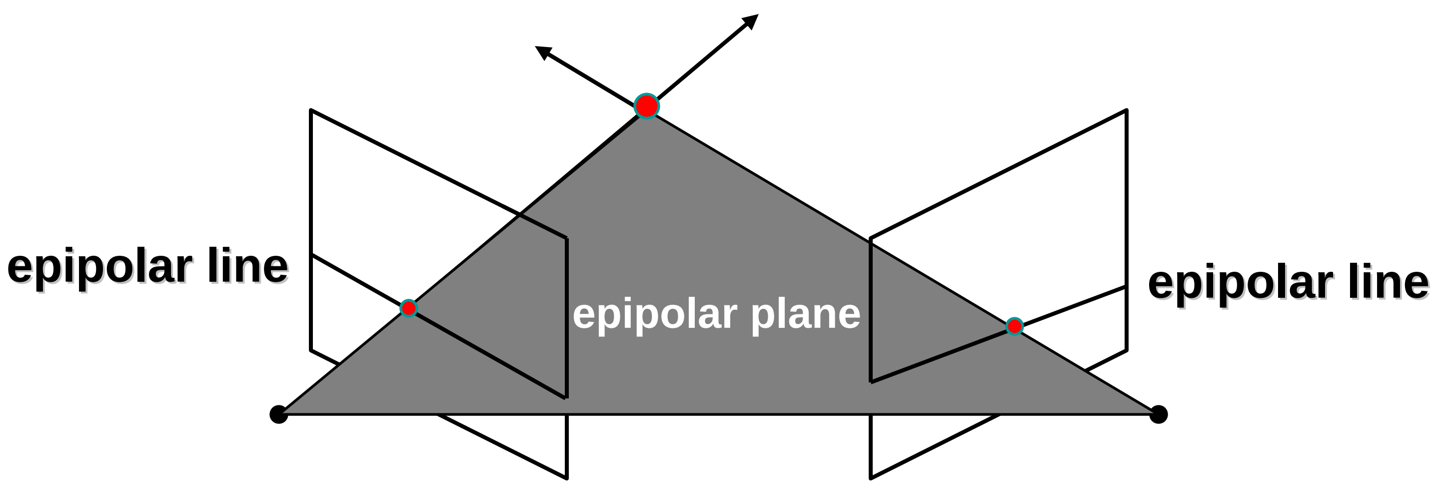
Stereopsis-II

Stereo correspondence

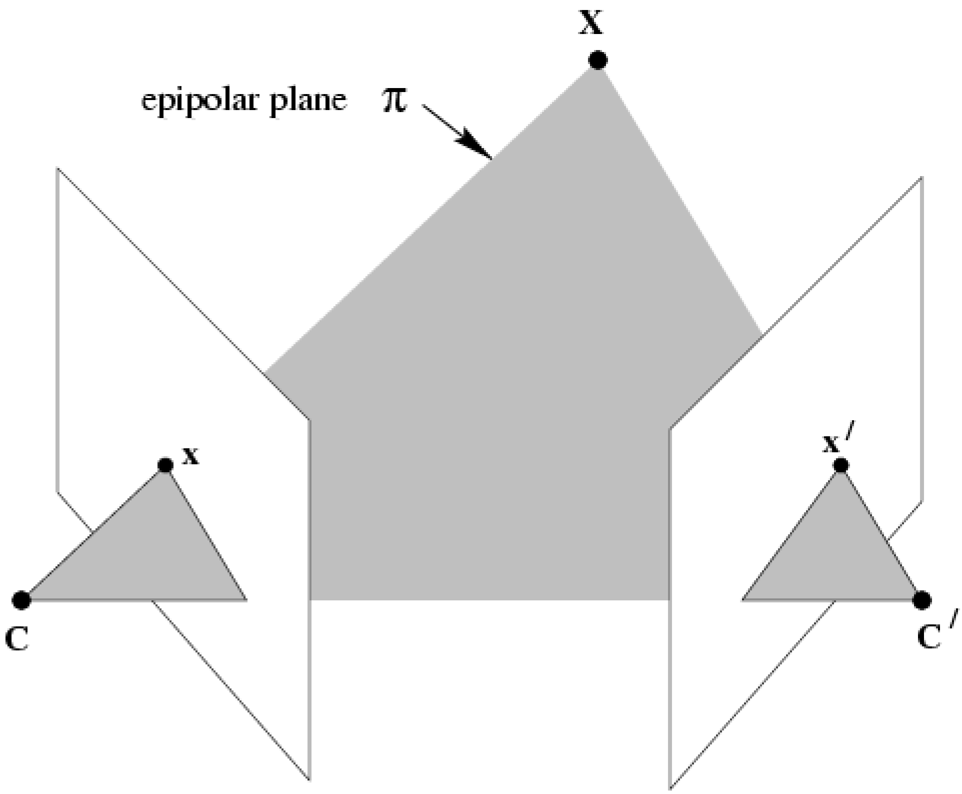
- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point



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

(Seitz)

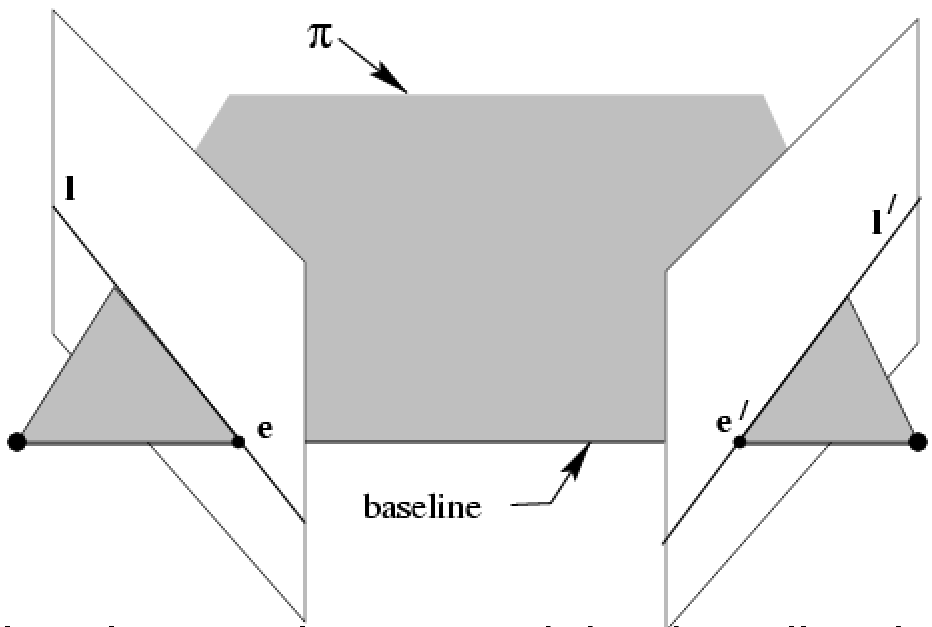
The epipolar geometry



C,C',x,x' and X are coplanar

The epipolar geometry

epipoles e,e'
= intersection of baseline with image plane
= projection of projection center in other image



an epipolar plane = plane containing baseline (1-D family)

