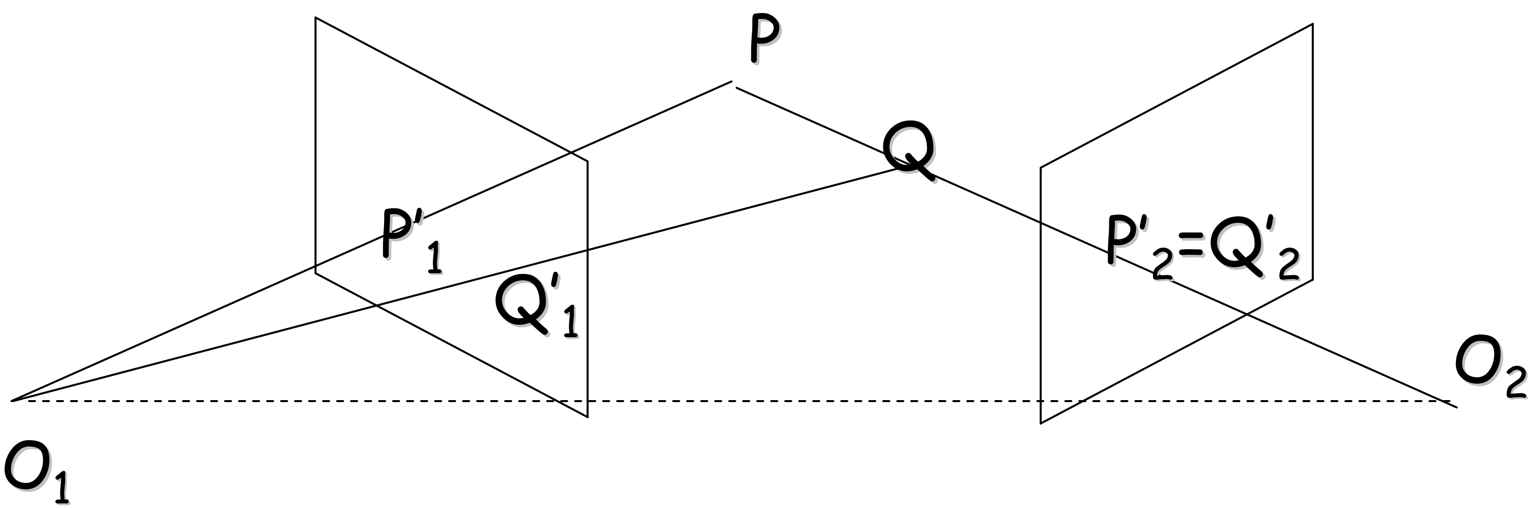


- 3D Points on the same viewing line have the same 2D image:
  - 2D imaging results in depth information loss

Stereo

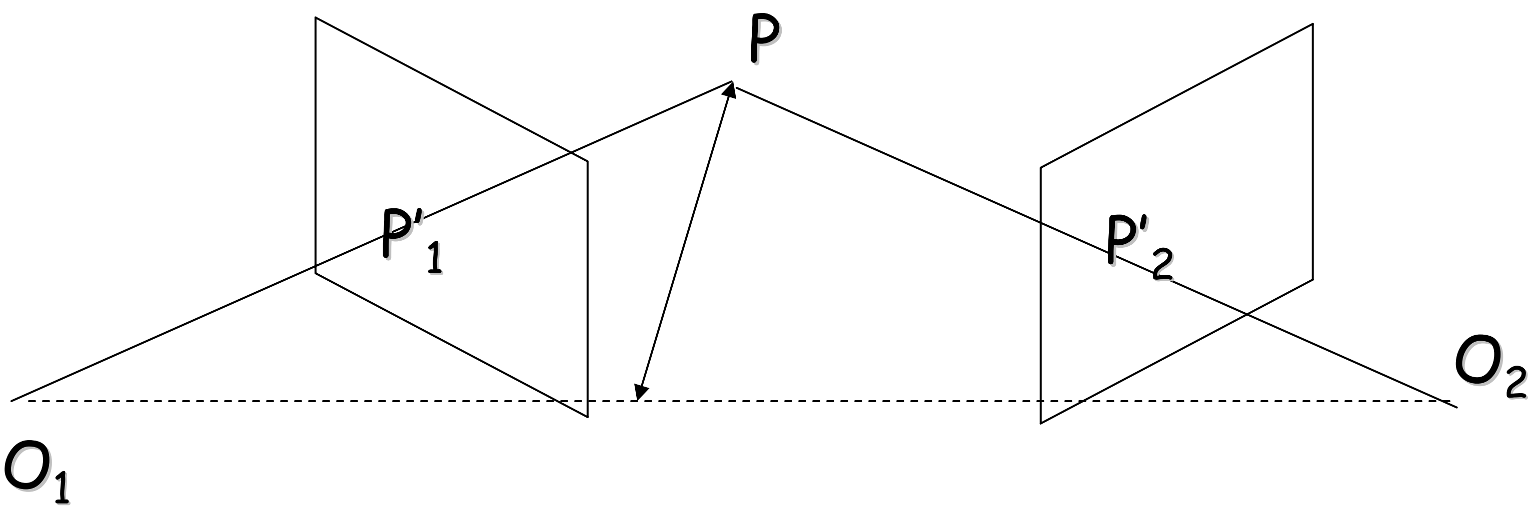
- Assumes (two) cameras.
- Known positions.
- Recover depth.

