Computing Simple Stereo

Background Reading:

- T&V Section 7.1
- Szeliski Chap 11

Recall: Simple Stereo System

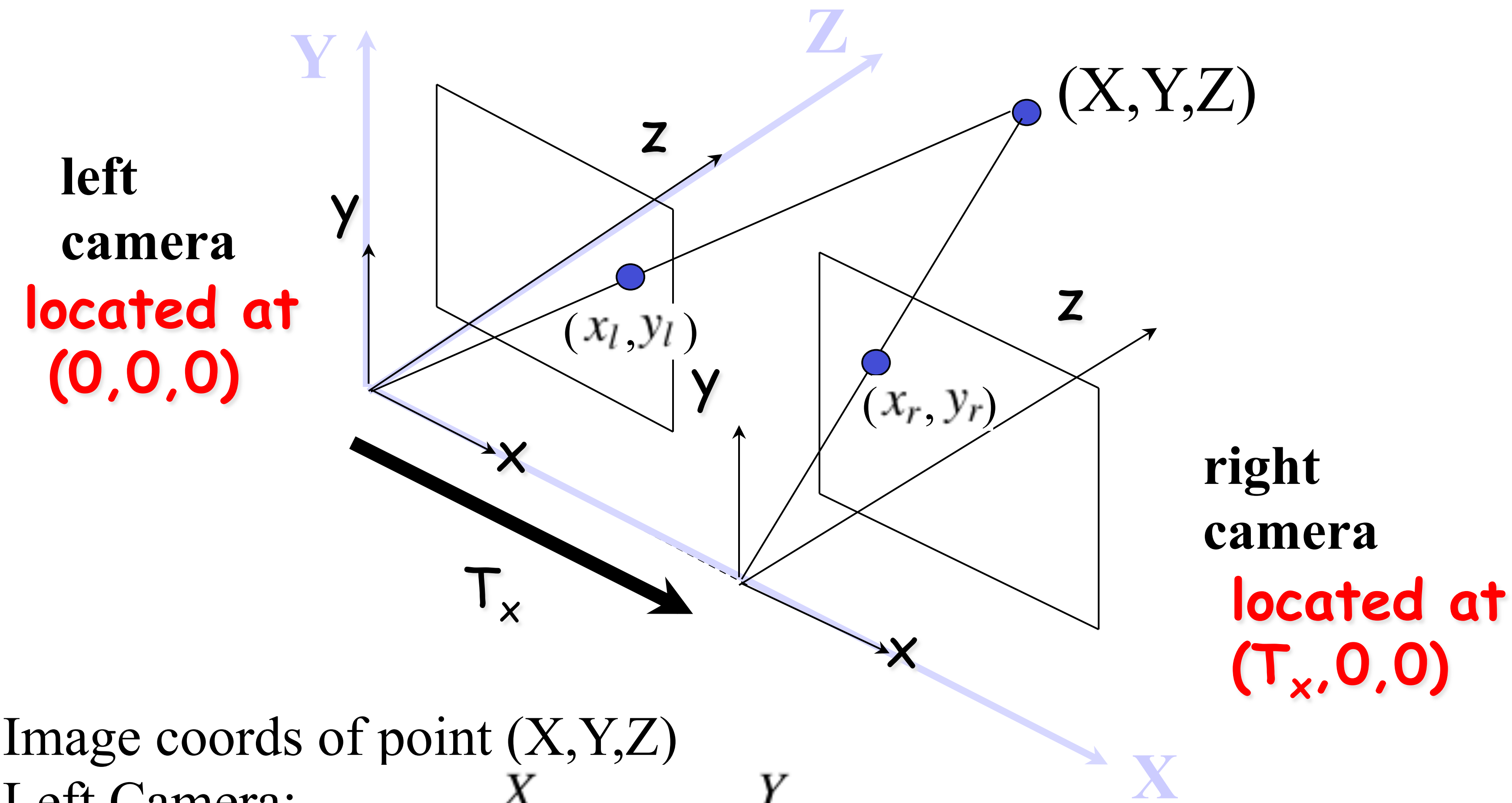


Image coords of point (X,Y,Z)

Left Camera: $x_l = f\,\frac{X}{Z} \quad y_l = f\,\frac{Y}{Z}$

Right Camera: $x_r = f\,\frac{X - T_x}{Z} \quad y_r = f\,\frac{Y}{Z}$

Robert Collins
CMPEN454

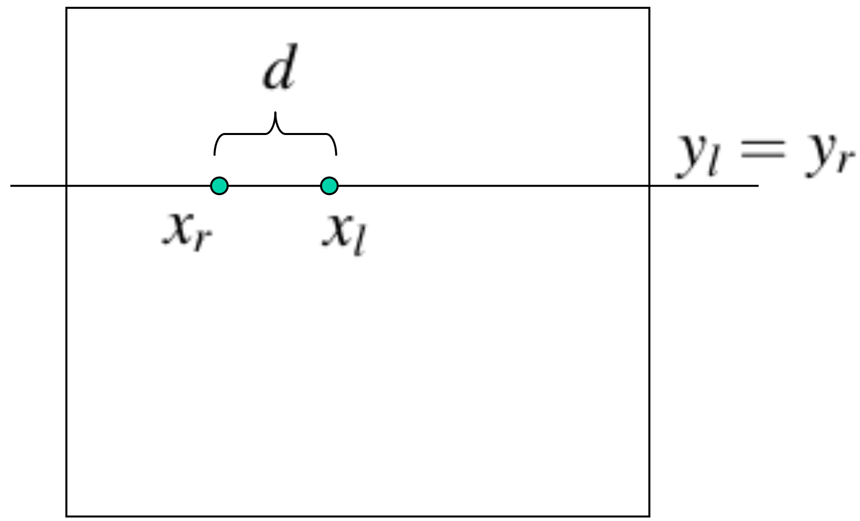
Recall: Stereo Disparity

Left camera

$$x_l = f\,\frac{X}{Z} \qquad y_l = f\,\frac{Y}{Z}$$

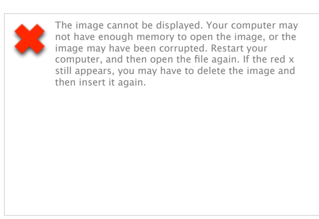
Right camera

$$x_r = f\,\frac{X - T_x}{Z} \qquad y_r = f\,\frac{Y}{Z}$$



Stereo Disparity

$$d = x_l - x_r = f\frac{X}{Z} - \left(f\frac{X}{Z} - f\frac{T_x}{Z}\right)$$



baseline

depth

$$Z = \frac{f\,T_x}{d}$$

disparity

Important equation!

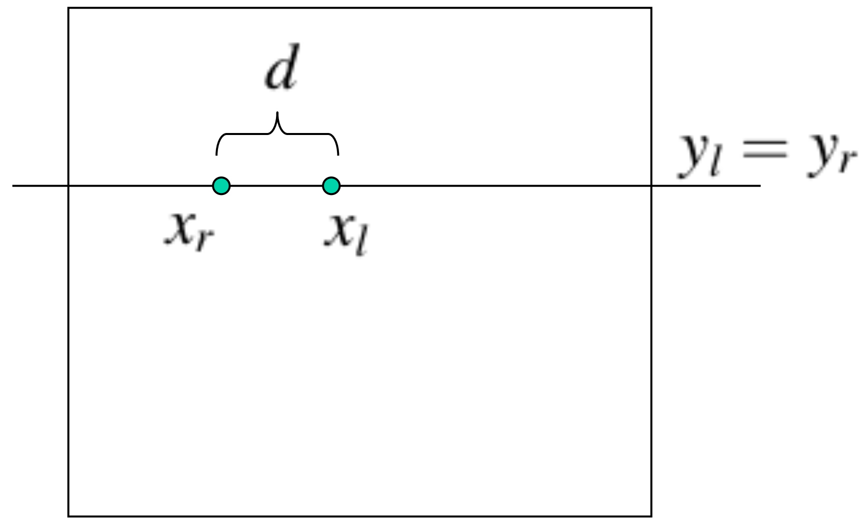
Recall: Stereo Disparity

Left camera

$$x_l = f\,\frac{X}{Z} \qquad y_l = f\,\frac{Y}{Z}$$

Right camera

$$x_r = f\,\frac{X - T_x}{Z} \qquad y_r = f\,\frac{Y}{Z}$$



Note: Depth and stereo disparity are inversely proportional

depth

$$Z = \frac{f\,T_x}{d}$$

disparity

Important equation!

Stereo Example

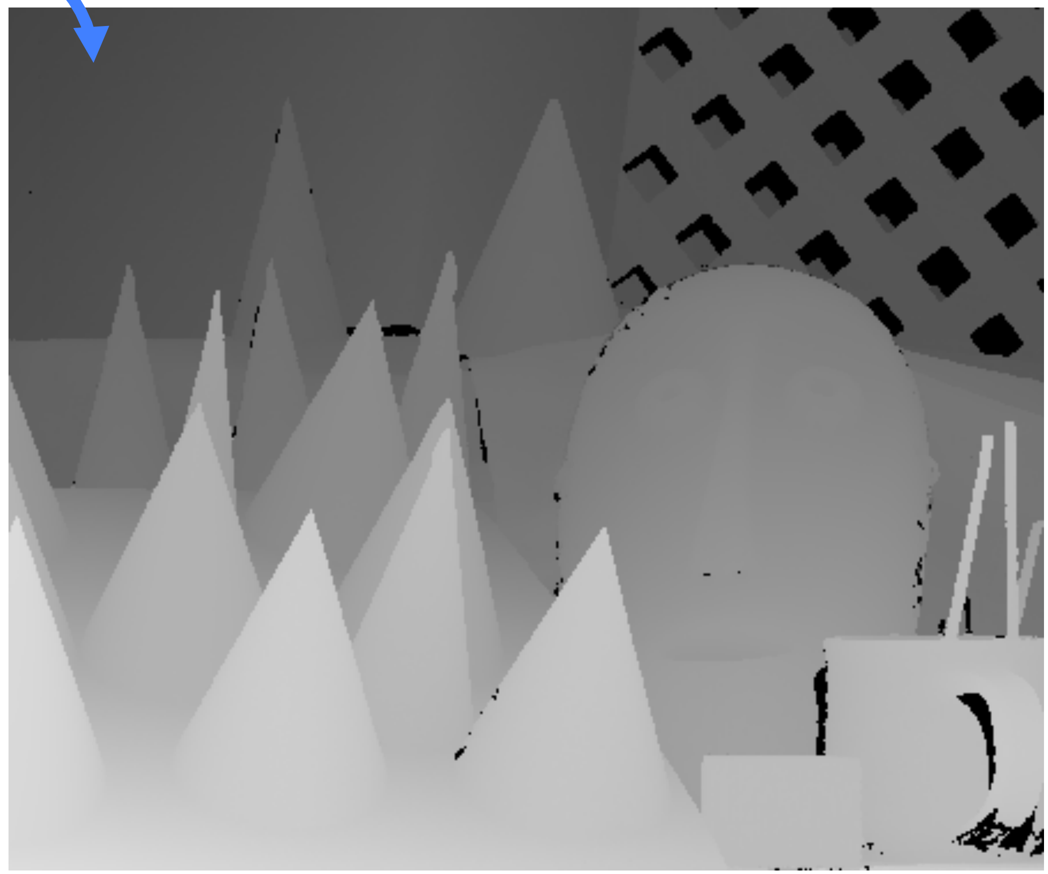
Left Image

Right Image

From Middlebury stereo evaluation page
http://www.middlebury.edu/stereo/

Stereo Example

Disparity values (0-64)



Note how disparity is larger (brighter) for closer surfaces.

Computing Disparity

- Correspondence Problem:
  - Determining which pixel in the right image corresponds to each pixel in the left image.
  - Disp = x_coord(left) - x_coord(right)

Recall our discussion of scores for measuring similarity/dissimilarity of image patches (Lecture 7).

Cfg - correlation of raw pixel values
SSD - sum of squared difference measure
NCC - normalized cross correlation measure

Epipolar Constraint

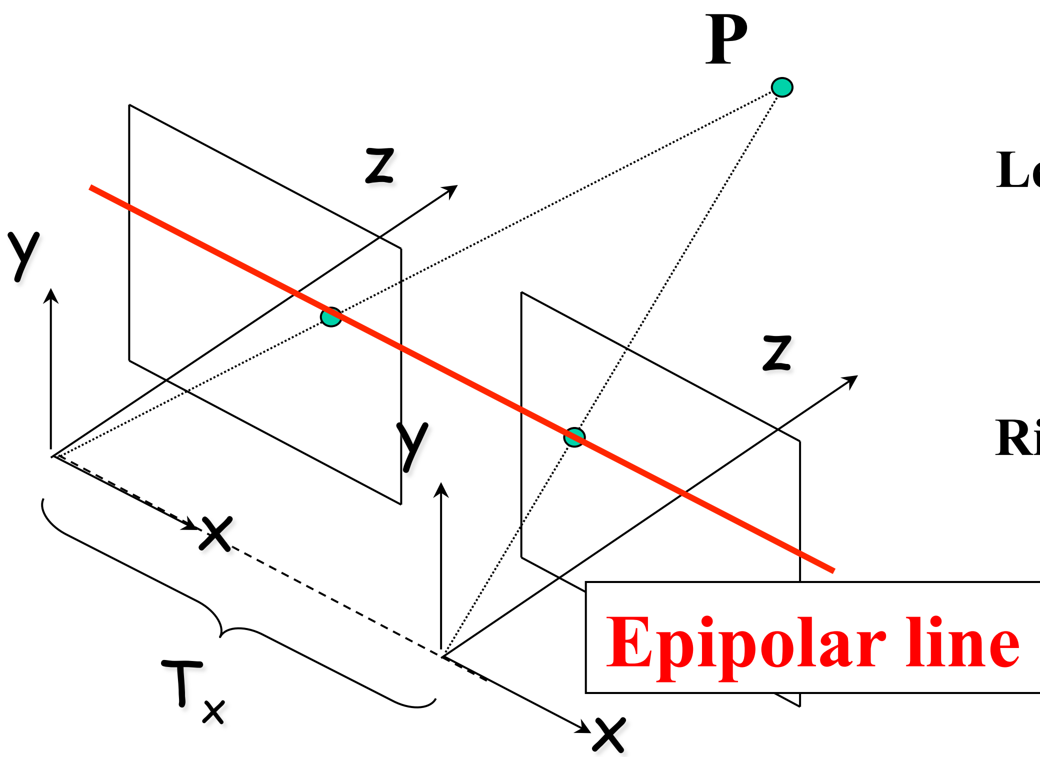
Important Concept:

For stereo matching, we don't have to search the whole 2D right image for a corresponding point.

The "epipolar constraint" reduces the search space to a one-dimensional line.