an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

Epipolar constraint

- Given an image point its image in the second image lies on the epipolar line

From geometry to algebra…

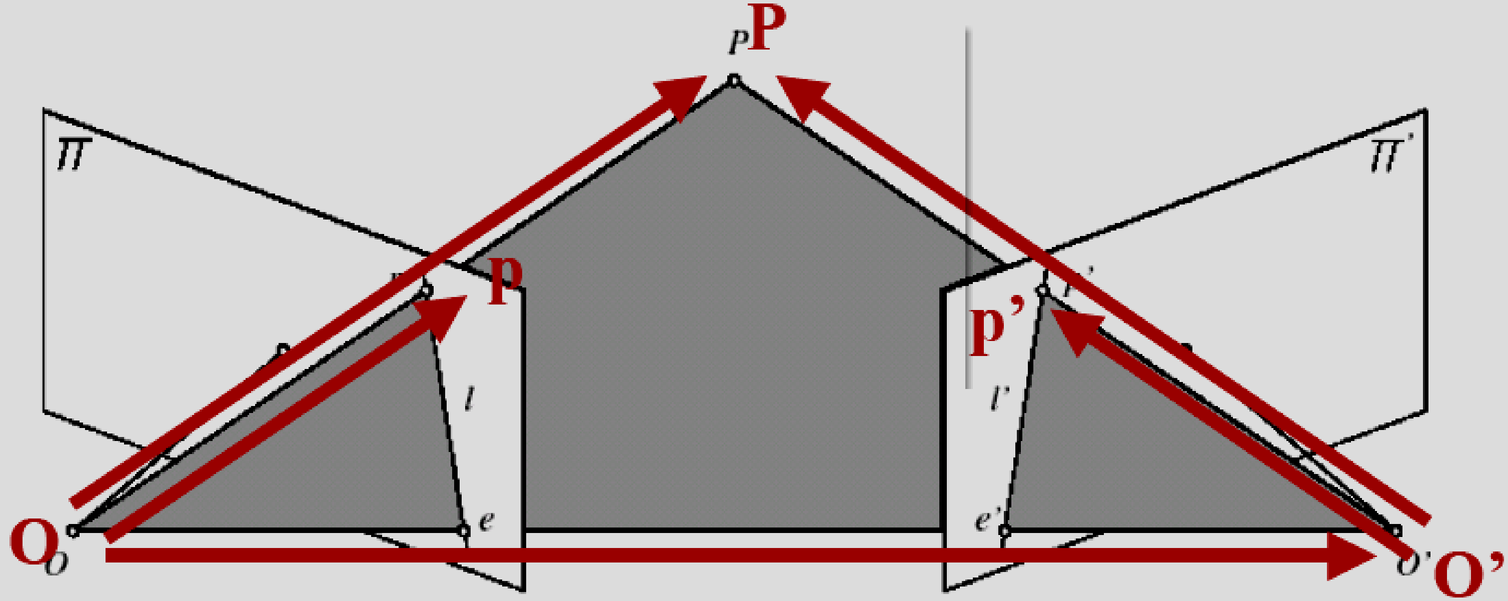


FIGURE 11.1: Epipolar geometry: the point P, the optical centers O and O' of the two cameras, and the two images p and p' of P all lie in the same plane.

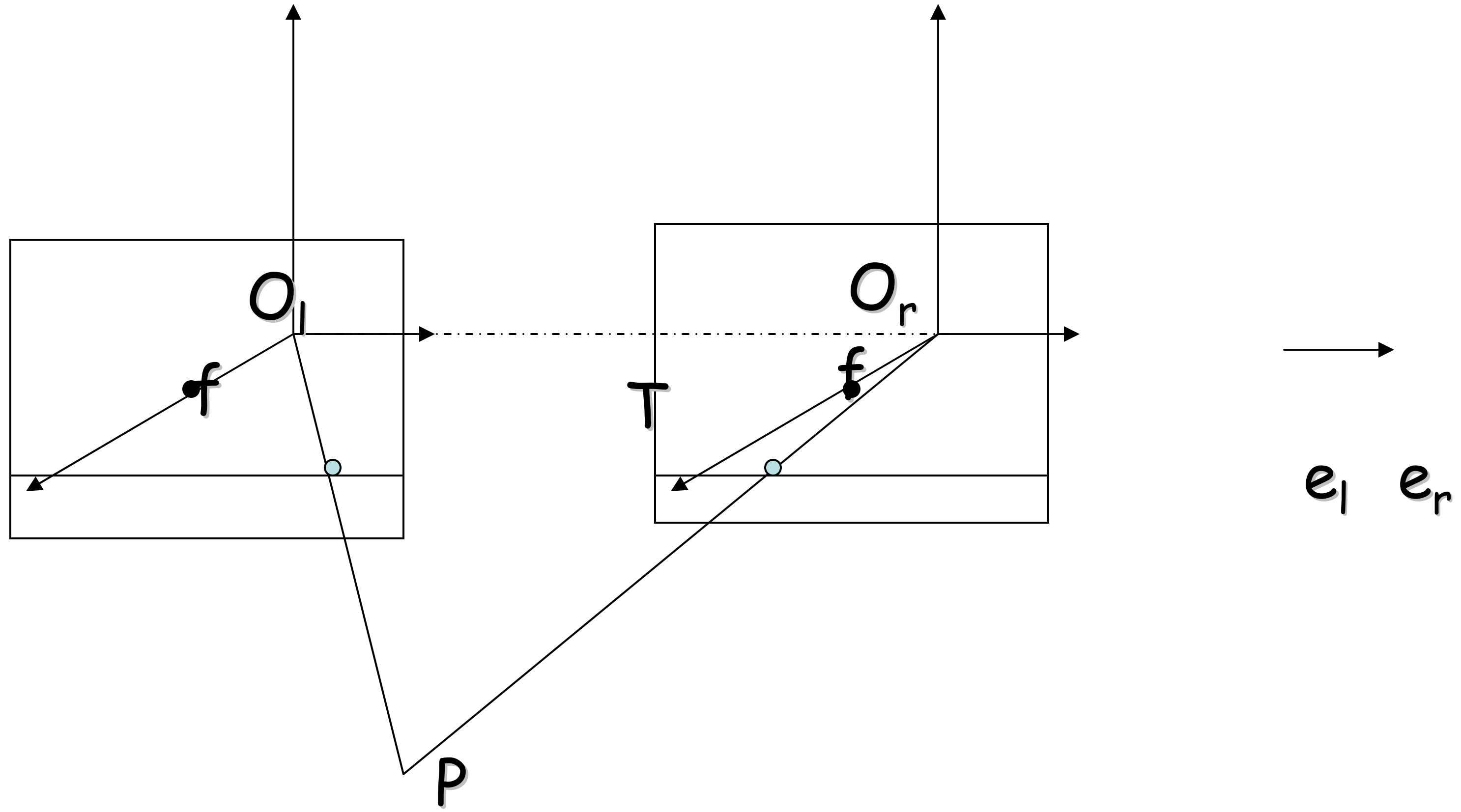
The epipolar constraint: these vectors are coplanar:

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$

Simplest Case

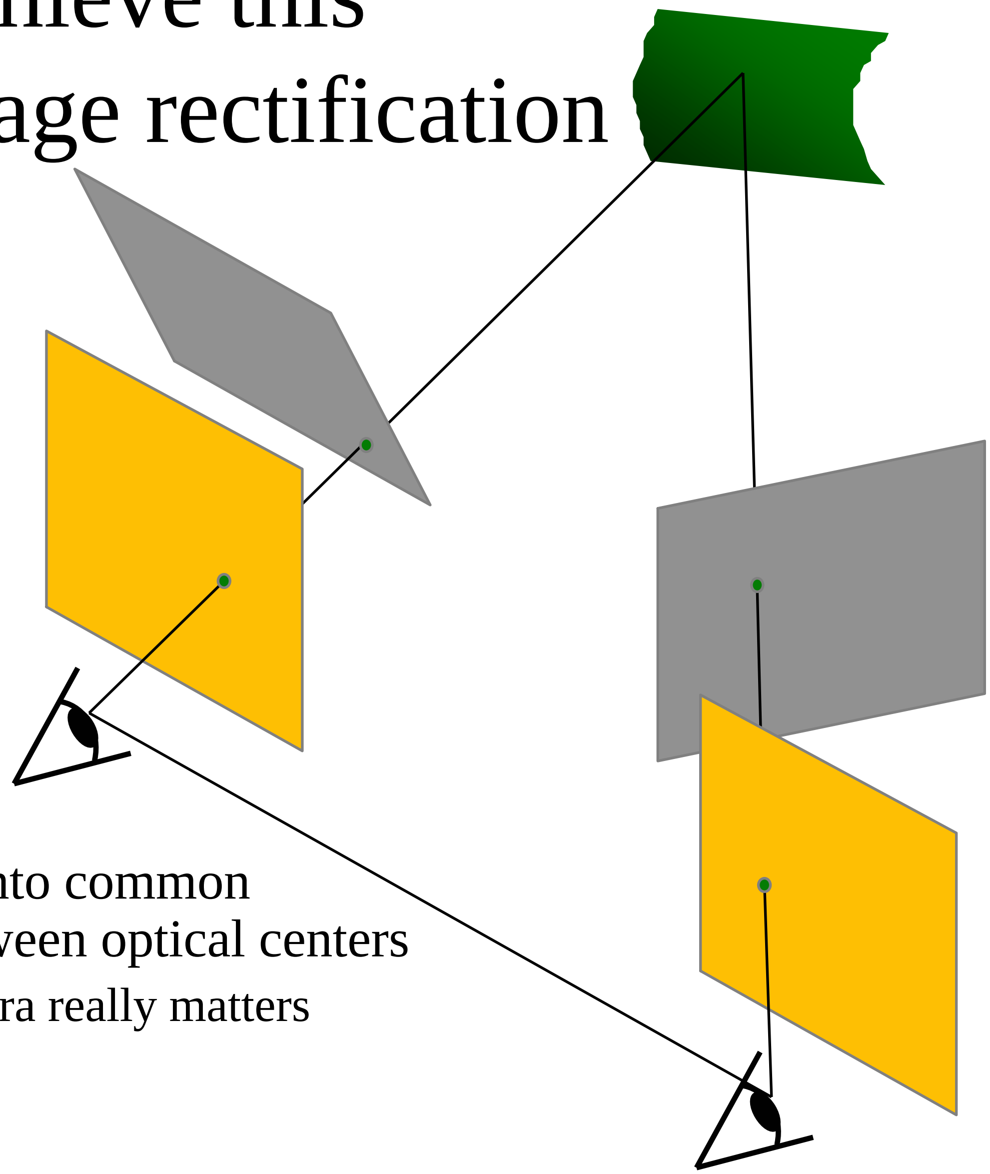
- Image planes of cameras are parallel.
- Focal points are at same height.
- Focal lengths same.
- Then, epipolar lines are horizontal scan lines.