Recovering Depth Information:



Depth can be recovered with two images and triangulation.

3D Reconstruction

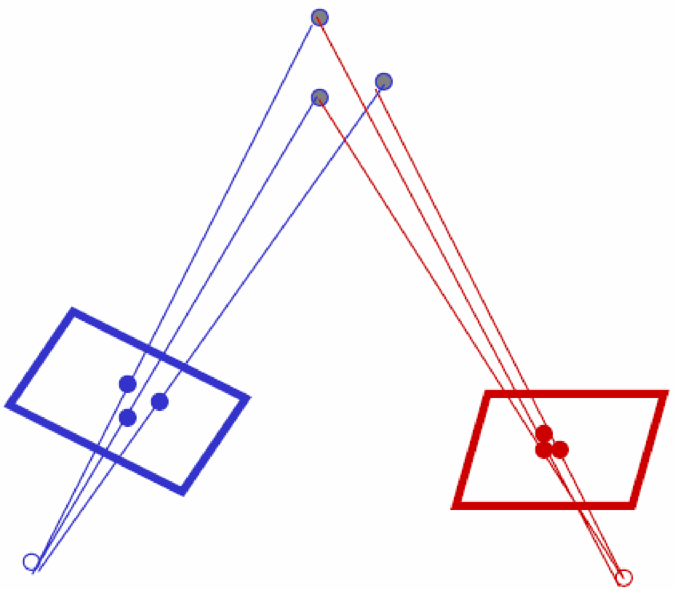


We must solve the correspondence problem first!

Multi-view geometry

Relate

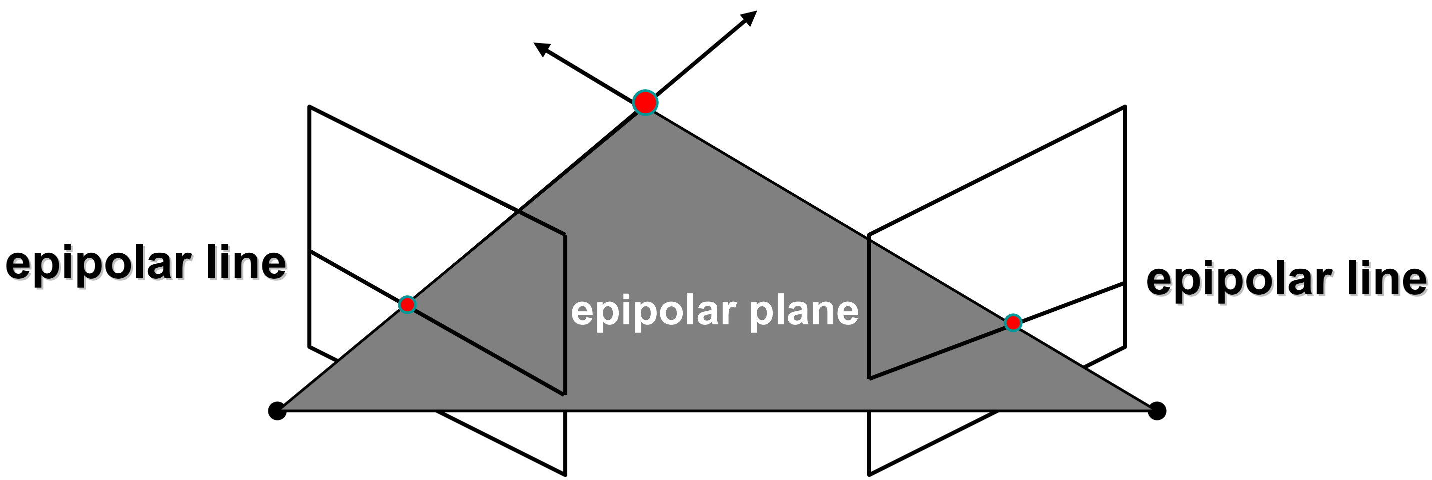
- 3-D points
- Camera centers
- Camera orientation
- Camera intrinsics