Robert Collins
CMPEN454

Recall : Simple Stereo System



Left camera

$$x_l = f\,\frac{X}{Z} \qquad \boxed{y_l = f\,\frac{Y}{Z}}$$

Right camera

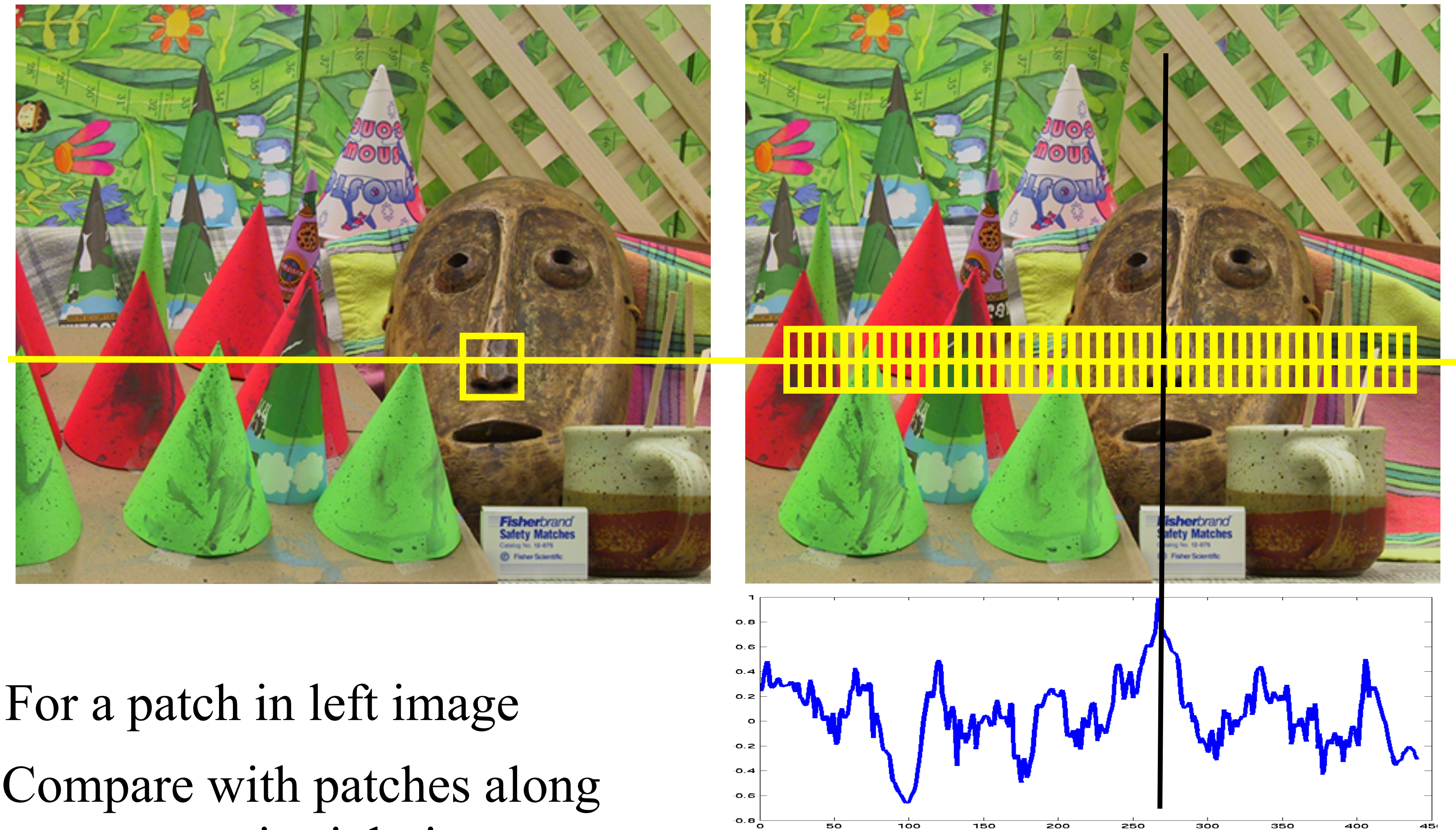
$$x_r = f\,\frac{X - T_x}{Z} \qquad \boxed{y_r = f\,\frac{Y}{Z}}$$

Same Y Coord!

Robert Collins
CMPEN454

Matching using Epipolar Lines

Left Image | Right Image



For a patch in left image

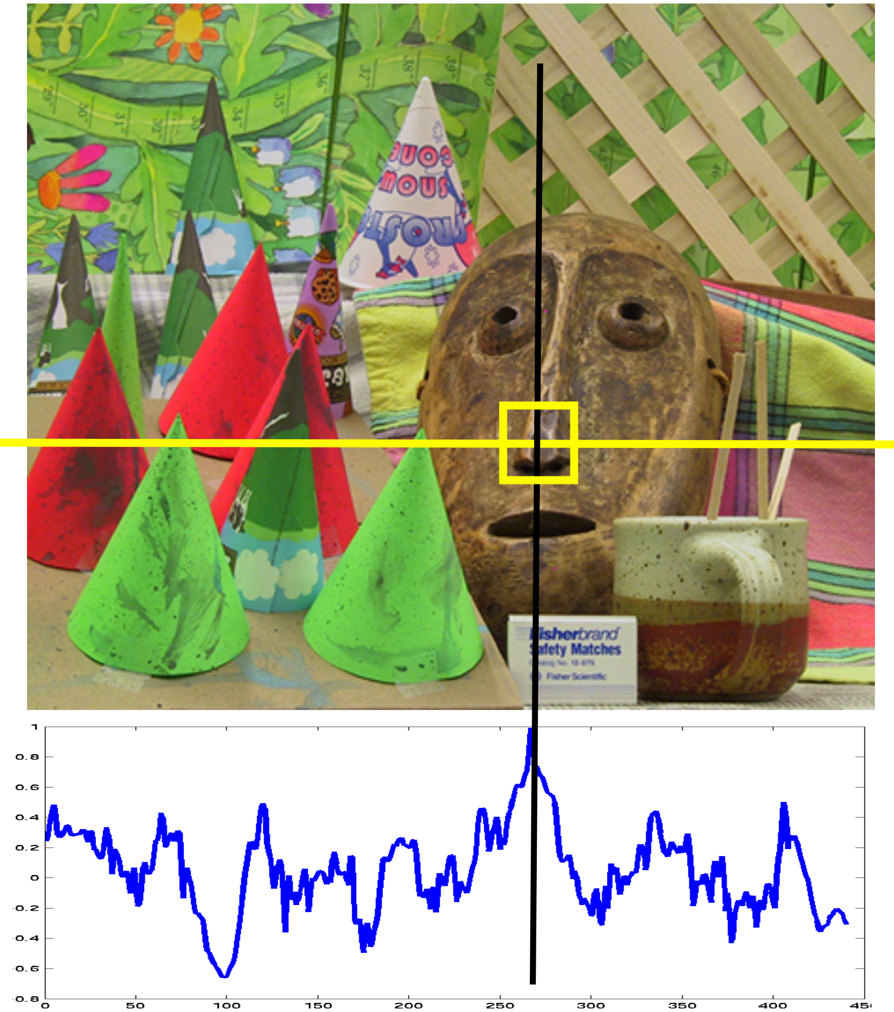
Compare with patches along same row in right image

Match Score Values

Matching using Epipolar Lines

Left Image

Right Image

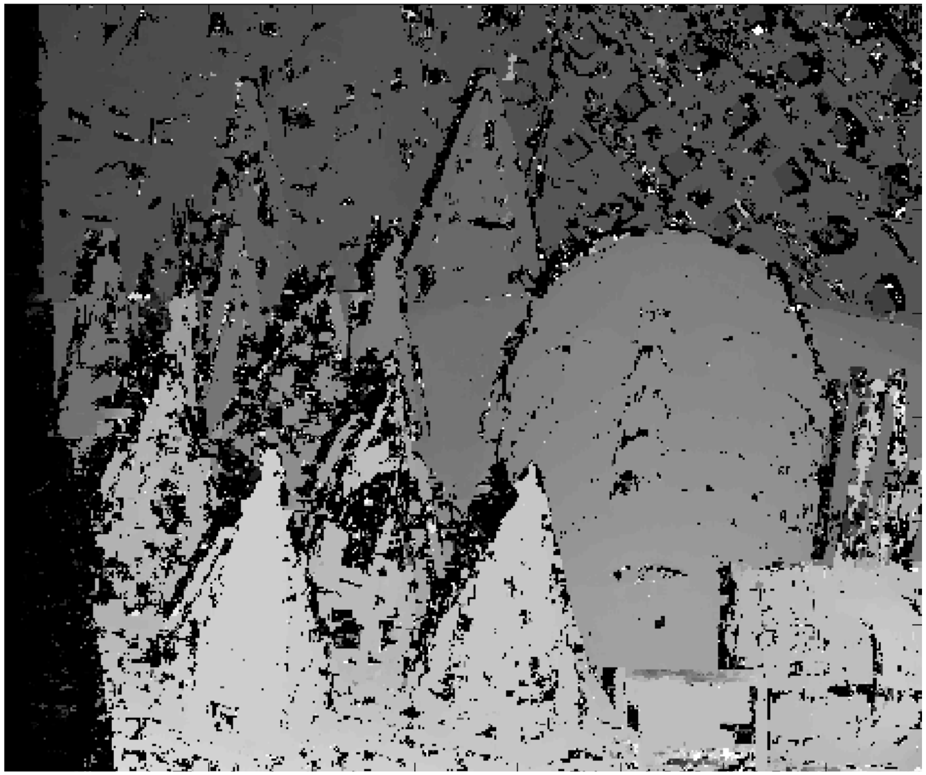


Select patch with highest
match score.

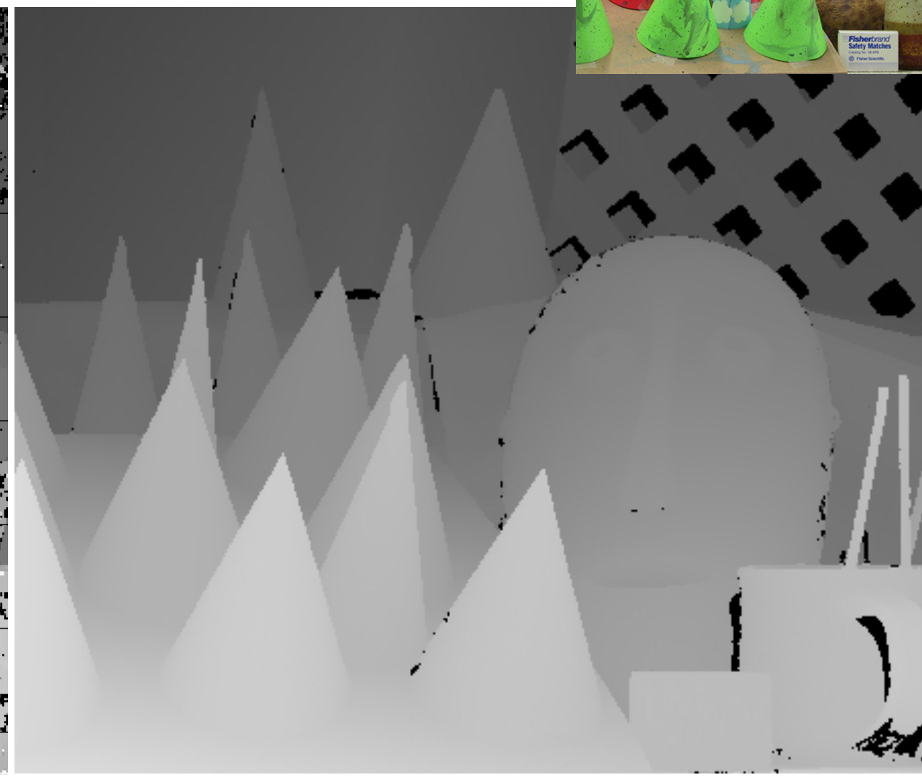
Repeat for all pixels in
left image.