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



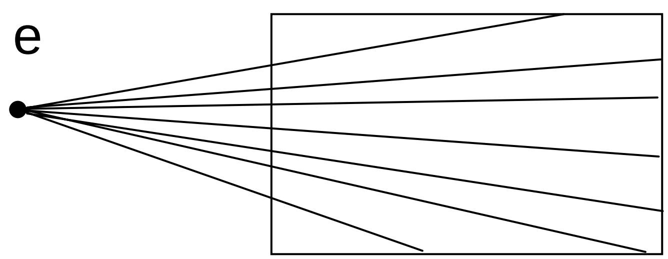
- Image Reprojection
  – reproject image planes onto common plane parallel to line between optical centers
- Notice, only focal point of camera really matters

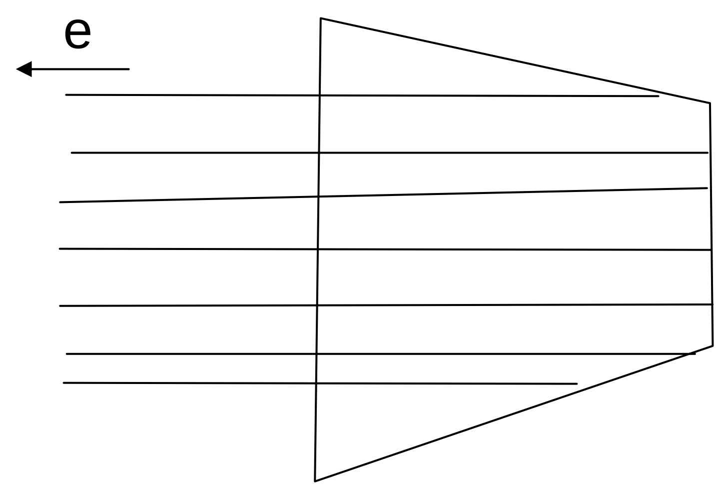
(Seitz)

Image pair rectification

simplify stereo matching
by warping the images

Apply projective transformation so that epipolar lines
correspond to horizontal scanlines





$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = He$$

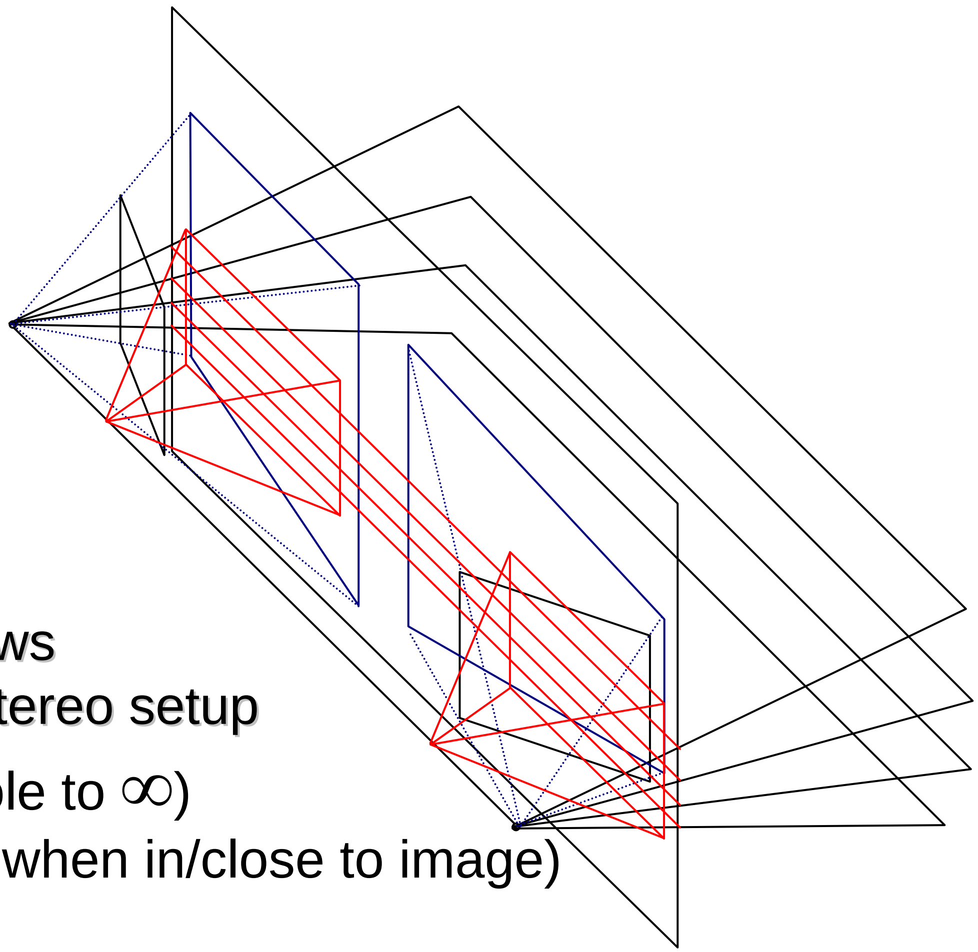
map epipole e to (1,0,0)

try to minimize image distortion

problem when epipole in (or close to) the image

Planar rectification

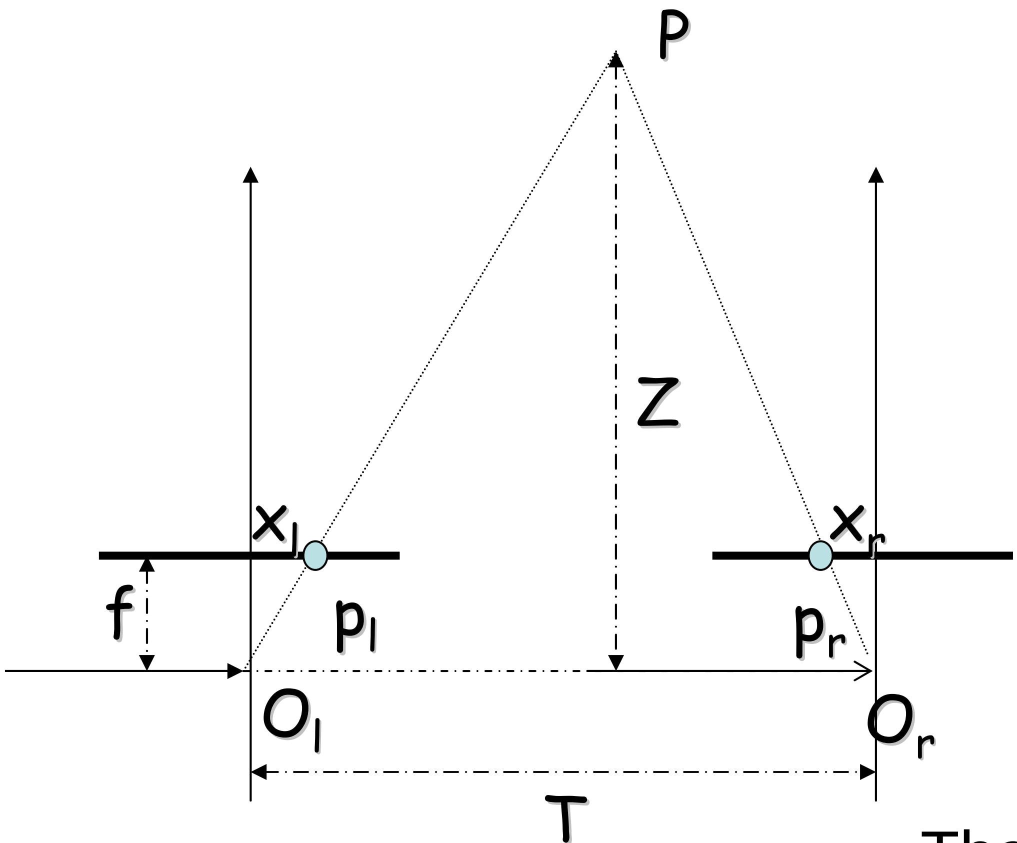
(standard approach)



Bring two views
to standard stereo setup

(moves epipole to ∞)
(not possible when in/close to image)

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: $d = x_l - x_r$

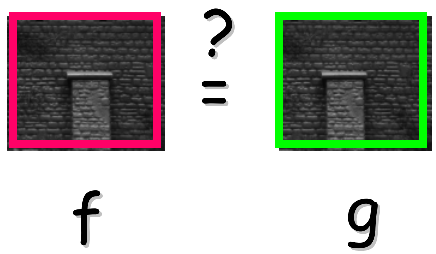
$$\boxed{Z = f\frac{T}{d}}$$

Then given Z, we can compute X and Y.