Slide adapted from: Darrell

Stereo correspondence

- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point



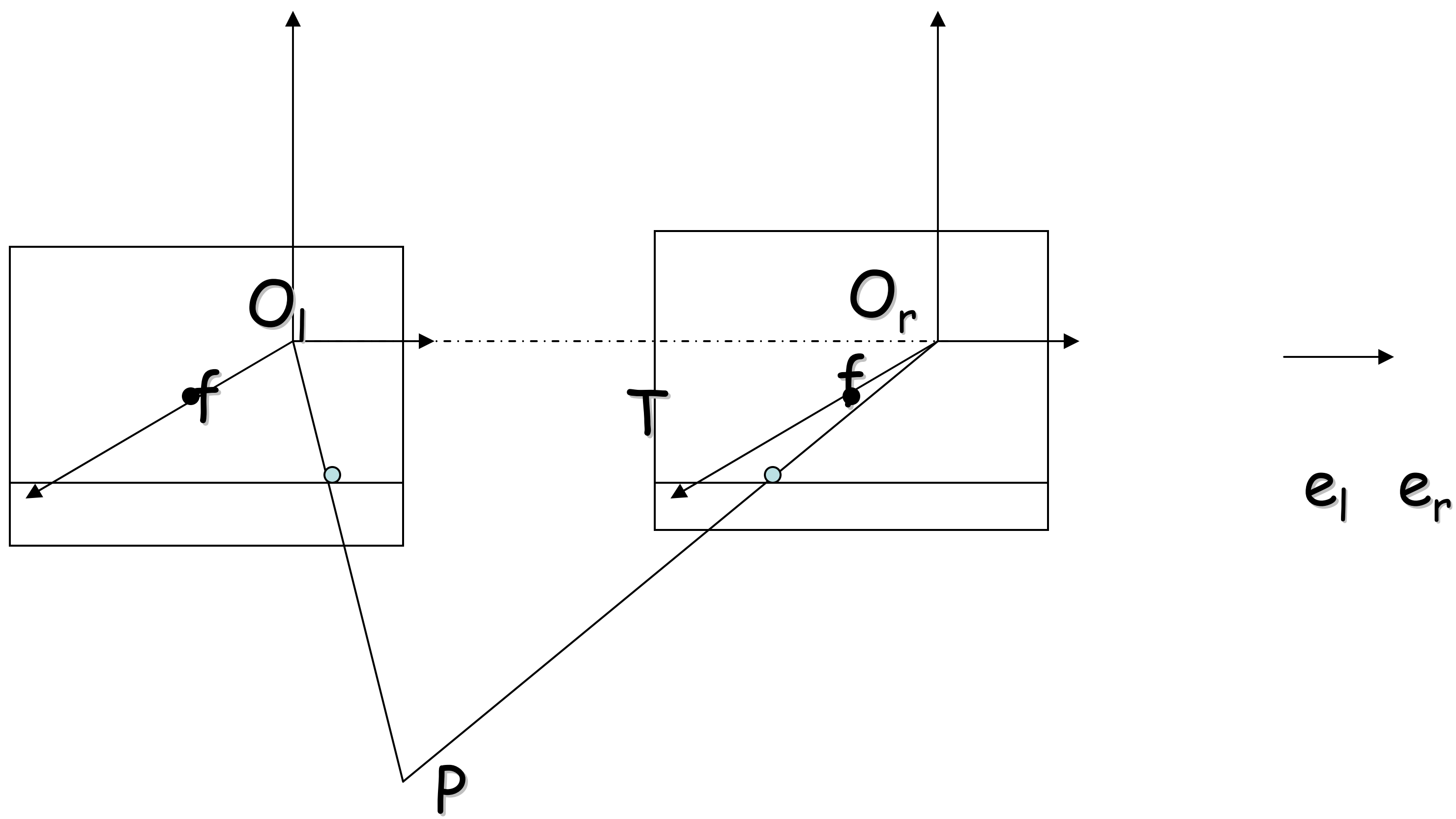
- Epipolar Constraint
  - Reduces correspondence problem to 1D search along *conjugate epipolar lines*

(Seitz)

Simplest Case

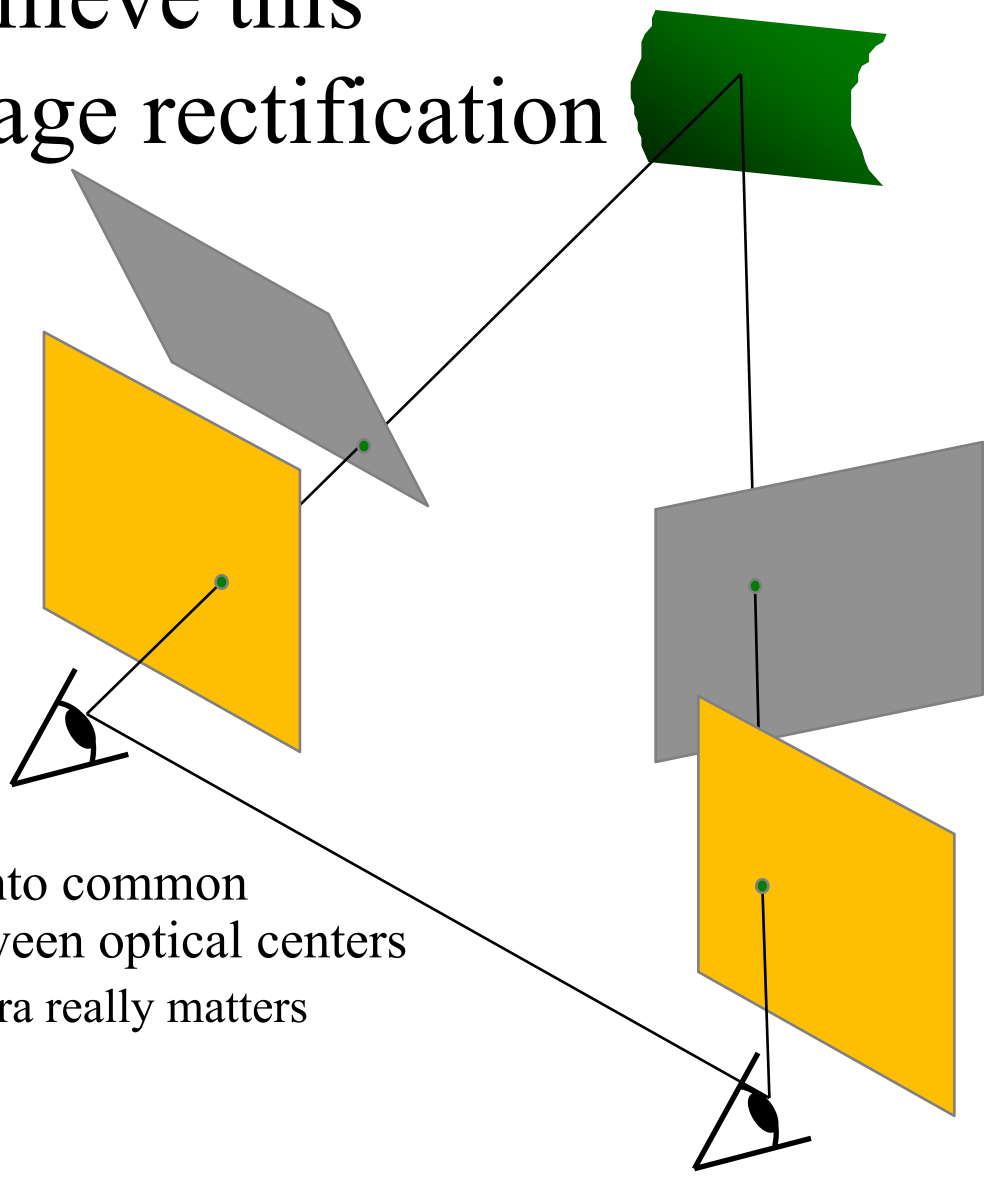
- Image planes of cameras are parallel.
- Focal points are at same height.
- Focal lengths same.
- Then, epipolar lines are horizontal scan lines.