Match Score Values

Example: 5x5 windows NCC match score



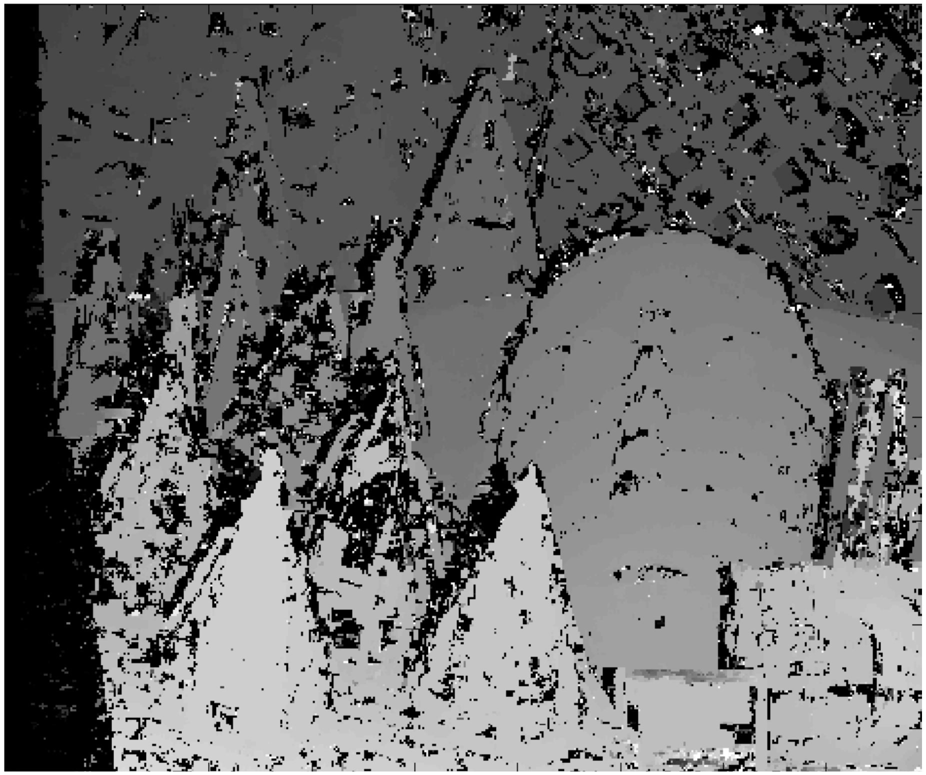
Computed disparities



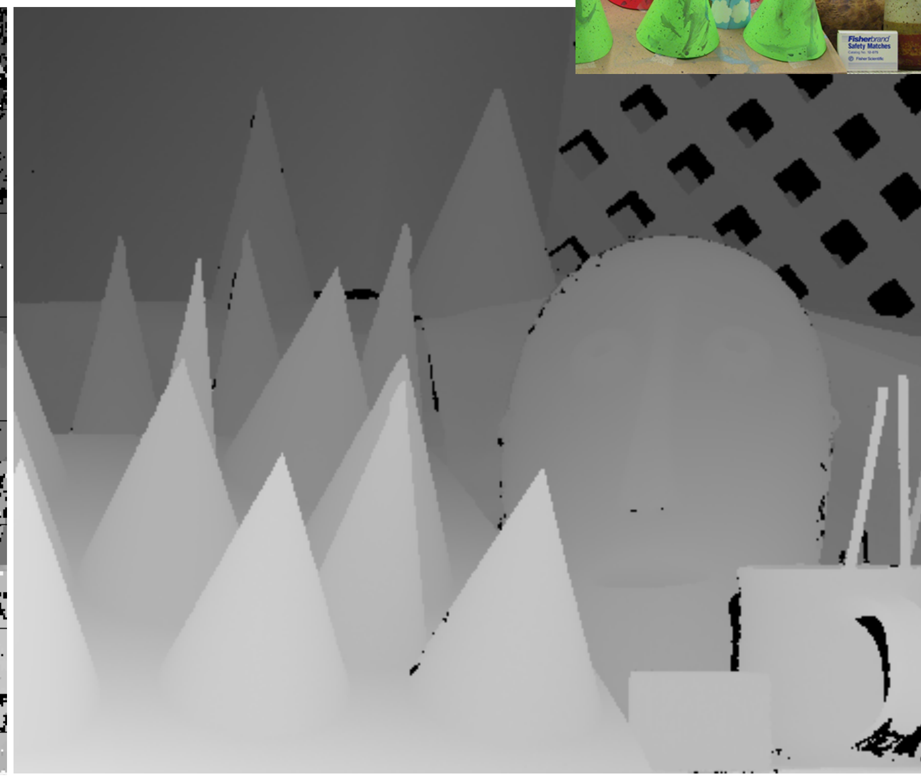
Ground truth

Black pixels: bad disparity values, or no matching patch in right image

Example: 5x5 windows NCC match score



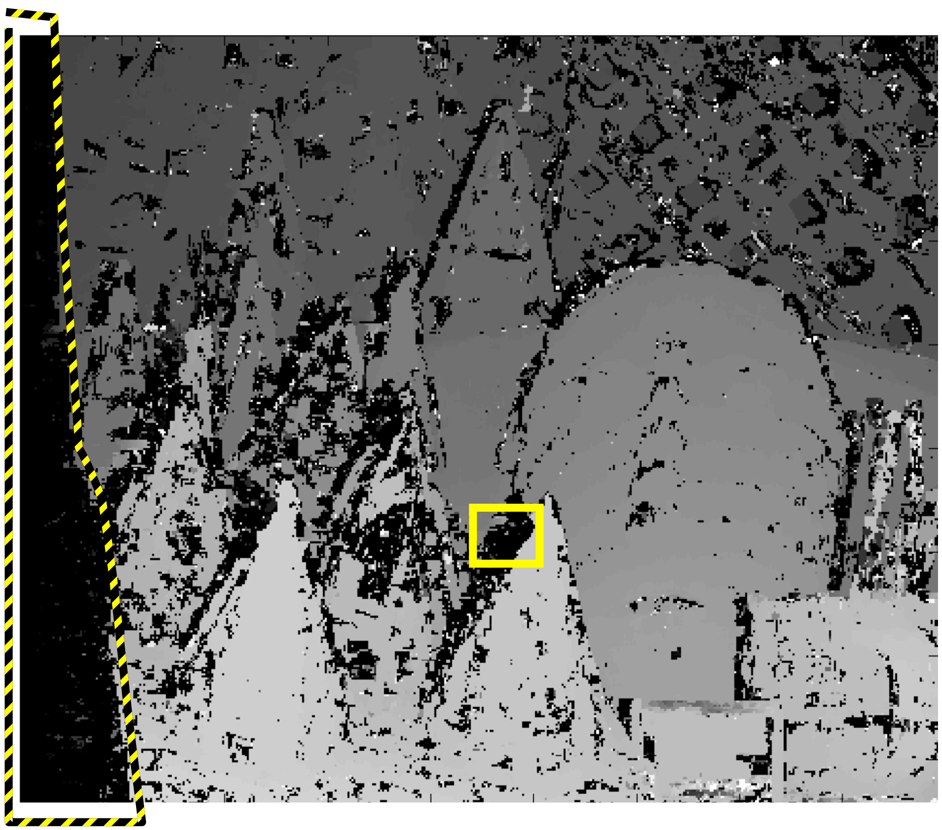
Computed disparities

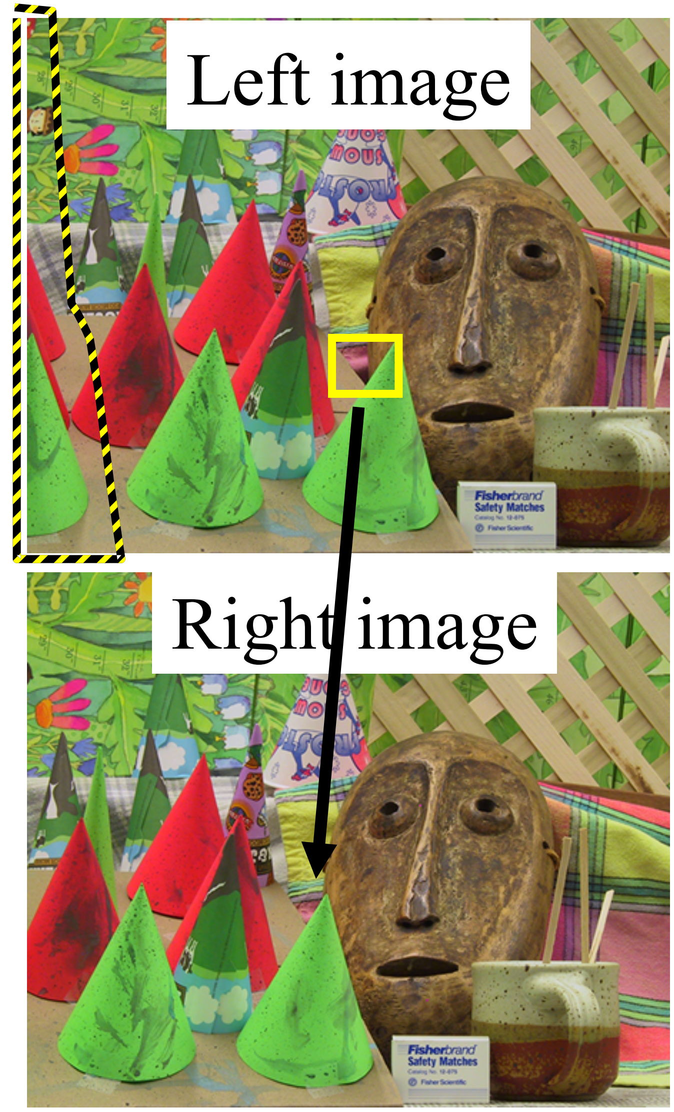


Ground truth

Black pixels: bad disparity values, or no matching patch in right image

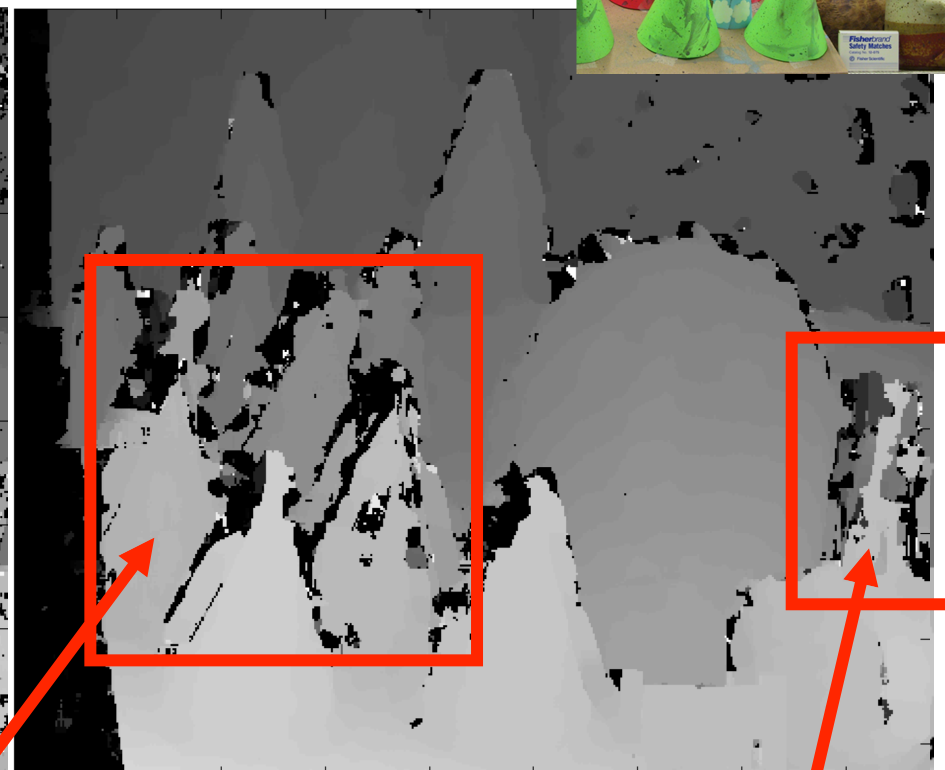
Occlusions: No matches





Effects of Patch Size

5x5 patches



11x11patches

Smoother in some areas

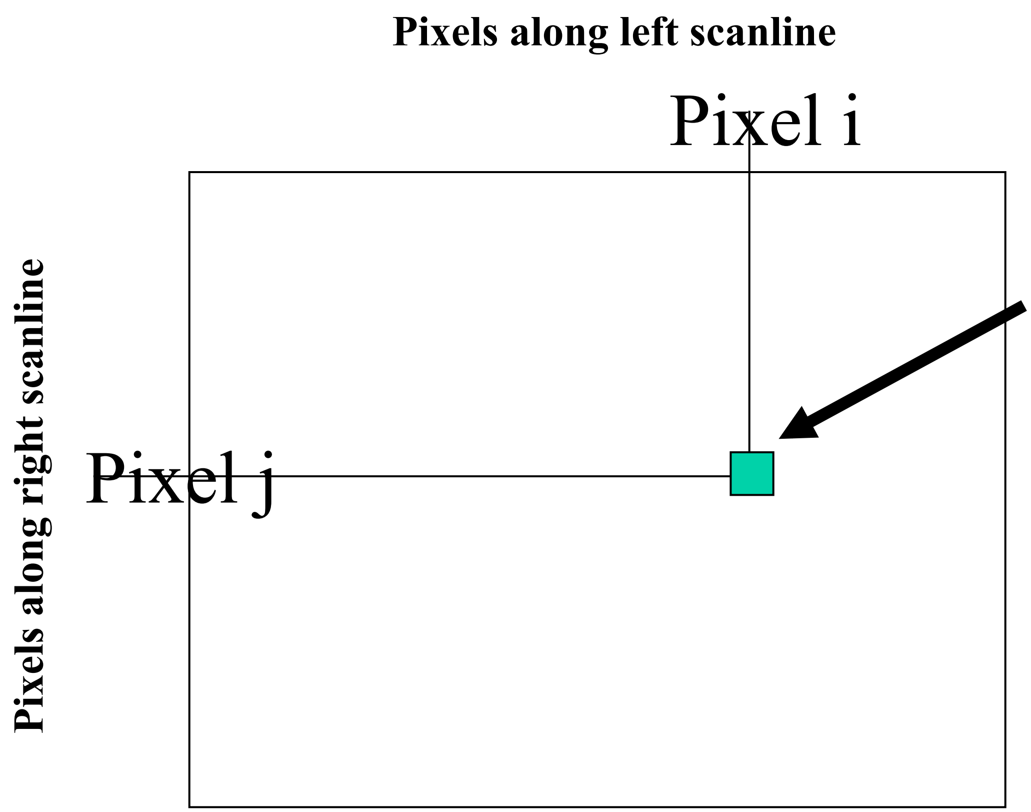
Loss of finer details

Limitations

- So far, each left image patch has been matched independently along the right epipolar line.
- This ignores some obvious constraints, namely that surfaces in the world tend to be smooth.
- We would like to at least enforce some consistency among matches in the same row (scanline).

Disparity Space Image

First we introduce the concept of DSI.
The DSI for one row represents pairwise match scores between patches along that row in the left and right image.

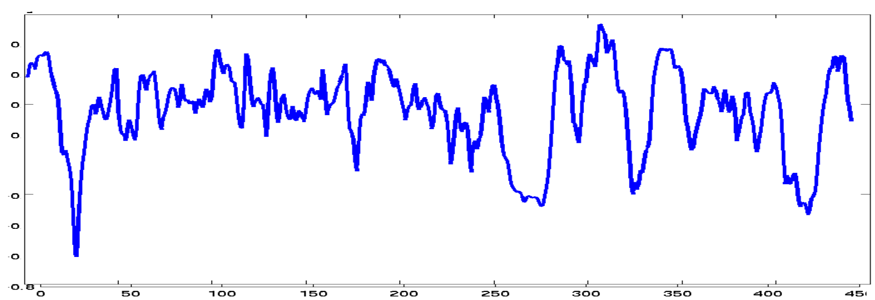


C(i,j) = Match score for patch centered at left pixel i with patch centered at right pixel j.