T is the stereo baseline

d measures the difference in retinal position between corresponding points

Comparing Windows:



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

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

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:

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

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

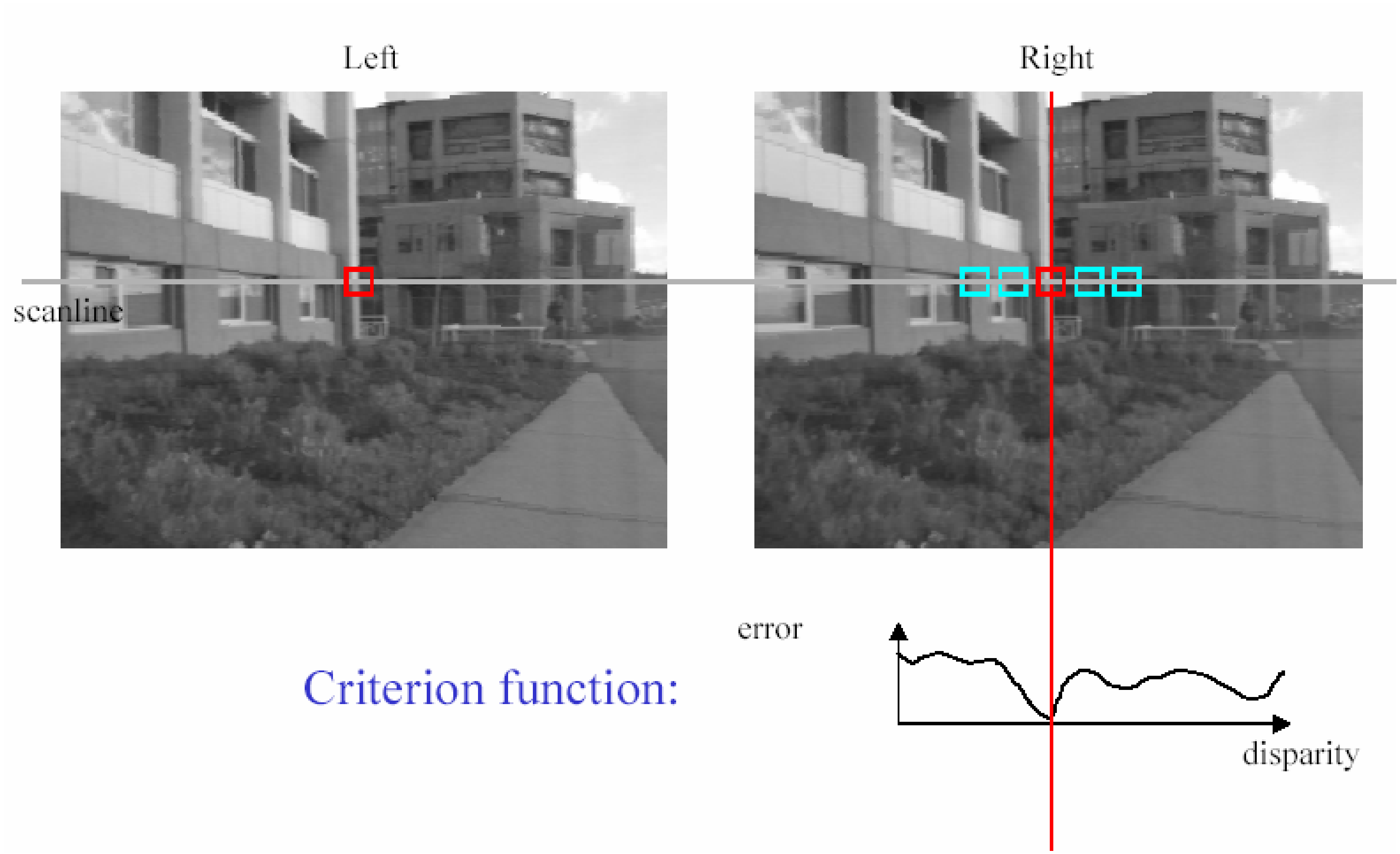
Other constraints

- 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.

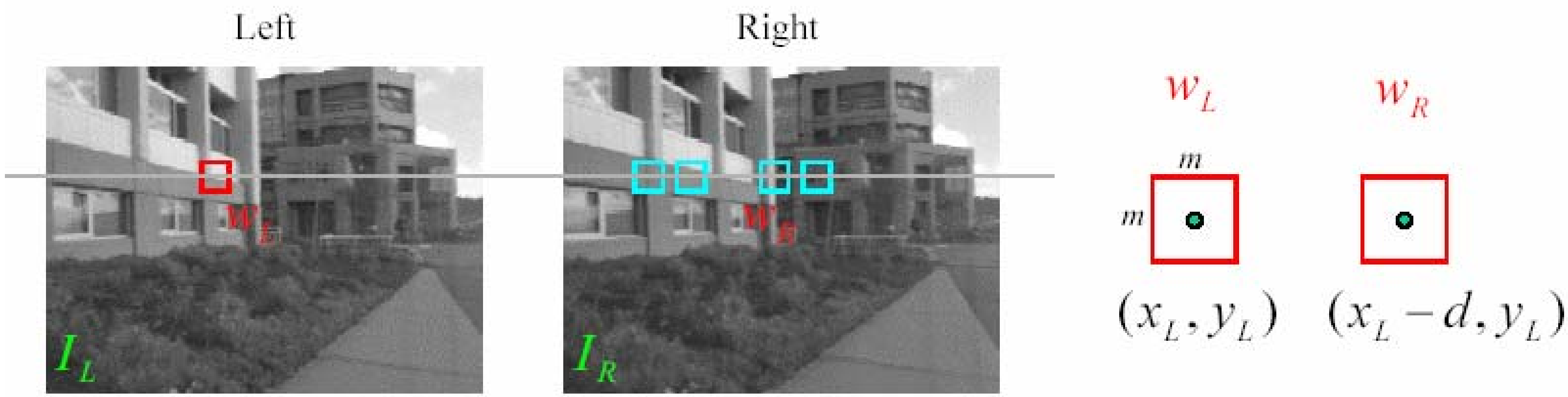
Stereo matching algorithms

- Match Pixels in Conjugate Epipolar Lines
  - Assume brightness constancy
  - This is a tough problem
  - Numerous approaches
    - dynamic programming [Baker 81,Ohta 85]
    - smoothness functionals
    - more images (trinocular, N-ocular) [Okutomi 93]
    - graph cuts [Boykov 00]
  - A good survey and evaluation: http://www.middlebury.edu/stereo/

Correspondence using Discrete Search



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 :

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

Image Normalization

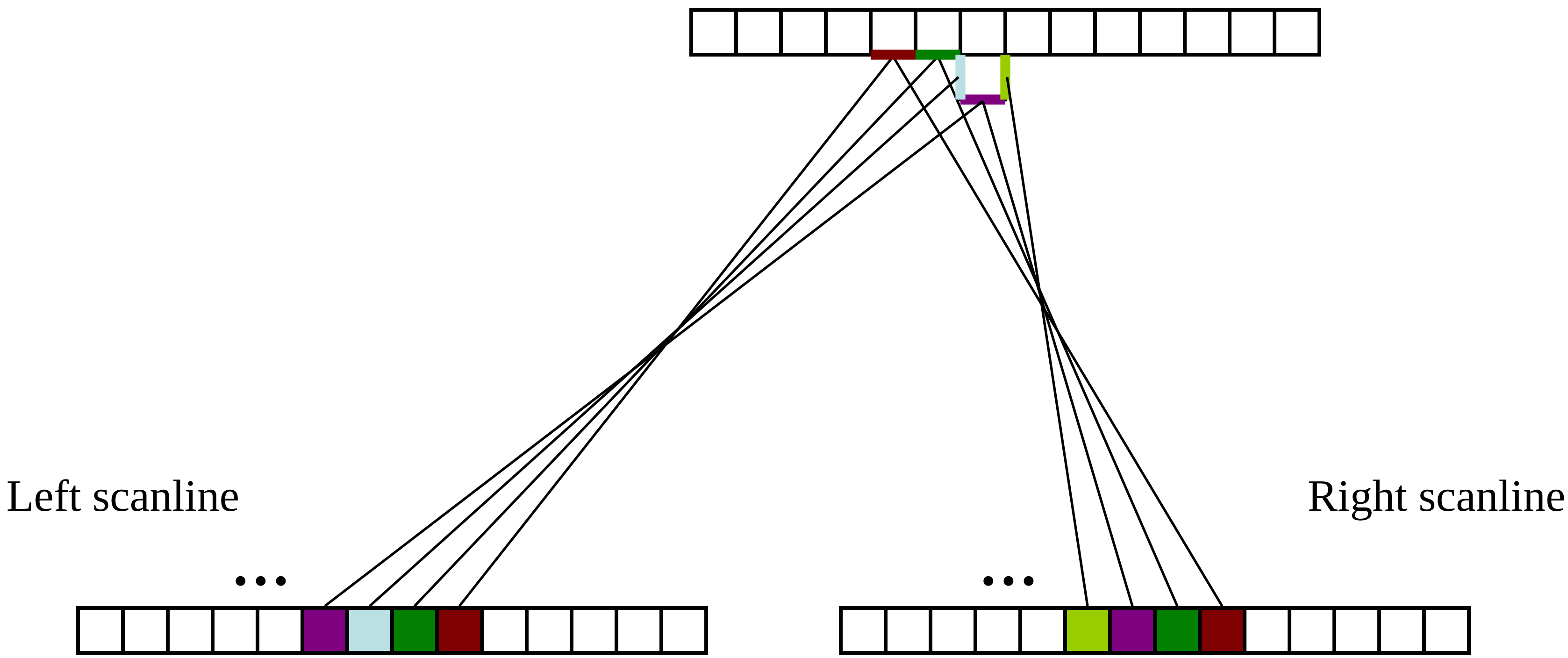
- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