Epipolar Geometry for Parallel Cameras



Epipoles are at infinity
Epipolar lines are parallel to the baseline

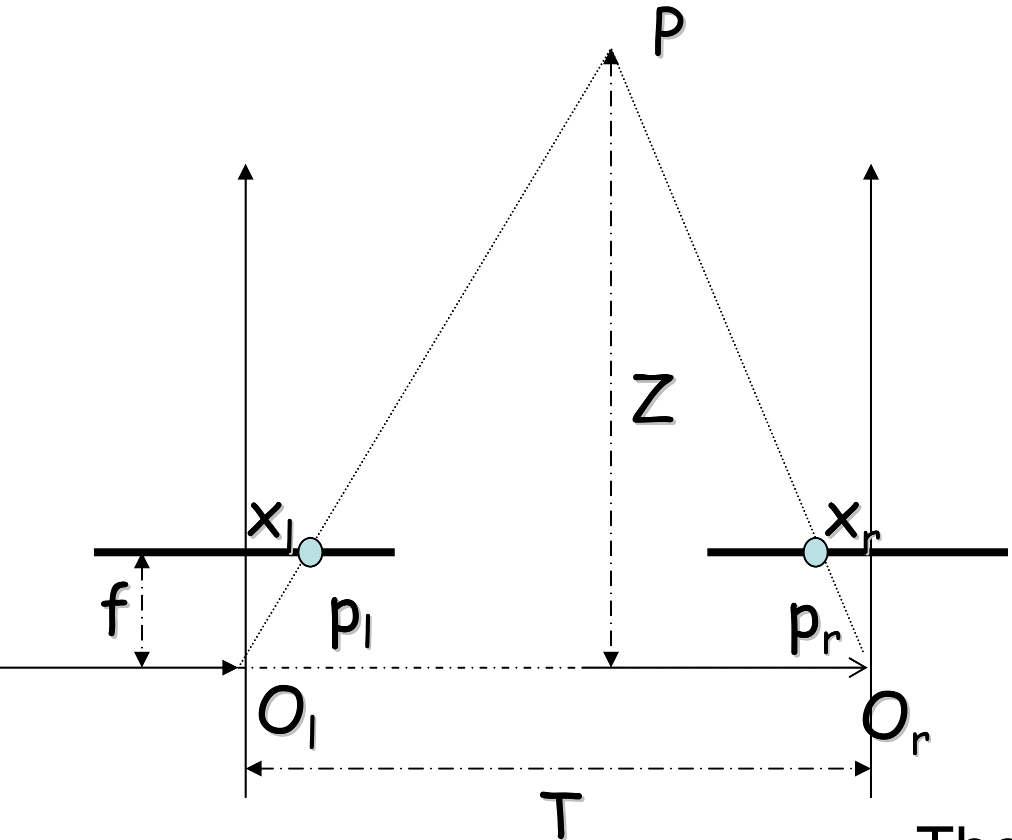
We can always achieve this geometry with image rectification



- Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers
- Notice, only focal point of camera really matters

(Seitz)

Let's discuss reconstruction with this geometry before correspondence, because it's much easier.



$$\frac{T + x_r - x_l}{Z - f} = \frac{T}{Z}$$

$$Z = f \frac{T}{x_l - x_r}$$

Disparity: $d = x_l - x_r$

$$\boxed{Z = f\frac{T}{d}}$$

Then given Z, we can compute X and Y.