Disparity Space Image (DSI)

Left Image

Right Image



Dissimilarity Values

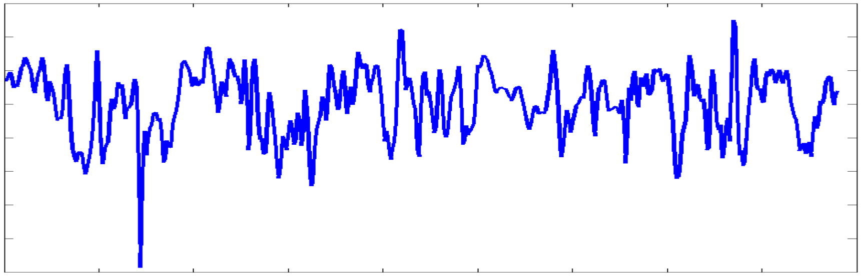
(1-NCC) or SSD

Robert Collins
CMPEN454

Disparity Space Image (DSI)

Left Image

Right Image

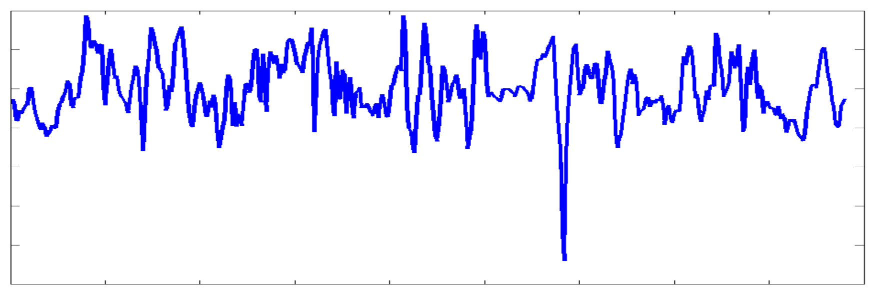


Dissimilarity Values
(1-NCC) or SSD

Disparity Space Image (DSI)

Left Image

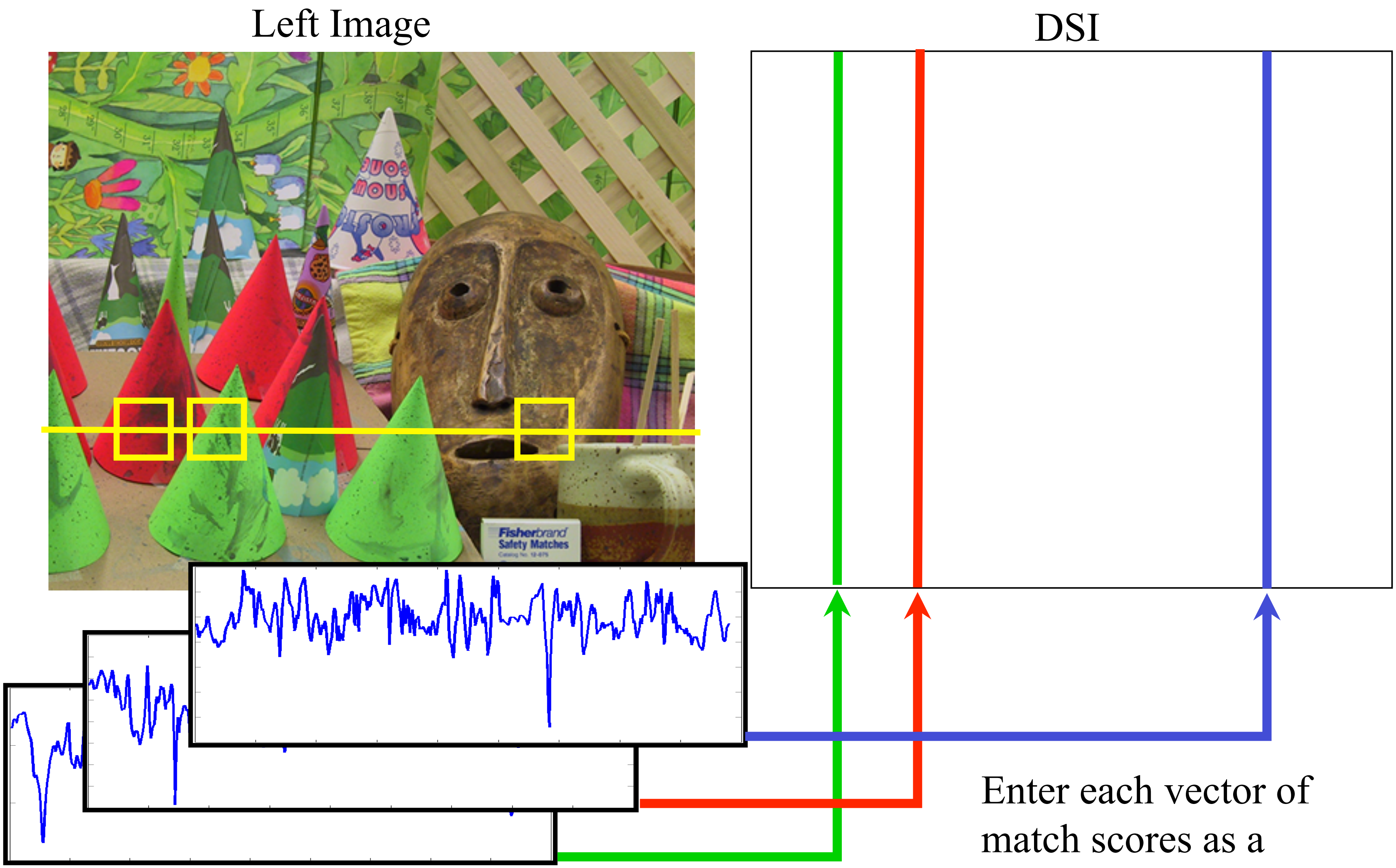
Right Image



Dissimilarity Values
(1-NCC) or SSD

Robert Collins
CMPEN454

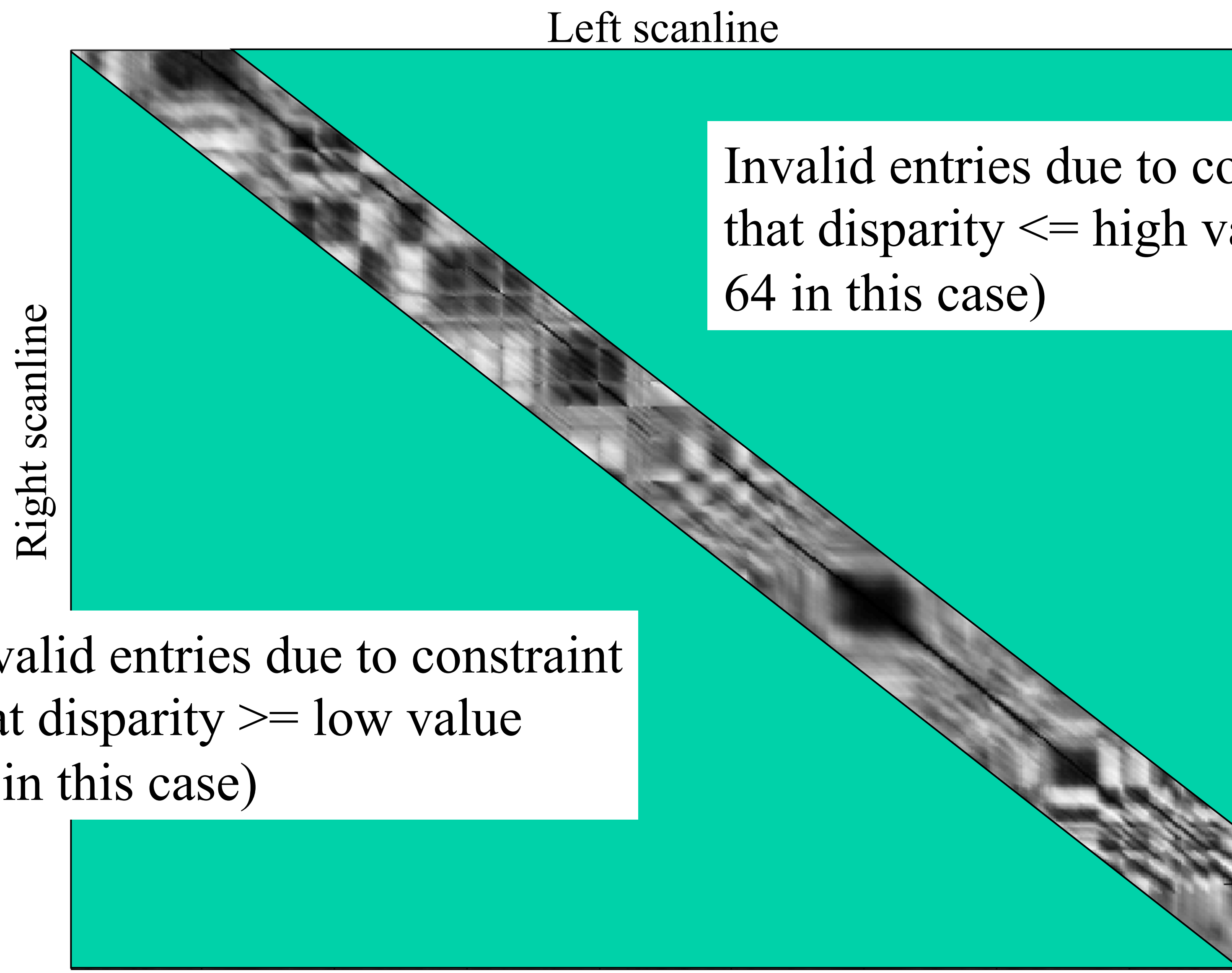
Disparity Space Image (DSI)



Dissimilarity Values

Enter each vector of match scores as a column in the DSI

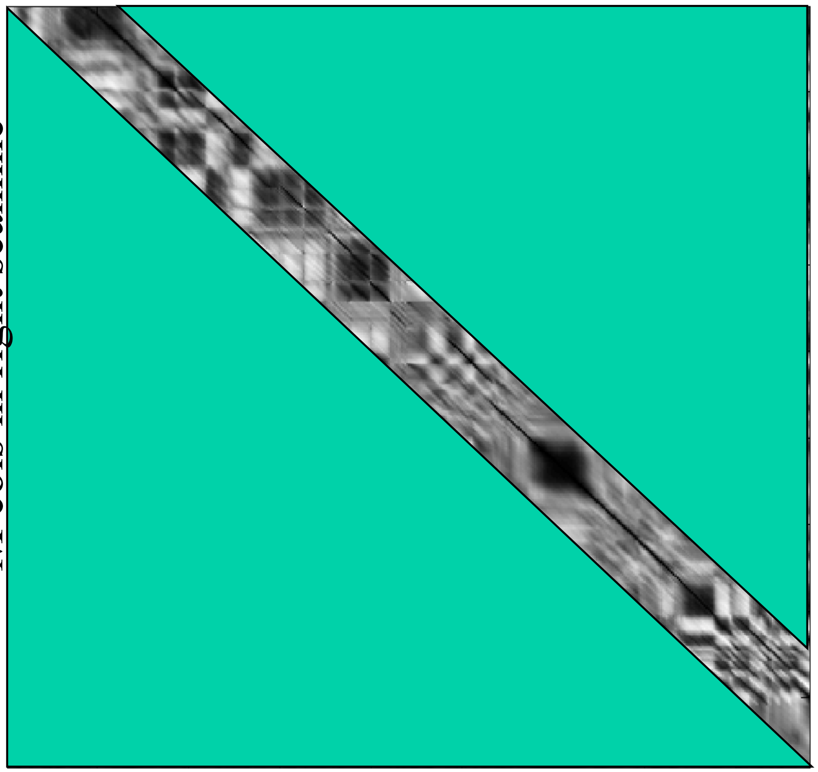
Disparity Space Image



Disparity Space Image

N cols in left scanline

M cols in right scanline

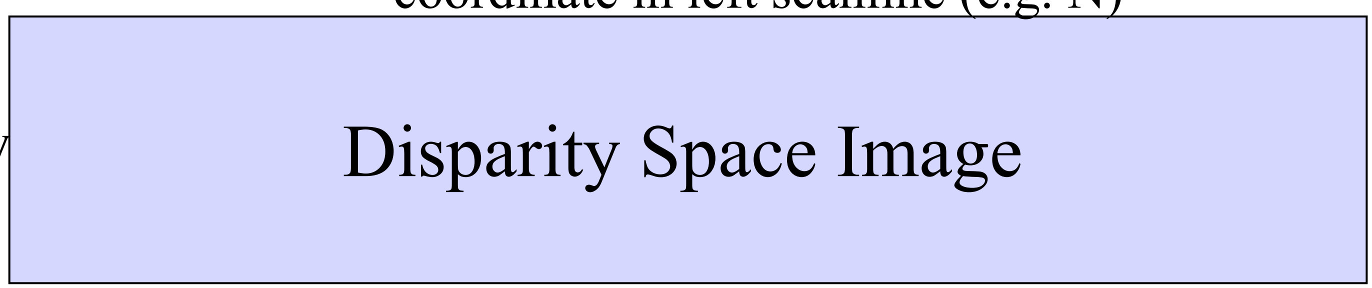


If we rearrange the diagonal band of valid values into a rectangular array (in this case of size 64 x N), that is what is traditionally known as the DSI

However, we're going to keep the full image around, including invalid values (I think it is easier to understand the pixel coordinates involved)

coordinate in left scanline (e.g. N)

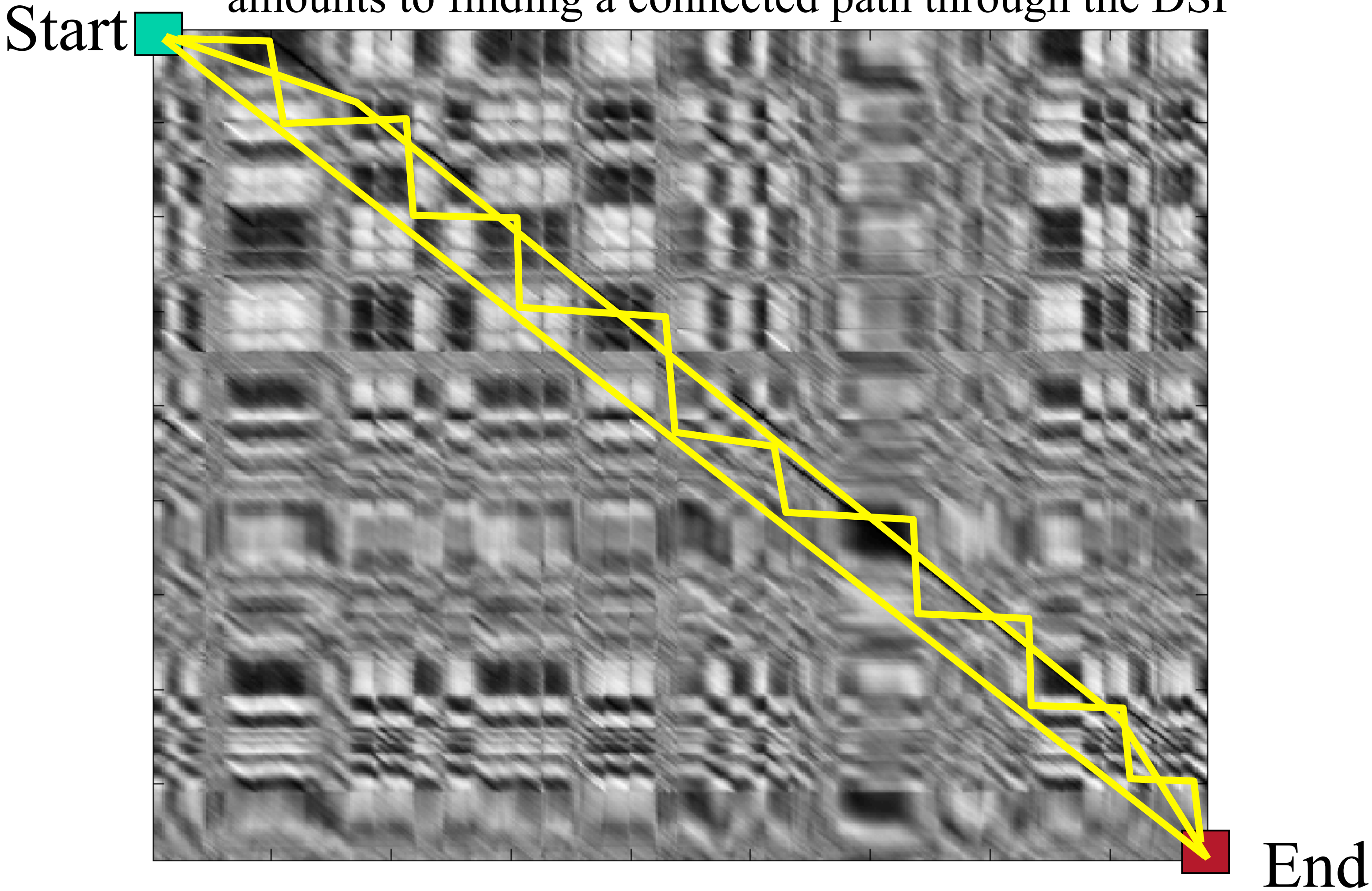
Disparity (e.g. 64)