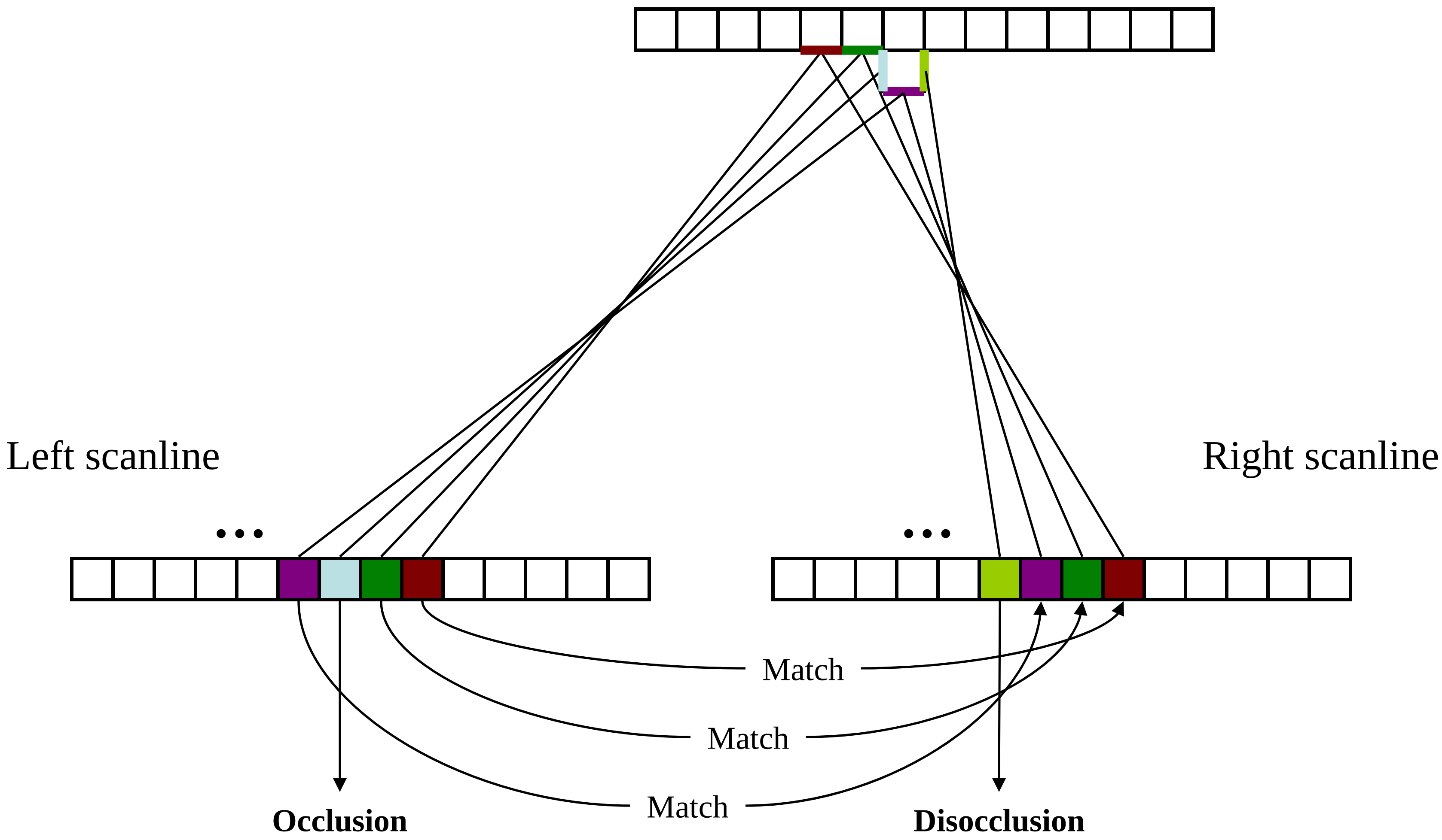
$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2}$$ Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}$$ Normalized pixel