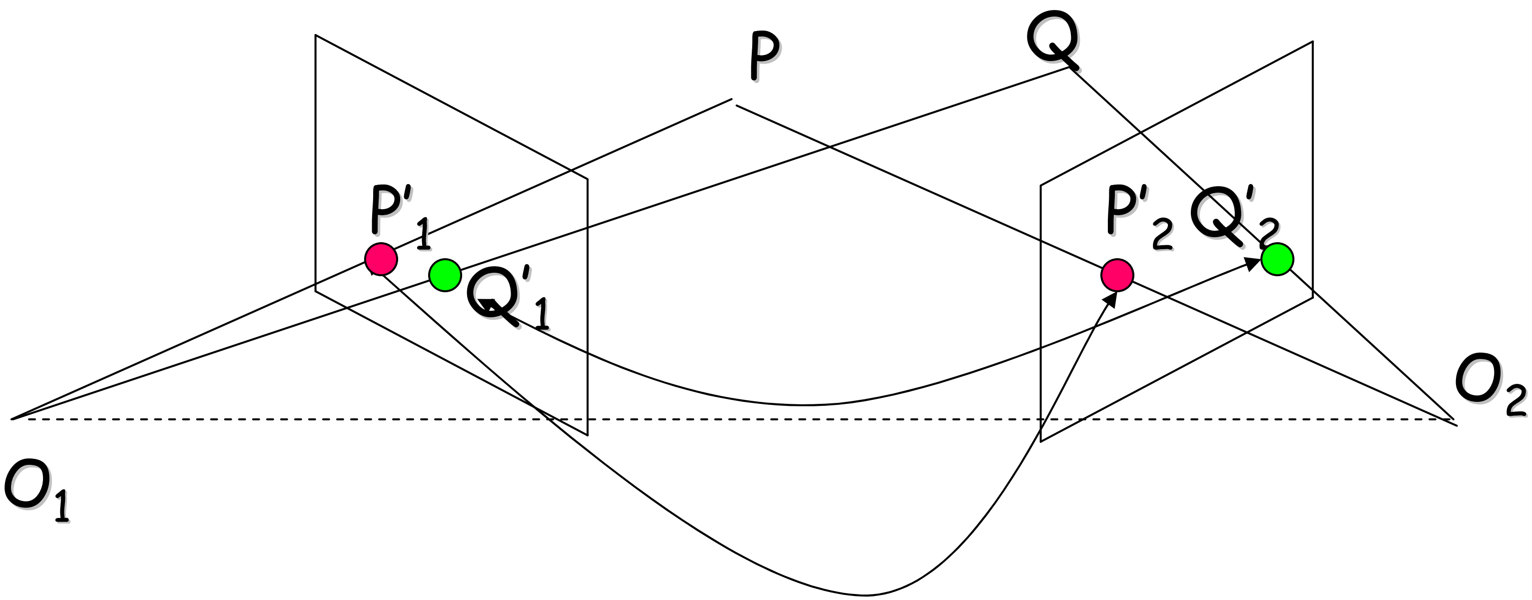
T is the stereo baseline

d measures the difference in retinal position between corresponding points

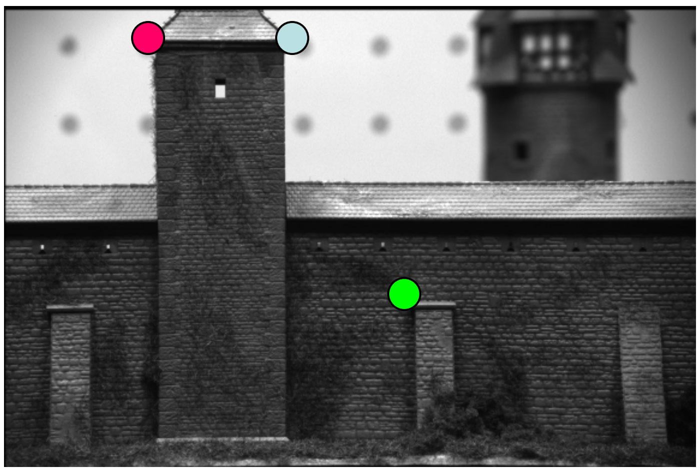
Correspondence: What should we match?

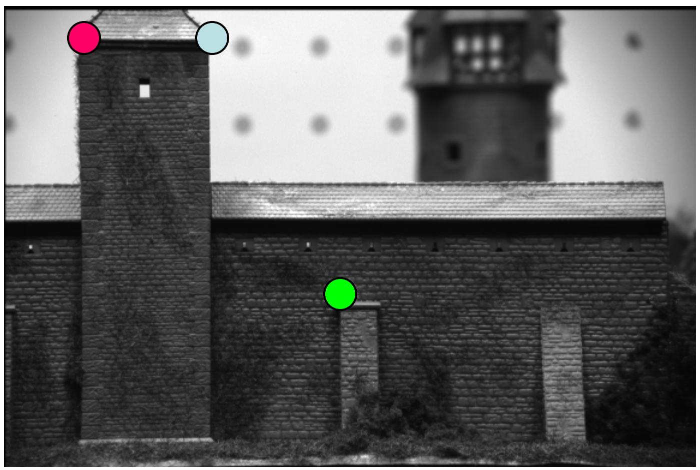
- Objects?
- Edges?
- Pixels?
- Collections of pixels?