Robert Collins
CMPEN454

DSI and Scanline Consistency

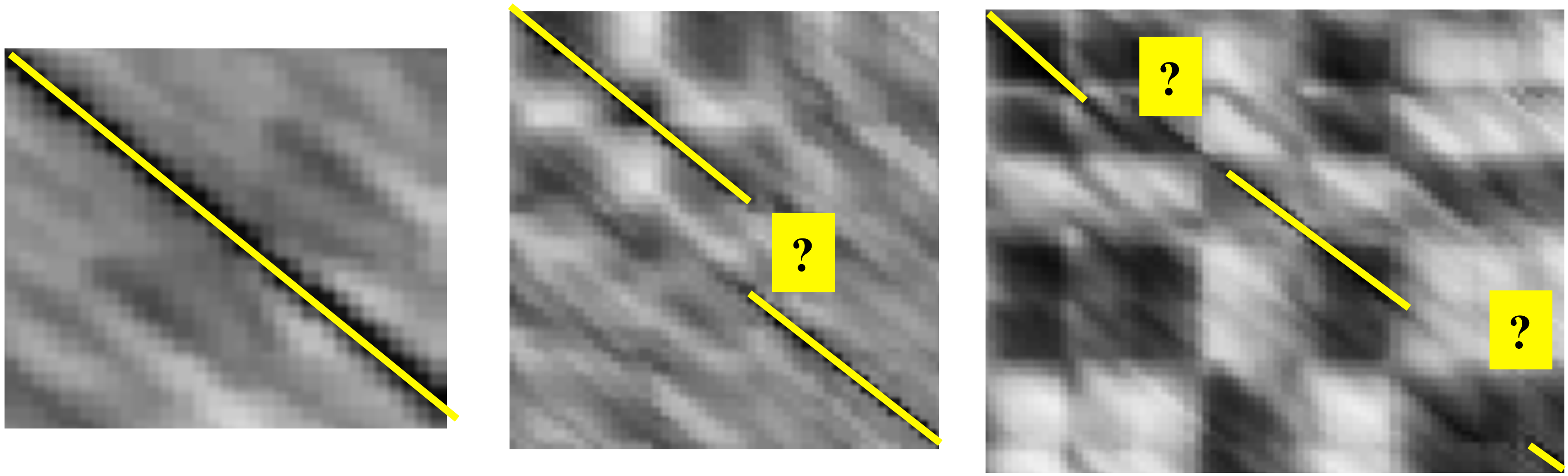
Assigning disparities to all pixels in left scanline now amounts to finding a connected path through the DSI



Lowest Cost Path

We would like to choose the "best" path.

Want one with lowest "cost" (Lowest sum of dissimilarity scores along the path)

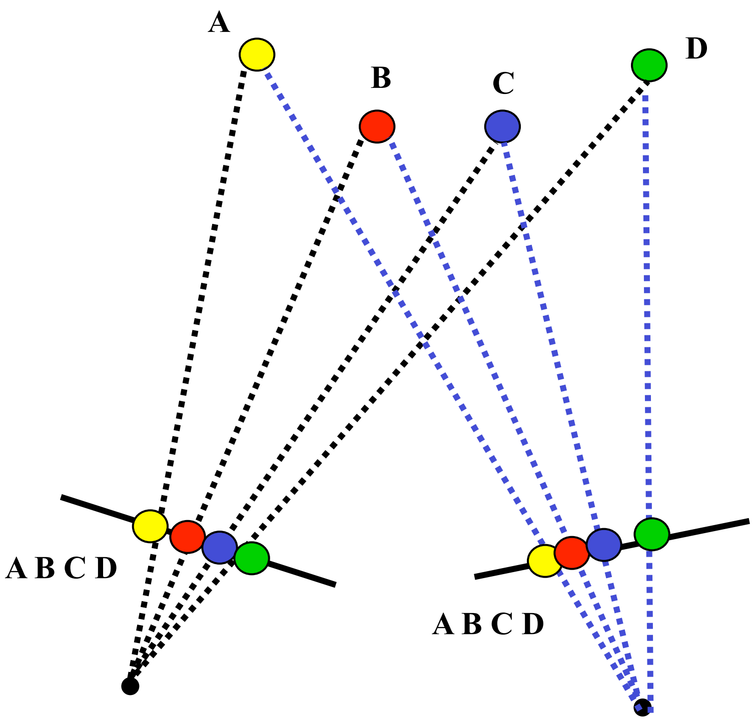


Other Constraints on Path

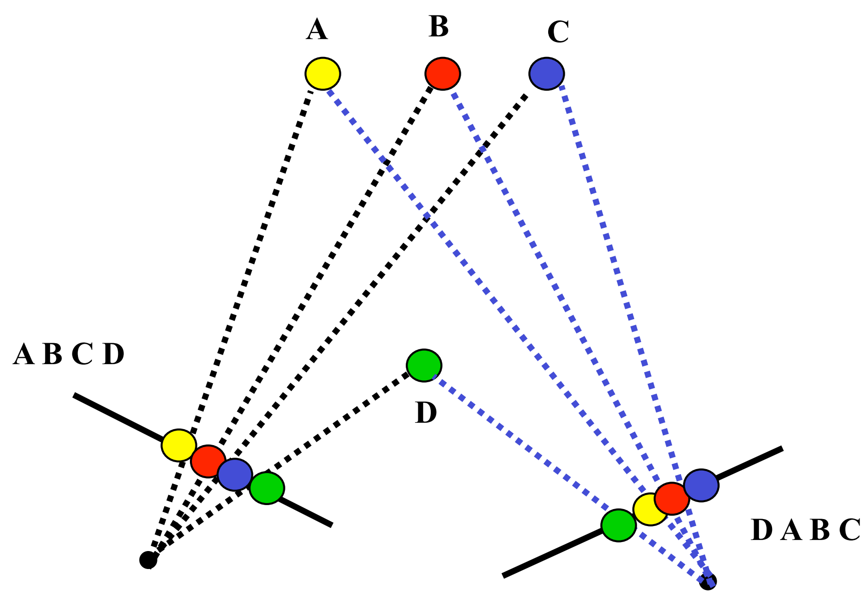
It is common to impose an ordering constraint on the path. Intuitively, the path is not allowed to “double back” on itself.