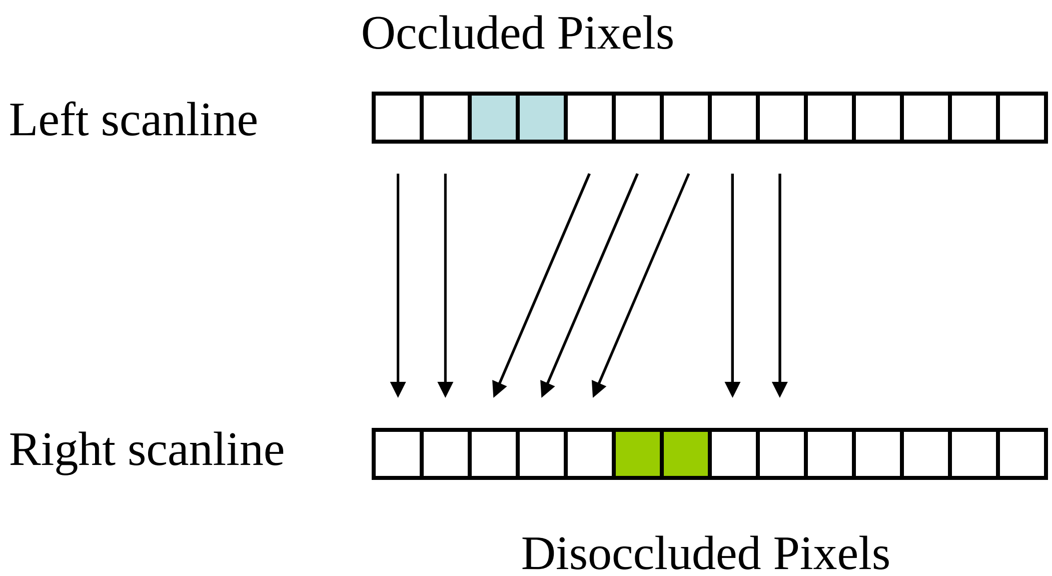
Stereo Correspondences



Stereo Correspondences



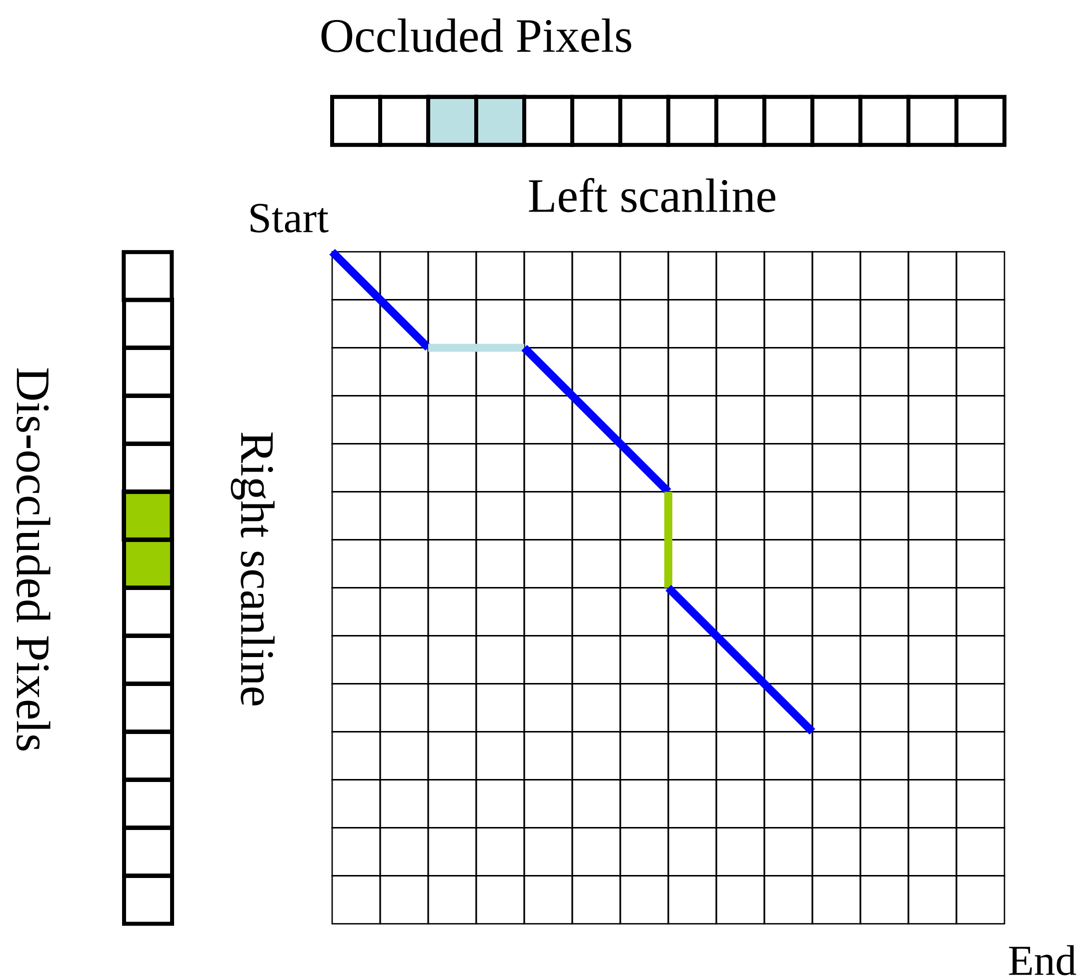
Search Over Correspondences



Three cases:

- Sequential – add cost of match (small if intensities agree)
- Occluded – add cost of no match (large cost)
- Disoccluded – add cost of no match (large cost)

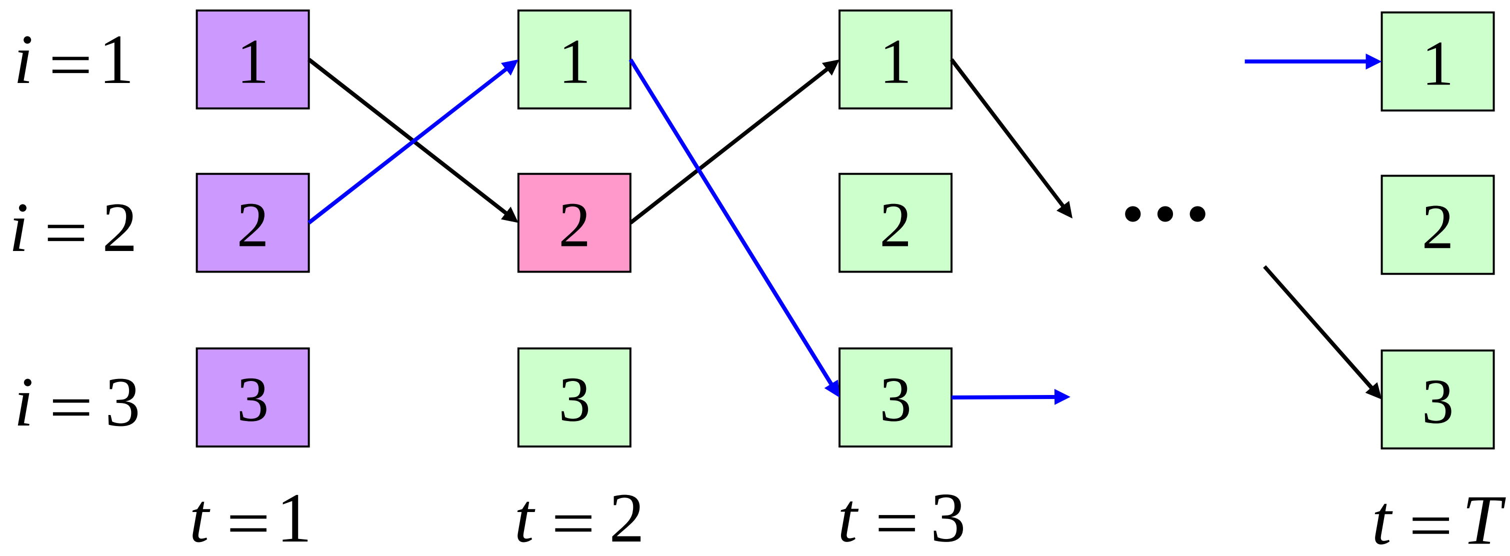
Stereo Matching with Dynamic Programming



Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint

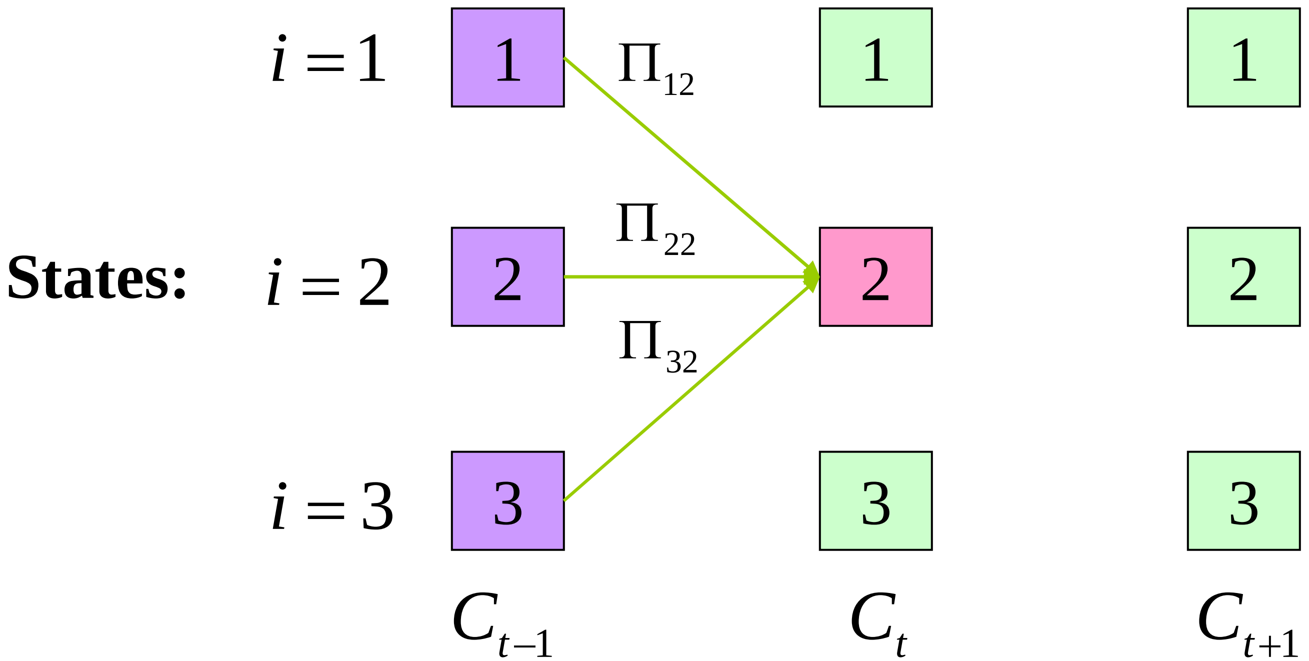
Dynamic Programming

- Efficient algorithm for solving sequential decision (optimal path) problems.