Finding Correspondences:



Finding Correspondences:



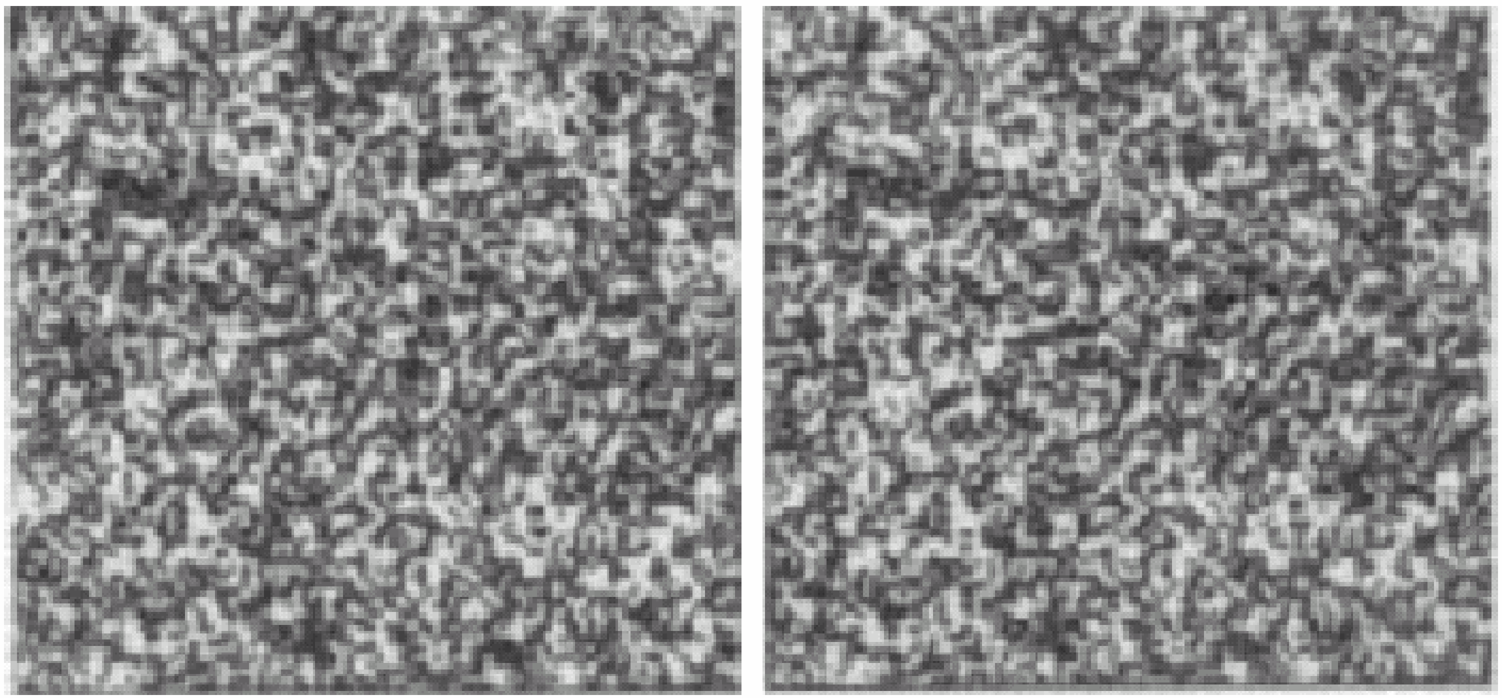


In principle any pixel can match to any pixel

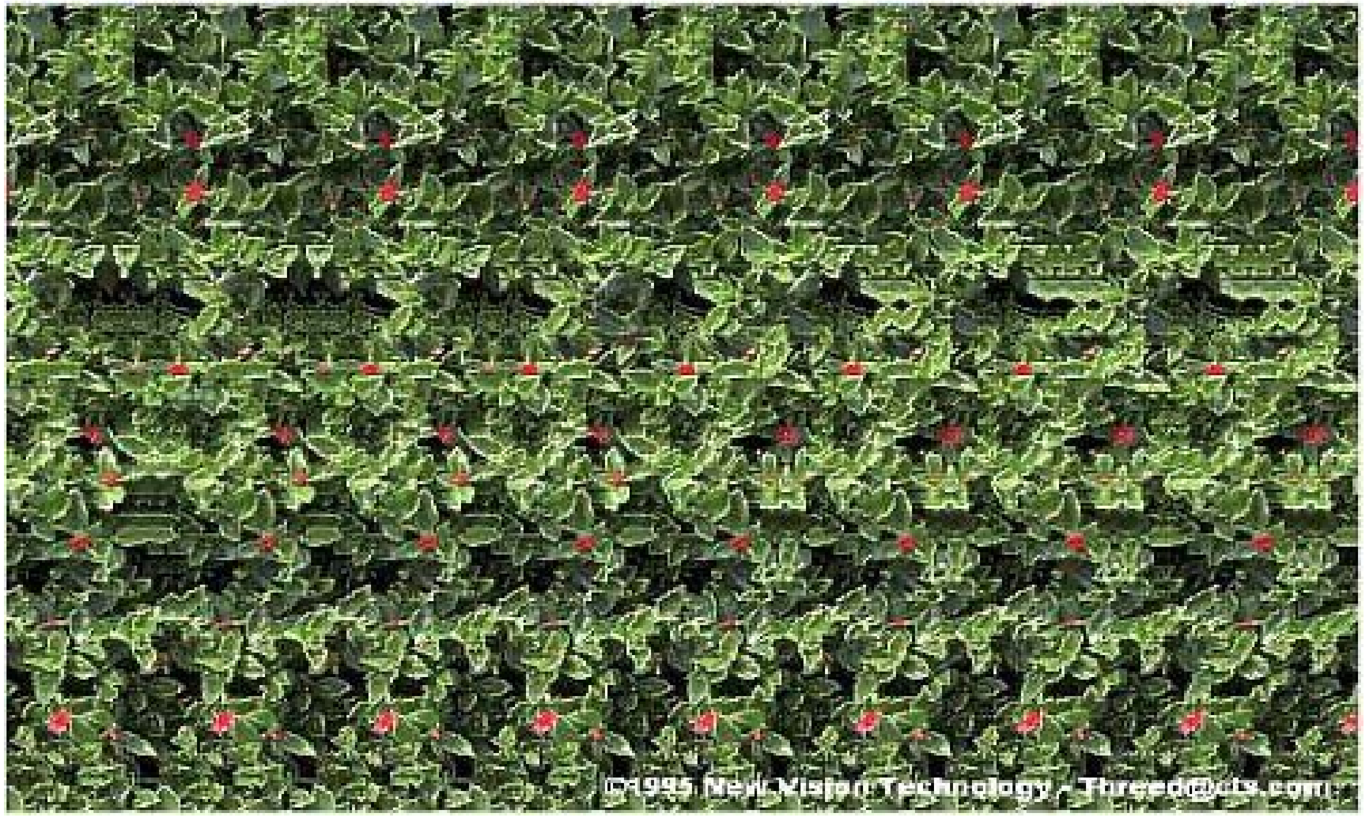
Finding correspondences is hard

Match “promising” features