Robert Collins
CMPEN454

Ordering Constraint

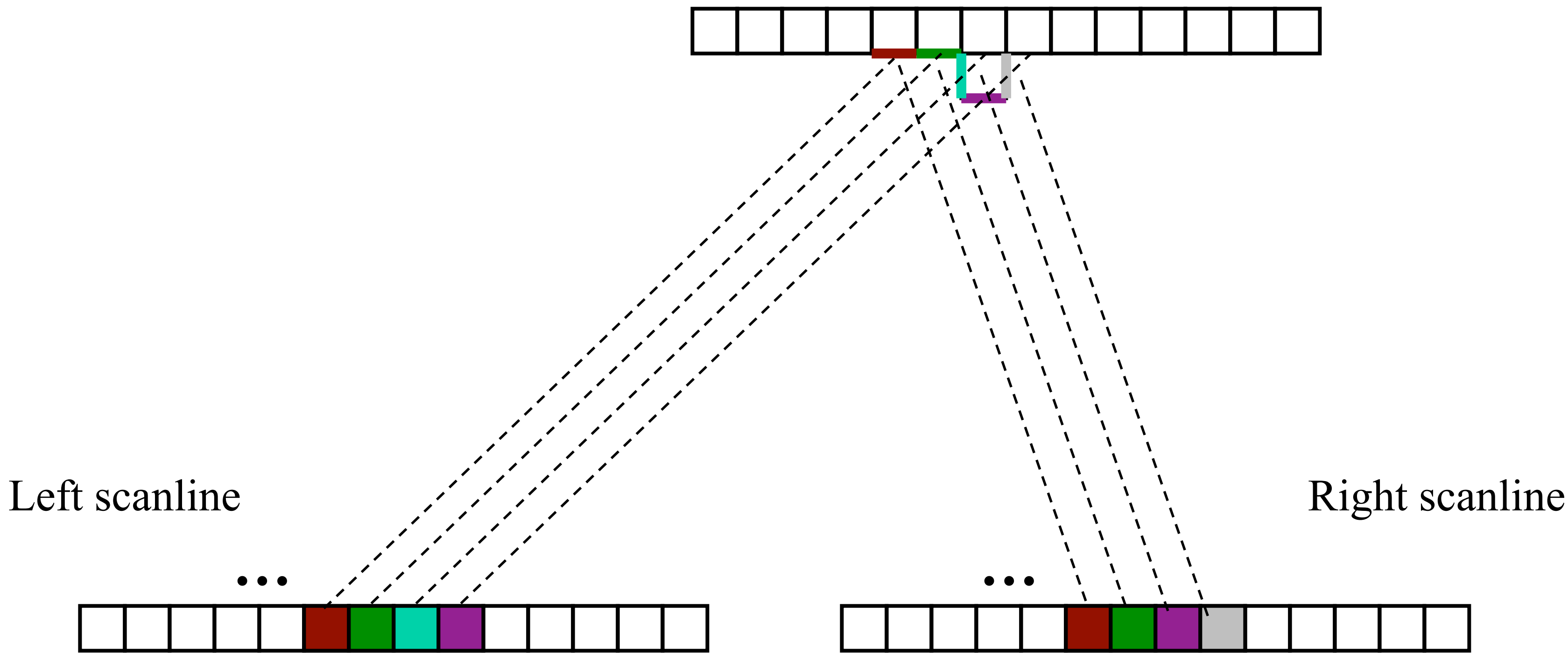


Ordering constraint…



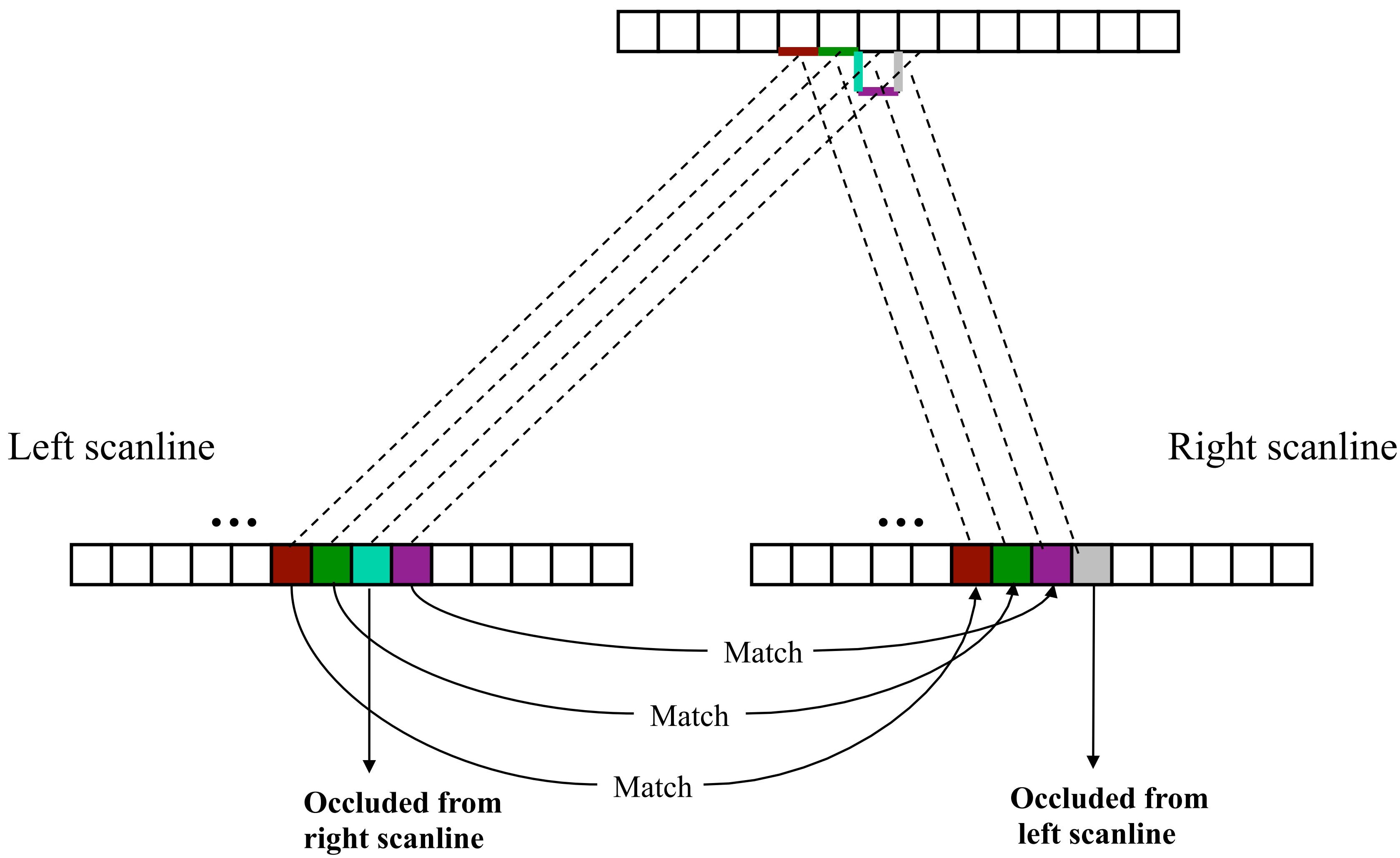
…and its failure

Occlusions Can Occur



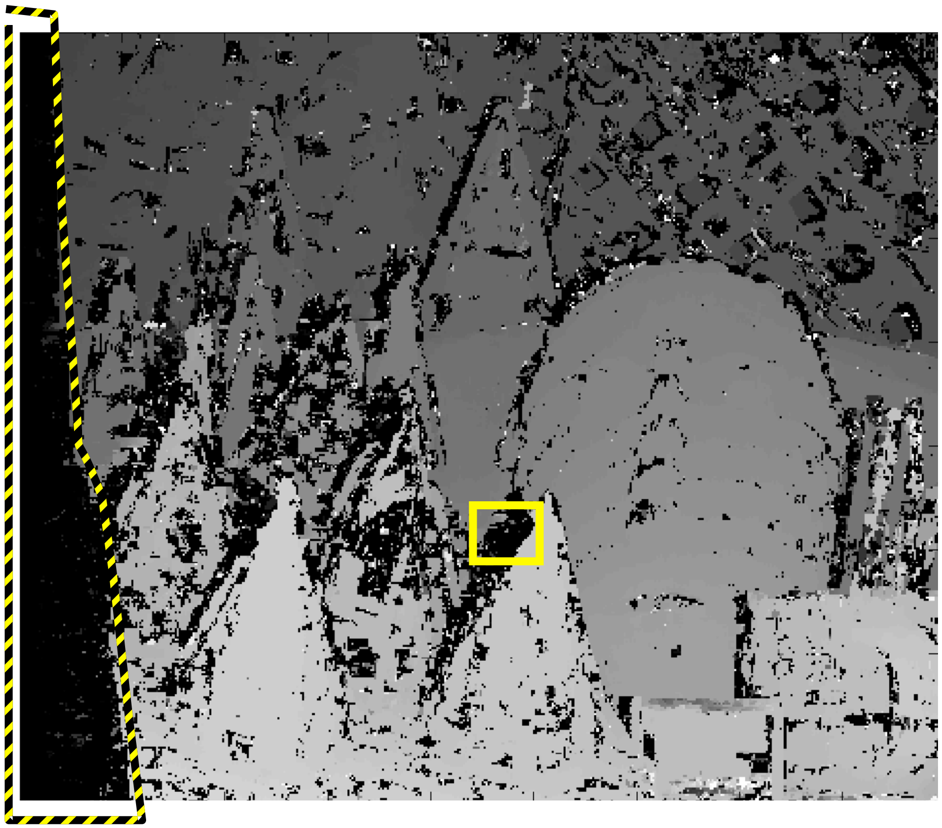
Robert Collins
CMPEN454

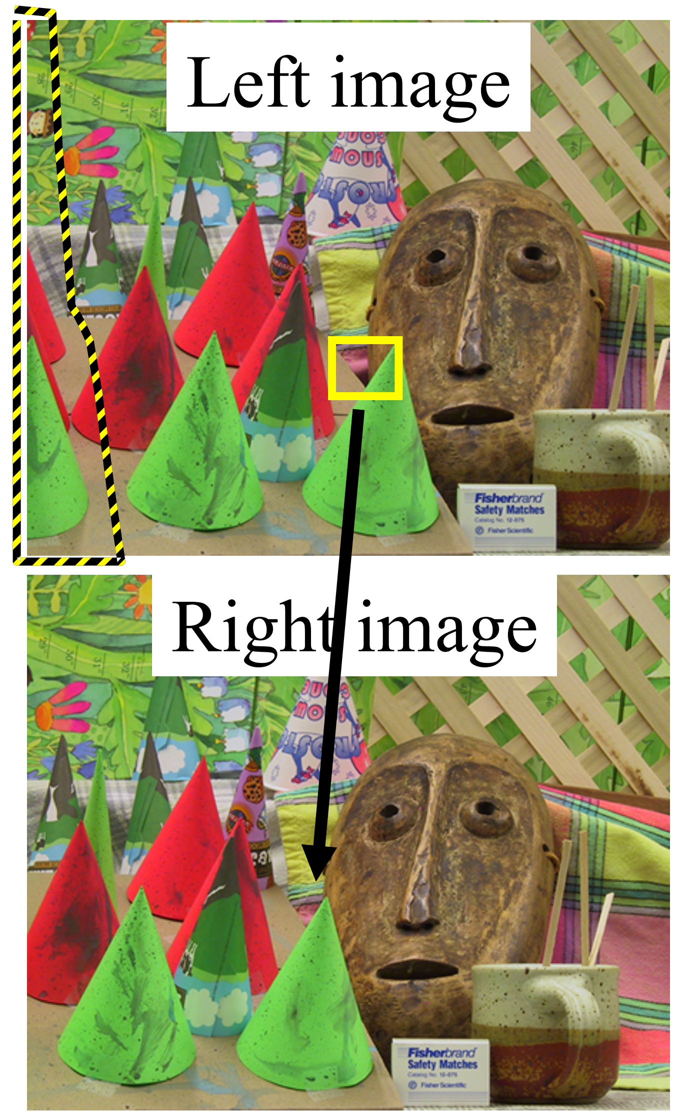
Occlusions Can Occur



However, note that the order of matching patches is preserved.

Occlusions: No matches





An Optimal Scanline Strategy

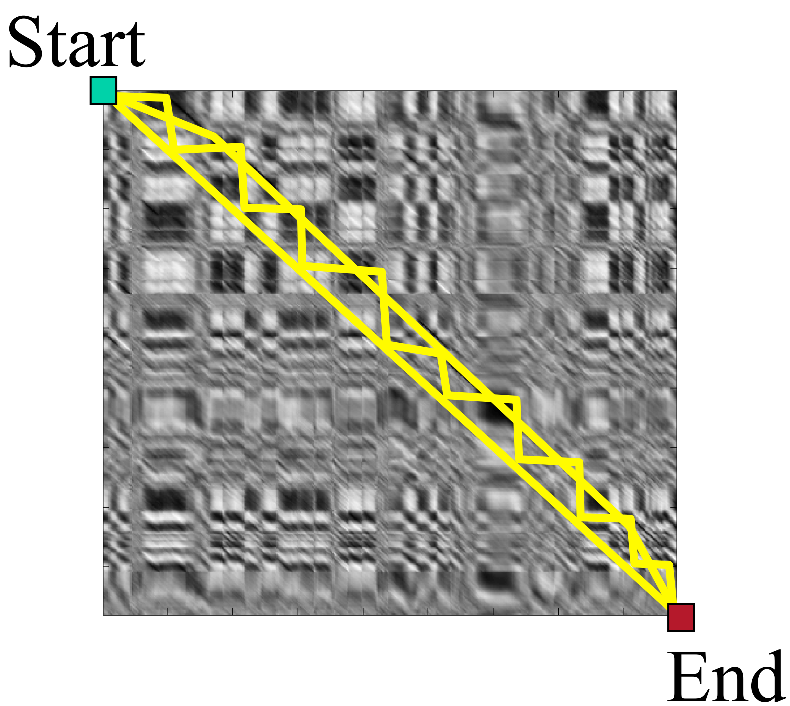
- We want to find best connected path, taking into account ordering constraint and the possibility of occlusions.

Practical algorithm:

Cox, Hingorani, Rao, Maggs, "A Maximum Likelihood Stereo Algorithm," Computer Vision and Image Understanding, Vol 63(3), May 1996, pp.542-567.

See also Ohta & Kanade '85

Cox et.al. Stereo Matching



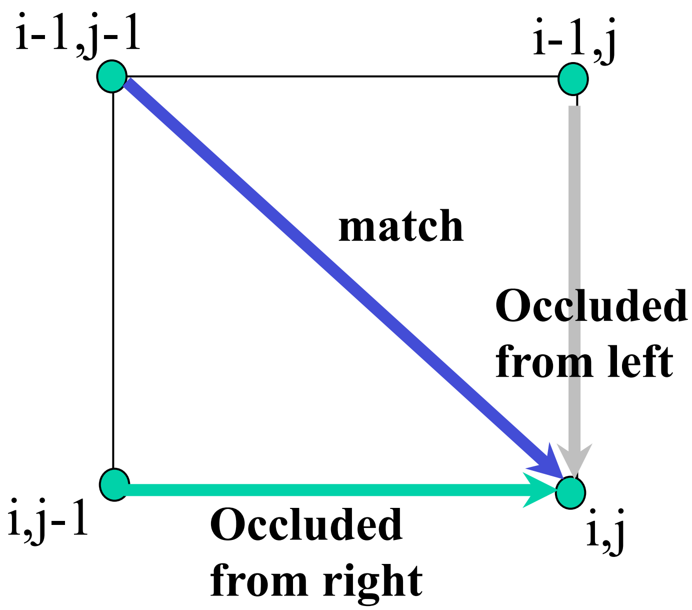
Recap: want to find lowest cost path from upper left to lower right of DSI image.

At each point on the path we have three choices: step left, step down, step diagonally.

Each choice has a well-defined cost associated with it.

This problem just screams out for Dynamic Programming!
(which, indeed, is how Cox et.al. solve the problem)

Cox et.al. Stereo Matching



Three cases:

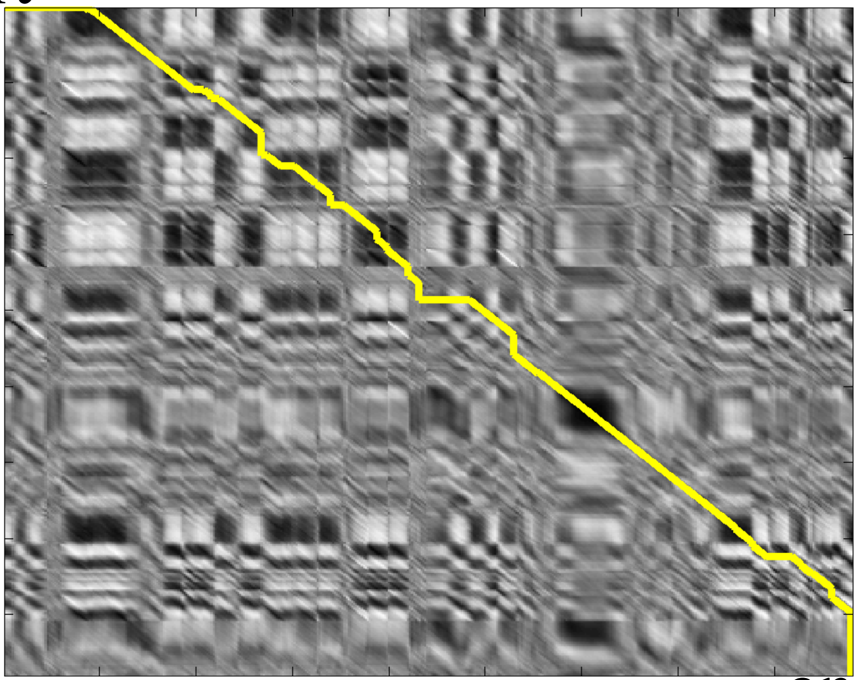
- Matching patches. Cost = dissimilarity score
- Occluded from right. Cost is some constant value.
- Occluded from left. Cost is some constant value.

```
C(i,j)= min([ C(i-1,j-1) + dissimilarity(i,j)
              C(i-1,j) + occlusionConstant,
              C(i,j-1) + occlusionConstant]);
```

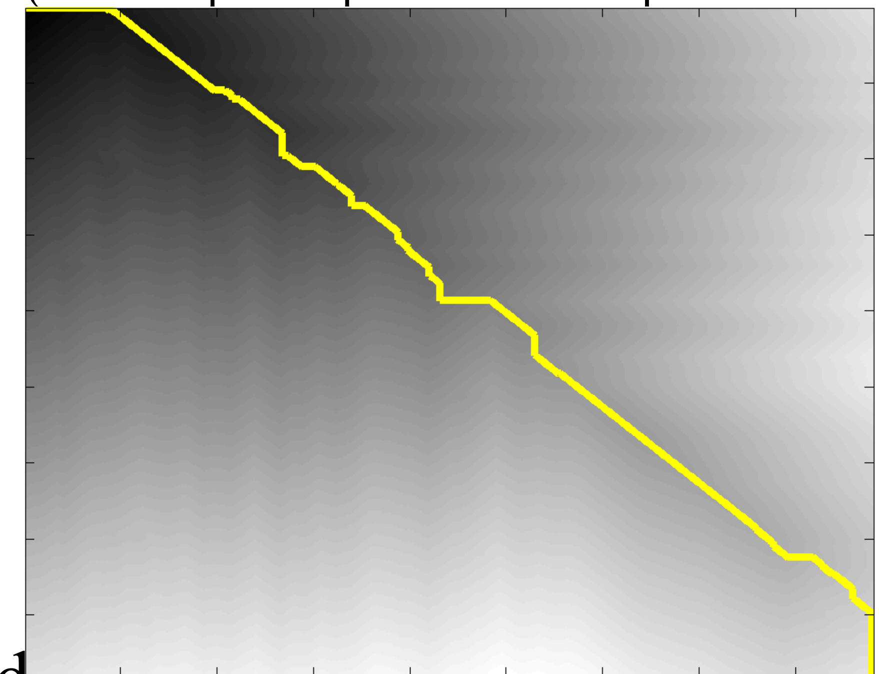
Real Scanline Example

start

DSI



DP cost matrix
(cost of optimal path from each point to END)



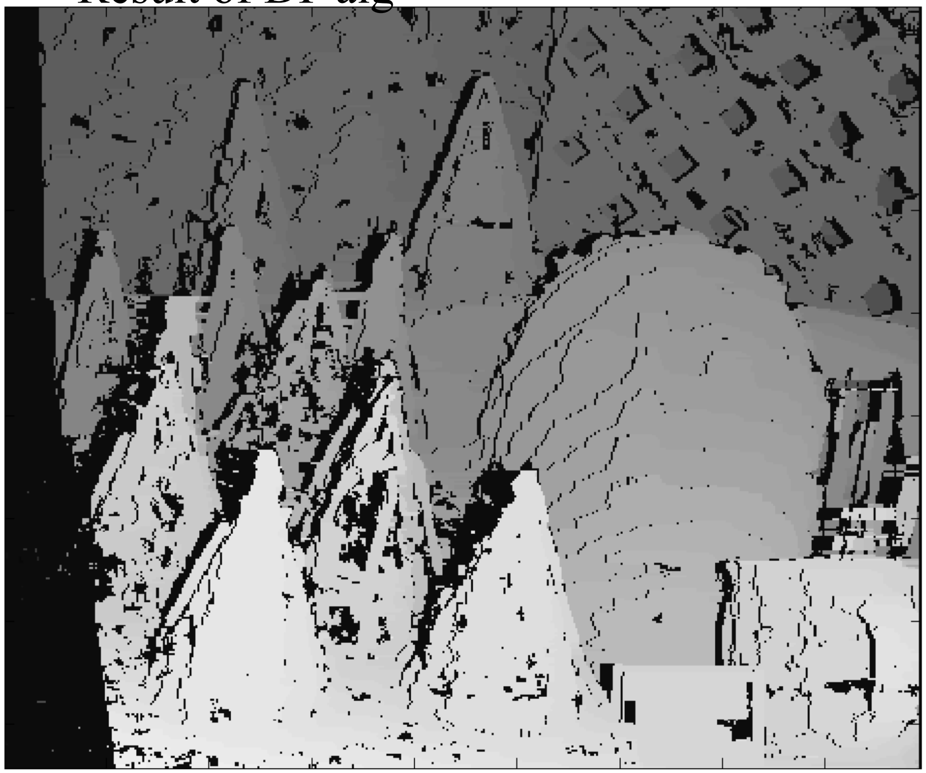
end

Every pixel in left column now is marked with either a disparity value, or an occlusion label.