How many paths through this trellis? 3^T

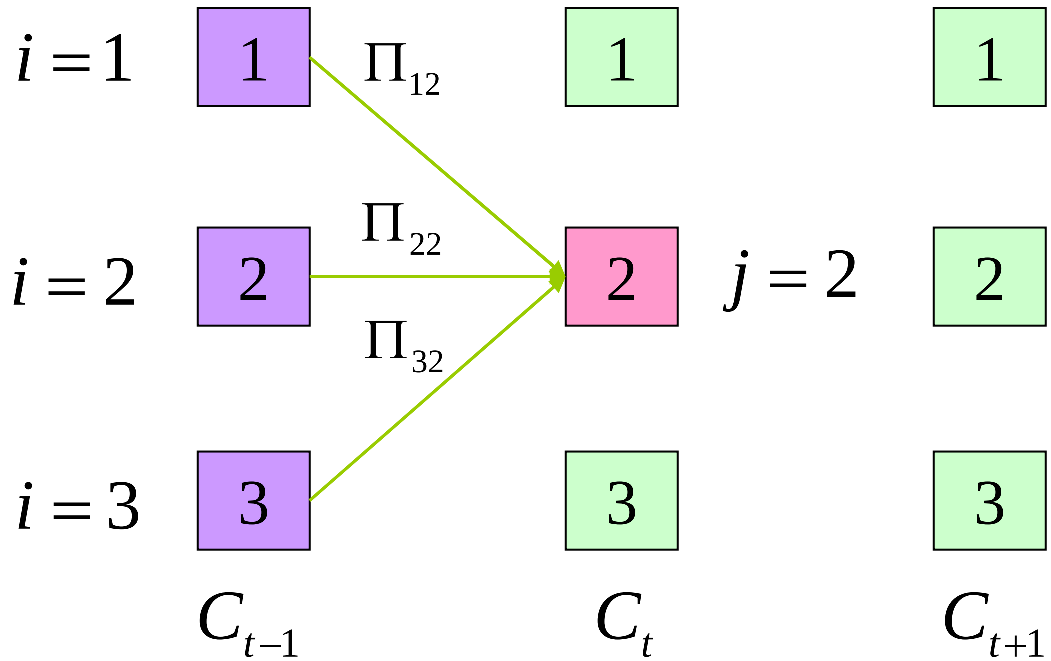
Dynamic Programming



Suppose cost can be decomposed into stages:

$$\Pi_{ij} = \text{Cost of going from state } i \text{ to state } j$$

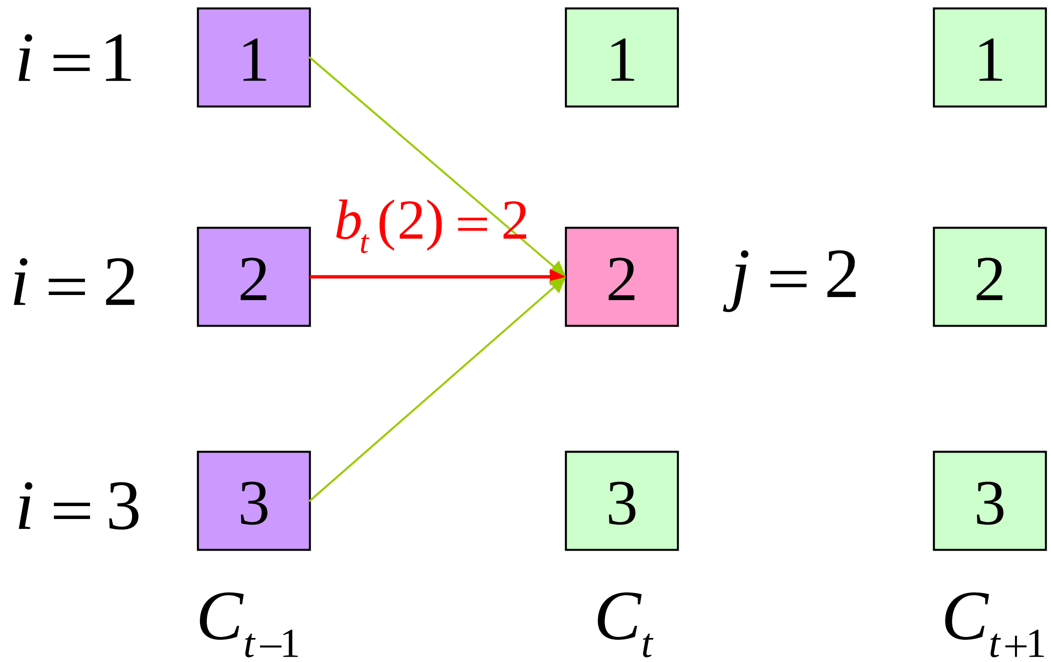
Dynamic Programming



Principle of Optimality for an n-stage assignment problem:

$$C_t(j) = \min_i(\Pi_{ij} + C_{t-1}(i))$$

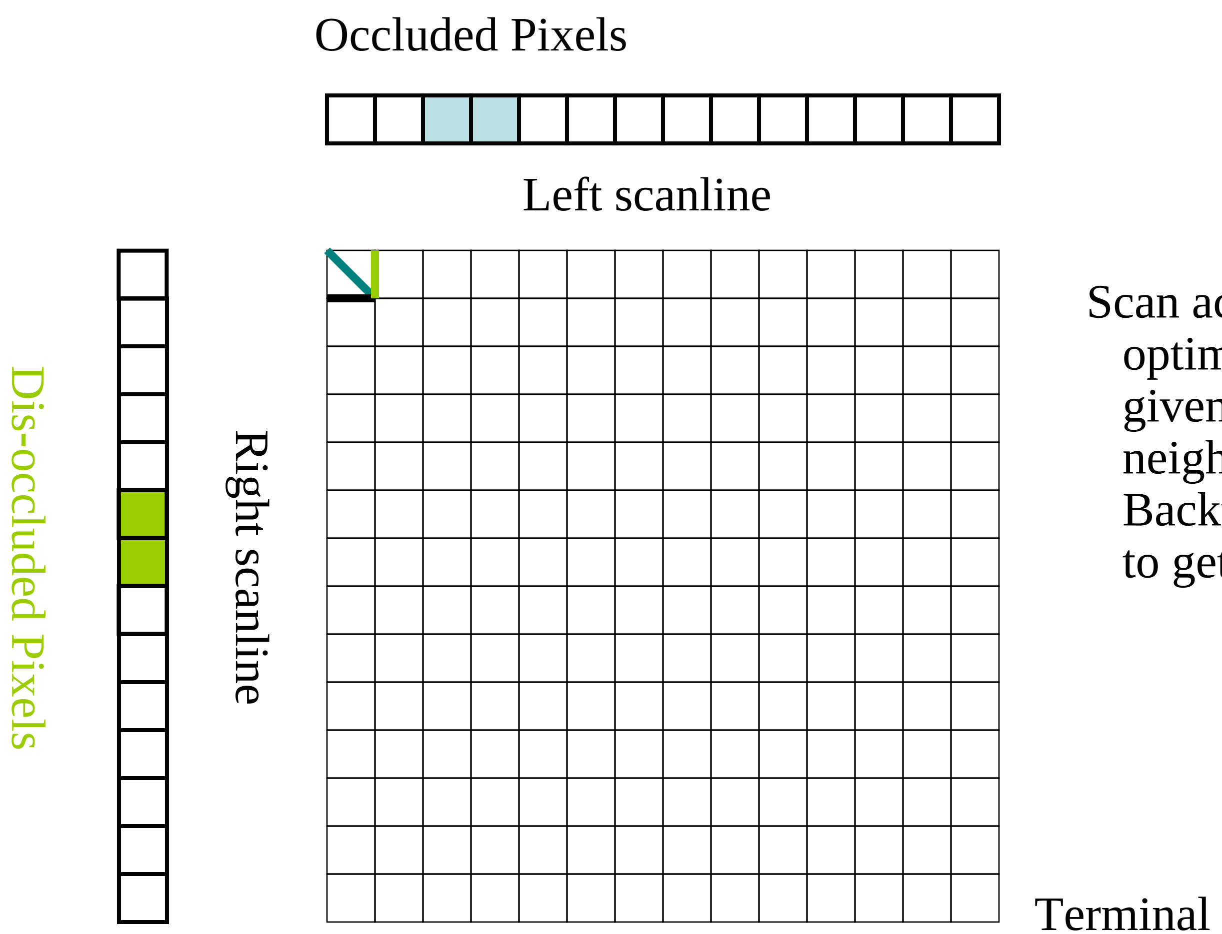
Dynamic Programming



$$C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i))$$

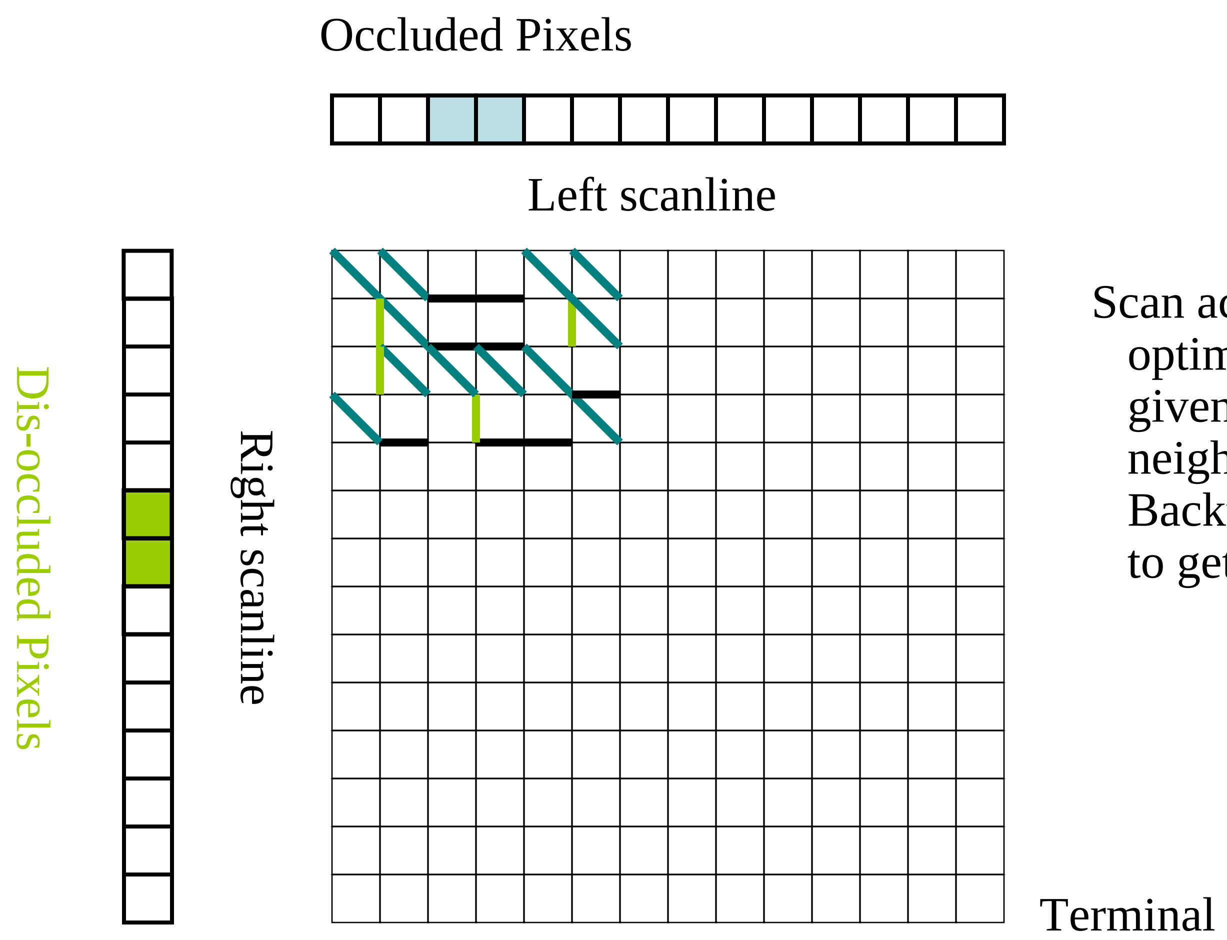
$$b_t(j) = \arg\min_i (\Pi_{ij} + C_{t-1}(i))$$

Stereo Matching with Dynamic Programming



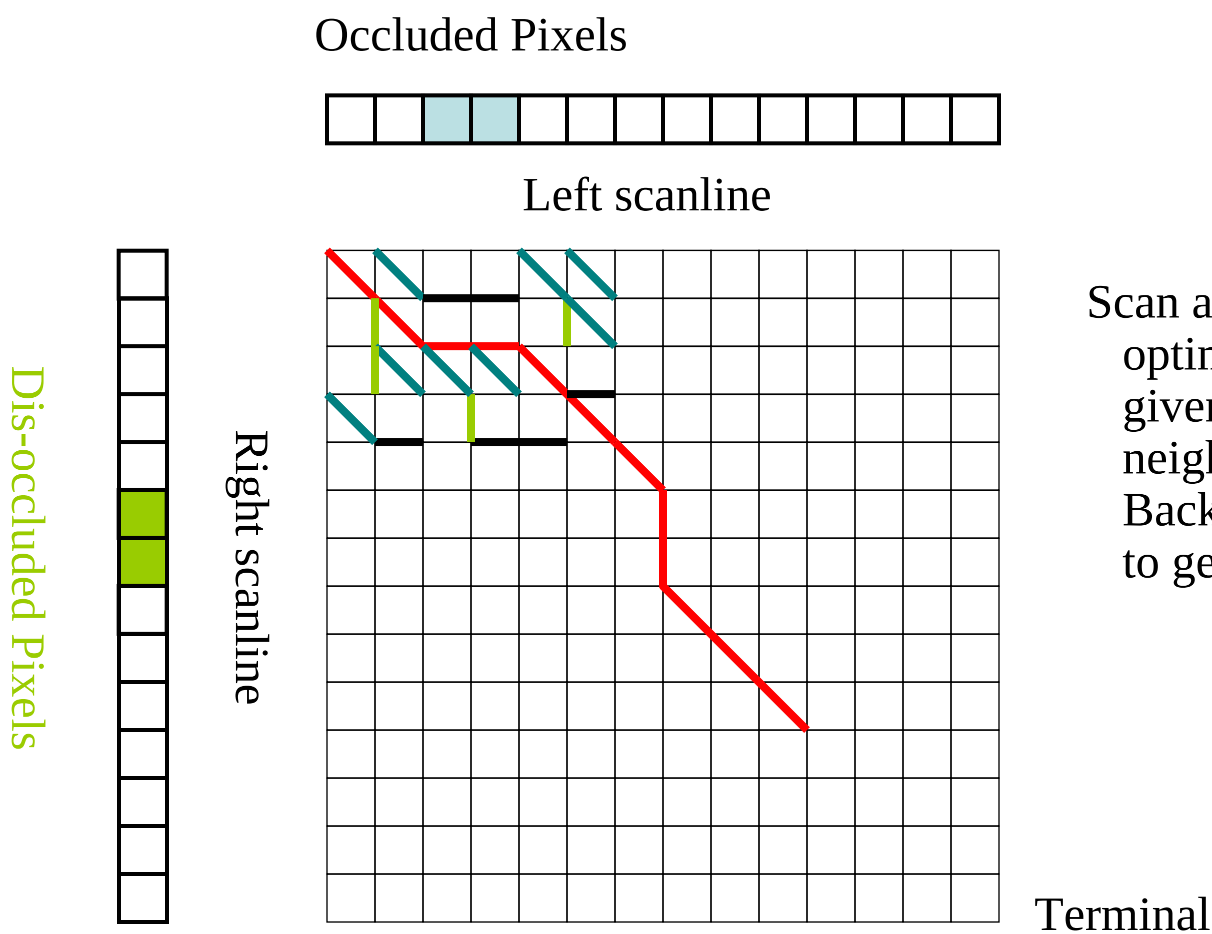
Scan across grid computing optimal cost for each node given its upper-left neighbors.
Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming



Scan across grid computing optimal cost for each node given its upper-left neighbors.
Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming



Scan across grid computing optimal cost for each node given its upper-left neighbors.
Backtrack from the terminal to get the optimal path.

Problems with window matching

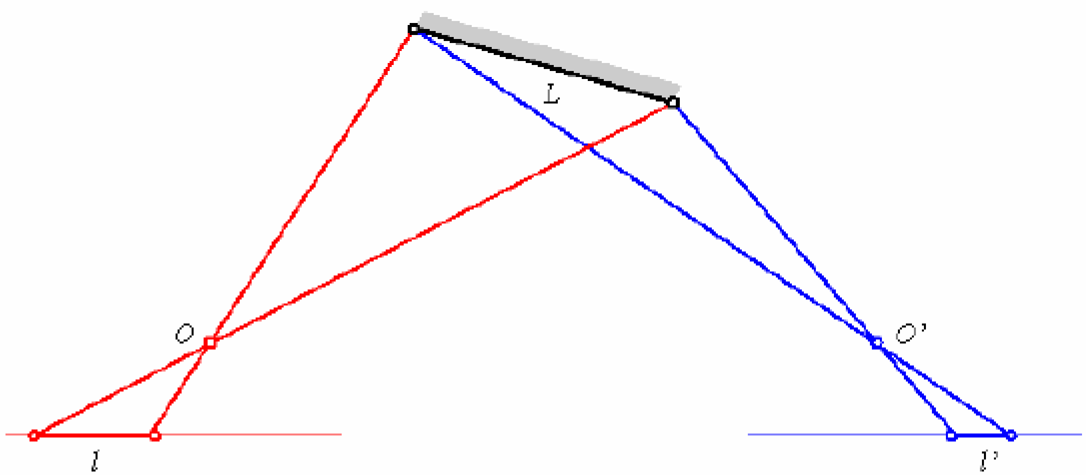
Patch too small?

Patch too large?

Can try variable patch size [Okutomi and Kanade], or arbitrary window shapes [Veksler and Zabih]

Foreshortening

Window methods assume fronto-parallel surface at 3-D point.



Initial estimates of the disparity can be used to warp the correlation windows to compensate for unequal amounts of foreshortening in the two pictures [Kass, 1987; Devernay and Faugeras, 1994].