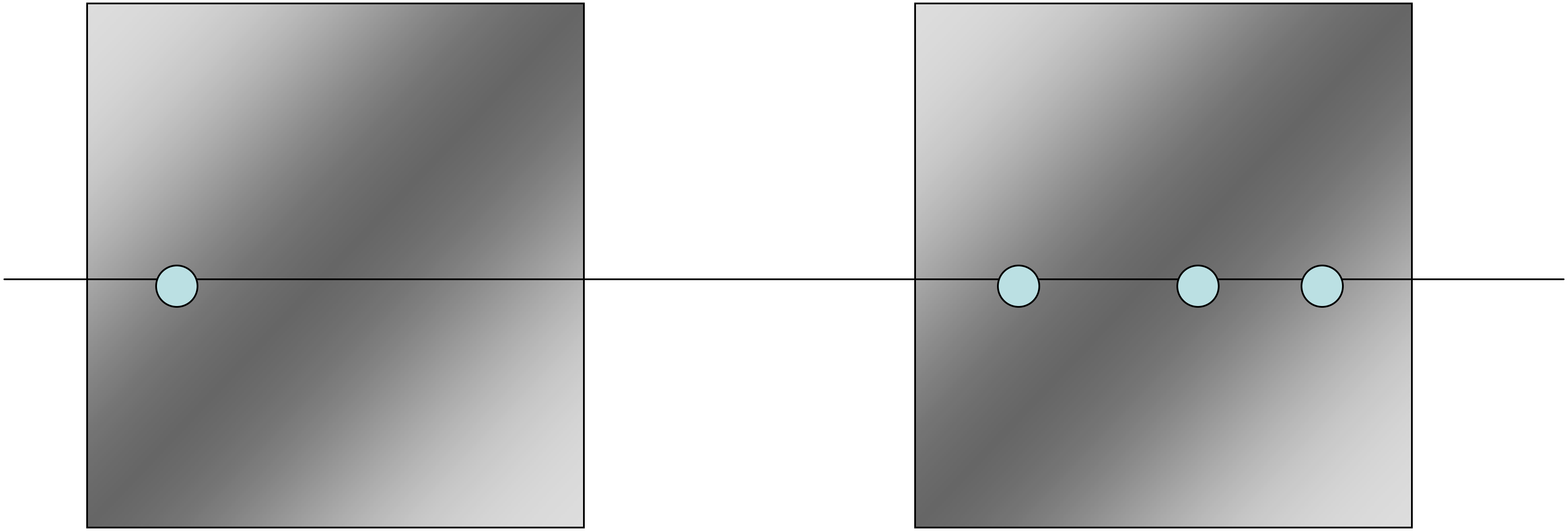
Random dot stereograms



Julesz: had huge impact because it showed that recognition not needed for stereo.


©1995 New Vision Technology - Threed@cts.com

Correspondence: Epipolar constraint.