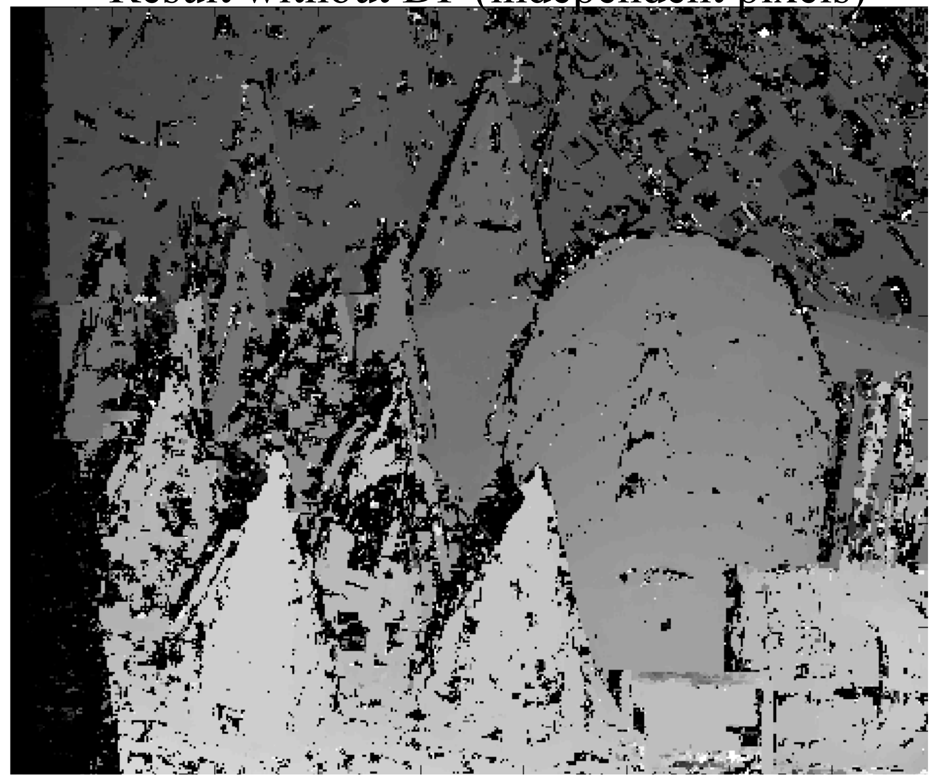
Proceed for every scanline in left image.

Example

Result of DP alg



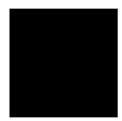
Result without DP (independent pixels)



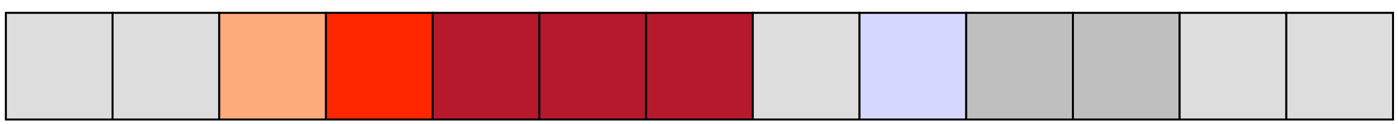
Result of DP alg. Black pixels = occluded.

Occlusion Filling

Simple trick for filling in gaps caused by occlusion.

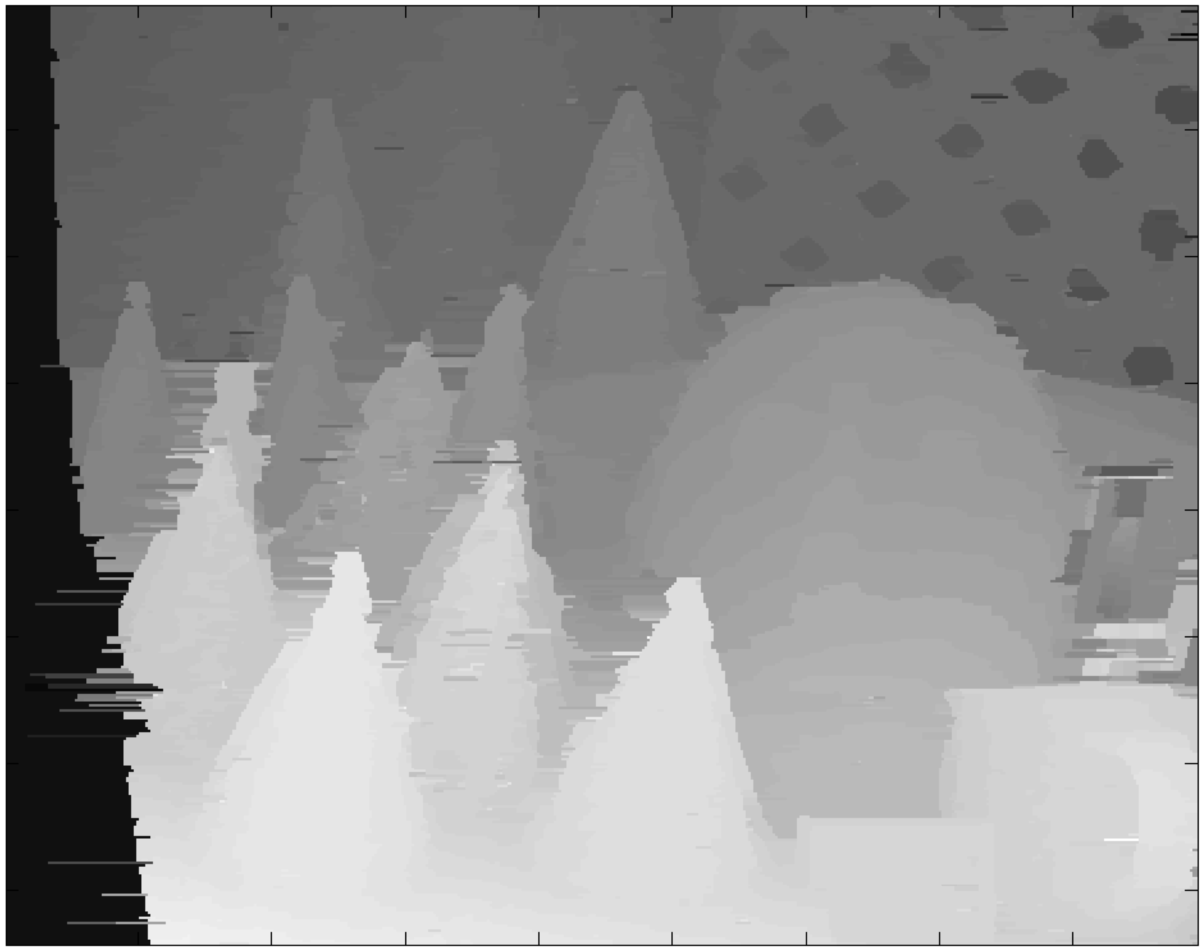
 = left occluded

Fill in left occluded pixels with value from the nearest valid pixel preceding it in the scanline.



Similarly, for right occluded, look for valid pixel to the right.

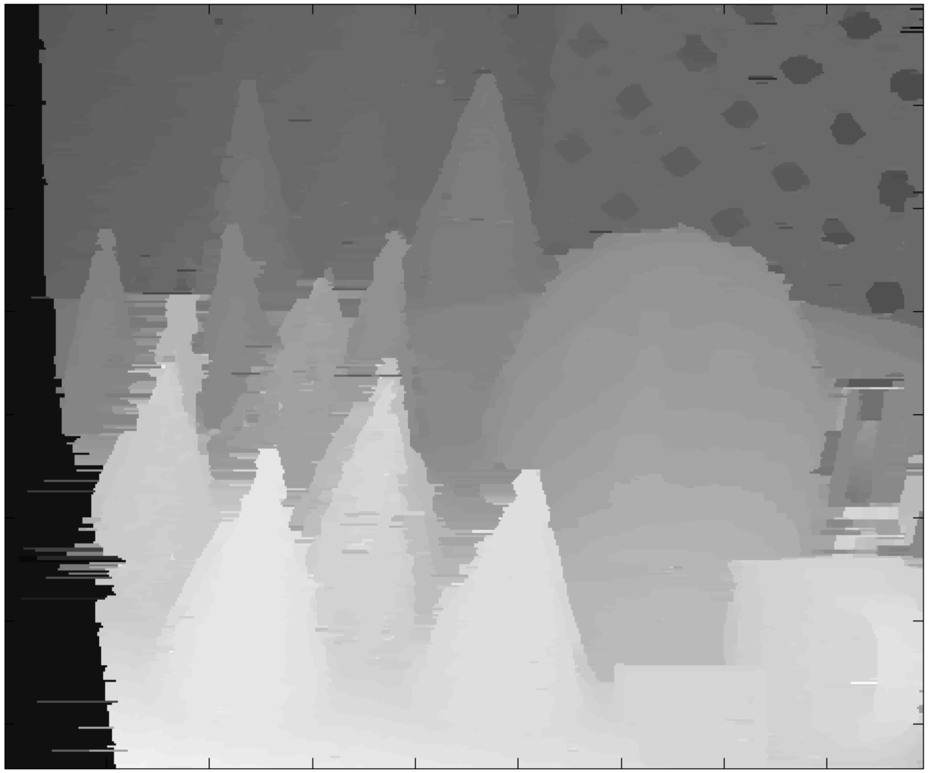
Example



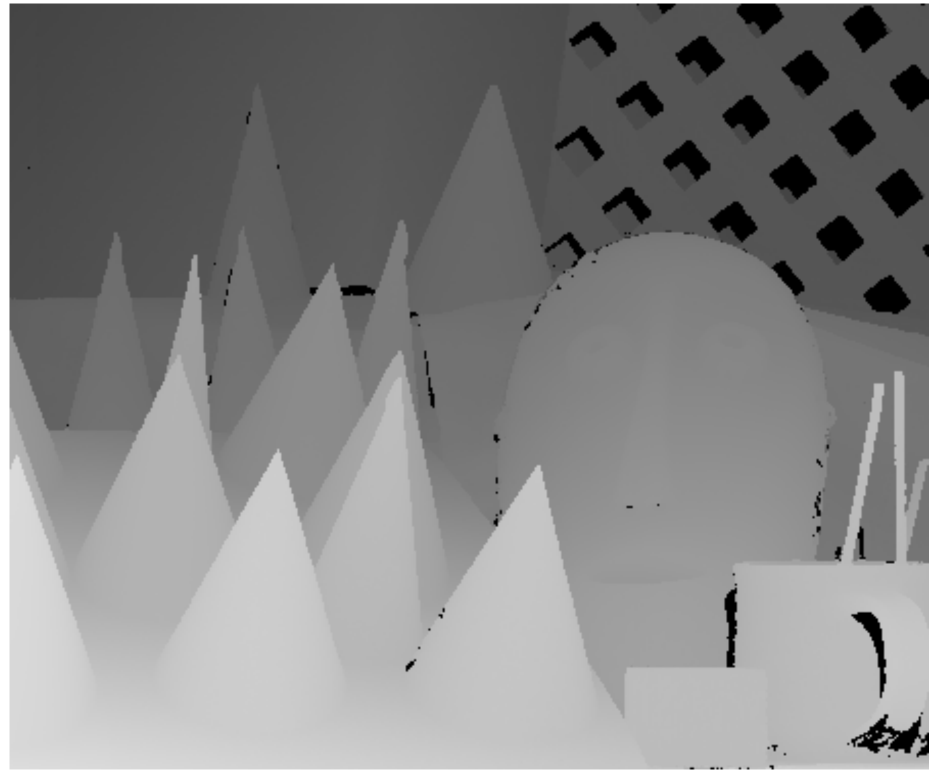
Result of DP alg with occlusion filling.

Example

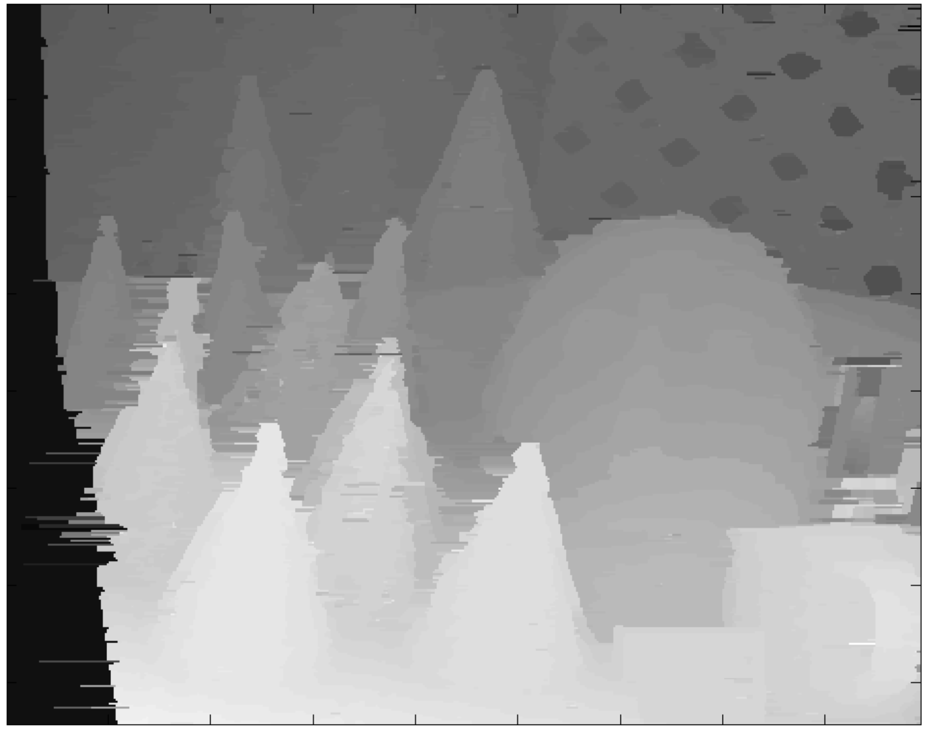
Result of DP alg with occlusion filling.



Ground truth

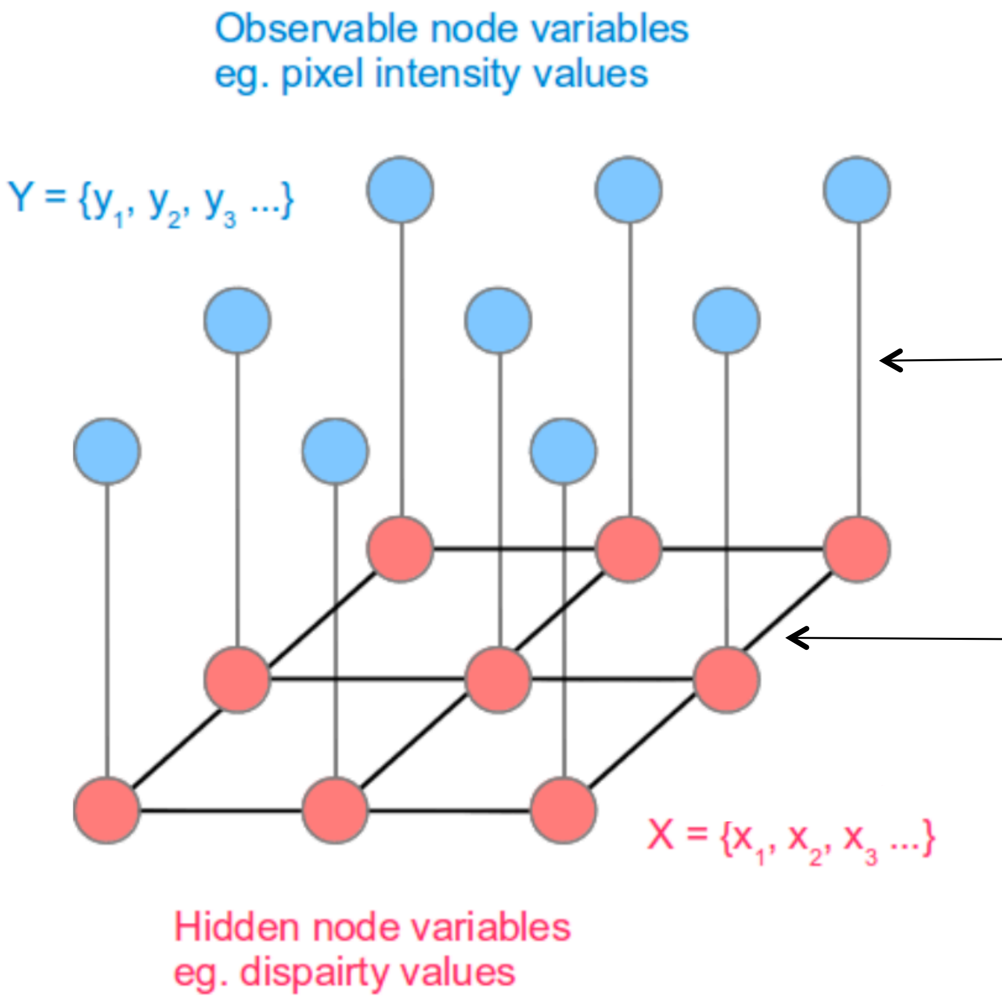


Scanline Algorithm Limitations



- Streaky results; each line being matched independently
- No obvious way to generalize dynamic programming from 1D scanline to 2D grid

Markov Random Field Stereo



P(image data | disparity)

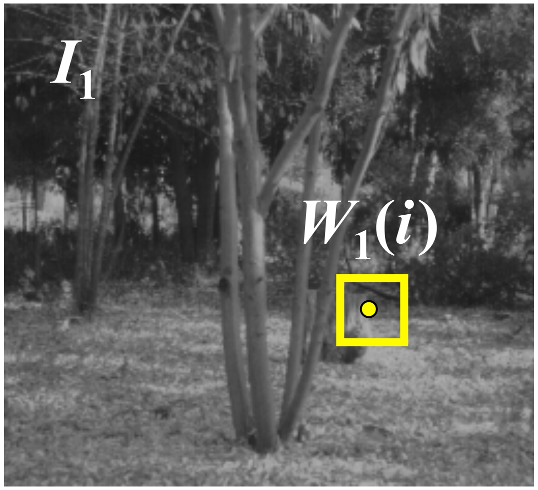
e.g. matching pixels should have similar intensity values

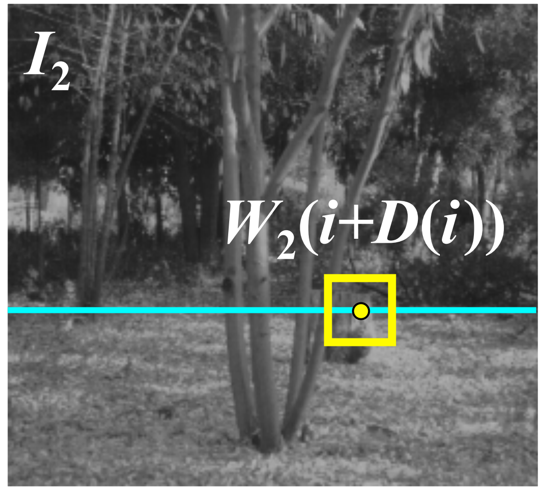
P(disparity)

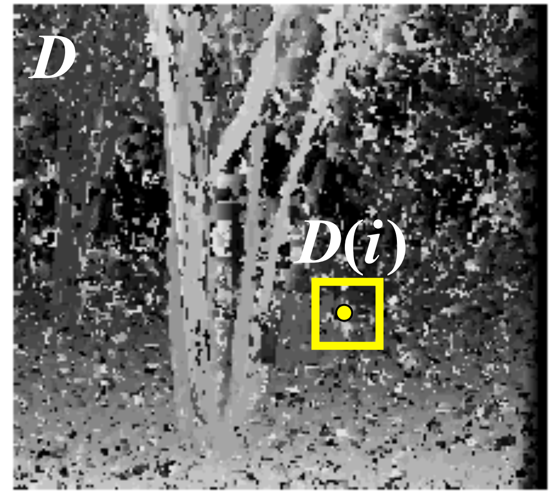
e.g. adjacent pixels should have same or similar disparity (depth)

Lazebnik, UNC

Stereo matching as MRF energy minimization







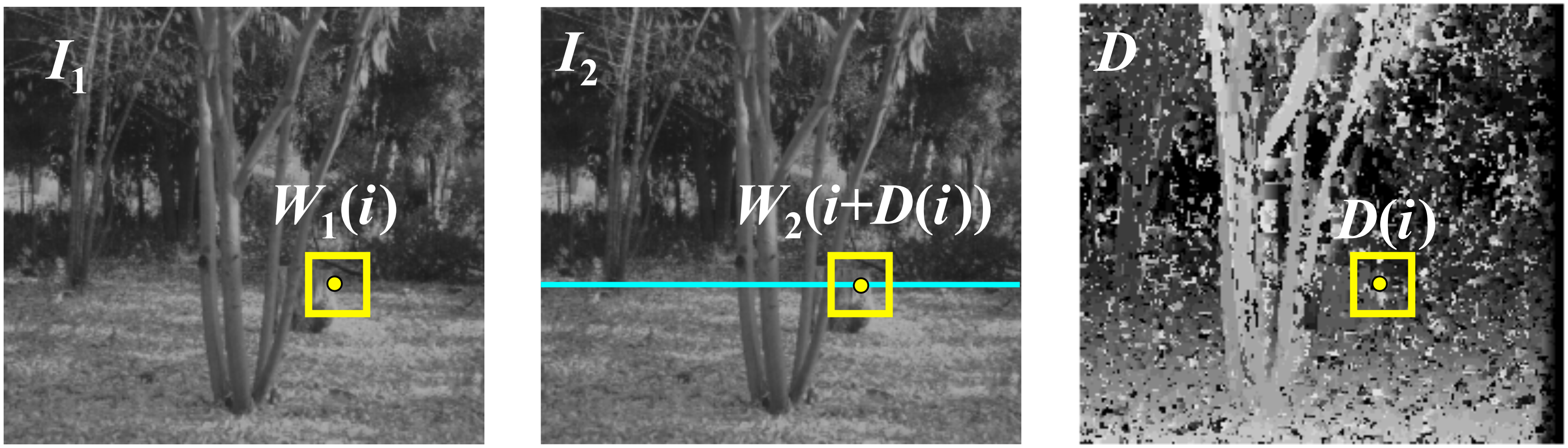
- Probabilistic interpretation: we want to find a Maximum A Posteriori (MAP) estimate of disparity image *D*:

$$P(D \mid I_1, I_2) \propto P(I_1, I_2 \mid D) P(D)$$

$$-\log P(D \mid I_1, I_2) \propto -\log P(I_1, I_2 \mid D) - \log P(D)$$

$$\boxed{E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)}$$

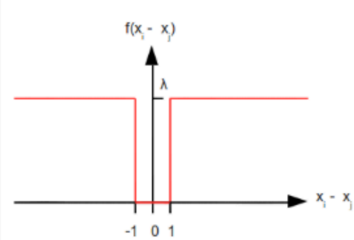
Stereo matching as MRF energy minimization



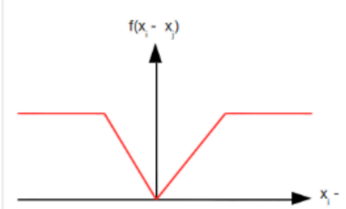
$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

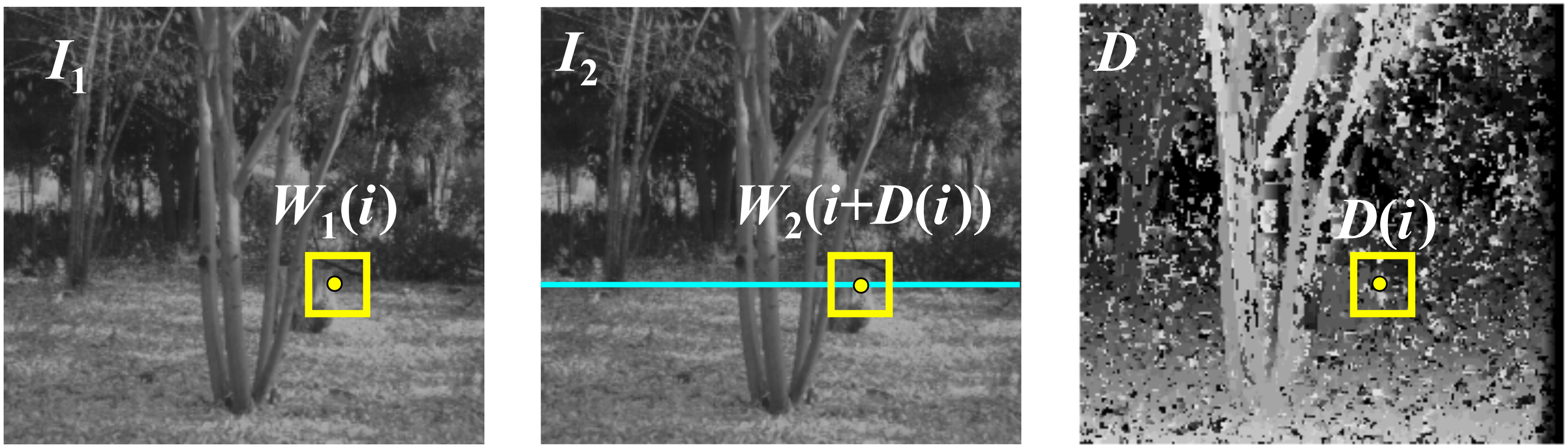


e.g. Potts model



e.g. truncated linear model

Stereo matching as MRF energy minimization



$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

- MRF energy functions of this form can be minimized efficiently using an algorithm known as *graph cuts*

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

V. Kolmogorov and R. Zabih,
Computing Visual Correspondence with Occlusions using Graph Cuts, ICCV 2001

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

vision.middlebury.edu

stereo • mview • MRF • flow • color

Stereo

Evaluation • Datasets • Code • Submit

Daniel Scharstein • Richard Szeliski • Heiko Hirschmüller

Welcome to the Middlebury Stereo Vision Page. This website accompanies our taxonomy and comparison of two-frame stereo correspondence algorithms [1]. It contains:

- An on-line evaluation of current algorithms
- Many stereo datasets with ground-truth disparities
- Our stereo correspondence software
- An on-line submission script that allows you to evaluate your stereo algorithm in our framework

How to cite the materials on this website:
We grant permission to use and publish all images and numerical results on this website. If you report performance results, we request that you cite our paper [1]. Instructions on how to cite our datasets are listed on the datasets page. If you want to cite this website, please use the URL "**vision.middlebury.edu/stereo/**".

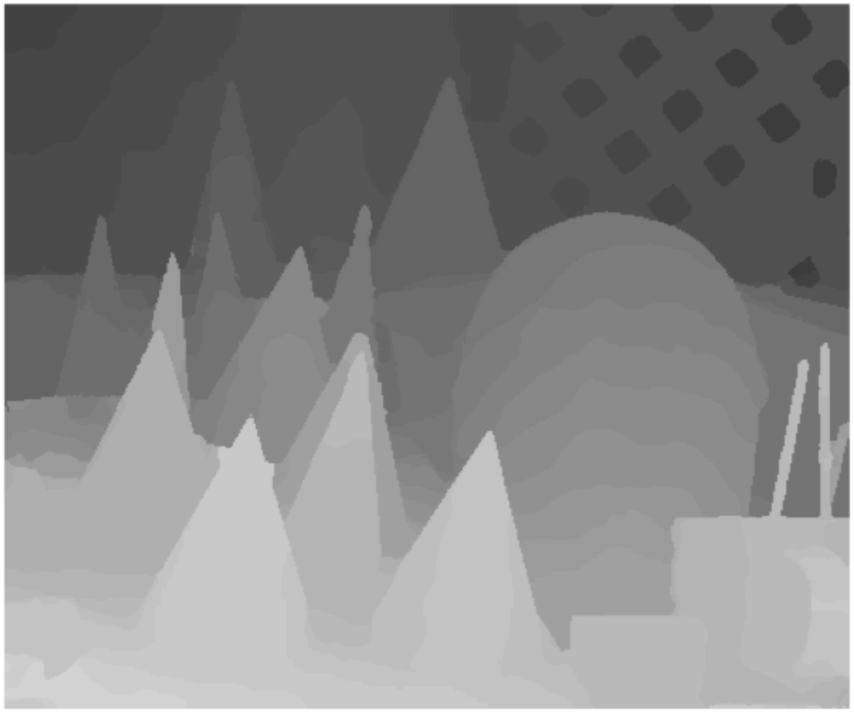
References:

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.
International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.
Microsoft Research Technical Report MSR-TR-2001-81, November 2001.

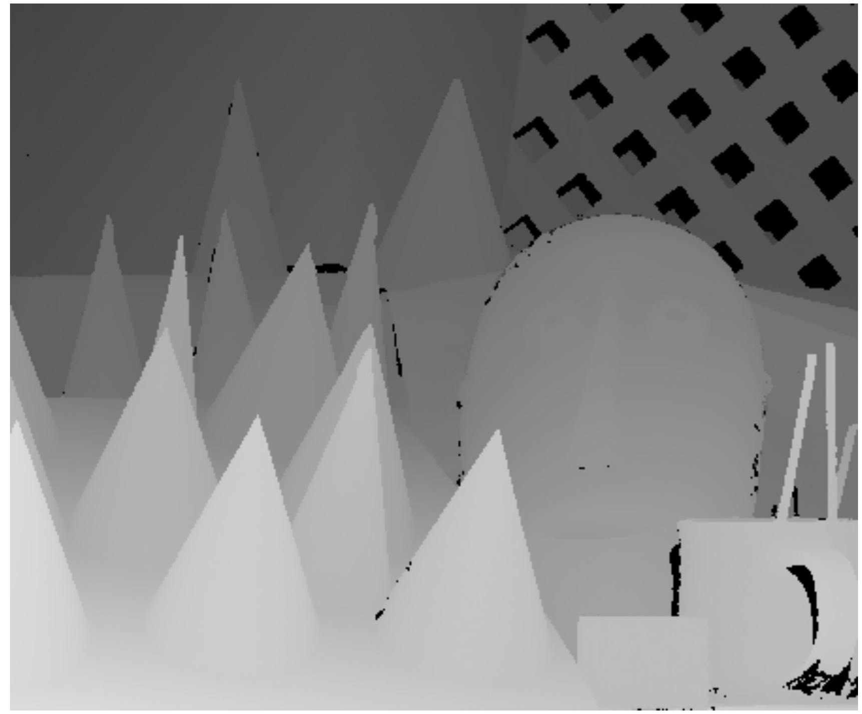
Other online stereo benchmarks and datasets:

- KITTI vision benchmark
- HCI robust vision challenge

State-of-the-Art Results



Algorithm Results



Ground truth

M. Mozerov and J. van Weijer. Accurate stereo matching by two step global optimization. IEEE Transactions on Image Processing 24(3), January 2015.