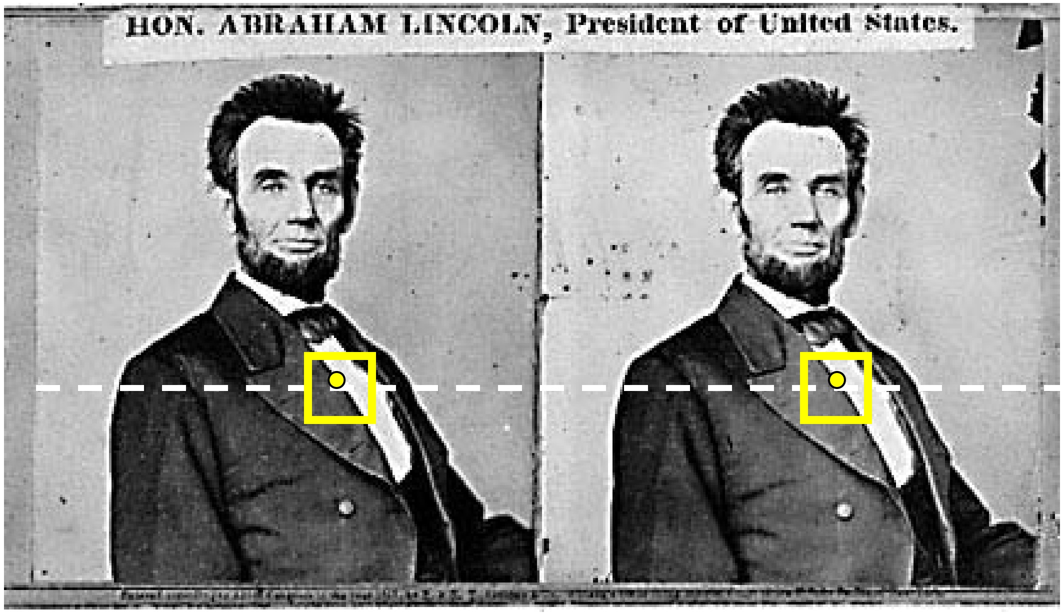
Correspondence Problem

- Two classes of algorithms:
  - Correlation-based algorithms
    - Produce a DENSE set of correspondences
  - Feature-based algorithms
    - Produce a SPARSE set of correspondences

Correspondence: Photometric constraint

- Same world point has same intensity in both images.
  - Lambertian fronto-parallel
  - Issues:
    - Noise
    - Specularity
    - Foreshortening

Using these constraints we can use matching for stereo



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
- This will never work, so:

Improvement: match ***windows***

Comparing Windows:

?
=

f g

$$SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2$$

$$C_{fg} = \sum_{[i,j] \in R} f(i,j) g(i,j)$$

Most popular

For each window, match to closest window on epipolar line in other image.

Minimize $$\sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$ Sum of Squared Differences

Maximize $$C_{fg} = \sum_{[i,j]\in R} f(i,j)g(i,j)$$ Cross correlation

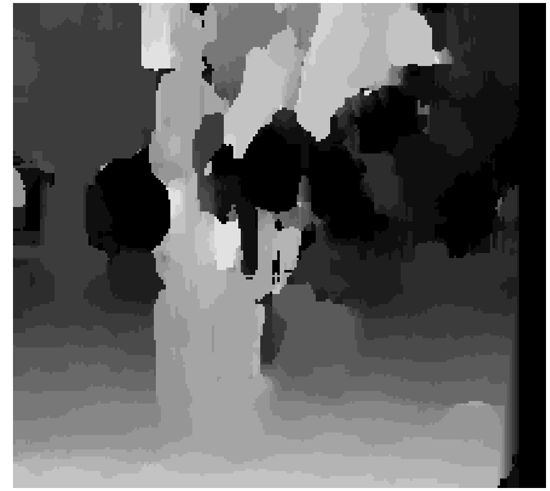
It is closely related to the SSD:

$$SSD = \sum_{[i,j]\in R} (f-g)^2 =$$

$$= \sum_{[i,j]\in R} f^2 + \sum_{[i,j]\in R} g^2 - 2 \sum_{[i,j]\in R} fg$$

Window size

W = 3



W = 20

- Effect of window size

Better results with *adaptive window*

- T. Kanade and M. Okutomi, *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*,, Proc. International Conference on Robotics and Automation, 1991.
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

(Seitz)

Other constraints

- Smoothness: disparity usually doesn't change too quickly.
  - Unfortunately, this makes the problem 2D again.
  - Solved with a host of graph algorithms, Markov Random Fields, Belief Propagation, ….
- Uniqueness constraint (each feature can at most have one match
- Occlusion and disparity are connected.

Feature-based Methods

- Conceptually very similar to Correlation-based methods, but:
  - They only search for correspondences of a sparse set of image features.
  - Correspondences are given by the most similar feature pairs.
  - Similarity measure must be adapted to the type of feature used.

Feature-based Methods:

- Features most commonly used:
  - Corners
    - Similarity measured in terms of:
      - surrounding gray values (SSD, Cross-correlation)
      - location
  - Edges, Lines
    - Similarity measured in terms of:
      - orientation
      - contrast
      - coordinates of edge or line’s midpoint
      - length of line

Example: Comparing lines

- l_l and l_r: line lengths
- θ_l and θ_r: line orientations
- (x_l,y_l) and (x_r,y_r): midpoints
- c_l and c_r: average contrast along lines
- ω_l ω_θ ω_m ω_c : weights controlling influence

$$S = \frac{1}{\omega_l(l_l - l_r)^2 + \omega_\theta(\theta_l - \theta_r)^2 + \omega_m[(x_l - x_r)^2 + (y_l - y_r)^2] + \omega_c(c_l - c_r)^2}$$

The more similar the lines, the larger S is!

Summary

- First, we understand constraints that make the problem solvable.
  - Some are hard, like epipolar constraint.
    - Ordering isn’t a hard constraint, but most useful when treated like one.
  - Some are soft, like pixel intensities are similar, disparities usually change slowly.
- Then we find optimization method.
  - Which ones we can use depend on which constraints we pick.